Data Collection

PRI 23/24 · 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 2 — were officially introduced: kiwi (Ki), mebi (Mi), gibi (Gi), tebi (Ti), pedi (Pi), exbi (Ei), zebi (Zi), yobi (Yi).
  - 1MiB (2^20 = 1024^2) = 1 048 576 bytes
  - 1MB (1000^2) = 1 000 000 bytes

"Binary prefix" on Wikipedia: https://en.wikipedia.org/wiki/Binary_prefix

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

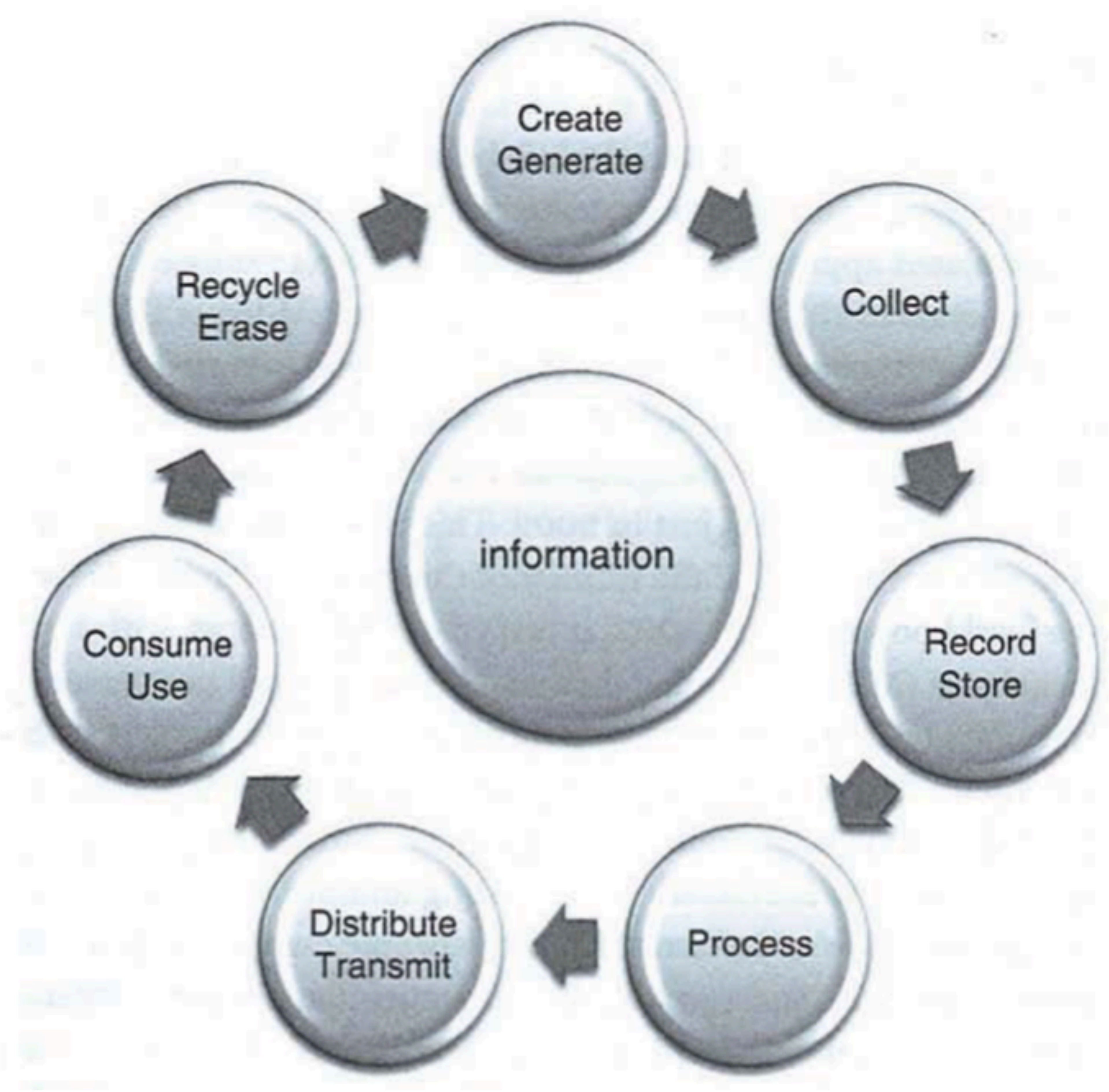


Image from Floridi, L. Information - A Very Short Introduction (2010).

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% were 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.

