

# Introduction to Data Science

WS 23/24, RWTH Aachen

October 31, 2023

## Contents

<b>1 Basics of data science</b>	<b>2</b>
1.1 Data science pipeline . . . . .	2
1.2 Types of data . . . . .	4
1.3 Supervised and unsupervised learning . . . . .	6
1.4 Data science process . . . . .	7
1.5 Challenges . . . . .	11
<b>2 Data visualization and exploration</b>	<b>13</b>
2.1 Data extraction . . . . .	13
2.2 Characterizing individual features . . . . .	15
2.3 Data quality . . . . .	21
2.4 Showing relations among features . . . . .	24
2.5 Preparing for analysis . . . . .	24
2.6 Good and poor visualizations . . . . .	24



# Introduction

Let's first introduce the general term of data science. It is a new and important discipline that can be viewed as:

- An amalgamation of classical disciplines such as statistics, data mining, databases, and distributed systems,
- With additional new challenges constantly emerging and making the field highly dynamic and appealing.

The problems grow in terms of size ("Big data") and complexity of the questions to be answered. But the basic job can be summarized as:

- Input: data  $\Rightarrow$  Processed by data scientist (with tools)  $\Rightarrow$  Output: value
- Where the skills of a data scientist are the combination of open mind, human interest, analytical skills, creativity, business-benefiting weighting, ...
- Or in other terms as can be seen in 0.1

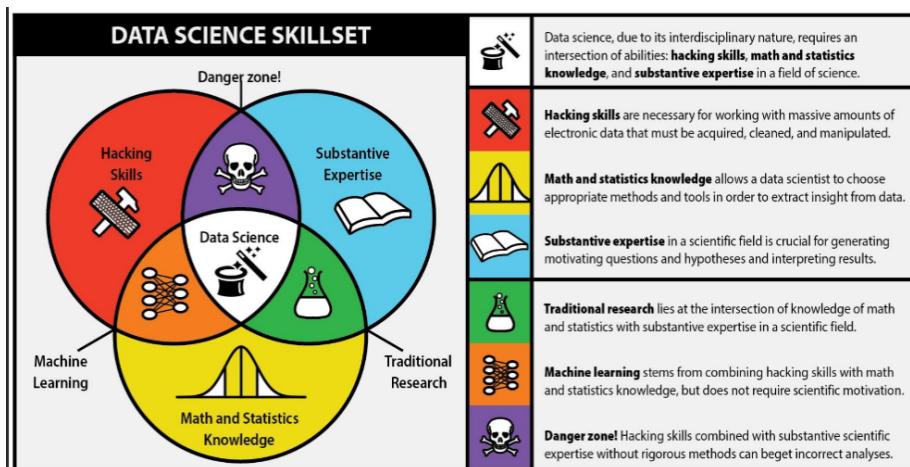


Figure 0.1: Skillset of a data scientist

With the growing importance of data and digitalization, organizations are looking for data scientists, maybe outnumbering computer scientists in the future. Important is the ability to handle data in any form, so basically the need for an all-around skilled "data wizard". This importance can be further highlighted when looking at the tech-development over the past 20 to 30 years. While the hardware got tremendously cheaper, faster, and more compact (20 times faster for MIP = mixed integer programs), also software has progressed in terms of speed (50 times faster for MIP). Interesting to look at is also the aspect of automation.

Dimensions of data science are:

- The different types of data (structured or unstructured, text, images, events, ...)
- The different types of tasks (supervised or unsupervised, ...)
- Human versus machine (Who does what?)
- Algorithm versus visualization (What is needed?)
- Flexibility versus usability

- Scalability versus quality (exact versus heuristics)
- Responsibility versus utility (accuracy and precision versus fairness, privacy, transparency, ...)

Besides raw data science, interesting to look at is also the connection to process science. The interplay between process and data science (PADS) leads to the term process mining. Imagine the connection as shown in 0.2.

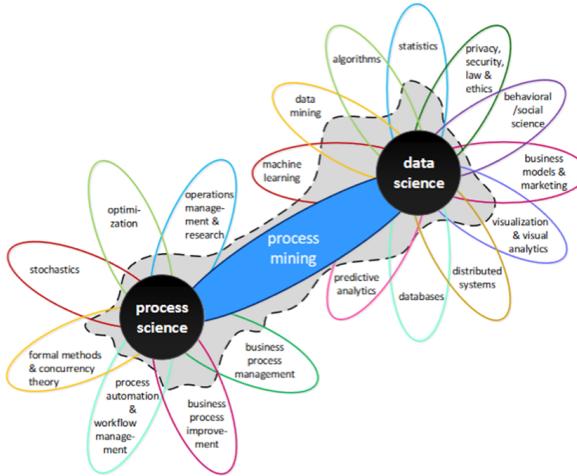


Figure 0.2: Interplay between process science and data science

As the final part of the introduction, we will now see the general covered topics in this course:

- Basic data exploration and visualization
- Decision trees, regression, support vector machines
- Neural networks, evaluation of supervised learning problems, clustering
- Frequent items sets, association rules, sequence mining, process mining, text mining
- Data preprocessing, data quality and binning, visual analytics and information visualization
- Responsible data science
- Big data technologies

# 1 Basics of data science

## 1.1 Data science pipeline

First, we are going to look at how data is processed in terms of the **data science pipeline** as it can be seen in 1.1.

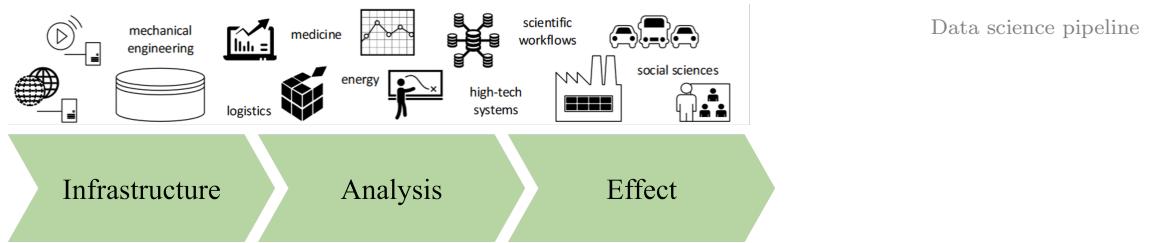


Figure 1.1: Pipeline of data science

Let's look at the individual components. The first step to pay attention to when wanting to handle data is the **infrastructure** with the keywords "**volume and velocity**". The main challenge is making things scalable and instant (responsiveness). Important terms are for example:

- Instrumentation
- Big data infrastructures, distributed systems
- Data engineering (databases and data management)
- Programming
- Security

Next, we have the step of the actual **analysis** concerned with **extracting knowledge** from data. The core challenge can be put as providing answers to known and unknown unknowns. Important terms are for example:

- Statistics, algorithms
- Data and process mining
- Machine learning, artificial intelligence
- Operations research
- Visualization

Finally, we also need to be concerned with the **effect** of our results on people, organizations, and society. The main challenge of this pipeline step is to do **responsibly** perform data handling. Important terms are for example:

- Ethics and privacy, and IT law
- Human-technology interaction
- Operations management
- Business models, entrepreneurship

This course will look into all the steps of the pipeline, but the main focus lies on the data analysis.

## Four generic data science questions

Important to answering all these questions is to keep attention to all three pipeline steps, so not only what analysis we need to perform to answer them, but also how we collect our input (data) and how to deal with our output (result).

Nonetheless, here are the four generic data science questions, with variety in terms of

difficulty and predicting the future:

1. **What** happened?
2. **Why** did it happen?
3. What will happen in the **future**?
4. What is the **best** that can happen?

## 1.2 Types of data

Now that we know that we have some kind of data as our input, we need to take a look at what this data can look like. Generally speaking, there are two types:

- |                   |  |
|-------------------|--|
| Structured data   | • Structured data like age, time, gender, class, etc., and |
| Unstructured data | • Unstructured data like text, audio, video, etc.          |

For **structured data** we have a further subdivision into structured data types. The data types depicted in 1.2 will be described in detail.

