

Max Garzon ·
Ching-Chi Yang · Deepak Venugopal ·
Nirman Kumar · Kalidas Jana · Lih-Yuan Deng
Editors

Dimensionality Reduction in Data Science



Springer

Dimensionality Reduction in Data Science

Max Garzon •
Ching-Chi Yang • Deepak Venugopal •
Nirman Kumar • Kalidas Jana • Lih-Yuan Deng
Editors

Dimensionality Reduction in Data Science



Editors

Max Garzon 
375 Dunn Hall
The University of Memphis
Memphis, TN, USA

Ching-Chi Yang 
373 Dunn Hall
The University of Memphis
Memphis, TN, USA

Deepak Venugopal
375 Dunn Hall
The University of Memphis
Memphis, TN, USA

Nirman Kumar 
375 Dunn Hall
The University of Memphis
Memphis, TN, USA

Kalidas Jana
Memphis, TN, USA

Lih-Yuan Deng 
373 Dunn Hall
The University of Memphis
Memphis, TN, USA

ISBN 978-3-031-05370-2 ISBN 978-3-031-05371-9 (eBook)
<https://doi.org/10.1007/978-3-031-05371-9>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Data science is about solving problems based on observations and data collected in the real world. Problems may range from the mundane to difficult scientific questions, for example, rating movies for recommendation systems, understanding the earning power of American taxpayers, increasing revenue for a business, spam, controlling the spread of misinformation through the internet, global warming, or the expansion of our universe. Our ability to generate, gather, and store volumes of data in the order of tera- and exo-bytes daily has far outpaced our ability to derive useful information from it in many fields, with available computational resources. The overarching goal of this book is to provide a practical and fairly complete, but not encyclopedic, review of Data Science (DS) through the lens of Dimensionality Reduction (DR).

The intended audience consists of professionals and/or students in any domain science who need to solve problems to answer questions about their domain based on data. Domain science is a fairly vague term that refers to a specialized area of human knowledge characterized by specific questions about a certain aspect of reality (like what is motion in *physics*, what are physical objects made of in *chemistry*, what is life in *biology*, and so forth.) In addition to the well-established sciences, they include just about any area where data can be recorded and analyzed to answer questions concerning the population of individuals or objects the data is about.

Data science presents a singular approach to problem solving when compared to more established sciences. Traditional sciences are motivated by understanding our world in order to survive and thrive. That requires a degree of analysis and theorizing to understand the specific phenomena involved and to enable predictive power. By contrast, with the advent of computer science and its abstractions into the information age (as embodied by the internet and web for example), tools have been created that can be used regardless of the specific domain. Once this threshold is crossed, then it is a natural next step from mathematics and statistics to synergistically combine them with the powerful computational tools developed by computer science to create a new science that is more than the sum of the parts, hence data science.

We have strived to leave our niche hats at the door and present an intuitive, integrative and synergistic approach that captures the best of the three worlds. That is the pervasive thread that readers will discover through examples and methods throughout the book. Sections begin with intuitive examples of a problem to be solved by the (perhaps new) concepts and results being described in the section. A professional with an undergraduate degree in any science, particularly quantitative, should be able to easily follow this part. These motivating examples are then followed by precise definitions of the technical concepts and presentation of the results in general situations. These require a degree of abstraction that can be followed by re-interpreting concepts like in the original example(s). Finally, each section closes with solutions to the original problem(s) afforded by these techniques, perhaps in various ways to compare and contrast dis/advantages of the various DR techniques based on quantitative and qualitative assessments back in the real world.

We are grateful to acknowledge support for this project from various sources. First, support from the University of Memphis CoRS (Communities of Research Scholars) program (through Deborah Hernandez) to start up research projects that facilitated initial interactions that eventually led to the interdisciplinary collaboration that produced this book, among other works, as well as access to the High Performance Cluster (HPC) used in development and testing of most results herein. Second, Lih-Yuan acknowledges support from the National Science Foundation under an award for The Learner Data Institute (NSF-1934745: The opinions, findings, and results are solely the authors' and do not reflect those of the funding agency.). Third, Max acknowledges faculty (Professor Luis Fernando Nino) and student support (especially Alfredo Bayuelo for TA support and help with cartoon design and rendering) at the National University of Colombia for their active feedback and project outcomes at the XI International Cathedra on *Data Science for Bioinformatics* in 2017, where several of the approaches, views, and some results included in this book were conceived.

Bon voyage!

Memphis, TN, USA
November 2021

Max Garzon
Ching-Chi Yang
Deepak Venugopal
Nirman Kumar
Kalidas Jana
Lih-Yuan Deng

Contents

1	What Is Data Science (DS)?	1
	Max Garzon, Ching-Chi Yang, and Lih-Yuan Deng	
2	Solutions to Data Science Problems	29
	Deepak Venugopal, Lih-Yuan Deng, and Max Garzon	
3	What Is Dimensionality Reduction (DR)?	67
	Lih-Yuan Deng, Max Garzon, and Nirman Kumar	
4	Conventional Statistical Approaches	79
	Ching-Chi Yang, Max Garzon, and Lih-Yuan Deng	
5	Geometric Approaches	97
	Nirman Kumar	
6	Information-Theoretic Approaches	127
	Max Garzon, Sambriddhi Mainali, and Kalidas Jana	
7	Molecular Computing Approaches	145
	Max Garzon and Sambriddhi Mainali	
8	Statistical Learning Approaches	169
	Ching-Chi Yang and Lih-Yuan Deng	
9	Machine Learning Approaches	179
	Deepak Venugopal and Max Garzon	
10	Metaheuristics of DR Methods	199
	Deepak Venugopal, Max Garzon, Nirman Kumar, Ching-Chi Yang, and Lih-Yuan Deng	
11	Appendices	219
	Max Garzon, Lih-Yuan Deng, Nirman Kumar, Deepak Venugopal, Kalidas Jana, and Ching-Chi Yang	

