

# Data Collection and Preparation

---

PRI 21/22 · Information Processing and Retrieval  
M.EIC · Master in Informatics Engineering and Computation

Sérgio Nunes  
Dept. Informatics Engineering  
FEUP · U.Porto

# Today's Plan

---

- Context and motivation for data processing
- Focus on data collection and preparation
  - Data collection and preparation tasks
  - Overview of a typical pipeline
  - Example projects
- Review of the first milestone delivery
- Review the plan for the next practical class

# Overview

# "Data"

---

- In Latin, data is the plural of datum.
- In specialized fields, data is treated as plural, e.g. "data were collected".
- Generally, it is treated as a mass noun, like "information", e.g. "data was collected".
- We adopt the use of "data" as mass noun.

# Terminology: Data, Metadata and Information

---

- **Data**

- is a measurement of something on a scale;
- a fact known by direct observation.

- **Metadata**

- is "data about data";
- not the content of data but data providing information about one or more aspects of the data, such as description (date, time, author), structure (format, version), administrative (permissions), legal, etc.

- **Information**

- is data with a context / meaning, thus enabling decision making;
- is data that has been processed, organized and structured.

# Terminology: Decimal and Binary Systems

---

- The binary system uses power of 2 units.
- The decimal system uses power of 10 units.
- In the International System of Units standard, kilo, mega, giga, correspond to powers of 1000 — thus decimal prefixes.
- Historically, the computer industry used the same prefix with two different meanings, i.e. 1MB could either be 1 048 576 bytes (binary) or 1 000 000 bytes (decimal).
- In 2008, binary prefixes — i.e. that refer to powers of 1024 — were officially introduced: kiwi (Ki), mebi (Mi), gibi (Gi), tebi (Ti), pedi (Pi), exbi (Ei), zebi (Zi), yobi (Yi).
  - 1MiB ( $1024^2$ ) = 1 048 576 bytes
  - 1MB ( $1000^2$ ) = 1 000 000 bytes

# Out of Scope

---

- Outside the scope of this course are:
  - Ethical dimensions of data and information
  - Economic aspects of data
  - Legal aspects
  - ....
- Note that these concepts should not be foreign to informatics or computer science.
- Informatics engineers need to be aware of many of these aspects, particular those with strong social impact, in their work. There are many recent examples, e.g. social media, cloud services.
- Possible topics for invited talks.

# Information Life Cycle

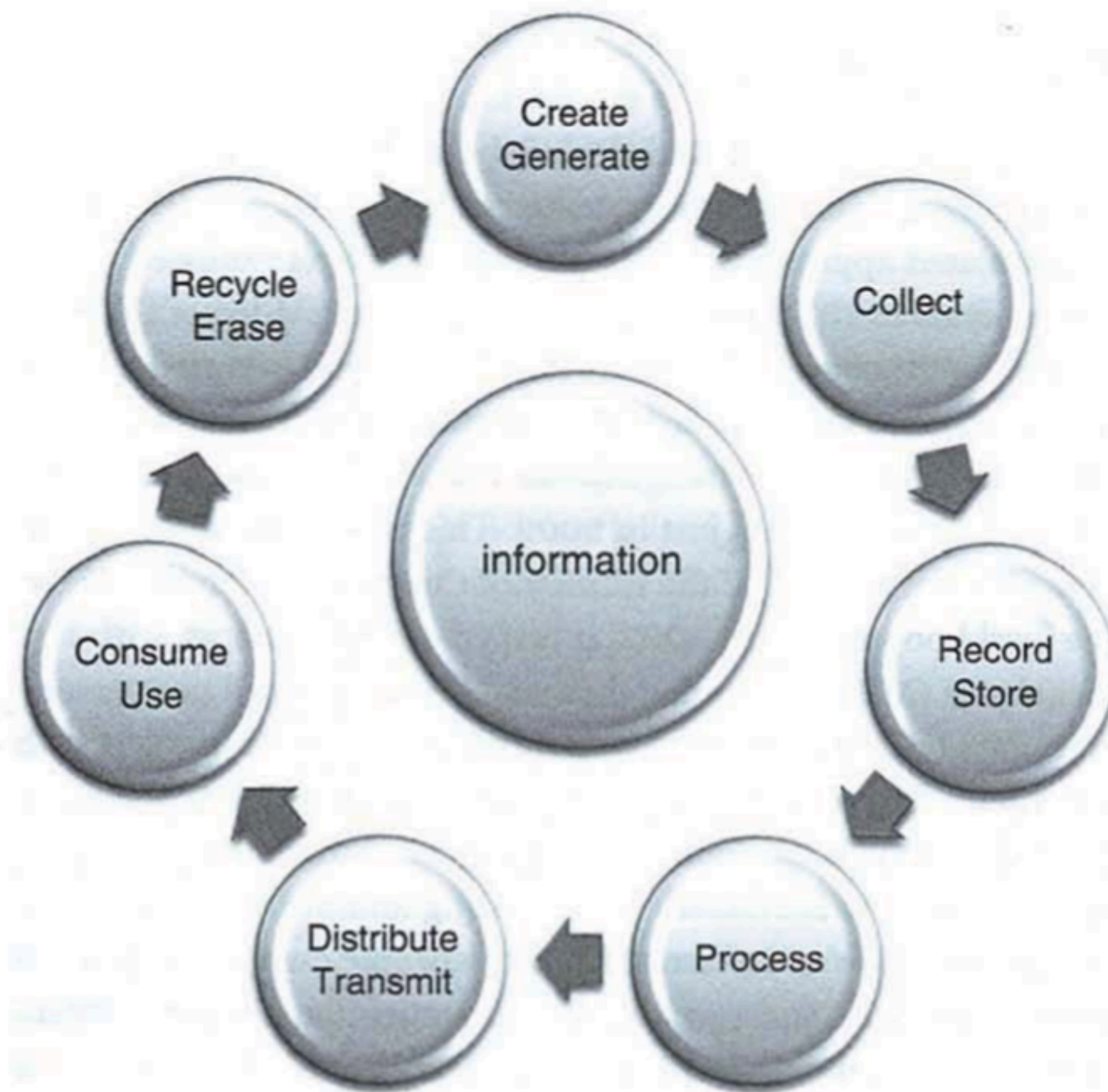
---

- In modern societies, progress and welfare is increasingly dependent on the successful and efficient management of the life cycle of information.
- The life cycle of information typically includes the following phases:
  - Occurrence: discover, design, author, etc;
  - Transmission: networking, accessing, retrieving, transmitting, etc;
  - Processing and Management: collecting, validating, modifying, indexing, classifying, filtering, sorting, storing, etc;
  - Usage: monitoring, explaining, planning, forecasting, decision-making, educating, learning, etc;
- Information and Communication Technologies (ICT) evolved from being mainly recording systems, to being communication systems, to also (and currently) being processing and producing systems.



# Information Life Cycle

---



# The zettabyte\* era

---

- A study from 2003, reported that humanity had accumulated approximately 12 exabytes of data until the emerge of the personal computer.
- Of these, 92% where stored on magnetic media (i.e. digital media).
- A more recent study from 2018, reported that *"the total amount of data created, captured, copied and consumed in the world was 33 zettabytes (ZB) – the equivalent of 33 trillion gigabytes. This grew to 59ZB in 2020 and is predicted to reach a mind-boggling 175ZB by 2025. One zettabyte is 8,000,000,000,000,000,000,000 bits"* \*\*
- Also, *"there are around 600 hyperscale data centres – ones with over 5,000 servers – in the world. Around 39% of them are in the US, while China, Japan, UK, Germany and Australia account for about 30% of the total"* \*\*
- With these trends, i.e. an annual growth rate of 50%, ~150 years from now the number of bits would surpass the number of atoms on Earth ...

\* 1 zettabyte = 1000 exabytes; 1 exabyte = 1000 petabytes; 1 petabytes = 1000 terabytes.

\*\* Vopson, M. *The world's data explained: how much we're producing and where it's all stored. The Conversation, 2021*  
<https://theconversation.com/the-worlds-data-explained-how-much-were-producing-and-where-its-all-stored-159964>

# Value in Data

---

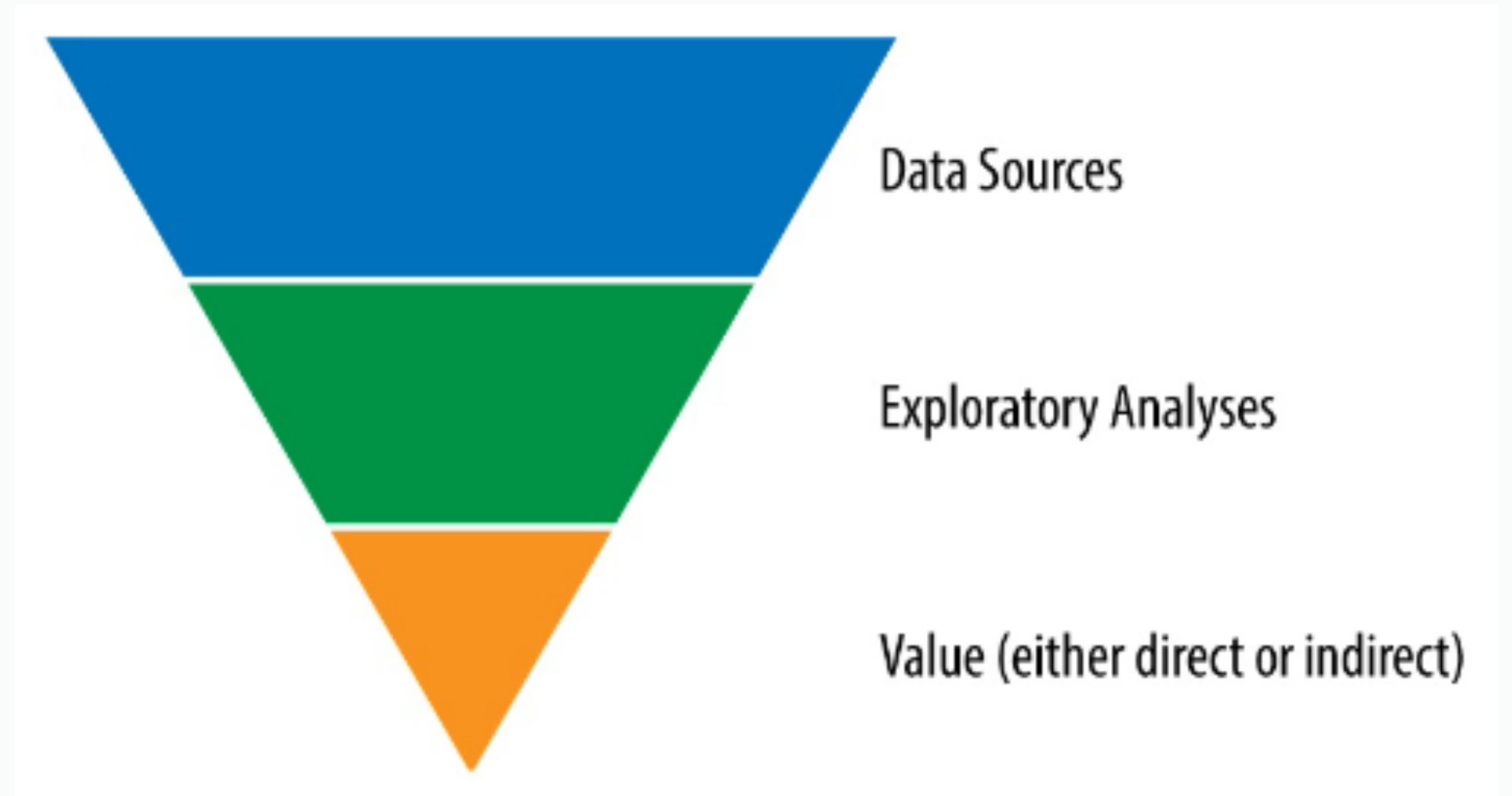
- "Data is the new oil", everybody (circa 20xx).
- Data is a source of value generation, providing evidence and content for the design of new products, new processes, and contribute to more efficient operations.
- In data-driven approaches, multidisciplinary teams experiment and explore large and diverse sources of data to "extract signal from the noise".
- Indirect value — data provides value by influencing of supporting decisions, e.g. risk analysis in insurance, purchase decisions in retail.
- Direct value — data provides value by feeding automated systems, e.g. search system, product recommendation system.



