Introduction to Data Science

WS 23/24, RWTH Aachen

October 31, 2023

Contents

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

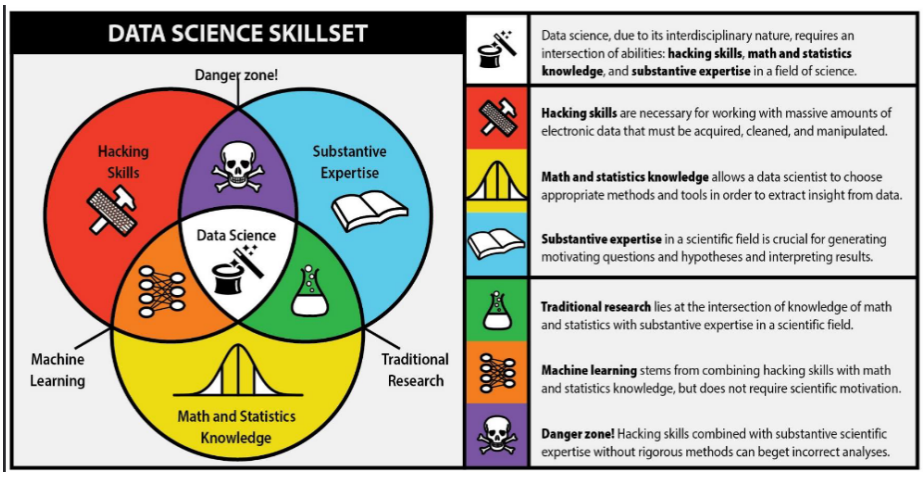


Figure 0.1: Skillset of a data scientist

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-developement 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

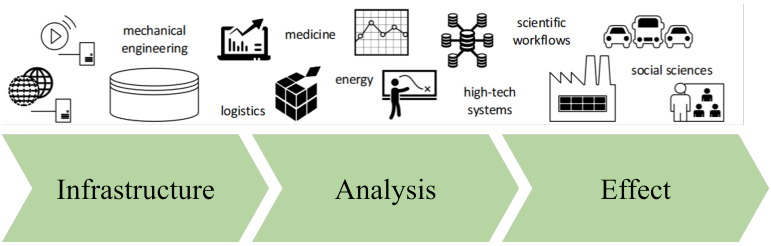
1 Basics of data science

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.

Infrastructure

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:



Data science pipeline

Figure 1.1: Pipeline of data science

- 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:

Analysis

- 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:

Effects

- 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.

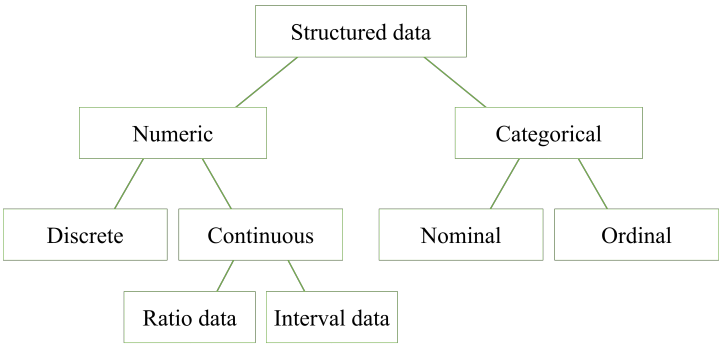


Figure 1.2: Overview structured data types

Categorical data
- **Categorical** 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, **nominal** 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
- **Ordinal** 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 **numerical** 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 $+, -, >, =, \ldots$).

Discrete data
- One subtype of numerical data is **discrete** 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 **continuous** 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 **interval** 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 $\cdot, /$). (Example: data=11-11-2018, temp=18.5°C)

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

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.

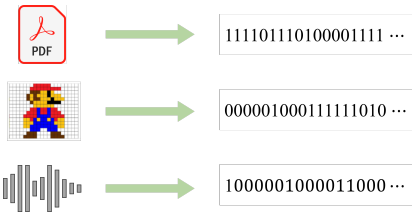


Figure 1.3: Input for unstructured data

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. Tabular data

feature

Order id	Product	Price	Date
32424	718 Cayman	66.000	21-10-2018
34535	911 Carrera	102.000	22-10-2018
43555	911 Turbo	154.000	24-10-2018
...	...	...	...
Ordinal	Nominal	Ratio data	Interval data

From	To	Message	Image
Sue	Peter	"How are you?"	
Peter	Sue	"Very good!"	
Peter	Mary	"Let's go out."	
...	...	...	...
Nominal			Unstructured

instance

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

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. Supervised learning

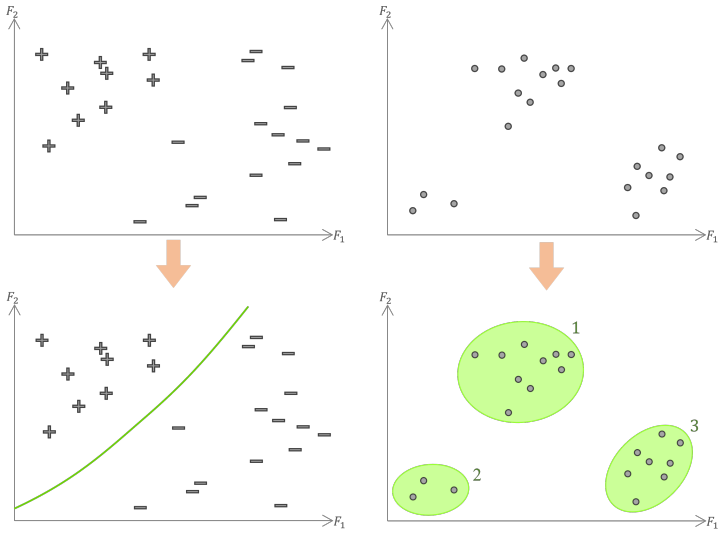


Figure 1.5: Comparing supervised (left) and unsupervised (right) learning

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.

Cluster

- **Clusters** are homogeneous sets of instances. Examples include finding similar groups of patients, students, customers, orders, cars, companies, and so on.

Pattern

- **Patterns** 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:
  - Customers who buy bread and butter typically pay by phone.
  - Patients who drink and smoke typically pay the hospital bill earlier than others.
  - 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.