Acronyms

General

DS, DR, ML

Data Science, Dimensionality Reduction, Machine Learning

P

Probability (measure)

p, n

Dimensionality p , size n of a dataset

Data Science

[IrisC]

Iris Flower Classification problem

[MalC]

Malware Classification problem

[BioTC]

Biotaxonomy Classification problem

[SynC]

Noisy Synoptic Data Classification problem

[AstroP]

Planetary Body Location Prediction problem

[PhenP]

Phenotypic Prediction problem

[RossP]

Rosette Area Prediction problem (of a plant)

[LocP]

Provenance of a Plant Prediction problem

MDS

Multi-Dimensional Scaling (classical or metric)

ISOMAP

Isometric Mapping

t -SNE

t -Stochastic neighbor embedding

RND

Random Scaling

MNIST

MNIST dataset

Statistical

Ω

A population/sample space

$\mathbf{D}, \mathbf{X}, \mathbf{Y}$

A sample or dataset from Ω

P

Probability (measure)

$P(B | A)$

Conditional Probability of event B given event A

$X, Y, \dots$

Random variables (RVs, possibly with subindices)

$E(X)$

Expected value (mean) of a RV X

$\text{Var}(X), \sigma(X)$

Variance, standard deviation of a RV X

H

Shannon Entropy

$N(0,1)$	Standard Normal Distribution
$N(\mu, \sigma)$	Normal Distribution with mean μ and standard deviation σ
EDA	Exploratory Data Analysis
PC, IC	Principal Component, Independent Component
PCA, ICA	Principal, Independent Component Analysis
KPCA	Kernel Principal Component Analysis
LDA,QDA	Linear, Quadratic Discriminant Analysis
GLM	Generalized Linear Model
MLR	MultiLinear Regression
BL	Bayesian Learning
SVD	Singular Value Decomposition
SVM	Support Vector Machine

Computational/Machine Learning

AP	Computational/Algorithmic problem
[CharRC]	Handwritten Character Recognition problem
[TSP]	Traveling Salesperson problem
$\mathbf{D}_m$	The DNA space of Watson-Crick paired oligomers (pmers) of length m
$h, h(x, y), *, * $	Hybridization h -distance in $\mathbf{D}_m$
E, P^i	Equator, i^{th} Parallel in DNA space $\mathbf{D}_m$ (possibly with subindices m)
[CWD]	Noncrosshybridizing (nxh) basis/microarray Code-word Design problem
GNB	Gaussian Naive Bayes
MLR	MultiLinear Regression
MLP	MultiLayer Perceptron
FFN, NN, CNN	Feed-Foward, Neural, Convolutational Neural Network
SVM	Support Vector Machine
RF	Random Forest
kNN	k-Nearest Neighbors
NFL	No Free Lunch Theorem
R^2 , AICS, BIC	Assessment metrics of statistical solutions: R-squared, Akaike's Information Criterion, Bayesian Information Criterion
$a, p, r, F1, RMSE, RE, s$	Assessment metrics of ML solutions: accuracy, precision, recall, F1-score, Root Mean Square Error, Relative Error, silhouette

Mathematical $\approx, \propto, \ll, \gg$ $\alpha, \beta, \lambda, \nu, \sigma, x, y, \dots$ $\mathbf{x}, \mathbf{y}, \mathbf{u}, \mathbf{v}, X, Y, \dots$ $\mathbf{C}, \mathbf{X}, \mathbf{Y}, \mathbf{W}, \dots$ $\mathbf{N}, \mathbf{Z}$ **Q****R** **$\mathbf{D}_m$** $|\ast, \ast|; h, h(x, y), |xy|$

NMF

Approximately equal to, proportional to, much smaller/greater than

Scalars (possibly with subindices or superindices)

Vectors (possibly with subindices or superindices)

Matrices (possibly with subindices)

The set of natural numbers $\{0, 1, 2, \dots\}$, integer $\{0, \pm 1, \pm 2, \dots\}$ numbersThe set of rational numbers $\{p/q : p, q \in \mathbf{Z}, q \neq 0\}$ The set of real numbers $(-\infty, +\infty)$ The DNA space of Watson-Crick paired oligomers (pmers) of length m Distance; Hybridization h -distance in **$\mathbf{D}_m$**

Nonnegative Matrix Factorization

Chapter 1

What Is Data Science (DS)?



Max Garzon , Ching-Chi Yang , and Lih-Yuan Deng

Abstract Our ability to generate, gather, and store volumes of data (order of tera- and exo-bytes (10^{12} – 10^{18} bytes) daily) has far outpaced our ability to derive useful information from it in many fields, with available computational resources. The theme of this book is a review of Data Science (DS) through the lens of Dimensionality reduction (DR). Data science is about solving problems based on observations of factors (referred to as *co-variables*, *predictors*, or just *features*) that may determine a solution. Typical kinds of problems are described, including classification, prediction, and clustering problems, as well as data collection methods.

1.1 Major Families of Data Science Problems

In this section, problems typically dealt with in data science are described and grouped into three major groups: **classification**, **prediction**, and **clustering**.

A *population* Ω is the entire group of individuals/objects of interest for a given problem, while a *sample* refers to a (relatively very small) subset $\mathbf{D} \subseteq \Omega$ of the population.

Data science problems can be regarded as computational problems in the sense that they call for certain types of inputs as data to a problem (as described in Sect. 1.3 below) and spell out an expectation in terms of the kind of result to be produced as a solution. The solution is usually some sort of model or program/code, ultimately to be translated into some sort of physical device to be run, typically a conventional computer. In computer science, these problems are referred to as *algorithmic problems* and are typically expressed in the forms of questions of a

M. Garzon (✉)

Computer Science, The University of Memphis, Memphis, TN, USA

e-mail: mgarzon@memphis.edu

C.-C. Yang · L.-Y. Deng

Mathematical Sciences, The University of Memphis, Memphis, TN, USA

e-mail: cyang3@memphis.edu; lihdeng@memphis.edu

similar kind being provided as inputs from a population of objects, as illustrated with the three major families described in the following subsections.