# Data Value Funnel

---

- A large number of data sources and exploratory analysis are required to produce a single valuable application of data.
- Minimize the time spent on non-relevant data by empowering business-experts (i.e. people who know about the business) to explore data.
- Additionally, make data analysis processes as efficient as possible, e.g. by implementing effective data processing workflows.



# Increasing Data Value

---

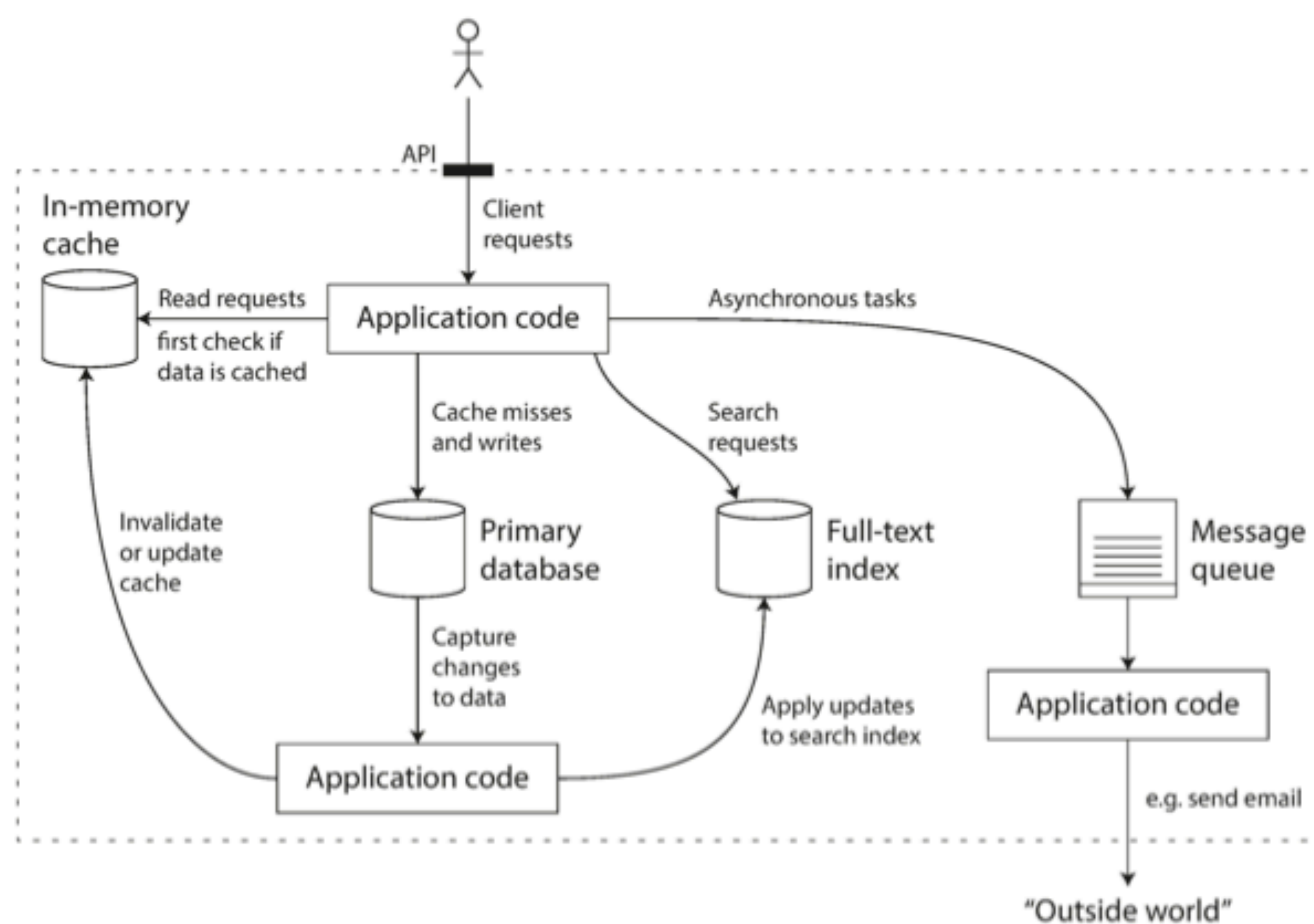
- Make data available, i.e. simply make previously inaccessible data, available.
- Combine data, i.e. create a single coherent whole from disperse data sources; e.g. collection of news from the main news outlets during a year.
- Clean data, i.e. eliminate problems such as incomplete values, duplicates; or create a subset according to specific criteria.
- Structure data, i.e. provide structure to unstructured data; e.g. derive a mentioned entities field from a textual field.
- Enrich data, i.e. complement existing data with data from other sources, including the computation necessary to do so.

# Data-Intensive Applications

---

- Many applications today are data-intensive, as opposed to computing-intensive.
- In this context, existing problems are:
  - the amount of data available;
  - the complexity of the data;
  - the speed at which it changes.
- Common building blocks in data-intensive application include:
  - Store data, for use or sharing (databases);
  - Remember the result of an expensive operation (caches);
  - Enable search and filtering (indexes)
  - Send messages between systems (stream processing);
  - Periodically process large amount of data (batch processing);

# Example of a Data System



# Data Stages

---

- Data moves through three main stages:
  - Raw — focus is on data discovery; the primary goals are ingestion, understanding, and metadata creation; common questions include: what kinds of records are in the data? how are record fields encoded?
  - Refined — focus is on data preparation for further exploration; tasks include removing unwanted parts, reshaping poorly formatted elements; establishing relationships between datasets; assessing data quality issues.
  - Production — focus is on integrating the data into production processes or products.
- Several data processing patterns exist in the literature, including: ETL, ELT, OSEMN.



# ETL Pattern

---

- The ETL framework (extract-transform-load) was coined in the 1970s and popularized in the context of data warehousing.
  - Extract, involves extracting data from the source system.
  - Transform, a series of operations or transformations are applied to the extracted data.
  - Load, involves publishing data to the target system, either simple flat files or other infrastructures.
- ETL is usually associated with classic centralized IT driven operations.

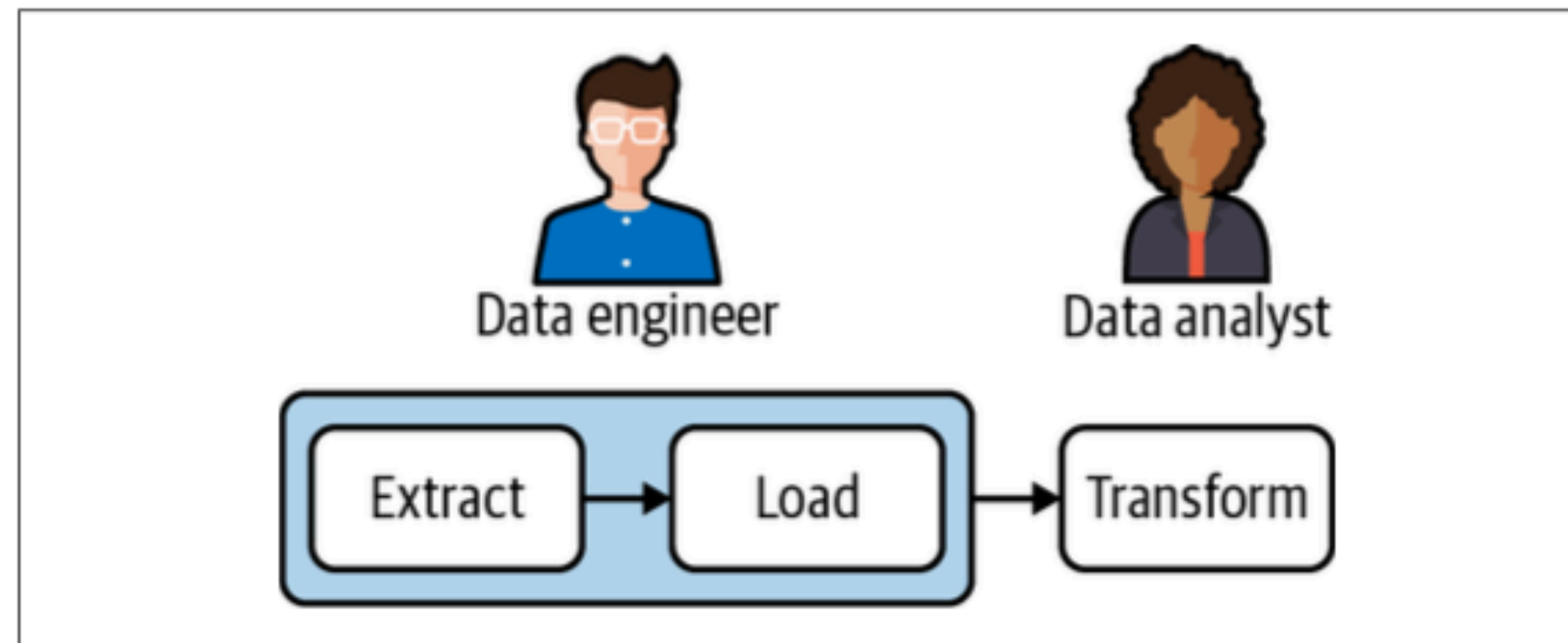
# ELT and EtLT Frameworks

---

- ELT (extract-load-transform) is a recent evolution over the ETL framework.
- Increasingly common access to data storage infrastructures capable of handling large volumes of data has lead to a more flexible pattern.
- Column-oriented data structures are particularly well-suited to typical data processing tasks, i.e. organizing operations per field or property.
- The sub-pattern EtLT introduces a transformation step before the loading, typically associated with data cleaning tasks.
- Load-transform, in contrast with transform-load, is a pattern more well-suited to the division of responsibilities in multidisciplinary teams.

