Data Science Basics 14

by **Patrick Boily** and **Jen Schellinck**

In 2012, the *Harvard Business Review* published an article calling data science the "sexiest job of the 21st century", describing data scientists as "hybrids of data hacker, analyst, communicator, and trusted adviser" [104].

Would-be data scientists are usually introduced to the field *via* machine learning algorithms and applications. While we will discuss these topics in later chapters, we would like to start with some of the important non-technical (and semi-technical) notions that are often unfortunately swept aside in favour of diving head first into murky analysis waters.

In this chapter, we focus on some of the fundamental ideas and concepts that underlie and drive forward the discipline of data science, as well as the contexts in which these concepts are typically applied. We also highlight issues related to the ethics of practical data science. We conclude by getting a bit more concrete and considering the analytical workflow of a typical data science project, the types of roles and responsibilities that generally arise during data science projects and some basics of how to think about data, as a prelude to more technical topics.

Note: we encourage readers to take a look at Chapter 1 (*Programming Primer*) before diving into this chapter.

14.1 Introduction

The main constituent of data science is, unsurprisingly, **data**. This seems obvious, as far as statements go, but the notion of "data" is more complex than first appears.

14.1.1 What Is Data?

It is surprisingly difficult to give a clear-cut definition of **data** – we cannot even seem to agree on whether it should be used in the singular or the plural:

"the data is " vs. "the data are "

From a strictly linguistic point of view, a *datum* (borrowed from Latin) is "a piece of information;" **data**, then, should mean "pieces of information." We can also think of it as a collection of "pieces of information", and we would then use *data* to represent the whole (being potentially greater than the sum of its parts) or simply the idealized concept.

When it comes to actual data analysis, however, is the distinction really that important? Is it even clear what data is, from the definition above, and where it comes from?

Without context, does it make sense to call the following "data"?

$$4,529 \quad \text{red} \quad 25.782 \quad Y$$

To paraphrase U.S. Justice Potter Stewart, while it may be hard to define what data is, "we know it when we see it." This position may strike some of you as unsatisfying; to overcome this (sensible) objection, we will think of data simply as a collection of facts about **objects** and their **attributes**.

For instance, consider the apple and the sandwich below.

Let us say that they have the following attributes:

- *Object:* apple
  - **Shape:** spherical
  - **Colour:** red
  - **Function:** food
  - **Location:** fridge
  - **Owner:** Jen
- *Object:* sandwich
  - **Shape:** rectangle
  - **Colour:** brown
  - **Function:** food
  - **Location:** office
  - **Owner:** Pat

As long as we remember that a person or an object is not simply **the sum of its attributes**, this rough definition should not be too problematic. Note, however, that there remains some ambiguity when it comes to **measuring** (and **recording**) the attributes.

We dare say that no one has ever beheld an apple quite like the one shown above: for starters, it is a 2-dimensional representation of a 3-dimensional object. Additionally, while the overall shape of the sandwich is vaguely rectangular (as seen from above, say), it is not an exact rectangle. While no one would seriously dispute the shape attribute of the sandwich being recorded as "rectangle", a **measurement error** has occurred.

For most analytical purposes, this error may not be significant, but it is impossible to dismiss it as such for all tasks.

More problematic might be the fact that the apple's shape attribute is given in terms of a volume, whereas the sandwich's is recorded as an area; the measurement types are **incompatible**. Similar remarks can be made about all the attributes – the function of an apple may be "food" from Jen's perspective, but from the point of view of an apple tree, that is emphatically not the case; the sandwich is definitely not uniformly "brown," and so on.

A number of potential attributes are not even mentioned: size, weight, time, etc. Measurement errors and incomplete lists are always part of the picture, but most people would recognize that the collection of attributes does provide a reasonable **description** of the objects.

This is the **pragmatic** definition of data that we will use throughout.

14.1.2 From Objects and Attributes to Datasets

Raw data may exist in any format; we will reserve the term **dataset** to represent a collection of data that could conceivably be fed into algorithms for analytical purposes.

Often, these appear in a **table**, with rows and columns;[1] attributes are the **fields** (or columns) in such a dataset; objects are **instances** (or rows).

Objects can then be described by their **feature vector** – the collection of attributes associated with value(s) of interest. The feature vector for a given observation is also know as its **signature**. For instance, the dataset of physical objects could contain the following items:

ID	shape	colour	function	location	owner
1	spherical	red	food	fridge	Jen
2	rectangle	brown	food	office	Pat
3	round	white	tell time	lounge	school
. . .	. . .	. . .	. . .	. . .	. . .

We will revisit this in Section 14.5.2 (*Structuring and Organizing Data*).

14.1.3 Data in the News

We collected a sample of headlines and article titles showcasing the growing role of **data science** (DS), **machine learning** (ML), and **artificial/augmented intelligence** (AI) in different domains of society.

While these demonstrate some of the functionality/capabilities of DS/ML/AI technologies, it is important to remain aware that new technologies are always accompanied by emerging (and not always positive) social consequences.

- "Robots are better than doctors at diagnosing some cancers, major study finds" [105]
- "Deep-learning-assisted diagnosis for knee magnetic resonance imaging: Development and retrospective validation of MRNet" [106]

[1] In practice, more complex **databases** are used.

- “Google AI claims 99% accuracy in metastatic breast cancer detection” [107]
- “Data scientists find connections between birth month and health” [108]
- “Scientists using GPS tracking on endangered Dhole wild dogs” [109]
- “These AI-invented paint color names are so bad they’re good” [110]
- “We tried teaching an AI to write Christmas movie plots. Hilarity ensued. Eventually.” [111]
- “Math model determines who wrote Beatles’ *In My Life*: Lennon or McCartney?” [112]
- “Scientists use Instagram data to forecast top models at New York Fashion Week” [113]
- “How big data will solve your email problem” [114]
- “Artificial intelligence better than physicists at designing quantum science experiments” [115]
- “This researcher studied 400,000 knitters and discovered what turns a hobby into a business” [116]
- “Wait, have we really wiped out 60% of animals?” [117]
- “Amazon scraps secret AI recruiting tool that showed bias against women” [118]
- “Facebook documents seized by MPs investigating privacy breach” [119]
- “Firm led by Google veterans uses A.I. to ‘nudge’ workers toward happiness” [120]
- “At Netflix, who wins when it’s Hollywood vs.the algorithm?” [121]
- “AlphaGo vanquishes world’s top Go player, marking A.I.’s superiority over human mind” [122]
- “An AI-written novella almost won a literary prize” [123]
- “Elon Musk: Artificial intelligence may spark World War III” [124]
- “A.I. hype has peaked so what’s next?” [125]
- “That Popular AI Photo App is Stealing from Human Artists – and Worse” [126]
- “Now AI can write students’ essays for them, will everyone become a cheat?” [127]

Opinions on the topic are varied – to some, DS/ML/AI provide examples of brilliant successes, while to others it is the dangerous failures that are at the forefront.

What do you think?

14.1.4 The Analog/Digital Data Dichotomy

Humans have been collecting data for a long time. In the award-winning *Against the Grain: A Deep History of the Earliest States*, J.C. Scott argues that data collection was a major enabler of the modern nation-state (he also argues that this was not necessarily beneficial to humanity at large, but this is another matter altogether) [128].

For most of the history of data collection, humans were living in what might best be called the **analogue world** – a world where our understanding was grounded in a continuous experience of **physical reality**.

Nonetheless, even in the absence of computers, our data collection activities were, arguably, the first steps taken towards a different strategy for understanding and interacting with the world. Data, by its very nature, leads us to conceptualize the world in a way that is, in some sense, **more discrete than continuous**.

By translating our experiences and observations into numbers and categories, we re-conceptualize the world into one with sharper and more definable boundaries than our raw experience might otherwise suggest. Fast-forward to the modern world and the culmination of this conceptual discretization strategy is clear to see in our adoption of the **digital computer**, which represents everything as a series of 1s and 0s.[2]

Somewhat surprisingly, this very minimalist representational strategy has been wildly successful at **representing our physical world**, arguably beyond our most ambitious dreams, and we find ourselves now at a point where what we might call the **digital world** is taking on a reality as pervasive and important as the physical one.

Clearly, this digital world is built on top of the physical world, but very importantly, the two do not operate under the same set of rules:

- in the physical world, the default is to **forget**; in the digital world, the default is to **remember**;
- in the physical world, the default is **private**; in the digital world, the default is **public**;
- in the physical world, copying is **hard**; in the digital world, copying is **easy**.

As a result of these different rules of operation, the digital is making things that were **once hidden, visible; once veiled, transparent**. Considering data science in light of this new digital world, we might suggest that data scientists are, in essence, scientists of the **digital**, in much the same way that regular scientists are scientists of the **physical**: data scientists seek to discover the **fundamental principles of data** and understand the ways in which these fundamental principles manifest themselves in different digital phenomena.

Ultimately, however, data and the digital world are **tied to the physical world**. Consequently, what is done with data has repercussions in the physical world; and it is crucial for analysts and consultants to have a solid grasp of the fundamentals and context of data work before leaping into the tools and techniques that drive it forward.

14.2 Conceptual Frameworks for Data Work

In simple terms, we use data to represent the world. But this is not the only strategy at our disposal: we might also (and in combination) describe the world using **language**, or represent it by building **physical models**. The common thread is the more basic concept of **representation** – the idea that one object can stand in for another, and be used in its stead in order to indirectly engage with the object being represented. Humans are representational animals *par excellence*; our use of representations becomes almost undetectable to us, at times.

[2] Or 'On' and 'Off', 'TRUE' and 'FALSE'.

On some level, we do understand that “the map is not the territory”, but we do not have to make much of an effort to use the map to navigate the territory. The transition from the **representation** to the **represented** is typically quite seamless. This is arguably one of humanity’s major strengths, but in the world of data science it can also act as an Achilles’ heel, preventing analysts from working successfully with clients and project partners, and from appropriately **transferring analytical results** to the real world contexts that could benefit from them.

The best protection against these potential threats is the existence of a well thought out and explicitly described **conceptual framework**, by which we mean, in its broadest sense:

- a **specification** of which parts of the world are being represented;
- **how** they are represented;
- the **nature of the relationship** between the represented and the representing, and
- **appropriate** and **rigorous strategies** for applying the results of the analysis that is carried out in this representational framework.

It would be possible to construct such a specification from scratch, in a piecemeal fashion, for each new project, but it is worth noting that there are some overarching **modeling frameworks** that are broadly applicable to many different phenomena, which can then be moulded to fit these more specific instances.

14.2.1 Three Modeling Strategies

We suggest that there are three main not mutually exclusive **modeling strategies** that can be used to guide the specification of a phenomenon or domain:

- **mathematical** modeling;
- **computer** modeling, and
- **systems** modeling.

We start with a description of the latter as it requires, in its simplest form, no special knowledge of techniques/concepts from mathematics or computer science.

Systems Modeling

General Systems Theory was initially put forward by L. von Bertalanffy, a biologist, who felt that it should be possible to describe many **disparate** natural phenomena using a **common conceptual framework** – one which would be capable of describing many disparate phenomena, all as systems of interacting objects.

Although Bertalanffy himself presented abstracted, mathematical, descriptions of his general systems concepts, his broad strategy is relatively easily translated into a purely conceptual framework.

Within this framework, when presented with a novel domain or situation, we ask ourselves the following questions:

- which objects seem most relevant or involved in the system behaviours in which we are most interested?
- what are the properties of these objects?
- what are the behaviours (or actions) of these objects?
- what are the relationships between these objects?
- how do the relationships between objects influence their properties and behaviours?

As we find the answers to these questions about the system of interest, we start to develop a sense that we **understand the system** and its **relevant behaviours**.

By making this knowledge **explicit**, e.g. *via* diagrams and descriptions, and by sharing it amongst those with whom we are working, we can further develop a **consistent**, **shared understanding** of the system with which we are engaged. If this activity is carried out prior to data collection, it can ensure that the **right data** is collected.

If this activity is carried out after data collection, it can ensure that the process of **interpreting what the data represents** and how the latter should be used going forward is on solid footing.

Mathematical and Computer Modeling

The other modeling approaches come with their own general frameworks for interpreting and representing real-world phenomena and situations, separate from, but still compatible with, this systems perspective.

These disciplines have developed their own mathematical/digital (logical) worlds that are distinct from the tangible, physical world studied by chemists, biologists, and so on. These frameworks can be used to describe real-world phenomena by **drawing parallels** between the properties of objects in these different worlds and reasoning *via* these parallels.

Why these **constructed worlds** and the conceptual frameworks they provide are so effective at representing and describing the actual world, and thus allowing us to understand and manipulate it, is more of a philosophical question than a pragmatic one.

We will only note that they are **highly effective** at doing so, which provides the impetus and motivation to learn more about how these worlds operate, and how, in turn, they can provide data scientists with a means to engage with domains and systems through a powerful, rigorous and shared conceptual framework.

14.2.2 Information Gathering

The importance of achieving **contextual understanding** of a dataset cannot be over-emphasized. In the abstract we have suggested that this context can be gained by using conceptual frameworks. But more concretely, how does this understanding come about?

It can be reached through:

- **field trips**;
- interviews with **subject matter experts** (SMEs);

- **readings/viewings**;
- **data exploration** (even just **trying to obtain** or gain access to the data can prove a major pain),
- etc.

In general, clients or stakeholders are **not a uniform** entity – it is even conceivable that client data specialists and SMEs will **resent the involvement** of analysts (external and/or internal). Thankfully, this stage of the process provides analysts and consultants the opportunity to show that everyone is pulling in the same direction, by

- asking **meaningful** questions;
- taking an interest in the SMEs'/clients' experiences, and
- acknowledging everyone's ability to contribute.

A little tact goes a long way when it comes to information gathering.

Thinking in Systems Terms

We have already noted that a **system** is made up of **objects** with **properties** that may change over time. Within the system we perceive **actions** and **evolving properties**, leading us to think in terms of **processes**.

In order to understand how various aspects of the world interact with one another, we need to **carve out chunks** corresponding to the aspects and define their boundaries. Working with other intelligences requires this type of **shared understanding** of what is being studied. **Objects** themselves have various properties.

Natural processes generate (or destroy) objects, and may change the properties of these objects over time. We **observe**, **quantify**, and **record** particular values of these properties at particular points in time.

This process generates data points in our attempt to **capture the underlying reality** to some acceptable degree of **accuracy** and **error**, but it remains crucial for data analysts and data scientists to remember that **even the best system model only ever provides an approximation of the situation under analysis**; with some luck, experience, and foresight, these approximations might turn out to be **valid**.

Identifying Gaps in Knowledge

A **gap in knowledge** is identified when we realize that what we thought we knew about a system proves **incomplete** (or blatantly false).

This can arise as the result of a certain naïveté *vis-à-vis* the situation being modeled, but it can also be emblematic of the nature of the project under consideration: with too many moving parts and grandiose objectives, there cannot help but be knowledge gaps.[3]

Knowledge gaps might occur **repeatedly**, at any moment in the process:

- data **cleaning**;
- data **consolidation**;
- data **analysis**,
- even during **communication of the results** (!).

[3] Note that it also happens with small, well-organized, and easily contained projects. It happens all the time, basically.

When faced with such a gap, the best approach is to be flexible: **go back**, **ask questions**, and **modify the system representation** as often as is necessary. For obvious reasons, it is preferable to catch these gaps early on in the process.

Conceptual Models

Consider the following situation: you are away on business and you forgot to hand in a very important (and urgently required) architectural drawing to your supervisor before leaving. Your office will send an intern to pick it up in your living space. How would you explain to them, by phone, how to find the document?

If the intern has previously been in your living space, if their living space is comparable to yours, or if your spouse is at home, the process may be sped up considerably, but with somebody for whom the space is new (or someone with a visual impairment, say), it is easy to see how things could get complicated.

But time is of the essence – you and the intern need to get the job done **correctly** as **quickly as possible**. What is your strategy?

Conceptual models are built using methodical investigation tools:

- **diagrams**;
- structured **interviews**;
- structured **descriptions**,
- etc.

Data analysts and data scientists should beware **implicit conceptual models** – they go hand-in-hand with knowledge gaps.

In our opinion, it is preferable to err on the side of "too much conceptual modeling" than the alternative (although, at some point we have to remember that every modeling exercise is wrong[4] and that there is nothing wrong with building better models in an iterative manner, over the bones of previously-discarded simpler models).

Roughly speaking, a **conceptual model** is a model that is not implemented as a scale-model or computer code, but one which exists only conceptually, often in the form of a diagram or verbal description of a system – boxes and arrows, mind maps, lists, definitions (see Figures 14.1 and 14.2).

Conceptual models do not necessarily attempt to capture specific behaviours, but they emphasize the **possible states** of the system: the focus is on object types, not on specific instances, with **abstraction** as the ultimate objective.

Conceptual modeling is not an exact science – it is more about making internal conceptual models **explicit** and **tangible**, and providing data analysis teams with the opportunity to **examine** and **explore** their ideas and assumptions. Attempts to formalize the concept include (see Figure 14.3):

- **Universal Modeling Language** (UML);
- **Entity Relationship Models** (ER), generally connected to relational databases.

[4] "Every model is wrong; some models are useful." *George Box*.

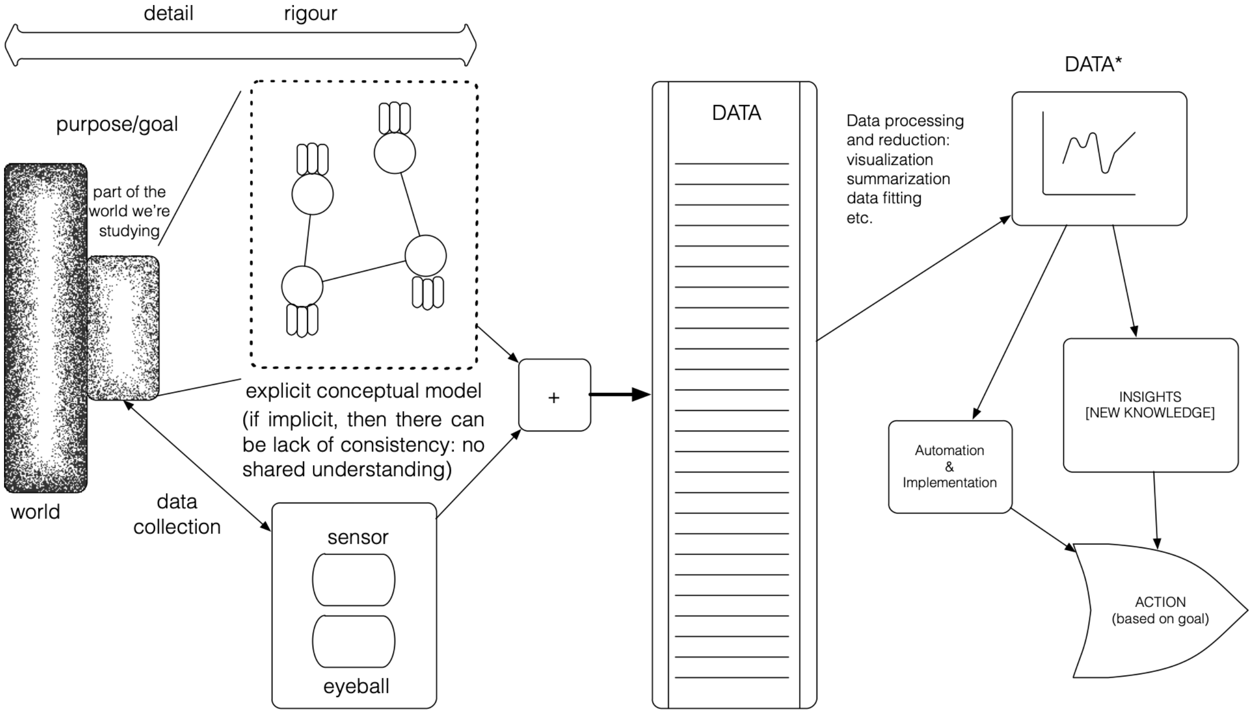

detail
rigour
purpose/goal
part of the world we're studying
world
explicit conceptual model (if implicit, then there can be lack of consistency: no shared understanding)
data collection
sensor
eyeball
+
DATA
Data processing and reduction: visualization summarization data fitting etc.
DATA*
Automation & Implementation
INSIGHTS [NEW KNOWLEDGE]
ACTION (based on goal)

Figure 14.1: A schematic diagram of systems thinking as it applies to a general problem.

