

Lecture Notes for Machine Learning and Data
Science Courses
Information School, University of Washington

Ott Toomet

July 21, 2023

Preface

This is a collection of notes made for INFO370, INFO371, IMT573 and IMT574 courses, taught at the Information School, University of Washington. It began as a collection of such topics where I could not find good material at suitable level and suitable coverage. Later I have also added some material where other good source exist, mainly to develop my presentation and have to deal with fewer reading sources. The pdf is available at my [UW faculty page](#). Check also out the [python companion](#) (preliminary) and [R companion](#) (even more preliminary).

The source of these notes is available at it's [bitbucket repo](#), feel free to leave feedback in it's issue tracker.

The text is licensed as [CC BY 4.0](#), the images have different copyrights, see the captions and the readme files in the corresponding folders in the [Bitbucket repo](#).

Contents

0.1 Notation	vi
1 Introduction to Statistics	1
1.1 Different Kind of Values	2
1.2 Doing Descriptive Statistics	6
1.3 Basics of Probability Theory	21
1.4 Distributions	48
1.5 Statistical Inference	66
1.6 Lies, Damned Lies, and Statistics	83
2 Linear Algebra	91
2.1 Why Linear Algebra in Machine Learning	91
2.2 Vectors and Vector Spaces	92
2.3 Matrices	103
2.4 Application: wireframe images	117
3 Regression Models	123
3.1 Linear Regression	123
3.2 Logistic Regression	179
3.3 Linear probability model	190
4 Causality	191
4.1 Introduction	192
4.2 What is cause?	193
4.3 Causality with data: three explanations	194
4.4 Strategies for Causal Inference	199
4.5 Causal inference in linear regression framework	205
4.6 Cognitive Illusions in Causal Inference	228
4.7 Causality and complex social problems	229
5 Predictive modeling and model goodness	231
5.1 Predictive modeling	231
5.2 Categorization	231
5.3 Overfitting and Validation	245
6 Machine Learning Models	253
6.1 Metric Distance: A Revisit	254

6.2	<i>k</i> -Nearest Neighbors	263
6.3	Trees and tree-based methods	267
6.4	Naïve Bayes	281
6.5	Support Vector Machines	296
6.6	Neural Networks	298
6.7	Comparison and Review	313
7	Different Types of Data	315
7.1	Numeric Data	315
7.2	Images	316
7.3	Text as Data	323
8	Machine Learning Techniques	337
8.1	Loss Function and Non-Linear Optimization	337
8.2	Gradient Ascent	341
8.3	OLS Example	344
8.4	Gradient Descent	345
8.5	Key Concepts	349
8.6	Feature Selection and Regularization	354
9	Unsupervised Learning	357
9.1	Introduction	357
9.2	Cluster Analysis	358
9.3	Principal Component Analysis	370
9.4	Comparison of Clustering and PCA	383
10	Applications	387
10.1	Recommender Systems	387
10.2	Generating Content: Generative Adversarial Networks	392
11	Responsible Data Science	395
11.1	Explainable AI	395
11.2	Social inequality	396
11.3	Fairness and discrimination	397
11.4	Human Versus Algorithmic Decision-Making	402
A	Mathematics	403
A.1	High-School Mathematics	403
A.2	Matrix calculus	404
B	Datasets	413
C	Exercise Solutions	415
C.1	Introduction to Statistics	415
C.2	Linear Algebra	421
C.3	Regression models	423
C.4	Causality	426

C.5 Predictive Modeling	427
C.6 Machine Learning Models	429
List of Cheatsheets	433
List of Examples	435
List of Figures	437
List of Tables	445
List of Exercises	449
Index	451
References	459

0.1 Notation

These notes contain a lot of mathematical notation. Here is a list of conventions and more common notation:

Greek alphabet Mathematical notation uses Greek alphabet extensively. Table 1 shows a complete list of it, both in upper and lower case form. Note that several upper case letters are identical with the corresponding Latin letters, and there are two ways to write certain lower case letters.

Table 1: Greek alphabet

Letter	Lower case	Upper case
Alpha	α	A
Beta	β	B
Gamma	γ	Γ
Delta	δ	Δ
Epsilon	ϵ, ε	E
Zeta	ζ	Z
Eta	η	E
Theta	θ, ϑ	Θ
Iota	ι	I
Kappa	κ	K
Lambda	λ	Λ
Mu	μ	M
Nu	ν	N
Omicron	\o	O
Pi	π	Π
Rho	ρ, ϱ	R
Sigma	σ	Σ
Tau	τ	T
Upsilon	υ	Y
Phi	ϕ, φ	Φ
Chi	χ	X
Psi	ψ	Ψ
Omega	ω	Ω

Numbers We generally follow the following notation:

- number of observations (cases) is denoted by N .
- number of variables in a model is denoted by K .
- predicted or estimated values are denoted by “hat” $\hat{\cdot}$, such as \hat{y} for predicted y and $\hat{\beta}$ for estimated value of β .

Sets

- general sets are denoted by calligraphic letters like \mathcal{S} , \mathcal{A} , \mathcal{Q} .
- number of elements in a set is denoted with the same symbol as norm, $||\mathcal{S}||$.
- Set of integers is denoted by \mathbb{Z} , set of pairs of integers is \mathbb{Z}^2 , and so on.
- Set of real numbers is denoted by \mathbb{R} , pairs of real numbers are \mathbb{R}^2 , etc.
- Intervals are denoted by $[a,b]$ for a closed interval from a to b , (a,b) for an open interval, and $[a,b)$ and $(b,a]$ for semi-open intervals.

Scalars, vectors, matrices

- scalar values (just numbers) are denoted with ordinary latin and Greek letters, such as a and v . Upper case letters denote integer constants (such as number of observations), lower case letters are both continuous values and integer indices. For instance, in case of x_i , x may be a continuous variable while i is an integer index.
- vectors are denoted in bold lower-case letters, for instance \mathbf{x} and ϵ .
- Matrices are written in sans-serif capital letters. For instance, \mathbf{A} and \mathbf{I} are matrices.
- unit matrix (identity matrix) is denoted by \mathbf{I} , or \mathbf{I}_n in case we want to stress it is $n \times n$ identity matrix.
- vectors are just $n \times 1$ or $1 \times n$ matrices. We denote by \mathbf{x} an $n \times 1$ column vector and \mathbf{x}^\top an $1 \times n$ row vector. When defining a column vector, we often use notation like $(1 \ 2 \ 3)^\top$, a row vector transposed, do denote the column vector $\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$.
 - Norm of vector is denoted by $||\cdot||$, e.g. $||\mathbf{v}||$.
 - We use dot, \cdot , to denote multiplication where it helps to understand the formulas. This applies to both scalar and matrix multiplication. So

$$\lambda \mathbf{x}^\top \mathbf{y} \quad \lambda \mathbf{x}^\top \cdot \mathbf{y} \quad \lambda \cdot \mathbf{x}^\top \cdot \mathbf{y} \quad (0.1.1)$$

are all equivalent and denote a product of three factors.

- We use \odot to denote elementwise product of matrices and vectors. For instance,

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \odot \begin{pmatrix} 10 & 20 \\ 30 & 40 \end{pmatrix} = \begin{pmatrix} 10 & 40 \\ 90 & 160 \end{pmatrix} \quad (0.1.2)$$

- we use large dot, \bullet , to denote “all indices at this dimension”. For instance $\mathbf{A}_{\bullet 2}$ means second column of matrix \mathbf{A} while $\mathbf{A}_{2\bullet}$ is its second row.

Functions We use notation $f : A \rightarrow B$ to denote a function that maps every element of set A to an element of set B . For instance, $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is a function that maps elements from \mathbb{R}^n to \mathbb{R}^m , i.e. assigns a m -dimensional real vector to every n -dimensional real vector. $g : \mathbb{R} \rightarrow \mathbb{R}$ is the “traditional” function that computes a new real number from every other real number.

We use the following special functions in the text:

- \log indicates natural log. 10-based or 2-based logs are denoted as \log_{10} and \log_2 .
- indicator function $\mathbb{1}(A)$:

$$\mathbb{1}(A) = \begin{cases} 1 & \text{if } A \text{ is true} \\ 0 & \text{otherwise.} \end{cases} \quad (0.1.3)$$

For instance, $\mathbb{1}(x > 0)$ is 1 if x is positive, and zero otherwise.

Indicator function is almost trivial to compute on computer: for the example here, just a logical operation $x > 0$ will work in many programming language.

Acronyms Here is a list of common acronyms used in the text:

- *c.d.f*: cumulative distribution function
- *CI*: confidence interval
- *CLT*: Central Limit Theorem (p 60)
- *GA*: Gradient Ascent (p 341)
- *i.i.d*: independently identically distributed
- *LN*: log-normal distribution (p 56)
- *p.d.f*: probability density function
- *p.m.f*: probability mass function
- *ReLU*: rectified linear unit (p 305)
- *RV*: random variable (p 37)

Chapter 1

Introduction to Statistics

This chapter gives an overview of the statistical methods that are needed for the machine learning methods later. ML is statistics-heavy, most of the models we discuss below are essentially statistical models, and statistics is also the tool that allows us to understand data and discuss the performance of the models.

Descriptive statistics is widely used to explore and summarize data. This includes computing means and variances, comparing certain obvious groups in data, analyzing data quality, and creating plots and tables. These methods are somewhat distinct from the formal and precise mathematical theory, *mathematical statistics*. Both branches of statistics are very important in data science. Much of “know your data”, and a large chunk of data visualizations and presentations can be counted as descriptive statistics; and machine learning is largely based on formal statistical models. We start with data description, thereafter continue with descriptive statistics, and consider certain concepts of mathematical statistics afterward. The final section discusses the problems related to understanding statistical results.

Contents

1.1	Different Kind of Values	2
1.1.1	Measures: Possible Mathematical Operations	2
1.1.2	Values: Which Values Are the Possible	5
1.2	Doing Descriptive Statistics	6
1.2.1	Data Quality	6
1.2.2	Describing Data	7
1.2.3	Sampling	18
1.3	Basics of Probability Theory	21
1.3.1	Events and Sample Space	21
1.3.2	Probability	24
1.3.3	Conditional Probability and Bayes Theorem	25
1.3.4	Random Variable	37
1.3.5	Expected Value and Variance	40
1.4	Distributions	48
1.4.1	Discrete Case	48

1.4.2	Continuous RV-s	52
1.4.3	Central Limit Theorem	60
1.4.4	Information and Entropy	63
1.5	Statistical Inference	66
1.5.1	Statistical Hypotheses and Hypothesis Testing	66
1.5.2	Doing Statistical Inference	71
1.5.3	Comparing Distributions	78
1.6	Lies, Damned Lies, and Statistics	83
1.6.1	Statistical Fallacies	83
1.6.2	Misusing Statistics	88

There are broadly three reasons we use statistics in data science and machine learning:

- Descriptive statistics is a good way to summarize data. For instance, GDP per capita (2017, nominal) in Madagascar is \$450 and in Canada it is \$44,841 ([world-o-meter data](#)). Despite all problems with reducing the complexity of an economy into a single index, in practice it is a very good proxy to describe life quality in various aspects in these places.
- Data is imprecise and we describe errors as random variables. This may include missings, measurement errors, computation errors, validity and reliability issues. This is one of the major motivations to use mathematical statistics for data analysis.
- The world is complex and unpredictable, and we model uncertain factors as random variables. This is the other reason mathematical statistics has done such strong inroads into econometrics and machine learning. It is also related to the previous problem, that of incomplete data—if we had better data, we would be able to predict the world better. But we have to live in this world, using data we have.

1.1 Different Kind of Values

Data contains values of different types. Here we discuss two potential ways to categorize the values: first, based on what kind of mathematical operations (comparison, addition, ...) the particular data type permits; and second, based on the possible values data can take.

1.1.1 Measures: Possible Mathematical Operations

The values are often categorized according to their *measure level*, namely *nominal* (no comparison possible), *ordinal* (comparison possible, but cannot compute difference), *interval* (can compute difference but not ratio), and *ratio* (can compute ratio too). Below we discuss these (and a few other) measure in a more detail.

Before nominal Nominal measures are usually exemplified with unique labels. However, there are important classes of data where such labeling does not make much sense. This includes text and images. Labeling all texts or images uniquely means not to label category but the text or image itself, so a single different letter or a single different pixel will result in a different label. While we can do this, such labeling is usually not helpful. We may use such approach if we are looking at very short standard texts (say, a review text is “wonderful”), but otherwise almost all texts and images are unique, and labeling will not help us to do any useful analysis. We have to use different tools for, e.g. categorizing images or extract the sentiment of the texts.

Nominal measures In many cases we have a limited number of different categories (and we can always lump too small categories into an “other” category). Such categories often do not follow any inherent order, and hence are not comparable (in a sense as smaller/larger, better/worse, ...). Examples include gender; name of the college attended; and membership of political parties. In such cases there are only limited number of mathematical operations possible:

- comparison: we can tell if two cases are equal
- mode: we can tell which category is most common.

But usually we cannot tell which case “precedes” another in any meaningful sense.

Ordinal measures Another large set of values are categorical with an inherent order, it is always possible to tell that one case is “larger” or “smaller” of another case (or maybe they are equal). Examples include various opinion questions like “do you support the president” with the answers ranging from 1 (not at all) to 5 (very much support); continuous values measured in brackets like income categories (0-10k, 10k-30k, 30k-60k, ...).

One can use ordinal measures for all operations as nominal measures, but now we can also compare the cases: which case is “smaller” or “larger”. This, in turn, allows us to order the observations, and compute the median (the middle value), and other sample quantiles.

However, the difference of such values carries little meaning. Sometimes the categories carry numeric label (like the opinion about president’s performance) but the difference between these numbers may be hard to interpret. There is no guarantee that the difference in the feeling about the president between strong and weak opponents (values 1 and 2) is similar as between weak and strong supporters (values 4 and 5).¹

Interval measures These are numeric values that are comparable like ordinal measures, but where also the differences are meaningful. The examples include temperature (in both degrees of C or F) and GPA.

In case of temperature we can, for instance, say that 2019 global temperature was 0.98C above the temperature of 1951-1980 base period, and that of 2001 was 0.54C

¹ Although, strictly speaking, one cannot compute the differences, it is fairly common in practice when comparing different samples. For instance, one may find that the average support is 4.0 among those without college degree and 3.5 among the college graduates. Such averages are handy for a quick comparison of groups.

above the same baseline.² Even more, these two figures, 0.98 and 0.54 are directly comparable, so we can say that the temperature anomaly in 2020 was 1.81 times larger. However, the temperature values in this sense are not comparable in the same way. The baseline temperature over this period was approximately 14C, and hence the corresponding values were 14.54 and 14.98C. Now it carries little meaning to say that the temperature in 2019 was 1.03 times larger than that of 2001. Celsius scale is based on the freezing point of water, and from the climate perspective, it is an arbitrary reference point.

With interval measures we can do all the operations as with ordinal ordinal measures, and one can also subtract and add two interval values. This allows to compute a number of common statistical measures, including mean, standard deviation, and variance.

Ratio measures These are numerical quantities that have well-defined zero. This includes various physical measures like height or area, age, income (in money, not in income categories) and the like. In case of ratio measures one can claim that one house is “twice as large” as the other house, or that the tree is “three times older than me”. Note that while ratio measures require the existence of a well-defined zero, they do not require any objects actually to be of measure 0. For instance, in case of human height, height 0 is very well defined despite of no human ever being of zero height.