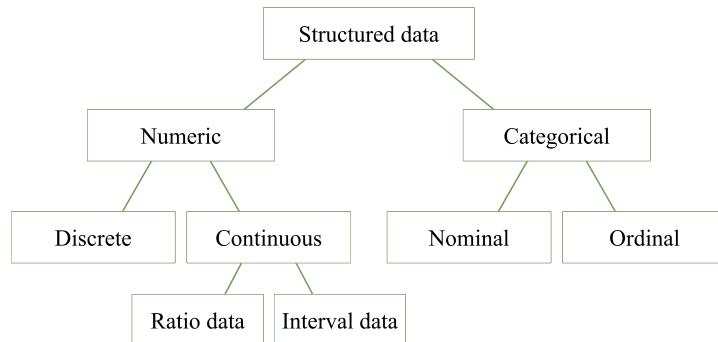


Figure 1.2: Overview structured data types

- |                  |  |
|------------------|--|
| Categorical data | • <b>Categorical</b> data can be stored and identified based on names or labels given to them and is also known as "qualitative" data. Matching can be applied, where data is grouped based on similarities.   |
| Nominal data     | • Concretely, <b>nominal</b> data or naming data has a label and its characteristic similar to a noun and doesn't imply an order. (Example: name, color=red, country=NL)   |
| Ordinal data     | • <b>Ordinal</b> data on the other hand is ranked, ordered, or used on a rating scale. This means, you can count and order ordinal data but are not able to measure it. (Example: risk=medium, score=good)   |
| Numerical data   | • In contrast to categorical data, we also have <b>numerical</b> data referring to data in the form of numbers instead of another language or descriptive form. It is also known as "quantitative" data. Important is the ability to be statistically and arithmetically calculated (allowing for $+, -, >, =, \dots$ ). |
| Discrete data    | • One subtype of numerical data is <b>discrete</b> data representing countable items, that are collected in a list (finite or infinite). (Example: number of items=5, age=18)  |
| Continuous data  | • Then, there's also <b>continuous</b> data in the form of intervals or ranges. The data represents measurements with their intervals falling on a number line (so counting isn't involved).   |
| Interval data    | • Continuous data can now be further distinguished. One subtype is <b>interval</b> data where the data can be measured only along a scale at equal distances from each   |

other, so only addition and subtraction operations are allowed. There is no true zero (and hence no  $,$   $/$ ). (Example: `data=11-11-2018, temp=18.5°C`)

- And finally, we have **ratio** data describing measurement with a defined (true) zero point. (Example: `dropout=33%, speed=128.34km/h`) Ratio data

For **unstructured data**, we just take the raw data and interpret it as a stream of bits. This goes for text, audio, images, signals, and videos exactly the same. Examples can be seen in 1.3.

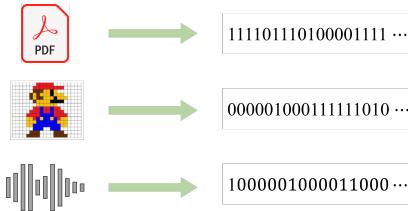


Figure 1.3: Input for unstructured data

Data can now be stored and ordered together by putting it into **tables**. Concretely, columns represent different features (can be different kinds of data types) whereas rows describe data instances (also known as individuals, entities, cases, objects, or records). Examples can be seen in 1.4.

feature				Tabular data			
Order id	Product	Price	Date	From	To	Message	Image
32424	718 Cayman	66.000	21-10-2018	Sue	Peter	"How are you?"	
34535	911 Carrera	102.000	22-10-2018	Peter	Sue	"Very good!"	
43555	911 Turbo	154.000	24-10-2018	Peter	Mary	"Let's go out."	
...	...	...	...	...	...	...	...

Ordinal
Nominal
Ratio data
Interval data
Nominal
Unstructured

Figure 1.4: Table data with data types

Features can now be raw or derived (e.g. `max, min, average, rank, bin, ...`). An important aspect is time, as it cannot decrease and we usually want to predict the future based on the past. Features

An important distinction to be made when it comes to tabular data is whether the items are labeled or not.

- In case of labelled data we have **descriptive features** and a **target feature**.
  - The descriptive features are also known as predictor variables or independent variables. Labelled data
  - Alternative names for target features are response variable, dependent variable, or also label. Descriptive features
- Unlabelled data on the other hand doesn't have a selected target feature. Target feature Unlabelled data

### 1.3 Supervised and unsupervised learning

Derived from the different kinds of tabular data, we have two fundamental learning paradigms. Exemplary input data and possible results for both paradigms can be seen in 1.5.

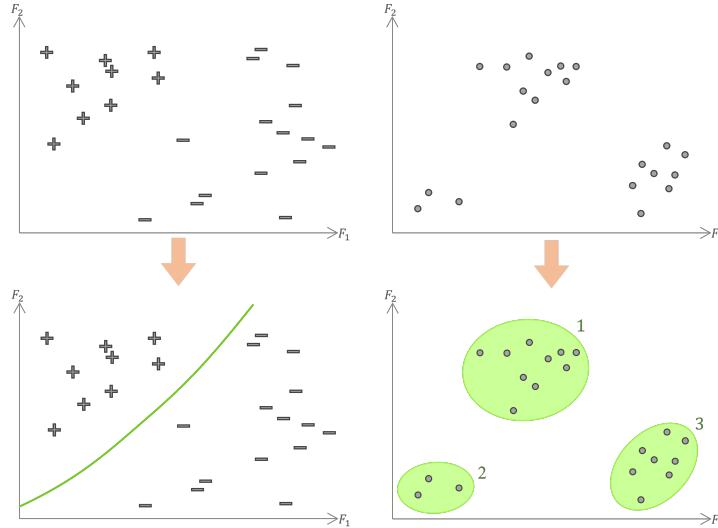


Figure 1.5: Comparing supervised (left) and unsupervised (right) learning

**Supervised learning** In the case of labeled data, we can apply **supervised learning**. The goal is to find a “rule” in terms of descriptive features explaining the target feature as well as possible. Examples include:

- Hospital environments where:
  - The target variable can be **recover** (yes or no), and
  - The descriptive variables can be **age**, **gender**, **smoking**, ...
- University environments where:
  - The target variable can be **drops out** (yes or no), and
  - The descriptive variables can be **mentor**, **prior education**, ...
- Production environments where:
  - The target variable can be **order is delivered in time** (yes or no), and
  - The descriptive variables can be **product**, **agent**, ...

**Unsupervised learning** In contrast to labeled data, we can also have instances without target labels, where we can only apply techniques of **unsupervised learning**. The goal is to find clusters or patterns.

- |                |   |
|----------------|---|
| <b>Cluster</b> | • <b>Clusters</b> are homogeneous sets of instances. Examples include finding similar groups of patients, students, customers, orders, cars, companies, and so on.  |
| <b>Pattern</b> | • <b>Patterns</b> on the other hand reveal hidden structures in the data, so basically the unknown unknowns. Rules of some form can be found in many environments and can for example look like this: <ul style="list-style-type: none"> <li>– Customers who buy bread and butter typically pay by phone.</li> <li>– Patients who drink and smoke typically pay the hospital bill earlier than others.</li> </ul> |

- Products produced by team A on Monday tend to be returned more frequently by customers.

Interesting to regard is process discovery as a form of unsupervised learning in the way that a process model is just a very sophisticated rule. Important to mention, that this task can get very complex very quickly.

## Terminology

Important to see for all of data science: many different names are used to refer to the key disciplines contributing to data science.

- This includes statistics, data analytics, data mining, machine learning, artificial intelligence, predictive analytics, process mining, generative AI, etc.
- Since frequently the same name is used for different concepts (names describe heavily overlapping areas), they really need to be put in context and interpreted accordingly.