# ELT Pattern

---



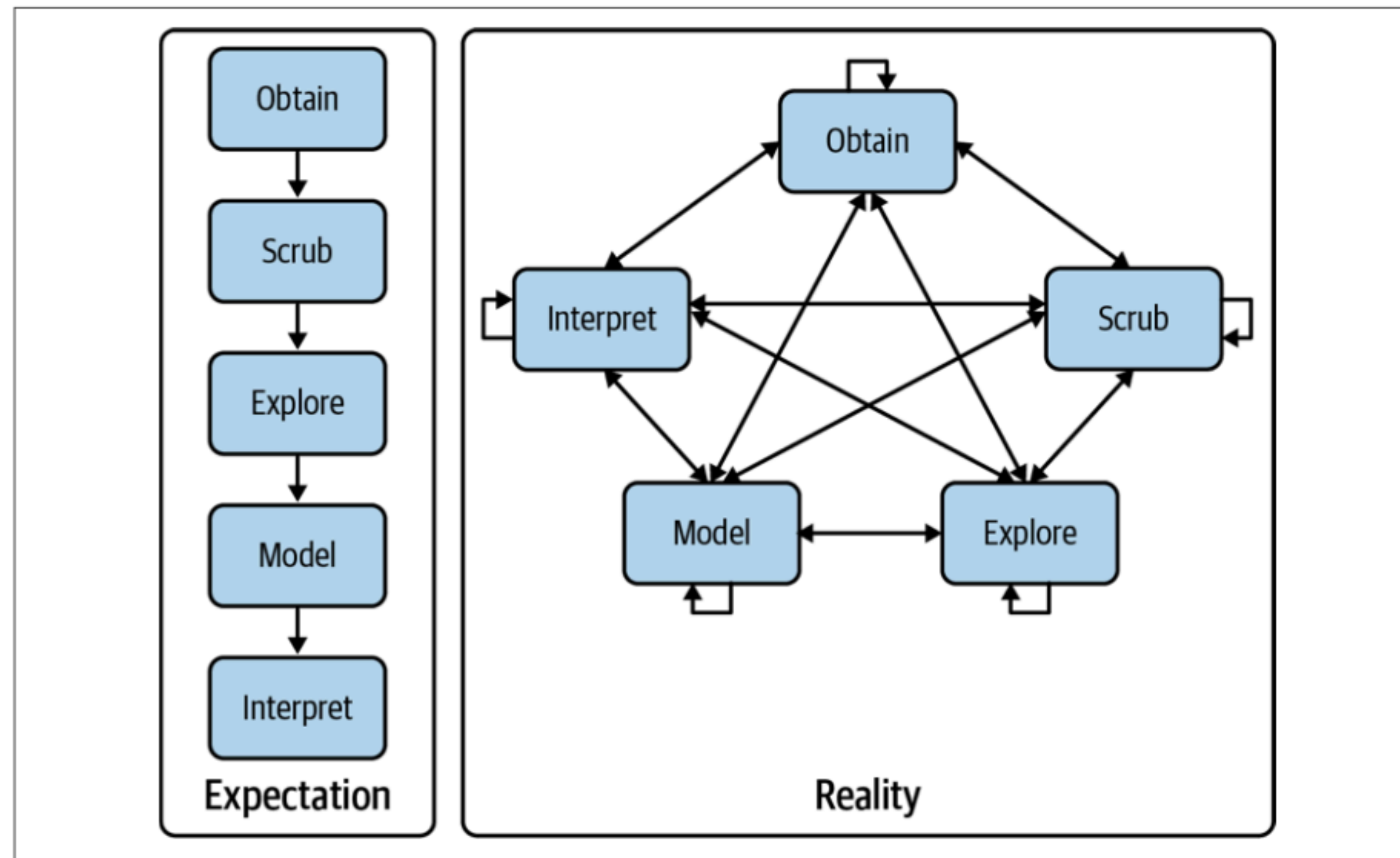
*Figure 3-3. The ELT pattern allows for a clean split of responsibilities between data engineers and data analysts (or data scientists). Each role can work autonomously with the tools and languages they are comfortable in.*

# OSEMN Framework

---

- In the context of Data Science, the OSEMN (pronounced awesome) was coined.
  - Obtain, gathering data.
  - Scrub, clear, arrange, prepare data.
  - Explore, observe, experiment, visualize.
  - Model, create a statistical model of the data.
  - Interpret, drawn conclusions, evaluating and communicating results.
- Although presented as a series of steps, real-world processes are typically non-linear.

# Iterative Process



*Figure 1-1. Doing data science is an iterative and nonlinear process*

# Data Engineers

---

- Data engineers have emerged as an autonomous key role in this context.
- Design, implement and maintain data processing pipelines.
- Work closely with data scientists and analysts to understand what will be done with the data.
- Wide range of technical skills:
  - SQL and Data Warehousing
  - Programming - Python, Java, Go (common in this context)
  - Distributed Computing
  - Cloud Infrastructures
  - System Administration

Data Collection



# Diversity of Data Sources

---

- Data sources vary across many dimensions:
  - Ownership — either owned or from third-parties; understanding ownership is central, i.e. know what data you have access to and what you can do with it;
  - Ingestion interface and structure — how do you get the data and in what form is in;
  - Volume —in each step of the pipeline, volume needs to be taken into account; high- and low- volume are difficult to define and depend on available infrastructures and algorithms;
  - Cleanliness and validity —duplicate data, missing or incomplete data, encoding, etc;
  - Latency and bandwidth of the source — need to consider internal update requirements and also source system limits, speed, timeouts, etc.



# Open Data

---

- The idea that data should be freely available to anyone, to use, modify, and republish for any purpose.
- Associated with a movement, see Open Knowledge Foundation, <https://okfn.org/>
- One of the most important forms of open data is open government data.
- Open data can also be linked, known as linked open data (LOD), see <https://www.w3.org/DesignIssues/LinkedData.html> (2009)
- "Web of data" is an expression to represent the set of technologies and practices that enable a space where data can be automatically discovered and accessed by machines.
- Also related is the concept of FAIR: findable, accessible, interoperable, and reusable; emphasizing machine-actionability over data.
- FAIR/O is used to indicate that a data source complies with FAIR and is also of open nature.

# Data Sources (Examples)

---

- Structured
  - Google Dataset Search, <https://toolbox.google.com/datasetsearch>
  - dados.gov, <https://dados.gov.pt>
  - Dados Abertos em Portugal, <http://dadosabertos.pt>
  - Dados Abertos do Parlamento, <https://www.parlamento.pt/Cidadania/Paginas/DadosAbertos.aspx>
- Unstructured
  - Legislação Portuguesa Consolidada, <https://dre.pt/web/guest/legislacao-consolidada>
  - Acórdãos do Tribunal Constitucional, <http://www.tribunalconstitucional.pt/tc/acordaos/>
- <https://web.fe.up.pt/~ssn/wiki/teach/pri/202122/datasets>

# Data Selection - Things to Consider

---

- Is the author a trustable source that can be contacted?
- Is the data regularly updated?
- Does the data include information about how, when it was acquired?
- Does it seem plausible through observation?

# Ingestion Interfaces and Data Structures

---

- Examples of ingestion interfaces include:
  - A relational database behind an application, such as PostgreSQL, SQLite or Oracle;
  - A layer of abstraction on top of a system, such as a REST API;
  - An endpoint to a message queue system, such as RabbitMQ;
  - A shared network filesystem, containing logs, CSV files, or other flat files.
- Examples of data structures include:
  - JSON from REST API;
  - Well-structured data from a relational database;
  - Semistructured log data;
  - CVS (comma-separated values) datasets;
  - PDF, or other proprietary format, files;
  - HTML, or other semi-structured, files;

# Data Formats

---

- Data formats enable the representation of different types of data in a computer-usable form.
- Common data representations:
  - Alphanumeric: Unicode, ASCII
  - Image (bitmap): PNG, JPG
  - Image (object / vector): PostScript, SVG
  - Sound: AVI, MP3, AAC
  - Documents: PDF, HTML, XML
- Formats used by individual applications are known as proprietary formats.
- Proprietary formats can be open if its specifications are published. Although not free of licensing.
- Some proprietary formats become "proprietary standards" when they become de facto standards due to general use.
- An open format is defined by a published specification, usually maintained by a standards organization (e.g. PNG, FLAC).

# Data Encoding

---

- Data can be encoded
  - in memory, in specific structures such as objects, lists, arrays;
  - as a self-contained sequence of bytes, for file storage or network transmission, e.g. JSON document.
- The process of translating from the in-memory representation to a byte sequence is called **encoding** (also known as serialization), and the inverse is called **decoding** (also parsing, deserialization).
- Most programming languages have built-in support for encoding (and decoding) in-memory data to byte sequences, e.g. Java's `java.io.Serializable`, Python's `pickle`, PHP's `serialize`.
- Useful for transient purposes but in general not adequate in data pipelines — limited to a programming language, reduced interoperability, lower performance, etc.

# JSON, XML Serialization

---

- JSON and XML are the most common text-based encoding standards.
- JSON is widely supported by many applications.
- CSV is also popular, but less powerful.
- These are (somewhat) human readable (an advantage).
- Are the *de facto* solution for data interchange, between organization, between applications, as export formats in applications, as API outputs.
- Limitations include: ambiguous support for number formats, limited support for binary data (e.g. images).



# Binary Serialization

---

- Binary serialization is more compact and faster to parse.
- Is a common solution for within organization data exchange.
- There are many binary formats for JSON - MessagePack, BSON, BSON, etc.
- Apache Thrift (originally Facebook) and Protocol Buffers (protobuf) are open-source binary encoding libraries that produce a binary encoding of a given record.