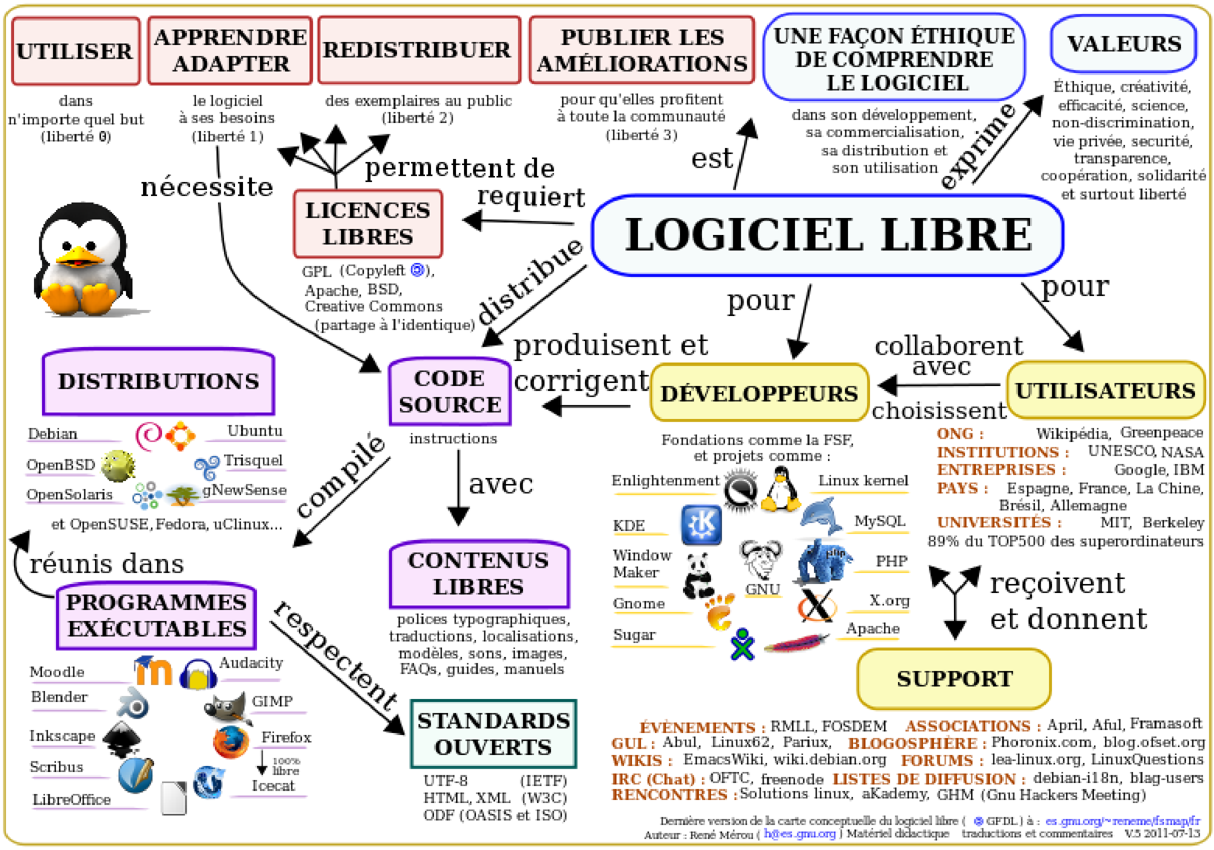

UTILISER
dans n'importe quel but (liberté 0)
APPRENDRE ADAPTER
le logiciel à ses besoins (liberté 1)
REDISTRIBUER
des exemplaires au public (liberté 2)
PUBLIER LES AMÉLIORATIONS
pour qu'elles profitent à toute la communauté (liberté 3)
UNE FAÇON ÉTHIQUE DE COMPRENDRE LE LOGICIEL
dans son développement, sa commercialisation, sa distribution et son utilisation
VALEURS
Éthique, créativité, efficacité, science, non-discrimination, vie privée, securité, transparence, coopération, solidarité et surtout liberté
nécessite
permettent de
requiert
est
exprime
LICENCES LIBRES
GPL (Copyleft), Apache, BSD, Creative Commons (partage à l'identique)
LOGICIEL LIBRE
distribue
pour
pour
produisent et corrigent
collaborent avec
choisissent
DISTRIBUTIONS
Debian
Ubuntu
OpenBSD
Trisquel
OpenSolaris
gNewSense
et OpenSUSE, Fedora, uClinux...
CODE SOURCE
instructions
DÉVELOPPEURS
Fondations comme la FSF, et projets comme :
UTILISATEURS
ONG : Wikipédia, Greenpeace
INSTITUTIONS : UNESCO, NASA
ENTREPRISES : Google, IBM
PAYS : Espagne, France, La Chine, Brésil, Allemagne
UNIVERSITÉS : MIT, Berkeley
89% du TOP500 des superordinateurs
compile
avec
Enlightenment
Linux kernel
KDE
MySQL
Window Maker
PHP
GNU
Gnome
X.org
Sugar
Apache
reçoivent et donnent
réunis dans
PROGRAMMES EXÉCUTABLES
CONTENUS LIBRES
polices typographiques, traductions, localisations, modèles, sons, images, FAQs, guides, manuels
respectent
Moodle
Audacity
Blender
GIMP
Inkscape
Firefox
100% libre
Scribus
Icecat
LibreOffice
STANDARDS OUVERTS
UTF-8 (IETF)
HTML, XML (W3C)
ODF (OASIS et ISO)
SUPPORT
ÉVÈNEMENTS : RMLL, FOSDEM
ASSOCIATIONS : April, Aful, Framasoft
GUL : Abul, Linux62, Pariux,
BLOGOSPHÈRE : Phoronix.com, blog.ofset.org
WIKIS : EmacsWiki, wiki.debian.org
FORUMS : lea-linux.org, LinuxQuestions
IRC (Chat) : OFTC, freenode
LISTES DE DIFFUSION : debian-i18n, blag-users
RENCONTRES : Solutions linux, aKademy, GHM (Gnu Hackers Meeting)
Dernière version de la carte conceptuelle du logiciel libre (GFDL) à : es.gnu.org/~reneme/fsmap/fr
Auteur : René Mérou (h@es.gnu.org) Matériel didactique traductions et commentaires V.5 2011-07-13

Figure 14.2: A conceptual model of the 'free software' system (in French) [129].

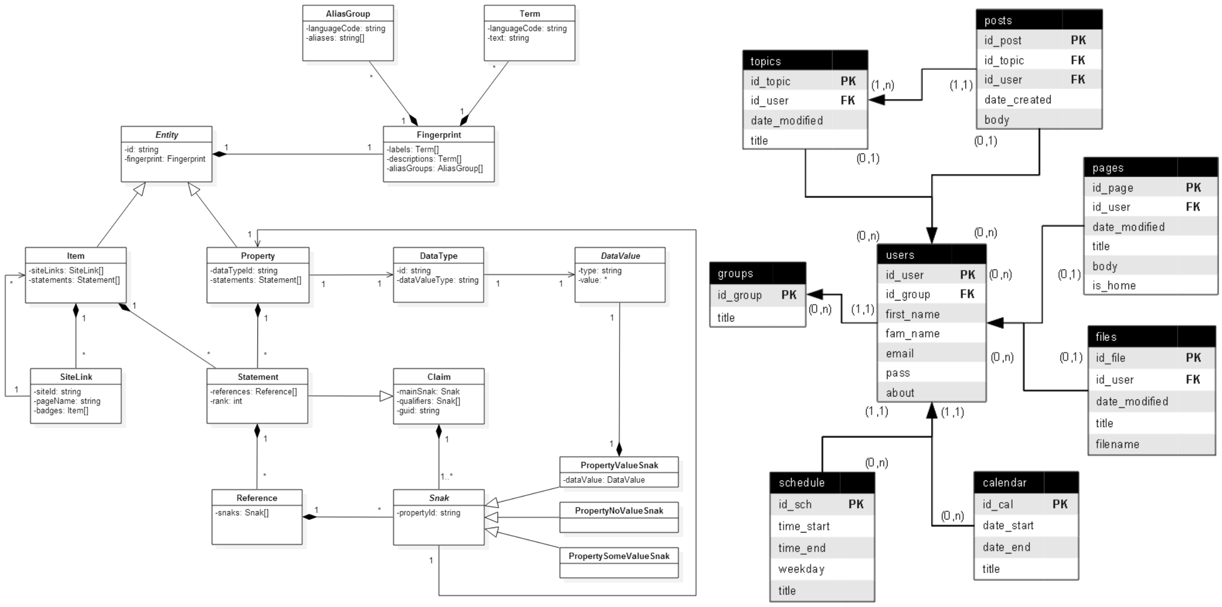
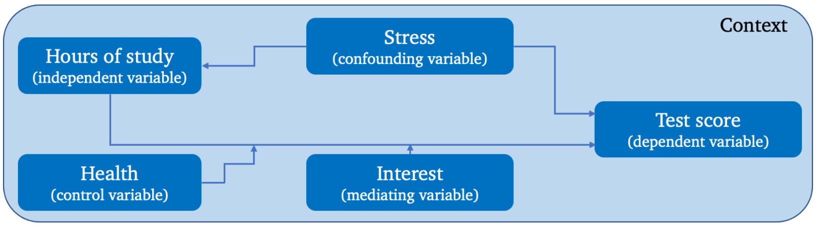


Figure 14.3: Examples of UML diagram (Wikibase Data Model, on the left [130]) and ER conceptual map (on the right [131]).

In practice, we must first select a system for the task at hand, then generate a conceptual model that encompasses:

- **relevant** and **key objects** (abstract or concrete);
- **properties** of these objects, and their values;
- **relationships between objects** (part-whole, is-a, object-specific, one-to-many), and
- **relationships between properties** across instances of an object type.

A simplistic example describing a supposed relationship between a **presumed cause** (hours of study) and a **presumed effect** (test score) is shown below:



Relating the Data to the System

From a pragmatic perspective, stakeholders and analysts alike need to know if the data which has been collected and analyzed will be useful to understand the system.

This question can best be answered if we understand:

- how the data is collected;
- the approximate nature of both data and system, and
- what the data represents (observations and features).

Is the **combination of system and data sufficient** to understand the aspects of the world under consideration? Once again, this is difficult to answer in practice. Contextual knowledge can help, but if the data, the system, and the world are **out of alignment**, any data insight drawn from mathematical, ontological, programmatical, or data models of the situation might ultimately prove useless.

14.2.3 Cognitive Biases

Adding to the challenge of building good conceptual models and using these to interpret the data is the fact that we are all vulnerable to a vast array of **cognitive biases**, which influence both how we construct our models and how we look for patterns in the data.

Such biases are difficult to detect in the spur of the moment, but making a conscious effort to identify them and setting up a clear and pre-defined set of thresholds and strategies for analysis will help reduce their negative impact. Here is a sample of such biases [132, 133]).[5]

Anchoring Bias causes us to rely too heavily on the first piece of information we are given about a topic; in a salary negotiation, for instance, whoever makes the first offer establishes a range of reasonable possibilities in both parties' minds.

Availability Heuristic describes our tendency to use information that comes to mind quickly and easily when making decisions about the future; someone might argue that climate change is a hoax because the weather in their neck of the woods has not (yet!) changed.

Choice-Supporting Bias causes us to view our actions in a positive light, even if they are flawed; we are more likely to sweep anomalous or odd results under the carpet when they arise from our own analyses.

Clustering Illusion refers to our tendency to see patterns in random events; if a die has rolled five 3's in a row, we might conclude that the next throw is more (or less) likely to come up a 3 (gambling fallacy).

Confirmation Bias describes our tendency to notice, focus on, and give greater credence to evidence that fits with our existing beliefs; gaffes made by politicians you oppose reinforces your dislike.

Conservation Bias occurs when we favour prior evidence over new information; it might be difficult to accept that there is an association between factors X and Y if none had been found in the past.

Ostrich Effect describes how people often avoid negative information, including feedback that could help them monitor their goal progress; a professor might chose to not consult their teaching evaluations, for whatever reason.

Outcome Bias refers to our tendency to judge a decision on the outcome, rather than on why it was made; the fact that analysts gave Clinton an 80% chance of winning the 2016 U.S.Presidential Election does not mean that the forecasts were wrong.

Overconfidence causes us to take greater risks in our daily lives; experts are particularly prone to this, as they are more convinced that they are right.

[5] 5: Other biases impacting our ability to make informed decisions include: bandwagon effect, base rate fallacy, bounded rationality, category size bias, commitment bias, Dunning-Kruger effect, framing effect, hot-hand fallacy, IKEA effect, illusion of explanatory depth, illusion of validity, illusory correlations, look elsewhere effect, optimism effect, planning fallacy, pro-innovation bias, representative heuristic, response bias, selective perception, stereotyping, etc.

Recency Bias occurs when we favour new information over prior evidence; investors tend to view today's market as the "forever' market and make poor decisions as a result.

Salience Bias describes our tendency to focus on items or information that are more noteworthy while ignoring those that do not grab our attention; you might be more worried about dying in a plane crash than in a car crash, even though the latter occurs more frequently than the former.

Survivorship Bias is a cognitive shortcut that occurs when a visible successful subgroup is mistaken as an entire group, due to the failure subgroup not being visible; when trying to get the full data picture, it helps to know what observations did not make it into the dataset.

Zero-Risk Bias relates to our preference for absolute certainty; we tend to opt for situations where we can completely eliminate risk, seeking solace in the figure of 0%, over alternatives that may actually offer greater risk reduction.

14.3 Ethics in the Data Science Context

> A lapse in ethics can be a conscious choice ... but it can also be negligence. [134]

In most empirical disciplines, **ethics** are brought up fairly early in the educational process and may end up playing a crucial role in researchers' activities. At Memorial University of Newfoundland, for instance, "proposals for research in the social sciences, humanities, sciences, and engineering, including some health-related research in these areas," must receive approval from specific Ethics Review Boards.

This could apply to research and analysis involving [135]:

- living human subjects;
- human remains, cadavers, tissues, biological fluids, embryos or foetuses;
- a living individual in the public arena if they are to be interviewed and/or private papers accessed;
- secondary use of data – health records, employee records, student records, computer listings, banked tissue – if any form of identifier is involved and/or if private information pertaining to individuals is involved, and
- quality assurance studies and program evaluations which address a research question.

In our experience, data scientists and data analysts who come to the field by way of mathematics, statistics, computer science, economics, or engineering, however, are not as likely to have encountered ethical research boards or to have had **formal ethics training.**[6]

Furthermore, discussions on ethical matters are often tabled, perhaps understandably, in favour of pressing technical or administrative considerations (such as algorithm selection, data cleaning strategies, contractual issues, etc.) when faced with hard deadlines.

[6] We are obviously not implying that these individuals have no ethical principles or are unethical; rather, that the opportunity to establish what these principles might be, in relation with their research, may never have presented itself.

The problem, of course, is that the current deadline is eventually replaced by another deadline, and then by a new deadline, with the end result being that the conversation may never take place. It is to address this all-too-common scenario that we take the time to discuss ethics in the **data science context**; more information is available in [136, 137].

14.3.1 The Need for Ethics

When large-scale data collection first became possible, there was to some extent a 'Wild West' mentality to data collection and use. To borrow from the old English law principle, whatever was not prohibited (from a technological perspective) was allowed.

Now, however, **professional codes of conduct** are being devised for data scientists [138–140], outlining responsible ways to practice data science – ways that are legitimate rather than fraudulent, and ethical rather than unethical.[7]

Although this shifts some added responsibility onto data scientists, it also provides them with protection from clients or employers who would hire them to carry out data science in questionable ways – they can refuse on the grounds that it is against their professional code of conduct.

14.3.2 What Is/Are Ethics?

Broadly speaking, ethics refers to the study and definition of right and wrong conduct. Ethics may consider what is right or wrong when it comes to actions in general, or consider how broad ethical principles are appropriately applied in more specific circumstances.

And, as noted by R.W. Paul and L. Elder, ethics is not (necessarily) the same as social convention, religious beliefs, or laws [144]; that distinction is not always fully understood. The following influential ethical theories are often used to frame the debate around ethical issues in the data science context.

- **Golden rule:** do unto others as you would have them do unto you;
- **Consequentialism:** the end justifies the means;
- **Utilitarianism:** act in order to maximize positive effect;
- **Moral Rights:** act to maintain and protect the fundamental rights and privileges of the people affected by actions;
- **Justice:** distribute benefits and harm among stakeholders in a fair, equitable, or impartial way.

In general, it is important to remember that our planet's inhabitants subscribe to a wide variety of ethical codes, including:

> Confucianism, Taoism, Buddhism, Shinto, Ubuntu, Te Ara Tika (Maori), First Nations Principles of OCAP, various aspects of Islamic ethics, etc.

It is not too difficult to imagine contexts in which any of these (or other ethical codes, or combinations thereof) would be better-suited to the task at hand – the challenge is to remember to **inquire** and to **heed the answers**.

[7] This is not to say that ethical issues have miraculously disappeared – Volkswagen, Whole Foods Markets, General Motors, Cambridge Analytica, and Ashley Madison, to name but a few of the big data science and data analysis players, have all recently been implicated in ethical lapses [141]. More dubious examples can be found in [142, 143].

14.3.3 Ethics and Data Science

How might these ethical theories apply to data analysis? The (former) University of Virginia's *Centre for Big Data Ethics, Law and Policy* suggested some specific examples of data science ethics questions [145]:

- who, if anyone, owns data?
- are there limits to how data can be used?
- are there value-biases built into certain analytics?
- are there categories that should never be used in analyzing personal data?
- should data be publicly available to all researchers?

The answers may depend on a number of factors, not the least of which is the matter of who is actually providing them to you. To give you an idea of some of the complexities, let us consider as an example the first of those questions: who, if anyone, owns data?

In some sense, the **data analysts** who transform the data's potential into usable insights are only one of the links in the entire chain. Processing and analyzing the data would be impossible without raw data on which to work, so the **data collectors** also have a strong ownership claim to the data.

But collecting the data can be a costly endeavour, and it is easy to imagine how the **sponsors** or **employers** (who made the process economically viable in the first place) might feel that the data and its insights are rightfully theirs to dispose of as they wish.

In some instances, the **law** may chime in as well. Indeed, one can easily list other players, but let it suffice to say that this simple question turns out to be far from easily answered, and may even change from case to case. Incidentally, this also highlights a hidden truth regarding the data analysis process: there is more to data analysis than *just* data analysis.

A similar challenge arises in regards to **open data**, where the "pro"and "anti" factions both have strong arguments (see [146–148], as well as [149] for a science-fictional treatment of the transparency vs security debate).

The answers to the above ethical questions aside, a general principle of data analysis is to **eschew the anecdotal** in favour of the **general** – from a purely analytical perspective, too narrow a focus on specific observations can end up obscuring the full picture (a vivid illustration can be found in [150]).

But data points are **not** solely marks on paper or electro-magnetic bytes on the cloud. Decisions made on the basis of data science (in all manners of contexts, from security, to financial and marketing context, as well as policy) may **affect living beings in negative ways**. And it can not be ignored that outlying/marginal individuals and minority groups often suffer disproportionately at the hands of so-called evidence-based decisions [151–153].

