Data Preparation

PRI 24/25 · Information Processing and Retrieval
M.EIC · Master in Informatics Engineering and Computation

Sérgio Nunes
Dept. Informatics Engineering
FEUP · U.Porto

Today's Plan

- Focus on data preparation
  - Data preparation tasks
  - Overview of a typical pipeline
  - Example projects

- Review of the first milestone delivery
- Review the plan for the next practical class

Quick Review

Status

- Starting the 2nd week of classes.
- Lectures: **data collection** +? **data preparation** >> data processing
- Practical: groups and topics >> **start working on data collection and preparation**
- Milestone 1 workshop **in 2 weeks**.

Project

- The development of an **information search system**, including work on
  - data collection and preparation,
  - information querying and retrieval,
  - and retrieval evaluation.

- Organized in **three milestones and workshops**:
  - M1: Data Preparation
  - M2: Information Retrieval
  - M3: Search System

Ad Hoc Search

- Focus is on the **ad hoc search task**
  - *standard retrieval task in which the user specifies his information need through a query which initiates a search (executed by the information system) for documents which are likely to be relevant to the user.*
- Examples (consider the collection, the data processing pipeline, the information needs)
  - Google search
  - Desktop search
  - Email search
- Other standard information retrieval tasks:
  - keyword extraction, summarization, question answering, information filtering …
  - not the focus of the project.

Milestone #1: Data Preparation

- **Preparation and characterization** of the datasets selected for the project.
- This task is heavily dependent on the datasets, which may require some extraction actions such as crawling or scraping.
- Work on these tasks depends on the nature, volume, organization and accessibility of the selected datasets. As a result of this milestone, a well-documented and reproducible pipeline of data processing is expected.

- M.IA students use already prepared datasets.

Milestone #1: Sample Actions

- The following list has a **sample of the actions which are required**:
  - search repositories for datasets;
  - select convenient data subsets;
  - assess the authority of the data source and data quality;
  - perform exploratory data analysis;
  - prepare and document a data processing pipeline;
  - characterize the datasets, identifying and describing some of their properties;
  - identify the conceptual model for the data domain;
  - define and characterize the documents in the final collection;
  - **identify follow-up information needs** in the data domain.