A special ratio-related measure is *percent*. By definition, this is a proportion and hence requires a ratio-type measure. For instance, if elevation of Mount Adams is 3,743 m, and that of Mount Hood 3,429 m, then Adams is $(3,743 - 3,429)/3,429 = 1.092$ times higher than Hood. This is 9.2% difference. Note that we have used Hood’s elevation as the base here, related to the expression “...higher than Hood”. Alternatively, we can also use Adams as the baseline: Hood is $3,429/3,743 = 0.9161$ times higher, i.e. 8.389% shorter than Adams.

A measure closely related to percents is *percentage points*. This is difference between two values, expressed in percentages. For instance, ECB deposit interest rate at the end of 2008 was 2.00 percent and refinancing rate was 2.50 percent. The difference between these rates was 0.5 *percentage point*. One can also say that refinancing rate was 25% higher than deposit rate as percent measures are ratios. However, such percent-of-percent figures are rarely used as this is a perfect source of confusion.

Table 1.1 summarizes the measures and some of the related descriptive statistics.

One should also be aware that the boundaries between the measure types may not be quite clear cut. As soon as one uses numeric labels for a variable, one can do all mathematical operations with these data. The question is whether the result of such operations have any applications. For instance, imagine financial data that contains a student status variable with two potential values *student* and *not student*. These are clearly nominal variables. But we can label “student” as 1 and “not student” as 2. These two numeric labels are as arbitrary as any other labels, but because they are numbers, we can still perform mathematical operations with these, e.g. compute the mean. The result, most likely a number between 1 and 2, will not carry much meaning



What is the coldest place in the universe?

By David Crookes about 5 hours ago

REFERENCE The coldest place in the universe is a teeth-chattering -459.67 degrees Fahrenheit: over three times icier than the chilliest location on Earth.

²NASA data

Numbers are sometimes used in a way that does not correspond to their measure levels. The claim that the coldest place in Universe is “three times icier” than Earth depends on the temperature scale. In Fahrenheits, the temperature ratio is $-458F / -136F \approx 3.4$ (space/earth correspondingly). In Celsius scale it is

Table 1.1: Quantitative measures and associated statistical operations

measure	operations	plots	examples
nominal	equality, count categories	(unordered) histogram	bicycle brands
ordinal	+ greater/smaller	(ordered) histogram, median	income categories
interval	+ add/subtract: mean, variance, standard deviation		temperature, IQ, GPA
ratio	+ divide		length, height, income

if applied to a particular person. However, it is a good descriptor of “studentness” of the dataset, i.e. what is the percentage of students or non-students in data. In a similar fashion, one can assign sequential numeric codes to ordinal measures, such as language skills, and compute the average or the standard deviation. This average by itself does not carry much meaning but is useful for comparing different samples.

Exercise 1.1: Of what measure type are these values?

Are these nominal, ordinal, interval, or ratio measures?

- Talent show result (e.g. first place, second place, 10th place...)
- Height in cm
- Height in feet, inches
- Colors by name
- Temperature in C
- IMDB movie ratings (on scale 1-10)

Solution on page [415](#).

1.1.2 Values: Which Values Are the Possible

While measures describe the nature of data from the mathematical operations’ point of view, they do not explain what kind of values are valid. When analyzing actual datasets one may encounter various invalid numbers, e.g. negative age, or percentage that is outside of $[0, 100]$ range. It is important to see if the values are valid when working with a new dataset.

Here we describe a few common types of values, and what kind of problems to look when working with that type data. It is a non-exhaustive list.

Discrete labels Quite often the values must belong to a pre-determined set of discrete labels. These are often nominal measures but they do not have to be nominal.

Example: students’ major must belong to a set of all majors offered in the college. Now if you notice that there is someone who is majoring in “witchcraft” then something must be wrong with your data (or maybe with the college).

In particular, you should look for empty strings, and labels like “NA”, “N/A” and similar. Such labels are frequently used in manually created datasets.

Counts Counts must be non-negative integers. Any other value is clearly erroneous.

Example: number of children in a family. Values like -1 or 2.7 are clearly impossible. But before just throwing out such values you should consult the documentation. Negative numbers are often used to denote various types of missing values (e.g. “-1” may mean “does not want to tell”). In a different type of data 2.7 children may mean a certain average value.

Continuous positive measures Certain values can only be non-negative. For instance, length or light intensity can take any fractional value but they must be non-negative.

But again, what constitutes “light intensity” in the dataset may not be that simple. E.g. VIIRS night light data applies complex processing to remove effects of sunlight, moonlight, lightning, fires, northern lights, snow reflections, drifting satellite orbits and instruments... ([Elvidge et al., 2017](#)). As a result, the light intensity values that the dataset includes can be negative.

Other limited values There are a plethora of other possible limitations. Some figures must fall in a certain range, for instance unemployment rate, defined as fraction of workforce out of work, must be between 0 and 1 by construction. In a similar fashion, percentages typically must fall in the [0,100] interval. But not always—for instance, an airplane can fly at 105% design speed.

1.2 Doing Descriptive Statistics

Descriptive statistics is largely data description. It serves several purposes, including to familiarize the analyst and the reader with the data, and to provide an easy overview of the traits in the data that are central for the current analysis. Descriptive statistics is also a useful way to summarize a huge amount of data into a few manageable figures.

1.2.1 Data Quality

In many cases it may not be necessary to inform the reader or to provide a summary of your data. But before ever using data, one should have an idea of its value. Can these data be trusted? What are the main traits and the main problems there? Do simple results on these data make sense?

There are many important questions we may want to answer using descriptive methods. Some of the answers can also be obtained from the documentation, but well-documented datasets are unfortunately a rare species. Moreover—even if high-quality documentation exists, we cannot be sure that the data actually correspond to what is stated there. The latter may be outdated, or the way the variables are encoded may have been changed later, protocols may have been violated, and there

may just be human errors. This is another reason why we may want to begin with descriptive analysis when working with a new dataset. Our analysis should address the following question:

- How is data collected? We cannot assess external validity of the results without knowing the sampling procedure. Ideally, the data represents the population under study well.

However, even if the authors were striving toward a particular sampling scheme, they may not have achieved this for various reasons. It is a common practice to test new data by calculating a selection of well-known results, e.g. relative population by region in case of a geographic dataset. In case of a representative dataset, these results should be close to what we can know from census or other sources.

See more in [Section 1.2.3 Sampling](#).

- Which variables/information are in data and how these are encoded? A good starting point may be to start from the documentation. There are well-documented or easy-to-understand datasets but too often one has to rely on just eyeballing the numbers and doing guesswork based on the variable names. As good documentation requires a lot of work, it is often skipped over, and you as an analyst will suffer as a result.
- Does it contain information you need? If we are interested in the relationship between income and education, it is not just enough to have a dataset that contains “income” and “education”. Both of the variables may be coded in a way that is not informative for our purpose. Imagine the case where we want to say something about how income is related to college degree, but the dataset only tells if someone has income or not.
- Missings and implausible entries. The variables we are interested may also suffer from many missing values, and from implausible entries. For instance, what should one do with negative income? Or with negative age?
- How are values coded? Sometimes it may be obvious, e.g. if variable *age* values fall between 18 and 81 then it is probably age in years. But if variable *sex* has values 1 and 2? Or maybe 1, 2, 9? Unless there is suitable documentation, it may not be possible to deduce the meaning of these values with certainty.

Understanding all such nuances is a substantial work, and in case of large datasets researchers usually try to stay within the realm of data they know.

1.2.2 Describing Data

When we have satisfactory answers to the questions above, we may want to get a quick overview of the content of data itself. This mainly serves the purpose of getting a broad understanding of the processes we are interested in, it may also be useful for assessing the informational content of data (we don’t need much data—we need much information!).

Here we describe three traits in data: *central tendency*, such as mean; *variability*, such as range and variance; and *distributions* in the form of histograms and other broad measures.

Central Tendency

One of the crucial bit of information about data is where are the data points located. And we may want to summarize the location with just a single number. Mean and median are the most popular such numbers.

Mean Mean is the most popular way to describe the location (the “center”) of the data. For N observations x_1, \dots, x_N it is defined as

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i. \quad (1.2.1)$$

Mean is very intuitive measure and humans have good innate abilities to estimate mean value by just looking at the sample.

Computing mean requires the data to be of interval measure, otherwise addition is not defined. However, also often sloppily applied to ordered measures when comparing distributions. In that case mean is, strictly speaking, not a central tendency measure but just a test statistic we are comparing across distributions.

The main disadvantage of mean is that it is sensitive to outliers and missing values. For instance, consider data 2.1, 2.2, 2.3. Its mean is $\bar{x} = 2.2$. However, if in case of a data entry error we have 2.1, 2.2, 23 instead, the mean will be 9.1. If any data point is missing then mean cannot be computed at all. Mean is not a *robust statistic*.

Mean may correspond to a non-existing or even impossible case. For instance, we may find that an average family has 0.5 children. One cannot conclude from this number that industry should supply more kids “half-beds”. But in order to evaluate the demand for daycare or school places, this number is very much applicable.

Mean is a good description for data that is fairly concentrated. For instance, if all employees have income between 40,000 and 60,000, the mean would describe all these salaries fairly well. But if our sample contains 100 people in poverty (income 10,000) and one billionaire (income 1,000,000,000), the average (9,900,000) does not describe the incomes well. One may wrongly assume that everyone in this sample has income around 10M, and hence poverty is not an issue.

Mean is the sample analog of [expected value](#) (see Section 1.3.5).

Median Median is another popular measure of data location. Median is the “middle value”, a value where there is an equal number larger and smaller values in the data. For instance, in a dataset 1, 2, 3, 4, 10, median is 3 as there are two smaller and two larger values. If there is no such datapoint, e.g. in a sample 1, 2, 3, 4, the median can be defined in different ways, one encounters values 2, 2.5 and 3.

Median is much less sensitive to outliers than mean. If we take the example above where instead of 2.1, 2.2, 2.3 we observe 2.1, 2.2, 23 due to a data entry error, we can see that the error leaves median, 2.2, unchanged. Median is a robust statistic. Median

is also less affected of missing values. For instance, consider the same data but now assume the last observation is not wrong but missing: $2.1, 2.2, NA$. While we cannot say anything about the mean, we can still say that $2.1 \leq median \leq 2.2$: if the missing value is larger than 2.2 then median is 2.2, if it is smaller than 2.1 then median is 2.1, and if it is somewhere in between, then the unknown value is also the median. We are not quite sure about the median value but in this example we can give fairly narrow bounds. Computing median involves just comparison and no addition as in case of mean. So it can be computed on ordered measures, interval properties are not needed.

Median describes well “typical” values in data but fails to capture information about “non-typical” values. For instance, in case of the poverty-billionaire example, the median income will be 10,000. The median person is in poverty. However, median does not provide any hint about the fact that we also have a billionaire in the sample. In a similar fashion, if we find that median household does not have any children, we cannot conclude that no household have any kids. In order to design family policies we have to incorporate other values than median.

Mode The third popular measure of data location is mode. Mode is just the most common data value. For instance, if data looks like $1, 2, 2, 3, 3, 3$, the mode is 3. Computing mode only requires comparing equality, so mode is well defined even for nominal measures. However, mode may not work well for continuous values. In case of discrete outcomes, there is only a limited set of possible values, but in continuous case it is unlikely that many data points have exactly the same value. Computing mode for continuous variables typically includes some sort of smoothing, and thereafter finding the maxima of the smoothed values.

Many types of data are *unimodal*, i.e. they have a single mode, often around the middle of the values (given the data is ordered). Other types of data are *bimodal* or *multimodal*, i.e. they have two or more different values that are most common. Normally one talks about bimodal distribution even if the two modes do not have the exact same frequency, but are clearly separated with less common values (see Distributions below).

Exercise 1.2: Mean, median, mode

Consider data $\mathbf{x} = (1, 2, 3, 3, 3, 5, 5, 10)$. Compute

1. mean
2. median
3. mode

Now assume the first observation is missing: $\mathbf{x} = (NA, 2, 3, 3, 3, 5, 5, 10)$. What can you tell about mean, median and mode?

Variability

While humans have very good intuitive idea of mean, our understanding of variability is not as good. We can understand the concept of range fairly well, but variance and standard deviation are much harder to grasp.

Range TBD: ends must be dense

Sample variance Variance and standard deviation are measures of variability that are extremely important from the theoretical point of view. Many common statistical tests, including t -test, are based on these values.

Sample variance is defined as average squared deviation from sample mean:

$$s^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2 \quad (1.2.2)$$

where \bar{x} is the sample mean. This is not an intuitive measure but we can think of variance as a “typical” squared deviation from the mean. Sample variance does not have standard notation but s^2 is a fairly common notation.

Variance has a number of advantages over just data range:

- Variance is much less sensitive to lone outliers. Although it is still sensitive—the definition involves the deviation squared, so large deviations have overly large influence—it also includes an average over all other data points. So variance is “made” of all data, not just of two extreme observations.
- Variance, in particular its [analogue for random variables](#) (see Section 1.3.5), is an extremely important theoretical concept.

There are two main disadvantages of variance: it is not an intuitive concept despite of its theoretical importance, and second, it is measured in squared units. For instance, if we are working with human age data, variance is measured in years squared. This is not a unit that we understand. Fortunately, this problem is easy to ameliorate. We can just take square root of variance, measured in the same units as data. This is called *standard deviation* or *standard error*. The difference between these two concepts is mostly semantic, they are equivalent descriptions of variability. The notion “standard deviation” is used primarily in the context where we talk about variability in data, “standard error” in a context where the variability describes uncertainty in our results. Standard deviation is denoted in various ways, in formulas it is often s (as square root of sample variance s^2), in text and tables it is often written as *std.dev* and *std.err*.

Variance can be computed using definition (1.2.2) above. Let us compute variance for data $(1, 2, 3)$. We do this by constructing a table for the auxiliary results (Table 1.2). The first column in the table just displays the data, average of which, \bar{x} is in the last row. In the second column we compute the deviation from the average, and the last column displays the deviation squared. The average of the latter is variance, in this example $s^2 = 2/3$. Note also that the middle column, $x - \bar{x}$, averages to 0. This is always true by the definition of mean, and explains why we square the deviations in the definition of variance.

Table 1.2: Computing variance. The last row displays the averages, the of those is just the sample average $\bar{x} = 2$, and the last one is variance s^2 . Note that the average of the middle column is 0. This is always true through the definition of mean.

x	$x - \bar{x}$	$(x - \bar{x})^2$
1	-1	1
2	0	0
3	1	1
average	2	0
		2/3

This approach, based on the definition (1.2.2) is easy when coding, when computing variance manually, it is easier to use the shortcut formula

$$s^2 = \bar{x}^2 - (\bar{x})^2, \quad (1.2.3)$$

the difference between mean of x^2 and the square of mean x . For the example in Table 1.2, we see immediately that $(\bar{x})^2 = 4$, and we can easily find that $\bar{x}^2 = (1+4+9)/3 = 4\frac{2}{3}$, and hence their difference is $s^2 = 2/3$, the same number we found when using the definition (1.2.2).

Proof 1.2.1: Where the shortcut formula is coming from

The shortcut formula can be derived from the definition of variance (1.2.2). We start by opening the parenthesis and re-arranging the terms:

$$\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2 = \frac{1}{N} \sum_{i=1}^N (x_i^2 - 2x_i \bar{x} + \bar{x}^2) = \quad (1.2.4)$$

$$= \frac{1}{N} \sum_{i=1}^N x_i^2 + \frac{1}{N} \sum_{i=1}^N (-2\bar{x} x_i + \bar{x}^2) = \dots \quad (1.2.5)$$

Here we already have the first term, \bar{x}^2 . Now we use the fact that \bar{x} does not depend on i can be take out of the sum:

$$\dots = \bar{x}^2 - 2\bar{x} \frac{1}{N} \sum_{i=1}^N x_i + \frac{1}{N} N \bar{x}^2 = \dots \quad (1.2.6)$$

Here we have used the fact that as $\sum_{i=1}^N \bar{x} = N \bar{x}$. And finally, using the definition of mean we have

$$\dots = \bar{x}^2 - 2\bar{x} \bar{x} + \bar{x}^2 = \quad (1.2.7)$$

$$= \bar{x}^2 - \bar{x}^2. \quad (1.2.8)$$

This is the shortcut formula.