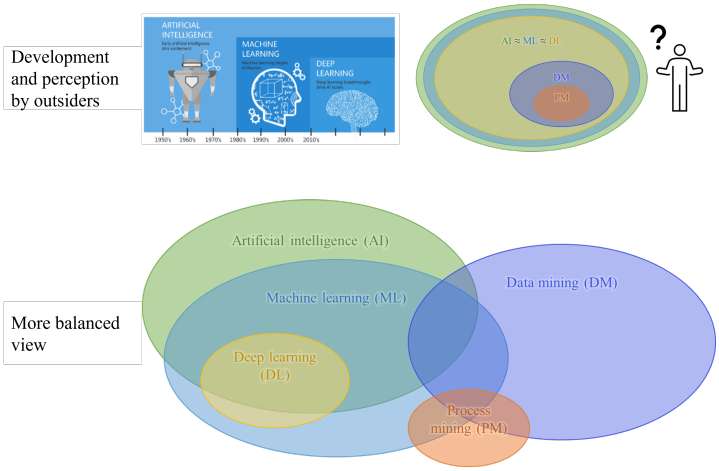


Figure 1.6: Terms around machine learning

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.

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

CRISP-DM

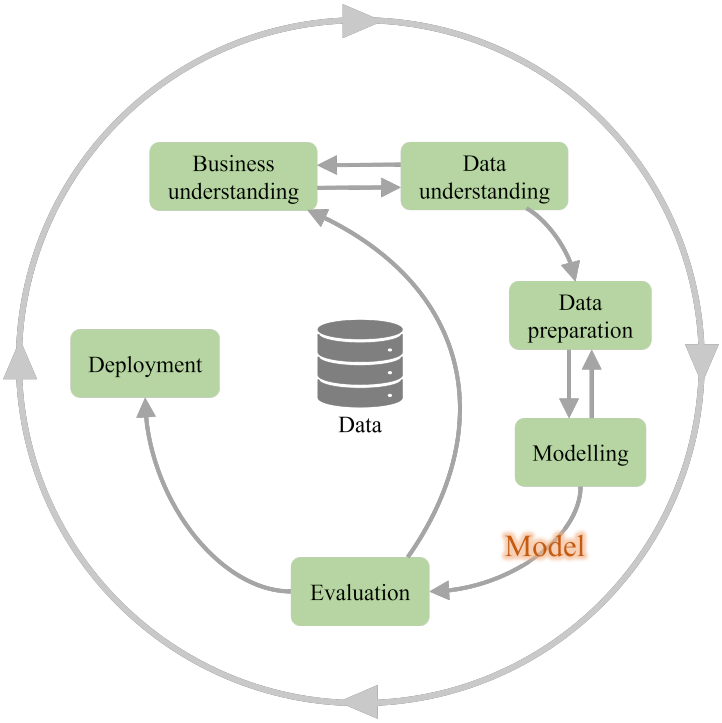


Figure 1.7: CRISP-DM process

Business understanding	
Determine business objective	Background, business objective, business success criteria
Situation assessment	Inventory of resources, requirements, assumptions, constraints, risks, contingencies, terminology, costs, benefits
Determine data mining goal	Data mining goals, data mining success criteria
Produce project plan	Project plan, initial assessment of tools and techniques
Data understanding	
Collect initial data	Initial data collection report
Describe and explore data	Data description, exploration report
Verify data quality	Data quality report
Data preparation	
Starting point: data set	Data set, data set description
Select data	Rationale for inclusion and exclusion
Clean data	Data cleaning report
Construct data	Derived attributes, generated records
Integrate and format data	Merged/reformatted data
Modeling[1]	
Select modeling technique	Modeling technique, modeling assumptions
Generate test design	Test design
Build model	Parameter settings, models, model description
Assess model	Model assessment, revised parameter settings
Evaluation	
Evaluate results	Assessment of data mining results w.r.t. business success criteria, approved models

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

Review process	Review of process
Determine next steps	List of possible actions settings
Deployment	
Plan deployment	Deployment plan
Plan monitoring and maintenance	Monitoring and maintenance plan
Produce final report	Final report and final presentation
Review project	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.

KDD

SEMMA

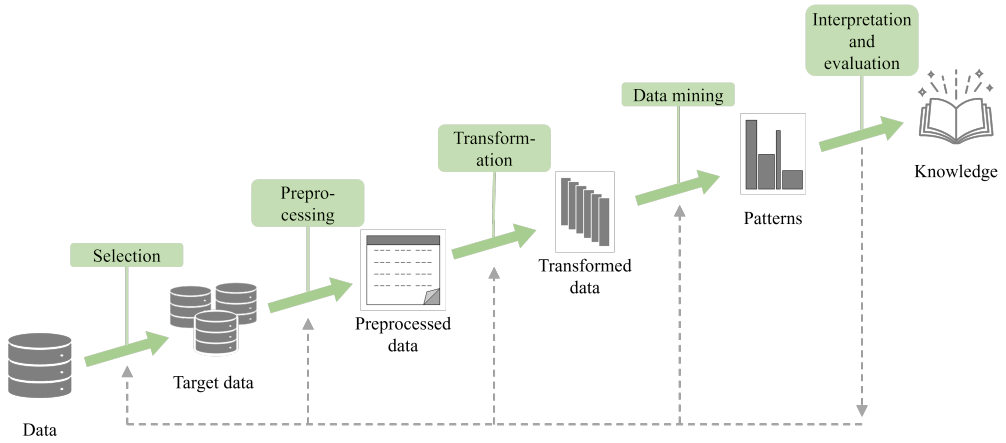


Figure 1.8: KDD process

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

L* lifecycle model

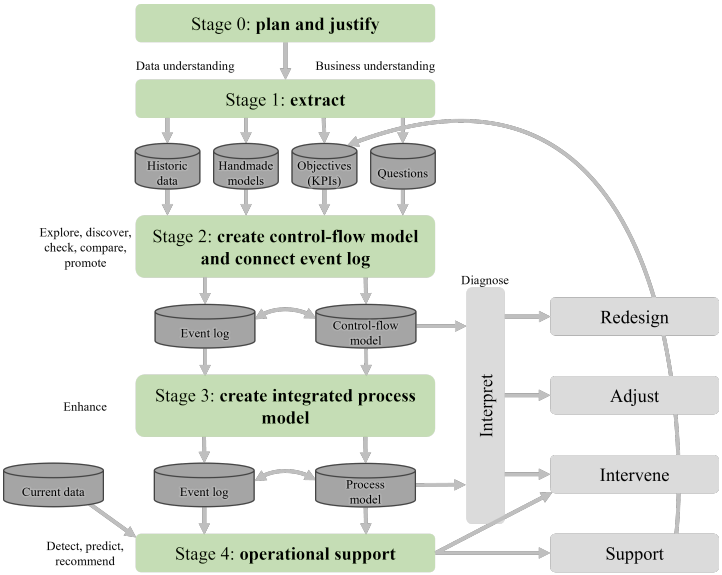


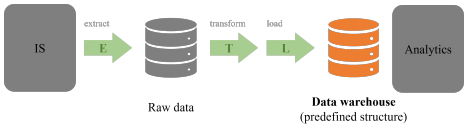
Figure 1.9: L* lifecycle model

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

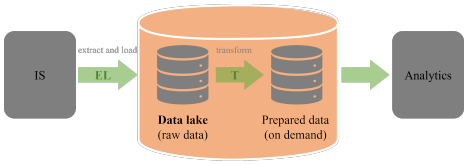
PDCA
- **PDCA** stands for Plan-Do-Check-Act and is a never-ending cycle with exactly these steps.