The point can be highlighted when looking at scoping machine learning. Here are examples of confusions:

- Sometimes "machine learning" is used as a synonym for "deep neural networks" and sometimes they cover the entire spectrum of learning techniques.
- Neural networks can be used as classifiers. But this doesn't imply that numerous classification techniques developed in data mining are part of machine learning in the narrow sense.

How confusing specifically the arrangement of terms around machine learning is and how fluent and unclear the actual terms are, is depicted in 1.6.

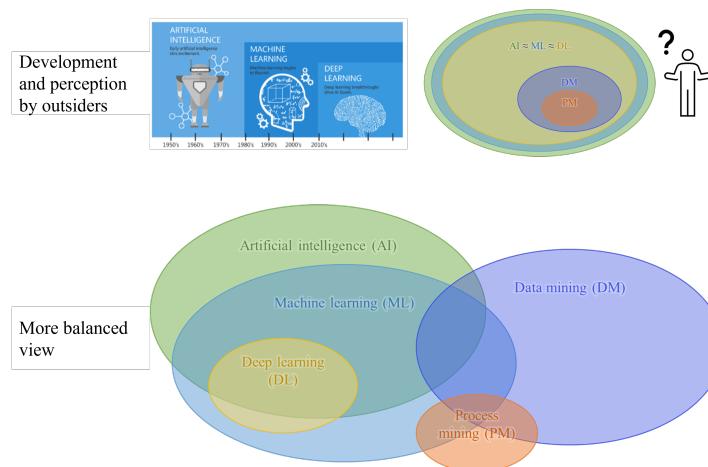


Figure 1.6: Terms around machine learning

## 1.4 Data science process

There are many different lifecycle models to describe phases in a data science project. This section will give a quick overview of some important ones.

CRISP-DM We'll start with **CRISP-DM** which stands for "Cross-industry standard process for data mining". It was developed in the late 1990s by different involved companies (SPSS, Teradata, Daimler AG, NCR Corporation, Ohra). The process consists of multiple steps playing together as visualized in 1.7

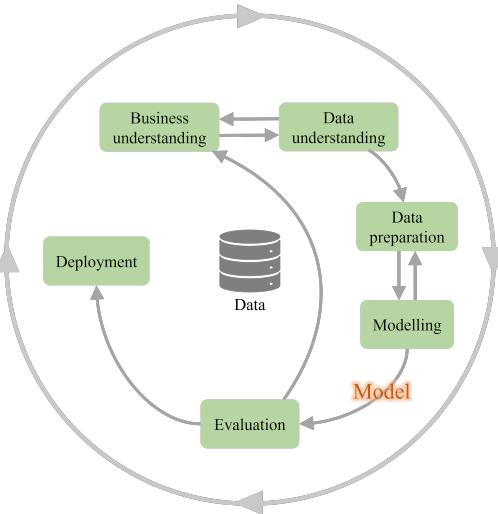


Figure 1.7: CRISP-DM process

### **Business understanding**

Determine business objective  
Situation assessment

Background, business objective, business success criteria  
Inventory of resources, requirements, assumptions, constraints, risks, contingencies, terminology, costs, benefits  
Data mining goals, data mining success criteria  
Project plan, initial assessment of tools and techniques

### **Data understanding**

Collect initial data  
Describe and explore data  
Verify data quality

Initial data collection report  
Data description, exploration report  
Data quality report

### **Data preparation**

Starting point: data set  
Select data  
Clean data  
Construct data  
Integrate and format data

Data set, data set description  
Rationale for inclusion and exclusion  
Data cleaning report  
Derived attributes, generated records  
Merged/reformatted data

### **Modeling<sup>1</sup>**

Select modeling technique  
Generate test design  
Build model  
Assess model

Modeling technique, modeling assumptions  
Test design  
Parameter settings, models, model description  
Model assessment, revised parameter settings

### **Evaluation**

Evaluate results  
Review process

Assessment of data mining results w.r.t. business success criteria, approved models  
Review of process

---

<sup>1</sup>The term "modeling" can be misleading, meant is the selection and assumptions (human) or automated learning by a tool or algorithm

Determine next steps

List of possible actions settings

## Deployment

- Plan deployment
- Plan monitoring and maintenance
- Produce final report
- Review project

- Deployment plan
- Monitoring and maintenance plan
- Final report and final presentation
- Experience documentation

Next, we have the **KDD** (Knowledge Discovery in Databases) process as shown in 1.8. Another process model also developed by SAS institute is called **SEMMA** consisting of the phases Sample, Explore, Modify, Model, and Assess.

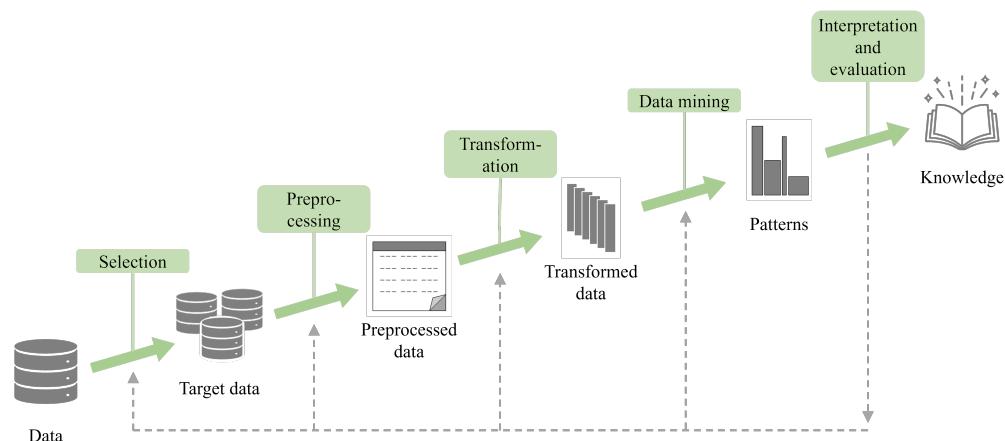


Figure 1.8: KDD process

The next process model is specifically developed for **L\* lifecycle model** with multiple stages as shown in 1.9.

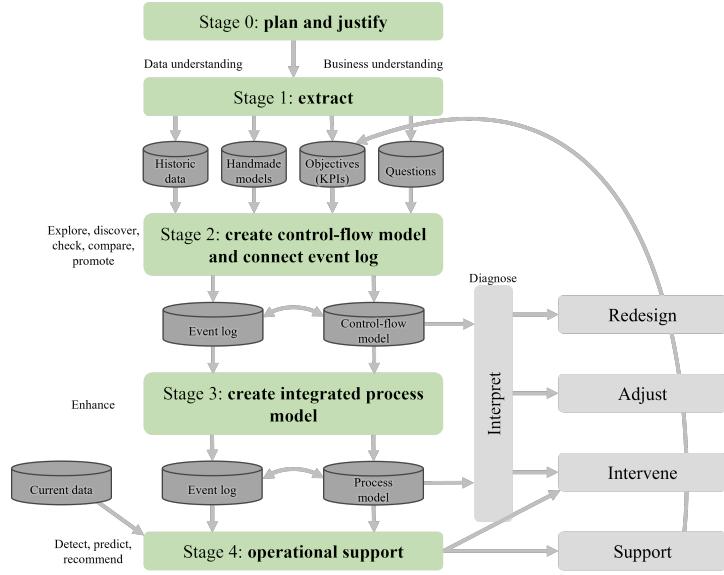
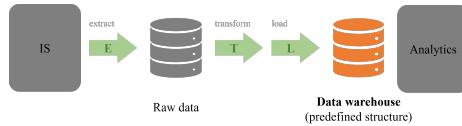


Figure 1.9:  $L^*$  lifecycle model

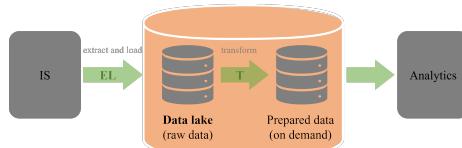
Furthermore, we have two methodologies the process model can be related to. Important to implement and solidify are improvements in both.