TBD: Exercise, properties: multiply by a constant

It also appears that the results computed from (1.2.2) on small sample tend to underestimate the variance on a larger sample of the same data. This can be easily understood when looking at a sample of a single observation only. Obviously, in this case $\bar{x} = x_1$ and hence $(x_1 - \bar{x})^2 = 0$ and we have the sample variance $s^2 = 0$. This is definitely an underestimate for anything besides constant values. The solution is to compute the corrected variance, often called *population variance*

$$s^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2. \quad (1.2.9)$$

The difference is the value $N - 1$ instead of N in the denominator. This inflates the sample variance estimator, and now the estimates are not too small even on small data. For instance, in our single observation example, this formula would give 0/0, an undefined value.

The bias correction, using $N - 1$ instead of N when computing the average, is related to the fact that we typically compute the average \bar{x} on the same data. This removes one degree of freedom from data, and hence we use $N - 1$, not N nor $N - 2$ when computing variance. One can intuitively see that data where the mean is “extracted” is “poorer” than the original data. Its average must now be 0, and hence if you know the values of $N - 1$ data points, you can immediately compute the value of the N -th data point. Only $N - 1$ data points are “free”, hence the degrees of freedom is $N - 1$. But if we compute \bar{x} through other means, not from the same data, the unbiased variance should be computed using N , not $N - 1$. See more in Section 1.3.5.

The two variance concepts, sample variance and population variance, are a source of a lot of confusion. Fortunately the difference in anything resembling a dataset of a reasonable size is minimal.

An additional source of confusion is caused by the theoretical concept corresponding to population variance that is also called **variance** (see Section 1.3.5). These concepts are related in a similar way as mean and expected value, but unfortunately they share the same name.

Distribution

Mean, range and variance and other descriptive figures give a few numbers that are useful in understanding the typical and extreme data values. But often we also want a visual representation about how common are certain data values. The popular tools are histograms and density plots.

Histogram *Histograms* are just counts of data points in bins of different values. Typically the variable range is split into equal amount of bins, and thereafter one counts how many observations fall into each bin. It is also common to present the histogram not as counts per bin, but as density, i.e. percentage of observations per unit width for each bin. The advantage of this is that the numeric values are approximately constant when changing the data size and number of bins. In practice, a good number of bins is square root of the number of data points.

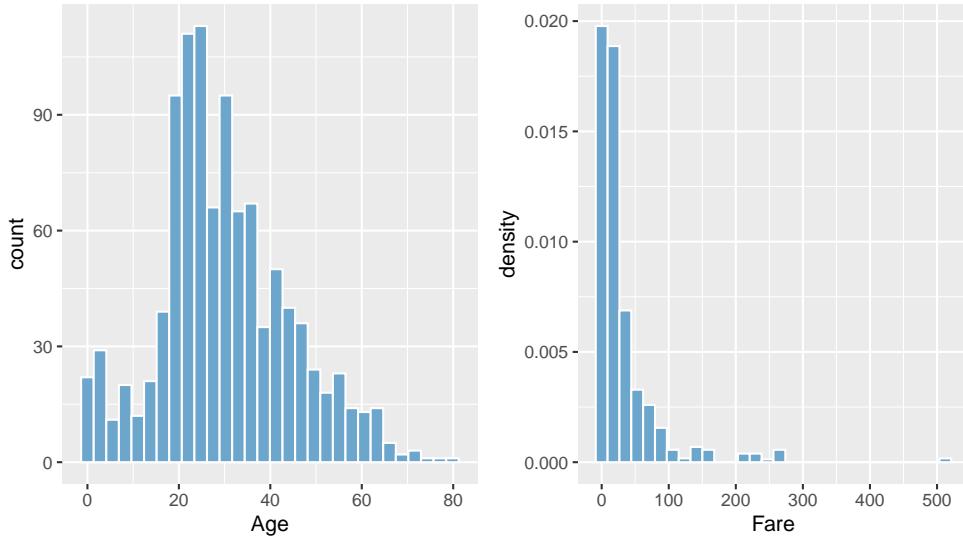


Figure 1.1: Histogram describing the age distribution of Titanic passengers (left panel) displayed as counts, and the distribution of fare they paid (right panel) as density. We can see that while age is approximately normal, fare is highly skewed.

Figure 1.1 depicts such histograms for age (left panel) and fare (right panel) of Titanic passengers. We can immediately see that these two variables are distributed in a rather different way. Age is broadly normally distributed while fare is extremely right-skewed: most people paid around 10£ (in fact, the highest fare paid was 512£) but a few passengers paid much more. We can also see that the age distribution is bi-modal: typical passengers were 20 to 40 years old, by we also see a peak among young children. These are probably emigrants' children, traveling with their parents.

So histograms allow us to quickly grasp several interesting features of the distribution, and in this sense they offer a much more detailed view than mean or variance. However, just by eyeballing we may not be able to estimate certain relevant features of the distributions, e.g. we may not be able to tell if mean of one sample differs from the mean of another sample.

TBD: Density plots, violin plots

Quantiles A popular method to quantify certain aspects of distributions is by using quantiles. *Quantile* is relative location in data. For instance, 0.2-th quantile is such a number that 20% of observations are smaller than this number, and 80% of observations are larger than this. There is a sibling measure of quantile, called *percentile*. These two are equivalent, 0.2-th quantile is exactly the same thing as 20th percentile. In order to compute 0.2-th quantile (20th percentile), we can first put data in an increasing order, and thereafter remove the first 20% of it. The smallest number that is left is the quantile value.

In practice we have to define it slightly differently to deal with repeated observations:

Definition 1 (Sample quantile). τ -th quantile of a sample is a number q , such that fraction τ of cases is no larger than q , and fraction $1 - \tau$ of cases are no smaller than q .

For example, consider sample $(1, 1, 1, 2)$. Its 0.5th quantile is 1: a half of the cases (1 and 1) are no larger than 1; and the other half (1 and 2) are no smaller than 1.

However, this definition is still not unique. For instance, 0.5-th quantile of $(1, 2)$ can be anything in the interval $[1, 2]$. Usually this does not matter in practice, but one must be aware of possible surprises, in particular if data contains a small number of discrete values. Also, different software packages may define quantile differently, in the case of this example you may find numbers like 1, 1.5 and 2, depending on what is the exact implementation.

Certain basic quantiles have separate names:

- *Median* is 0.5-th quantile: it is the middle value, i.e. a half of the sample is no larger than median, and the other half is no smaller than median.
- *Tertiles* (or *terciles*) are 1/3 and 2/3-th quantiles
- *quartiles* are 1/4, 1/2 and 3/4-th quantiles
- *quintiles* are quantiles that split data into five parts (0.2, 0.4, 0.6, 0.8-th).
- *deciles* split data into 10 equal parts.

Quantiles that are close to median are fairly robust with respect to outliers, but extreme quantiles (such as 0.01-th or 0.999-th) may be very sensitive.

Exercise 1.3: Compute Sample Quantiles

Consider data $(1, 2, 3, 1, 2, 1)$. Compute:

1. 0.5-th quantile (median)
2. 0.25-th quantile (lower quartile)
3. 0.8-th quantile (upper quintile)

Solution on page [416](#).

Other descriptive measures

Inequality Another common feature we analyze in data is inequality. There are various ways to measure it, e.g. by Gini coefficient, the quintile share ratio, Pareto ratio, and many others. Note that inequality is only well defined for ratio measures, this is because when comparing inequality, we almost invariably talk in relative terms. For instance, very feel that \$100,000 difference in income describes we different inequality for two persons earning \$50,000 and \$150,000, compared to two persons earning \$1,050,000 and \$1,150,000. This is why we need ratio measures to discuss inequality.

A number of inequality measures also cannot handle negative and zero values: it is true that someone owning \$10 owns infinitely more in relative terms than someone with no money, but such ratios are typically not useful for any practical applications.

Ratio measure has well-defined zero. See [Section 1.1.1 Measures: Possible Mathematical Operations](#), page 2.

Below, we look at two measure—*quintile share ratio* and *Pareto rule*.

Quintile share ratio Quintile Share Ratio (*QSR*, also $S80/S20$) is a popular and simple inequality measure. It is the ratio of the total wealth owned by the wealthiest 20% to the total wealth owned by the poorest 20%. It can be computed as a sum of all values above the 0.8-th percentile, divided by the sum of values below 0.2-th percentile.

Obviously, we can compute QSR for all sorts of different variables, not just wealth. For instance, look at the house prices in Windsor,³ the distribution is shown in Figure 1.10, page 56. The average house in the dataset costs \$68,100, and all in all, we have data about 546 homes. The bottom 20th percentile of the house values is \$47,000 and the top 20th percentile is \$87,000. The total value in the bottom quintile, the total price of all 107 homes with price below \$47,000 is \$4,145,400. The total value of the top quintile, the total value of 109 homes above \$87,000 is \$12,021,700. Hence the quintile share ratio

$$QSR = \frac{\$12,021,700}{\$4,145,400} \approx 2.9. \quad (1.2.10)$$

For the house values example, QSR is well defined and easy to understand. This is because in normal housing market, all houses command a positive price. But certain distributions have a large number of zeros. If the distribution contains more than 20% of zero values, the total value in the bottom quintile is zero and QSR is infinite. It carries little information in such case. Unfortunately, zero values are very common in all kinds of income, wealth, and popularity data. For instance, those who do not work have no income. Those who do not own a home have zero housing wealth. Websites that are not accessed have zero number of hits. In that case one may compute SQR for the positive values only, but this approach ignores the presence of zeros in data.

The problem boils down to a conceptual issue—we want to use the inequality measure to describe life quality difference between different groups of people. But no income does not mean zero life quality, most of zero-income people are either drawing down their own savings, or so are supported by others. Income is only a proxy for life quality.

Pareto ratio Pareto ratio is another popular measure of inequality. It is often called 80/20 ratio after the observation that for many phenomena, 20% of the cases are responsible for 80% of the outcomes. This includes wealth inequality (“20% of the richest control 80% of the wealth”), but also computer code (“You spend 20% of time to get your code to work on 80% of data, and you spend 80% of your time on the last 20% of cases...”).

The exact figure depends on the data distribution, and despite we may call 80/20 ratio, it is not usually the case that the top 20% controls 80% of all outcomes. If this is indeed the case, i.e. if the richest 20% control 80% of total wealth, then we have a rather unequal distribution. For instance, in case of Windsor housing wealth example above, the most expensive 43% of the houses contain 57% of the housing value instead (see Figure 1.2); but in case of research paper citations, the most cited

³Data “Housing” in R package *Ecdat*

17.5% of papers capture roughly 82.5% of all citations. So among these datasets, housing values are much more equal than citations. See also Table 1.7 for related ratios for log-normal distribution.



Figure 1.2: Windsor housing data (R package Ecdat). Pareto rule states that the top $x\%$ of population possesses as much as the lower $100 - x\%$ of the population. Here the most expensive 43% of houses cost as much as the cheapest 57% houses. Housing value inequality is low in this neighborhood.

Note that by construction, the top 50% will always own at least 50% of the total wealth.

Cheatsheet 1.1: Descriptive Statistics

Central tendency What are the “typical” values.

Mean (average) $\bar{x} = \frac{1}{N} \sum_i x_i$. Need interval measure. Easy and intuitive, good for aggregate data; sensitive to outliers, the value may not exist.

Median middle value: value where half of the sample is smaller than this, and another half is larger than this. Need ordinal measure. Less sensitive to outliers; less intuitive.

Mode most common value. Any measure will do. Intuitive for discrete values, needs assumptions for continuous values.

Variability How are the values spread around.

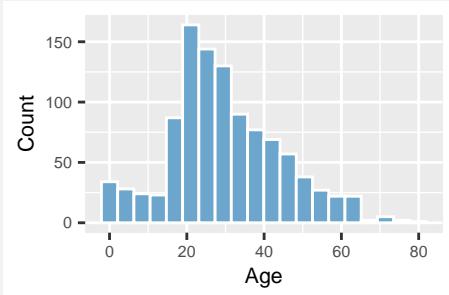
Range minimum and maximum value. Need ordinal measure. Easy and intuitive; extremely sensitive to outliers.

Variance average squared deviation from mean: $s^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2$. Very important theoretical measure; not intuitive, measured in squared units that are hard to interpret.

Standard deviation square root of variance. Measured in the same units as data; less desirable theoretical properties.

Distribution What values are more common and less common.

Histogram count and plot values in pre-determined bins:



Quantile 0.2-th quantile is such a number so that 20% of values are smaller than that, and 80% of values are larger than that.

Inequality

QSR (quintile share ratio) is the ratio of total value of top 20% cases (top quintile) and the bottom 20% of cases (bottom quintile). For instance, *the top 20% of jobs pay 8 times more than bottom 20% of jobs*.

Pareto ratio x so that the largest $x\%$ of cases “owns” $100 - x\%$ of total value. For instance, *the richest 30% own 70% of all wealth*. Note: x does not have to be 20%, the exact value depends on the distribution!

1.2.3 Sampling

Sample and Population

In statistics-related tasks one typically wants to make conclusions about the whole “population” based on a small “sample”. Intuitively, we need a large enough sample, that covers all the important subgroups. Here we call *population* the whole set of cases we want to apply our results to. In contrast, *sample* is the set of observations we have access to. For instance, based on a sample of 1000 voters, a consultant may make claims about the election outcome, decided by the population of all voters.

Why do we need to consider sampling and sampling designs?

- It is rarely possible to collect data on the complete population. Even if possible, sometimes it is cheaper to collect and analyze a sample, and generalize the results to the population.
- Sometimes it is inherently infeasible to collect data about the whole population. This includes cases where we are concerned about future or the past.
- Other times we can collect the data about “everything” but we still want to generalize our results to even larger populations. For instance, we can easily collect grades of *all* students of a particular course. Do we still need to generalize? This depends on the question:
 - If we are only interested in students of that course then we have the full population. We do not have to consider generalization issues. In this case the sample and the population are the same. Say, if the instructor is concerned that the grades are too low and considers curving those up, there is no need to think about generalization of any kind.
 - If we are interested in “all” students in “similar” courses then our data only represents a sample. For instance, we may be concerned that the course may be hard for students with certain background. This concern is a generic one, also applying to the future students in similar courses. In this case we need to generalize from the sample (students of this course) to the population.

Sampling Process

Getting a sample of data typically involves many steps, some of these deliberate choices, and some caused by external factors, such as data or funding availability.

Sampling contains broadly such steps:

1. Theoretical population. Who (or what) do we want our results to generalize to?
2. Study population. What part of the theoretical population can we access?
3. Sampling frame. What part of the population will be studied? This is the part of the population that is accessible from the practical point of view, i.e. we have information about their presence and location, and can realistically access them. It is often based on proxy information, for instance phone directory when surveying humans, or geographic location when counting wildlife.

4. Sample. What part of the population do you end up getting data for?

Example 1.1: Predicting election results

Look at the sampling process when predicting election results.