14.3.4 Guiding Principles

Under the assumption that one is convinced of the importance of proceeding ethically, it could prove helpful to have a set of guiding principles to aid in these efforts.

In his seminal science fiction series about *positronic robots*, Isaac Asimov introduced the now-famous *Laws of Robotics*, which he believed would have to be built-in so that robots (and by extension, any tool used by human beings) could overcome humanity's *Frankenstein*'s complex (the fear of mechanical beings) and help rather than hinder human social, scientific, cultural, and economic activities [154]:

> **1.** A robot may not injure a human being or, through inaction, allow a human being to come to harm.
>
> **2.** A robot must obey the orders given to it by human beings, except where such orders would conflict with the 1st Law.
>
> **3.** A robot must protect its own existence as long as such protection does not conflict with the 1st and 2nd Law.

Had they been uniformly well-implemented and respected, the potential for story-telling would have been somewhat reduced; thankfully, Asimov found entertaining ways to break the Laws (and to resolve the resulting conflicts) which made the stories both enjoyable and insightful.

Interestingly enough, he realized over time that a Zeroth Law had to supersede the First in order for the increasingly complex and intelligent robots to succeed in their goals. Later on, other thinkers contributed a few others, filling in some of the holes.

> **Asimov's (expanded)** ***Laws of Robotics*****:**
>
> **00.** A robot may not harm sentience or, through inaction, allow sentience to come to harm.
>
> **0.** A robot may not harm humanity, or, through inaction, allow humanity to come to harm, as long as this action/inaction does not conflict with the 00th Law.
>
> **1.** A robot may not injure a human being or, through inaction, allow a human being to come to harm, as long as this does not conflict with the 00th or the 0th Law.
>
> **2.** A robot must obey the orders given to it by human beings, except where such orders would conflict with the 00th, the 0th or the 1st Law.
>
> **3.** A robot must protect its own existence as long as such protection does not conflict with the 00th, the 0th, the 1st or the 2nd Law.
>
> **4.** A robot must reproduce, as long as such reproduction does not interfere with the 00th, the 0th, the 1st, the 2nd or the 3rd Law.
>
> **5.** A robot must know it is a robot, unless such knowledge would contradict the 00th, the 0th, the 1st, the 2nd, the 3rd or the 4th Law.

We cannot speak to the validity of these laws for **robotics** (a term coined by Asimov, by the way), but we do find the entire set satisfyingly complete.

What does this have to do with data science? Various thinkers have discussed the existence and potential merits of different sets of Laws ([155]) – wouldn't it be useful if there were *Laws of Analytics*, **moral principles that could help us conduct data science ethically**?

Best Practices

Such universal principles are unlikely to exist, but best practices have nonetheless been suggested over the years.

"Do No Harm": Data collected from an individual **should not be used to harm the individual**. This may be difficult to track in practice, as data scientists and analysts do not always participate in the ultimate decision process.

Informed Consent: Covers a wide variety of ethical issues, chief among them being that **individuals must agree to the collection and use** of their data, and that they must have a **real understanding of what they are consenting to**, and of **possible consequences** for them and others.

The Respect of "Privacy": This principle is dearly-held in theory, but it is hard to adhere to it religiously with robots and spiders constantly trolling the net for personal data. In the *Transparent Society*, D. Brin (somewhat) controversially suggests that privacy and total transparency are closely linked [147]:

> "And yes, **transparency is also the trick to protecting privacy**, if we empower citizens to notice when neighbors [*sic*] infringe upon it. Isn't that how you enforce your own privacy in restaurants, where people leave each other alone, because those who stare or listen risk getting caught?'

Keeping Data Public: Another aspect of data privacy, and a thornier issue – should some data be kept private? Most? All? It is fairly straightforward to imagine scenarios where adherence to the principle of public data could cause harm to individuals (for instance, revealing the source of a leak in a country where the government routinely jails members of the opposition), thereby contradicting the first principle against causing harm. But it is just as easy to imagine scenarios where keeping data private would have a similar effect.

Opt-in/Opt-out: Informed consent requires the ability to **not consent**, i.e., to opt out. Non-active consent is not really consent.

Anonymize Data: Identifying fields should be removed from the dataset **prior** to processing and analysis. Let any temptation to use personal information in an inappropriate manner be removed from the get-go, but be aware that this is easier said than done, from a technical perspective.

Let the Data Speak: It is crucial to absolutely restrain oneself from **cherry-picking** the data. Use all of it in some way or another; validate your analysis and make sure your results are repeatable.

14.3.5 The Good, the Bad, and the Ugly

Data projects could whimsically be classified as **good**, **bad** or **ugly**, either from a technical or from an ethical standpoint (or both). We have identified instances in each of these classes (of course, our own biases are showing):

- **good** projects increase knowledge, can help uncover hidden links, and so on: [106–108, 112, 115, 116, 122, 156–163]
- **bad** projects can lead to bad decisions, which can in turn decrease the public's confidence and potentially harm some individuals: [109, 113, 120, 121, 150]
- **ugly** projects are, flat out, unsavoury applications, even if the initial impetus for the work was noble; either they are poorly executed from a technical perspective, or they put a lot of people at risk; these (and similar approaches/studies) should be avoided: [118, 119, 151–153, 164]

14.4 Analytics Workflows

An overriding component of the discussion so far has been the **importance of context**. And although the reader may be eager at this point to move into data analysis proper, there is one more bit of context that should be considered first – the **project context**.

We have alluded to the idea that data science is much more than merely data analysis, and this is apparent when we look at the typical steps involved in a data science project. Inevitably, data analysis pieces take place within this larger project context, as well as in the context of a larger **technical infrastructure** or **pre-existing system**.

14.4.1 The "Analytical" Method

As with the **scientific method**, there is a "step-by-step" guide to data analysis:

1. statement of objective
2. data collection
3. data clean-up
4. data analysis/analytics
5. dissemination
6. documentation

Notice that **data analysis** only makes up a small segment of the entire flow.

In practice, the process often end up being a bit of a mess, with steps taken out of sequence, steps added-in, repetitions and re-takes (see Figure 14.4).

And yet ... it tends to work on the whole, if conducted correctly.

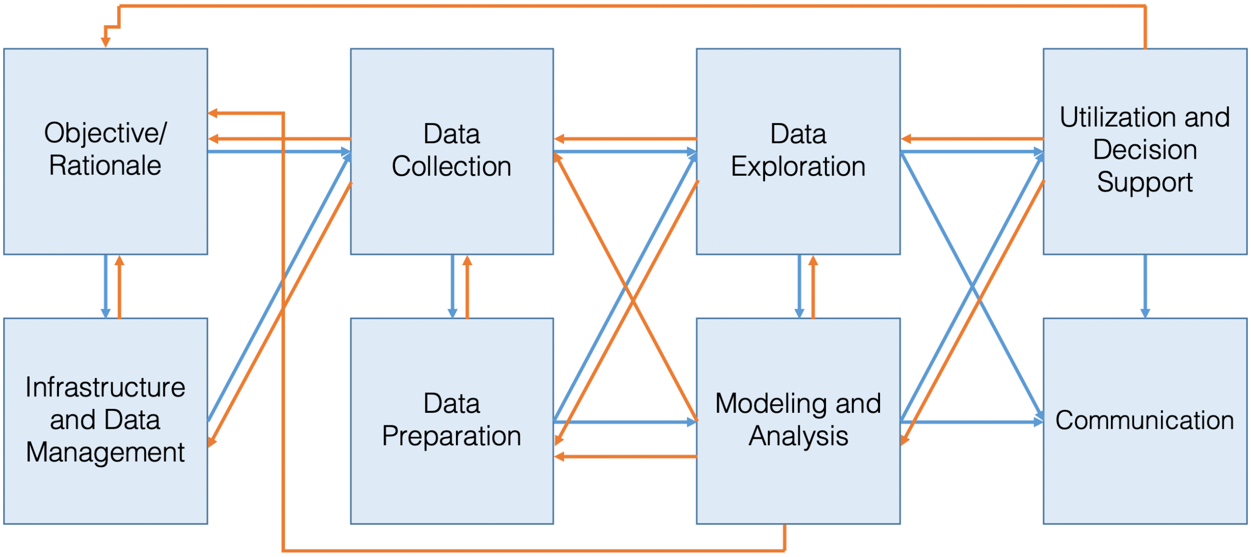
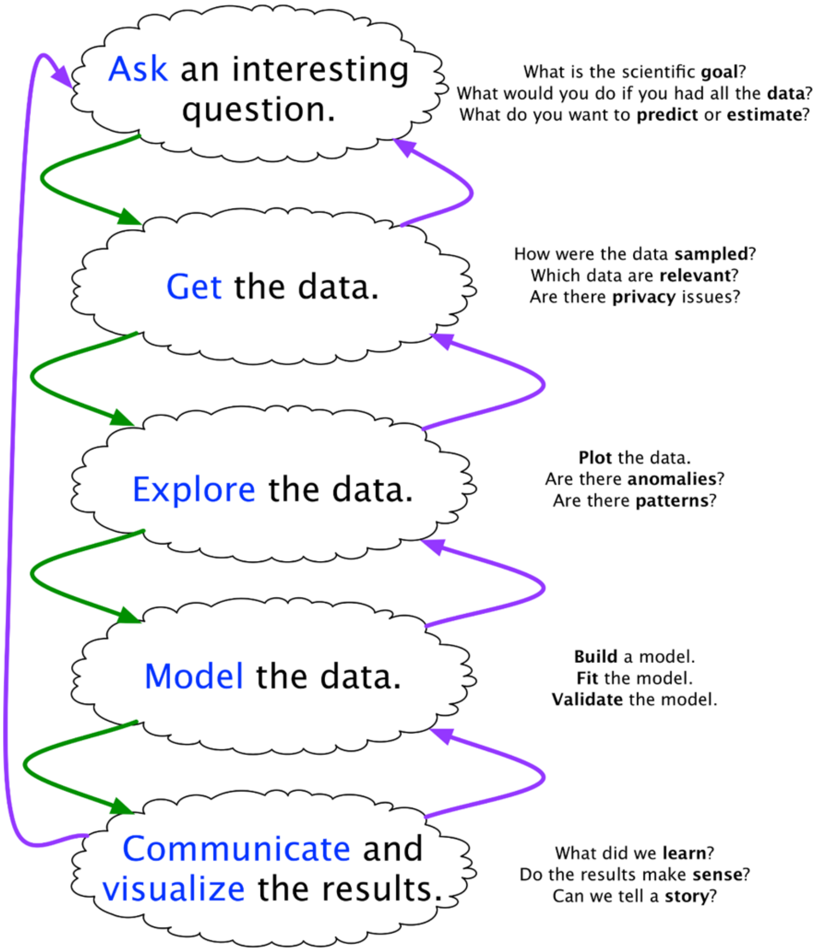


Figure 14.4: The reality of the analytic workflow – definitely not a linear process!

Blitzstein and Pfister (who teach a well-rated data science course at Harvard) provide their own workflow diagram, but the similarities are easy to spot (see below).


Ask an interesting question.
What is the scientific goal?
What would you do if you had all the data?
What do you want to predict or estimate?
Get the data.
How were the data sampled?
Which data are relevant?
Are there privacy issues?
Explore the data.
Plot the data.
Are there anomalies?
Are there patterns?
Model the data.
Build a model.
Fit the model.
Validate the model.
Communicate and visualize the results.
What did we learn?
Do the results make sense?
Can we tell a story?

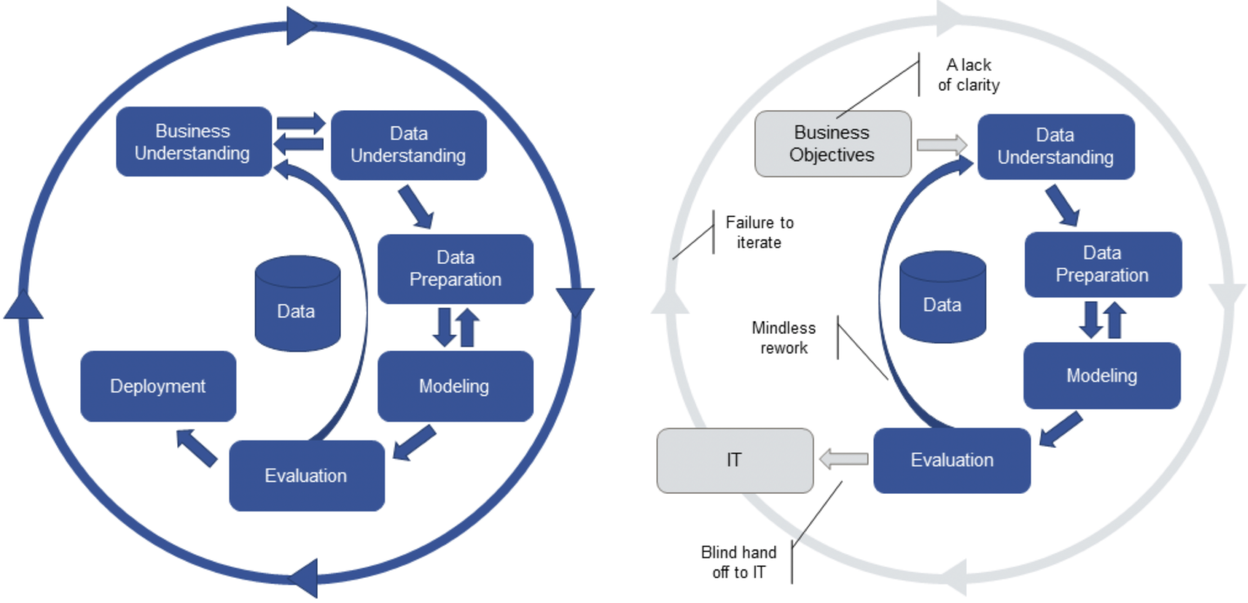


Figure 14.5: Theoretical (on the left) and corrupted (on the right) CRISP-DM processes [165].

The **Cross Industry Standard Process, Data Mining (CRISP-DM)** is another such framework, with projects consisting of 6 steps:

1. business understanding
2. data understanding
3. data preparation
4. modeling
5. evaluation
6. deployment

The process is iterative and interactive – the dependencies are highlighted in Figure 14.5. In practice, data analysis is often corrupted by:

1. lack of clarity;
2. mindless rework;
3. blind hand-off to IT, and
4. failure to iterate.

CRISP-DM has a definite old-hat flavour (as exemplified by the use of the outdated expression “data mining”), but it can be useful to check off its sub-components, if only as a **sanity check**.

Business Understanding
- understanding the business goal
- assessing the situation
- translating the goal in a data analysis objective
- developing a project plan

Data Understanding
- considering data requirements
- collecting and exploring data

Data Preparation
- selection of appropriate data
- data integration and formatting
- data cleaning and processing

Modeling

- selecting appropriate techniques
- splitting into training/testing sets
- exploring alternatives methods
- fine tuning model settings

Evaluation

- evaluation of model in a business context
- model approval

Deployment

- reporting findings
- planning the deployment
- deploying the model
- distributing and integrating the results
- developing a maintenance plan
- reviewing the project
- planning the next steps

All these approaches have a common core: data science projects are **iterative** and (often) **non-sequential**. Helping the clients and/or stakeholders recognize this central truth will make it easier for analysts and consultants to **plan the data science process** and to obtain **actionable insights** for organizations and sponsors.

The main take-away from this section, however, is that there is a great deal to consider in advance of modeling and analysis – once more, **data science is not solely about data analysis**.

14.4.2 Collection, Storage, Processing, Modeling

Data enters the **data science pipeline** by first being **collected**. There are various ways to do this:

- data may be collected in a **single pass**;
- it may be collected in **batches**, or
- it may be collected **continuously**.

The **mode of entry** may have an impact on the subsequent steps, including how frequently models, metrics, and other outputs are **updated**.

Once it is collected, data must be **stored**. Choices related to storage (and **processing**) must reflect:

- how the data is collected (mode of entry);
- how much data there is to store and process (small vs. big), and
- the type of access and processing that will be required (how fast, how much, by whom).

Unfortunately, stored data may go **stale** (both *figuratively*, as in, for example, addresses no longer accurate, names have changed, etc., and *literally*, as in the physical decay of the data and storage space); regular data audits are recommended.

The data must be **processed** before it can be analyzed. This is discussed in detail in Chapter 15 (*Data Preparation*), but the key point is that **raw data** has to be converted into a format that is **amenable to analysis**, by:

- identifying **invalid**, **unsound**, and **anomalous** entries;
- dealing with **missing values**;
- **transforming** the variables and the datasets so that they meet the requirements of the selected algorithms.

In contrast, the **analysis** step itself is almost anti-climactic – simply run the selected methods/algorithms on the processed data. The specifics of this procedure depend, of course, on the choice of method/algorithm.

We will not yet get into the details of how to make that choice[8] , but data science teams should be familiar with a fair number of techniques and approaches:

- data cleaning
- descriptive statistics and correlation
- probability and inferential statistics
- regression analysis (linear and other variants)
- survey sampling
- bayesian analysis
- classification and supervised learning
- clustering and unsupervised learning
- anomaly detection and outlier analysis
- time series analysis and forecasting
- optimization
- high-dimensional data analysis
- stochastic modeling
- distributed computing
- etc.

These only represent a **small slice** of the analysis pie. It is difficult to imagine that any one analyst/data scientist could master all (or even a majority of them) at any moment, but that is one of the reasons why data science is a team activity (more on this in Section 13.1.3, *Roles and Responsibilities*).

14.4.3 Model Assessment and Life After Analysis

Before applying the findings from a model or an analysis, one must first confirm that the model is reaching valid conclusions about the system of interest.

All analytical processes are, by their very nature, **reductive** – the raw data is eventually transformed into a small(er) **numerical outcome** (or summary) by various analytical methods, which we hope is still **related** to the system of interest, see Section 14.2 (*Conceptual Frameworks for Data Work*).

Data science methodologies include an **assessment** (evaluation, validation) phase. This does not solely provide an analytical sanity check;[9] it can also be used to determine when the system and the data science process have stepped out of alignment.

Note that past successes can lead to reluctance to re-assess and re-evaluate a model (the so-called **tyranny of past success**); even if the analytical approach has been vetted and has given useful answers in the past, it may not always do so.

[8] Truth be told, choosing wisely is probably the the most **difficult** aspect of a data science project.

[9] Are the results analytically compatible with the data, say?

At what point does one determine that the current data model is **out-of-date**? At what point does one determine that the current model is no longer **useful**? How long does it take a model to react to a **conceptual shift**?[10]

This is another reason why regular audits are recommended – as long as the analysts remain in the picture, the only obstacle to performance evaluation might be the technical difficulty of conducting said evaluation.

When an analysis or model is 'released into the wild' or delivered to the client, it often takes on a life of its own. When it inevitably ceases to be **current**, there may be little that (former) analysts can do to remedy the situation.

Data analysts and scientists rarely have full (or even partial) control over **model dissemination**. Consequently, results may be misappropriated, misunderstood, shelved, or failed to be updated, all without their knowledge. Can conscientious analysts do anything to prevent this?

Unfortunately, there is no easy answer short of advocating that analysts and consultants not only focus on data analysis, but also recognize the opportunity that arises during a project to **educate clients and stakeholders** on the importance of these auxiliary concepts.

Finally, because of **analytic decay**, it is crucial not to view the last step in the analytical process as a **static dead end**, but rather as an invitation to return to the beginning of the process.

14.4.4 Automated Data Pipelines

In the **service delivery context**, the data analysis process is typically implemented as an **automated data pipeline** to enable the analysis process to occur repeatedly and automatically.