# Data Quality

---

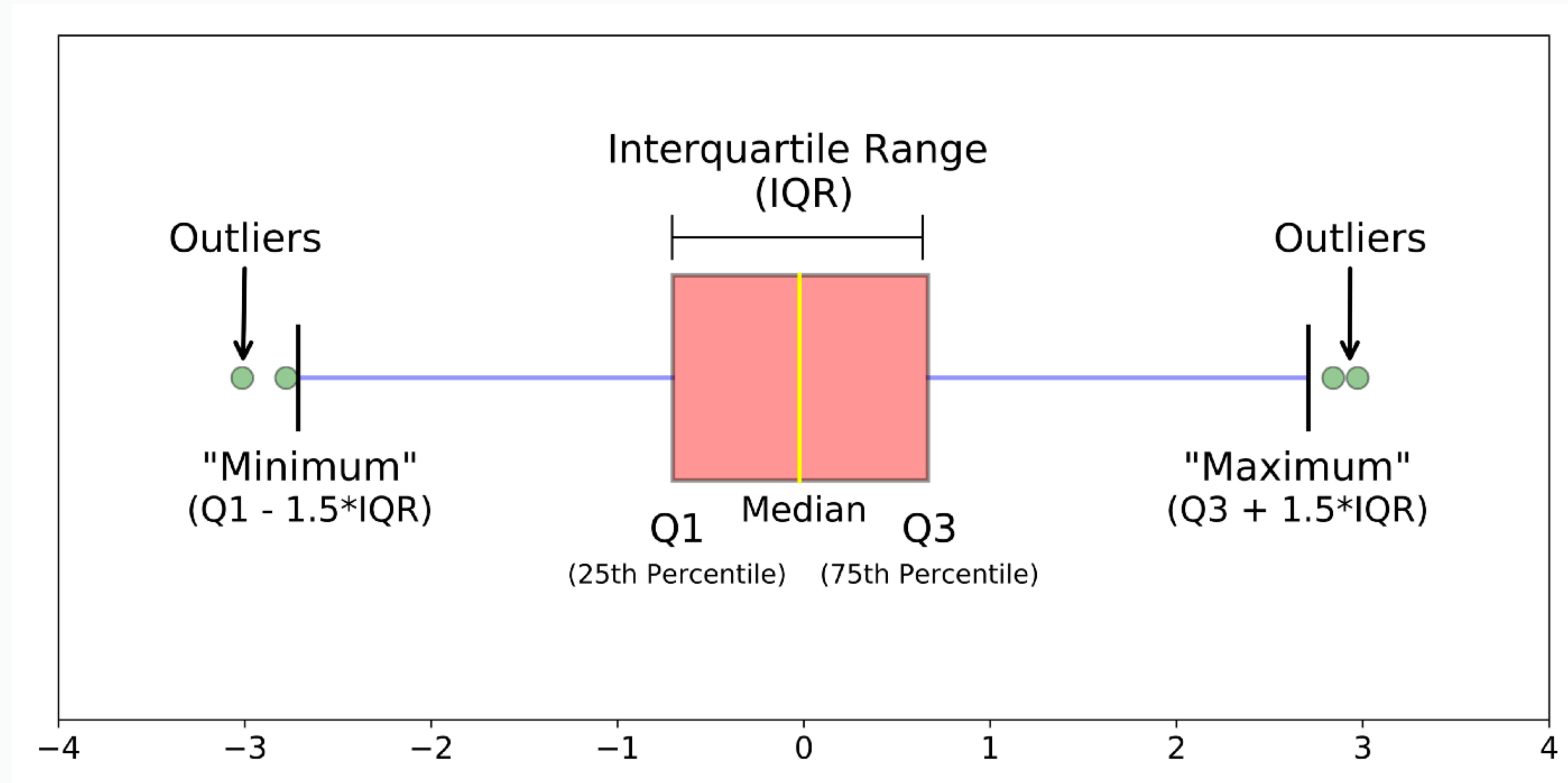
- Common problem affecting data quality:
  - Missing data — due to error, specific meaning (e.g. n/a, 0, NULL).
  - Inconsistent values — distinct timezones in time/dates, multiple units (e.g. m, km).
  - Precision problems — rounding decisions may result in fake patterns (e.g. maps).
  - Duplicate values — due to errors, or valid data.
  - Many other: text encoding problems, mislabeled data, incomplete, outdated, etc.
- There is a need to investigating and understand data properties during the data selection phase — often called data investigation or assessment.

# Overall View of the Distribution

---

- Descriptive statistics are commonly used to begin the investigation.
- Common measures and techniques estimate the:
  - Central tendency, i.e. central, average value of observations:
    - Mean (sum divided by the number of items), median (middle value that separates the observations), mode (most frequent value);
  - Dispersion, i.e. how much the values vary:
    - Standard deviation, interquartile range (difference between values in upper and lower quartile), difference between the maximum and minimum.
- Box plots are a methods that graphically depicts most of these descriptive statistics.

# Box plots

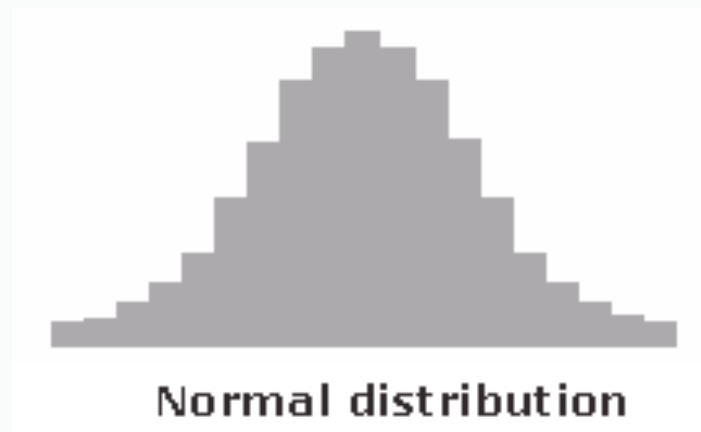


- Image from Understanding Boxplots, Towards Data Science  
[ <https://towardsdatascience.com/understanding-boxplots-5e2df7bcbd51> ]

# Frequency Histograms

---

- For large data volumes, observing distributions of attribute values can be done using histograms, which represent how numerical data is distributed.
- A key aspect of producing histograms is exploring with different bin sizes.
- Numerous distributions exist and are described in detail in statistical literature.

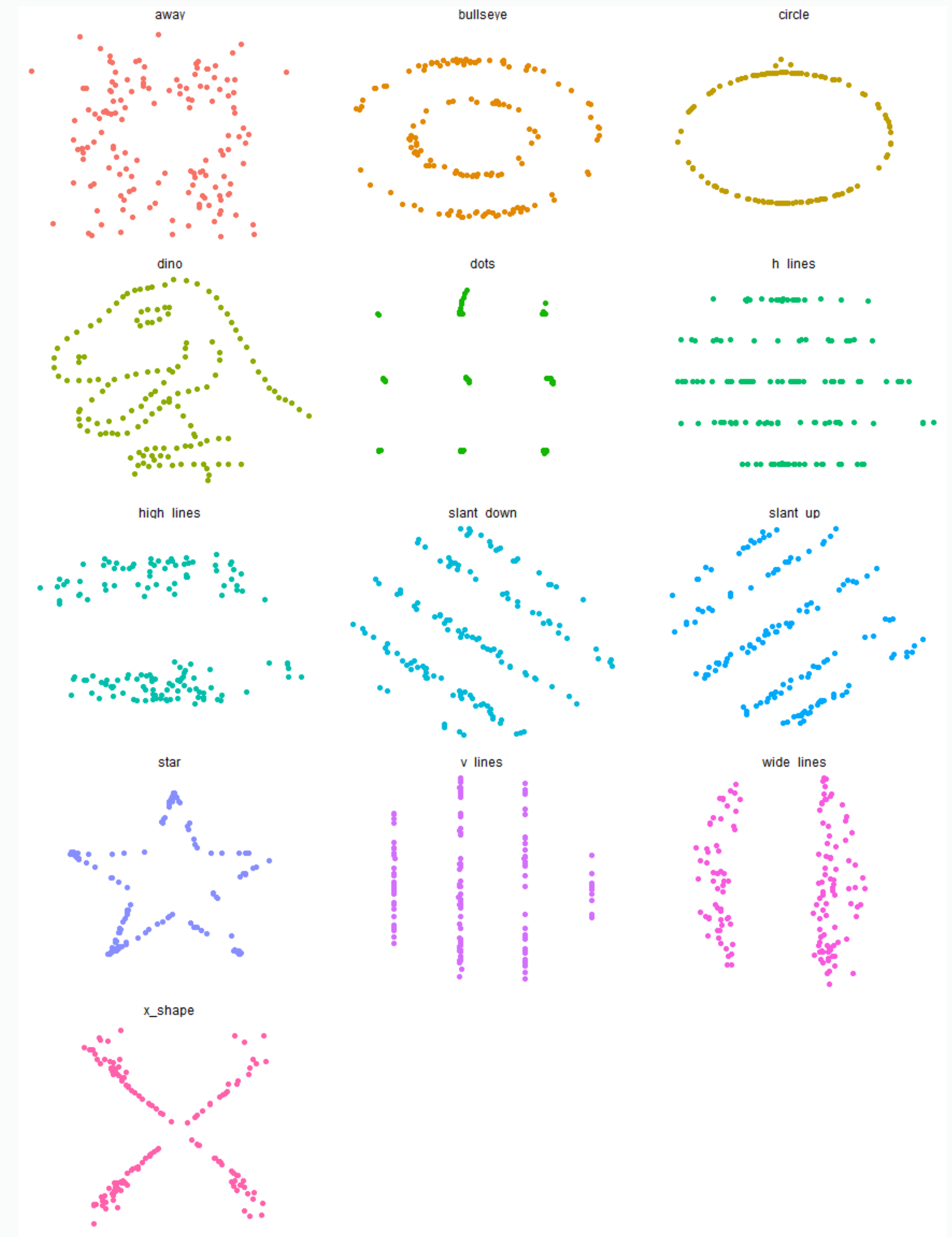


- Examples from "What are Histograms?", ASQ  
[<https://asq.org/quality-resources/histogram>]



# Descriptive statistics only provide a summary

- The image on the right depicts 13 sets of x-y data, where basic descriptive statistics have the same values, i.e. x-mean, y-mean, x-std, y-std), but look very different.
- Don't rely only on descriptive statistics, include exploratory visualization in your process.
- Image from the The Datasaurus data package [ <https://cran.r-project.org/web/packages/datasauRus/vignettes/Datasaurus.html> ]



# Outliers

---

- Outliers are items that differ significantly from others.
- It is necessary to understand if these values are exception but valid cases or if are errors that need to be removed. Expert domain-knowledge is often necessary.
- Errors resulting in outliers may be the result of:
  - Problems in the data collection procedure;
  - Hardware or software problems in data collection tools;
  - Human mistakes in data recording.
- Outliers may significantly distort descriptive statistics or visualizations.



# Missing Data

---

- Missing data is an important aspect of data quality that always needs a detailed investigation to determine its origin and impact on following steps.
- Isolated instances of missing data aren't usually a problem, however if the missing data is not randomly distributed or occurs in large numbers globally or in specific variables, the data set will be biased and thus not appropriate for a valid analysis.
- Missing data can also be an indicator of flaws in the data collection process.

# Data Quality Summary

---

- Investigating the properties of the data at its origin and at different points of the data processing pipeline is a key aspect of a solid data-based project, helping on:
  - Deciding on the data sources to select;
  - Determining possible bias and limitations of the data *a priori*;
  - Detecting and correcting problems in the pipeline;
  - Framing the conclusions or built products;
- Data quality investigations should rely on multiple methods, from descriptive statistics to exploratory visualization.
- Best practices: clean and validate in the best system to do so; validate often.