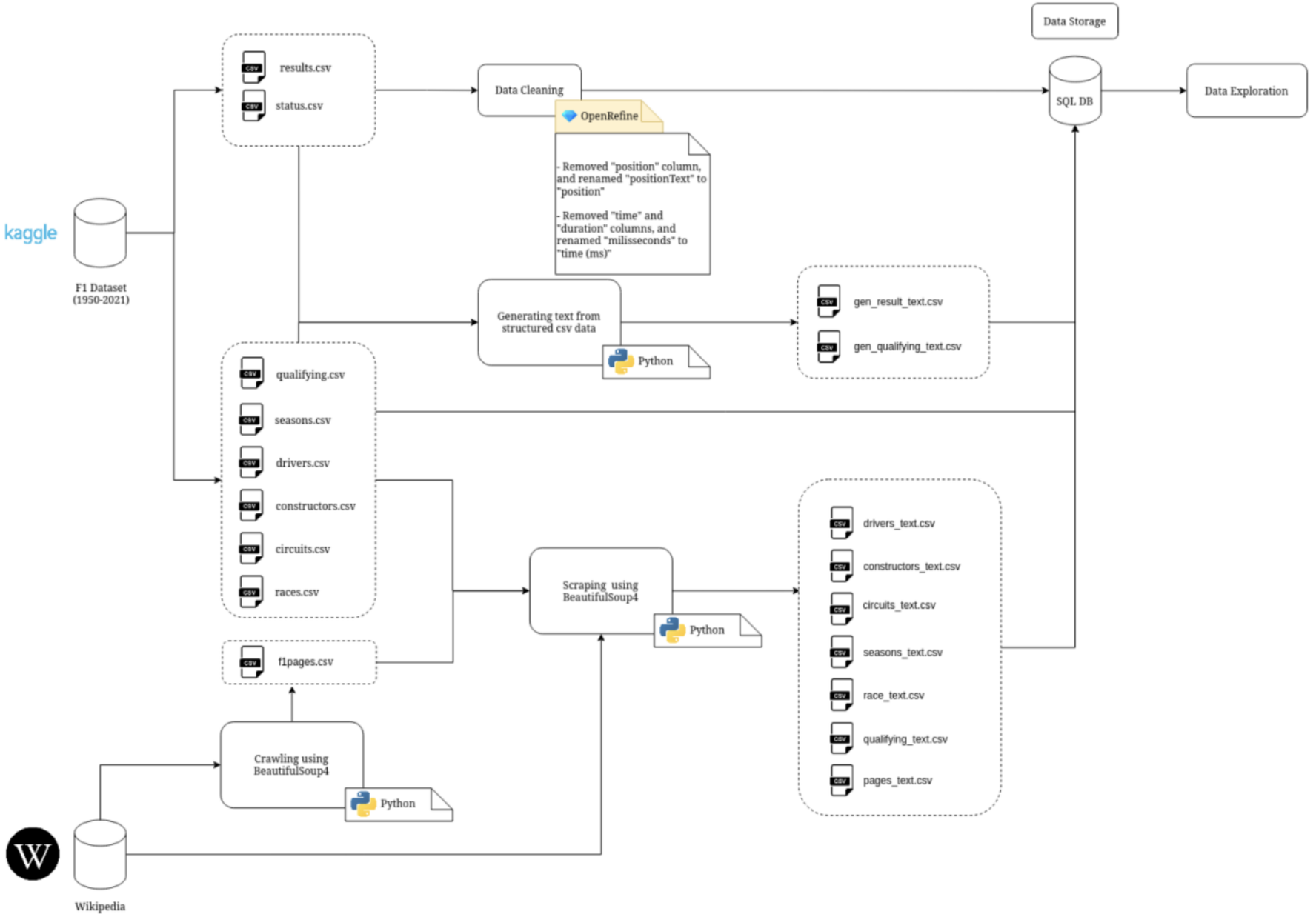
Examples

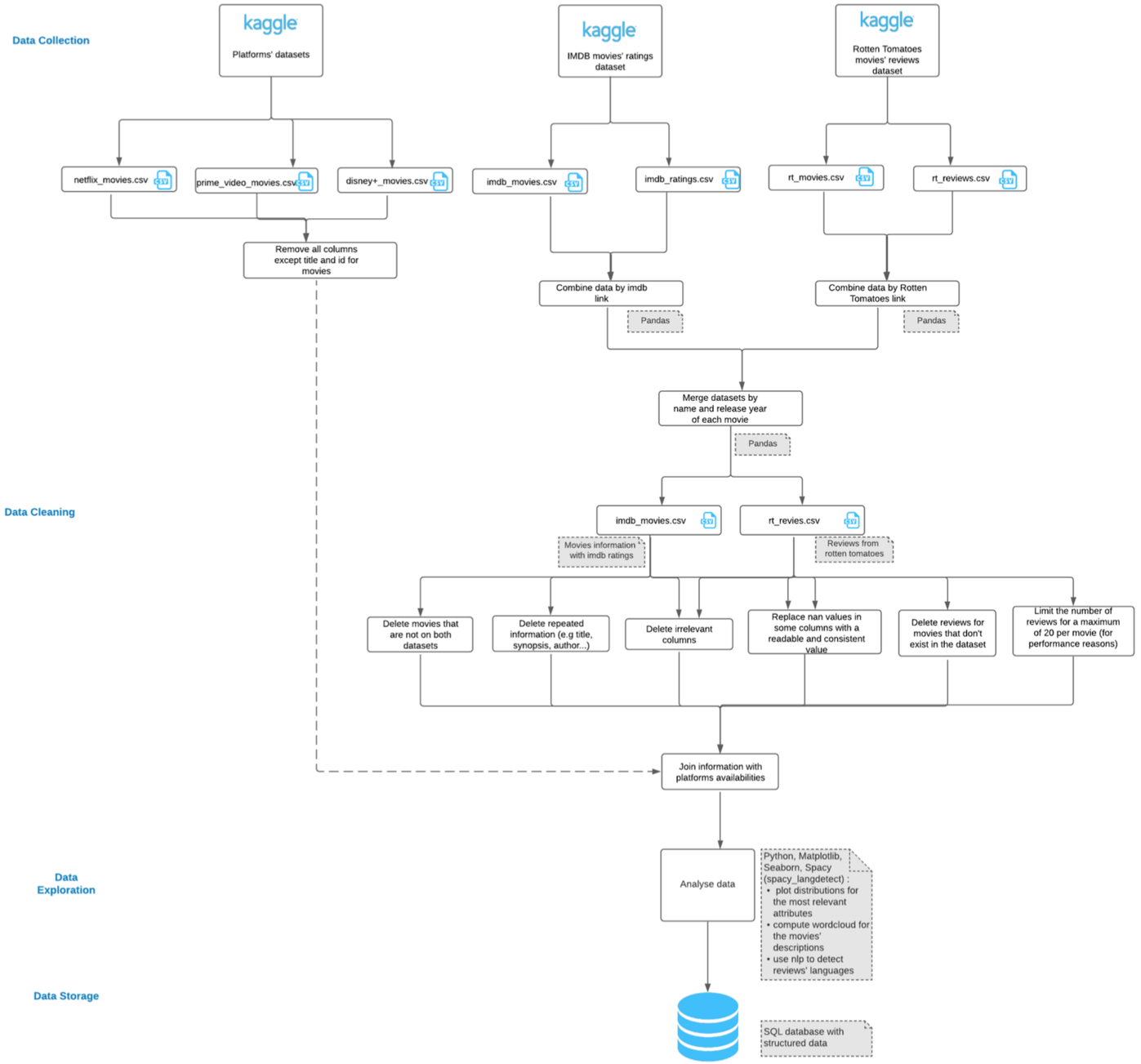
Topic Examples (1)