- |       |  |
|-------|--|
| PDCA  | <ul style="list-style-type: none"> <li>• <b>PDCA</b> stands for Plan-Do-Check-Act and is a never-ending cycle with exactly these steps.</li> </ul>   |
| DMAIC | <ul style="list-style-type: none"> <li>• The other one <b>DMAIC</b> stands for Define-Measure-Analyze-Improve-Control, with the following subtasks:           <ul style="list-style-type: none"> <li>– Define: launch team, establish charter, plan project, gather VOC/VOB, plan for change</li> <li>– Measure: document process, collect baseline data, narrow project focus</li> <li>– Analyze: analyze data, identify root causes, identify and remove waste</li> <li>– Improve: generate, evaluate, and optimize solutions, pilot, plan and implement</li> <li>– Control: control the process, validate project benefits</li> </ul> </li> </ul> |

Finally, we have two processes with the same components, but different ordering of the ETL steps as can be seen in 1.10. The short terms for the processes are **ETL** (extract, transform, load) and **ELT** (extract, load, transform).



(a) ETL with data warehouse



(b) ELT with data lake

Figure 1.10: Processes with extraction, transform, and load steps

As a final note on which steps are usually the most time-expensive: there is a so-called "80/20 rule" stating:

- 80% of a data scientist's time is spent on finding, cleaning, preprocessing, and organizing data. This leaves only 20% to actually perform an analysis.
- On the other hand, we have 20% effort determining 80% of the final result.

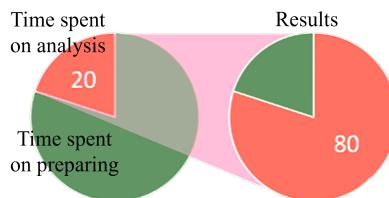


Figure 1.11: 80-20 rule

## 1.5 Challenges

To finalize the overview and basics of data science, let's look at the typical challenges.

First, we have the challenge of **finding data**. There may be hundreds or thousands of tables, for example in the case of SAP the numbers can easily go up to **800'000**. But, different entities differ in their relevance, meaning some are less relevant than others.

The next challenge is the **transformation of data**, meaning reorganization of data, filtering, extraction of relevant features, and so on. Not only for transformations, but also in general other challenges are **dealing with big data and streaming data**. The challenge of big data evolved over the last few decades, meaning typical stochastic methods try to solve the problem of saying something about entities given only a small amount of samples, whereas now we have a very high load of data, and need to solve the problem of dealing with these large amounts in a correct way. Also for streaming data, new approaches need to be thought of. Additionally, we also need to **deal with a concept shift**.

Another huge challenge is ensuring **data quality**. This goes especially, since our provided

data may be incomplete, invalid, inconsistent, imprecise, and/or outdated. Consider for example timestamps. They might be

- Incomplete (event is missing),
- Invalid (e.g. 14-14-2018),
- Inconsistent (14-07-2018 in contrast to 7-14-2018), or
- Imprecise (only regard part of available data: 2018-09-21<sup>T</sup>'13:00:10).

A very typical problem is **overfitting and underfitting** as it can be seen 1.12.

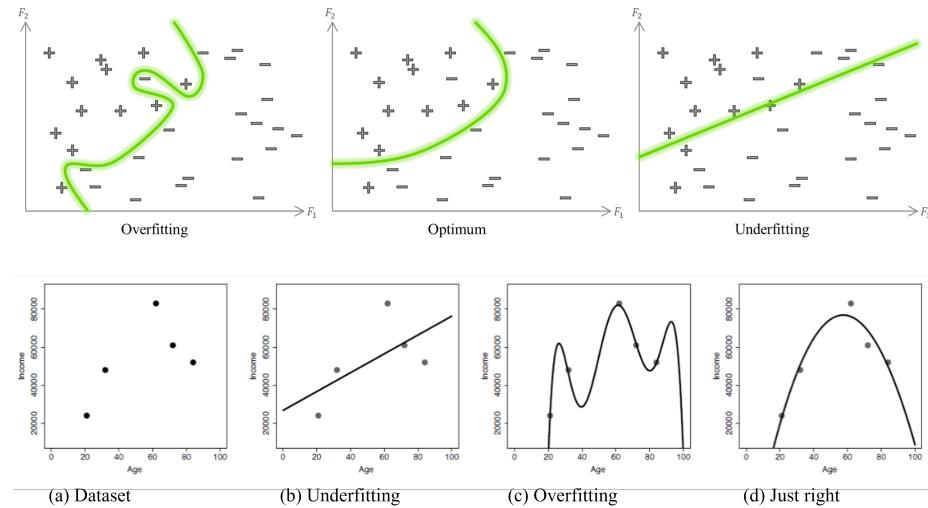


Figure 1.12: Over- and underfitting visualized

The next challenge is the distinction of **correlation and causation**, explicitly that correlation does not imply causation. Consider this example:

- Sunburn and ice cream have a strong correlation. When only these two features are considered, one might derive that either ice cream causes sunburn, or the other way around.
- We know of course, that this is not correct and instead an additional factor causes both phenomena: if the sun is shining, it's warm and people eat ice cream, and also sun directly causes sunburn.

Besides the accuracy of our results, we also need to look into whether our results are valuable. Concretely, **results** should be **made actionable**. This means, that analysis results should be relevant, specific, timely, novel, and clear. Our goal is to go from "data" to "insight" and finally "action". Consider these examples:

- Warning about a traffic jam should come before entering said traffic jam.
- That it's currently raining is not too helpful information. Preferably is a notice ahead of time.

Responsible data science      The last, but very important challenge is **responsible data science (RDS)** . This includes ensuring of:

- **Fairness**, meaning data science should exclude prejudice (How to avoid unfair conclusions even if they are true?)

- **Accuracy**, so data science without guesswork (How to answer questions with a guaranteed level of accuracy?)
- **Confidentiality** (How to answer questions without revealing secrets?)
- **Transparency** (How to clarify answers such that they become indisputable?)

## 2 Data visualization and exploration

### 2.1 Data extraction

Generally, we have a bunch of different data sources, with a multitude of different standards, features, and also information that can be extracted. The general problem is depicted in 2.1.

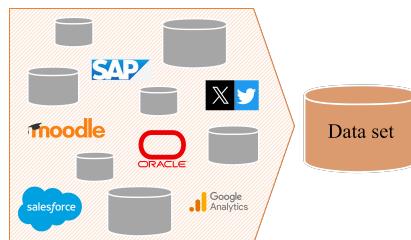


Figure 2.1: Data extraction from different sources

The different datatypes were already analyzed in the previous chapter, but still 2.2 shows a quick recap. Important to mention, that any unstructured data is considered a bit stream.

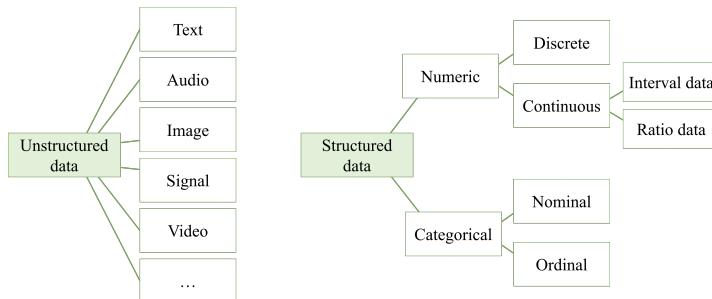


Figure 2.2: Recap: overview types of data

Important when wanting to obtain any object is of course the feature extraction. The data described by features are usually captured in a tabular form, with rows as the instances and columns as the features. There exist some special features:

- **Time** usually always plays a role in data observation, which is why it is usually one of the recorded features.
- Then there are also the **target features**, in contrast to the descriptive features. The concept was introduced in the last section as part of supervised learning.