1. The population of interest is clearly all actual voters (i.e. those who will cast their vote, not all registered voters), and we are interested in finding their favored candidates.
2. Study population is a list of voters we have records about, either their address, phone, or just the fact that they exist. If the analysts have access to all registered voters records, then the study population will almost overlap with the theoretical population. However, the overlap may still not be perfect, as between now and the election day, more people may register as voters, and some in our records will die (or otherwise leave the records).
3. Sampling frame is a (potential) voter register with contact information. In the best case this is the actual voter register, but it may also be any other accessible proxy, e.g. phone directory, street maps, or lists of public places to visit.
4. Sample. This is taken from the sampling frame, the voters we were able to access and who did answer the questions about their (prospective) voting behavior.

Each step in the sampling process can introduce a corresponding error.

- *External validity* concerns the generalizability of study population to the theoretical population. For instance, are the results collected for current students also valid for future students?
- *Coverage error* are errors, resulting from non-perfect overlap between study population and sampling frame. If many voters do not have phone, we miss those voters.
- *Sampling error* arises from the fact that instead of the whole population, we only work with a small population. Sampling errors can be addressed by increasing the sample size, if feasible. If the sampling process is well known, the errors can also be quantified, and taken into account through confidence intervals .

Sampling bias is a combination of coverage error and external validity problems. Sampling bias can sometimes accounted for if we know the sources of coverage error and validity problems. But full extent of these problems is rarely known and hence the sampling bias is a common issue, and the results may lack external validity.

Sample without replacement:

- Everyone is sampled either 0 or 1 times
- If sampled, you are removed from the “at risk” population

Example: urn with 2 white and 2 black balls. What is the probability to sample 2 black balls?

- The probability to sample 1 black ball is $1/2$ (2 out of 4)
- The probability to sample 2nd black ball is $1/3$ (now 1 out of 3 is left black)
- Hence the answer $1/2 \cdot 1/3 = 1/6$
- Everyone can be sampled 0 or more times
- If sampled, you are put back to the “at risk” population

Example: urn with 2 white and 2 black balls. What is the probability to sample 2 black balls?

- The probability to sample 1 black ball is $1/2$
- The probability to sample 2nd black ball is $1/2$ (still 2 out of 4 is left black).
- Hence the answer $1/2 \cdot 1/2 = 1/4$

1.3 Basics of Probability Theory

This section discusses probability theory, in particular the concepts of *random variable*, *expected value* and *variance*. We use these concepts extensively later in statistics.

1.3.1 Events and Sample Space

Before we get into discussing the concepts in more details, we should make clear *what are we analyzing*. The two central concepts in probability theory are *events* and *probability*.

Event is something that may or may not take place, and where we typically do not know if it occurs. Sure, we can also talk about things that take place for sure (*certain events*) or that will never take place (*impossible events*), but we do not really need the concept of probability to analyze such cases. A few examples of events we may be interested in include

- Flip a coin. An event is *get heads*;
- We play a dice game and roll two dice. We may be interested in an event *get at least one six*;
- We are going to pick up a friend at airport. We are concerned about the event *flight arrives in time*.

When talking about events in the probability theory sense, we are always thinking about some kind of *stochastic* phenomenon, or in a stochastic experiment. *Stochastic* refers to phenomena that are not completely predictable, at least not in terms of the information and tools that we have at our disposal. For instance, your friend's arrival time may be very well predictable if we know the exact position and speed of the airplane, the wait time at immigration, and whether all the luggage bands at the airport are working. But as we don't have this information, we may just go to the airport in time and hope for the best. Arrival time is a stochastic process from our perspective.

All possible events together form *sample space* \mathcal{S} . So sample space is a set of all kind of events that can occur in the phenomenon we are considering. Although the concept may feel trivial, it is extremely helpful when thinking about random outcomes. Here are a few examples:

- Toss a coin. There are only two options, heads and tails, so $\mathcal{S} = \{H, T\}$
- Roll two dice. Each die can come up with sides 1 to 6, so the sample space is a set of tuples (ordered pairs)

$$\mathcal{S} = \left\{ \begin{array}{l} (1,1), (1,2), \dots, (1,6) \\ (2,1), (2,2), \dots, (2,6) \\ \dots \\ (6,1), (6,2), \dots, (6,6) \end{array} \right\}.$$



Flipping a coin is a popular way to create random outcomes—heads or tails. Historically, one side of the coin frequently represented the head of the monarch, the tail side depicted other symbols of power. Five roubles in gold, Nicholas II of Russia. By Unwrecker, CC BY-SA 3.0, via Wikimedia Commons

Note that we distinguish $(1, 6)$ and $(6, 1)$, i.e. we distinguish between the first and second die: in the first event the first die comes out with one and the second with six, in the second event it is the way around. These two simple events make a compound event *one and six* (see below). If both dies are similar and hard to distinguish, then we may consider these two events to be a simple event instead.

- Flight delay. This can be any number, and we cannot really put a lower or upper limit on it in general, so we can consider the sample space to be

$$\mathcal{S} = (-\infty, \infty)$$

The first two of these examples are finite discrete sample spaces. The third one is a continuous sample space. Note also that the first two are not numeric: when tossing coins, we receive heads and tails, not numbers. When rolling two dice, we receive pairs of numbers, not numbers. (Or, to be even more precise, we receive pairs of sides with a certain dot patterns on them.) Finally, the third example, the flight delay, is numeric, but not just numeric as it also has a unit (say, minutes). This is because these events describe the physical world.

Example 1.2: Monty Hall Problem

The concept of sample space allows us to analyze and understand certain problems that are otherwise hard to grasp. Monty Hall problem, a game in a TV-show hosted by TV-host Monty Hall, is the following:

You are in a room with three closed doors. You know that behind one of the doors is the price, and the other two doors are empty. The host knows where is the price but you do not know. You pick one door (but do not open it). Now the host opens one of the other two doors, one that is empty. Now you can either stay at your current door, or switch to the other closed door. Finally, the door you chose is opened, and if you picked the correct door, you'll win the price.

Should you switch the door after the host opened an empty door?

To a big surprise for most of us, including trained mathematicians, it is worthwhile to switch. This will increase the chance of winning from $1/3$ to $2/3$. Why such a counter-intuitive result?

The problem is easy to assess when using the concept of sample space. Let's label the doors 1, 2, and 3, and assume (without loss of generality) that the price is behind the door 1 (see the figure below). If you first pick door 1, the host will open either door 2 or 3, and importantly, you should stay where you are. However, because you don't know where the price is, you pick the correct door only $1/3$ of time; and hence $1/3$ is your winning chance if you stay where you are. However, if you pick a wrong door, for instance 2, the host has only one option to open an empty door, namely 3. Now you should switch to door 1. You do not know if you initially picked an empty door, but just by chance this happens $2/3$ of time. So the second strategy gives you a win in $2/3$ of cases, the former strategy in $1/3$ of cases.

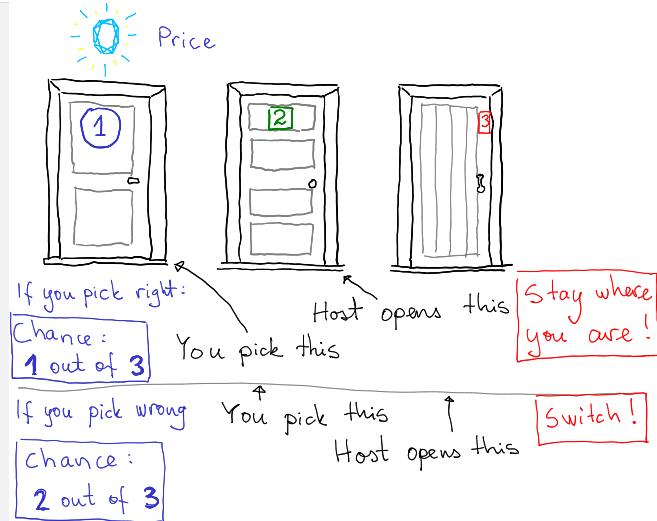


Figure 1.3: Monty Hall problem: you should stay at the door you picked if you picked the right one. But that happens only $1/3$ of time. $2/3$ of time you should switch.

It is important to understand the role of the host. The host is not acting randomly but instead she modifies the setup in a precise way. When you pick a door, there is $1/3$ probability that the price is behind your door, and $2/3$ chance that it is behind another door. When the host opens the other door, it is still $1/3$ probability that the price is behind your door. But now all the rest of $2/3$ probability is concentrated behind the other closed door. In order to make this point more clear, you can imagine a similar game with 100 doors. Again, you start by choosing one door and thereafter the host open 98 other doors so that only two doors remain closed. As your initial guess was correct only $1/100$ of time, the price is most likely behind the other one.

It is often useful to distinguish between *simple events* and *compound events*. Simple events are such events that cannot be partitioned into anything simpler, while compound events can be partitioned. As an example, event of *heads in a coin flip* cannot be divided into anything more basic. It is a simple event. But *get a six when rolling two dice* can be any of $(1, 6)$, $(2, 6)$, $(6, 6)$ or a number of other possibilities. It is a compound event. In a similar fashion, *plane arrives in time* may mean it arrived *exactly* in time, or in colloquial language it may also have arrived (exactly) 2 minutes early. In its colloquial meaning it is a compound event.

Such distinction is often very useful when we compute the corresponding probabilities. It is typically easier to compute probabilities of simple events than of compound events. For instance, consider an experiment: *Flip two coins. What is the probability to get exactly one head?* When working with simple events, the sample space $\mathcal{S} = \{(H, H), (T, H), (H, T), (T, T)\}$. Importantly, as the coins are independent, all these four events are equally likely (with probability $1/4$). Our compound event of interest, exactly one head, is made of two mutually exclusive simple events (H, T)

and (T, H) . Hence the probability of this compound event is $1/2$.

Exercise 1.4: Rolling two dice

Take the example of rolling two dice. Compute the probability of the compound event *get at least one six*.

Hint: sketch the sample space in simple events. Are these events equally likely? Which simple events constitute the compound event of interest?

Solution on page [416](#).

Another important concept is *mutually exclusive events*. It is fairly easy to understand—events are mutually exclusive if they cannot occur at the same time. For instance, sides 1 and 2 cannot occur in the same experiment when rolling a single die. However event “an even side” and “2” can occur at the same time and hence these are not mutually exclusive events. All simple events are mutually exclusive.

1.3.2 Probability

Now we have discussed the events. But probability theory is concerned about *probability* of events. What is probability? It turns out that it is not quite obvious. There are at least two different answers.

The easiest answer to understand the concept is related to repeated events. Probability is “tendency” of the event to occur if we repeat the experiment many times. For instance, when tossing a fair coin 100 times, we will get around 50 heads, i.e. in average, we get heads in approximately 50% of cases. One can imagine that the average number of heads gets close to the true probability when we increase the number of experiments without limit. This concept is called *frequentist probability*.

But not all experiments can be repeated a large number of times. For instance, what would be a frequentist answer to the question “what is the probability that there will be a nuclear war with North Korea”? Do you really want to poke mister Kim Young Un 1000 times to see how many times a war breaks out? Even more, there are a plethora of phenomena that can never be repeated. For instance, probability of rain tomorrow. There is only one tomorrow, and in that tomorrow it will either be raining or not. What is the probability here?

In such cases we have to resort to a definition like “tendency for the event to happen given the information we know”. In case of rain tomorrow, the “information we know” may be a weather model. Professional weather models typically contain many random processes, processes that are impractical or impossible to model precisely. But as we know the properties of these processes in the model, we can compute the probability of rain. This concept is related to *propensity probability* and *Bayesian probability*.

In everyday life we perform somewhat similar calculations. For instance, when deciding when to head to the airport to pick up your friend, you may have heard that today flights are an hour late. When you hear this, you may head to the airport only a half an hour late because you “think” that it is “unlikely” the flight is delayed by less than 30 minutes. We do not perform explicit computations but just “feel” what is an appropriate estimate.

1.3.3 Conditional Probability and Bayes Theorem

Prerequisites: events, sample space

TBD: history

Bayes theorem is a rule about computing conditional probabilities—probabilities that something happens, given something else happened. Conditional probabilities play a very important role in statistics and machine learning, in a sense all supervised learning is about computing conditional probabilities. For instance, if you are predicting house prices based on house size, you are asking questions like “what is the probability that this house costs over \$500,000 given its size is 200 m²? ”

In a similar fashion as we introduced events and thereafter probability in Sections 1.3.1 and 1.3.2, we introduce conditional events first and conditional probability thereafter.

Conditional events, Venn diagram, and conditional probability

Let us start with a simple example: you roll a die and you get an even number. What is the probability that you got six? It is fairly obvious that the answer is 1/3: there are only three even numbers (2, 4, 6), they are all equally likely, and hence you get six in one third of the cases.

This example demonstrates all the basics about conditional events. We roll a die. It has full sample space of six events: 1, 2, …, 6. However, we also have a conditioning event, “even number”. It is a compound event containing simple events 0, 2, 4, 6, 8, … . The conditioning event “carves” (*partitions*) the sample space into two parts: one contains the feasible events (here these are even numbers that are possible to get on a die, i.e. 2, 4, 6) and the other partition contains infeasible events (here 1, 3, 5). Now we can work on the feasible partition only, e.g. we compute the probability of interest as one out of three feasible events.

What we did above is the essence of working with conditional events. We partition the sample space into two parts: the feasible partition (conditioning event), and the rest, the infeasible region. Thereafter we only consider what happens in the feasible region, the region where the conditioning event occurred. The infeasible region can essentially be ignored. Below we introduce the idea more formally and provide more complex examples.

The conditioning of sample space is often illustrated using *Venn Diagrams*. Figure 1.4 displays one such Venn diagram. It is just a picture of sample space where we mark the set of events we are interested, and the set of events we are conditioning on. Figure 1.4 depicts the sample space S (the white rectangle) and two events, A and B . In this figure we have to think of both events as compound events, consisting of single points as simple events. The events overlap to a certain extent, i.e. it is possible that both A and B occur. But they do not overlap perfectly, so if A occurs, it is not certain that B will occur and the way around. Finally, as the events do not occupy the full sample space, it is possible that neither will happen.

An example of such two events may be $A = \text{it is raining}$, and $B = \text{the class is canceled}$. Obviously, it is possible that neither of these two events happens (it is not raining and the class is not canceled), so these two events do not make a complete sample

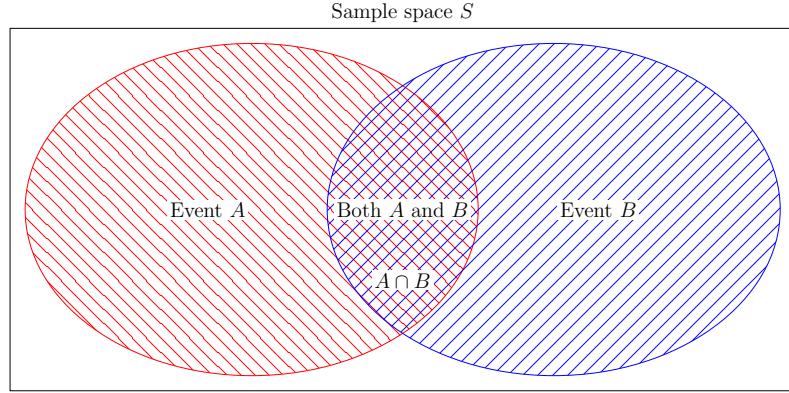


Figure 1.4: Venn diagram. The events A and B overlap partially, so it is possible that only A happens, only B happens, and both A and B happen. As A and B do not sum to the whole sample space (there is a white “leftover area” in the sample space box), it is also possible that neither happens.

space. The “not raining and not canceled” is the white area, surrounding events A and B . Alternatively, it is also possible that only A happens (it is raining but the class takes place), only B happens (it is not raining but the class is canceled), and finally both of these may happen too.