DMAIC
- The other one **DMAIC** stands for Define-Measure-Analyze-Improve-Control, with the following subtasks:
  - Define: launch team, establish charter, plan project, gather VOC/VOB, plan for change
  - Measure: document process, collect baseline data, narrow project focus
  - Analyze: analyze data, identify root causes, identify and remove waste
  - Improve: generate, evaluate, and optimize solutions, pilot, plan and implement
  - Control: control the process, validate project benefits

ETL
ELT
Finally, we have two processes with the same components, but different ordering of the 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.

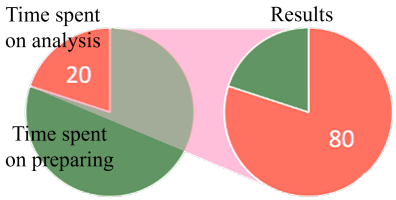


Figure 1.11: 80-20 rule

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 for 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~~'T'13:00:10~~).

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

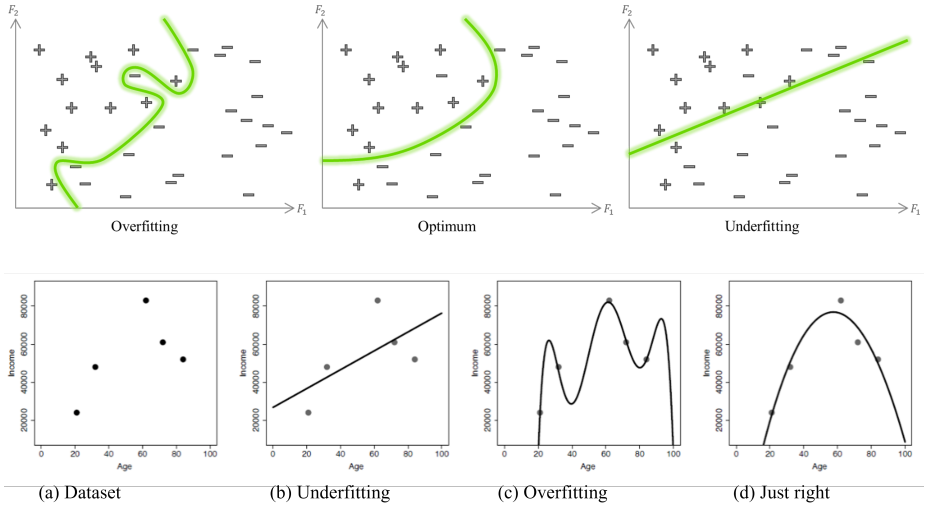


Figure 1.12: Over- and underfitting visualized

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.

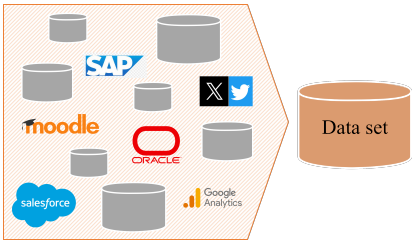


Figure 2.1: Data extraction from different sources

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.

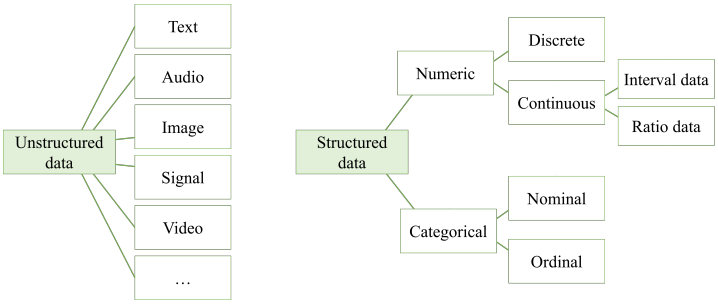


Figure 2.2: Recap: overview types of data

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.

	I		II		III		IV	
	x	y	x	y	x	y	x	y
	10	8.04	10	9.14	10	7.46	8	6.58
	8	6.95	8	8.14	8	6.77	8	5.76
	13	7.58	13	8.74	13	12.74	8	7.71
	9	8.81	9	8.77	9	7.11	8	8.84
	11	8.33	11	9.26	11	7.81	8	8.47
	14	9.96	14	8.1	14	8.84	8	7.04
	6	7.24	6	6.13	6	6.08	8	5.25
	4	4.26	4	3.1	4	5.39	19	12.5
	12	10.84	12	9.13	12	8.15	8	5.56
	7	4.82	7	7.26	7	6.42	8	7.91
	5	5.68	5	4.74	5	5.73	8	6.89
Mean of x	9		9		9		9	
Sample variance of x	11		11		11		11	
Mean of y	7.5		7.5		7.5		7.5	
Sample variance of y	4.125		4.125		4.125		4.125	
Correlation between x / y	0.816		0.816		0.816		0.816	
Linear regression line $y(x)$	$3 + 0.5\,x$		$3 + 0.5\,x$		$3 + 0.5\,x$		$3 + 0.5\,x$	

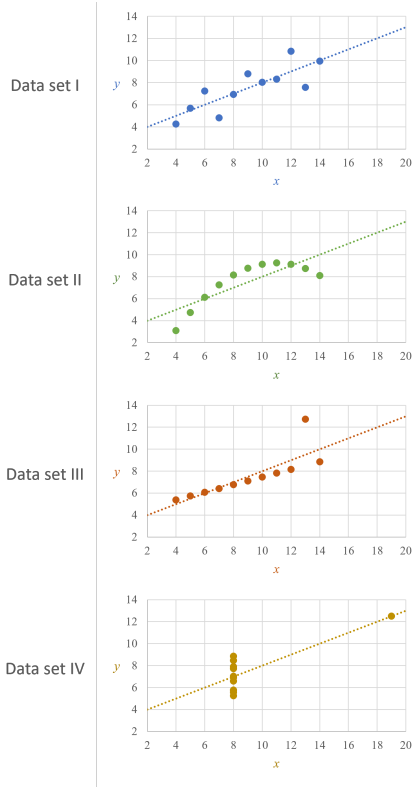


Figure 2.3: Anscombes quartet

- 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.