## Importance of visualization

We now looked at how we can represent our data in a very machine-friendly represented way. The following subsection shows, why visualization of data has any importance, even though tabular data already captures the features nicely.

It is important as a human to explore your data before applying mathematical operations to see, which techniques make sense to apply to the provided data. As an extreme example, we will take a look at **Anscombe's quartet** created by Francis Anscombe in 1973.

- You can see the raw data of all four datasets in the table in 2.3. Since the format isn't human-friendly to read, you might not see any significant differences in the data.
- Now consider applying the evaluation depicted below the data table. As you can see: all of the properties that are evaluated are exactly the same for all datasets.
- BUT, if you visualize the datasets, e.g. as simple scatter plots, you see, how drastically they vary. These show the importance of first exploring your data, to then have a better evaluation foundation for the applied techniques.

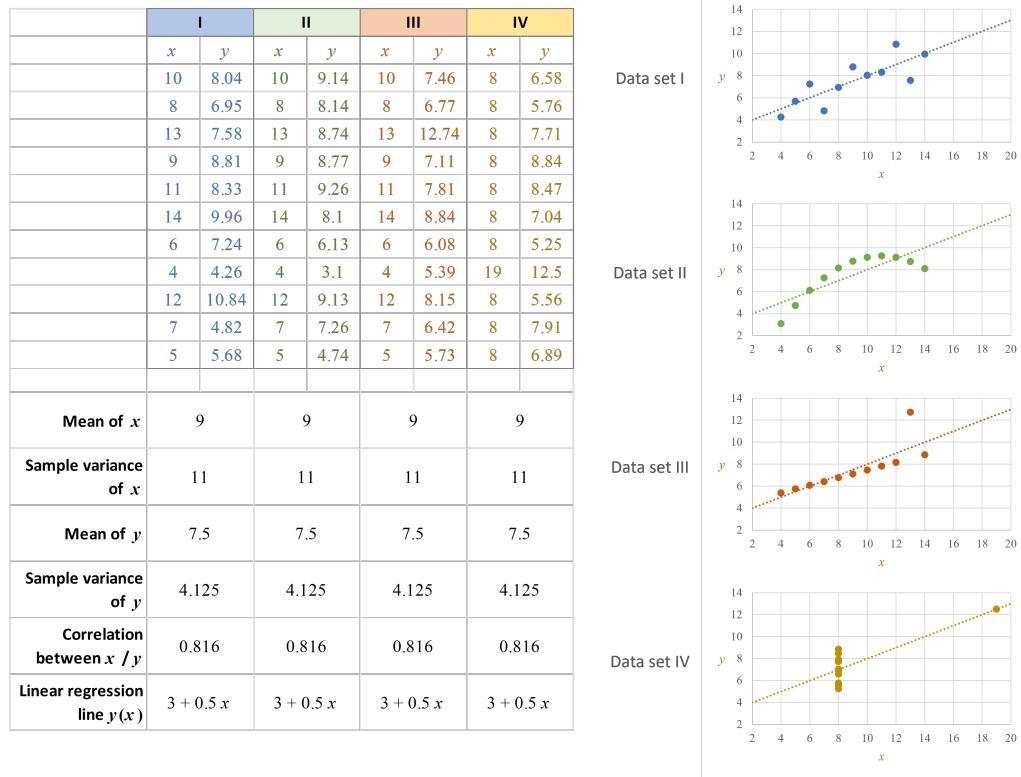


Figure 2.3: Anscombes quartet

The next example highlighting the importance of visualization and especially of a fitting and well-thought-out visualization is the diagram as shown in 2.4. The chart shows the following aspects, which are quite a lot, while still keeping a good overview:

- The number of men in Napoleon's 1812 Russian campaign army,

- Their movements (direction),
- The temperature encountered on the return path,
- All given a specific geographic point.

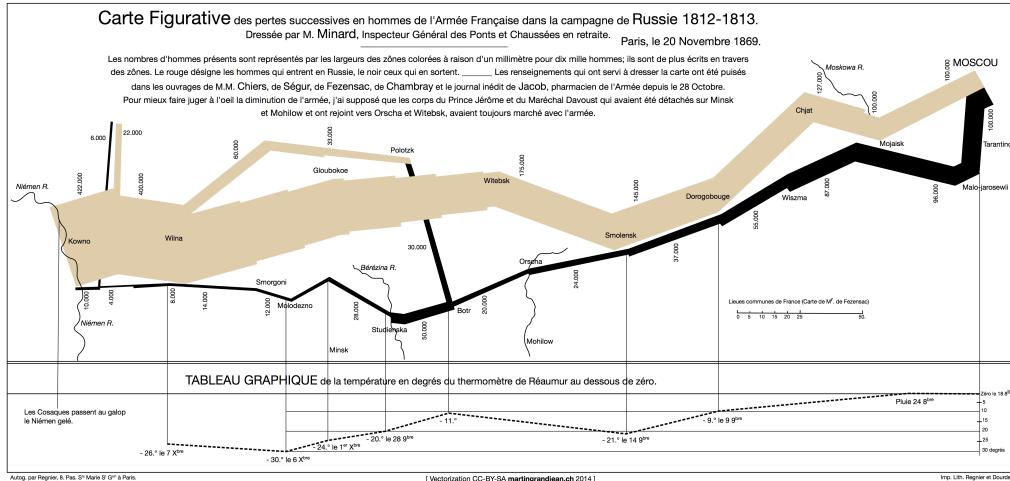


Figure 2.4: Multi-feature visualization (Napoleon's army)

As a final example: when we have given different event data, it often helps to plot this in some way as can be seen in 2.5. With the visualization, some sort of trend, similarities, certain batching areas, etc. can be directly seen.

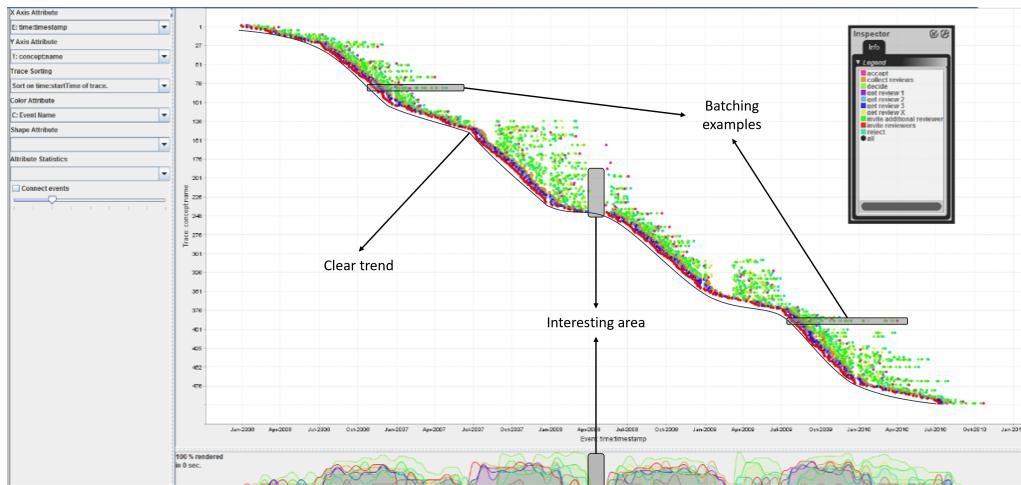


Figure 2.5: Visualization of event data

## 2.2 Characterizing individual features

As a first step of actual data exploration, we are now going to look at which information we can get from a single data feature. In terms of tabular data: we're going to focus on a single column.

What kind of data we can derive from the feature, depends of course on the data type.

Properties from  
investigating  
individual  
continuous feature

Generally deriving features from other ones is of course done best when dealing with structured data. Here, we have two types from which we can derive different properties.