However, in other type of examples not all four options may be possible. For instance, in case of coin toss, heads H and tails T are mutually exclusive events and hence it is not possible that both of these occur simultaneously. Even more, there are no more possible events in the sample space and hence either H or T occurs for sure.

TBD: Exercise: draw a Venn diagram of some sort of either complete event, mutually exclusive events, or maybe where the event of interest is part of the conditioning event.

Conditional probability is basically just probability, computed on the smaller, feasible partition of the sample space that was carved out by the conditioning event. We denote the conditional probability of event A happening given event B happens as $\Pr(A|B)$. For instance, in the house price/house size example, the question can be written as $\Pr(\text{price} > 500,000 | \text{size} = 200)$.

Formally, we denote the probabilities related to the Venn diagram 1.4 as follows. First, $\Pr(A)$ is the probability that event A occurs and $\Pr(B)$ is the probability that event B occurs. These are called *unconditional probabilities* or just probabilities, as these are not related to the other event occurring. Next, we denote by $\Pr(A, B) \equiv \Pr(A \cap B)$ the probability that both A and B occurred. Normally we prefer the shorter notation $\Pr(A, B)$ to denote joint events (this is also common in the literature), but sometimes we want to stress that this is an overlap of A and B and we write $\Pr(A \cap B)$. Finally, we denote the probability that A occurs conditional on B occurring as $\Pr(A|B)$; and the opposite probability, that B occurs given that A happens, as $\Pr(B|A)$. In machine learning context, a major application of conditional probability is predictive modeling. Using statistical tools we are trying to answer the question:

what is the probability of outcome Y given the data X ?

Example 1.3: Red and green, nice and bad

Consider the the following situation: Police is attempting to catch “Bad guys”. But whether someone is good or bad will be clear first after arrest and a costly investigation. But people are of different color, *Red* and *Green*, and the color is immediately visible. For some reason, however, there are more nice guys among Reds and more bad guys among Greens. It can be depicted using the following Venn diagram:

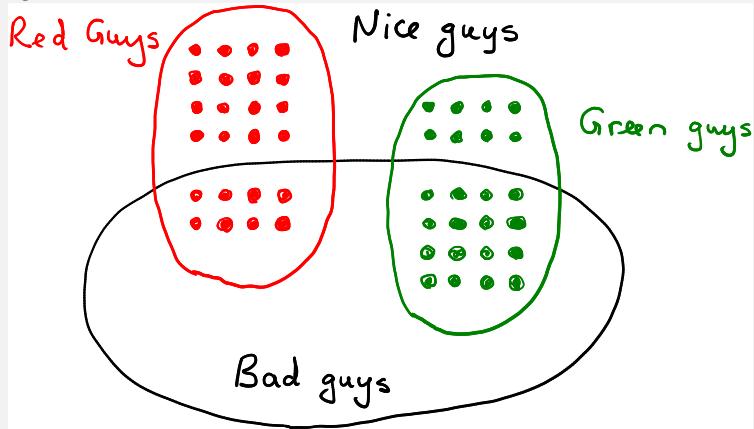


Figure 1.5: Venn diagram of three events: Red, Green, and Bad.

The diagram displays three events, *Red R*, *Green G*, and *Bad B*. There is also a fourth event, *Nice*, but as this is just the complement of *Bad*, we will not discuss it further. In this example, *R* and *G* are mutually exclusive, but events *R* and *B* have some overlap, and so have *G* and *B*. There are 24 reds and 24 greens, so (unconditional) probability to get a green is $\Pr(G) = 0.5$. In a similar fashion, there are 24 Nice-s and 24 Bad-s, so the unconditional probability to find a Bad person is $\Pr(B) = 0.5$.

What is the probability that a person whom the police detains is bad? This depends on whom the police targets:

- Color-blind: arrest persons at random. As there are 24 nice guys and 24 bad guys, the probability that police arrests a bad guy is

$$\Pr(B|Arrest) = \Pr(B) = \frac{24}{48} = \frac{1}{2}.$$

- Target greens: arrest greens only. As there are 8 good and 16 bad Greens, the probability of detaining a bad guy is

$$\Pr(B|Arrest) = \Pr(B|G) = \frac{16}{24} = \frac{2}{3}.$$

- Target reds: arrest reds only. Now the probability to get a bad guy is just

$$\Pr(B|Arrest) = \Pr(B|R) = \frac{8}{24} = \frac{1}{3}.$$

In this example, the police may be tempted to target greens, no matter what is the wider impact to the society.

Exercise 1.5: First class survivors

Consider Titanic passengers. By survival and passenger class, their count is

Class	Survived	Count
1	0	123
1	1	200
2	0	158
2	1	119
3	0	528
3	1	181

Compute:

1. $\Pr(\text{survived}|\text{traveled in 1st class})$
2. $\Pr(\text{traveled in 1st class}|\text{survived})$.

Solution on page 416. See also Exercise 1.7.

Next, let's look at the following, somewhat more complex problem: *Roll two dice. What is the probability to get at least one six, given one of the dies comes with an odd side up?* Let's call the event of interest, "at least one six", A , and conditioning event, "odd side up", B . The corresponding sample space is shown in the Figure 1.6. In essence it is a Venn diagram, exactly as on the Figure 1.4. It is just a more complex one, and it is displayed as a table, not as surface areas. Every simple event in the bottom row and in the rightmost column in the table constitutes the compound event A , we have marked it with blue. In a similar fashion, every odd row and odd column in the table corresponds to the conditioning event B , and we have marked it with pink. The table cells where A and B overlap, e.g. cells (1,6) and (5,6), are marked with purple. Intuitively, it is easy to see that the event we are interested are made of the 6 purple cells, and the conditioning event A are made of the 27 pink (and purple) cells. As all the cells (simple events) are equally likely, the probability of interest, denoted by $\Pr(A|B)$ is $\Pr(A|B) = 6/27 \approx 22.2\%$.

More formally, the conditioning event B partitions the sample space into two parts: one part corresponds to B and the other part corresponds to non- B (Table 1.3). When conditioning on B , we are only interested in the left panel that depicts those cells that were pink on the previous figure. The non- B events (right-hand panel) are irrelevant. So we can just divide the count of simple events of interest (blue cells) by the total number of feasible simple events (cells in the table, 27).

Note that conditioning is not necessarily related to timing or causality. There is

		Die 2						
		1	2	3	4	5	6	
Die 1		1	(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)
		2	(2,1)	(2,2)	(2,3)	(2,4)	(2,5)	(2,6)
		3	(3,1)	(3,2)	(3,3)	(3,4)	(3,5)	(3,6)
		4	(4,1)	(4,2)	(4,3)	(4,4)	(4,5)	(4,6)
		5	(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	(5,6)
		6	(6,1)	(6,2)	(6,3)	(6,4)	(6,5)	(6,6)

Figure 1.6: Roll two dice, get at least one six, given one die has an odd number. This is Venn diagram for a discrete sample space.

Table 1.3: Partitioning the sample space into two subsets. The left side contains all simple events in A , the right side the simple events not in A .

		A occurs						A does not occur							
		Die 2						Die 2							
		1	2	3	4	5	6			1	2	3	4	5	6
Die 1	1	(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)	1							
	2	(2,1)		(2,3)		(2,5)		2		(2,2)		(2,4)		(2,6)	
	3	(3,1)	(3,2)	(3,3)	(3,4)	(3,5)	(3,6)	3							
	4	(4,1)		(4,3)		(4,5)		4		(4,2)		(4,4)		(4,6)	
	5	(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	(5,6)	5							
	6	(6,1)		(6,3)		(6,5)		6		(6,2)		(6,4)		(6,6)	

nothing wrong to ask questions like “what is the probability that the weather is dry today, given the class will be canceled tomorrow?” Also, conditional probability is not the same as causal relationship. While house size is definitely part of factors that determine the house price, the conditioning event is not always a cause. For instance, when computing probability that an email is spam given it contains the word “viagra”, we cannot say that the word “causes” email to be spam. Email is either spam or not, and spam emails are more likely to contain certain words than non-spam emails.

Bayes theorem

Let us now discuss $\Pr(A|B)$. Intuitively, computing conditional probability involves conditioning, focusing on event B only. Essentially we analyze now a smaller sample space, $S_B = \mathcal{S} \cap B$, the blue oval in Figure 1.4. This is equal to B as $B \subseteq \mathcal{S}$. In this new smaller sample space, the event A transforms to $A_B = A \cap B$, the red and blue overlap area in the Figure. In the new, conditioned-on- B -world, the probability of B (or more precisely, $\Pr(B|B)$) is one. We just ignore all events that do not involve B . One can intuitively see that $\Pr(A|B)$ depends on the “size” of $A \cap B$ relative to the size of B . If all points inside of B and A are equally likely, we can find the conditional probability just by dividing the area of $A \cap B$ by the area of B . Now the events A and B do not have anything like “size”⁴ but they have well-defined probability. Hence we

⁴What corresponds to the intuitive concept of “size” is called *measure* in probability theory.

compute the conditional probability as

$$\Pr(A|B) = \frac{\Pr(A \cap B)}{\Pr(B)} = \frac{\Pr(A, B)}{\Pr(B)} \quad (1.3.1)$$

Example 1.4: Gender and Titanic Survival

Consider sinking of RMS Titanic in 1912. She had 1309 passengers, ^a 843 male and 466 female, out of whom 161 male and 339 female survived (see Figure 1.7 below). What is the survival probability, given the passenger was female? Intuitively, it is just the number of female survivors divided by the number of female passengers:

$$\Pr(\text{survived}|\text{female}) = \frac{\text{female survivors}}{\text{all females}} = \frac{339}{466} = 0.727.$$

This calculation is essentially an application of Bayes theorem. Consider the figure below.

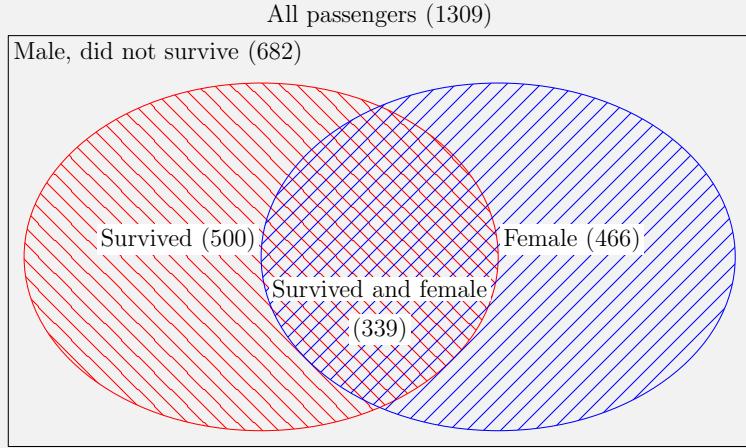


Figure 1.7: Gender distribution among Titanic passengers, displayed as a Venn diagram. Out of 1309 passengers, 466 were female, 500 survived, and 339 were both female and survived.

Our sample space (the box) is “made of” all 1309 passengers. Out of these passengers, 466 were female (blue on the figure), i.e. $\Pr(\text{female}) = 466/1309 = 0.356$. 500 passengers survived (red on the figure), so $\Pr(\text{survived}) = 500/1309 = 0.382$. But there is also an overlap—339 females who survived (red/blue cross shaded in the figure). We can compute this probability (out of all passengers) as $\Pr(\text{survived, female}) = 339/1309 = 0.259$. However, we are not interested in $\Pr(\text{survived, female})$, probability that a random passenger was female and survived, but in the conditional probability $\Pr(\text{survived}|\text{female})$, probability that a random female passenger survived. So we should divide the cross-shaded overlap area with the blue female area:

$$\Pr(\text{survived}|\text{female}) = \frac{339/1309}{466/1309} = \frac{339}{466} = 0.727.$$

This is the same result we got above.

^aThe dataset we refer here contains information about 1309 passengers. There were probably more, and it also carried approximately 885 crew members.

Exercise 1.6: A family has two children...

(This problem is known as *two daughter problem*)

Consider a family with two children. We know that one of these is a girl. What is the probability that the other one is also a girl? Assume gender of children is independent, and $\Pr(\text{boy}) = \Pr(\text{girl}) = 0.5$.

Hint: what is the sample space and the conditioning event in this case? If thinking in terms of sample space in abstract terms is too hard then it is useful to imagine it in terms of a concrete number, e.g. 100 families that all have two children. How many of those belong to the groups of interest?

Solution on page [416](#).

Obviously, because the problem is symmetric—we can just swap A and B in (1.3.1) and have

$$\Pr(B|A) = \frac{\Pr(B, A)}{\Pr(A)}. \quad (1.3.2)$$

Note also that the probability that both events occur, $\Pr(A, B) = \Pr(B, A)$. So we can isolate $\Pr(A, B)$ from (1.3.2) as

$$\Pr(B, A) = \Pr(A, B) = \Pr(B|A) \cdot \Pr(A) \quad (1.3.3)$$

and insert this into (1.3.1) to get

$$\Pr(A|B) = \frac{\Pr(A, B)}{\Pr(B)} = \frac{\Pr(B|A) \cdot \Pr(A)}{\Pr(B)}. \quad (1.3.4)$$

This relationship plays quite a big role when working with conditional probabilities—sometimes it is easy to compute $\Pr(B|A)$ but not $\Pr(A|B)$, and (1.3.4) shows how we can get the latter from the former.

In practical applications, we often have “outcome” in place of event A and “data” in place of event “B”. In this context (1.3.4) shows what is the probability to get “outcome” given we have “data”. This is essentially a predictive modeling problem.

When the counts are given then computing conditional probability reduces to dividing of the two counts of interest. But sometimes the probabilities are more easily available. Consider a diagnosis problem: a doctor meets a patient with certain symptoms, e.g. runny nose, watery eyes, and cough. Does the patient have flu? This is a prediction problem that we can state as $\Pr(\text{flu}|\text{symptoms})$. This is indeed very much the problem the doctors face when encountering a patient. How can we calculate this probability? First, we can use (1.3.4) to express the diagnosis task as

$$\Pr(\text{flu}|\text{symptoms}) = \frac{\Pr(\text{symptoms}|\text{flu}) \cdot \Pr(\text{flu})}{\Pr(\text{symptoms})}. \quad (1.3.5)$$

What are the four probabilities we need to compute the diagnosis?

- $\Pr(\text{symptoms}|\text{flu})$ is the probability to observe symptoms given someone has flu. This data probably exists in hospitals.
- $\Pr(\text{flu})$ is probability that a patient has flu. This data is also likely to exist.
- Finally, $\Pr(\text{symptoms})$ is probability that a patient has such symptoms, flu or no flu. This data is also likely to be present in medical records.

So given our doctor has access to medical data, she is now able to compute the diagnosis!

Example 1.5: Probability of diagnosis

Assume we learn from the medical records that:

- $\Pr(\text{symptoms}|\text{flu}) = 0.3$: 30% of patients with flu have such symptoms.
- $\Pr(\text{flu}) = 0.2$: only 20% of patients have flu. data is also likely to exist.
- Finally, $\Pr(\text{symptoms}) = 0.1$. Such symptoms are observed on 10% of patients.

Now the probability of flu is

$$\Pr(\text{flu}|\text{symptoms}) = \frac{\Pr(\text{symptoms}|\text{flu}) \cdot \Pr(\text{flu})}{\Pr(\text{symptoms})} = \frac{0.3 \cdot 0.2}{0.1} = 0.6.$$

So given such symptoms, it is 60% likely that the patient has flu.