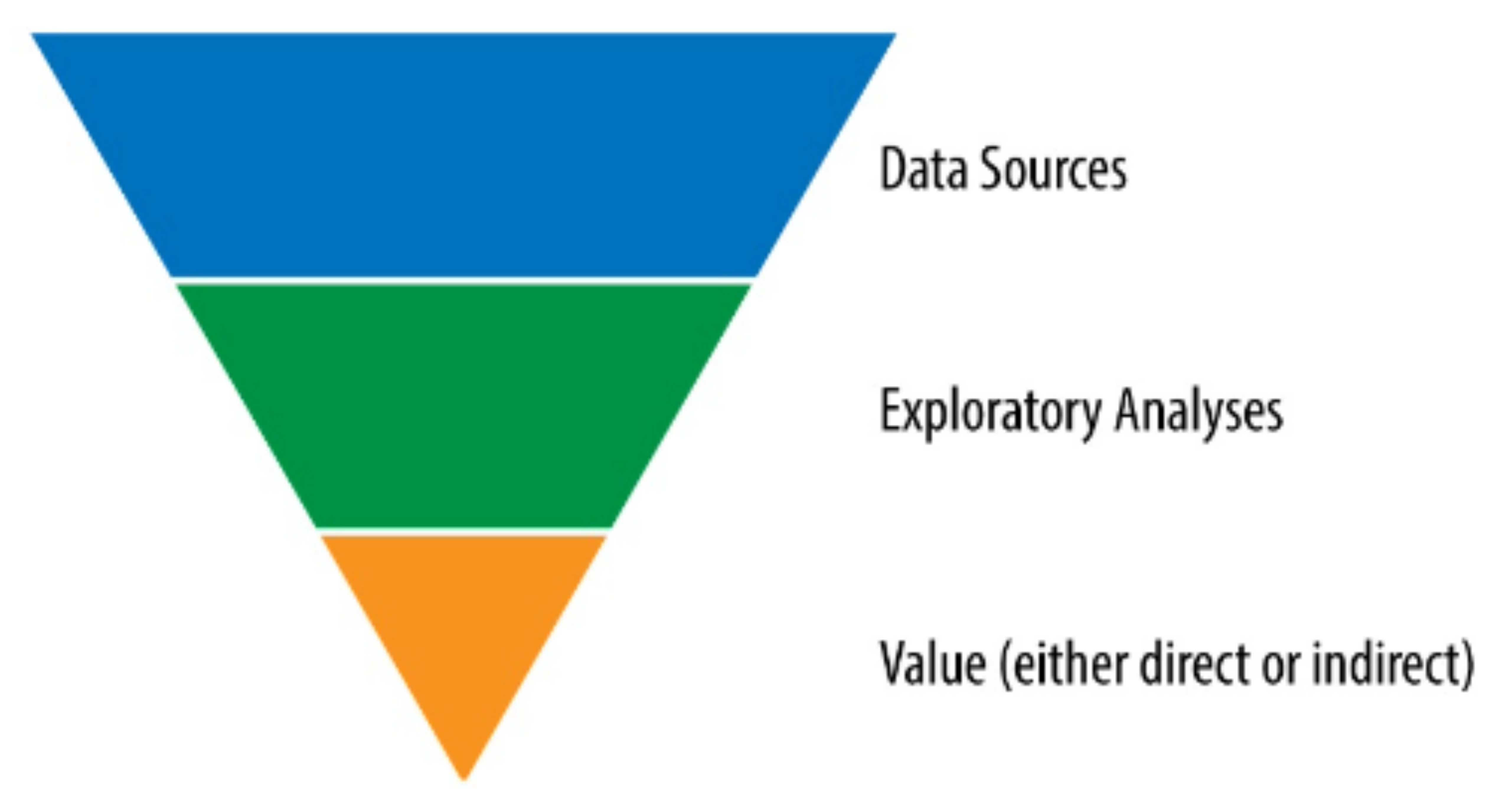


Image from Densmore, J. Data Pipelines Pocket Reference (2021).

Increasing Data Value

- Make data available, i.e. simply make previously unaccessible 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