From **continuous features**, we can derive

- count** : Number of instances having this feature
- % miss** : Percentage of missing information (how many instances don't have this feature)
- card** : Number of unique values (cardinality)
- min** : Minimal value over all instances
- 1<sup>st</sup> qrt** : 25<sup>th</sup> percentile (largest value of the quarter of instances having the lowest values)
- mean** : Average value over all instances
- median** : Middle value of all instances
- 3<sup>rd</sup> qrt** : 75<sup>th</sup> percentile (smallest value of the quarter of instances having the highest values)
- max** : Maximal value over all instances
- std. dev** : Standard deviation over all instances

Properties from  
investigating  
individual  
categorical feature

From **categorical features**, we can derive <sup>2</sup>

- count** : Number of instances having this feature
- % miss** : Percentage of missing information (how many instances don't have this feature)
- card** : Number of unique values (cardinality)
- mode** : Most common value
- mode frequ** : Frequency of the mode
- mode %** : Percentage of the mode
- 2<sup>nd</sup> mode** : Second most common value
- 2<sup>nd</sup> mode frequ** : Frequency for the second mode
- 2<sup>nd</sup> mode %** : Percentage of the second mode

To get a better idea of how to get these properties for all of the features, we will look at an example. Consider a table containing information about insurance claims fraud. The dataset contains 500 instances (claims) and a bunch of different features such as type, claim amount, etc. Now first, determine the data type of each feature, and then create one table for the numerical and one for the categorical features and fill it with the according information. The original data and the resulting feature-describing tables can be found in 2.6.

---

<sup>2</sup>obvious: **min**, **max**, **mean**, etc. can't be computed

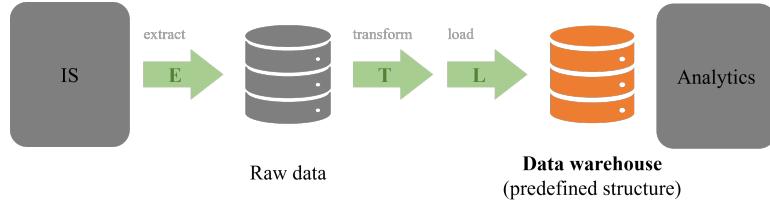
ID	TYPE	INC.	MARITAL STATUS	NUM CLMNTS.	INJURY TYPE	HOSPITAL STAY	CLAIM AMNT.	TOTAL CLAIMED	NUM CLAIMS	NUM SOFT TISS.	% SOFT TISS.	CLAIM AMT RCVD.	FRAUD FLAG
1	CI	0		2	Soft Tissue	No	1,625	3250	2	2	1.0	0	
2	CI	0		2	Back	Yes	15,028	60,112	1	0	0	15,028	0
3	CI	54,613	Married	1	Broken Limb	No	-99,999	0	0	0	0	572	0
4	CI	0		4	Broken Limb	Yes	5,097	11,661	1	1	1.0	7,864	0
5	CI	0		4	Soft Tissue	No	8869	0	0	0	0	0	1
6	CI	0		1	Broken Limb	Yes	17,480	0	0	0	0	17,480	0
7	CI	52,567	Single	3	Broken Limb	No	3,017	18,102	2	1	0.5	0	1
8	CI	0		2	Back	Yes	7463	0	0	0	0	7,463	0
9	CI	0		1	Soft Tissue	No	2,067	0	0	0	0	2,067	0
10	CI	42,300	Married	4	Back	No	2,260	0	0	0	0	2,260	0
300	CI	0		2	Broken Limb	No	2,244	0	0	0	0	2,244	0
301	CI	0		1	Broken Limb	No	1,627	92,283	3	0	0	1,627	0
302	CI	0		3	Serious	Yes	270,200	0	0	0	0	270,200	0
303	CI	0		1	Soft Tissue	No	7,688	92,806	3	0	0	7,688	0
304	CI	46,365	Married	1	Back	No	3,217	0	0	0	0	1,653	0
458	CI	48,176	Married	3	Soft Tissue	Yes	4,653	8,203	1	0	0	4,653	0
459	CI	0		1	Soft Tissue	Yes	881	51,245	3	0	0	0	1
460	CI	0		3	Back	No	8,688	729,792	56	5	0.08	8,688	0
461	CI	47,371	Divorced	1	Broken Limb	Yes	3,194	11,668	1	0	0	3,194	0
462	CI	0		1	Soft Tissue	No	6,821	0	0	0	0	0	1
491	CI	40,204	Single	1	Back	No	75,748	11,116	1	0	0	0	1
492	CI	0		1	Broken Limb	No	6,172	6,041	1	0	0	6,172	0
493	CI	0		1	Soft Tissue	Yes	2,569	20,055	1	0	0	2,569	0
494	CI	31,951	Married	1	Broken Limb	No	5,227	22,095	1	0	0	5,227	0
495	CI	0		2	Back	No	3,813	9,982	3	0	0	0	1
496	CI	0		1	Soft Tissue	No	2,118	0	0	0	0	0	1
497	CI	29,280	Married	4	Broken Limb	Yes	3,199	0	0	0	0	0	1
498	CI	0		1	Broken Limb	Yes	32,469	0	0	0	0	16,763	0
499	CI	46,683	Married	1	Broken Limb	No	179,448	0	0	0	0	179,448	0
500	CI	0		1	Broken Limb	No	8,259	0	0	0	0	0	1
Continuous/ numerical features													
Feature	Count	Miss.	Card.	Min	1 <sup>st</sup> Quart.	Mean	Median	3 <sup>rd</sup> Quart.	Max	Std. Dev.			
INCOME	500	0.0	171	0.0	13,740.0	0.0	33,918.5	71,284.0	20,081.5				
NUM CLAIMANTS	500	0.0	4	1.0	1.0	1.9	2	3.0	4.0	1.0			
CLAIM AMOUNT	500	0.0	493	-99,999	3,322.3	16,373.2	5,663.0	12,245.5	270,200.0	29,426.3			
TOTAL CLAIMED	500	0.0	235	0.0	0.0	9,597.2	0.0	11,282.8	729,792.0	35,655.7			
NUM CLAIMS	500	0.0	7	0.0	0.0	0.8	0.0	1.0	56.0	2.7			
NUM SOFT TISSUE	500	2.0	6	0.0	0.0	0.2	0.0	0.0	5.0	0.6			
% SOFT TISSUE	500	0.0	9	0.0	0.0	0.2	0.0	0.0	2.0	0.4			
AMOUNT RECEIVED	500	0.0	329	0.0	0.0	13,051.9	3,253.5	8,191.8	295,303.0	30,547.2			
FRAUD FLAG	500	0.0	2	0.0	0.0	0.3	0.0	1.0	1.0	0.5			
Categorical features													
Feature	Count	Miss.	Card.	Mode	Mode Freq.	Mode %	2 <sup>nd</sup> Mode	2 <sup>nd</sup> Mode Freq.	2 <sup>nd</sup> Mode %				
INSURANCE TYPE	500	0.0	1	CI	500	100.0	—	—	—				
MARITAL STATUS	500	61.2	4	Married	99	51.0	Single	48	24.7				
INJURY TYPE	500	0.0	4	Broken Limb	177	35.4	Soft Tissue	172	34.4				
HOSPITAL STAY	500	0.0	2	No	354	70.8	Yes	146	29.2				

Figure 2.6: Example for single feature investigation (insurance claim fraud)

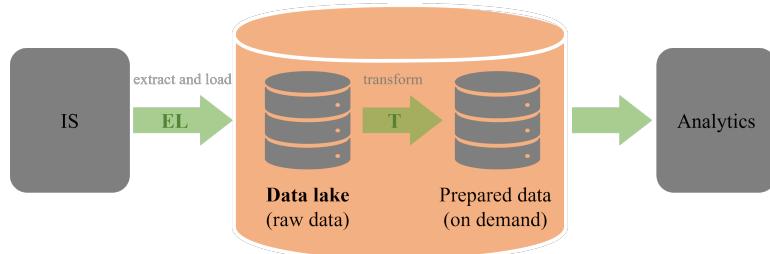
To investigate the raw data further, let's visualize it. More specifically, we're going to visualize the distributions of the different features. Figure 2.7 shows exemplary plots.