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:

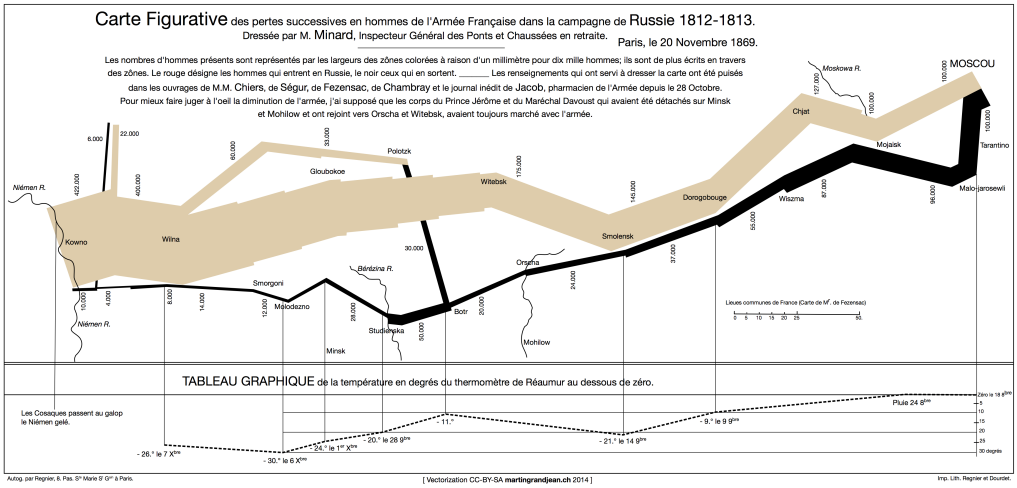


Figure 2.4: Multi-feature visualization (Napoleon's army)

- 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.

The final example highlights the importance of visualizing event or process data. In 2.5 we see the plot of different given event data input. With the visualization, some sort of trend, similarities, certain batching areas, etc. can be directly seen.

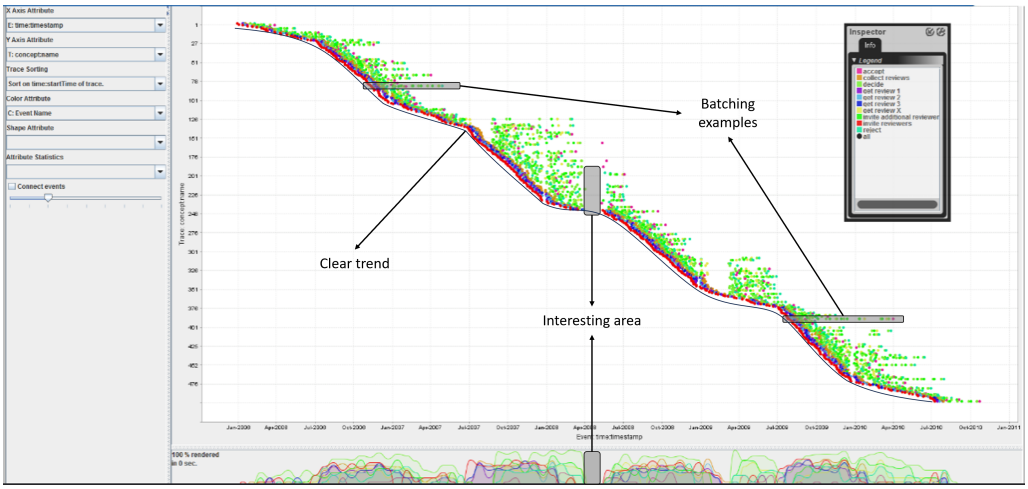


Figure 2.5: Visualization of event data

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. 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.

Investigation of individual **continuous** features

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^{st} qrt : 25^{th} 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^{rd} qrt : 75^{th} 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

Investigation of individual **categorical** feature

From **categorical features**, we can derive [2]

- 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^{nd} mode : Second most common value
- 2^{nd} mode frequ : Frequency for the second mode
- 2^{nd} 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 raw data can be found in 2.6.

To investigate the raw data further, let's first extract the resulting feature-describing tables and then also visualize the data. More specifically for the visualization, we're going to show the distributions of the different features. Figure 2.7 shows both the

[2] 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	1
2	CI	0		2	Back	Yes	15,028	60,112	1		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	7,668	92,806	3	0	0	7,668	0
304	CI	46,365	Married	1	Back	No	3,217	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	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	9,882	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	179,448	0
500	CI	0		1	Broken Limb	No	8,259	0	0	0	0	0	1

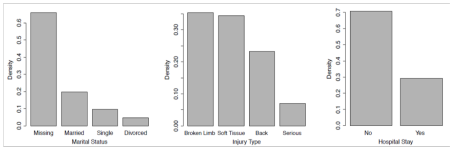
Figure 2.6: Example for single feature investigation (insurance claim fraud)

properties of the features and examplary 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/infinitely many possible feature values, group items (**binning**) and then visualize the resulting histogram.

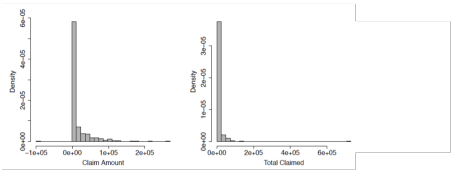
Binning

Feature	Count	% Miss.	Card.	Mode	Mode Freq.	Mode %	2nd Mode	2nd Mode Freq.	2nd 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



(a) Categorical features

Feature	Count	% Miss.	Card.	Min	1st Qrt.	Mean	Median	3rd Qrt.	Max	Std. Dev.
Income	500	0.0	171	0.0	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



(b) Numerical (continuous) features

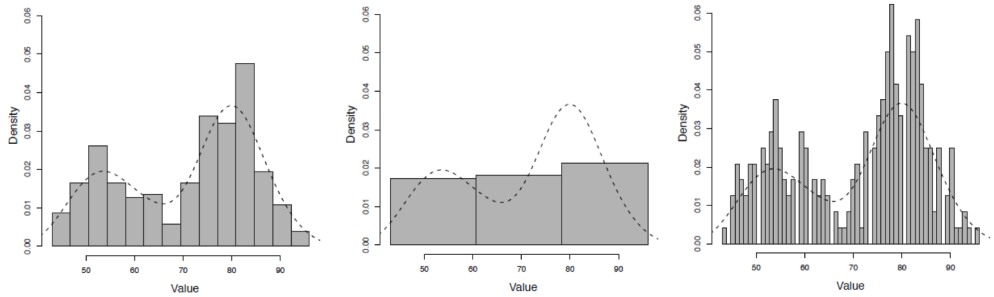
Figure 2.7: Feature-discribing table and distribution visualization

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:

Over- and underfitting for binning

- In the case of underfitting, the true function is not at all matched.