Data pipelines usually consist of 9 components (5 **stages** and 4 **transitions**, as in Figure 14.9):

1. data collection
2. data storage
3. data preparation
4. data analysis
5. data presentation

Each of these components must be **designed** and then **implemented**. Typically, at least one pass of the data analysis process has to be done **manually** before the implementation is completed. We will return to this topic in Section 14.5.2 (*Structuring and Organizing Data*).

14.5 Getting Insight From Data

With all of the appropriate context now in mind, we can finally turn to the main attraction, **data analysis** proper. Let us start this section with a few definitions, in order to distinguish between some of the common categories of data analysis.

[10] How long does it take Netflix to figure out that you no longer like action movies and want to watch comedies instead, say? How long does it take Facebook to recognize that you and your spouse have separated and that you do not wish to see old pictures of them in your feed?

What is Data Analysis?

We view **finding patterns in data** as being data analysis's main goal. Alternatively, we describe the data analysis process as **using data to**:

- answer specific questions;
- help in the decision-making process;
- create models of the data;
- describe or explain the situation or system under investigation,
- etc.

While some practitioners include other analytical-like activities, such as testing (scientific) hypotheses, or carrying out calculations on data, we think of those as separate activities.

What is Data Science?

One of the challenges of working in the data science field is that nearly all quantitative work can be described as data science (often to a ridiculous extent). Our simple definition paraphrases T. Kwartler: data science is the collection of processes by which we extract **useful** and **actionable insights** from data. Robinson [166] further suggests that these insights usually come *via* **visualization** and (manual) **inferential analysis**.

The noted data scientist H. Mason thinks of the discipline as "the **working intersection** of statistics, engineering, computer science, domain expertise, and 'hacking' " [167].

What is Machine Learning?

Starting in the 1940s, researchers began to take seriously the idea that machines could be taught to **learn**, **adapt** and **respond** to novel situations. A wide variety of techniques, accompanied by a great deal of theoretical underpinning, were created in an effort to achieve this goal.

Machine learning is typically used to obtain "predictions" (or "advice"), while reducing the operator's analytical, inferential and decisional workload (although it is still present to some extent) [166].

What is Artificial/Augmented Intelligence?

The science fiction answer is that artificial intelligence is **non-human intelligence** that has been **engineered** rather than one that has evolved naturally. Practically speaking, this translates to "computers carrying out tasks that only humans can do". A.I. attempts to remove the need for oversight, allowing for automatic "actions" to be taken by a completely unattended system.

These goals are laudable in an academic setting, but we believe that stakeholders (and humans, in general) should not seek to abdicate all of their agency in the decision-making process. As such, we follow the lead of various thinkers and suggest further splitting A.I. into **general A.I.** (who would operate independently of human intelligence) and **augmented** intelligence (which enhances human intelligence).

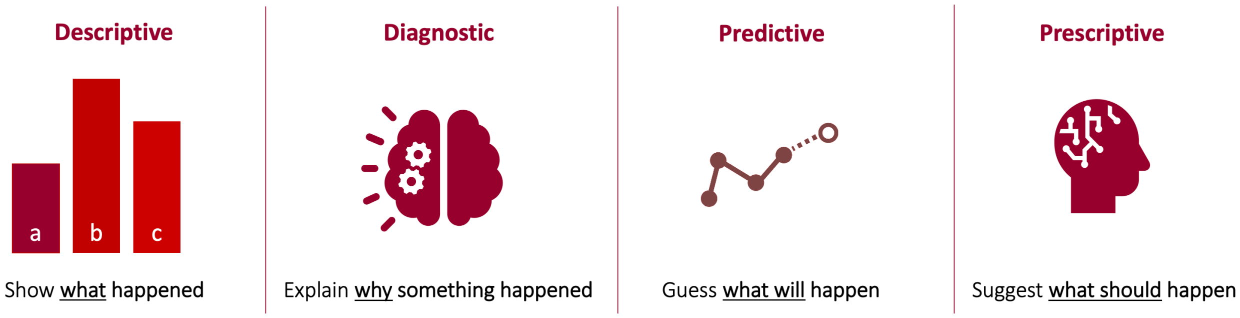


Figure 14.6: Analysis/data science buckets [Marwan Kashef].

These approaches can be further broken down into 4 **core key buckets** (see Figure 14.6), moving roughly from **low value/low difficulty** propositions (left) to **high value/high difficulty** propositions (right).

For instance, a shoe store could conduct the following analyses:

Descriptive Sales report
Diagnostic Why did the sales take a large dip?
Predictive What is the sales forecast next quarter?
Prescriptive: How should we change the product mix to reach our target sales goal?

14.5.1 Asking the Right Questions

Definitions aside, however, data analysis, data science, machine learning, and artificial intelligence are about **asking questions** and **providing answers** to these questions. We might ask various types of questions, depending on the situation.

Our position is that, from a quantitative perspective, there are only really three types of questions:

- **analytics** questions;
- **data science** questions, and
- **quantitative methods** questions.

Analytics questions could be something as simple as:

> how many clicks did a specific link on my website get?

Data science questions tend to be more complex – we might ask something along the lines of:

> if we know, historically, when or how often people click on links, can we predict how many people from Winnipeg will access a specific page on our website within the next three hours?

Whereas analytics-type questions are typically answered by **counting things**, data science-like questions are answered by using historical patterns to **make predictions**.

Quantitative methods questions might, in our view, be answered by making predictions but not necessarily based on historical data. We could build a model from **first principles** – the "physics" of the situation, as it were – to attempt to figure out what might happen.

For instance, if we thought there was a correlation between the temperature in Winnipeg and whether or not people click on the links in our website, then we might build a model that predicts "how many people from Winnipeg will access a page in the next week?", say, by trying to predict the weather instead,[11] which is not necessarily an easy task.

Analytics models do not usually predict or explain anything – they just **report** on the data, which is itself meant to represent the situation. A data mining or a data science model tends to be **predictive**, but **not necessarily explanatory** – it shows the existence of connections, of correlations, of links, but without explaining why the connections exist.

In a quantitative method model, we may start by assuming that we know what the links are, what the connections are – which presumably means that we have an idea as to why these connections exist[12] – and then we try to **explore the consequences** of the existence of these connections and these links.

This leads to a singular realization that we share with new data scientists and analysts, potentially the single most important piece of advice they will receive in their quantitative career[13] :

> **not every situation calls for analytics, data science, statistical analysis, quantitative methods, machine learning, A.I.**

Take the time to identify instances where more is asked out of the data than what it can actually yield, and be prepared to warn stakeholders, as early as possible, when such a situation is encountered.

If we cannot ask the right questions of the data, of the client, of the situation, and so on, any associated project is doomed to fail from the very beginning. Without questions to answer, analysts are wasting their time, running analyses for the sake of analysis – **the finish line cannot be reached if there is no finish line**.

In order to help clients/stakeholders, data analysts and scientists need:

- questions **to answer**;
- questions that **can be answered** by the types of methods and skills at their disposal, and
- answers that will be **recognized as answers**.

"How many clicks did this link get?" is a question that is easily answerable if we have a dataset of links and clicks, but it might not be a question that the client cares to see answered. Data analysts and scientists often find themselves in a situation where they will ask the types of questions that can be answered with the **available data**, but the answers might not actually prove useful.

From a data science perspective, the right question is one that leads to **actionable insights**. And it might mean that old data is discarded and new data is collected in order to answer it. Analysts should beware: given the sometimes onerous price tag associated with data collection, it is not altogether surprising that there will sometimes be pressure from above

[11] Questions can also be asked in an **unsupervised** manner, see [4, 168], among others, and Section 14.5.5 (*Quantitative Methods*), briefly.

[12] Unless we're talking about quantum physics and then all bets are off – nobody has the slightest idea why things happen the way they do, down there.

[13] We are not even sure we are joking when we say this...

to keep working with the available data. Stay strong – analysis on the wrong dataset is the wrong analysis!

The Wrong Questions

Wrong questions might be:

- questions that are **too broad** or **too narrow**;
- questions that **no amount of data could ever answer**,
- questions for which **data cannot reasonably be obtained**, etc.

One of the issues with "wrong" questions is that they do not necessarily "break the pipeline":

- in the **best-case scenario**, stakeholders, clients, colleagues will still recognize the answers as irrelevant.
- in the worst-case scenario, policies will erroneously be implemented (or decisions made) on the basis of answers that have not been identified as misleading and/or useless.

Framing Questions

In general, data science questions are used to:

- **solve problems** (fix pressing issues, understand why something is or isn't happening, etc.);
- **create meaningful change** (create new standards in the company, etc.),
- **support gut feelings** (approve or disprove blind intuition).

One thing to note is that individuals prefer to **answer a question quickly**, especially in their area of expertise. It is also **strongly** suggested that analysts avoid glancing over the data before they settle on the question(s), to avoid "begging the question". Finally, not that just as we can be blinded by love, we can also be blinded by solutions: the right solution to the right question is not necessarily the "sexiest" solution.

The website kdnuggets.com ↗ suggests the following roadmap to framing questions:

- Understand the problem (opportunity vs problem)
- What initial assumptions do I have about the situation?
- How will the results be used?
- What are the risks and/or benefits of answering this question?
- What stakeholder questions might arise based on the answer(s)?
- Do I have access to the data necessary for answering this question?
- How will I measure my "success" criteria?

Example: Should I buy a house? But this is a bit vague; perhaps, instead, the question could be: should I buy a single house in Scotland? [based on an example by M. Kashef]

Answer: Let's use the roadmap.

- **Understand the problem.** I've been renting for two years and feel like I'm throwing my money away. I want a chance to invest in my own space instead of someone else's.
- **What initial assumptions do I have about the situation?** It's going to be expensive but worth it – it'll be an investment that appreciates over time.
- **How will the results be used?** Either to buy a house or rent a bit longer to save more for a larger down payment.
- **What are the risks and/or benefits of answering this question?** Risk: I could put myself under immense debt and become "house poor". Benefits: I could get into the market just in time to make a fortune, and I won't have to live under the uncertainty from my landlord possibly selling his home.
- **What stakeholder questions might arise based on the answer(s)?** Would this new home be in an area that's safe for kids? Will it be close to my workplace?
- **Do I have access to the data necessary to answer this question?** Yes, through my real estate agent and online real estate brokerages, I can keep my finger on the pulse of the market.
- **How will I measure my "success" criteria?** If I manage to buy a forever home within my $600k budget, say.

Additional Considerations

Specific questions are preferred over vague questions; questions that encourage qualification/quantification are preferred over **Yes/No questions**. Here are a few examples of questions to avoid [Health Families BC]:

- Is our revenue increasing over time? Has it increased year-over-year?
- Are most of our customers from this demographic?
- Does this project have valuable ambitions for the broader department?
- How great is our hard-working customer success team?
- How often do you triple check your work?

Consider using the following questions, instead:

- What's the distribution of our revenues over the past three months?
- Where are our top 5 high-spending cohorts from?
- What are the different benefits of pursuing this project?
- What are three good and bad traits of our customer success team?
- What kind of quality assurance testing do you carry out on your deliverables?

Question Audit Checklist [The Head Game]:

1. Did I avoid creating any yes/no questions?
2. Would anyone in my team/department understand the question irrespective of their backgrounds?
3. Does the question need more than one sentence to express?
4. Is the question 'balanced' - scope is not too broad that the question will never truly be answered, or too small that the resulting impact is minimal?

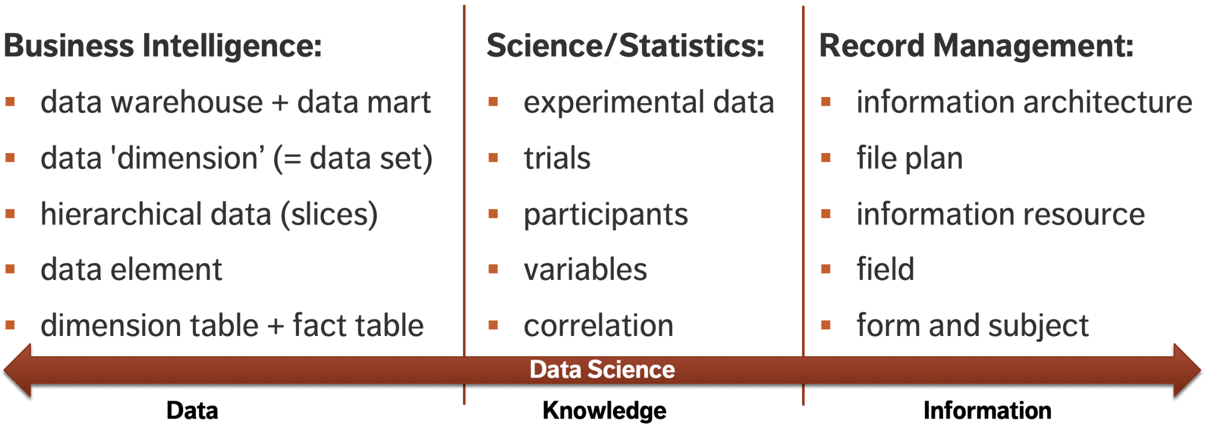


Figure 14.7: Different data cultures and terms.

5. Is the question being skewed to what may be easier to answer for my/my team's particular skillset(s)?

14.5.2 Structuring and Organizing Data

Let us now resume the discussion that was cut short in Sections 14.1.1 (*What Is Data?*) and 14.1.2 (*From Objects and Attributes to Datasets*).

Data Sources

We cannot have insights from data without data. As with many of the points we have made, this may seem trivially obvious, but there are many aspects of **data acquisition**, **structuring**, and **organization** that have a sizable impact on what insights can be squeezed from data.

Specifically, there are a number of questions that can be considered:

- why do we collect data?
- what can we do with data?
- where does data come from?
- what does "a collection" of data look like?
- how can we describe data?
- do we need to distinguish between data, information, knowledge?[14]

Historically, data has had three functions:

- **record keeping** – people/societal management;
- **science** – new general knowledge, and
- **intelligence** – business, military, police, social, domestic, personal.

Traditionally, each of these functions has:

- used different **sources** of information;
- collected **different types of data**, and
- had **different data cultures** and **terminologies**.

[14] According to the adage, "data is not information, information is not knowledge, knowledge is not understanding, understanding is not wisdom." (C.Stoll, attributed).

As data science is an interdisciplinary field, it should come as no surprise that we may run into all of them on the same project (see Figure 14.7). Ultimately, data is generated from making observations about and taking measurements of the world. In the process of doing so, we are already imposing particular **conceptualizations** and **assumptions** on our raw experience.

More concretely, data comes from a variety of sources:

- records of activity;
- (scientific) observations;
- sensors and monitoring, and
- more frequently lately, from computers themselves.

As discussed in Section 14.1.4 (*The Analog/Digital Data Dichotomy*), although data may be collected and recorded by hand, it is fast becoming a **mostly digital phenomenon**.

Computer science (and information science) has its own theoretical, **fundamental** viewpoint about data and information, operating over data in a fundamental sense – 1s and 0s that represent numbers, letters, etc. Pragmatically, the resulting data is now stored on computers, and is accessible through our world-wide computer network.

While data is necessarily a representation of **something**, analysts should endeavour to remember that the data itself still has **physical properties**: it takes up physical space and requires energy with which to work. In keeping with this physical nature, data also has a shelf life – it ages over time. We use the phrases "**rotten data**/**decaying data**" in two senses:

- **literally**, as the data storage medium might decay, but also
- **metaphorically**, as when it no longer accurately represents the relevant objects and relationships (or even when those objects no longer exist in the same way) – compare with "analytical decay", see Section 14.4.3 (*Model Assessment and Life After Analysis*).

Useful data must stay 'fresh' and 'current', and avoid going 'stale' – but that is both **context-** and **model-dependent**!

Before the Data

The various data disciplines share some **core concepts** and elements, which should resonate with the systems modeling framework previously discussed in Section 14.2 (*Conceptual Frameworks for Data Work*):

- all objects have **attributes**, whether concrete or abstract;
- for multiple objects, there are **relationships** between these objects and attributes, and
- all these elements evolve over time.

The **fundamental relationships** include:

- part–whole;
- is–a;
- is–a–type–of;
- cardinality (one-to-one, one-to-many, many-to-many),
- etc.,

while **object-specific relationships** include:

- ownership;
- social relationship;
- becomes;
- leads-to,
- etc.

Objects and Attributes

We can examine concretely the ways in which objects have properties, relationships and behaviours, and how these are captured and turned into data through observations and measurements, *via* the apple and sandwich example of Section 14.1.1 (*What Is Data?*).

There, we **made measurements** on an apple instance, **labeled the type of observations** we made, and **provided a value describing** the observation. We can further use these labels when observing other apple instances, and associate new values for these new apple instances.

Regarding the fundamental and object specified relationships, we might be able to see that:

- an apple is a type of fruit;
- a sandwich is part of a meal;
- this apple is owned by Jen;
- this sandwich becomes fuel, etc.

It is worth noting that while this all seems tediously obvious to adult humans, it is not so from the perspective of a toddler, or an artificial intelligence. Explicitly, “understanding” requires a basic grasp of:

- categories;
- instances;
- types of attributes;
- values of attributes, and
- which of these are important or relevant to a specific situation or in general terms.

From Attributes to Datasets

Were we to run around in an apple orchard, measuring and jotting down the height, width and colour of 83 different apples completely haphazardly on a piece of paper, the resulting data would be of limited value; although it would technically have been recorded, it would be lacking in **structure**.

We would not be able to tell which values were heights and which were widths, and which colours or which widths were associated with which heights, and *vice-versa*. **Structuring** the data using **lists**, **tables**, or even **tree structures** allows analysts to **record** and **preserve** a number of important relationships:

- those between object types and instances, property/attribute types (sometimes also called fields, features or dimensions), and values;

- those between one attribute value and another value (i.e., both of these values are connected to this object instance);
- those between attribute types, in the case of hierarchical data, and
- those between the objects themselves (e.g., this car is owned by this person).

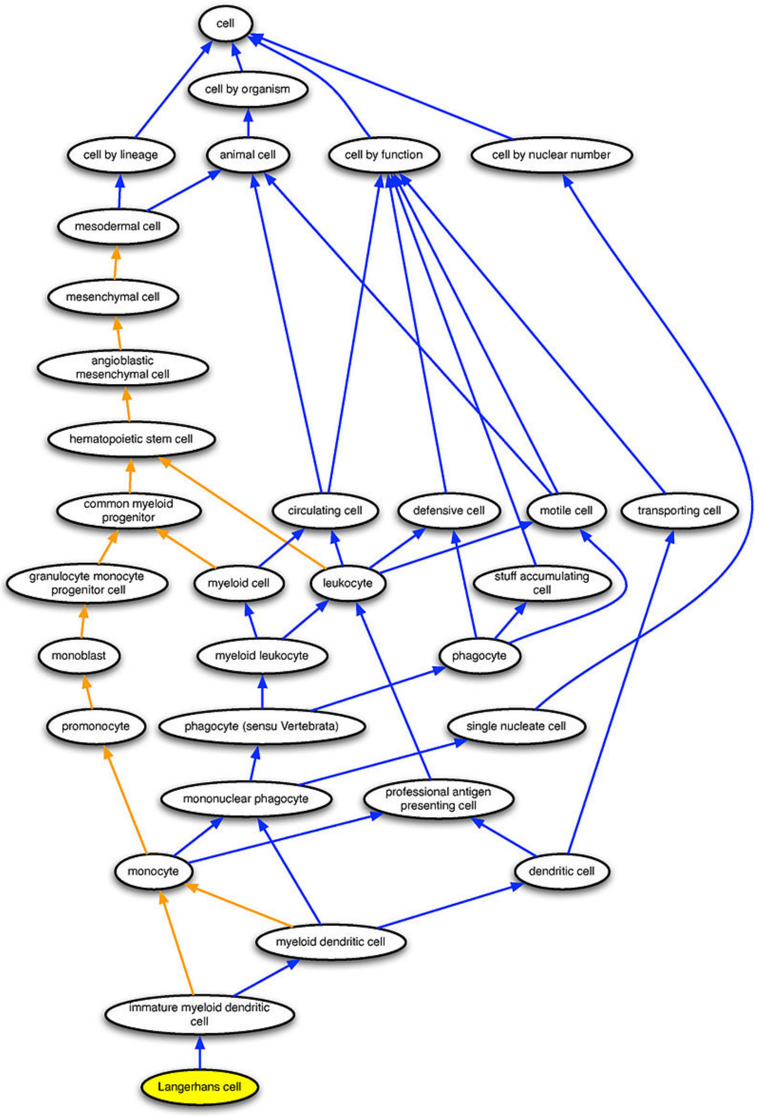
Tables, also called flat files, are likely the most familiar strategy for structuring data in order to preserve and indicate relationships. In the digital age, however, we are developing increasingly sophisticated strategies to store the **structure of relationships** in the data, and finding new ways to work with these increasingly complex relationship structures.