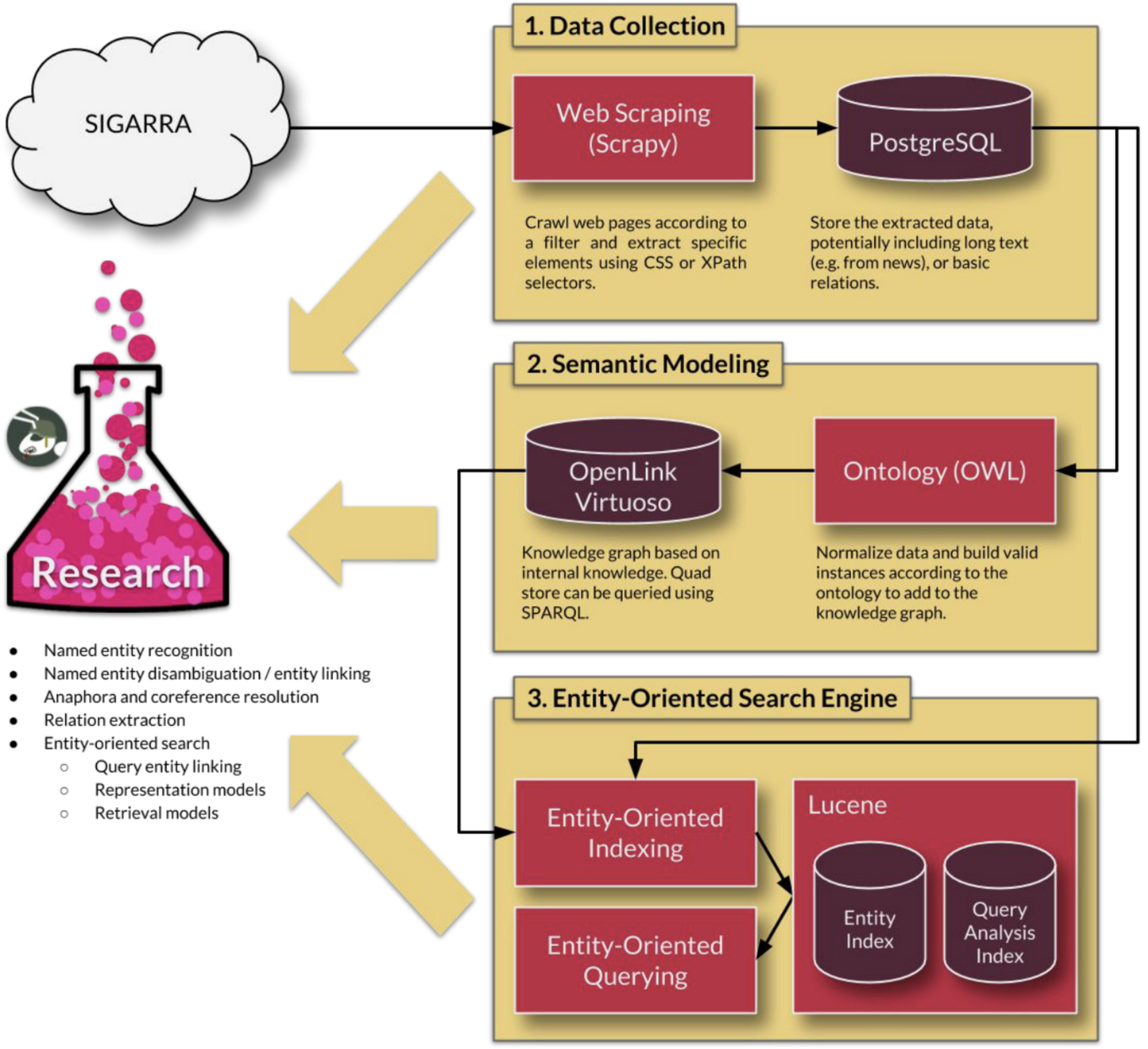
- Search over MIEIC / M.EIC dissertations:
  - Data sources: U.Porto Open Repository
  - (Fallback: scientific articles)
  - Collection: web scraping
  - Processing: PDF parsing
- Search over political parties web pages throughout history:
  - Data source: arquivo.pt API
  - Collection: JSON API requests
  - Processing: parsing HTML

Topic Examples (2)

- Search over movies subtitles:
  - Data sources: existing datasets + Wikipedia / Wikidata
  - Document: individual sentence; other ?
  - Document enrichment with: movies datasets; wikipedia pages, etc.
    - Include speaker / actor, time, movie, director. Many opportunities for filtering.
- Search for European universities:
  - Data sources: existing datasets + Wikipedia / Wikidata
  - Document: university info including city info
  - Document enrichment with: schools and courses?


Data Storage
results.csv
status.csv
Data Cleaning
SQL DB
Data Exploration
OpenRefine
- Removed "position" column, and renamed "positionText" to "position"
- Removed "time" and "duration" columns, and renamed "milisseconds" to "time (ms)"
kaggle
F1 Dataset (1950-2021)
Generating text from structured csv data
Python
gen_result_text.csv
gen_qualifying_text.csv
qualifying.csv
seasons.csv
drivers.csv
constructors.csv
circuits.csv
races.csv
Scraping using BeautifulSoup4
Python
drivers_text.csv
constructors_text.csv
circuits_text.csv
seasons_text.csv
race_text.csv
qualifying_text.csv
pages_text.csv
f1pages.csv
Crawling using BeautifulSoup4
Python
Wikipedia


Data Collection
kaggle
Platforms' datasets
kaggle
IMDB movies' ratings dataset
kaggle
Rotten Tomatoes movies' reviews dataset
netflix_movies.csv
prime_video_movies.csv
disney+_movies.csv
imdb_movies.csv
imdb_ratings.csv
rt_movies.csv
rt_reviews.csv
Remove all columns except title and id for movies
Combine data by imdb link
Pandas
Combine data by Rotten Tomatoes link
Pandas
Merge datasets by name and release year of each movie
Pandas
Data Cleaning
imdb_movies.csv
rt_revies.csv
Movies information with imdb ratings
Reviews from rotten tomatoes
Delete movies that are not on both datasets
Delete repeated information (e.g title, synopsis, author...)
Delete irrelevant columns
Replace nan values in some columns with a readable and consistent value
Delete reviews for movies that don't exist in the dataset
Limit the number of reviews for a maximum of 20 per movie (for performance reasons)
Join information with platforms availabilities
Data Exploration
Analyse data
Python, Matplotlib, Seaborn, Spacy (spacy_langdetect) :
• plot distributions for the most relevant attributes
• compute wordcloud for the movies' descriptions
• use nlp to detect reviews' languages
Data Storage
SQL database with structured data