# Tools for Data Collection

---

- SQL for extracting data from databases.
- Custom code for APIs.
- Unix commands to work with external sources, e.g. curl.
- Web crawling platforms:
  - Scrapy, <https://scrapy.org/> [Python]
  - Internet Archive Heritrix, <https://github.com/internetarchive/heritrix3> [Java]
  - Apache Nutch, <http://nutch.apache.org> [Java]

# Data Preparation

# Data Preparation

---

- Real-world data is "messy".
- In practice, 50% to 80% of the time spent in data processing is spent in data preparation tasks.
- Data preparation, often called "data wrangling", captures activities like:
  - Understand what data is available;
  - Choose what data to use and at what level of detail;
  - Understand how to combine multiple sources of data;
  - Deciding how to distill the results to a size and shape that enables the follow-up steps.

# Data Preparation

---

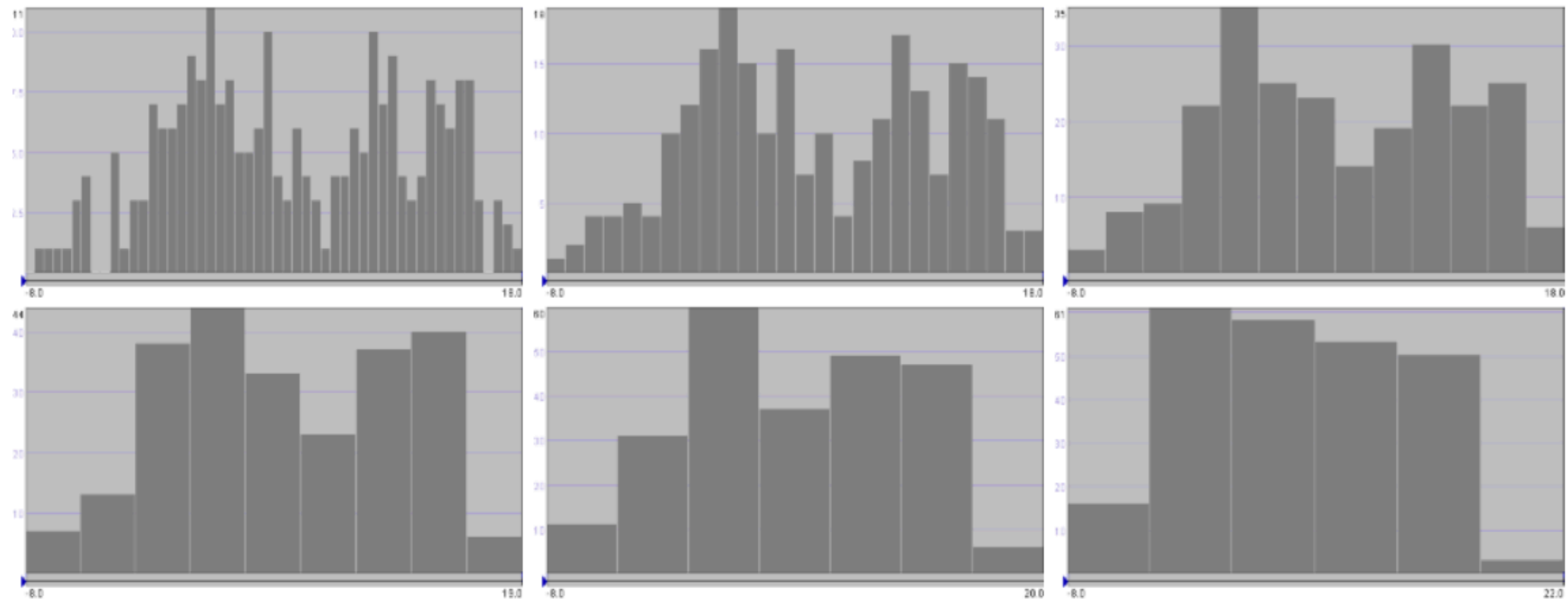
- After data properties are investigated and understood, a data preparation phase is generally needed to make the data suitable for the follow-up phases.
- Common data preparation tasks include:
  - Data cleaning — identify and fix data quality issues;
  - Data transformation — transform data to improve analysis or manipulation;
  - Synthesis of Data — create new attributes derived from existing data;
  - Data integration — combine data from different sources;
  - Data reduction or selection — eliminate data from the collection.

# Data Transformation

---

- Data transformation operations can be performed to improve or facilitate data handling in subsequent stages.
- Transformation of data elements include:
  - Normalization of values to a comparable scale;
  - Scaling values to the same range (e.g. 0 to 1 for a follow-up method);
  - Non-linear transformations to deal with skewed distributions;
  - Discretization or binning, which transforms a numeric continuous value into ordered categorical values (e.g. having bins of the same size vs. having bins with the same amount of items)
- Note that all operations over the data introduce layers of distortion and bias that are dependent on assumptions and interpretations. Documentation is key.

# Discretization



**Fig. 3.13:** Frequency histograms of the temperature data with the bin sizes equal to 0.5, 1, 2, 3, 4, and 5 degrees Celsius, correspondingly (left to right, top to bottom).

- Image from "Visual Analytics for Data Science", Springer (2020).

# Synthesis of Data

---

- Combine existing attributes to produce new additional attributes that are more convenient to analyze or use as input in follow-up phases. Examples:
  - a new attribute representing the difference between two timestamps (e.g. duration);
  - the maximum value from a series of numerical attributes;
  - an integrated score that combines several attributes;
  - splitting an existing numerical series in two independent series (e.g. day and night);
  - most important keywords or topics extracted from a textual field;
  - etc.



# Data Integration

---

- Combine data that originally exists in multiple sources.
- Key task in many projects, e.g. integrate data from multiple databases in an organization; enrich individual records with data from external sources.
- Complexity of data integration tasks differs significantly, from using join operations in a single database management system, to downloading, processing and linking external data sources.
- A central step of many data integration tasks is linking the corresponding records. A task that is easier if unique identifiers are available (rarely!), but of high complexity if based on other information (e.g. names, birth dates, addresses).



# Data Reduction or Selection

---

- Data reduction or selection may be justified due to several reasons, namely:
  - data is not relevant;
  - data is outdated;
  - data volume exceeds the existing capacity for processing it;
  - existing precision is excessive, while lower precision is sufficient.
- Techniques for performing data reduction or selection include:
  - data filtering
  - data sampling
  - data aggregation

# Data Filtering

---

- Data filtering is used to remove data from the dataset, e.g. items with unsuitable values, data that is irrelevant for the scope of the project, outdated data items, data items that must be removed due to legal reasons, etc.
- Data filtering can also be used during development to test the planned approach using a manageable portion of the original collection.
- Data filtering operations are deterministic in nature, e.g. remove data from a given year, remove all references to a specific keyword, remove a specific data attribute, etc.

# Data Sampling

---

- Data sampling is a non-deterministic process that takes a random subset of the data items of a requested size.
- When designing the sampling method it is important to ensure that the resulting sample is representative of the complete collection.
- Thus, it is important to analyze the distribution of data attributes before and after the sampling process.

# Data Aggregation

---

- Data aggregation may be used to reduce excessive detail in data, decisions require:
  - choice of the grouping method, depending of domain-specific criteria;
  - selection of the aggregation operator (e.g. mean, median, min, max, percentile).

# Visualization in Data Preparation

---

- Data preparation requires a good understanding of the data properties, thus data visualization is usually also used at this stage.
- The role of visualization at this stage is to visually present the result of the execution of different methods, e.g. outlier removal.
- Visualization in data preparation can be applied:
  - Before the application of computational methods to explore the properties of the data, e.g. choose an adequate computational method, detect disparities in data.
  - During the application of computational methods to inspect how data is being processed, e.g. observe intermediate data structures, track the data pipeline execution.
  - After the application of computational methods to evaluate the quality of the results, compare the results between different executions of the pipeline.

# Tools for Data Preparation

# Tools for Data Cleaning

---

- Existing tools combine visual and computational techniques for identifying and fixing data problems.
- A freely available open-source reference software is OpenRefine (formerly Google Refine), available at <https://openrefine.org/>
  - Focus on tabular data;
  - Supports data cleaning operations on numerical data, dates and time, and textual data;
  - Data exploration with descriptive statistics, facets, and histograms;
  - Data transformation with batch detection and replacement of inconsistent or missing values, time and date formatting, textual misspellings;
  - Integration with Wikidata for both obtaining and publishing data.
- Explore available tutorials in lab classes.



# OpenRefine

OpenRefine

100 Most Populous Cities from Wikidata

Permalink

Open...

Export ▾

Help

Facet / Filter

Undo / Redo 0 / 0

Refresh

Reset All

Remove All

countryLabel

change

20 choices

Sort by: name count

Cluster

Bangladesh 1

Brazil 2

Chile 1

Democratic Republic of the Congo 1

India 8

Indonesia 1

Iran 1

Japan 2

Manchukuo 1

Nigeria 2

100 rows

Extensions: Wikidata ▾

Show as: rows records Show: 5 10 25 50 rows

« first < previous 1 - 10 next > last »

		cityLabel	population	countryLabel
★	🗨	1. Shanghai	23390000	People's Republic of China
★	🗨	2. Beijing	21710000	People's Republic of China
★	🗨	3. Lagos	21324000	Nigeria
★	🗨	4. Dhaka	16800000	Bangladesh
★	🗨	5. Mumbai	15414288	India
★	🗨	6. Istanbul	14657434	Turkey
★	🗨	7. Tokyo	13942856	Japan
★	🗨	8. Tianjin	13245000	People's Republic of China
★	🗨	9. Guangzhou	13080500	People's Republic of China
★	🗨	10. São Paulo	12106920	Brazil

55 records

Schema \*

Issues

Preview

Extensions: Wikidata ▾

The Wikidata schema below specifies how your tabular data will be transformed into Wikidata edits. You can drag and drop the column names below in most input boxes: for each row, edits will be generated with the values in these columns.

Save schema

Discard changes

Author Title Publication year Literary genre Country Reference

Title

remove

Terms

no labels, descriptions or aliases added

+ add term

Statements

author

Author

remove

+ add qualifier

1 references

copy

reference URL

Referen...

remove

remove

+ add

+ add reference

+ add value

publication date

Publication y...

remove

+ add qualifier

1 references

copy

reference URL

Referen...

remove

remove

+ add

+ add reference

+ add value

genre

Literary ge...

remove

+ add qualifier

1 references

copy

reference URL

Referen...

remove

remove

+ add

+ add reference

+ add value

+ add statement



# Tools for Data Preparation

---

- Parse and extract text and metadata from files.
  - Apache Tika (e.g. PDF, PPT, XLS), <https://tika.apache.org>
  - BeautifulSoup (HTML), <https://www.crummy.com/software/BeautifulSoup/>
- NLP toolkits provide natural text processing tools
  - spaCy, <http://spacy.io>
  - NLTK, <http://nltk.org>
  - Apache OpenNLP, <http://opennlp.apache.org>
- Data processing
  - R, <http://www.r-project.org>
  - Pandas, <https://pandas.python.org>

# Cloud Computing

---

- Cloud-based services are not a focus of this course but are a relevant piece in modern data processing and analytics pipelines.
- Managed services make building and deploying data pipelines more accessible, due to the elastic properties of the service, and the outsourcing of the setup and configuration part.
- Models based on pay-per-use, enable setting up and discarding data processing pipelines as needed.
- Cloud solutions are a good option to ad-hoc, highly diverse needs.