- In the case of overfitting, there exist very steep valeys, the provided data points are more learned by heart rather than abstracting to a function.



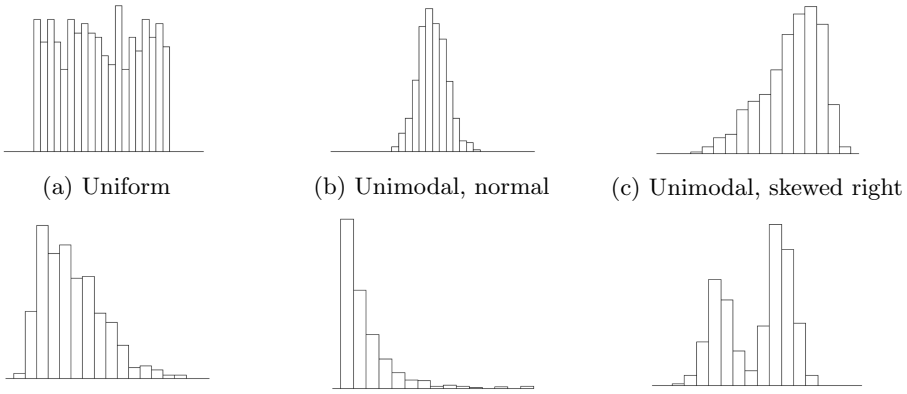
(a) Good approximation
14 bins

(b) Underfitting
3 bins

(c) Overfitting
60 bins

Figure 2.8: Binning for continuous variables

The histograms furthermore can show different types, as depicted in 2.9.



(a) Uniform

(b) Unimodal, normal

(c) Unimodal, skewed right

(d) Unimodal, skewed left

(e) Exponential

(f) Multimodal

Figure 2.9: Histogram types

Here are some further notes on the types:

Uniform
Unimodal
Multimodal
Exponential

- **Uniform** means all items have the same likelyhood (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 descrease in the likelihood over all instances.

Normal distribution

Normal distribution

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:

Mean

- The **mean** μ, so the expected value also characterizing the peak, and

- The **standard deviation** σ characterizing how narrow the peak, or the distribution around the peak, is. Standard deviation

(a) Variation of mean μ (b) Variation of standard deviation σ

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 1σ-distance of the mean,
- 95% of all observations will be within 2σ-distance of the mean, and
- 99.7% of all observations will be within 3σ-distance of the mean

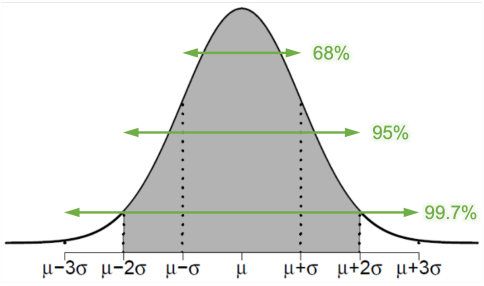


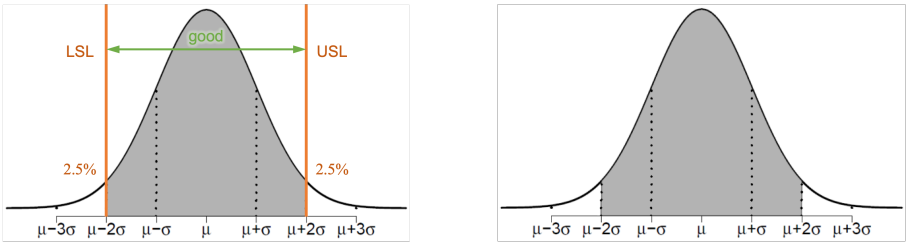
Figure 2.11: 68-95-99.7-rule

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 2σ-range, so with:
  - **LSL** (lower spcification 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 LSL
  - **USL** (upper spcification 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. USL

  Combined, $100\% - 2.5\% - 2.5\% = 95\%$ have a deviation from the mean lying withing 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 to late.



(a) Example 1 (defect tolerance) (b) Example 2 (delivery times)

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 operatiing with "six sigma quality" are assumed to have < 3.4 defects per million instances, so $\Pr(\text{error}) = 0.0000034$. It characterizes a process improvement approach. This likelyhood is a combination of $\pm 6\sigma$ tolerance and a "drift" of $\pm 1.5\sigma$.

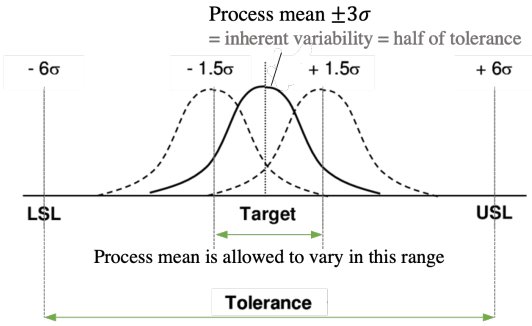


Figure 2.13: Lean six sigma

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
- **Inconsitency** - 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

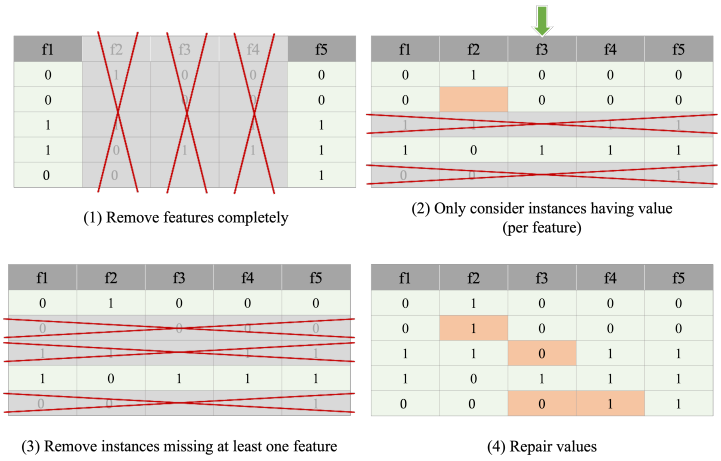
Imagine different missing features of some instances. Since some data is missing, we need to deal with this in some way. Here are the possible options: Missing features

1. Remove feature completely (for all instances)
2. Only consider instsances 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.

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

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
- 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 distribtion over months and days, days 1 to 12 are more frequent than days 13 to 31[3]

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 acutally valid.