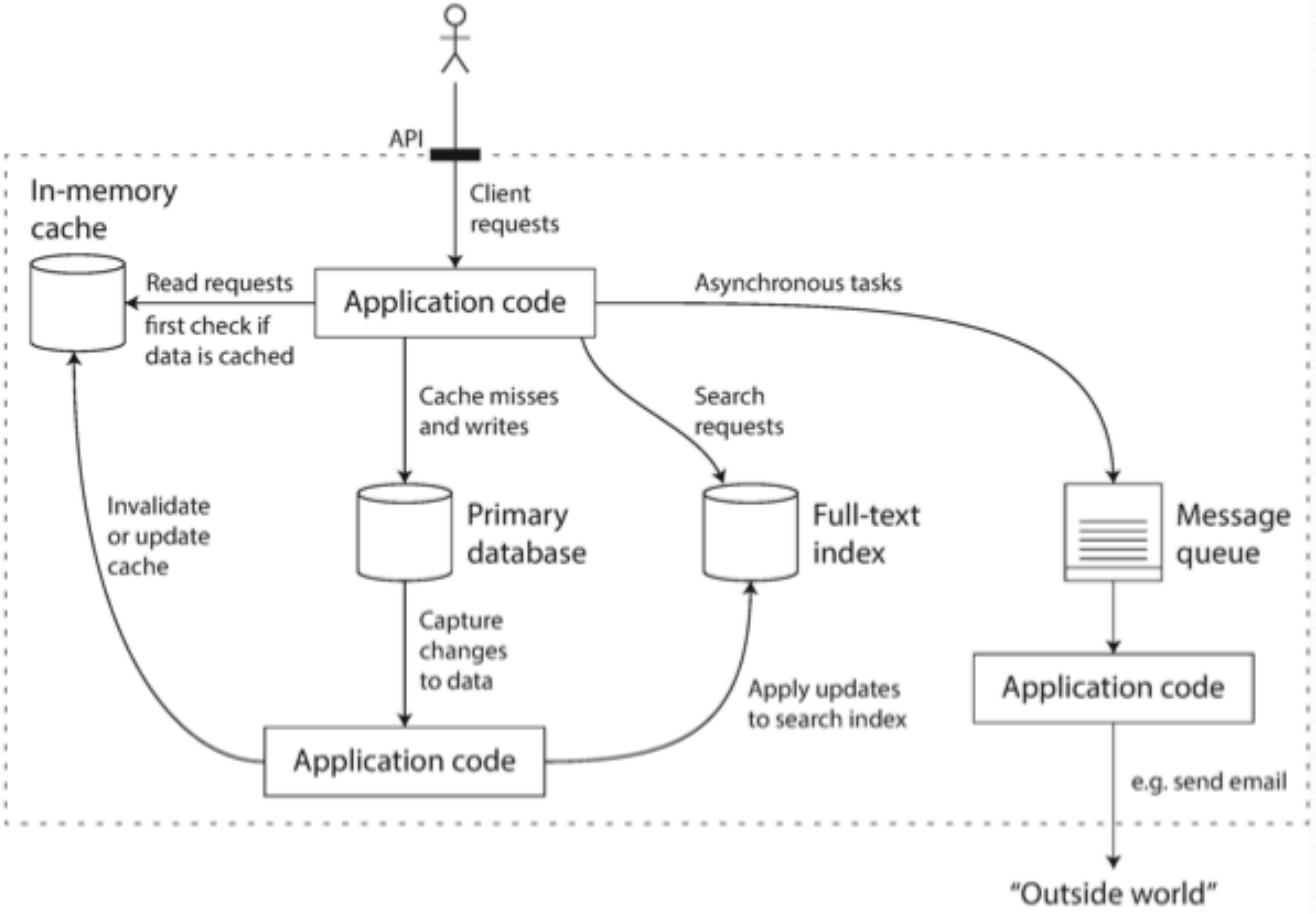


Image from Kleppmann, M. Designing Data-Intensive Applications (2017).

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 pattern 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 simples 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.
- Load-transform, in contrast with transform-load, is a pattern more well-suited to the division of responsibilities in multidisciplinary teams.
- 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.