# Data Pipelines

# Data Pipelines

---

- Data pipelines are sets of processes that move and transform data from various sources to various destinations where new value can be derived.
- Complexity of data pipelines varies widely, depending on size, state, structure of the data sources as well as the needs of the project.
  - Simple data pipeline example: obtain data from a REST API and load it to a relational database;
  - Complex data pipeline example: obtain data from multiple web pages at regular intervals; parse HTML and extract main keyword from each page; automatically identify main trending topics using a previously trained machine learning model; update model with new data; integrate topics and web pages in a cloud-based storage linked to a live dashboard;

# Characteristics of a Data Pipeline

---

- A data pipeline is a software system, thus general software best practices apply. Those highlighted here are particularly relevant in the context of data pipelines.
  - Reliable — work as expected even in face of adversity (i.e. software, hardware or human faults).
  - Scalable — cope with increased load (in volume, traffic, complexity) in a manageable way (e.g. adding resources, distributing load).
  - Maintainable — evolve through time and teams (reduce complexity, make changes easier).
- Data collection and preparation is usually an ad-hoc and exploratory process, easily leading to a dispersion in threads and activities. Adoption of pipeline management systems (e.g. Makefiles) and a complete and detailed documentation is key.
- Treating data processing pipelines as software makes them more maintainable, versionable, testable and collaborative.

# Tools for Data Pipelines



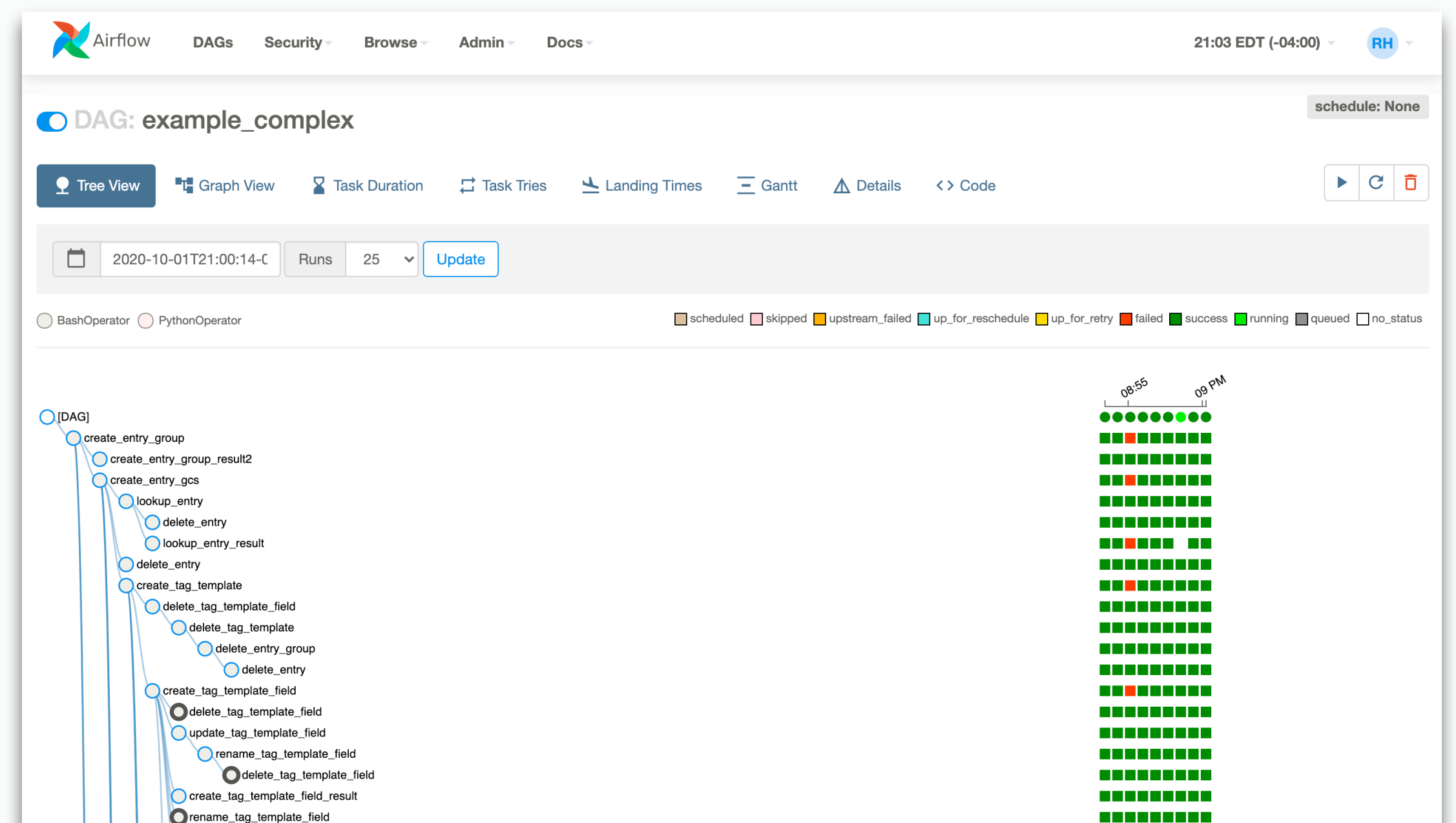
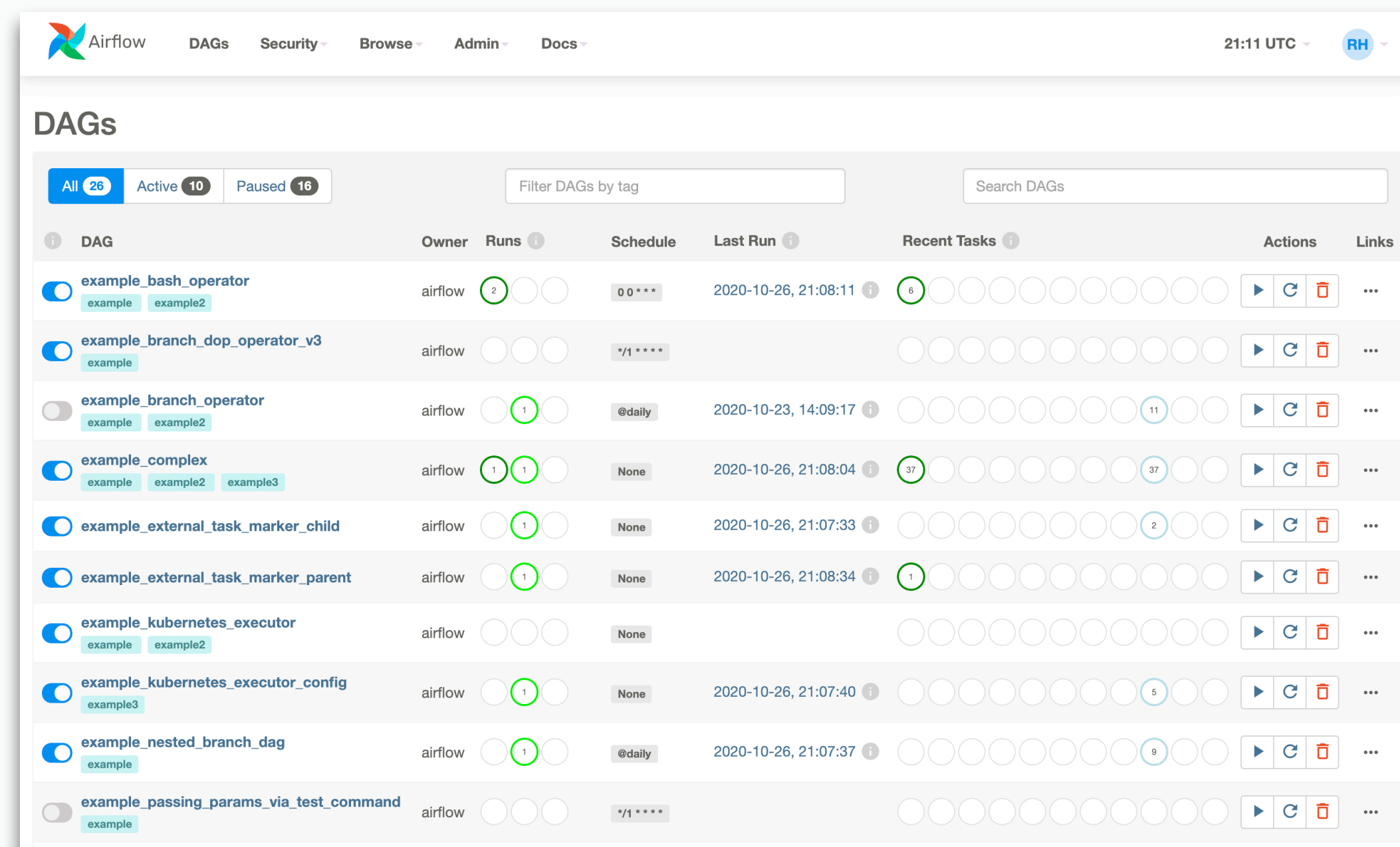
# Makefiles

---

- Makefiles are used to automate software build processes, by defining targets and rules to execute. The underlying abstraction is of a dependency graph, where tasks depend on the execution of other tasks.
- Make is language agnostic and implementations exist for most operating systems.
- Can be used to document and setup data pipelines.
- Expected to be used in the final project.
- To be explored in a lab class.

# Apache Airflow

- A Python-based, open-source platform, to "programmatically author, schedule and monitor workflows" [ <https://airflow.apache.org/> ]



# Documentation of Data Pipelines

# Data Documentation

---

- Documentation is central in any engineering process.
- It distinguishes between ad-hoc and repeatable, inspectable, shareable processes.
- It is essential to document:
  - Data formats across the pipeline (input, output, as well as intermediary files);
  - Platform and software versions (OS, tools);
  - Operations performed (extract, sample, link, etc);
- For understanding underlying assumptions (bias) in data documentation is a key element.

# Data Flow Diagrams (DFD)


---

- Data-flow diagrams can be used to represent the flow of data from external entities into the system, show how data moves from one process to another, and data's logical storage.
- The notation includes four symbols:
  - **Squares** represent external entities, i.e. sources or destinations of data
  - **Rounded rectangles** represent processes, which takes data as input, perform operations over it, and then output it.
  - **Arrows** represent data flows, i.e. how data moves around.
  - **Open-ended rectangles** represent data stores, e.g. databases, files.