- For finite amounts of possible feature classes, simply visualize the distribution as a bar diagram with the different classes as entries on the x-axis. The y-axis can either be the frequency or a percentage.
- For continuous features with continuous variables/ininitely many possible feature values, group items (**binning**) and then visualize the resulting histogram.

Binning



(a) Distribution of categorical features



(b) Distribution of numerical (continuous) features

Figure 2.7: Distribution visualization

The binning comes with some challenges. When we select the amount of bins with evenly distributed width of each individual bin, we need to be aware not to **over- or underfit**. Examples can be found in 2.8. As one can see:

- In the case of underfitting, the true function is not at all matched.
- In the case of overfitting, there exist very steep valesys, the provided data points are more learned by heart rather than abstracting to a function.

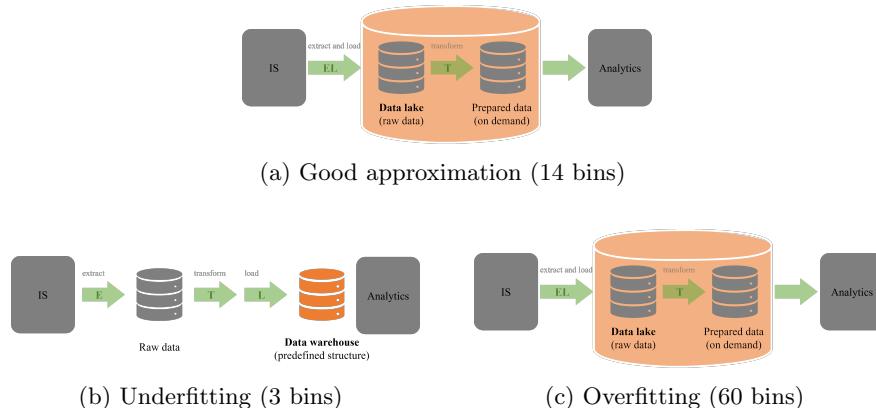


Figure 2.8: Binning for continuous variables

The histograms furthermore can show different types, as depicted in 2.9. Here are some further notes on the types:

- **Uniform** means all items have the same likelihood (within a range).
- **Unimodal** means we have one peak (can be tilted to one side), whereas **multimodal** means there are multiple distinct ones.

- **Exponential** means we have an exponential decrease in the likelihood over all instances.

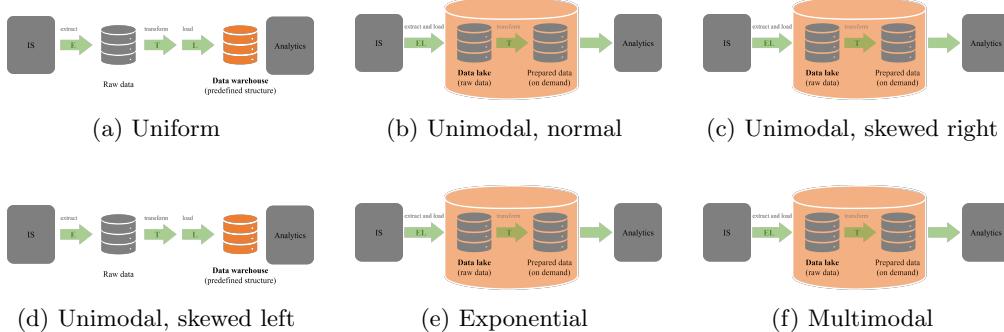


Figure 2.9: Histogram types

One of the types mentioned, we're now gonna investigate a bit further. The **normal distribution** is described by two important variables, whose effects on the distribution are shown in 2.10:

- The **mean  $\mu$** , so the expected value also characterizing the peak, and
- The **standard deviation  $\sigma$**  characterizing how narrow the peak, or the distribution around the peak, is.

Normal distribution

Mean

Standard deviation

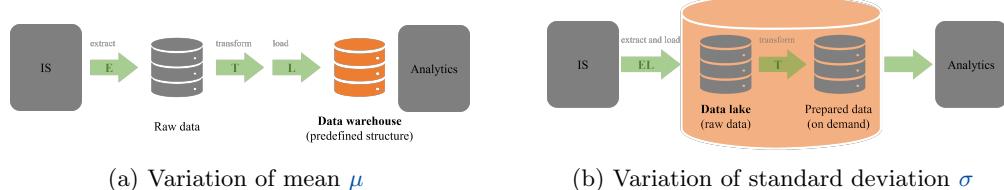


Figure 2.10: Normal distribution

The normal probability distribution over  $x$  is defined as:

$$x \sim \mathcal{N}(\mu, \sigma)$$

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp \left[ -\frac{1}{2} \left( \frac{x-\mu}{\sigma} \right)^2 \right]$$

Interesting are now precise areas we instantly know something about. The **68-95-99.7-rule** tells us, as depicted in 2.11.

68-95-99.7-rule

- **68%** of all observations will be within  $1\sigma$ -distance of the mean,
- **95%** of all observations will be within  $2\sigma$ -distance of the mean, and
- **99.7%** of all observations will be within  $3\sigma$ -distance of the mean

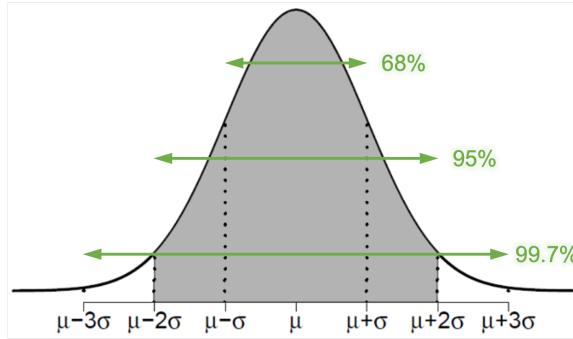


Figure 2.11: 68-95-99.7-rule

From this rule, we can now derive probabilities for different events. Consider the following examples, also visualized in 2.12:

- Example 1 is interested in the amount of defects for some produced item. The **tolerance** can be defined as within the  $2\sigma$ -range, so with:

- |     |   |
|-----|---|
| LSL | – <b>LSL</b> (lower specification limit) at $\mu - 2\sigma$ , meaning only 2.5% of the instances have a larger deviation into the negative direction from the mean than this limit, and |
| USL | – <b>USL</b> (upper specification limit) at $\mu + 2\sigma$ , meaning only 2.5% of the instances have a larger deviation into the positive direction from the mean than this limit.     |

Combined,  $100\% - 2.5\% - 2.5\% = 95\%$  have a deviation from the mean lying within the defined range.

- Example 2 is interested in how many deliveries are too late. Therefore, it **only** defined an **upper bound** with the USL at  $\mu + 2\sigma$ . This means,  $100\% - 2.5\% = 97.5\%$  of the deliveries are not too late.

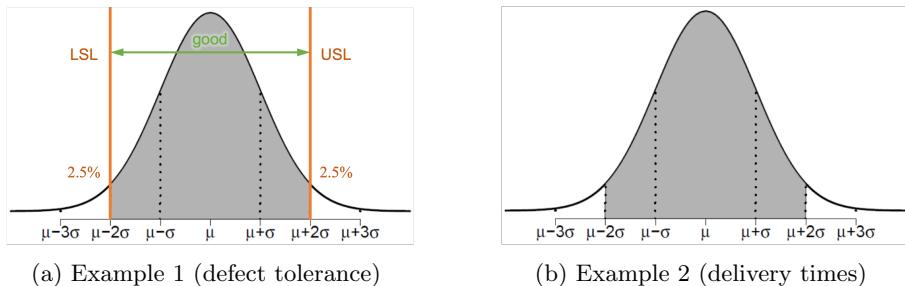


Figure 2.12: Examples for 68-95-99.7-rule

Furthermore, the rule also introduces the term of **six sigma** or also lean six sigma. It basically means: Processes operating with "six sigma quality" are assumed to have  $< 3.4$  defects per million instances, so  $\text{Pr}(\text{error}) = 0.0000034$ . It characterizes a process improvement approach. This likelihood is a combination of  $\pm 6\sigma$  tolerance and a "drift" of  $\pm 1.5\sigma$ .