SIGARRA
1. Data Collection
Web Scraping (Scrapy)
PostgreSQL
Crawl web pages according to a filter and extract specific elements using CSS or XPath selectors.
Store the extracted data, potentially including long text (e.g. from news), or basic relations.
2. Semantic Modeling
OpenLink Virtuoso
Ontology (OWL)
Knowledge graph based on internal knowledge. Quad store can be queried using SPARQL.
Normalize data and build valid instances according to the ontology to add to the knowledge graph.
3. Entity-Oriented Search Engine
Entity-Oriented Indexing
Entity-Oriented Querying
Lucene
Entity Index
Query Analysis Index
Research
Named entity recognition
Named entity disambiguation / entity linking
Anaphora and coreference resolution
Relation extraction
Entity-oriented search
Query entity linking
Representation models
Retrieval models

Lab #2: Class Plan

- Data collection and processing.
- Tools exploration and evaluation (OpenRefine, Apache Tika, etc).
- Work on practical tutorials #1 (command line) and #2 (datasets).
- Continue work on project milestone 1.
  - Clear vision for the documents and information needs.
  - Obtaining datasets from the domain chosen for practical work.
  - Datasets storage.
  - Conduct exploratory data analysis.
  - Conceptual domain modeling.

Data Preparation

Data Preparation

- Real-word 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

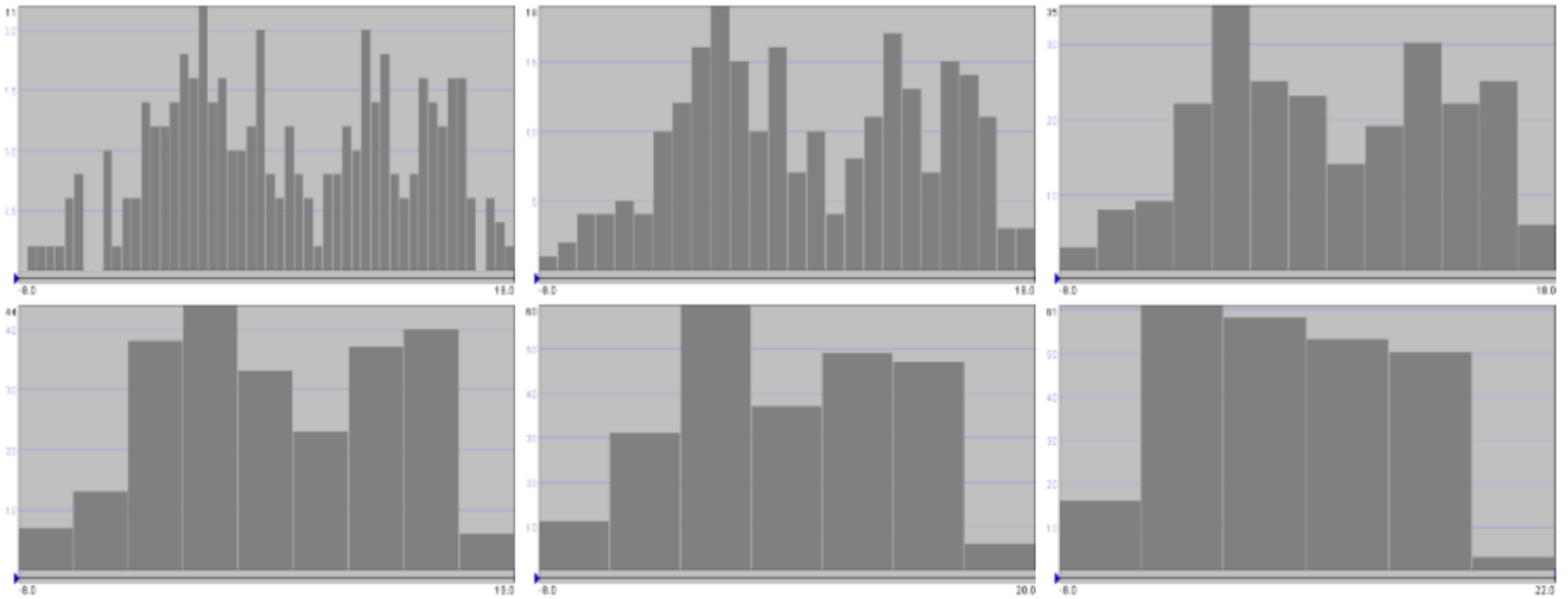


Fig. 3.13: Frequency histograms of the temperature data with the bin sizes equal to 0.5, 1, 2, 3, 4, and 5 degrees Celsius, correspondingly (left to right, top to bottom).

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 exiting 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 processes, e.g. observe intermediate data structures, track the data pipeline execution.
  - After the application of computational methods to evaluate the quality if the results, compare the results between different executions of the pipeline.