# Example Projects



ant.fe.up.pt/search?q=MIEIC



MIEIC

TodosEstudantesPessoalNotíciasCadeirasDepartamentosSalasCursosFerramentas de Pesquisa

Em alternativa, repita a pesquisa na web: MIEIC


3442 resultados (0.04 segundos)

Mestrado Integrado em Engenharia Informática e Computação

Cursohttps://sigarra.up.pt/feup/pt/cur\_geral.cur\_view?pv\_ano\_lectivo=2020...  
Faculdade de Engenharia da Universidade do Porto (FEUP)

Áreas Científicas Predominantes: Engenharia Informática e Computação

Diretores: João Carlos Pascoal Faria, Maria Cristina de Carvalho Alves Ribeiro



Rodrigo Manuel Lopes de Matos Moreira

Estudantehttps://sigarra.up.pt/feup/pt/vld\_entidades\_geral.entidade\_pagina?pct...  
Faculdade de Engenharia da Universidade do Porto (FEUP)

Curso: Programa Doutoral em Engenharia Informática

Código: 200403627

I003

Salahttps://sigarra.up.pt/feup/pt/instal\_geral.espaco\_view?pv\_id=74766  
Estudantes de MIEIC , Faculdade de Engenharia da Universidade do Porto (FEUP)

Responsáveis: Hugo José Sereno Lopes Ferreira

Edifício: Electrotecnia (I) Andar: 0

Física I

Cadeirahttps://sigarra.up.pt/feup/pt/ucurr\_geral.ficha\_uc\_view?pv\_ocorrencia...  
Faculdade de Engenharia da Universidade do Porto (FEUP)

Cursos Responsáveis: Mestrado Integrado em Engenharia Informática e Computação (MIEIC)

Docentes: Francisco Carvalho Neto de Queiros Pimenta, Jaime Enrique Villate Matiz, Joana dos Santos Brojo Ascenso, Maria Helena Sousa Soares de Oliveira Braga, Mercedes Esteves Filho

Visão por Computador

Cadeirahttps://sigarra.up.pt/feup/pt/ucurr\_geral.ficha\_uc\_view?pv\_ocorrencia...  
Faculdade de Engenharia da Universidade do Porto (FEUP)

Cursos Responsáveis: Mestrado Integrado em Engenharia Informática e Computação (MIEIC)

Docentes: Jorge Alves da Silva, Luís Filipe Pinto de Almeida Teixeira

Provas Públicas de Dissertação do MIEIC| 1.º Semestre, do ano letivo 2019.20

Notíciahttps://sigarra.up.pt/feup/pt/noticias\_geral.ver\_noticia?p\_nr=13334  
Faculdade de Engenharia da Universidade do Porto (FEUP) - 27 Março 2020 às 00h00


As sessões agendadas podem ser consultadas no Calendário das provas do Mestrado Integrado em Engenharia Informática (MIEIC). As provas agendadas no ...

Palestra MIEIC: A Plataforma TOCOnline, Software as a Service na Cloudware

Notíciahttps://sigarra.up.pt/feup/pt/noticias\_geral.ver\_noticia?p\_nr=76170  
Faculdade de Engenharia da Universidade do Porto (FEUP) - 18 Maio 2018 às 00h00

No contexto da unidade curricular de LBAW do MIEIC, vai ter lugar a seguinte palestra convidada na próxima segunda-feira, dia ...

ant.fe.up.pt/search?q=tecnologias%20web



tecnologias web

TodosNotíciasCadeirasPessoalEstudantesCursosSalasDepartamentosFerramentas de Pesquisa

Qualquer unidadeQualquer cursoQualquer intervalo de créditos

Qualquer unidade

Faculdade de Engenharia da Universidade do Porto (FEUP)301

Faculdade de Ciências da Universidade do Porto (FCUP)183

Faculdade de Letras da Universidade do Porto (FLUP)86

Faculdade de Belas Artes da Universidade do Porto (FBAUP)48

Faculdade de Medicina da Universidade do Porto (FMUP)47

Instituto de Ciências Biomédicas Abel Salazar (ICBAS)36

Faculdade de Economia da Universidade do Porto (FEP)25

Faculdade de Farmácia da Universidade do Porto (FFUP)23

Faculdade de Medicina Dentária da Universidade do Porto (FMDUP)17

Faculdade de Arquitectura da Universidade do Porto (FAUP)11

Linguagens e Tecnologias Web

Cadeirahttps://sigarra.up.pt/feup/pt/ucurr\_geral.ficha\_uc\_view?pv\_ocorrencia...  
Faculdade de Engenharia da Universidade do Porto (FEUP)

Cursos Responsáveis: Mestrado Integrado em Engenharia Informática e Computação (MIEIC)

Docentes: André Monteiro de Oliveira Restivo, Mafalda Falcão Torres Veiga de Ferreira, Sérgio Sobral Nunes, Tiago Nuno Mesquita Folgado leitão Devezas

Web Design I

Cadeirahttps://sigarra.up.pt/fbaup/pt/ucurr\_geral.ficha\_uc\_view?pv\_ocorrenci...  
Faculdade de Belas Artes da Universidade do Porto (FBAUP)

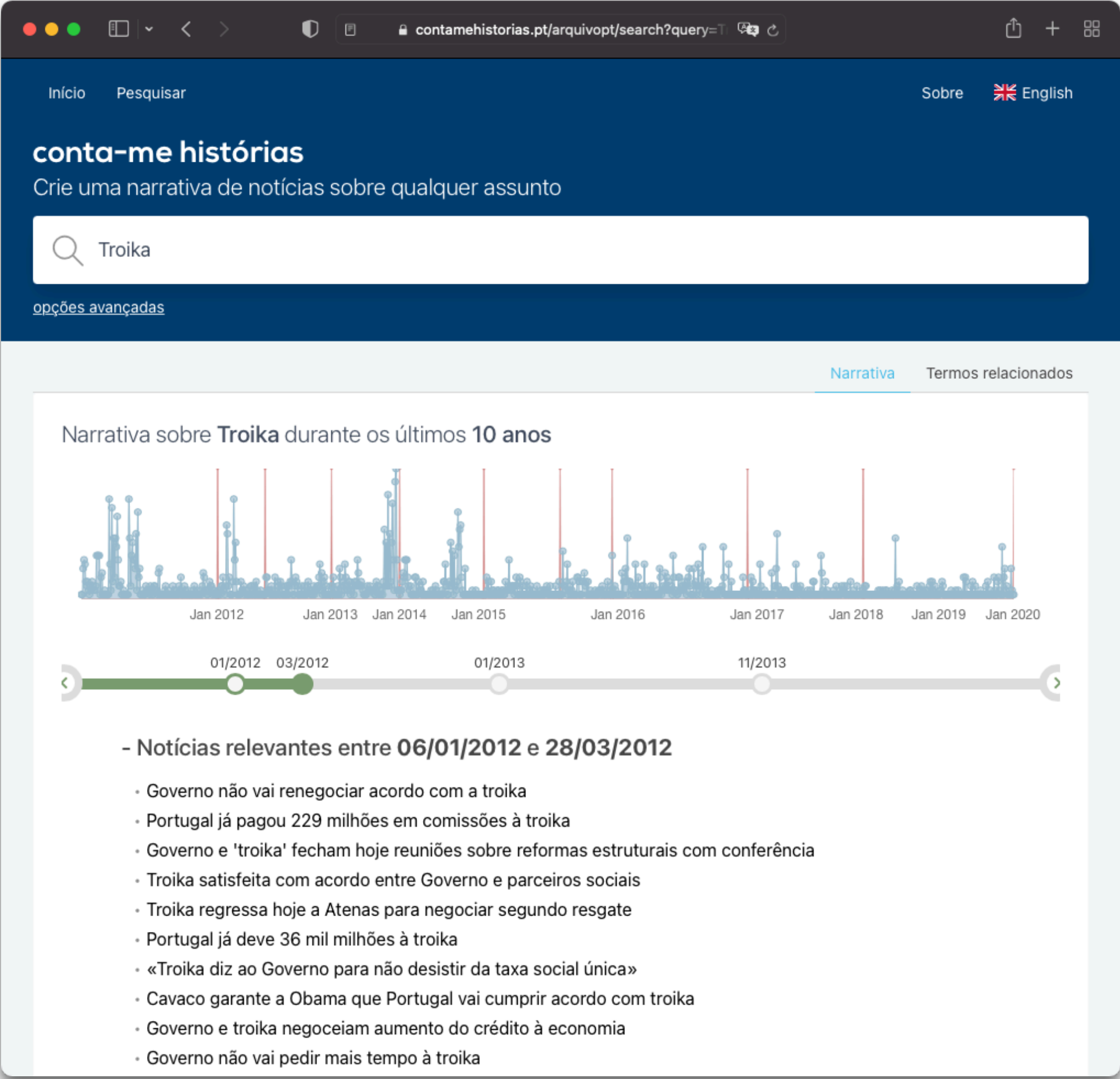
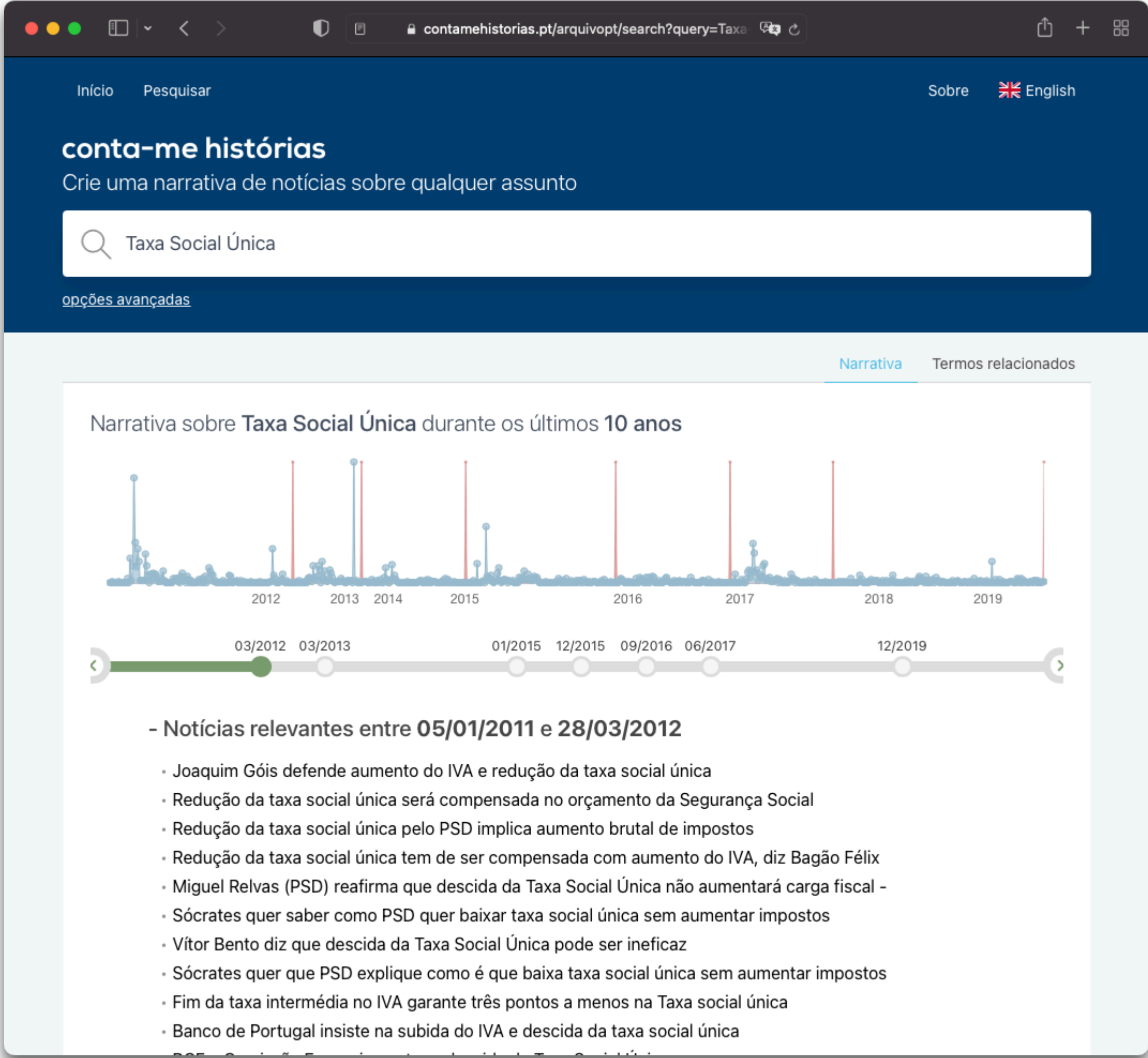
Cursos Responsáveis: Design de Comunicação (DC)