Formally, a **data model** is an abstract (logical) description of both the **dataset structure** and the **system**, constructed in terms that can be implemented in data management software.In a sense, data models lie halfway between **conceptual models** and **database implementations**. The data proper relates to **instances**; the model to **object types**.

Ontologies provide an alternative representation of the system: simply put, they are **structured**, **machine-readable** collections of **facts** about a domain.[15] In a sense, an ontology is an attempt to get closer to the level of detail of a full conceptual model, while keeping the whole machine-readable.

For instance, the image below is a representation of the Langerhans cells in the *Cell Ontology* [169].

[15] We could facetiously describe ontologies as "data models on steroids."



Every time we move from a conceptual model to a specific type of model (a data model, a knowledge model), we lose some information. One way to preserve as much context as possible in these new models is to also provide rich **metadata** – data about the data! Metadata is crucial when it comes to successfully working with and across datasets. **Ontologies** can also play a role here, but that is a topic for another day.

Typically data is stored in a **database**. A major motivator for some of the new developments in types of databases and other data storing strategies is the increasing availability of **unstructured** and (so-called) '**BLOB**' data.

- **Structured data** is labeled, organized, and discrete, with a pre-defined and constrained form. With that definition, for instance, data that is collected *via* an e-form that only uses drop-down menus is structured.
- **Unstructured data**, by comparison, is not organized, and does not appear in a specific pre-defined data structure – the classical example is text in a document. The text may have to subscribe to specific syntactic and semantic rules to be understandable, but

in terms of storage (where spelling mistakes and meaning are irrelevant), it is highly unstructured since any data entry is likely to be completely different from another one in terms of length, etc.
- The acronym "BLOB" stands for **B**inary **L**arge **Ob**ject data, such as images, audio files, or general multi-media files. Some of these files can be structured-like (all pictures taken from a single camera, say), but they are usually quite unstructured, especially in multi-media modes.

Not every type of database is well-suited to all data types. Let us look at four currently popular database options in terms of fundamental **data and knowledge** modeling and structuring strategies:

- key-value pairs (e.g. JSON);
- triples (e.g. resource description framework – RDF);
- graph databases, and
- relational databases.

Key-Value Stores

In these, all data is simply stored as a giant list of keys and values, where the 'key' is a name or a label (possibly of an object) and the 'value' is a value associated with this key; **triple** stores operate on the same principle, but data is stored according to 'subject – predicate – object'.

The following examples illustrate these concepts:

1. The *apple type – apple colour* key-value store might contain
   - `Granny Smith -- green`, and
   - `Red Delicious -- red`.
2. The *person – shoe size* key-value store might contain
   - `Jen Schellinck -- women's size 7`, and
   - `Colin Henein -- men's size 10`.
3. Other key-value stores: *word – definition*, *report name – report (document file)*, *url – webpage*.
4. Triples stores just add a *verb* to the mix: *person – is – age* might contain
   - `Elowyn -- is -- 20`;
   - `Llewellyn -- is -- 9`, and
   - `Gwynneth -- is -- 6`;

   while *object – is-colour – colour* might contain
   - `apple -- is-colour -- red`, and
   - `apple -- is-colour -- green`.

Both strategies results in a large amount of flexibility when it comes to the 'design' of the data storage, and not much needs to be known about the data structure prior to implementation. Additionally, missing values do not take any space in such stores.

In terms of their **implementation**, the devil is in the details; note that their extreme flexibility can also be a flaw [170], and it can be difficult to query them and find the data of interest.

Graph Databases

In **graph databases**, the emphasis is placed on the relationships between different **types of objects**, rather than between an object and the properties of that object:

- the objects are represented by **nodes**;
- the relationships between these objects are represented by **edges**, and
- objects can have a relationship with other objects of the same type (such as *person is-a-sibling-of person*).

They are fast and intuitive when using relation-based data, and might in fact be the only reasonable option to use in that case as traditional databases may slow to a crawl. But they are probably too specialized for non relation-based data, and they are not yet widely supported.

Relational Databases

In **relational databases**, data is stored in a series of tables. Broadly speaking, each table represents a type of object and some properties related to this type of object; special columns in tables connect object instances across tables (the entity-relationship model diagram (ERD) of Figure 14.3 is an example of a relational database model).

For instance, a person lives in a house, which has a particular address. Sometimes that property of the house will be stored in the table that stores information about individuals; in other cases, it will make more sense to store information about the house in its own table.

The form of relational databases are driven by the **cardinality** of the relationships (one-to-one, one-to-many, or many-to-many). These concepts are illustrated in the cheat sheet found in Figure 14.8.

Relational databases are widely supported and well understood, and they work well for many types of systems and use cases. Note however, that it is difficult to modify them once they have been implemented and that, ironically, they do not really handle relationships all that well.

Spreadsheets

We have said very little about keeping data in a single giant table (**spreadsheet**, **flat file**), or multiple spreadsheets (we purposely kept it out of the original list of modeling and structuring strategies).

On the positive side, spreadsheets are efficient when working with:

- **static data** (e.g., it is only collected once), or
- data about **one particular type of object** (e.g., scientific studies).

Most implementations of analytical algorithms require the data to be found in **one location** (such as an `R` data frame). Since the data will eventually need to be exported to a flat file anyway, why not remove the middle step and work with spreadsheets in the first place?

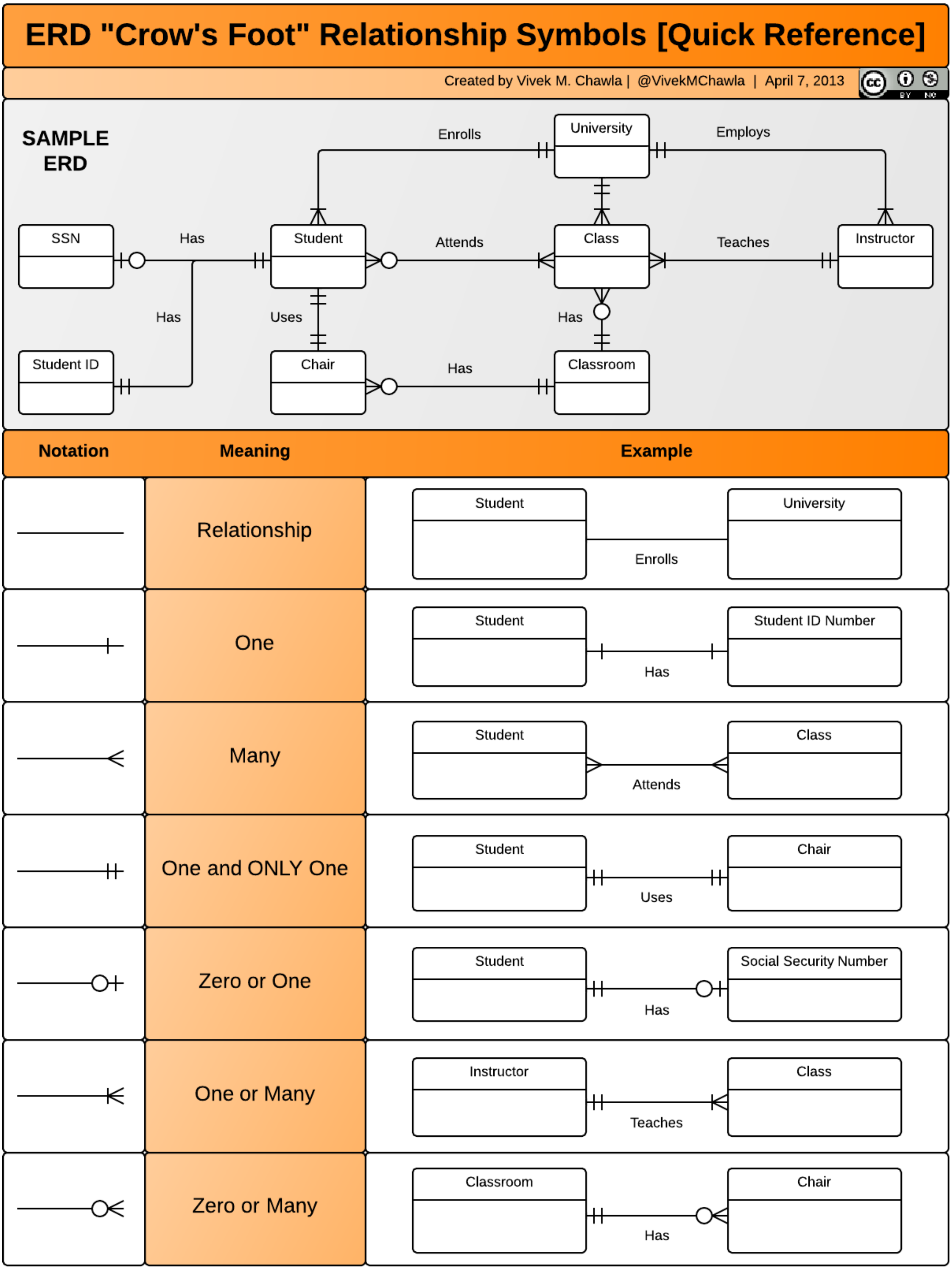


Figure 14.8: Entity-relationship model diagram (so-called) crow's foot relationship symbols cheat sheet [171].

The problem is that it is hard to manage **data integrity** with spreadsheets over the long term when data is collected (and processed) **continuously**. Furthermore, flat files are not ideal when working with systems involving many different types of objects and their relationships, and they are not optimized for querying operations.

For small datasets or quick work, flat files are often a reasonable option; we should look for alternatives when working on **large scale projects**.

All in all, we have provided very little in the way of concrete information on the topic of databases and data stores. Be aware that, time and time again, projects have **sunk** when this aspect of the process has not been taken seriously. Simply put, serious analyses cannot be conducted properly without the **right data infrastructure**.

Implementing a Model

In order to **implement** the data/knowledge model, data engineers and database specialists need access to **data storage** and **management software**. Gaining this access might be challenging for individuals or small teams as the required software traditionally runs on **servers**.

A server allows multiple users to access the database **simultaneously**, from different client programs. The other side of the coin is that servers make it difficult to 'play' with the database.

User-friendly **embedded database software** (vs client-server database engines) such as SQLite can help overcome some of these obstacles. **Data management software** lets human agents interact easily with their data – in a nutshell, they are a **human–data interface**, through which

- data can be **added** to a data collection;
- subsets can be extracted from a data collection based on certain filters/criteria, and
- data can be deleted from (or edited in) a data collection.

But *tempora mutantur, nos et mutamur in illis*[16] – we used to speak of:

- databases and database management systems;
- data **warehouses** (data management system designed to enable **analytics**);
- data **marts** (used to retrieve client-facing data, usually oriented to a specific business line or team);
- **Structured Query Language** (SQL, a commonly-used programming language that helps manage (and perform operations on) relational databases),

we now speak of (see [172]):

- data **lakes** (centralized repository in which to store structured/unstructured data alike);
- data **pools** (a small collection of shared data that aspires to be a data lake, someday);
- data **swamps** (unstructured, ungoverned, and out of control data lake in which data is hard to find/use and is consumed out of context, due to a lack of process, standards and governance);
- database **graveyards** (where databases go to die?),

[16] "Times change, and we change with them." *C.Huberinus*

and data might be stored in **non-traditional** data structures, such as

> Popular NoSQL database software include: ArangoDB, MongoDB, Redis, Amazon DynamoDB, OrientDB, Azure CosmosDB, Aerospike, etc.

Once a logical data model is complete, we need only:

1. **instantiate** it in the chosen software;
2. **load** the data, and
3. **query** the data.

Traditional relational databases use SQL; other types of databases either use **other query languages** (AQL, semantic engines, etc.) or rely on **bespoke (tailored) computer program**s (e.g. written in R, Python, etc.).

Once a data collection has been created, it must be **managed**, so that the data remains **accurate**, **precise**, **consistent**, and **complete**. **Databases decay**, after all; if a data lake turns into a data swamp, it will be difficult to squeeze usefulness out of it!

Data and Information Architectures

There is no single correct structure for a given collection of data (or dataset).

Rather, consideration must be given to:

- the **type of relationships** that exist in the data/system (and are thought to be important);
- the **types of analysis** that will be carried out, and
- the **data engineering requirements** relating to the time and effort required to extract and work with the data.

The chosen structure, which stores and organizes the data, is called the **data architecture**. Designing a specific architecture for a data collection is a necessary part of the data analysis process. The data architecture is typically embedded in the larger **data pipeline infrastructure** described in Section 14.4.4 (*Automated Data Pipelines*).

As another example, **automated data pipelines** in the **service delivery context** are usually implemented with 9 components (5 **stages**, and 4 **transitions**, as in Figure 14.9):

1. data collection
2. data storage
3. data preparation
4. data analysis
5. data presentation

Note that **model validation** could be added as a sixth stage, to combat model “drift”.

By analogy with the human body, the **data storage** component, which houses the data and its architecture, is the “heart” of the pipeline,[17] whereas the **data analysis** component is its “brain.”[18]

[17] The engine that makes the pipeline go

[18] What does that make the other components?

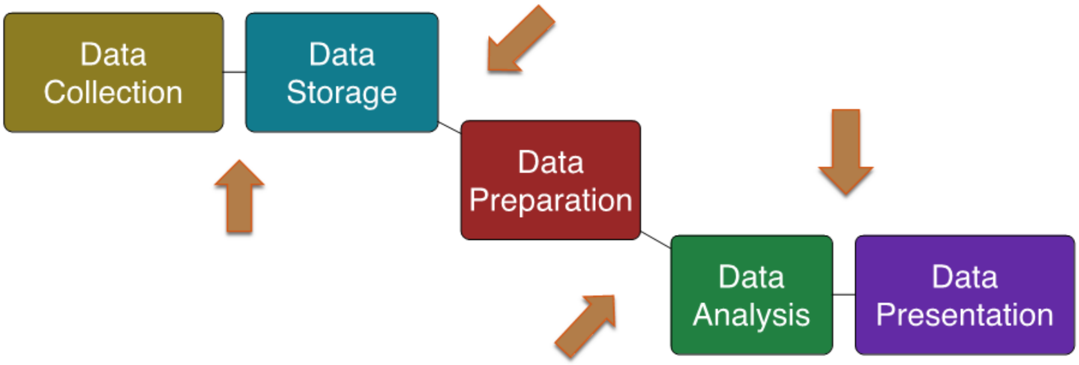


Figure 14.9: An implemented automated pipeline; note the transitions between the 5 stages.

Most analysts are familiar with mathematical and statistical models which are implemented in the data analysis component. **Data models**, by contrast, tend to get constructed separately from the analytical models at the data storage phase. This separation can be problematic if the analytical model is not compatible with the data model.[19]

If the data comes from forms with various fields stored in a relational database, the discrepancy could create difficulties on the data preparation side of the process. Building both the analytical model and the data model off of a **common conceptual model** might help the data science team avoid such quandaries.

In essence, the task is to structure and organize both data and knowledge so that it can be:

- stored in a useful manner;
- added to easily;
- usefully and efficiently extracted from that store (the "**extract-transform-load**" (ETL) paradigm), and
- operated over by humans and computers alike (programs, bots, A.I.) with minimal external modification.

14.5.3 Basic Data Analysis Techniques

Business Intelligence (BI) has evolved over the years:

1. we started to recognize that data could be used to **gain a competitive advantage** at the end of the 19th century;
2. the 1950s saw the first **business database** for decision support;
3. in the 1980s and 1990s, computers and data became increasingly available (**data warehouses, data mining**);
4. in the 2000s, the trend was to take business analytics out of the hands of data miners (and other specialists) and into the hands of **domain experts**,
5. now, **big data** and specialized techniques have arrived on the scene, as have **data visualization**, **dashboards**, and **software-as-service**.

Historically, BI has been one of the streams contributing to modern-day data science, via:

- **system of interest:** the commercial realm, specifically, the market of interest;

[19] As an example, if an analyst needs a flat file (with variables represented as columns) to feed into an algorithm implemented in R, say.

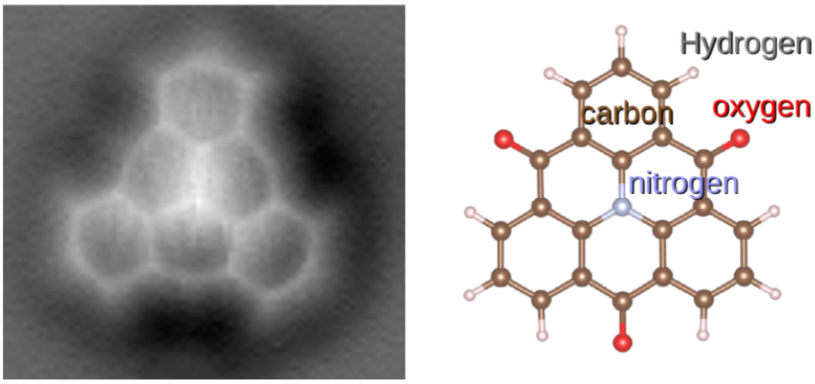


Figure 14.10: AFM image of 1,5,9-trioxo-13-azatriangulene (left) and its chemical structure model (right) [173].

- **sources of data:** transaction data, financial data, sales data, organizational data;
- **goals:** provide awareness of competitors, consumers and internal activity and use this to support decision making,
- **culture and preferred techniques:** data marts, key performance indicators, consumer behaviour, slicing and dicing, business ‘facts’.

But no matter the realm in which we work, the ultimate goal remains the same: **obtaining actionable insight into the system of interest**. This can be achieved in a number of ways. Traditionally, analysts hope to do so by seeking:

- **patterns** – predictable, repeating regularities;
- **structure** – the organization of elements in a system, and
- **generalization** – the creation of general or abstract concepts from specific instances (see Figure 14.10).

The underlying analytical **hope** is to find patterns or structure in the data from which **actionable insights** arise.

While finding patterns and structure can be interesting in its own right (in fact, this is the ultimate reward for many scientists), in the data science context it is how these discoveries are used that trumps all.

Variable Types

In the example of a conceptual model shown in Figure @ref(fig:simple-conceptual-model), we have identified different types of variables. In an **experimental setting**, we typically encounter:

- **control/extraneous variables** – we do our best to keep these controlled and unchanging while other variables are changed;
- **independent variables** – we control their values as we suspect they influence the dependent variables,
- **dependent variables** – we do not control their values; they are generated in some way during the experiment, and presumed dependent on the other factors.

For instance, we could be interested in the **plant height** (dependent) given the **mean number of sunlight hours** (independent), given the **region of the country** in which each test site is located (control).