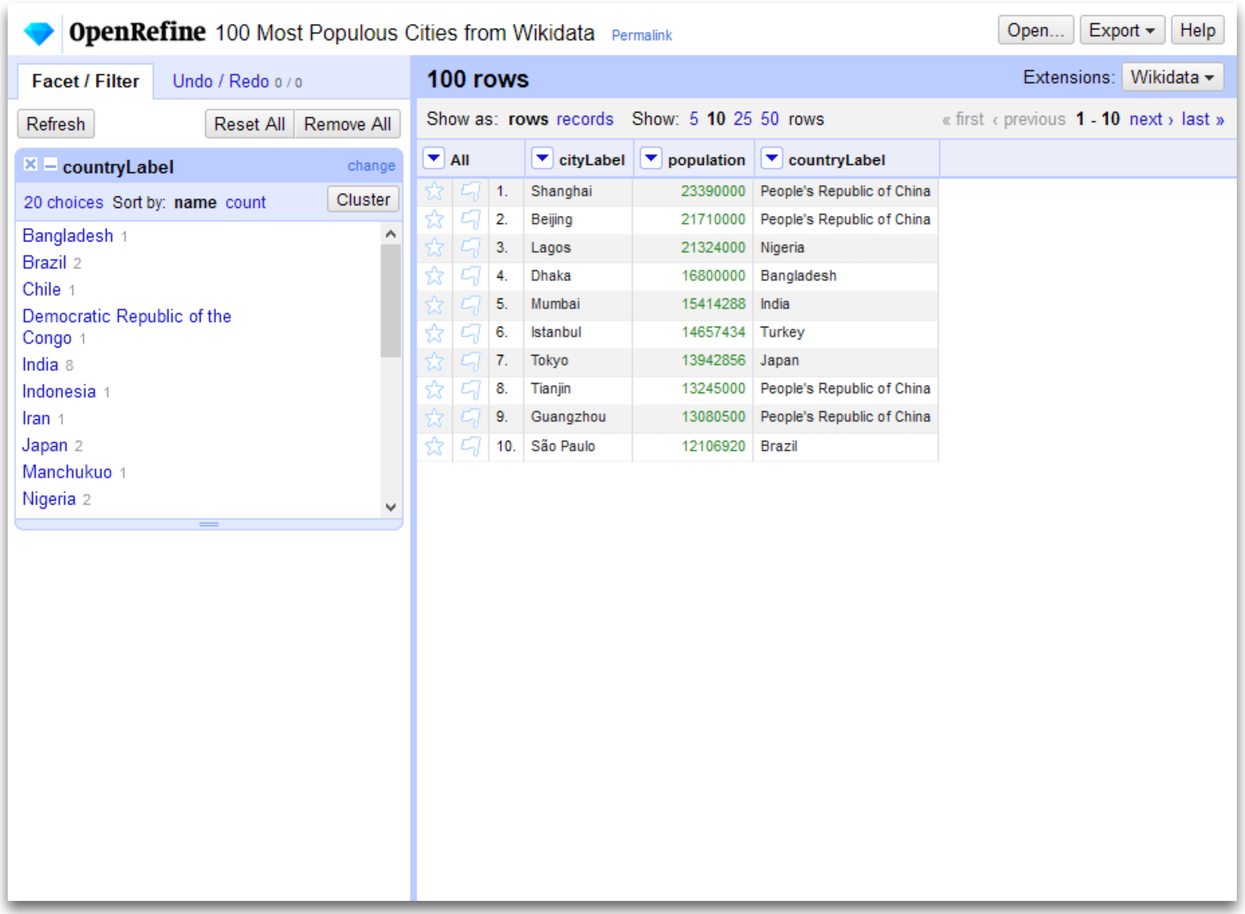
Tools for Data Preparation

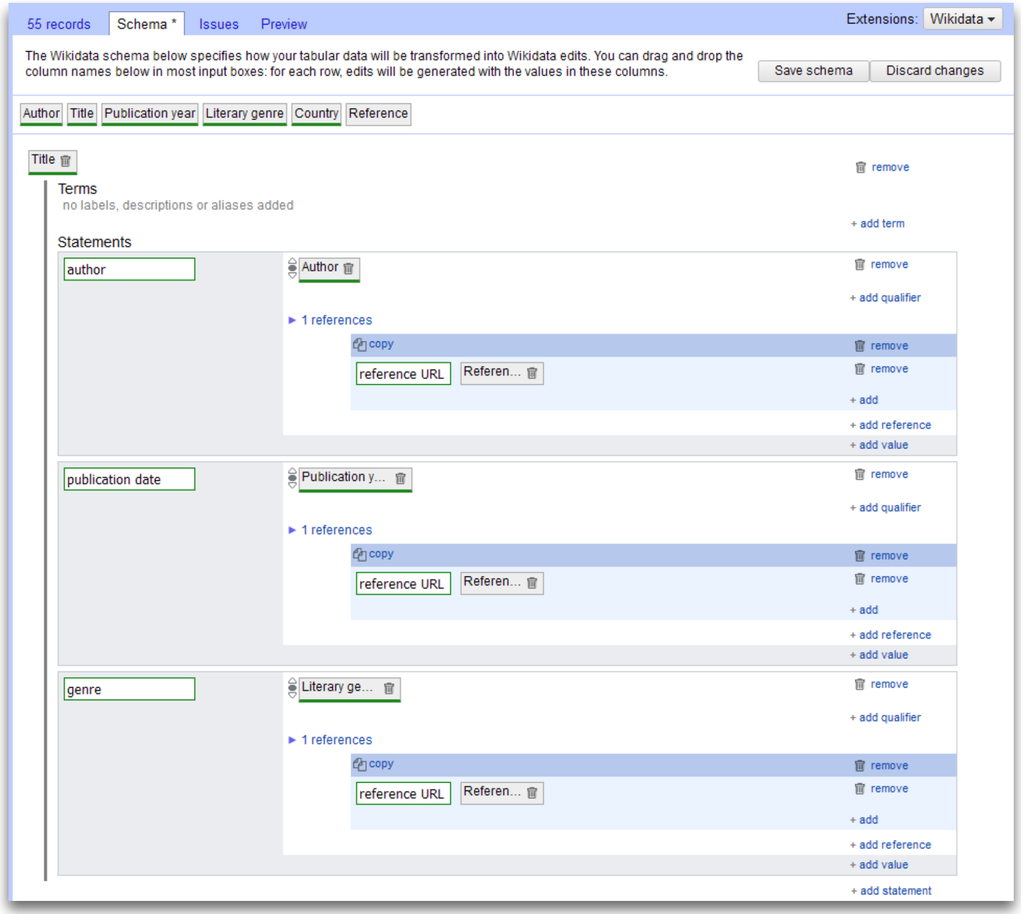
Tools for Data Cleaning

- 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)
  - 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 »
All
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
remove
reference URL
Referen...
remove
+ add
+ add reference
+ add value
publication date
Publication y...
remove
+ add qualifier
▸ 1 references
copy
remove
reference URL
Referen...
remove
+ add
+ add reference
+ add value
genre
Literary ge...
remove
+ add qualifier
▸ 1 references
copy
remove
reference URL
Referen...
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
  - Beautiful Soup (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.
- However, the **risk of lock-in** is central and must be taken into consideration.

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 example: obtain data from a REST API and load it into a relational database;
  - Complex example:
    - obtain data from multiple web pages at regular intervals;
    - parse HTML and extract main keyword from each page;
    - automatically identity main trending topics using a previously trained machine learning model;
    - update model with new data (keep an archive of previous models);
    - 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 (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