1.1.1 Classification Problems

One of the most common kind of problems is *classification*.

Example 1.1 Iris flower classification is a relatively simple example. The population of objects under consideration for inputs is a feature vector x giving the length and width of sepals and petals of an iris flower. The flower is to be placed in one of the three categories. These categories must be specified in advance for the problem and determine the type of classification being made into disjoint and exhaustive classes, i.e., they constitute a *partition* Π of the population. If the categories change, then we are dealing with a different problem since the answers will have to change across the inputs. The problem can be stated precisely as follows, with $\Pi_1 = \{\textit{Setosa}, \textit{Versicolor}, \textit{Virginica}\}$. $\square$

[IrisC] IRIS FLOWER CLASSIFICATION (Π_1)

INSTANCE: A feature vector $x = (\text{sepal length, sepal width, petal length, petal width})$ (in cms) describing an iris flower, e.g., (4.6, 3.1, 1.5, 0.2)
 QUESTION: Which kind of flower in Π_1 is x ?

The brackets contain a mnemonic name **[IrisC]** used to refer to the problem in the sequel. A dataset for this problem is described in more detail in Sect. 11.5.

Example 1.2 Malware classification is a more complex example. The population of objects under consideration for inputs is a computer program/code that causes harm when run on a computer, usually referred to as *malware*. A sample from this population is discussed above in Sect. 1.1, where a given piece of malware is expected to be categorized into nine (9) classes or categories, as shown in Table 1.1. $\square$

The problem can then be stated precisely as follows.

[MalC] MALWARE CLASSIFICATION (Π_2)

INSTANCE: A piece of malware x
 QUESTION: Which category c in Π_2 does x belong in?

The Malware Classification Challenge (arxiv.org/abs/1802.10135, accessed October 2021) dataset was released in 2015 and is publicly available through *kaggle.com*. An available dataset consists of a set of 10,868 known malware files representing a mix of 9 different malware families, as summarized in Table 1.1. Each datapoint contains both the raw binary content of the malware file as well as metadata information extracted using the IDA disassembler tool, for a total of

Table 1.1 Microsoft’s malware classification problem. The labels for the classes in the partition Π are in the first column

Labels	Class name	Type of malware	Count
1	Ramnit	Worm	1541
2	Lollipop	Adware	2478
3	Kelihos_ver3	Backdoor	2942
4	Vundo	Trojan	475
5	Simda	Backdoor	42
6	Tracur	TrojanDownloader	751
7	Kelihos_ver1	Backdoor	398
8	Obfuscator.ACY	Any kind of obfuscated malware	1228
9	Gatak	Backdoor	1013
			10,868

1804 raw features. The classification challenge is to correctly assign every piece of malware in the population to one of the nine (9) possible malware classes defined with the challenge. A dataset for [MalC] is described in more detail in Sect. 11.5.

Example 1.3 A third and more difficult example of a classification problem is species identification in biology for a given group of species (a taxon) T . The population of objects under consideration consists of all living organisms belonging to a species in T . The taxon T is the union of all specimens/individuals in the species in the second column of Table 1.2, according to a given biological taxonomy, while the partition labels Π are shown in the first column. □

For a data sample, representative genes were obtained from mitochondrial DNA (specifically, cytochrome Oxidase genes COI, COII, and COIII, CytB) of 249 organisms (as shown in the fourth column of Table 1.2) collected from biological genetic repositories (e.g., GenBank at www.ncbi.nlm.nih.gov/genbank/). A dataset for this problem is described in more detail in Sect. 11.5. The problem can then be stated precisely as follows for an arbitrary taxon T .

[BioTC] BIOTAXONOMIC CLASSIFICATION

INSTANCE: A DNA sequence representing a living organism x
QUESTION: What species in T does x belong in?

In summary, a *classification problem* is defined by a partition Π of a population Ω (into a number of *mutually disjoint classes exhaustive of* Ω) and the problem is to place an arbitrary element of the population into the right class. A solution to the problem is a model that will make a (hopefully correct) decision for every element from the population (*not just the sample*) as to which class it belongs to.

Table 1.2 Organisms in the sample data for [BioTC] of a biological taxon. The 21 classes/labels in the partition I_7 are in the first column

Label	T : genus species	Common name	Count
1	<i>Apis mellifera</i>	Western honey bee	4
2	<i>Arabidopsis thaliana</i>	Cress	5
3	<i>Bacillus subtilis</i>	Hay/grass bacillus	18
4	<i>Branchiostoma floridae</i>	Florida lancelet	18
5	<i>Caenorhabditis elegans</i>	Round worm	6
6	<i>Canis lupus</i>	Wolf	18
7	<i>Cavia porcellus</i>	Pork	4
8	<i>Danio rerio</i>	Zebra fish	9
9	<i>Drosophila melanogaster</i>	Fruit fly	18
10	<i>Gallus gallus</i>	Red junglefowl	18
11	<i>Heterocephalus glaber</i>	Naked mole rat	3
12	<i>Homo sapiens</i>	Human	18
13	<i>Macaca mulatta</i>	Rhesus macaque	8
14	<i>Mus musculus</i>	House mouse	18
15	<i>Neurospora crassa</i>	Red bread mold	5
16	<i>Oryza sativa</i>	Asian rice	12
17	<i>Pseudomonas fluorescens</i>	Infectious bacterium	6
18	<i>Rattus norvegicus</i>	Brown rat	18
19	<i>Rickettsia rickettsii</i>	Tick-born bacterium	18
20	<i>Saccharomyces cerevisiae</i>	Yeast	18
21	<i>Zea mays</i>	Corn/maize	7
			249

1.1.2 Prediction Problems

A second kind of problems is *prediction*.