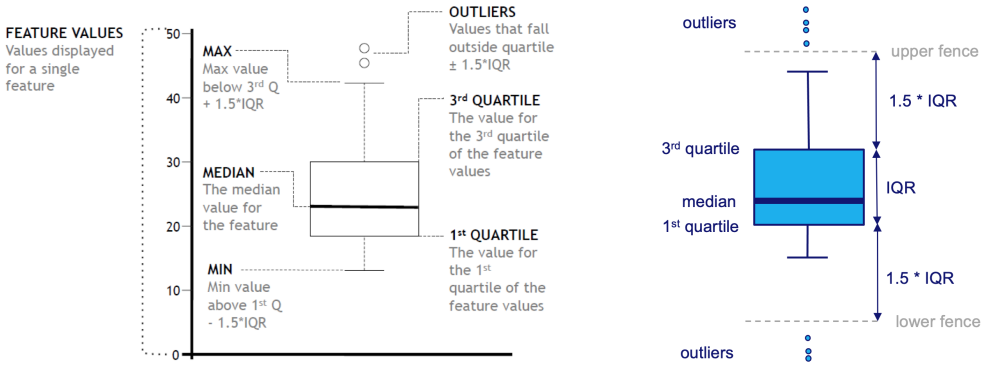
Outlier values

Outlier values

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

Box plot

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.



(a) Properties on a box plot

(b) Construction of a 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 is depicted by the "Bar" in the center.
  - The median is the "middle" value, so the number halway between lowest and highest number.

IQR

- The **IQR** , so the interquartile range, covering 50% of the "middle instances" is depicted by the "Box".
  - The first quartile is the number halway between lowest and middle number, the third halway between middle and highest number.
  - The IQR is the distance between first and third quartile.
- The upper whisker indicates the **maximal** value below the $3^{\text{rd}}\text{quartile} + 1.5 \cdot IQR$, whereas

[3] NOT the case anymore if date format DD-MM-YYYY and MM-DD-YYYY are mixed

- The lower whisker indicates the **minimal** value above the $1^{st}\text{quartile} - 1.5 \cdot IQR$.
- 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:

$$\{_{1:}\ 1,\ _{2:}\ 2,\ _{3:}\ 5,\ _{4:}\ 7,\ _{5:}\ 8,\ _{6:}\ 8,\ _{7:}\ 9,\ _{8:}\ 9,\ _{9:}\ 9,\ _{10:}\ 10,\ _{11:}\ 10,\ _{12:}\ 10,\ _{13:}\ 11,\ _{14:}\ 12,\ _{15:}\ 14,\ _{16:}\ 19,\ _{17:}\ 23\}$$

Then we construct the box diagram like this:

- The median value is 9 (at position 9).
- The first quatile 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 existance can be indicated by claming the removed values to these thresholds. The process is shown in 2.16.

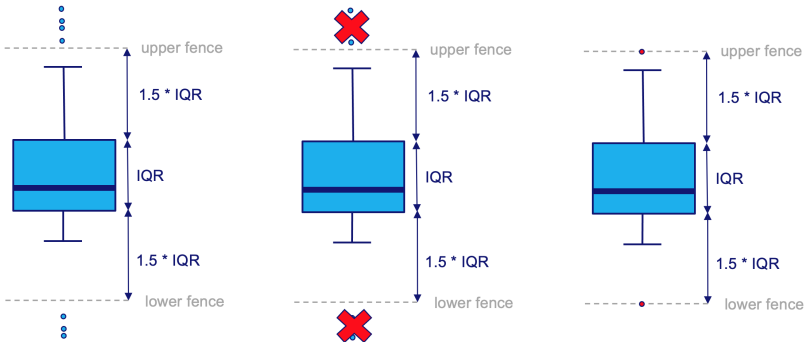


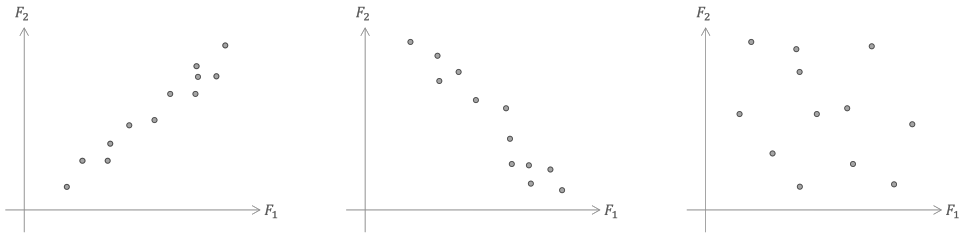
Figure 2.16: Handling outliers in box plots

2.4 Relations among features

So far, we only really looked into features separately. This section will now focus on showing how multiple features are related. We will only consider relating TWO features, but the techniques are also applicable to more.

Scatter plots

As a first step to see whether a relation exists, first plot the two features. Examples for typical resulting **correlations** is shown in 2.17. Correlation



(a) Positive correlation (b) Negative correlation (c) No correlation

Figure 2.17: Scatter plots visualizing correlations

As an example for detecting correlations, we have a data set about basketball players with different features. The raw data will not be included here. Instead, we will directly look at all correlations of the features simultaneously. This is visualized using a **scatter plot matrix** (SPLOM) as in 2.18.