ELT Pattern

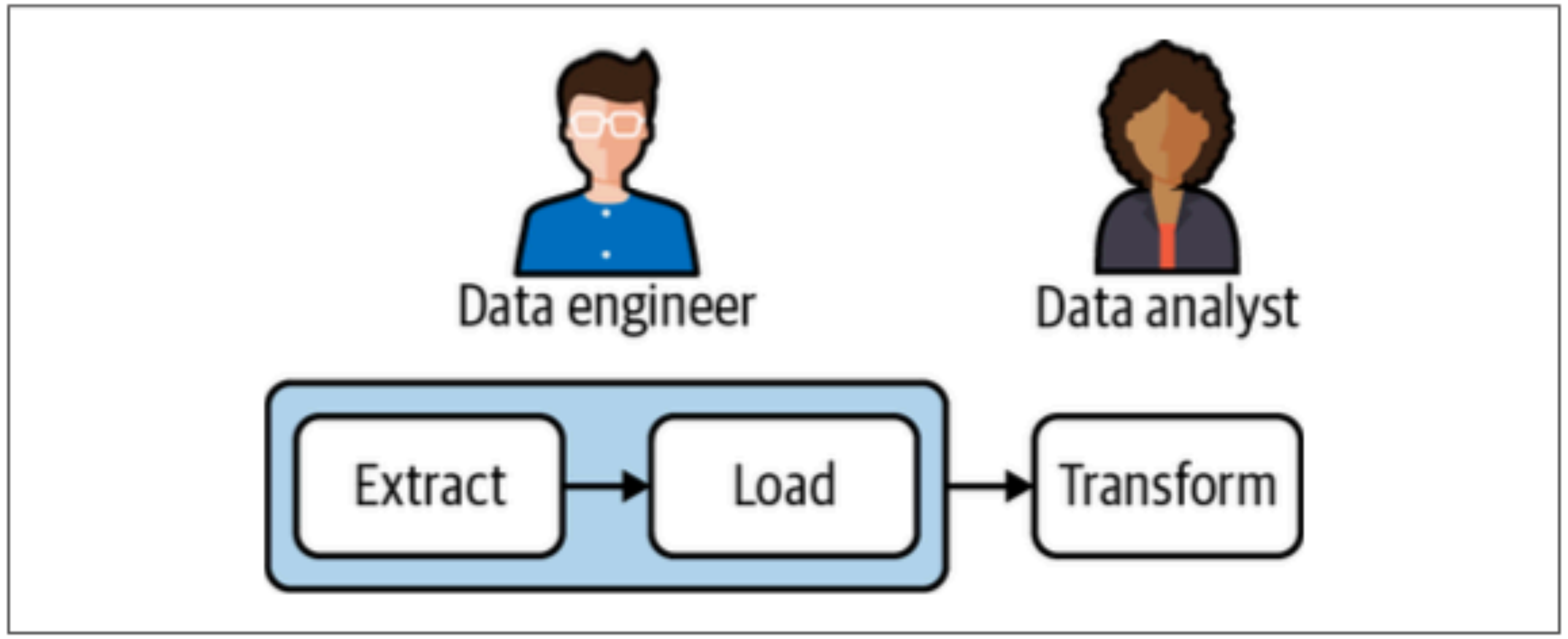


Figure 3-3. The ELT pattern allows for a clean split of responsibilities between data engineers and data analysts (or data scientists). Each role can work autonomously with the tools and languages they are comfortable in.

Image from Densmore, J. Data Pipelines Pocket Reference (2021).

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-word processes are typically non-linear.