Example 1.4 The population could be plants x (e.g., a specimen of *A. thaliana*) and the question of interest is to know the area of the rosette of the plant covered by the spread of its leaves at maturity. The problem can then be precisely defined as follows for a taxon of plants T .

[RossP] ROSETTE AREA OF A PLANT (T)

INSTANCE: a (long) DNA sequence x (over the alphabet $\{a, c, g, t\}$) representative of a plant in T ?

QUESTION: What is the area of x 's rosette when fully grown?

Note that there is a dependency (not necessarily functional since environmental factors appear to be involved, e.g., how well the plant was fed) that roughly determines the rosette area given the genome of the plant. A similar problem can be posed for the weight of a person as a function of their height. $\square$

Example 1.5 Another example is the problem **[LocP]** of determining the origin of the plant for a taxon of plants T based on DNA sequence alone, as stated below. $\square$

[LocP] PROVENANCE OF A PLANT (T)

INSTANCE: a (long) DNA sequence x (over the alphabet $\{a, c, g, t\}$) describing a plant in T

QUESTION: Where on Earth (latitude, longitude) was x grown?

In summary, a *prediction* problem is defined by a function f on a population that associates numerical values to every element in the population (usually hard to measure directly) and the problem is to find the value $f(x)$ of the function f for an arbitrary element x in the population Ω . A solution is a model to determine that value based on other features identifying the input elements x from Ω . Since this may be difficult, one may need to settle for just approximate values.

1.1.3 Clustering Problems

A third kind of common problem is *clustering*.

Example 1.6 When a typical US town grows to a certain size, it is desirable to set up some fire stations just in case a house or building catches fire. Naturally, when your house is on fire, you would like the station to be as close as possible. To conceptualize the problem, we can specify a number n of points c_i ($i = 1, \dots, m$) in a 2D Cartesian plane to designate the positions of the fire stations. Thus, any fire emergency at a house located at any given position x will be tended to by fire trucks at station c_{i_0} closest (at minimum distance) to it in distance, i.e., so that

$$i_0 = \arg \min_i \|c_i - x\|.$$

So, if there are two stations c_1, c_2 , locations get clustered in two clusters, defined by half-planes separated by the perpendicular bisector of the segment joining them. If there are three stations, the regions are like cake slices, as shown in Fig. 1.1a, with boundaries being the relevant fragments of the perpendicular bisectors of all three pairs of stations. More stations will complicate the partition of the 2D plane, as illustrated in Fig. 1.1b, but along the same ideas. This kind of diagram is called a *Voronoi diagram* and is defined by the fire station locations $c_1, \dots, c_m$, which are generically referred to as *centroids*. This diagram is a solution to the problem of clustering the population into groups that are each served by the same fire station according to the ordinary Euclidean (ℓ_2) distance. $\square$

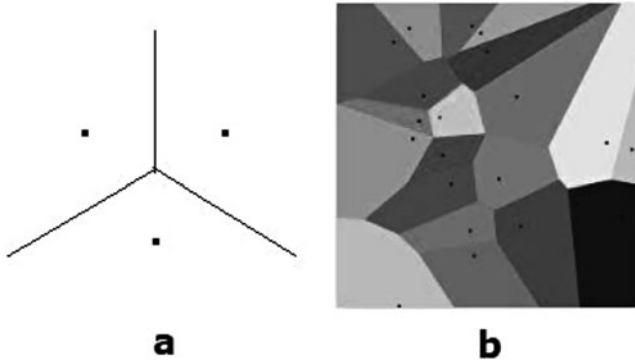


Fig. 1.1 Voronoi diagrams as solutions to the clustering problem defined by the Euclidean distance in the population of the 2D plane for (a) three points (e.g., fire stations) and (b) 20 points

In summary, a *clustering problem* is defined by a metric (usually a distance function, defined precisely below) function between elements in a population to capture degrees of (dis)similarity, and the problem is to produce a partition (*unlike a classification problem, where the partition is given*). A solution is to find a model that will produce a partition such that elements in any one cluster are more similar among each other than to elements in the other clusters.

It may be difficult to make a crisp distinction between these kinds of problems. For example, a partition Π in a given classification problem can be regarded as a function f on the same population assigning a numerical label in an enumeration of the classes in some order, so it would appear that every classification problem is a prediction problem (especially if a solution to the problem is used to label unknown inputs). Conversely, if the range of f in a prediction problem is small, one can regard it as a classification problem as well. Thus, we will speak of prediction problems only when the range of f is a fairly large set of (perhaps infinitely many) analog values. Therefore, typical classification problems usually involve a few categories in their partitions (in the order of at most tens). Likewise, once a solution to a clustering problem is obtained, one can use the clusters as a partition in a classification problem to predict the clusters for unknown elements in the population.

Other kinds of problem besides these three categories can certainly arise in data science. An example involves dynamical sequences of actions as inputs (e.g., a video of a pedestrian crossing a street) and questions concerning the next expected action (e.g., will the pedestrian run across or turn back on an upcoming car). There is no telling what kind of problem could arise in any domain science. However, these three main categories can always be used, at least in the first approximation to a given problem. For example, the answers to the pedestrian problem can range from a simple classification (forward/stop/back) to a prediction of the next step, to prediction of complex sequence of moves. (What sequence of actions will the pedestrian exactly take in the next 5 seconds?)

1.2 Data, Big Data, and Pre-processing

In this section, a characterization of what is understood by data and major techniques to conceptualize, visualize, and understand data are reviewed. Understanding data, particularly large datasets and big data, can prove to be extremely difficult, particularly when faced with acting on decisions based on data analytics discussed in later chapters to sell a solution of a particular problem to management. Further, the results of data analysis may impact an individual's health, or even large communities when applied at large scales in the real world, so understanding the scope and rationale for the results of the analyses is critical.