Data Types

Variables need not be of the same **type**. We may encounter:

- **numerical** data – integers or numerics: 1, −7, 34.654, 0.04, etc.;
- **text** data – strings of text, which may be restricted to a certain number of characters, such as “Welcome to the park”, “AAAAA”, “345”, “45.678”, etc.;
- **categorical** data – are variables with a fixed number of values, may be numeric or represented by strings, but for which there is no specific or inherent ordering, such as (‘red’,‘blue’,‘green’), (‘1’,‘2’,‘3’), etc.,
- **ordinal** data – categorical data with an inherent ordering; unlike **integer** data, the spacing between values is not well-defined (very cold, cold, tepid, warm, super hot).

We use the following artificial dataset to illustrate some of the concepts.

Creating the artificial dataset

```
set.seed(0)      # for replicability
n.sample = 165   # num. of observations

colour=factor(c("red","blue","green"))   # var 1: colour
p.colour=c(40,15,5)                      # parameters

year=factor(c(2012,2013))                # var 2: year
p.year=c(60,40)                          # parameters

quarter=factor(c("Q1","Q2","Q3","Q4"))   # var 3: quarter
p.quarter=c(20,25,30,35)                 # parameters

signal.mean=c(14,-2,123)                 # var 4: signal
p.signal.mean=c(5,3,1)                   # parameters
signal.sd=c(2,8,15)
p.signal.sd=c(2,3,4)

s.colour <- sample(length(colour),       # var 1: colour
                   n.sample,             # sample
                   prob=p.colour,
                   replace=TRUE)

s.year <- sample(length(year),           # var 2: year
                 n.sample,               # sample
                 prob=p.year,
                 replace=TRUE)

s.quarter <- sample(length(quarter),     # var 3: quarter
                    n.sample,            # sample
                    prob=p.quarter,
                    replace=TRUE)

s.mean <- sample(length(signal.mean),    # var 4: signal
                 n.sample,               # sample (mean)
                 prob=p.signal.mean,
                 replace=TRUE)

s.sd <- sample(length(signal.sd),        # var 4: signal
               n.sample,                 # sample (sd)
               prob=p.signal.mean,
               replace=TRUE)

signal <- rnorm(n.sample,                # var 4: signal
                signal.mean[s.mean],     # sample
                signal.sd[s.sd])

new_data <- data.frame(colour[s.colour], # creating a
                       year[s.year],     # data frame
                       quarter[s.quarter],
                       signal)

colnames(new_data) <- c("colour",        # renaming the
                        "year",          # variables
                        "quarter",
                        "signal")

new_data |>                              # displaying the
  dplyr::slice_head(n = 6)               # first 6 obs
```

ID	colour	year	quarter	signal
1	blue	2013	Q2	22.998
2	red	2012	Q1	12.456
3	red	2012	Q4	9.935
4	red	2012	Q3	5.047
5	blue	2013	Q2	6.142
6	red	2012	Q4	13.498

(Do you understand what the code does?)

We can transform categorical data into numeric data by generating **frequency counts** of the different values/levels of the categorical variable; regular analysis techniques could then be used on the now numeric variable.[20]

```
table(new_data$colour)
```

colour	Freq
blue	41
green	10
red	114

Categorical data plays a special role in data analysis:

[20] A similar approach underlies most of modern text mining, natural language processing, and categorical anomaly detection. Information usually gets lost in the process, which explains why meaningful categorical analyses tend to stay fairly simple.

- in data science, categorical variables come with a **pre-defined** set of values;
- in experimental science, a **factor** is an independent variable with its levels being defined (it may also be viewed as a category of treatment),
- in business analytics, these are called **dimensions** (with members).

However they are labeled, these variable can be used to **subset** or **roll up/summarize** the data.

Hierarchical / Nested / Multilevel Data

When a categorical variable has multiple levels of abstraction, new categorical variables can be created from these levels. We can view these levels as new categorical variables, in a sense. The 'new' categorical variable has pre-defined relationships with the more detailed level.

This is commonly the case with time and space variables – we can 'zoom' in or out, as needed, which allows us discuss the **granularity** of the data, i.e., the 'maximum zoom factor' of the data.

For instance, observations could be recorded hourly, and then further processed (mean value, total, etc.) at the daily level, the monthly level, the quarterly level, the yearly level, etc., as seen below.

Let us start with the number of observations by year and quarter:

```
library(tidyverse)    # to be able to use
                      # group_by() and summarise()
new_data |>
  group_by(year, quarter) |>
  summarise(n = n())
```

year	quarter	n	year	quarter	n
2012	Q1	21	2013	Q1	14
2012	Q2	17	2013	Q2	11
2012	Q3	30	2013	Q3	20
2012	Q4	37	2013	Q4	15

We can also roll it up to the number of observations by year:

```
new_data |>           # no need to load tidyverse again
  group_by(year) |>
  summarise(n = n())
```

year	n
2012	105
2013	60

Data Summaries

The **summary statistics** of variables can help analysts gain basic **univariate insights** into the dataset (and hopefully, into the system with which it is associated).

These data summaries do not typically provide the full picture and connections/links between different variables are often missed altogether. Still, they often give analysts a **reasonable sense** for the data.[21]

Common summary statistics include:

- **min** – smallest value taken by a variable;
- **max** – largest value taken by a variable;
- **median** – "middle" value taken by a variable;
- **mean** – average value taken by a variable;
- **mode** – most frequent value taken by a variable;
- **# of obs** – number of observations for a variable;
- **missing values** – # of missing observations for a variable;
- **# of invalid entries** – number of invalid entries for a variable;
- **unique values** – unique values taken by a variable;
- **quartiles**, **deciles**, **centiles**;
- **range**, **variance**, **standard deviation**;
- **skew**, **kurtosis**,
- **total**, **proportion**, etc.

We can also perform operations over subsets of the data – typically over its columns, in effect **compressing** or '**rolling up**' multiple data values into a single **representative value**, as below, say.

We start by creating a mode function (there isn't one in R):

Defining the mode function

```
mode.R <- function(x) {
   unique.x <- unique(x)
   unique.x[which.max(tabulate(match(x, unique.x)))]
}
```

Data scientists often have to create their own routines/functions from scratch; there is nothing wrong with borrowing from sites such as StackOverflow, but it is important to make sure that we understand what those routines do.

The data can then also be summarized using:

Summarizing the data I

```
new_data |>     # no need to load tidyverse anew
 summarise(n = n(),
           signal.mean=mean(signal),
           signal.sd=sd(signal),
           colour.mode=mode.R(colour))
```

[21] At least for **a first pass**.

n	signal.mean	signal.sd	colour.mode
165	20.70894	38.39866	red

Typical roll-up functions include the '`mean`', '`sum`', '`count`', and '`variance`', but these do not always give sensical outcomes: if the variable measures a proportion, say, the sum of that variable over all observations is a meaningless quantity, on its own.

We can apply the same roll-up function to many different columns, thus providing a **mapping** (list) of columns to values (as long as the computations all make sense – this might mean that all variables need to be of the same type in some cases).

We can map the mode to some dataset variables:

Summarizing the data II

```
new_data |>      # still no need to re-load the tidyverse
 summarise(year.mode=mode.R(year),
           quarter.mode=mode.R(quarter),
           colour.mode=mode.R(colour))
```

year.mode	quarter.mode	colour.mode
2012	Q4	red

Datasets can also be summarized *via* contingency and pivot tables. A **contingency table** is used to examine the relationship between two **categorical** variables – specifically the frequency of one variable relative to a second variable (this is also known as **cross-tabulation**).

Here is a contingency table, by colour and year:

Contingency table (by colour and year)

```
table(new_data$colour,new_data$year)
```

	2012	2013
blue	21	20
green	6	4
red	78	36

A contingency table, by colour and quarter:

Contingency table (by colour and quarter)

```
table(new_data$colour,new_data$quarter)
```

	Q1	Q2	Q3	Q4
blue	5	8	16	12
green	2	0	5	3
red	28	20	29	37

A contingency table, by year and quarter:

Contingency table (by year and quarter)

```
table(new_data$year,new_data$quarter)
```

	Q1	Q2	Q3	Q4
2012	21	17	30	37
2013	14	11	20	15

A **pivot table**, on the other hand, is a table generated in a software application by applying operations (e.g. `sum`, `count`, `mean`) to variables, possibly based on another (categorical) variable. Here is a pivot table of signal characteristics by colour:

Pivot table (signal characteristics by colour)

```
new_data |> group_by(colour) |>
 summarise(n = n(),
           signal.mean=mean(signal),
           signal.sd=sd(signal))
```

colour	n	signal.mean	signal.sd
blue	41	25.58772	40.64504
green	10	30.79947	49.71225
red	114	18.06916	36.51887

Contingency tables are special instances of pivot tables, where the roll-up function is `count`.

Analysis Through Visualization

Consider the broad definition of analysis as:

- identifying patterns or structure, and
- adding meaning to these patterns or structure by **interpreting** them in the context of the system.

There are two general options to achieve this:

1. use analytical methods of varying degrees of sophistication, and/or
2. **visualize** the data and use the brain's analytic (perceptual) power to reach meaningful conclusions about these patterns.

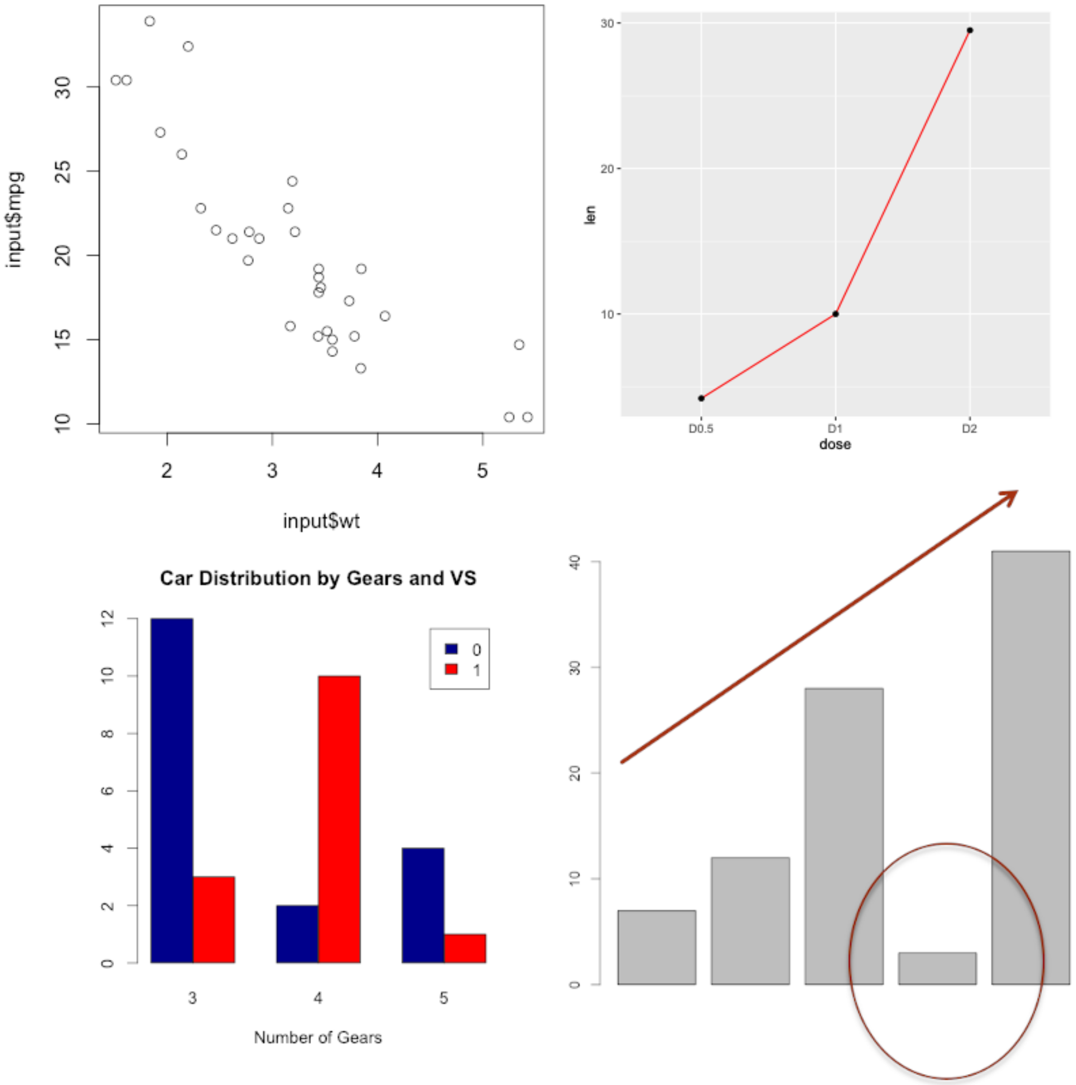


Figure 14.11: Analysis and pattern-reveal through visualization.

At this point, we will only list some simple visualization methods that are often (but not always) used to reveal patterns:

- **scatter plots** are probably best suited for two numeric variables;
- **line charts**, for numeric variable and ordinal variable;
- **bar charts** for one categorical and one numeric, or multiple categorical/nested categorical data,
- **boxplots**, **histograms**, **bubble charts**, **small multiples**, etc.

An in-depth discussion of data visualization is given in Chapter 16 (*Data Visualization*); best practices and a more complete catalogue are provided in [25].

14.5.4 Common Statistical Procedures in R

The underlying goal of **statistical analysis** is to reach an **understanding of the data**. In this section, we show how some of the most common **basic** statistical concepts that can help analysts reach that goal are implemented in R; a more thorough treatment of probability and statistics notions can be found in Chapters 6 (*Probability and Applications*), 7 (*Introduction to Statistical Analysis*), and 8 (*Classical Regression Analysis*).

Once the data is properly organized and visual exploration has begun in earnest, the typical next step is to describe the distribution of each variable numerically, followed by an exploration of the relationships among selected variables.

The objective is to answer questions such as:

- What kind of mileage are cars getting these days? Specifically, what's the distribution of miles per gallon (mean, standard deviation, median, range, and so on) in a survey of automobile makes and models?
- After a new drug trial, what is the outcome (no improvement, some improvement, marked improvement) for drug versus placebo groups? Does the sex of the participants have an impact on the outcome?
- What is the correlation between income and life expectancy? Is it significantly different from zero?
- Are you more likely to receive imprisonment for a crime in different regions of Canada? Are the differences between regions statistically significant?

Basic Statistics

When it comes to calculating **descriptive statistics**, `R` can basically do it all.

We start with functions that are included in the base installation. We will then look for extensions that are available through the use of user-contributed packages.

For illustrative purposes, we will use several of the variables from the *Motor Trend Car Road Tests* (`mtcars`) dataset provided in the base installation: we will focus on miles per gallon (`mpg`), horsepower (`hp`), and weight (`wt`):

```
myvars <- c("mpg", "hp", "wt")
head(mtcars[myvars])
```

	mpg	**hp**	**wt**
Mazda RX4	21.0	110	2.620
Mazda RX4 Wag	21.0	110	2.875
Datsun 710	22.8	93	2.320
Hornet 4 Drive	21.4	110	3.215
Hornet Sportabout	18.7	175	3.440
Valiant	18.1	105	3.460

Let us first take a look at descriptive statistics for all 32 models. In the base installation, we can use the `summary()` function.

```
summary(mtcars[myvars])
```

mpg	hp	wt
Min.: 10.40	Min.: 52.0	Min.: 1.513
1st Qu.: 15.43	1st Qu.: 96.5	1st Qu.: 2.581
Median: 19.20	Median: 123.0	Median: 3.325
Mean: 20.09	Mean: 146.7	Mean: 3.217
3rd Qu.: 22.80	3rd Qu.: 180.0	3rd Qu.: 3.610
Max.: 33.90	Max.: 335.0	Max.: 5.424

The `summary()` function provides the minimum, maximum, quartiles, and mean for numerical variables, and the respective frequencies for factors and logical vectors.

In base `R`, the functions `apply()` or `sapply()` can be used to provide any descriptive statistics. The format in use is:

```
sapply(x, FUN, options)
```

where x is the data frame and `FUN` is an arbitrary function. If options are present, they are passed to `FUN`. Typical functions that can be used include:

- `mean()`
- `sd()`
- `var()`
- `min()`
- `max()`
- `median()`
- `length()`
- `range()`
- `quantile()`
- `fivenum()`

The next example provides several descriptive statistics using `sapply()`, including the **skew** and the **kurtosis**.

```
mystats <- function(x, na.omit=FALSE){
                    if (na.omit)
                        x <- x[!is.na(x)]
                    m <- mean(x)
                    n <- length(x)
                    s <- sd(x)
                    skew <- sum((x-m)^3/s^3)/n
                    kurt <- sum((x-m)^4/s^4)/n - 3
                    return(c(n=n, mean=m, stdev=s,
                             skew=skew, kurtosis=kurt))
                  }
```

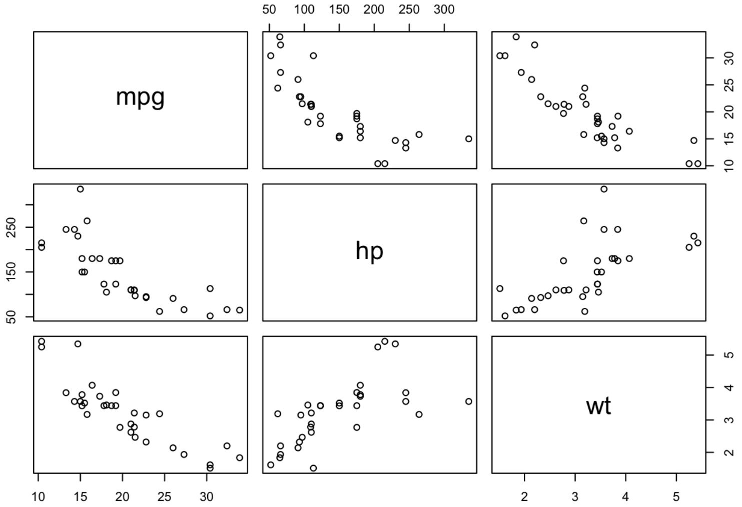
Let us apply `mystats()` to the data frame of interst.

```
sapply(mtcars[myvars], mystats)
```

	mpg	hp	wt
n	32	32	32
mean	20.090625	146.6875	3.21725
stdev	6.026948	68.5628685	0.9784574
skew	0.610655	0.7260237	0.4231465
kurtosis	−0.372766	−0.1355511	−0.0227108

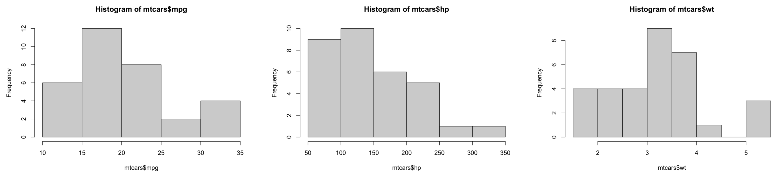
We can plot the pairwise scatterplots for the three variables.

```
plot(mtcars[myvars])
```



For cars in this sample, the mean mpg is 20.1, with a standard deviation of 6.0. The distribution is skewed to the right (+0.61) and is somewhat flatter than a normal distribution (−0.37). This is most evident if we build histograms of the data.

```
hist(mtcars$mpg)
hist(mtcars$hp)
hist(mtcars$wt)
```



To omit missing values for the computations, we would use the option na.omit=TRUE.

Since there are no missing observations in the dataset, we create a version of mtcars with some missing values, then we provide a `mystats()` summary.

Adding missing values

```
my.mtcars <- mtcars
my.mtcars[2,1] <- NA
my.mtcars[17,1] <- NA
sapply(my.mtcars[myvars], mystats, na.omit=TRUE)
```

	mpg	**hp**	**wt**
n	30	32	32
mean	20.24	146.6875	3.21725
stdev	6.1461847	68.5628685	0.9784574
skew	0.5660728	0.7260237	0.4231465
kurt	−0.4870340	−0.1355511	−0.0227108

Notice the changes in the `mpg` summary.