Iterative Process

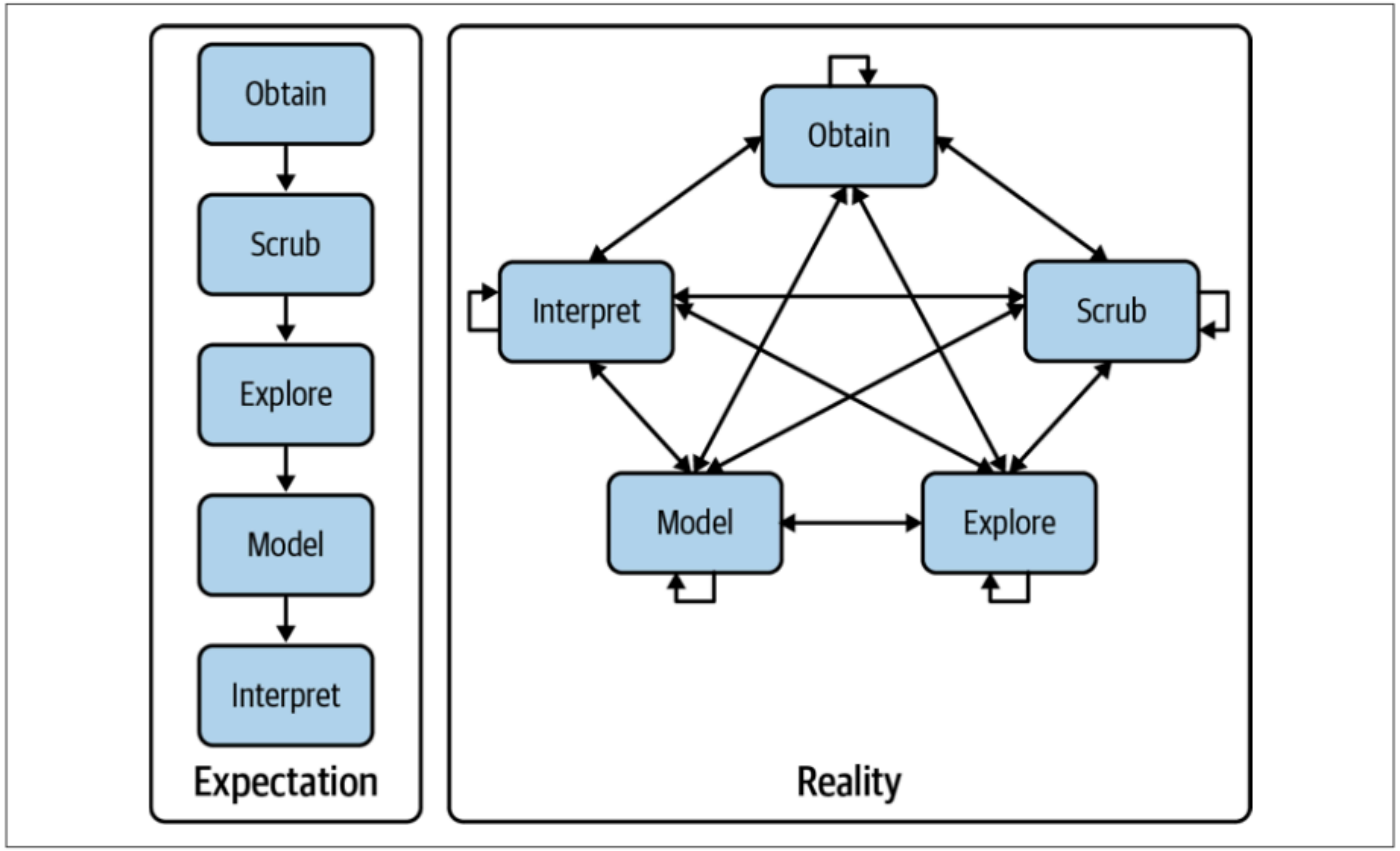


Figure 1-1. Doing data science is an iterative and nonlinear process

Image from Janssens, J. Data Science at the Command Line (2021).

Data Analytics Process in Organizations

- Data analytics processes can either be centralized, typically in an IT department, or decentralized, in specialized teams.
- Benefits of centrally controlling data processing:
  - Controlled data governance;
  - Efficiency gains due to reuse of practices, methods, and expertise.
- Drawbacks of centralization:
  - Frequent bottlenecks due to the time taken to dependency upon the IT department
- Challenge: expanding the range of users who have access to raw data and provide them with the necessary training and skills.

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 coined 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 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/
- *See Moodle for a list of selected data sources and example 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 and 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;
  - CSV (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, BJSON, etc.
- Apache Thrift (originally Facebook) and Protocol Buffers (protobuf) are open-source binary encoding libraries that produce a binary encoding of a given record.