Let us take another look at (1.3.4). In essence it is an updating rule. The doctor starts the diagnosis $\Pr(\text{flu})$. It is called *prior probability* or just *prior*, this is the probability of flu given the doctor hasn't learned about any symptoms. This is the prediction based on no data. However, if we collect data (i.e. observe symptoms), then the prior will be updated by multiplying it with $\frac{\Pr(\text{symptoms}|\text{flu})}{\Pr(\text{symptoms})}$. The product, $\Pr(\text{flu}|\text{symptoms})$, the probability we are interested in, is called *posterior*. So the essence of Bayesian theorem is to update the prior based on data—if we get new information (data), we should update our initial guess (prior) into posterior.

Exercise 1.7: First class given survived

What is the probability that a titanic passenger was traveling in first class given they survived, $\Pr(C = 1|S = 1)$? We know that the percentage of first class passengers who survived was 0.619, percentage of first class passengers was 0.247, and probability of survival was 0.382.

Solution at page 417. See also Exercise 1.5.

The above example assume we know $\Pr(\text{symptoms})$, also called *normalizer*.⁵ This was a reasonable assumption in the diagnosis problem above, but it is not always the case. Consider (1.3.4) again, but now assume we do not know $\Pr(B)$. However, we can compute it if we know both of the following probabilities: the conditional probability

⁵It is called “normalizer” because it “takes care of” that the result will be a valid probability. See more in Section 6.4 Naive Bayes, page 281.

of B given A happens, $\Pr(B|A)$, and the conditional probability of B given A *does not happen*, we denote it by $\Pr(B|\bar{A})$. Now the normalizer can be computed as

$$\Pr(B) = \Pr(B|A) \cdot \Pr(A) + \Pr(B|\bar{A}) \cdot \Pr(\bar{A}). \quad (1.3.6)$$

This is essentially an application of expected value. It is intuitively a fairly obvious rule: B may happen both in case A happens, and in case A does not happen. These events happen with probability $\Pr(A)$ and $\Pr(\bar{A})$. The probability B will happen differs by these two cases, being $\Pr(B|A)$ and $\Pr(B|\bar{A})$ correspondingly.

Expected value is similar to average over a large sample, see more in [Section 1.3.5 Expected Value](#), page 40.

Example 1.6: Do you have cancer?

True positives: one has cancer and test is positive; false positives: one does not have cancer but the test is still positive. See more in [Section 5.2.1 Confusion matrix and related concepts](#), page [232](#).

Consider a very unfortunate situation you may happen to get: you take a test for cancer and the test comes back positive. Do you really have cancer with all its awful consequences on the rest of your life? But tests, in particular the first cheap tests people do, are far from perfect. Maybe the test is wrong?

Denote by $T = 1$ the event of test being positive and $C = 1$ one having cancer. Assume the test is fairly good at spotting true positives, $\Pr(T = 1|C = 1) = 0.99$, but it also reports a large number of false negatives, $\Pr(T = 1|C = 0) = 0.1$, i.e. in 10% of cases where one does not have cancer, the test is still positive. Finally, assume cancer is rare, $\Pr(C = 1) = 0.001$, and hence no-cancer is very common, $\Pr(C = 0) = 0.999$. What is the probability that you actually have cancer given you have a positive test result, $\Pr(C = 1|T = 1)$?

We use Bayes theorem ([1.3.4](#)) to invert the conditional probability:

$$\Pr(C = 1|T = 1) = \frac{\Pr(T = 1|C = 1) \cdot \Pr(C = 1)}{\Pr(T = 1)}.$$

From data above we know the two probabilities in the numerator, but we still have to compute the denominator:

$$\begin{aligned} \Pr(T = 1) &= \Pr(T = 1|C = 1) \cdot \Pr(C = 1) + \Pr(T = 1|C = 0) \cdot \Pr(C = 0) = \\ &= 0.99 \cdot 0.001 + 0.1 \cdot 0.999 = 0.10089. \end{aligned}$$

Now we can just plug this number into the Bayes theorem above and we get

$$\Pr(C = 1|T = 1) = \frac{0.99 \cdot 0.001}{0.10089} = 0.009813.$$

So despite returning with a positive test, it is still less than 1% likely that you actually have cancer!

It is a somewhat counter-intuitive example, where the actual computations are not well-aligned with the intuitive understanding (positive result means cancer). To be more precise, it is not so much “counter-intuitive” as “non-intuitive” as our intuition usually cannot come up with anything reasonable, we just do not have enough experience with similar probability calculations.

Here it is fairly easy to see why the test is not very informative: because of the very large false positive rate, approximately 100 people out of 1000 get the positive result. This dwarfs to 1 person out of 1000 that actually has cancer. Hence most likely (99%) likely you have no reason to worry.

Exercise 1.8: Two bags of M&M

There are two kind of m&m bags, A and B , and they are equally likely. The probability to get a red m&m in bag A is $2/3$ and in bag B it is $1/3$.

1. Dai-yu takes a candy from the bag and gets a red one. What is the probability that it is an A -bag?
2. Now she takes two candies out of a bag, and both are red. What is the

probability that this is a B -bag?

Assume that there are many candies in the bag, so the probability does not change when one is removed.

Solution on page [417](#)

Exercise 1.9: Smile or fights?

You are a caveman 100,000 years ago. You hear someone moving in darkness near your campfire. Is this your friendly neighbor, or a hungry lion? Should you wait and smile, or grab a burning stick and stand ready for fight?

Assume $\Pr(\text{steps}|\text{neighbor}) = 0.2$ (neighbor is fairly quiet) and $\Pr(\text{steps}|\text{lion}) = 0.6$ (lion is fairly noisy). Assume also that $\Pr(\text{neighbor}) = 0.9$ (neighbor is around frequently) and $\Pr(\text{lion}) = 0.1$ (there are not many lions). Compute the relevant probabilities, and try to answer the questions above.

Solution on page [418](#).

Independent Events

Events are *independent* if their joint probability can be factored into a product of two individual event probability. In case of two events:

$$\Pr(A, B) = \Pr(A) \cdot \Pr(B) \quad (1.3.7)$$

In case of K events:

$$\Pr(A_1, A_2, \dots, A_K) = \prod_{k=1}^K \Pr(A_k) \quad (1.3.8)$$

TBD: Examples

Cheatsheet 1.2: Events, Probability and Conditional Probability

Event A possible outcome in random experiment or phenomenon. Example: heads H is an event when flipping a coin.

Sample space Set of all possible events. Example: sample space for coin flip is $\{H, T\}$.

Simple event Event that cannot be divided into more basic events. Example: roll a die, event “1”.

Compound event Event that can be divided into simpler events. Example: roll a die, get an even number.

Mutually exclusive events Events that cannot occur at the same time. Example: roll a die, “1” and “2” are mutually exclusive.

Frequentist probability tendency of an event to happen in a given percentage of trials. Example: toss coin 1000 times, you get H approximately 50% of

times.

Bayesian probability best estimate given available information for how likely is something to happen. Example: *what is the probability it is sunny tomorrow when I know it is raining today?*

Conditional event one event happening given that the other event also happens. Example: *roll a die, get “1” given you get an odd number.*

Conditional probability $\Pr(A|B)$ probability that event A happens given that the event B happens. Example: *roll a die, what is probability of “1” given you got an odd number?, denoted by bar symbol as $\Pr(1|\text{odd number})$.*

Remember: **conditioning event is after the bar symbol!**

Bayes Theorem $\Pr(A|B) = \frac{\Pr(A,B)}{\Pr(B)} = \frac{\Pr(B|A) \cdot \Pr(A)}{\Pr(B)}$.

1.3.4 Random Variable

Random variable (RV) is a central concept that connects probability theory to statistics. In particular, it makes numbers out of events so that one can use the mathematical apparatus to analyze the random processes. The concept of RV is somewhat complex, so we start with the easy part—it is easy to remember what RV is not. First, random variable is not random; and second, random variable is not a variable.

But then what *is* RV? In essence, it is a rule that assigns a number to each event in the sample space \mathcal{S} . Most of the events we care about occur to our physical world and are not numbers, but we need numbers to use the mathematical apparatus. So we need a RV that links the outcomes of the stochastic phenomenon we are analyzing with some sort of numbers.

RV-s are typically denoted by capital Latin letters, such as X or Z . Formally, a RV X is $X : \mathcal{S} \rightarrow \mathbb{R}$ where \mathcal{S} is the sample space of the phenomenon we are analyzing. For instance, if our experiment is flipping a coin, then its sample space is $\{H, T\}$, i.e. it contains just possible events, heads and tails. We can assign zero to tails and one to heads, and define the RV X as

$$X(E) = \begin{cases} 0 & \text{if } E = T \\ 1 & \text{if } E = H. \end{cases} \quad (1.3.9)$$

Obviously, one can also define it in the opposite way $X(H) = 0$ and $X(T) = 1$; and in a myriad of other ways. For instance, if we want the expected value to be zero, (see [Section 1.3.5](#) for explanation) we can define $X(T) = -1$ and $X(H) = 1$.

Now when we actually conduct the experiment and toss the coin, we will receive either H or T . We use RV X to convert the realized outcome into a number and hence we get either 1 or 0. These are two possible *observed values* or *realizations* of the RV. Let's repeat here: H and T are *events*. The corresponding numbers, 0 and 1, are *realizations* (observed values). While RV-s are traditionally denoted by upper case Latin letters, such as X or Y , their observed values (realizations) are denoted with the corresponding lower case letters, such as x and y . If we observe many realizations (e.g. toss the coin multiple times), we usually denote those using a subscript like x_1, x_2, \dots, x_N .

So RV is not random. It is two things:

- a phenomenon or experiment with well-defined properties; and
- a rule how to assign numeric labels to the events in that phenomenon.

But the realizations of RV-s are random.

It is extremely important to be able to distinguish between a non-random RV and its random realizations! Part of the confusion arises from how the word “random” is used: *random* in the concept *random phenomenon* refers to the fact that this phenomenon can produce random realizations. However, the properties of the phenomenon are not random, they are fixed and well defined. But *random* in *random outcomes* refers to the fact that the outcomes are unpredictable, random.

Cheatsheet 1.3: Random variable and realization

- **Random variable** (RV) is two things:
 - a phenomenon or experiment with well-defined properties; and
 - a rule how to assign numeric labels to the events in that phenomenon.

It is a rule, and it is not random.
- **Realizations** are numbers, resulting in a random experiment when converting the outcomes (events) to numbers using a RV. These are random.
- A number of realizations together form a **sample**.

Different ways to define RV-s is related to questions about the physical world we are interested in. Take example of rolling two dice (see Section 1.3.1). For instance, if we are just interested in different outcomes, we can enumerate the combinations by defining

$$Y(E) = \begin{cases} 1 : \text{if } E = (1,1) \\ 2 : \text{if } E = (1,2) \\ \dots \\ 36 : \text{if } E = (6,6). \end{cases} \quad (1.3.10)$$

Now the RV will tell us if we got (1,5), (5,1) or (2,4). All these combinations correspond to different values. However, we may not be interested in the different combinations but instead in the sum of the points, whichever sides come up. Now we can define

$$Z(E) = \begin{cases} 2 : \text{if } E = (1,1) \\ 3 : \text{if } E = (1,2) \text{ or } E = (2,1) \\ 4 : \text{if } E = (1,3) \text{ or } E = (2,2) \text{ or } E = (3,1) \\ \dots \\ 12 : \text{if } E = (6,6). \end{cases} \quad (1.3.11)$$

Exercise 1.10: Rolling two dice

Take the example of two dice. Construct a random variable that answers the question: did we get any 6-s?

Solution on page 418.

The RV outcomes have, in general, different probabilities as they correspond to different events in the sample space. In case of the coin-toss RV X , the value 0 corresponds only to event T and the value 1 to the event H . Both of these events have probability 0.5 if it is a fair coin, and hence the values 0 and 1 will also have equal probability. However, this is not the case for RV Z above that counts the points on two dice. Although all the atomic events are equiprobable, the RV values are not because those correspond to different compound events. Value 2 corresponds only to a single atomic event (1,1) and hence has probability 1/36. Value 3 corresponds to two atomic events, (1,2) and (2,1) and hence has probability 2/36. Probability of 4 is 3/36 and so on.

Exercise 1.11: Find $\Pr(Z = 6)$

Consider the RV Z as defined above. Find $\Pr(Z = 6)$, the probability that rolling two dice will give you sum 6.

Hint: consider drawing a 6×6 table of faces and marking the sum of the dots in each table cell.

Obviously, when the sample space is discrete—there is only a limited number of different events—then there is also only a finite number of possible different RV values. We talk about *discrete random variables*. Discrete random variables can be presented

as probability table. For instance, when tossing a fair coin and denoting heads by 1 (RV X on page 37) we can represent the values as a table:

Value	Probability
0	0.5
1	0.5

Such a table is very convenient when computing expectation, variance, and other properties of the RV.

A few more words about the notation. The random variable, the process of tossing a coin and counting heads, is typically denoted by a capital letter like X . Individual realizations, the actual number of heads that we get when we roll the dice, are typically denoted with lower case letters x . As we usually consider many individual outcomes, we denote those by subscripts x_1 , x_2 , and so on, i.e. x_1 denotes the number of heads in the first toss, x_2 the second toss etc. But in different situations the subscripts may mean different things. In other times we may want to enumerate the different possible outcomes, the lines in the probability table. Now for instance $x_1 = 0$ and $x_2 = 1$. In this case $\Pr(X = x_1)$ means $\Pr(X = 0)$, “the probability to receive no heads when tossing a coin”. This is what we do when talking about expectation below. The notation can be quite confusing, and one has to understand what exactly x_i means in each case.

In case of continuous sample space we may have an infinite number of possible values and we talk about *continuous random variables*. For example, flight delay in minutes or temperature in degrees are continuous random variables (given we measure not just in minutes and degrees but also include the corresponding fractions). There are also different ways to define RV-s in case of continuous sample space like flight delay. The first and most obvious case is just to use the length of the delay d in minutes. Alternatively, if we are not interested in early arrivals, we may construct a different RV: what was the delay, given the flight was delayed?

$$X = \max(0, d) \tag{1.3.12}$$

where d is the delay in minutes.

TBD: independent random variables

1.3.5 Expected Value and Variance

Expected Value

The section 1.2.2 above discusses mean as a way to characterize the central tendency in case of sample of data. Intuitively, one can easily see that as the sample grows, its mean will converge to the “true mean”. For instance, one can immediately understand that the “true mean” when tossing the coin should be 0.5. When thinking about “true mean” we intuitively have in mind a more general sample, the “population”, or perhaps a stochastic process, where the current data is sampled from. This population or stochastic process is essentially a RV and the “true mean” is a certain property of this RV. The property is called *expected value* or *expectation*. It is usually denoted by capital “ \mathbb{E} ”, e.g. $\mathbb{E} X$ means the expected value of random variable X . Its numeric

value is often denoted by μ . Unfortunately it is also common to refer to the expected value as “mean”, e.g. when talking about distributions. So “mean” can refer to either sample mean or to the expected value of a RV. However, “average” is not used to denote the expected value.

For discrete RVs, expectation can be computed as the weighted average of possible outcome values where the weights are the corresponding probabilities:

$$\mathbb{E} X = \sum_i p_i \cdot x_i \quad (1.3.13)$$

where i counts over all possible outcomes of X , denoted by x_i .⁶ Consider the coin toss example where we assigned 1 to heads and 0 to tails. The expected value of X is

$$\mathbb{E} X = 0.5 \cdot 0 + 0.5 \cdot 1 = 0.5. \quad (1.3.14)$$

This is intuitively obvious: in average, we get heads half of the times.