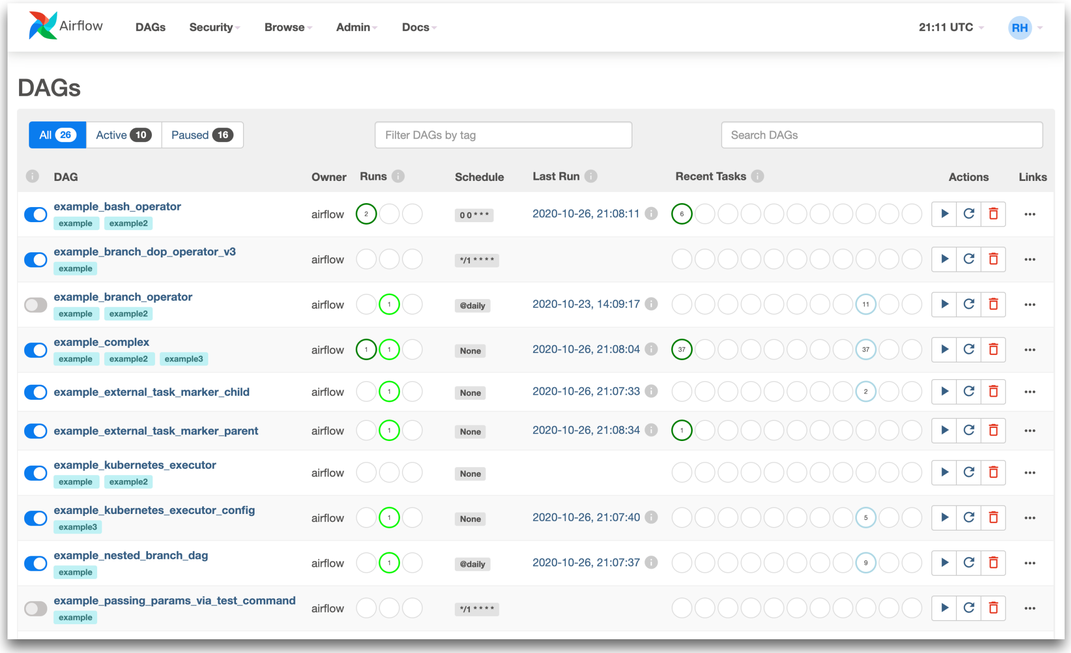
Makefiles

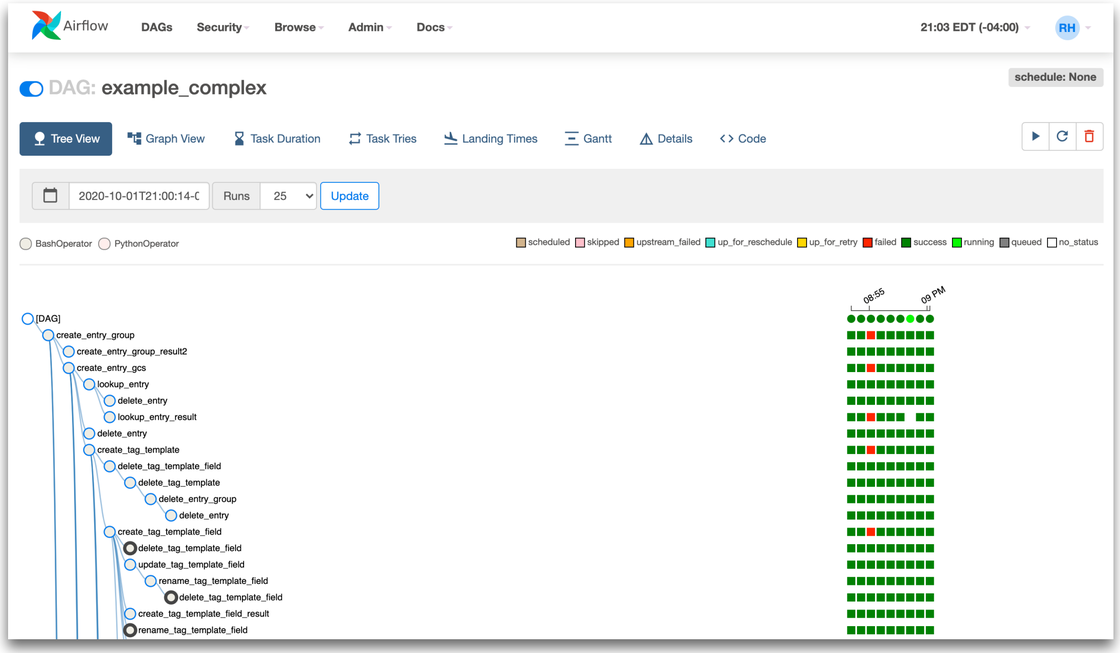
- Makefiles are used to automate software build processes, by defining targets and rules to execute. The underlying abstraction is of a dependency graph, where tasks depend on the execution of other tasks.
- Make language agnostic, and implementations exist for most operating systems.