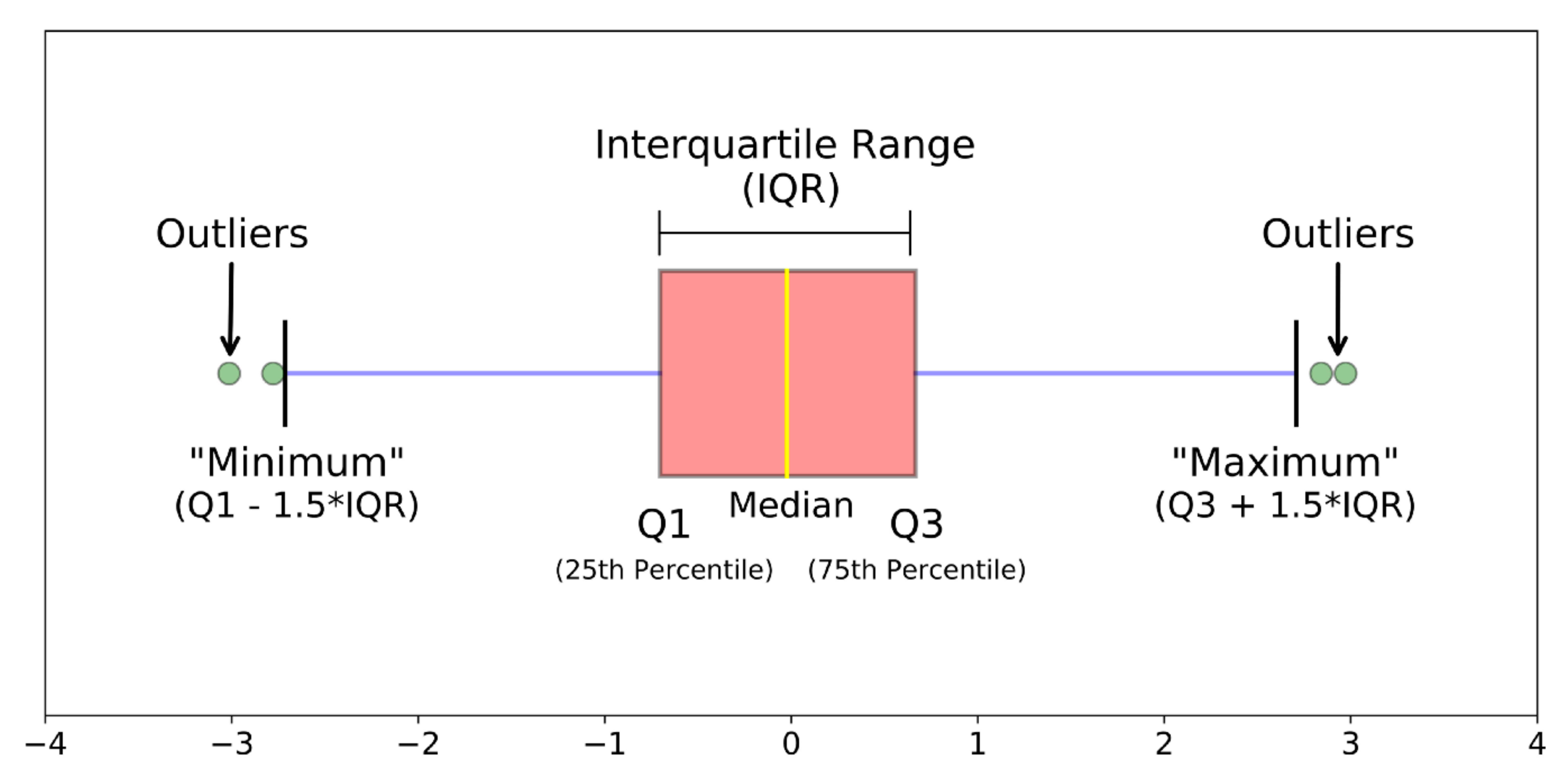
Data Quality

- 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.