1.2.1 What Is Data?

A basic question that any book on data science must address is, what exactly is *data*? Many answers have been given to this question since the dawn of the information age in the 1940s. For example, well-known computer scientists (such as Peter Naur of BNF: Backus-Naur Normal Form fame) wanted to redefine computer science as *data science* back in the mid-1950s. So have well known statisticians (such as C.F. Jeff Wu of EM optimization and '*Statistics = Data Science*' fame in the 1970s), implying that data is about observations in the form of numbers, that it can be used to make inferences about entire populations and that there is a computational component to it. Like an iron age man on attempting to answer the question *What is iron?*, we can tell data when we see it but are very hard pressed when asked to give a rigorous noncircular definition of data. Plain numbers in abstract or computer programs to crunch them cannot really be data as it is understood today. We will thus adopt the following characterization.

Data is an objective recording of one or several event(s) in the real world that is accessible at later times for perusal and analysis by at least one person.

Example 1.7 Thus, the following do *not* qualify as data: the plain occurrence of an event; a memory in someone's head as a witness of an event (unless an objective recording is made of the witness describing or reliving the event -humans update their memories as they remember them); the contents of a computer's RAM memory (unless it is being recorded); and so forth. On the other hand, the following can be considered data: website contents containing inaccurate facts and spreading misinformation; inaccurate scientific observations (due to faulty equipment or even malicious intent); fossils of dead or extinct animals; videos of a person (even if no longer living); and statistics collected through observations or machines and stored on paper or machine. Thus, data requires the so-called four Ws: *What, Who from, Where, and When*. □

It will be useful to make further distinctions with other related concepts, such as *information* and *knowledge* (*wisdom* is out of reach). In this book, data differs

from information in that information requires, in addition, a purposeful observer, for example, trying to solve a (data science) problem, as discussed below in Sect. 1.3. Data becomes information if it can be used to extract a solution to the problem with some processing, as illustrated by many examples in the following chapters. (The more complex concept of knowledge will be re-visited in Sect. 10.2).

Data can assume a variety of formats, for example, numbers, text, DNA or protein sequences, voice recordings, images, x-rays, videos, browser histories, time stamps, and so forth. These recordings are considered to be *raw* or *primary* data. They usually require transformation and processing to enable the kind of analysis done in data science to solve a problem, as will be discussed in this section and later chapters.

The most common format for the simplest kind of data is a two-dimensional (henceforth, just 2D) *table* consisting of rows and columns, usually organized so that attributes of the data are placed in the columns and each specific observation (sometimes referred to as a *datum*) occupies an entry in the rows. Herein, they will be referred to as *features* and *records*, respectively. Therefore,

a *record* is a p -dimensional (henceforth pD) vector of features values (datums, or its latin plural, *data*). A table can be regarded as an nD vector of pD vectors, the records. The *dimensionality (size)* of the data/sample is p (n , respectively.)

Mathematically, a table can be regarded as a sample and its columns can be regarded as random variables X , as described above in Sect. 1.1 (see Sect. 11.1 for statistical background concepts). Single tables are good enough for small datasets but a more complex dataset or problem may require many such tables to form a *data corpus* for the problem simply because there may be many very different kinds of observations and attributes about objects in the same population the corpus is all about. Data corpora raise the problem of properly organizing the data so that humans can comprehend it. A criterion or logic to *organize* a dataset or data corpus is usually called an *ontology* for the data.

Example 1.8 If the problem of interest is the earning power of American taxpayers, the underlying population is, of course, American citizens who must pay income taxes by law. An important table here would contain a taxPayerID, a fiscal year, an AGI (AdjustedGrossIncome), and possibly taxDue. Another table may include demographic information, including taxPayerID, taxPayerFullName, CurrentAddress, priorAddress, and bankAccountNr as features. A third table may contain information about itemizedDeductions for a given year, and so forth. Many other tables are probably necessary to keep a full history of a taxPayerID in the full data corpus. The ontology in this case is the grouping of the information by taxPayerID and into a number of categories (like tax data for the first table, demographic data for the second, itemized deductions for the third) for each fiscal year. Other ontologies are certainly possible (e.g., brute force, a single table with one record for each American tax payer, and hundreds or even thousands of attributes for each of them), although probably less desirable because it is less useful to work with and/or understand the data. □

On closer examination, a feature and its values (datums) in a table can be classified in various ways. For example, they could be qualitative or quantitative variables. A *qualitative* variable only observes values in a limited and fixed set of possible outcomes. If the list of outcomes can be sorted naturally, they are referred to as *ordinal*. Otherwise, the feature is called *nominal*. On the other hand, a *quantitative* variable observes values with a very large number of possibilities, usually the result of measuring something. They can be further separated into *discrete* or *continuous* variables. A discrete variable takes a value from a set of finite or countable numbers (indexable by natural numbers), while a continuous variable takes values from a set of uncountable numbers (like real numbers).

Example 1.9 (Qualitative or Quantitative) In order to distinguish qualitative and quantitative variables, a useful criterion is, *Can the values of the variable be added?* If a variable represents what is in a shopping bag, the bag may contain *oranges* and *apples* as answers. Since oranges and apples cannot be added (orange-apple makes no sense as a fruit), the list of items in the bag is qualitative data. However, if one is counting items in the bag, the data will be quantitative, say $2+3=5$ items. $\square$

Example 1.10 (Ordinal or Nominal) It is hard to objectively decide on a natural order for apples, oranges, and bananas (though their *names* can easily be sorted alphabetically). The variable typeOfFruit is nominal. On the other hand, a Likert scale for answering whether one likes a particular fruit is ordinal (e.g., *Like*, *Neutral*, *Dislike*). $\square$

Example 1.11 (Discrete or Continuous) The criterion for distinguishing discrete or continuous variables can be, *Could a decimal number be an answer?* For example, if one is counting how many people are in a theater, a number such as 3.14 is nonsense. The number of people in a theater is discrete. However, if one considers the average number of people in a theater for one show over a period of one year, the number 3.14 could be an answer; this variable is continuous. $\square$

1.2.2 Big Data

A similar problem to that in Example 1.8 arises with financial credit card transactions for a given bank or credit card company and many other problems for government and private organizations in every country. The data corpora can become really huge and they are usually referred to as *big data* (in the order of terabytes and larger). This fact raises the important issue of *understanding* datasets and data corpora. A good ontology will enable a human to grasp the large picture of what the data is all about (the four Ws) and make it possible to develop effective strategies to solve problems about the given population.