- Can be used to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

- Not mandatory for the PRI project, but a good practice.
- A tutorial is available to explore in lab classes.

Apache Airflow

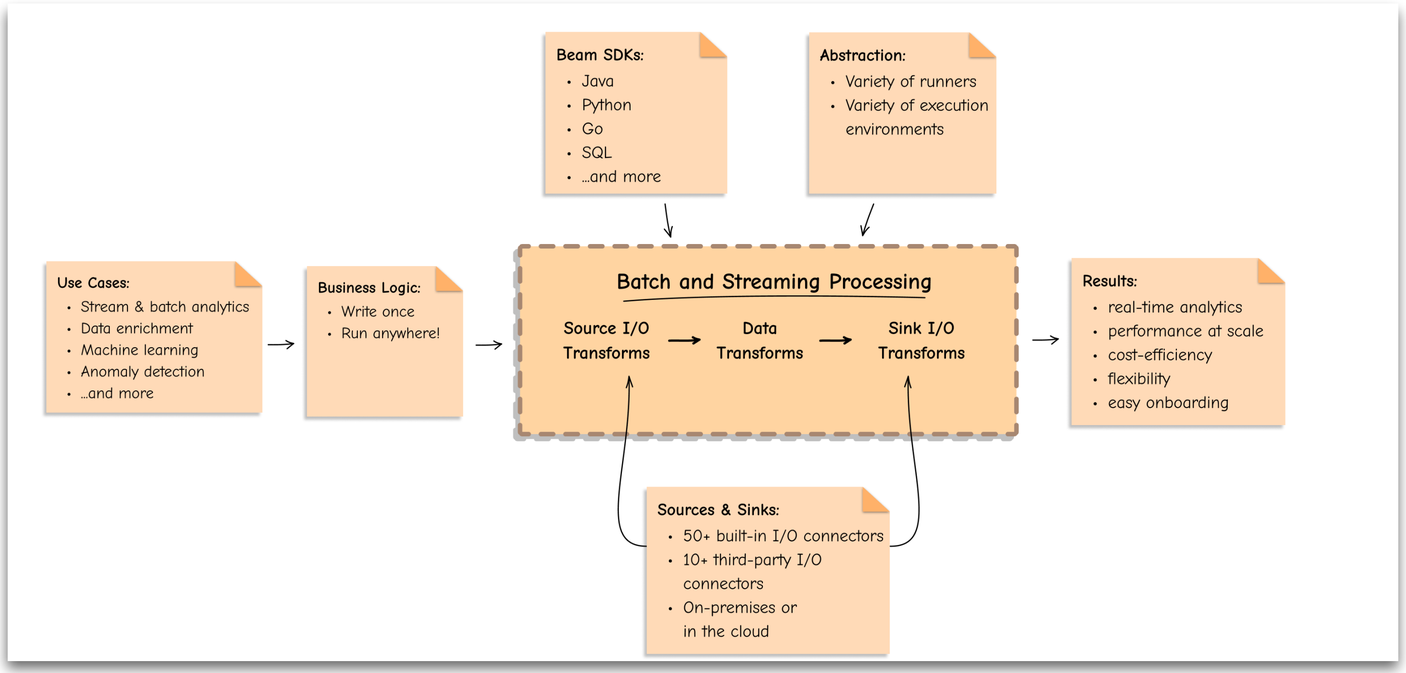
- An open-source platform, to "programmatically author, schedule and monitor workflows" [airflow.apache.org]. Supports Python and Bash scrips.