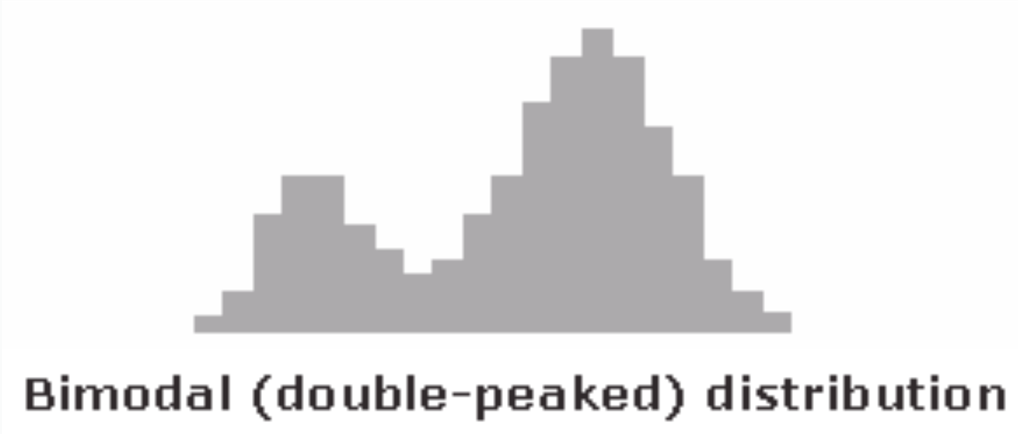
Box plots



- 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.