The same table can be obtained using the `dplyr` package functions instead (`skewness()` and `kurtosis()` are available in `e1071` package).

```
mpg = dplyr::summarise(mtcars, n=n(), mean=mean(mpg),
                stdev=sd(mpg), skew=e1071::skewness(mpg),
                kurt=e1071::kurtosis(mpg))
hp = dplyr::summarise(mtcars, n=n(), mean=mean(hp),
                stdev=sd(hp), skew=e1071::skewness(hp),
                kurt=e1071::kurtosis(hp))
wt = dplyr::summarise(mtcars, n=n(), mean=mean(wt),
                stdev=sd(wt), skew=e1071::skewness(wt),
                kurt=e1071::kurtosis(wt))

pivot = t(rbind(mpg,hp,wt))
colnames(pivot) <- c("mpg","hp","wt")
```

	mpg	**hp**	**wt**
n	32	32	32
mean	20.090625	146.6875	3.21725
stdev	6.026948	68.5628685	0.9784574
skew	0.610655	0.7260237	0.4231465
kurt	−0.372766	−0.1355511	−0.0227108

`Hmisc` and `pastecs`

Several packages offer functions for descriptive statistics, including `Hmisc` and `pastecs` (as do `dplyr` and `e1071`).

`Hmisc`'s `describe()` function returns the number of variables and observations, the number of missing and unique values, the mean, quantiles, and the five highest and lowest values.

```
Hmisc::describe(mtcars[myvars])
```

```
mtcars[myvars]

 3  Variables      32  Observations
--------------------------------------------------------------------------------
mpg
       n  missing distinct     Info     Mean      Gmd      .05      .10
      32        0       25    0.999    20.09    6.796    12.00    14.34
     .25      .50      .75      .90      .95
   15.43    19.20    22.80    30.09    31.30

lowest : 10.4 13.3 14.3 14.7 15.0, highest: 26.0 27.3 30.4 32.4 33.9
--------------------------------------------------------------------------------
hp
       n  missing distinct     Info     Mean      Gmd      .05      .10
      32        0       22    0.997    146.7    77.04    63.65    66.00
     .25      .50      .75      .90      .95
   96.50   123.00   180.00   243.50   253.55

lowest :  52  62  65  66  91, highest: 215 230 245 264 335
--------------------------------------------------------------------------------
wt
       n  missing distinct     Info     Mean      Gmd      .05      .10
      32        0       29    0.999    3.217    1.089    1.736    1.956
     .25      .50      .75      .90      .95
   2.581    3.325    3.610    4.048    5.293

lowest : 1.513 1.615 1.835 1.935 2.140, highest: 3.845 4.070 5.250 5.345 5.424
--------------------------------------------------------------------------------
```

The `pastecs` package includes the function `stat.desc()` that provides a wide range of descriptive statistics:

```
stat.desc(x, basic=TRUE, desc=TRUE, norm=FALSE, p=0.95)
```

where x is a data frame or a time series.

If `basic=TRUE` (the default), the number of values, null values, missing values, minimum, maximum, range, and sum are provided. If `desc=TRUE` (also the default), the median, mean, standard error of the mean, 95% confidence interval for the mean, variance, standard deviation, and coefficient of variation are also provided. Finally, if `norm=TRUE` (not the default), normal distribution statistics are returned, including skewness and kurtosis (with statistical significance) and the Shapiro–Wilk test of normality.

The p option is used to calculate the confidence interval for the mean (.95 by default).

For instance, we may obtain:

```
pastecs::stat.desc(mtcars[myvars])
```

	mpg	**hp**	**wt**
nbr.val	32	32	32
nbr.null	0	0	0
nbr.na	0	0	0
min	10.4	52	1.513
max	33.9	335	5.424
range	23.5	283	3.911
sum	642.9	4694	102.952
median	19.2	123	3.325
mean	20.0906250	146.6875000	3.2172500
SE.mean	1.0654240	12.1203173	0.1729685
CI.mean.0.95	2.1729465	24.7195501	0.3527715
var	36.3241028	4700.8669355	0.9573790
std.dev	6.0269481	68.5628685	0.9784574
coef.var	0.2999881	0.4674077	0.3041285

We take this opportunity to caution users against relying too heavily on a single (or even a few) specific packages.

Correlations

Correlation coefficients are used to describe relationships among quantitative variables. The sign ± indicates the direction of the relationship (positive or inverse), and the magnitude indicates the strength of the relationship (0: no linear relationship; 1: perfect linear relationship).

In this section, we look at a variety of correlation coefficients, as well as tests of significance. We will use the `state.x77` dataset available in the base R installation. It provides data on the population, income, illiteracy rate, life expectancy, murder rate, and high school graduation rate for the 50 US states in 1977. There are also temperature and land-area measures, but we will not be using them.[22]

`R` can produce a variety of correlation coefficients, including:

- **Pearson's product-moment** coefficient (degree of linear relationship between two quantitative variables);
- **Spearman's rank-order** coefficient (degree of relationship between two rank-ordered variables), and
- **Kendall's tau** coefficient (nonparametric measure of rank correlation).

The `cor()` function produces all three correlation coefficients, whereas the `cov()` function provides covariances. There are many options, but a simplified format for producing correlations is

```
cor(x, use=OPT , method=METHOD)
```

where x is a matrix or a data frame, and `use` specifies the handling of missing data; its options are

- `all.obs` (assumes no missing data);

[22] 22: In addition to the base installation, we will also be using the `psych` and `ggm` packages.

- `everything` (any correlation involving a case with missing values will be set to missing);
- `complete.obs` (listwise deletion), and
- `pairwise.complete.obs` (pairwise deletion).

The `method` specifies the type of correlation; its options are `pearson`, `spearman`, and `kendall`.

The default options are `use ="everything"` and `method= "pearson"`.

For the built-in dataset `state.x77`, which contains socio-demographic information about the 50 U.S. states from 1977, we find the following correlations:

Correlations in the state.x77 data

```
states <- state.x77[,1:6]
cor(states)
```

	Population	**Income**	**Illiteracy**	**Life Exp**	**Murder**	**HS Grad**
Population	1.0000000	0.2082276	0.1076224	−0.0680520	0.3436428	−0.0984897
Income	0.2082276	1.0000000	−0.4370752	0.3402553	−0.2300776	0.6199323
Illiteracy	0.1076224	−0.4370752	1.0000000	−0.5884779	0.7029752	−0.6571886
Life Exp	−0.0680520	0.3402553	−0.5884779	1.0000000	−0.7808458	0.5822162
Murder	0.3436428	−0.2300776	0.7029752	−0.7808458	1.0000000	−0.4879710
HS Grad	−0.0984897	0.6199323	−0.6571886	0.5822162	−0.4879710	1.0000000

This produces the Pearson product-moment correlation coefficients. We can see, for example, that a strong positive correlation exists between `income` and `HS Grad` rate and that a strong negative correlation exists between `Illiteracy` and `Life Exp`.

A **partial correlation** is a correlation between two quantitative variables, controlling for one or more other quantitative variables;[23] the `pcor()` function in the `ggm` package provides partial correlation coefficients (this package is not installed by default, so it must be installed before first use).

The format is:

```
pcor(u, S)
```

where `u` is a vector of integers, with the

- first two entries representing the indices of the variables to be correlated, and
- remaining numbers being the indices of the conditioning variables (that is, the variables being **partialled out**),

and where `S` is the covariance matrix among the variables.

Partial correlations in the state.x77 data I

```
colnames(states)
```

[23] The use of partial correlations is common in the social sciences, but not so much in other fields.

```
ggm::pcor(c(1,5,2,3,6), cov(states))
```

```
"Population" "Income" "Illiteracy" "Life Exp" "Murder" "HS Grad"
0.3462724
```

In this case, 0.346 is the correlation between `Population` (variable 1) and the `Murder` rate (variable 5), controlling for the influence of `Income`, `Illiteracy`, and `HS Grad` (variables 2, 3, and 6 respectively).

We see that the partial correlations only change slightly if we condition against a different subset of values.

Partial correlations in the state.x77 data II

```
ggm::pcor(c(1,5,2,3), cov(states))
ggm::pcor(c(1,5,2), cov(states))
```

```
0.3621683
0.4113621
```

How do these three values compare to the direct correlation between `Population` and `Murder`?

Simple Linear Regression

In many ways, **regression analysis** is at the heart of statistics. It is a broad term for a set of methodologies used to predict a response variable (also called a dependent, criterion, or outcome variable) from one or more predictor variables (also called independent or explanatory variables).

In general, regression analysis can be used to:

- identify the explanatory variables that are related to a response variable;
- describe the form of the relationships involved, and
- provide an equation for predicting the response variable from the explanatory variables.

For example, an exercise physiologist might use regression analysis to develop an equation for predicting the expected number of calories a person will burn while exercising on a treadmill.

In this example, the response variable is the number of calories burned (calculated from the amount of oxygen consumed), say, and the predictor variables might include:

- duration of exercise (minutes);
- percentage of time spent at their target heart rate;
- average speed (mph);
- age (years);
- gender, and
- body mass index (BMI).

From a practical point of view, regression analysis could help answer questions such as:

- how many calories can a 30-year-old man with a BMI of 28.7 expect to burn if he walks for 45 minutes at an average speed of 4 miles per hour and stays within his target heart rate 80% of the time?
- what's the minimum number of variables needed in order to accurately predict the number of calories a person will burn when walking?

`R` has powerful and comprehensive features for fitting regression models – the abundance of options can be confusing. The basic function for fitting a linear model is `lm()`. The format is

```
fit <- lm(formula, data)
```

where `formula` describes the model to fit and `data` is the data frame containing the data that is used in fitting the model. The resulting object (`fit`, in this case) is a list that contains extensive information about the fitted model.

The formula is typically written as

```
Y ~ X1 + X2 + ... + Xk
```

where ~ separates the response variable on the left from the predictor variables on the right, and the predictor variables are separated by + symbols.

In addition to `lm()`, there are several functions that are useful when generating regression models. Each of these functions is applied to the object returned by `lm()` in order to generate additional information based on the fitted model.

Function	**Action**
`summary()`	Displays detailed results for the fitted model
`coefficients()`	Lists the model parameters (intercept and slopes) for the fitted model
`confint()`	Provides confidence intervals for the model parameters (95% by default)
`residuals()`	Lists the residual values in a fitted model
`anova()`	Generates an ANOVA table for a fitted model, or to compare 2+ fitted models
`plot()`	Generates diagnostic plots for evaluating the fit of a model
`fitted()`	Extracts the fitted values for the dataset
`predict()`	Uses a fitted model to predict response values for a new dataset

As an example, the `women` dataset in `R`'s base installation provides the heights and weights for a set of 15 women aged 30 to 39. Assume that we are interested in predicting the weight of an individual from their height.[24]

[24] An equation for predicting weight from height could help identifying individuals who are possibly overweight (or underweight), say.

The linear regression on the data is obtained as follows:

Regression on the women dataset

```
fit <- lm(weight ~ height, data=women)
summary(fit)
```

```
Call:
lm(formula = weight ~ height, data = women)

Residuals:
    Min      1Q  Median      3Q     Max
-1.7333 -1.1333 -0.3833  0.7417  3.1167

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -87.51667    5.93694  -14.74 1.71e-09 ***
height        3.45000    0.09114   37.85 1.09e-14 ***
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Residual standard error: 1.525 on 13 degrees of freedom
Multiple R-squared:  0.991, Adjusted R-squared:  0.9903
F-statistic:  1433 on 1 and 13 DF,  p-value: 1.091e-14
```

From the output, you see that the prediction equation is

$$\widehat{\texttt{weight}} = -87.52 + 3.45 \times \texttt{height}.$$

Because a height of 0 is impossible, there is no sense in trying to give a physical interpretation to the intercept – it merely becomes an adjustment constant (in other words, 0 is **not in the domain** of the model).

From the `P(>|t|)` column, we see that the regression coefficient (3.45) is significantly different from zero ($p < 0.001$), which indicates that there’s an expected increase of 3.45 pounds of weight for every 1 inch increase in height. The multiple R-squared coefficient (0.991) indicates that the model accounts for 99.1% of the variance in weights.

The individual weights (in pound) are:

```
women$weight
```

```
115 117 120 123 126 129 132 135 139 142 146 150 154 159 164
```

and their fitted values (and residuals) are

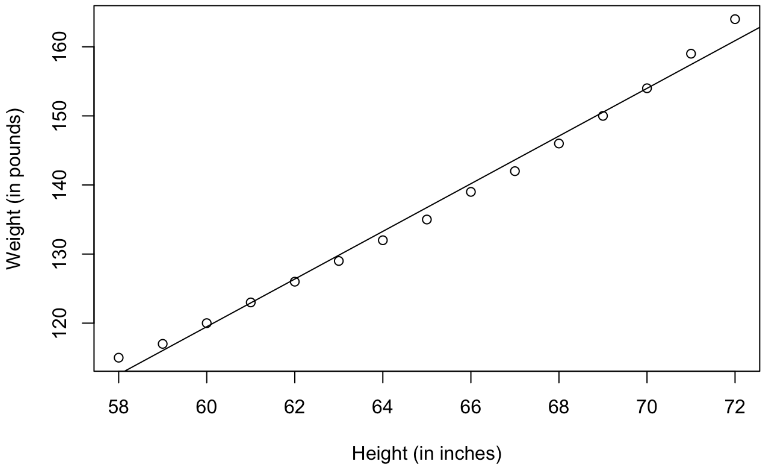
```
fitted(fit)
residuals(fit)
```

```
fitted:
       1        2        3        4        5        6        7        8 
112.5833 116.0333 119.4833 122.9333 126.3833 129.8333 133.2833 136.7333 
       9       10       11       12       13       14       15 
140.1833 143.6333 147.0833 150.5333 153.9833 157.4333 160.8833 

residuals:
      1       2       3       4       5       6       7       8 
 2.4167  0.9667  0.5167  0.0667 -0.3833 -0.8333 -1.2833 -1.7333 
      9      10      11      12      13      14      15 
-1.1833 -1.6333 -1.0833 -0.5333  0.0167  1.5667  3.1167 
```

We can see that the linear fit is decent (although the residual structure suggests that a quadratic fit would probably be better).

```
plot(women$height,women$weight,
     xlab="Height (in inches)", ylab="Weight (in lbs)")
abline(fit)
```



Bootstrapping

Bootstrapping is a powerful and elegant approach to estimating the sampling distribution of specific statistics. It can be implemented in many situations where asymptotic results are difficult to find or otherwise unsatisfactory.

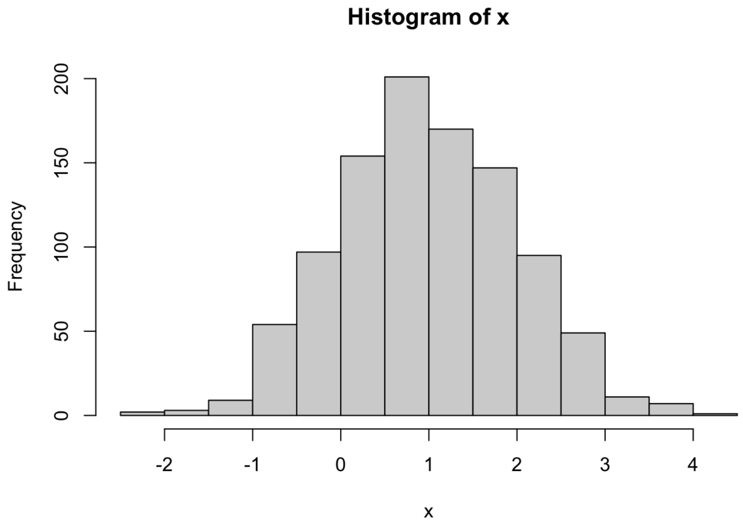
Bootstrapping proceeds using three steps:

1. resample the dataset (with replacement) many times over (typically on the order of 10,000);
2. calculate the desired statistic from each resampled dataset,
3. use the distribution of the resampled statistics to estimate the standard error of the statistic (normal approximation method) or construct a confidence interval using quantiles of that distribution (percentile method).

There are several ways to bootstrap in R. As an example, say that we want to estimate the standard error and 95% confidence interval for the **coefficient of variation** (CV), defined as σ/μ, for a random variable X.

We will illustrate the procedure with 1000 generated values of $X \sim \mathcal{N}(1,1)$:

```
set.seed(0)              # for replicability
x = rnorm(1000, mean=1, sd=1)
hist(x)
```



On this sample, the coefficient of variation is:

```
(cv=sd(x)/mean(x))
```

```
1.014057
```

We must define a function to compute the statistic of interest in R.

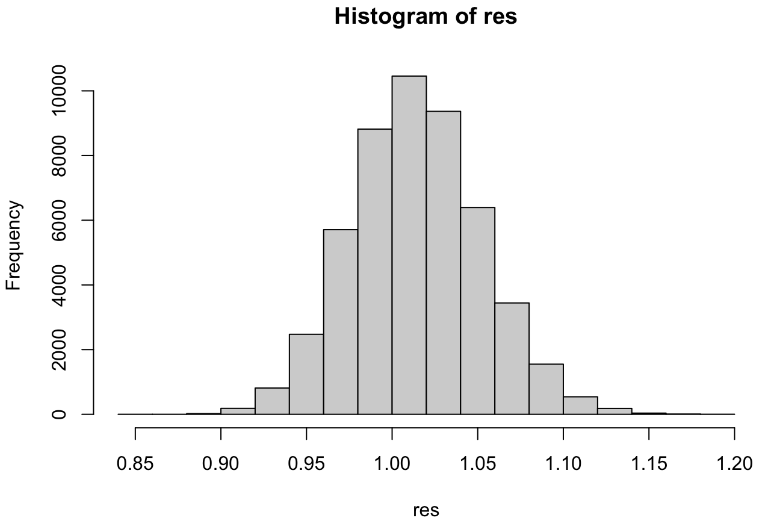
Defining the coefficient of variation in R

```
cvfun = function(x) {
    return(sd(x)/mean(x))
}
```

The `replicate()` function is the base R tool for repeating function calls. We nest a call to `cvfun()` and a call to sample the data with replacement using the `sample()` function (with 50000 replicates).

Bootstrap distribution of CV(x)

```
res = replicate(50000, cvfun(sample(x, replace=TRUE)))
hist(res)
```



We can also compute quantiles, as below:

95% Confidence Interval for CV(x)

```
quantile(res, c(.025, .975))
```

```
     2.5%     97.5%
0.9432266 1.0917185
```

This seems reasonable, as we would expect the CVs to be centered around 1, given that $\mu = \sigma = 1$ (in this example).

The percentile interval is easy to calculate from the observed bootstrapped statistics. If the distribution of the bootstrap samples is approximately normally distributed, a $t-$interval could be created by calculating the standard deviation of the bootstrap samples and finding the appropriate multiplier for the confidence interval. Plotting the bootstrap sample estimates is helpful to determine the form of the bootstrap distribution.

The framework can also be extended to include **non-linear** models, **correlated variables**, **probability estimation**, and/or **multivariate** models; any book on statistical analysis contains at least one chapter or two on the topic (see [41, 174], for instance).

We will not pursue the topic further except to say that regression analysis and bootstrapping are two of the arrows that every data scientist should have in their quiver.

14.5.5 Quantitative Methods

We provided a list of quantitative methods in Section 14.4.2 (*Data Collection, Storage, Processing, and Modeling*); we finish this section by expanding on a few of them.