Big data is usually characterized by the so-called five V's:

- *Volume* (raw number of records/amount of data),
- *Variety* (how diverse is the type, nature, and format of the data),

- *Velocity* (speed of data generation),
- *Veracity* (trustworthiness/quality of captured data), and
- *Value* (insight and impact afforded by the data about the overall population).

Example 1.12 (Mobile Data with High Volume and Velocity) According to *ericsson.com*, total global mobile data traffic reached 49 exo-bytes (EB) per month at the end of 2020. It is expected to grow with the launch of the 5G internet. By comparison, the Iris dataset only contains 2.6×10^3 bytes, whereas a month of total global mobile data contains 4.9×10^{18} bytes. The data is generated extremely fast www.ericsson.com/en/mobility-report/dataforecasts/mobile-traffic-forecast, so this dataset exhibits very high volume and velocity. □

Example 1.13 (Data Variety) Long ago, data might have referred to a well-organized spreadsheet, e.g., .txt or .csv files. Variety refers to any data that could be mixedly generated by humans or by machines. For example, a Fitbit generates sensor data of a body by continuously monitoring the body movements. Browser history is generated by human activity. Twitch stream is a live video that records the show and chats between hosts and viewers. A data corpus containing these chats has high variety. □

Example 1.14 (Anonymity on the Internet) For freedom of speech, websites may offer online anonymity. However, it comes with the problem of data veracity. It is easy to lie and the information therein cannot be trusted. Moreover, fake information usually spreads faster than facts. □

Example 1.15 (Educational Data) Data value is a subtle concept. It refers to how useful the data is to answer research questions and turn it into business intelligence. In the case of educational data, students' progress/scores are stored from K-12th grades. If the data are just stored without further use or analysis, it offers little value. However, if one can utilize the data to further identify personalized key learning strategies for students, its value increases tremendously for the benefit of society. □

Computational resources and techniques today are not able to process the extraordinary full volume of data being generated in so many fields. Because of the overwhelming growth of data collection, it can be challenging to extract useful information from big data with current computational resources. Choosing a good subset of the data as the training sample to build some suitable models is a critical issue. For proper big data modeling and analysis, it is common to use some statistical and machine learning methods for preliminary pre-processing in order to make big data amenable to analysis. It helps to reduce the computational time of the analysis from years to feasible.

1.2.3 Data Cleansing

Raw data, i.e., data that has not been processed and is not ready for analysis, requires *pre-processing* or *cleansing* (e.g., (re)formatting, organization, and selective extraction) to become data ready-to-use (sometimes referred to as *cooked data*). The process is also referred to as data *cleaning*, *scrubbing*, *wrangling*, and so forth. Cleansing can be manually done or automatically done by a machine for large datasets.

Example 1.16 (Handwritten Character Recognition) In handwritten character recognition, a trace of handwriting is processed (e.g., zip codes on ordinary mail envelopes in order to route the envelopes properly). Further details about this dataset can be found in Sect. 11.5. □

Example 1.17 (Malware Detection) Malware Classification ([MalC]) requires professional/domain knowledge to extract the information from an executable (.exe file) into cleanly labeled data suitable for analysis, where the rows are the pieces of observed malware and columns are the features extracted from the executable file. □

Example 1.18 (Stock Market) In the stock market, a trade can be made and recorded nowadays in approximately $\frac{1}{64}$ millionth of a second, i.e., at the rate of 64M per second. It is difficult to analyze this raw data because of its volume. It has to be organized and transformed for further analysis/prediction, e.g., by trading volume or opening/closing price in a day, to be of value. □

In Examples 1.16–1.18, raw data are not ready for answering research questions. Additional procedures are required to organize observations and their features.

Cleansing is the process of screening and/or transforming data to enable processing and analysis to derive value from it.

Needless to say, cleansing must be conducted prior to data analysis, although it is sometimes done later upon failure to get useful results from an analysis due to multiple reasons. The data may be *incomplete*, *noisy*, *inconsistent*, *duplicated*, *irrelevant*, or simply be *missing* values. The aim of cleansing and pre-processing is to remove noise, standardize the format, and retain useful information, so as to make sure the scripts for analyses that expect a common format will not crash due to formatting errors or missing values. This is a pre-condition to extracting value from the data analyses. There are several common ways to accomplish this goal.

1.2.3.1 Duplication

For example, if a minority opinion is surveyed repeatedly and a large number of duplicates are added to the dataset, the minority opinion might become the majority opinion. By removing duplication, one can avoid biased inferences and misleading conclusions.

Example 1.19 (Online Sweepstakes) Online sweepstakes offer great prizes. If one enters multiple times, the case is duplicated. To be fair, the website keeper should remove the duplication or prevent a person from entering the sweepstakes twice. $\square$

1.2.3.2 Fixing/Removing Errors

It is always recommended to re-check the source to obtain the correct values for the missing values. However, this is not always possible. Certain values cannot be recovered in many situations. One way to handle it is to change the erroneous datum to a missing one and handle it accordingly.

1.2.3.3 Missing Data

Missing data/value refers to a datum being unavailable or corrupted. Data can be missing due to many reasons. For example, in a survey, an individual chose not to report an answer; the datum was not observable/available; or the datum went missing during data (pre-)processing. *Missing value imputation* is one common technique to obtain a missing value. Some common imputation methods are:

1. *Mean imputation*

The missing value x_{ij} for the variable X_j can be filled in as the mean, i.e., $\bar{X}_j$. This method is generally used for a continuous variable since the mean of a discrete variable or a categorical variable might not make sense or be appropriate.