SPLOM

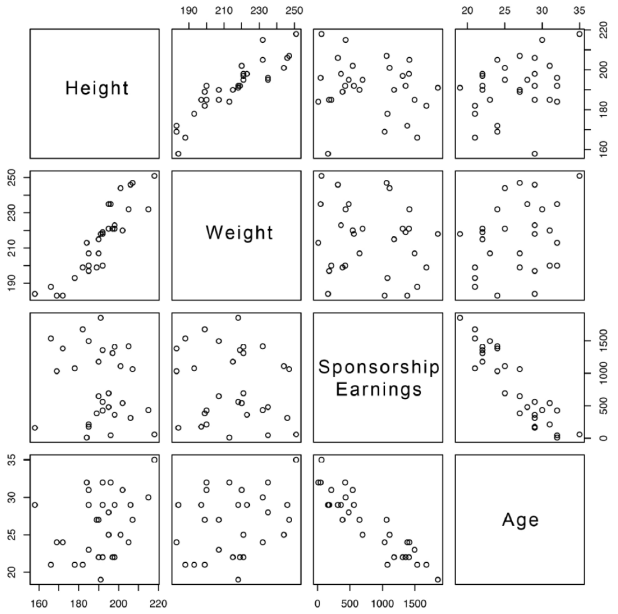


Figure 2.18: Scatter plot matrix for four features

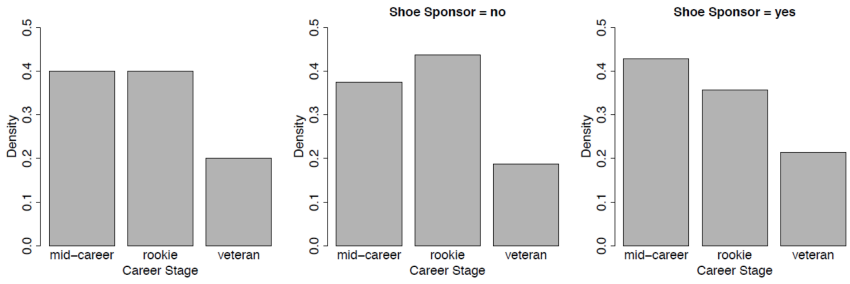
Interesting to see in such a SPLOM is the mirror axis in opposite tiles of the matrix which is an absolutely linear line (so absolute correlation, or identity, as would be displayed if a feature would be displayed on a scatter plot compared to itself). This mirroring doesn't change the nature of a correlation. If feature A is positively correlated to feature B, the same goes in the other direction (equivalent for negative correlation).

Collection of bar plots

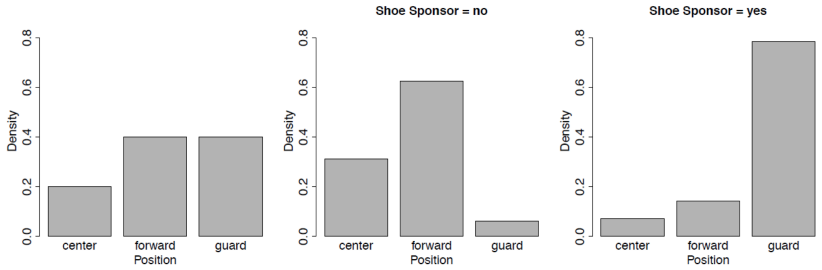
Collection of small multiple bar plots

Another way to display relations is via a collection of small multiple **bar plots** . Consider the plots in 2.19

No relation is implied when the differently conditioned and non-conditioned bar plots don't show any significant difference, as for the example of career stages. On the other



(a) Career stage: No relation to shoe sponsor



(b) Position: Strong relation to show sponsor

Figure 2.19: Collection of (conditioned) bar plots (Conditioned on shoe sponsor)

hand, a relation can be seen when the bar plots show specific differences. For example in the position case it can be seen that "guards" are more likely to have a shoe sponsor.

The difference and implied relation can be further highlighted by using **stacked bar plots** that show the conditioned percentage, as can be seen in 2.20.

Stacked bar plots

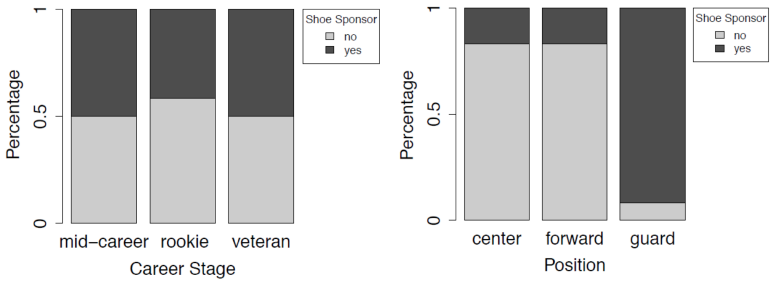
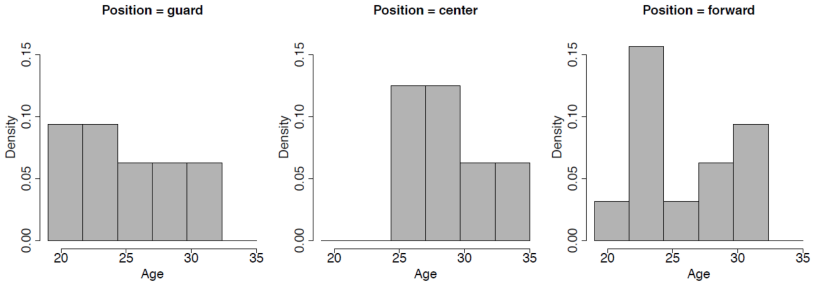


Figure 2.20: Stacked bar plots (for both career and position conditioned on shoe sponsor)

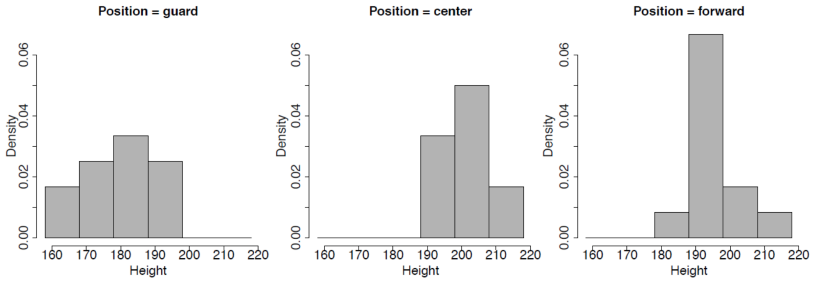
Collection of histograms and box plots

In the case of continuous variables, instead of bar plots we can use a collection of small multiple **histograms** , as displayed in 2.21.

Collection of small multiple histograms



(a) Age (6 bins): No strong relation to position

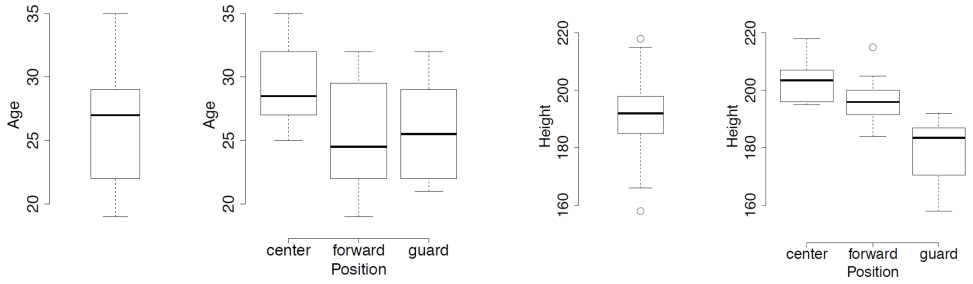


(b) Height (6 bins): Relation to position

Figure 2.21: Collection of (conditioned) histograms (Conditioned on position)

Collection of box plots

Alternatively, **box plots** can be collected and utilised to identify relations. Figure 2.22 conducts the same relation between age or height to position. It further highlights the relatively strict separation of ages for individual positions.