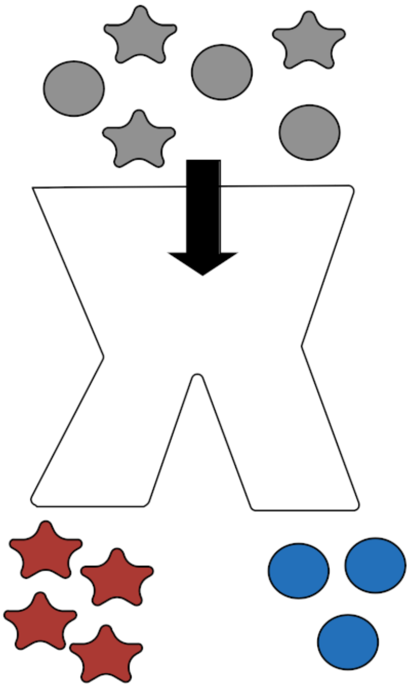


Figure 14.12: The trousers of classification.

Classification and Supervised Learning Tasks

Classification is one of the cornerstones of machine learning. Instead of trying to predict the numerical value of a response variable (as in regression), a **classifier** uses **historical data**[25] to identify general patterns that could lead to observations belonging to one of several **pre-defined categories**.

For instance, if a car insurance company only has resources to investigate up to 20% of all filed claims, it could be useful for them to predict:

- whether a claim is likely to be fraudulent;
- whether a customer is likely to commit fraud in the near future;
- whether an application for a policy is likely to result in a fraudulent claim,
- the amount by which a claim will be reduced if it is fraudulent, etc.

Analysts and machine learning practitioners use a variety of different techniques to carry this process out (see Figure 14.12 for an illustration, and Chapters 19 (*Introduction to Machine Learning*) and 21 (*Focus on Classification*), as well as [2, 5, 6] for more details), but the general steps always remain the same:

1. use **training data** to teach the classifier;
2. test/validate the classifier using **hold-out** data,
3. if it passes the test, use the classifier to classify **novel instances**.

Some classifiers (such as deep learning neural nets) are '**black boxes**': they might be very good at classification, but they are not **explainable**. In some instances, that is an acceptable side effect of the process, in others, it might not be – if an individual is refused refugee status, say, they might rightly want to know **why**.

[25] This training data usually consists of a **randomly** selected subset of the **labeled** (response) data.

Unsupervised Learning Techniques

The hope of artificial intelligence is that intelligent behaviours will eventually be able to be **automated**. For the time being, however, that is still very much a work in progress.[26]

Classification, for instance, is the prototypical supervised task: can we learn from historical/training examples? It seems like a decent approach to learning: evidence should drive the process.

But there are limitations to such an approach: it is difficult to make a **conceptual leap** solely on the basis of training data [if our experience in learning is anything to go by. . .], if only because the training data might not be representative of the system, or because the learner target task is **too narrow**.

In **unsupervised** learning, we learn without examples, based solely on what is found in the data. There is no specific question to answer (in the classification sense), other than "what can we learn from the data?"

Typical unsupervised learning tasks include:

- **clustering** (finding novel categories);
- **association rules mining**,
- **recommender systems**, etc.

For instance, an online bookstore might want to make recommendations to customers concerning additional items to browse (and hopefully purchase) based on their buying patterns in prior transactions, the similarity between books, and the similarity between **customer segments**:

- but what are those patterns?
- how do we measure similarity?
- what are the customer segments?
- can any of that information be used to create promotional bundles?
- etc.

The lack of a specific target makes unsupervised learning much more **difficult** than supervised learning, as does the challenges of **validating the results**.[27]

More general information and details on clustering can be found in Chapters 19 (*Introduction to Machine Learning*) and 22 (*Focus on Clustering*), as well as in [4, 5, 175].

Other Machine Learning Tasks

These scratch but a **minuscule** part of the machine learning ecosystem. Other common tasks include [168]:

- profiling and behaviour description;
- link prediction;
- data reduction,
- influence/causal modeling, etc.

to say nothing of more sophisticated learning frameworks (semi-supervised learning, reinforcement learning [176], deep learning [177], etc.).

[26] One of the challenges in that process is that not every intelligent behaviour arises from a supervised process.

[27] This contributes to the proliferation of clustering algorithms and cluster quality metrics.

Time Series Analysis and Process Monitoring

Processes are often subject to **variability**:

- variability due the **cumulative effect** of many small, essentially unavoidable causes, such as regular variations in the weather or in the quality of materials (a process that only operates with such **common causes** is said to be **in (statistical) control**),
- variability due to **special causes**, such as improperly adjusted machines, poorly trained operators, defective materials, etc. (the variability is typically much larger for special causes, and such processes are said to be **out of (statistical) control**).

The aim of **statistical process monitoring** (SPM) is to identify occurrence of special causes. This is often done *via* **time series analysis**.

Consider n observations $\{x_1, \dots, x_n\}$ arising from some collection of processes. In practice, the index i is often a **time index** or a **location index**, i.e., the x_i are observed in **sequence** or in **regions**.[28]

The processes that generate the observations could change from one time/location to the next due to:

- **external factors** (war, pandemic, regime change, election results, etc.), or
- **internal factors** (policy changes, modification of manufacturing process, etc.).

In such case, the mean and standard deviation alone might not provide a useful summary of the situation.

To get a sense of what is going on with the data (and the associated system), it could prove preferable to **plot the data** in the **order that it has been collected** (or according to geographical regions, or both).

The horizontal coordinate would then represent:

- the **time of collection** t (order, day, week, quarter, year, etc.), or
- the **location** i (country, province, city, branch, etc.).

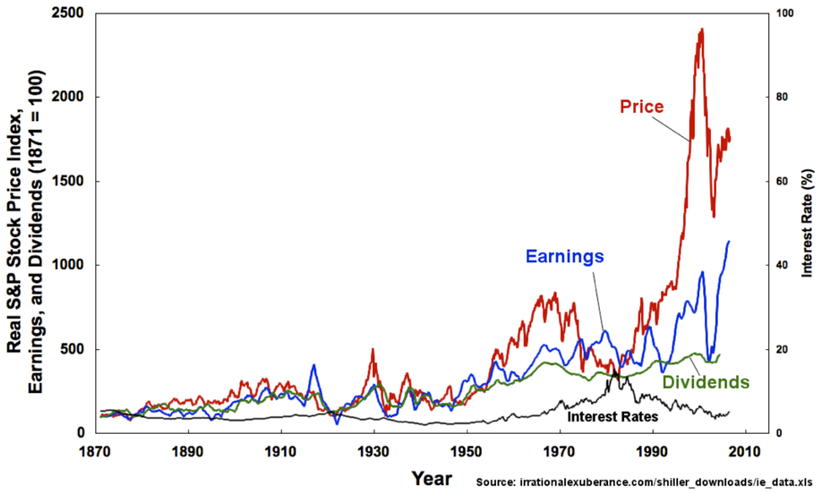
The vertical coordinate represents the observations of interest x_t or x_i (see Figure 14.13 for an example). In process monitoring terms, we may be able to identify potential special causes by identifying **trend breaks**, **cycles discontinuities**, or **level shifts** in time series.

For instance, consider the three time series of Figure 14.14.

Is any action required?

- in the first example (left), there are occasional drops in sales from one year to the next, but the **upward trend** is clear – we see the importance of considering the full time series; if only the last two points are presented to stockholders, say, they might conclude that action is needed, whereas the whole series paints a more positive outlook;
- in the second case (middle), there is a **cyclic effect** with increases from Q1 to Q2 and from Q2 to Q3, but decreases from Q3 to Q4 and from Q4 to Q1. Overall, we also see an upward trend – the presence of regular patterns is a positive development,

[28] In the first situation, the observations form a **time series**.



Source: irrationalexuberance.com/shiller_downloads/ie_data.xls

Figure 14.13: Real S&P stock price index (red), earnings (blue), and dividends (green), together with interest rates (black), from 1871 to 2009 [R.J. Shiller].

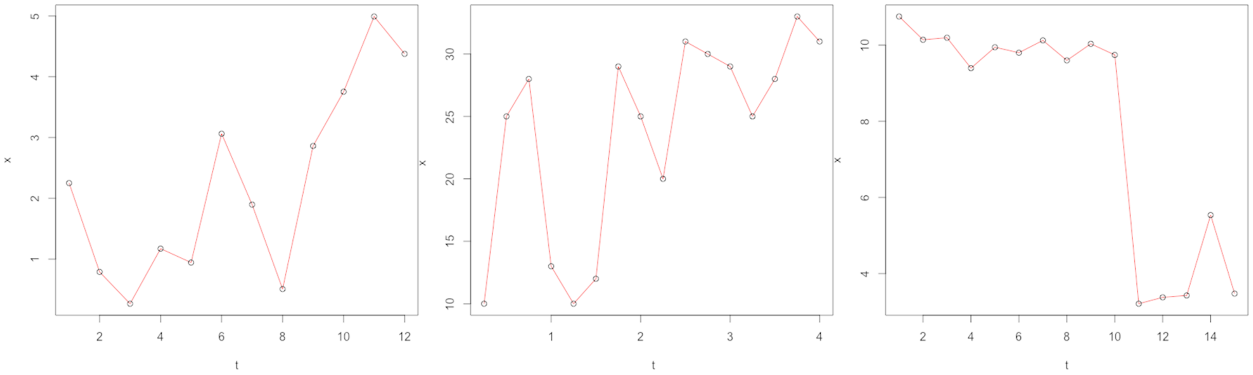


Figure 14.14: Sales (in $10,000's) for 3 different products – years (left), quarters (middle, but labeled in years), weeks (right).

- finally, in the last example (right), something clearly happened after the tenth week, causing a **trend level shift**. Whether it is due to internal or external factors depends on the context, which we do not have at our disposal, but some action certainly seems to be needed.

We might also be interested in using historical data to **forecast** the future behaviour of the variable. These are the familiar analysis goals:

- **finding patterns** in the data, and
- **creating a (mathematical) model** that captures the essence of these patterns.

Time series patterns can be quite complex and must often be **broken down** into multiple component models (trend, seasonal, irregular, etc.).

Typically, this can be achieved with fancy analysis methods, but it is not a simple topic, in general (some details can be found in Chapter **??**, *Time Series and Forecasting*). Thankfully, there are software libraries that can help.

Anomaly Detection

The special points from process monitoring are anomalous in the sense that something unexpected happens there, something that changes the nature of the data pre- and post-break.

In a more general context, **anomalous observations** are those that are **atypical** or **unlikely**.

From an analytical perspective, anomaly detection can be approached using supervised, unsupervised, or conventional statistical methods.

The discipline is rich and vibrant (and the search for anomalies can end up being an arms race against the "bad guys"), but it is definitely one for which analysts should heed contextual understanding – blind analysis leads to blind alleys!

A more thorough treatment is provided in Chapter 27 (*Anomaly Detection and Outlier Analysis*).

14.5.6 Quantitative Fallacies

Quantitative fallacies and **misinterpretations** are a consequence of our notoriously poor skills at quantitative interpretation, which manifest themselves through incorrect reasoning or the misuse of statistics (by design or by accident).

- **Correlation is not causation:** causality is one kind of correlation but correlation is not necessarily causal – it's conceivable that a man who purchases diapers also decides to buy beer, but the purchase of diapers does not cause the purchase of beer. The statement is sometimes extended to imply that while correlation is not causation, it can be highly suggestive.
- **Odd results sometimes happen:** the patterns in subgroups of the data may not align with pattern in the entire dataset, thanks to Simpson's paradox (see Figure 14.15).

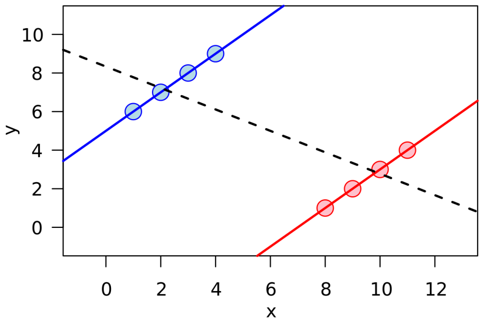


Figure 14.15: Simpson's paradox: the slope of the line of best fit in each of the two subgroups (blue and red) is positive, while the slope of the line of best fit for the entire dataset (black) is negative [author unknown].

- **Extreme patterns can mislead:** rare patterns need to be considered separately from the rest of the data. They are either invalid patterns and need to be removed altogether, or interesting and they could reveal that the story has more depth. The presence of extreme patterns or cases in the modeling process could introduce biases to the model and the final model is less likely to be a good fit for the data. For instance, a severe snowstorm hit Ottawa on Feb 16, 2016, causing a large number of road collisions – if we want to predict the number of road collisions on an average day in Ottawa per year, keeping this day in the model may skew the results.
- **Leaving a study's range:** a fallacy can occur when an assumption is replaced by a seemingly similar one which turns out not to be interchangeable. For instance, when a snow storm drops 20cm of

snow in Ottawa, traffic may be delayed slightly, but it's business as usual for most citizens; we might expect a similar reaction in Winnipeg, but the same 20cm would paralyze Beijing and block sewers. The effects of a snow storm may not be transferable.

- **Keeping the base rate in mind:** the base rate fallacy occurs when the underlying characteristics of a subgroup are ignored. As an example, the likelihood that an individual will die of lung cancer depends on whether he or she is a smoker; if this information is not known, the prediction will also depend on the likelihood of the individual being a smoker. This fallacy is best avoided through the application of Bayesian analysis.
- **Randomness plays a role:** if a situation has occurred more frequently in the past, it is possible that it will be more likely to happen again in the future, but it is also possible that it happened more frequently in the past by chance alone. Statistical analysis will help to separate the Gambler's fallacy from the presence of a signal in the data.
- **There is a human component to any analytical activity:** it is impossible to avoid human bias altogether when analyzing data. The ultimate choice of explanatory parameters or of the final model (to name but these two) can never be done with complete and total detachment.
- **Small effects can be significant:** a statistically significant result does not need to be large, it just needs to be unlikely to be due to chance alone. The terminology is partly to blame for the confusion: in the statistical context, **significance** is not the same as **importance**.
- **Misinterpretation of $p-$values:** the $p-$value reveals the probability of observing a result given that the null hypothesis is true, rather than the probability that the null hypothesis is true. As an example, suppose a Department wants to find out whether a reported increase in efficiency is due to the implementation of a new policy; the null hypothesis would be that the new policy has no effect on efficiency. Using available data, the model produces a $p-$value of 0.05; we cannot conclude that there is a 95% probability that the null hypothesis is false and that the policy change had an effect on efficiency. We can only conclude that there is a 5% probability that our model would show an effect even if the none was present.

There is a lot more to say on the topic of data analysis – we will delve into various topics in detail in subsequent chapters.

14.6 Exercises

1. Are the following examples of good questions? Are they vague or specific? What are the ranges of answers we could expect? How would you improve them?

   a. How does rain affect goal percentage at a soccer match?
   b. Did the Toronto Maple Leafs beat the Edmonton Oilers?
   c. Did you like watching the Tokyo Olympics?

   d. What types of recovery drinks do hockey players drink?
   e. How many medals will Canada win at the Paris 2024 Olympics?
   f. Should we fund the Canadian Basketball team more than the Canadian Hockey team?

2. Write a paper discussing some of the ethical issues surrounding the use of artificial intelligence (A.I.), data science (D.S.), and/or machine learning (M.L.) algorithms in the public sector, the private sector, or in academia.

   a. Establish a list of the 3 most important ethical principles that the use of such algorithms should abide by. Explain why you have selected each of these principles.
   b. Describe (at least) 2 instances of the use of A.I./D.S./M.L. in the public sector, the private sector, or in academia, when the ethical principles you have chosen were violated. Discuss how the failure to abide by your selected ethical principles have caused (or could cause) harm to individuals, organizations, countries, etc.
   c. Suggest how the projects discussed above could have been modified so that their use of A.I./D.S./M.L. algorithms would abide by your selected ethical principles.

3. Provide additional data summaries and some simple visual summaries of the artificial dataset of pages 693-694.
4. Select a data project of interest to you (either personally or professionally) and provide a first planning draft for it, touching on the topics discussed in this module and in Chapter 13 (*Non-Technical Aspects of Quantitative and Data Work*). The following questions can help guide your proposal:

   - What are some questions associated with the project?
   - What is the conceptual model of the underlying situation?
   - What kind of dataset(s) exist that could help you answer these questions?
   - Are there data or analytical limitations?
   - Do you need to collect new data to handle such questions?
   - How is the data stored/accessed? What are the infrastructure requirements?
   - What do deliverables look like?
   - How would successes be quantified/qualified?
   - What are your timelines and availability?
   - What skillsets are required to work on this project?
   - Would you work on this alone or as part of a team?
   - How costly would it be to initiate and complete this project?
   - What does the data analysis pipeline look like?
   - What software and analytical methods will be used?

5. The file `cities.txt` ↗ contains population information about a country's cities. A city is classified as "small" if its population is below 75K, as "medium" if it falls between 75K and 1M, and as "large" otherwise.

   - Locate and load the file into the workspace of your choice. How many cities are there? How many in each group?
   - Display summary population statistics for the cities, both overall and by group.

6. Find examples of recent "Data in the News" stories. Were they successes or failures? What social consequences could emerge from the technologies described in the stories?
7. In what format is your organization's data available? Are you able to access it easily? Is it updated regularly? Are there data dictionaries? Have you read them?
8. Consider the following situation: you are away on business and you forgot to hand in a very important (and urgently required) architectural drawing to your supervisor before leaving. Your office will send an intern to pick it up in your living space. How would you explain to them, by phone, how to find the document? If the intern has previously been in your living space, if their living space is comparable to yours, or if your spouse is at home, the process may be sped up considerably, but with somebody for whom the space is new (or someone with a visual impairment, say), it is easy to see how things could get complicated. Time is of the essence – you and the intern need to get the job done correctly as quickly as possible. What is your strategy?
9. Translate the cognitive biases to analytical contexts. What cognitive biases are you, your team, and your organization most susceptible to? Least?
10. Research the recent data ethics scandals involving Volkswagen, Amazon, Whole Foods Markets, Cambridge Analytica, Ashley Madison, General Motors, or any other organization. What transpired? Who was affected? What were the consequences to the general public, the organization, the data community? How could it have been avoided?
11. Establish a statement of ethics for your data work. Are there areas that you are unwilling to work on?
12. The remaining exercises use the Gapminder Tools ↗ (there is also an offline version ↗).

    a. Take some time to explore the tool. In the online version, the default starting point is a bubble chart of 2020 life expectancy vs. income, per country (with bubble size associated with total population). In the offline version, select the "Bubbles" option.
    b. Can you identify the available variable categories and some of the variables? [You may need to dig around a bit.]
    c. Why do you think that Gapminder has selected Life Expectancy and Income as the default plotting variables?
    d. Replace Life Expectancy by Babies per woman. Observe and discuss the changes from the default plot.
    e. Formulate a few questions that could be answered with the default data.
    f. Formulate a few questions that could be answered using some of the other variables.
    g. At what point in the data science workflow do you think that visualizations of this nature could be useful?
    h. Do these visualizations provide a sound understanding of the system under investigation (the geopolitical Earth)?
    i. What do you think the data sources are for the underlying dataset? [You may need to dig around the internet to answer this question].

j. Are all variables and measurements equally trustworthy? How could you figure this out?

k. Is the underlying dataset structured or unstructured?

l. Provide a potential data model for the dataset.

m. What are the types of the 4 default variables (Life Expectancy, Income, Population, World Regions)?

n. Play around with the charts for a bit. Can you find pairs of variables that are positively correlated? Negatively correlated? Uncorrelated?

o. Among those variables that are correlated, do any seem to you to exhibit a dependent-independent relationship? How could you identify such pairs?

p. Can you provide an eyeball estimate of the mean, the median, and the range of various numerical variables?

q. Can you provide an eyeball estimate of the mode of the categorical variables?

r. Can you identify epochal moments (special temporal points) in the data where a shift occurs, say?

s. Is the tool and its underlying dataset useable? What factors does your answer depend on?