2. *Interpolation and extrapolation* (for numerical data)

The missing value can be imputed by prediction. While interpolation implies the prediction falls within the range of data points, extrapolation implies the prediction falls outside the range of data points. Basically, by connecting two close data points without missing values with a straight line, the missing values can be imputed as the value on the fitted line.

3. *Regression imputation*

The missing value can also be imputed by a model for the data obtained later by analysis. One type of model is the regression type, which will be discussed in detail in Sect. 2.1.

1.2.3.4 Outliers

An *outlier* is a data point containing a datum that is out of the range of the other points in the feature. While the multivariate situation is considered, a distance measurement, e.g., Mahalanobis distance, will be used to disrobe how far the observation is from the bulk of the data. The goal of a whole research area called *outlier detection* (also *anomaly detection*) is to estimate this distance and identify the outliers. It can be used for detecting unusual observations, such as bank fraud.

Example 1.20 (Bank Fraud Detection) It is essential to detect fraudulent transactions not only for a personal account but also for money laundering. If an account usually has transactions for grocery shopping, the occurrence of a large transaction appears as an outlier. A bank will issue a hold on this type of activity until confirmed. How large is considered “large” requires fine-tuning. (While keeping its customers’ assets safe is important, the bank will not want to issue a hold every time and ruin most honest users’ experience.) □

1.2.3.5 Multicollinearity

Multicollinearity (also collinearity) refers to the case where one predictor variable in a multiple regression model (described in Sect. 2.1) can be linearly predicted from the others with high accuracy. This technique is similar to combined inter-extrapolation, but with more than one missing value. (Some consider dependencies of this type to be “dirty” and require cleansing.)

1.2.4 Data Visualization

One of the major problems with data, particularly big or complex data, is to make it *understandable* to humans. This is particularly important for humans to verify, interpret, and/or rationalize the results of any analytics to act on a solution to a problem as being consistent with the data that led to it. In Human–Computer Interaction (HCI), the problem is usually solved by reaching into the human’s head and finding a metaphor to transform and present the data to enable his/her brainpower with an appropriate transformation into a more familiar situation.

Example 1.21 A typical example is the problem of navigation on the metro system in a big city. A computer system can only store a list of metro lines and the locations (latitude or longitude) of their stop stations. Presenting a time table of these stations to a human will not enable easy navigation to instruct her to move from point A to point B. The visualization in the form of a graph in various colors (the lines) roughly following the layout of the city and their points of intersections and stops along the lines is a familiar and effective solution because humans easily understand space, distances, and orientation to navigate the city. □

In statistics, data visualization refers to exploratory data analysis (EDA) as an approach to analyzing datasets to summarize their main characteristics, often using graphics about statistics of the data features. Based on the variable’s type (qualitative/quantitative, discrete/continuous), different EDA methods can be used. Table 1.3 and Figs. 1.2, 1.3, 1.4, 1.5, and 1.6 illustrate various techniques with the Iris dataset.

Table 1.3 A stem-and-leaf plot of the sepal length for the Iris dataset

Key: 4 3 = 4.3 cm	
4	3444
4	566667788888999999
5	00000000001111111122223444444
5	55555556666667777777888888999
6	0000001111112222333333334444444
6	555556677777778889999
7	0122234
7	677779

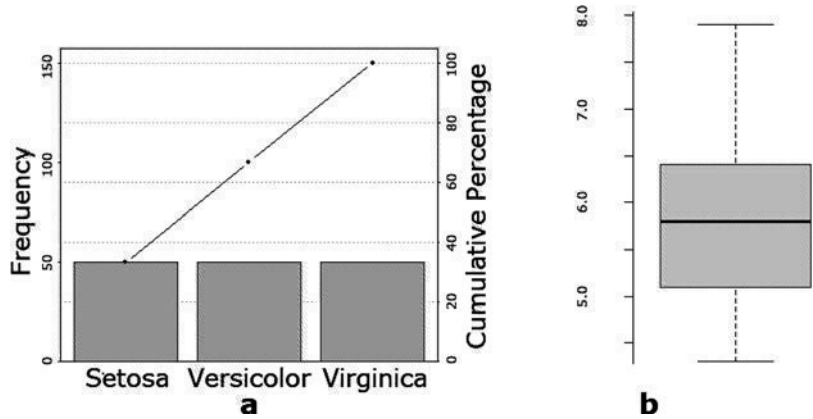


Fig. 1.2 (a) A Pareto chart with the count for the species and (b) a boxplot for the sepal length of the species in the species in the Iris dataset

Example 1.22 (Boxplot of a Continuous Variable) Boxplots are useful to represent the five number summary of a feature: the *minimum*, the *first quartile*, the *second quartile*, the *third quartile*, and the *maximal* of a variable. Figure 1.2b shows a boxplot of the sepal length for the Iris dataset.

Note that *quantiles* are points that divide the range of the observations in a sample variable into subranges (in a sequence) with equal probabilities. The most common quantiles are the 4-quantiles (are also called *quartiles*), the 10-quantiles (are also called *deciles*), and the 100-quantiles (are also called *percentiles*). □

Example 1.23 (Pareto Chart of a Variable) A *Pareto chart* is a bar graph. Figure 1.2a shows a Pareto chart of the three species in the Iris data sample: Virginica, Versicolor, and Setosa. The lengths of the bars are proportional to the frequency of each category and the curve presents their cumulative distribution. □

Example 1.24 (Histogram of a Continuous Variable) A *histogram* is an approximate representation of the distribution of numerical data. A histogram of the sepal length for the Iris dataset is shown in Fig. 1.3. □