(a) Age: Weaker relation to position

(b) Height: Stronger relation to position

Figure 2.22: Collection of (conditioned) box plots (Conditioned on position)

Descriptive statistics

To not only see the relation, but also classify it with values, we know introduce some basic descriptive statistics. Based on n values $a_1, \ldots, a_n$, we have the **sample mean** $\overline{a}$ and **sample variance** $var(a)$ and **standard deviation** $sd(a)$ as:

Sample mean

$$\overline{a} = \frac{1}{n} \sum_{i=1}^{n} a_i$$

Sample variance

$$var(a) = \frac{\sum_{i=1}^{n} (a_i - \overline{a})^2}{n-1}$$

$$sd(a) = \sqrt{var(a)} = \sqrt{\frac{\sum_{i=1}^{n}(a_i - \overline{a})^2}{n-1}}$$ Standard deviation

A quick note on why we devide by $n-1$ and not n for the calculation of $var(a)$. This is due to the estimated mean $\overline{a}$ instead of actual or true one $\hat{a}$. Simply put, since we can only estimate, we would rather overestimate the variance (devide by smaller number) instead of underestimating (devide by learger number) it. We would call a variance calculated by deviding by n a biased estimator.

To classify the relation between features, based on n pairs of values $(a_1, b_1), \ldots, (a_n, b_n)$ we have the **sample covariance** $cov(a, b)$ and the **correlation** $corr(a, b)$ as:

$$cov(a, b) = \frac{1}{n-1}\sum_{i=1}^{n}((a_i - \overline{a}) \times (b_i - \overline{b}))$$ Sample covariance

$$corr(a, b) = \frac{cov(a, b)}{sd(a) \times sd(b)}$$ Correlation

Covariance and correlation have the following properties:

$cov(a, b) \in [-\infty, \infty]$ (unbounded)

$$cov(a, b) \text{ is } \begin{matrix}\text{positive } (+) \\ \text{negative } (-)\end{matrix} \text{ if } a \begin{matrix}(\pm) \\ (\pm)\end{matrix} \text{ and } b \begin{matrix}(\pm) \\ (\mp)\end{matrix} \text{ are } \begin{matrix}\text{the same} \\ \text{different}\end{matrix}$$

$corr(a, b) \in [-1, 1]$ (normalized)

$$corr(a, b) \begin{matrix}> 0 \\ < 0 \\ = 0\end{matrix} \text{ if } a \text{ and } b \text{ are } \begin{matrix}\text{positively correlated} \\ \text{negatively correlated} \\ \text{independent}\end{matrix}$$

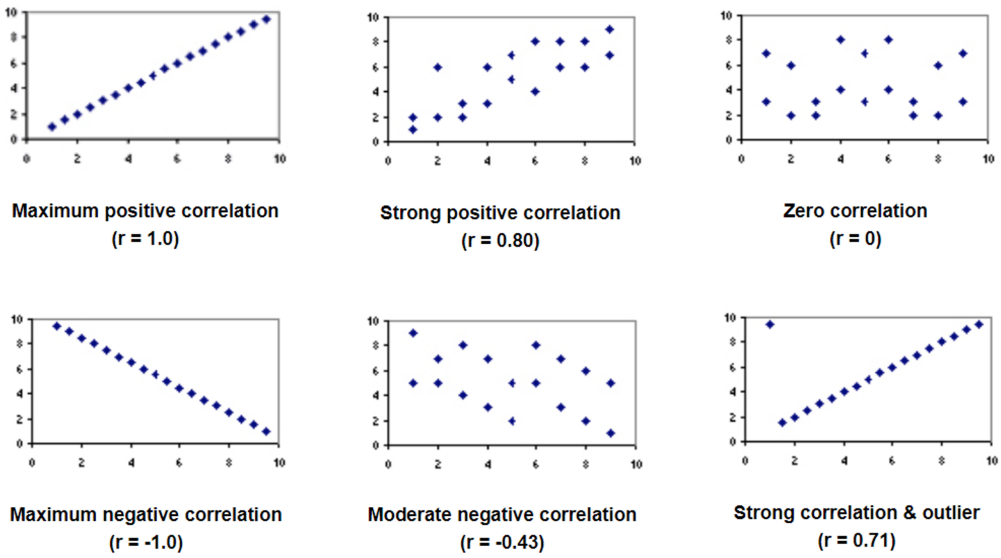


Figure 2.23: Different correlation examples with scatter plot

With the new knowledge, we can expand our previous SPLOM diagram with the **correlation matrix** values. A correlation matrix looks as follows:

Correlation matrix

$$corr\underset{\{a,b,\ldots,z\}}{\text{matrix}} = \begin{pmatrix} corr(a,a) & corr(a,b) & \cdots & corr(a,z) \\ corr(b,a) & corr(b,b) & \cdots & corr(b,z) \\ \vdots & \vdots & \ddots & \vdots \\ corr(z,a) & corr(z,b) & \cdots & corr(z,z) \end{pmatrix}$$

with $corr(f_1, f_2) = corr(f_2, f_1)$ for all $f_1, f_2 \in \{a, b, \ldots, z\}$ making the correlation matrix symmetric.

The updated SPLOM diagram, which basically contains the same information twice (just flipped) now also shows the correlation values as can be seen in 2.24.

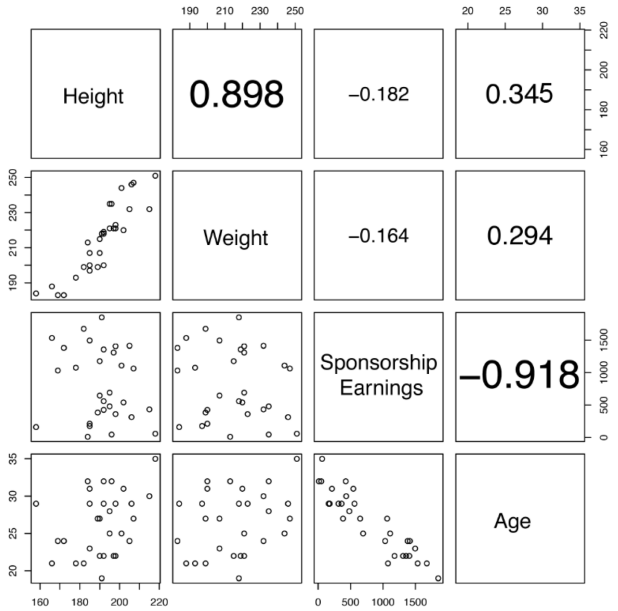


Figure 2.24: Scatter plot matrix for four feature with according *corr*-values

2.5 Preparing for analysis

2.6 Good and poor visualizations