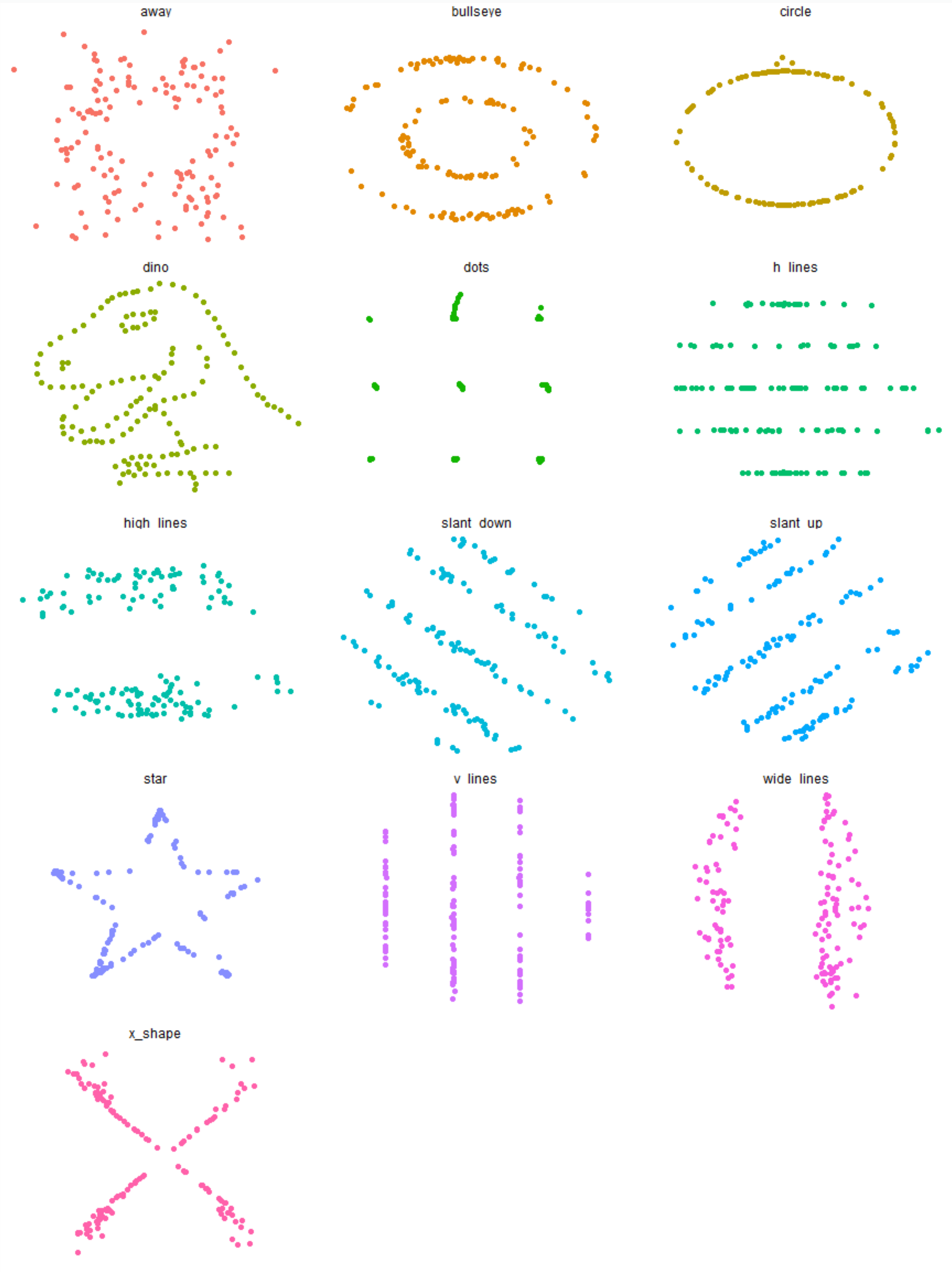
- Examples from "What are Histograms?", ASQ [https://asq.org/quality-resources/histogram]

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 if 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 of 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, wget.
- Web crawling platforms:
  - Scrapy, https://scrapy.org/ [Python]
  - Internet Archive Heritrix, https://github.com/internetarchive/heritrix3 [Java]
  - Apache Nutch, http://nutch.apache.org [Java]

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?