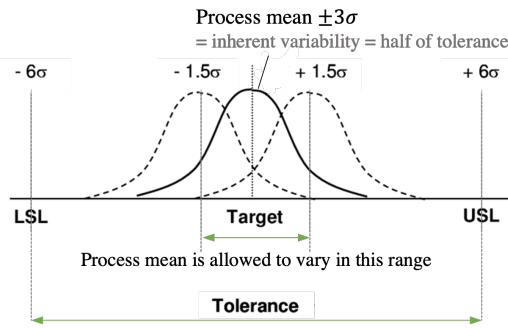


Figure 2.13: Lean six sigma

## 2.3 Data quality

In the introduction, we already looked at some key challenges regarding data quality. In this subsection, we will investigate and search solutions for the some of the following typical data quality problems in detail:

- **Incompleteness** - missing instances or attributes
- **Invalidity** - impossible values
- **Inconsistency** - conflicting values
- **Imprecision** - approximates or rounded values
- **Outdated** - values based on old observations

For that, we will take a look at missing, invalid, unlikely, and outlier values.

### Missing values

Imagine different missing features of some instances. Since some data is missing, we need to deal with this in some way. Here are the possible options: Missing features

1. Remove feature completely (for all instances)
2. Only consider instances that have a value (this is done for per-feature-evaluation)
3. Remove all instances that have one of the features missing
4. Repair missing features (imputation)

The problem setting and the possible solutions are visualized in 2.14.

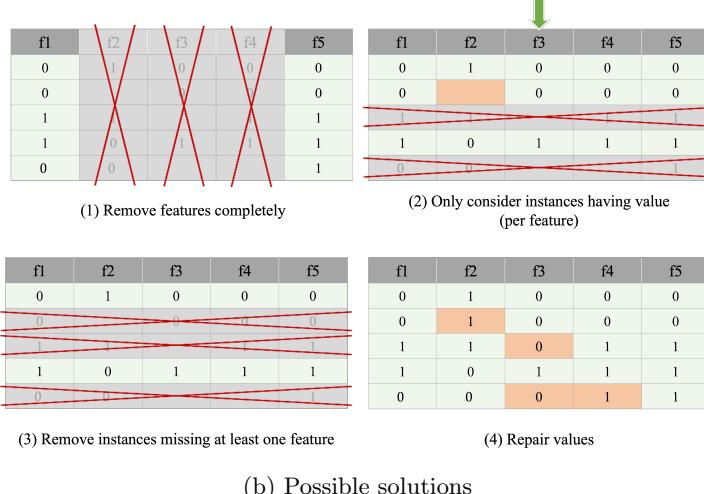
### Impossible values

The next typical challenge are impossible values that by some mistake were entered as data. Examples are: Impossible values

- Wrong date format: instead of **2018-10-18**, we would have **18-10-2018**
- Completely impossible date or time: **2018-13-51, 23:61**
- There can be spelling errors, for example for colors: **Bllue**

f1	f2	f3	f4	f5
0	1	0	0	0
0		0	0	0
1	1		1	1
1	0	1	1	1
0	0			1

(a) Problem setting



(b) Possible solutions

Figure 2.14: Missing values

- The data type might not make sense with the feature, like number of members as a float: **6.5 member**

The handling of this problem is solved just as for missing features.

### Unlikely values

In contrast to impossible values, unlikely values are theoretically possible, but just not common to appear. Examples are:

- Age: **123** is rather unlikely, but possible
- Price: **120.000\$** in a store where the other prices lie in the range of **5\$** to **150\$**
- Dates: even on dates, where one would usually expect a uniform distribution over months and days, days **1** to **12** are more frequent than days **13** to **31**<sup>3</sup>

Unlikely values, domain knowledge Whether a value is unlikely or not is identified based on **domain knowledge**. They can then be further investigated to see, whether the unlikely value is actually valid.

### Outlier values

Outlier values In contrast to unlikely values, **outlier values** are identified based on the distribution.

<sup>3</sup>NOT the case anymore if date format **DD-MM-YYYY** and **MM-DD-YYYY** are mixed

An especially popular technique to visualize distributions and outliers are **box plots**. They were first introduced by John Tukey in the book "Exploratory data analysis" in 1977. Figure 2.15 shows the properties visualized by a box plot and also how to construct one given a data set.

Box plot

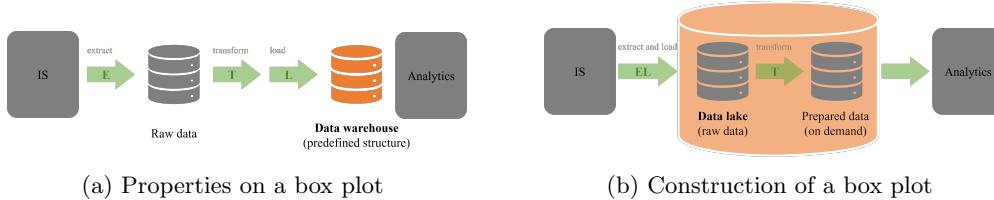


Figure 2.15: Box plot

Let's first take a look at the properties one can see in the box diagram.

- The **median** value<sup>4</sup> is depicted by the "Bar" in the center.
- The **IQR**, so the interquartile range<sup>5</sup>, covering 50% of the "middle instances" is IQR depicted by the "Box".
- The upper whisker indicates the **maximal** value below the  $3^{\text{rd}} \text{ quartile} + 1.5 \cdot IQR$ , whereas
- The lower whisker indicates the **minimal** value above the  $1^{\text{st}} \text{ quartile} - 1.5 \cdot IQR$ .
- Finally, the **outliers** are drawn separately.

The description already contained a bit of the construction details, which will now be explained in more detail with an example. Consider the (already ordered) data set  $\{1, 2, 5, 7, 8, 8, 9, 9, 9, 10, 10, 10, 11, 11, 12, 12, 14, 14, 19, 23\}$ . Then we construct the box diagram like this:

- The median value is 9 (at position 9).
- The first quartile has the value 8 (at position 5), the third one has the value 11 (at position 13) resulting in an  $IQR = 11 - 8 = 3$ .
- This means we have an upper fence  $11 + 1.5 \cdot 3 = 15.5$ , and the upper whisker as the maximum value below this fence at 14 (position 15).
- The lower fence has the value  $8 - 1.5 \cdot 3 = 3.5$ , the lower whisker therefore the minimum value above this fence value at 5 (position 3).
- Finally, the outliers are 1, 2, 19, 23 (position 1, 2, 16, 17).

Those are all the necessary components to construct the box diagram.

Now, one final detail about box diagrams and also the topic of this paragraph is the handling of the outliers. They can first be removed (meaning remove values above and below the upper and lower fences), and their existence can be indicated by claming the removed values to these thresholds. The process is shown in 2.16.

---

<sup>4</sup>Median = The "middle" value, so the number halfway between lowest and highest number.

<sup>5</sup>First to third quartile, where the first quartile is the number halfway between lowest and middle number, and third quartile is the number halfway between middle and highest number.

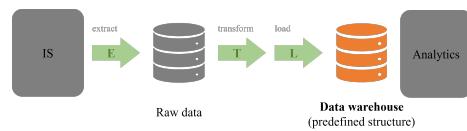


Figure 2.16: Handling outliers in box plots

## 2.4 Showing relations among features

## 2.5 Preparing for analysis

## 2.6 Good and poor visualizations