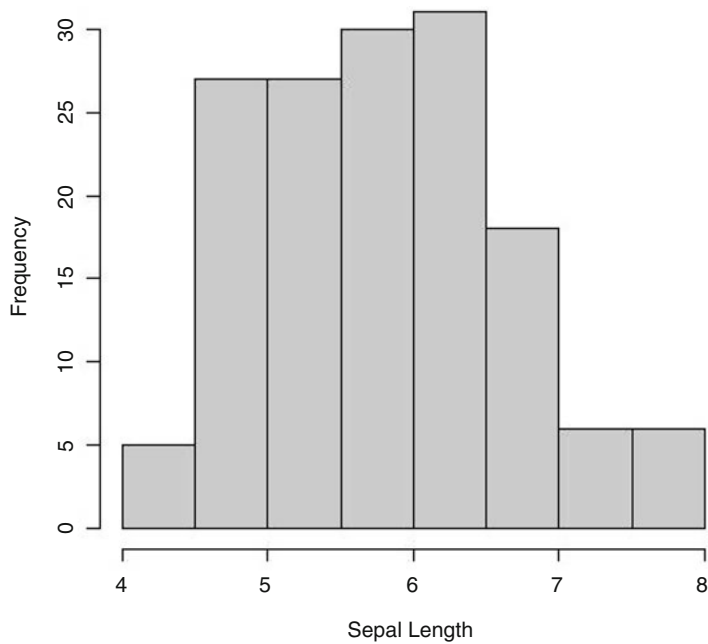


Fig. 1.3 A histogram of the sepal length in the Iris dataset

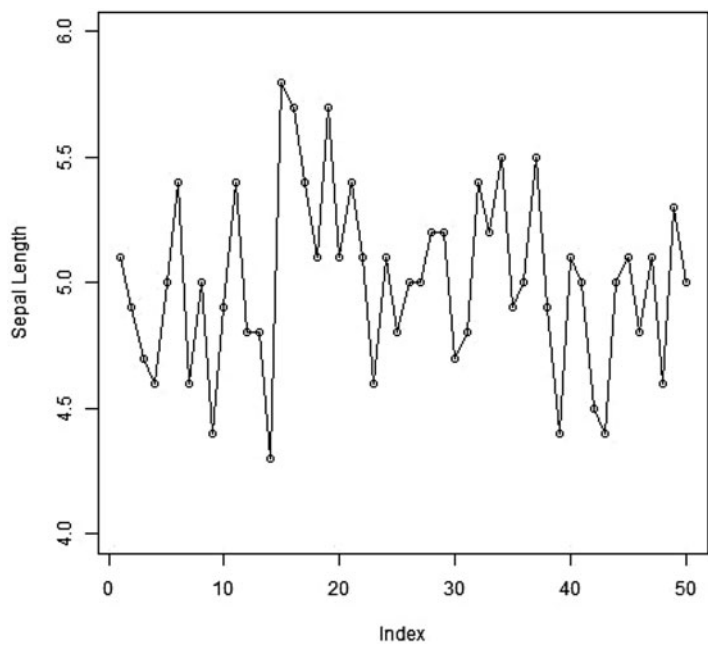


Fig. 1.4 A run chart of the sepal length in the Iris dataset

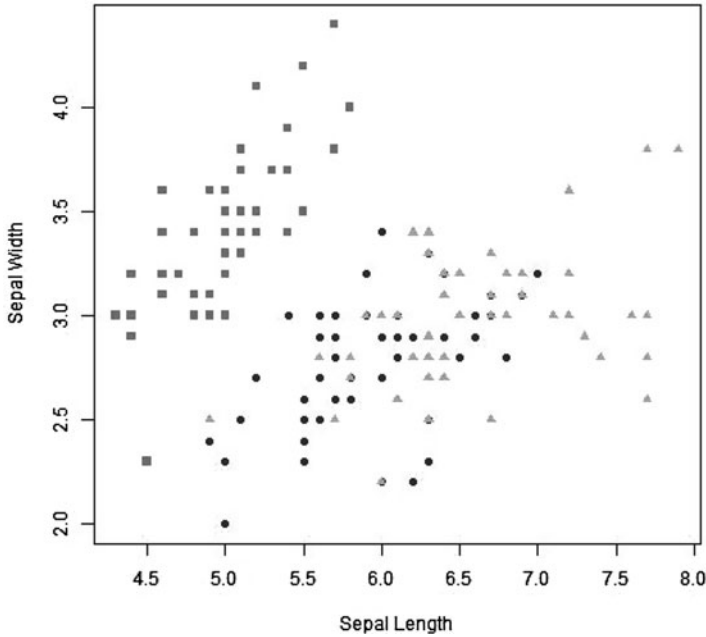


Fig. 1.5 A scatter plot of sepal length and width for the Iris dataset

Example 1.25 (Stem-and-Leaf Plot for a Quantitative Variable) A *stem-and-leaf plot* is a way of presenting a quantitative variable. The plot can also be treated as a table. For example, a stem-and-leaf plot of the sepal length in the Iris dataset is shown in Table 1.3. By studying a stem-and-leaf plot, one can achieve a visualization similar to the one obtained from a histogram. □

Example 1.26 (Run Chart)

To understand the dynamic change of a variable over time, a run chart is usually used. If the first 50 observations are recorded in a sequence over a time period of time for the Iris dataset, one may study whether the size of the flower will change over time. A run chart of the sepal length for the Iris dataset is shown in Fig. 1.4. □

Example 1.27 (Scatter Plot) It is common to have more than one variable (feature) in a dataset. A *scatter plot* represents an observation by a dot in the figure for two features (x -axis and y -axis). A scatter plot of the sepal length and sepal width in the Iris dataset is shown in Fig. 1.5. □

Example 1.28 (Multivariate Chart) A *multivariate chart* is used to combine different visualizations of multiple variables into one single chart to enable an overall comparison. A multivariate chart for the Iris dataset is shown in Fig. 1.6. □

Example 1.29 (Targeted Projection Pursuit) In big data, there are too many variables in the data corpus. It is hard to visualize all of them at once. *Targeted*