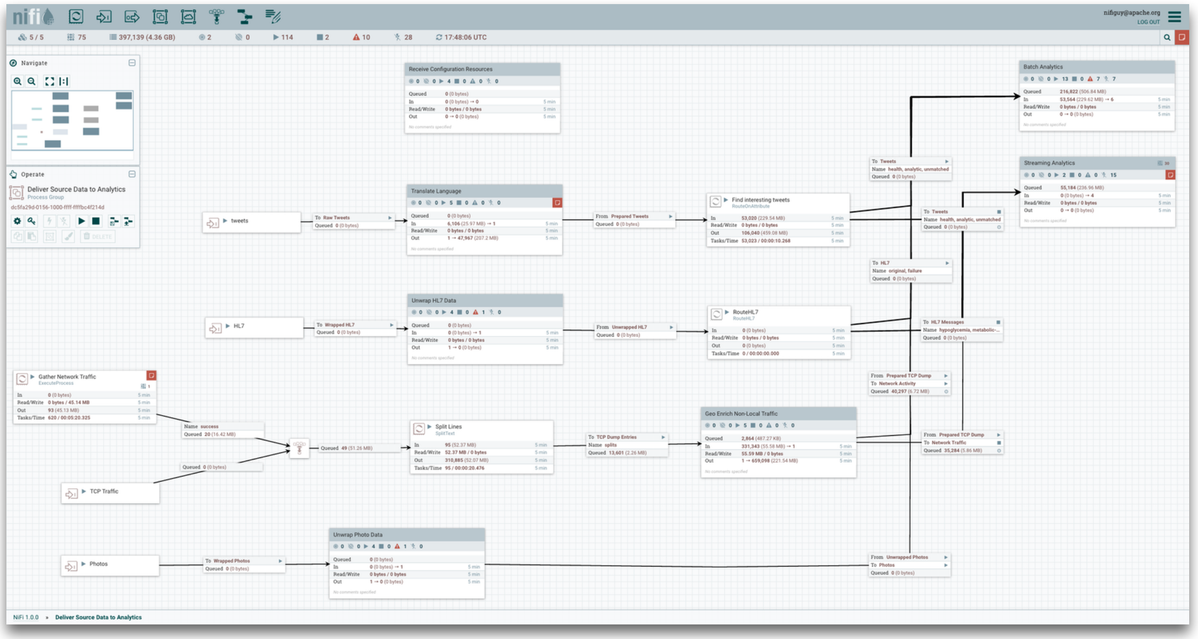
Apache Beam

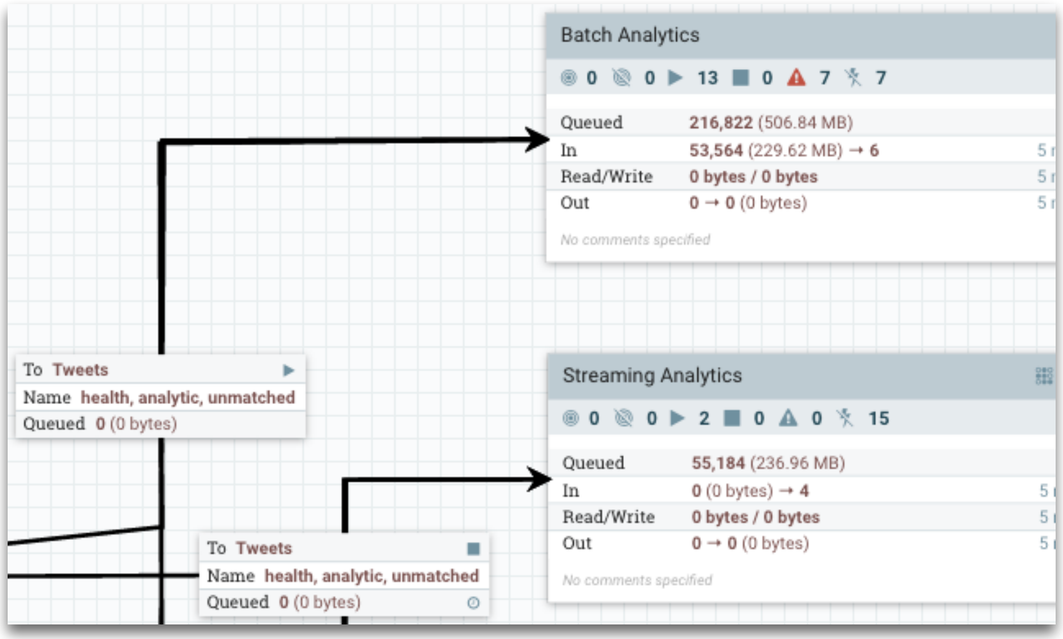
- "Apache Beam is an open-source, unified programming model for batch and streaming data processing pipelines that simplifies large-scale data processing dynamics." [beam.apache.org]



Apache NiFi

- "Apache NiFi is designed to automate the flow of data between software systems, leveraging the ETL pattern (extract, transform, load).” [nifi.apache.org]





Data Streams Processing

- **Apache Kafka**, kafka.apache.org
  - "An open-source distributed event store and stream-processing platform, providing a unified, high-throughput, low-latency platform for handling real-time data feeds."
- **Apache Flinks**, flink.apache.org
  - "A framework and distributed processing engine for stateful computations over unbounded and bounded data streams, designed to run in all common cluster environments, perform computations at in-memory speed and at any scale.”

Documentation of Data Pipelines

Data Documentation

- Documentation is central in any engineering process.
- It distinguishes between ad-hoc processes and repeatable, inspectable, shareable processes.
- It is essential to document:
  - Data selection and sampling criteria, sources, filtering, etc.
  - 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.

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