Example 1.7: Expectation of a 3-valued RV

Consider a more complex example. Take a RV

$$Y = \begin{cases} 0 & \text{with probability } 0.5 \\ 1 & 0.25 \\ 2 & 0.25. \end{cases} \quad (1.3.15)$$

Its expectation is $\mathbb{E} Y = 0.5 \cdot 0 + 0.25 \cdot 1 + 0.25 \cdot 2 = 0.75$

Exercise 1.12: Expected value of die

Consider rolling a die as a RV D . Denote its values by $1, 2, \dots, 6$. What is its expected value $\mathbb{E} D$?

Exercise 1.13: How many sixes do we get?

Consider an experiment of rolling two dice. We are interested in how many sixes did we get. The corresponding RV will look like

N	Probability
0	25/36
1	10/36
2	1/36

1. Show that these probabilities are correct.
2. Compute the expected value of this RV, the expected number of sixes when rolling two dice.

⁶Note: here i enumerates the possible outcomes, not consecutive experiments. See page 40 for comments on notation.

Solution on page [419](#).

The weighted sum in the definition of the expected value ([1.3.13](#)) transforms to an integral in case of continuous RV-s, see [Section 1.4.2 What are continuous RV-s](#), page [52](#) and equation ([1.4.13](#)) below.

Note that expected value is not a random variable, nor is it random in any other way. It is just a number.⁷ As the expected value is just a number, its expectation, in turn, is just the same number. So when we sometimes need to compute expected value of expected value, we have $\mathbb{E}(\mathbb{E} X) = \mathbb{E} X$.

It is important to keep in mind that expectation is not sample mean and the way around. Expectation is a property of random variable, a precisely defined stochastic process. Sample mean is a property of sample. Even if expectation is sometimes called “mean”, it is important to realize that sample and RV have different properties. For instance, sample mean is random and it fluctuates depending on what is sampled. But expectation is constant and does not change. Say, tossing coin a few times may result in a different mean, but the expected number of heads is always 0.5. One can also compute mean for every sample but not every RV has expectation (see, e.g. [Pareto distribution](#) below).

Theorem 1 (Law of large numbers, LLN).⁸ Let x_1, x_2, \dots, x_N be independent realizations of a RV X . Assume the expected value $\mathbb{E} X = \mu$ exists. Now the sample average converges to the expected value.

$$\frac{1}{N} \sum_{i=1}^N x_i \equiv \bar{X}_n \xrightarrow{P} \mu. \quad (1.3.16)$$

TBD: A more complex example

TBD: Conditional expectations

Variance

While expectation is similar to the sample mean, variance is similar to the sample variance. Variance is a much less intuitive concept than expectation, in exactly the same way as sample variance is much less intuitive than sample mean. The naming convention is not helpful either: unlike expectation versus mean, both of these concepts are called “variance”. Usually, the context makes it clear whether we are talking about variance as the property of RV, or about the sample variance (is it a sample? is it a RV?). But where needed, we indicate the type of the concept by writing “variance of the RV” or “theoretical variance” when we talk about random variables, and “sample variance” when we talk about data.

Variance is typically denoted by Var , e.g. $\text{Var } X$ is variance of the random variable X . Its numerical values are often denoted by σ^2 , stressing that its definition is related

⁷If we want, we can imagine that numbers are degenerate RV-s where all realizations are the same, so these RV-s will have 100% probability on the only outcome. Obviously, the expected value of such a RV is the outcome value.

⁸Symbol \xrightarrow{P} means *convergence in probability*: as N grows, the probability $\Pr(|\bar{X}_N - \mu| > \epsilon)$ gets arbitrarily small for every positive ϵ .

to squared deviations. As a bonus, when denoting variance by σ^2 we can denote the standard deviation by just σ .

Variance is one of the most important statistical concepts, most of the statistical inference is in fact based on variance in one way or another. It is defined in the same way as sample variance while replacing means with expectations. So variance of RV X is defined as

$$\text{Var } X = \mathbb{E}(X - \mathbb{E} X)^2. \quad (1.3.17)$$

Let us explain what this means. First, $\mathbb{E} X$ in the parenthesis is the expected value of X . It is just a number, a constant. Next, $X - \mathbb{E} X$ is the deviation of X from its expected value. It is just X minus a number. As X is a RV, so is $X - \mathbb{E} X$. Third, $(X - \mathbb{E} X)^2$ is just a squared value of the deviation. As the deviation is a RV, so is its square. And finally, $\mathbb{E}(X - \mathbb{E} X)^2$ is the expected value of that RV. So variance can be computed in a similar fashion as expectations. Let us consider an example. Take the RV from Example 1.7:

y	$\Pr(Y = y)$
0	0.50
1	0.25
2	0.25

Above we computed $\mathbb{E} Y = 0.75$. Let us now compute its variance using the definition. The most straightforward approach is to extend the table above with the auxiliary RV-s (Table 1.4). The first two columns represent the RV realizations y and the corresponding probabilities $\Pr(Y = y)$, it is just a copy of the definition table above. The third column is the deviation from the expected value, $y - \mathbb{E} Y$. The fourth column is the deviation squared. The variance is just the expected value of the fourth column. The probability values in the second column are not affected by the other operations—computing the deviation and squaring it. Hence the variance is $\mathbb{E}(Y - \mathbb{E} Y)^2 = 0.5 \cdot 0.5625 + 0.25 \cdot 0.0625 + 0.25 \cdot 1.5625 = 0.6875$.

Table 1.4: Computing variance of a discrete random variable

y_i	$\Pr(Y = y_i)$	$y_i - \mathbb{E} Y$	$(y_i - \mathbb{E} Y)^2$
0	0.50	-0.75	0.5625
1	0.25	0.25	0.0625
2	0.25	1.25	1.5625

The easiest way to compute the variance of a RV by using the definition is to add columns for $Y - \mathbb{E} Y$ and $(Y - \mathbb{E} Y)^2$ in the table of RV values. Variance is simply the expected value of the last column, here 0.6875. See explanations in text.

In practice it is somewhat easier to use another formula

$$\text{Var } X = \mathbb{E}(X^2) - (\mathbb{E} X)^2. \quad (1.3.18)$$

Sample variance is the average squared deviation from the mean in a dataset:
 $s^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2$. See Section 1.2.2 on page 10.

This involves computing the expected value of X^2 . It is easy to show that this formula is equivalent to the definition of variance (1.3.17). Let us re-compute the variance we did above using this formula. First, we have to find $\mathbb{E} X^2$. This is $\mathbb{E} X^2 = 0.5 \cdot 0^2 + 0.25 \cdot 1^2 + 0.25 \cdot 2^2 = 0.25 + 1 = 1.25$. Hence the variance is $\text{Var } X = \mathbb{E} X^2 - (\mathbb{E} X)^2 = 1.25 - 0.75^2 = 1.25 - 0.5625 = 0.6875$. This is the same number we found above.

Exercise 1.14: Compute variance of a RV

Consider a RV

x	$\Pr(X = x)$
-1	0.25
0	0.50
1	0.25

Compute its variance using a) the definition formula (1.3.17); and b) the shortcut formula 1.3.18.

Solution on page 419

Exercise 1.15: Variance of Bernoulli RV

Bernoulli(p) RV (see Section 1.4.1 Bernoulli Distribution, page 49) is a RV that can take values

x	$\Pr(X = x)$
0	$1 - p$
1	p

Compute its variance using both the definition formula (1.3.17) and the shortcut formula (1.3.18).

Solution on page 420

Expected value and variance of functions of RV-s

Quite often we want to compute not just expected value $\mathbb{E} X$ but expected value of a certain function of the RV, for instance, $\mathbb{E} 2X$, $\mathbb{E}[X + Y]$ or $\text{Var } e^X$. Below, we discuss two special cases: expected value and variance of a RV multiplied by a scalar, and of a sum of two RV-s. We show that how to generalize it to arbitrary sums of independent RV-s. However, in general, $\mathbb{E} f(X) \neq f(\mathbb{E} X)$.

Functions of RV-s Before we get to the results, a brief explanation about what does a function of RV mean. It means performing the function on the *values* of the RV, while leaving the probabilities unchanged. For instance, consider the RV, described in Table 1.4. We compute the following functions: $2 \cdot Y$ and e^Y (see Table 1.5 below). The process involves in just computing the corresponding values $2y_i$ and e^{y_i} for all possible outcomes i . But the probabilities, $\Pr(Y = y_i)$ will remain unaffected.

Table 1.5: Functions of RV-s. The table shows the RV Y , and values of the corresponding functions $2Y$ and e^Y .

y_i	$\Pr(Y = y)$	$2y_i$	e^y
0	0.50	0	1
1	0.25	2	2.7183
2	0.25	4	7.3891

RV multiplied by a scalar Intuitively, it is easy to see that when we multiply the RV with a scalar, the expected value will be just the original expected value, multiplied by the same scalar. For instance, if we flip a coin and label heads as 1 and tails as 0, then the expected value is 0.5. When we multiply the RV by two, i.e. we label heads by 2 and tails by 0, then the expected value will be 1. In case of the RV $2Y$ in Table 1.5, we need to compute $\mathbb{E}[2Y] = 0.5 \cdot 0 + 0.25 \cdot 2 + 0.25 \cdot 4 = 1.5$ while $\mathbb{E}e^Y = 0.5 \cdot 1 + 0.25 \cdot 2.7183 + 0.25 \cdot 7.3891 = 3.0268$.

Scalar is just a number.

We can state this as a theorem

Theorem 2 (Expected value of RV multiplied by scalar).

$$\mathbb{E} \lambda \cdot X = \lambda \cdot \mathbb{E} X. \quad (1.3.19)$$

The proof is fairly obvious and is left out.

TBD: proof as an exercise

TBD: variance of this

Sum of two independent RV-s Other times we need to compute, e.g. $\mathbb{E}[X+Y]$ where X and Y are different independent RV-s. Here we only discuss the case where both X and Y are similar, for instance you flip two coins, and X describes the outcome of the first coin, and Y that of the second coin. But the idea carries over to different RV-s too, as long as they are independent. As in case of when multiplying the RV-s with a scalar, the outcome here is fairly intuitive. Imagine you flip two coins and label heads with 1 and tails with 0. What is the expected number when you add the values together? As both coins will have an expected value of 0.5, the sum of their outcomes will just be sum of these expected values, i.e. 1. We state the intuitive result here as a theorem, and prove it underneath.

Theorem 3 (Sum of independent RV-s). If X and Y are independent RV-s, the expected value of their sum is

$$\mathbb{E}[X + Y] = \mathbb{E} X + \mathbb{E} Y. \quad (1.3.20)$$

Proof 1.3.1: Sum of independent RV-s

Let X and Y be two discrete random variables where X has possible outcomes x_1, x_2, \dots, x_N with the corresponding probabilities p_1, p_2, \dots, p_N ; and Y has possible outcomes y_1, y_2, \dots, y_M with the corresponding probabilities q_1, q_2, \dots, q_M .

(1.3.13): $\mathbb{E} X = \sum_i p_i \cdot x_i$

By definition (1.3.13), the expected value of $X + Y$ is

$$\begin{aligned}\mathbb{E}[X + Y] &= \sum_{i=1}^N p_i \left[\sum_{j=1}^M q_j (x_i + y_j) \right] = \\ &= \sum_{i=1}^N p_i \left[x_i \sum_{j=1}^M q_j + \sum_{j=1}^M q_j y_j \right] = \\ &= \sum_{i=1}^N p_i x_i + \sum_{i=1}^N p_i \mathbb{E} Y = \mathbb{E} X + \mathbb{E} Y.\end{aligned}\quad (1.3.21)$$

The proof uses a number of facts:

- X and Y are independent, and hence the probability that x_i and y_j happen is $p_i q_j$.
- as q_j is not related to i , we can sum just compute the $\sum_{j=1}^M q_j = 1$.
- $\sum_{j=1}^M q_j y_j = \mathbb{E} Y$ by the definition of the expected value.

TBD: some sort of example, exercise

TBD: add to cheatsheet

TBD: non-linear functions

Cheatsheet 1.4: Expected value, mean, variance

Realization is the actual outcome you get in a random experiment, such as coin toss. See Section 1.3.4, page 37.

- **Sample mean** (aka *average*) is just an average of the random realizations, the sample. Mean of x_1, x_2, \dots is often denoted by \bar{x} .

Example: toss coin 4 times and mark the heads as “1” and tails as “0”. The realizations (sample) may be 0, 0, 1, 0. The sample mean $\bar{x} = 0.25$.

- **Expected value** (aka *expectation, mean*) is a property of random variable, the random process we are analyzing. It is not random, i.e. it is not related to sample. Expected value of RV X is denoted by $\mathbb{E} X$, in discrete case it can be computed as

$$\mathbb{E} X = \sum_i p_i \cdot x_i$$

where i counts the different possible outcomes and p_i is the corresponding probability.

For instance, if you toss a fair coin then the expected value $\mathbb{E} X = 0.5$.

- **Sample variance** (aka *variance*) is a standardized measure of variation in the sample. It is often denoted by s^2 :

$$s^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2 = \bar{x}^2 - (\bar{x})^2$$

where N is sample size. The first formula is the definition, the second one is easier to use.

Sample variance depends on the sample. For instance, the variance of the coin toss example above is 0.1875.

- **Variance** (aka *variance of RV*) is a standard measure of variation of a RV. It is often denoted by σ^2 :

$$\sigma^2 = \mathbb{E} (X - \mathbb{E} X)^2 = \mathbb{E} X^2 - (\mathbb{E} X)^2$$

The first formula is the definition, the second one is easier to use.

It is a property of RV and does not depend on sample. For instance, the variance of the RV that describes tossing a fair coin is 0.25.

- *Law of Large Numbers* tells that if sample gets large, sample mean and sample variance will be close to the corresponding expected value and variance.

Note that the word “variance” may mean both sample and RV variance. The word is the same but the concepts are different.

1.4 Distributions

Now when we have introduced RV-s, we want to describe their properties and to distinguish between different RV-s. Some of the most useful and most widely used properties are expected value and variance. But often we want more information, in fact the complete information about random variables. This is where distributions come into the play.

For RV-s, distributions describes the “frequency” of different values. In a similar fashion as we have pairs of concepts like expectation and mean to describe the RV and the sample, we can talk about distribution (in case of RV) and histogram (in case of sample). When the sample size gets large, the histogram becomes similar to the distribution, in a similar fashion as sample mean approaches the expected value. But note that in practice the word *distribution* means both, the property of RV-s and the sample histogram (in a similar fashion as “variance” means both RV and sample property).

We start with discrete RV-s where the corresponding function, *probability mass function* (p.m.f), corresponds to the theoretical frequency of different values, and move to continuous RV-s thereafter where *probability density function* (p.d.f) has a slightly different interpretation. Another popular measure, *cumulative distribution function* (c.d.f) gives the probability that the observed values is *less* than its argument.

1.4.1 Discrete Case

Let’s start with a simple example: we toss two coins and count the number of heads (assume both are fair coins). The outcomes and their probabilities are in Table 1.6. This table essentially describes what is known as *probability mass function* (p.m.f)

Table 1.6: Possible outcomes number of heads when tossing two coins, and corresponding probabilities.

x	$\Pr(X = x)$
0	0.25
1	0.5
2	0.25

in case of discrete RV-s. p.m.f is a function that for each possible value of x assigns the corresponding probability. (It can also be trivially extended by assigning the probability 0 to every impossible value.):

$$f(x) = \Pr(X = x) \quad (1.4.1)$$

So we can, somewhat trivially, restate the table as p.m.f:

$$f(x) = \begin{cases} 0.25 & \text{if } x = 0 \\ 0.5 & \text{if } x = 1 \\ 0.25 & \text{if } x = 2 \end{cases} \quad (1.4.2)$$

In a general discrete case, p.m.f must be described as the table above, or a corresponding graph. However, there are numerous processes that generate p.m.f-s with a given structure, for instance the example case of tossing two coins results in a binomial distribution, more precisely in $\text{Binom}(2, 0.5)$.