Docentes: Pedro Jorge Couto Cardoso

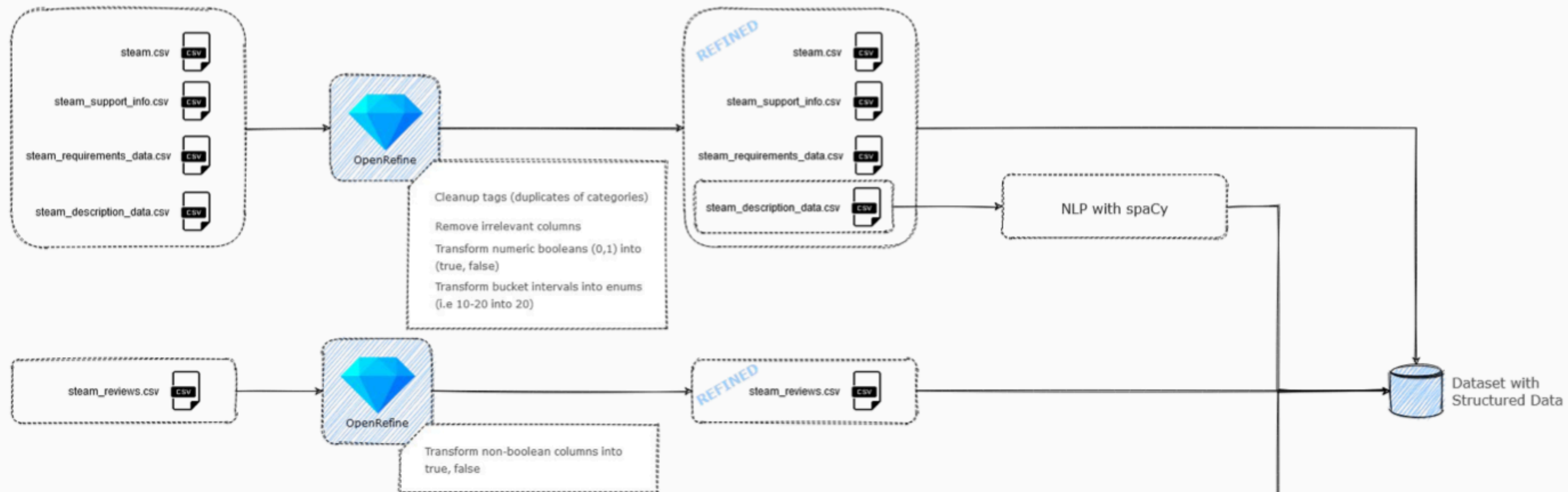
Web Design II

Cadeirahttps://sigarra.up.pt/fbaup/pt/ucurr\_geral.ficha\_uc\_view?pv\_ocorrenci...  
Faculdade de Belas Artes da Universidade do Porto (FBAUP)

# Conta-me Histórias







Wikipedia API



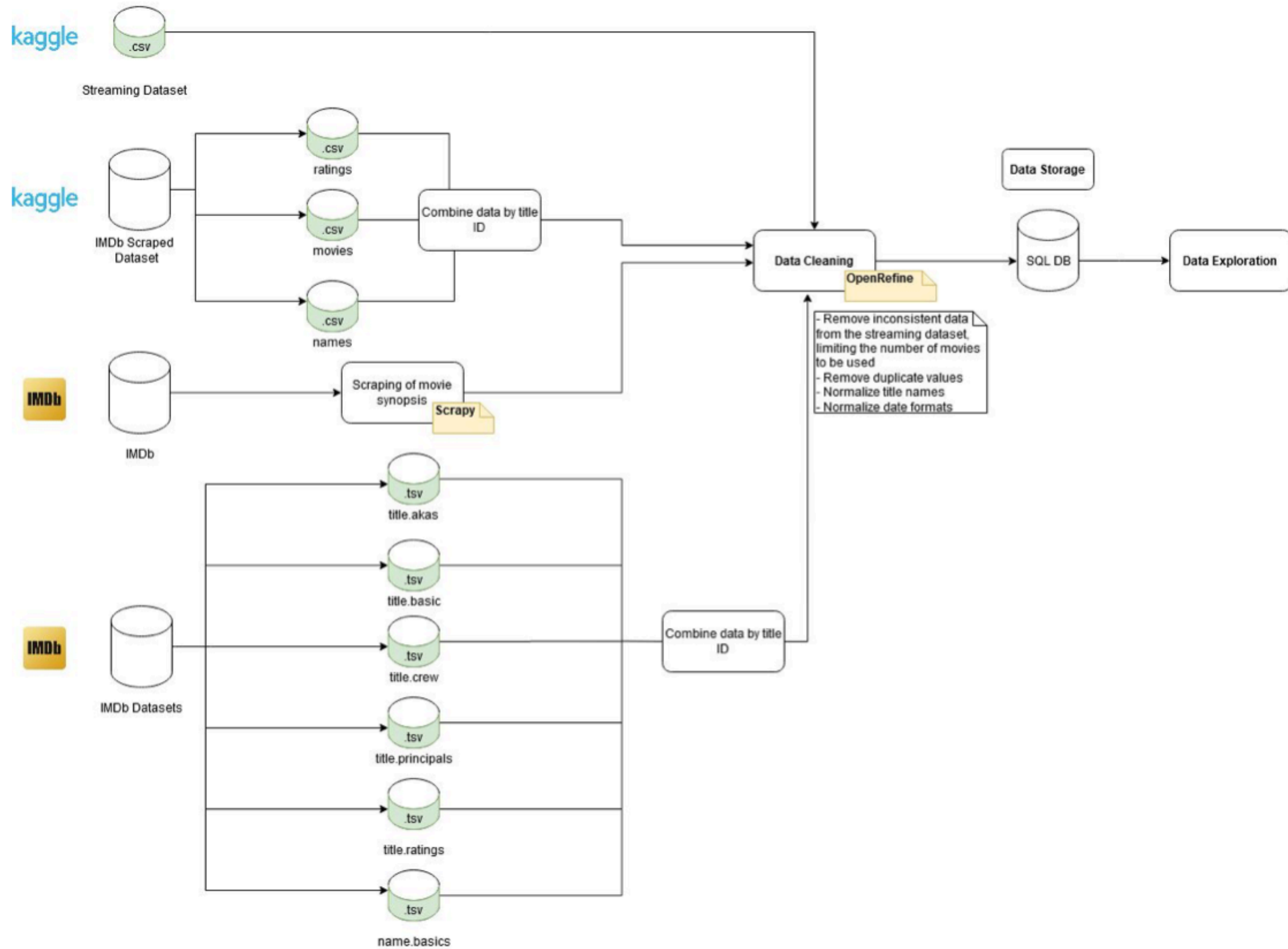
WikiData API

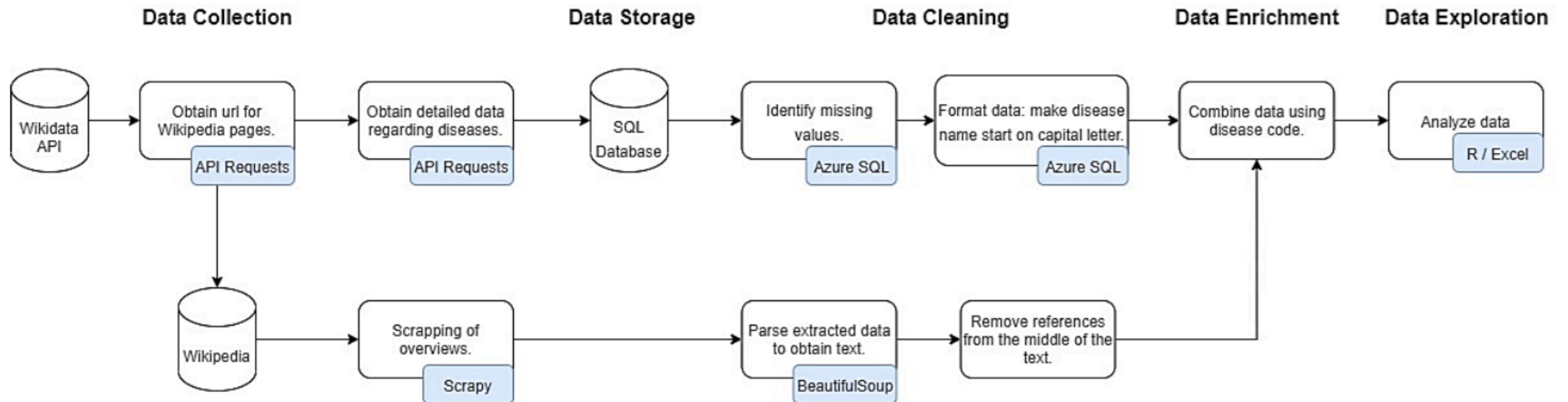
Get Publisher and Developer for each game

Parsing  
Take the first paragraph from the wikipage

NLP with spaCy

Dataset with Structured Data





# Bibliography and Further Reading

---

- **Visual Analytics for Data Scientists** [VADS20]  
Andrienko, N., Andrienko, G., Fuchs, G., Slingsby, A., Turkay, C., and Wrobel, S. Springer, 2020
- **Designing Data-Intensive Applications** [DDIA17]  
Kleppmann, M. O'Reilly, 2017
- **Principles of Data Wrangling** [PDW17]  
Rattenbury, T., Hellerstein, J. M., Heer, J., Kandel, S., and Carreras, C. O'Reilly, 2017
- **Data Pipelines Pocket Reference** [DPPR21]  
Densmore, J. O'Reilly, 2021
- **Information - A Very Short Introduction** [IVSI10]  
Floridi, L. Oxford University Press, 2010

Questions or comments?