Another widely used function is *cumulative distribution function* (c.d.f). It answers the question “what is the probability that the outcome is no larger than a given number”:

$$F(x) = \Pr(X \leq x). \quad (1.4.3)$$

If f is the p.m.f, one can easily compute c.d.f as

$$F(x) = \sum_{x' \leq x} f(x'). \quad (1.4.4)$$

In the example case above we have

$$F(x) = \begin{cases} 0 & \text{if } x < 0 \\ 0.25 & \text{if } 0 \leq x < 1 \\ 0.75 & \text{if } 1 \leq x < 2 \\ 1 & \text{if } x \geq 2 \end{cases} \quad (1.4.5)$$

Exercise 1.16: Measure for c.d.f

What kind of measure—nominal, ordinal, difference, or ratio—must X be for its c.d.f to be well defined?

Bernoulli Distribution

Bernoulli RV is perhaps the easiest RV. It is a process that can result in two events: event E with probability p , and event *non- E* with probability $1 - p$. Normally we denote E by 1 and non- E (\bar{E}) by 0. An easy example is flipping a fair coin: with probability $p = 0.5$ the event “heads” will occur, and with probability $1 - 0.5$ the event “heads” will not occur. This is often written as *Bernoulli*(0.5) process. Figure 1.8 demonstrates the corresponding p.m.f.

Bernoulli RV is widely used in practice because many interesting questions can be described as the interesting event occurs versus it does not occur. For instance: does a patient have the illness or not? Will the customer buy the product or not? Does this image depict a cat or not? Many other processes are based on Bernoulli (e.g. Binomial), or partly based on Bernoulli process (e.g. zero-inflated distributions).

The expected value of Bernoulli is simple and intuitive but its variance is not. It is good to know how to compute it because it is so widely used in practice. This will be important below when computing standard errors for sample fraction in [Section 1.5.3 Comparing Distributions](#), page 78.

The expected value of Bernoulli process is very intuitive:

$$\mathbb{E} X = p \cdot 1 + (1 - p) \cdot 0 = p. \quad (1.4.6)$$

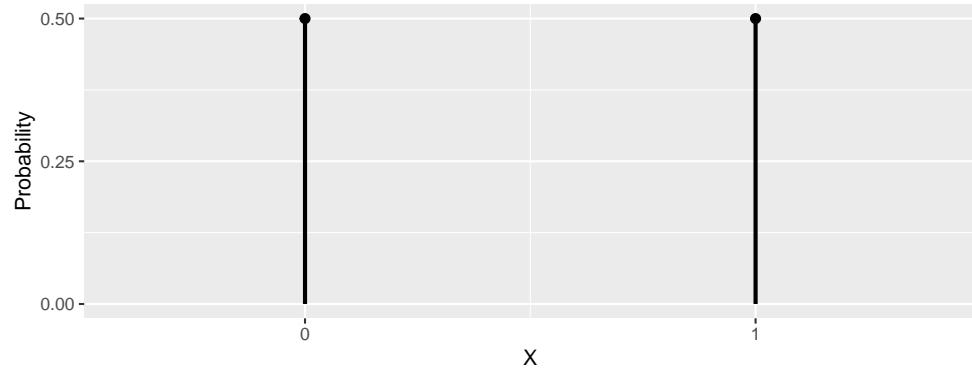


Figure 1.8: Bernoulli(0.5) p.m.f. Event E occurs with probability 0.5, and the event \bar{E} (non- E) also occurs with probability 0.5.

Its variance is also simple although not intuitive (see Exercise 1.15). We can compute it from the variance formula (1.3.18):

$$\text{Var } X = \mathbb{E} X^2 - (\mathbb{E} X)^2 = p - p^2 = p(1 - p). \quad (1.4.7)$$

We used the fact that for Bernoulli distribution, $X^2 = X$ as we only have values 0 and 1, and hence $\mathbb{E} X^2 = \mathbb{E} X = p$.

Binomial Distribution

- Repeat (independent) Bernoulli- p process S times.
 - Count “successes”
- $$x = \sum_i x_i$$
- Example: toss 4 coins. How many heads you get?
 - Example: look at 10 Titanic passengers. How many of them did survive?

Discrete Uniform Distribution

This is a discrete distribution where all possible outcomes have equal probability. The examples include toss of fair coin, roll of fair die, or suit of a random card, drawn from the complete deck. Discrete uniform distribution over elements of set \mathcal{S} assigns equal probability on each element of \mathcal{S} . Its p.m.f is simply

$$f(x) = \begin{cases} \frac{1}{|\mathcal{S}|} & \text{if } x \in \mathcal{S} \\ 0 & \text{otherwise} \end{cases} \quad (1.4.8)$$

When we talk about “random sample” or “selecting at random”, we usually mean a discrete uniform selection where all possible events have equal probability to end up

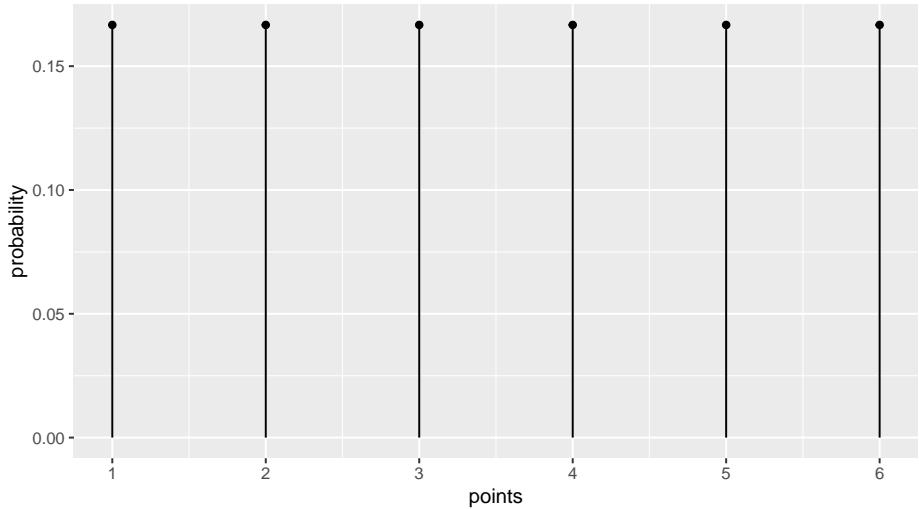
being selected. Strictly speaking, it does not have to be so—random sample is still a random sample even if the subjects are not picked with the equal probability, but such usage of the word is much less common (a related concept is *stratified sample*).

Example 1.8: Rolling a die

On a fair die, all size sides are equally likely and hence the p.m.f is

$$f(x) = \begin{cases} 1/6 & \text{if } x = 1 \\ 1/6 & \text{if } x = 2 \\ 1/6 & \text{if } x = 3 \\ 1/6 & \text{if } x = 4 \\ 1/6 & \text{if } x = 5 \\ 1/6 & \text{if } x = 6 \end{cases} \quad (1.4.9)$$

The function can be depicted graphically as follows:

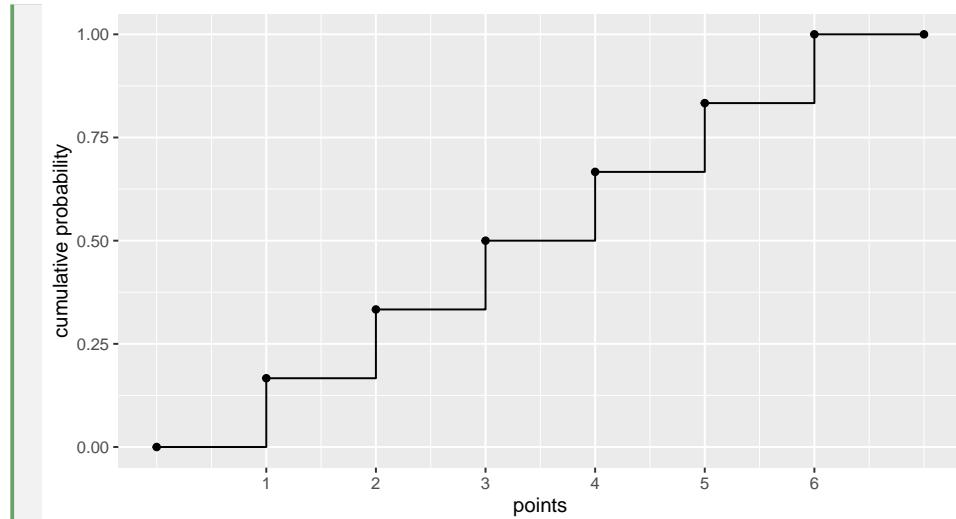


Each bar corresponds to the probability to get the corresponding number on a die, in case of uniform distribution these are all of equal length.

The respective c.d.f is given by

$$F(x) = \begin{cases} 0 & \text{if } x < 1 \\ 1/6 & \text{if } 1 \leq x < 2 \\ 2/6 & \text{if } 2 \leq x < 3 \\ 3/6 & \text{if } 3 \leq x < 4 \\ 4/6 & \text{if } 4 \leq x < 5 \\ 5/6 & \text{if } 5 \leq x < 6 \\ 1 & \text{if } x \geq 6 \end{cases} \quad (1.4.10)$$

and looks like a staircase on graph:



The dots stress that the function has achieved the “upper” level at these points, for instance $F(1) = 1/6$, and not zero.

1.4.2 Continuous RV-s

What are continuous RV-s

Many phenomena can have not just a limited set of values, but values that are everywhere in a continuous interval. For instance, flight delay, human height, or human income can be essentially every single number (within a reasonable range). Hence we cannot create a table of possible values like in Table 1.6. First, there are infinite number of possible values, and second, every particular value is extremely rare. How often it happens that your flight is delayed by exactly 54.321 minutes? Such phenomena are described by *continuous random variables*.

But we can still get close to the frequency tables and p.m.f plots. The trick is to partition the sample space into “bins” and treat the bins as discrete values. The more data we have, the narrower bins we can create, and as a result we get smoother and smoother pictures. Figure 1.9 displays this process. On the two upper panels we have 25 realizations. The top-left plot puts the results into five separate bins. The tallest bins are in the middle with the corresponding counts being 12 and 9. All other counts are much smaller. On the top-right panel we repeat the process with 100 bins. Most of the bins are empty now, but we still see that the tallest bin contains three realizations.

The two lower panels repeat the process with 10000 realizations. The bottom-left panel splits the values into four bins, and now the result is fairly symmetric around zero. Bottom-right panel uses 100 bins that display a fairly smooth bell-curve. Note that the bottom figures do not display counts but frequencies—counts divided by the width of the corresponding bins. We do this because frequencies remain roughly equal if we use a large number of narrow bins instead of a small number of wide bins. Two

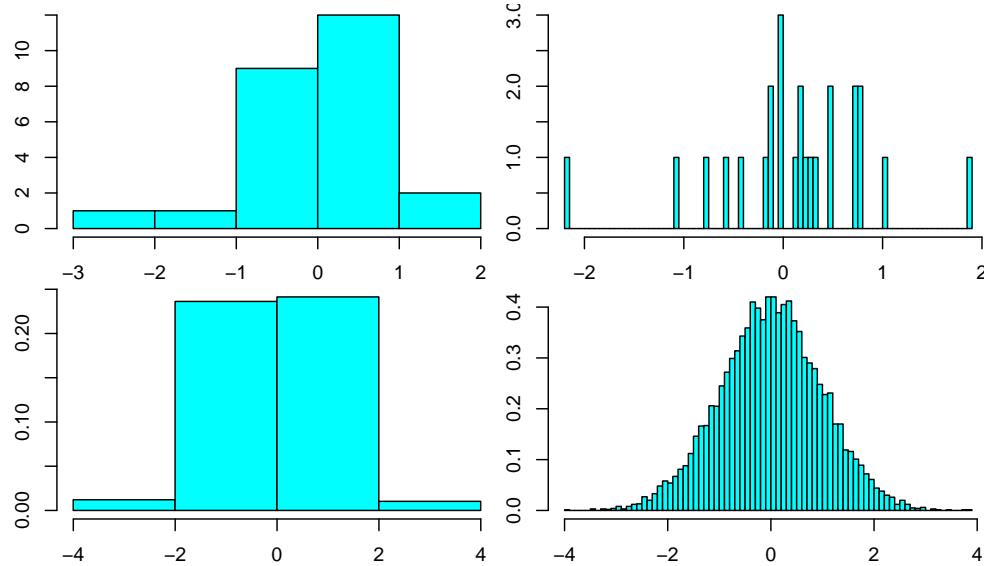


Figure 1.9: Moving from discrete to continuous values. All histograms display random normal realizations. On the left panels we partition the value range into a small number of intervals, and on the right panel into a large number intervals. We are effectively looking at a discrete distribution with that many possible values. The upper panel displays a sample of 30 random values, the lower panel 10,000 random values. The upper panel displays counts of values in each bin, the lower panel the relative frequency.

bins of half width instead of one will both contain roughly a half of the cases of the original wide bin. But by dividing the count by only a half of the original width, we retain (roughly) equal frequency.

We can continue this process as we get more and more data. At the limit when looking at intervals of infinitesimal width, the resulting frequencies are called *probability density function* (p.d.f.). This is the RV counterpart of histogram, in a similar fashion as expected value (property of RV) corresponds to average (property of sample). In a similar fashion as the sample average converges to the expected value as $N \rightarrow \infty$, the histogram converges to the p.d.f.

p.d.f is often denoted by $f(x)$ and can define it as

$$f(x) = \lim_{dx \rightarrow 0} \frac{\Pr(X \in [x, x + dx])}{dx}. \quad (1.4.11)$$

If it looks like definition of derivative to you then you are right, p.d.f is a derivative of a closely related function, cumulative distribution function (see below). p.d.f can also be defined as integral of the function over an interval is the probability that the value falls into this interval:

$$f(x) : P(x \in B) = \int_B f(x) dx \quad (1.4.12)$$

The definition of expectation and variance are intuitively the same in case of continuous RV-s as in case of discrete RV-s, just we have to replace the sums by respective integrals. Expected value is defined as

$$\mathbb{E} X = \int_{-\infty}^{\infty} f(x)x dx, \quad (1.4.13)$$

essentially a weighted average where weights is the corresponding p.d.f. The variance definition can be written exactly the same way as for discrete RV-s (see (1.3.17)), just keep in mind that the expectation must now be calculated using the integral (1.4.13):

$$\text{Var } X = \mathbb{E}[(X - \mathbb{E} X)^2] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2 \quad (1.4.14)$$

where the first part is the definition and the second part the shortcut formula (see (1.3.18)).

Example 1.9: Expected value of uniform RV

Let's calculate the expected value for standard uniform RV. Standard uniform is a continuous distribution where all values between 0 and 1 are equally likely, and other values are impossible. Its density function is just $f(x) = \mathbb{1}(x \in [0,1])$, so we just integrate over the $[0,1]$ interval:

$$\mathbb{E} X = \int_0^1 1 x dx = \frac{1}{2} x^2 \Big|_0^1 = \frac{1}{2}. \quad (1.4.15)$$

This is an intuitive result: if all values between 0 and 1 are equally likely, we get the average between these numbers.

$\mathbb{1}(\cdot)$ is the indicator function. It is just a shorthand to write $f(x) = 1$ if $x \in [0,1]$, otherwise $f(x) = 0$. See Section 0.1, page viii.

TBD: Distribution vs process

Popular Distributions

Here we discuss and give examples of a few commonly used distributions.

TBD: Uniform

Exercise 1.17: Quantiles of standard uniform distribution

Consider standard uniform distribution

$$f(x) = \begin{cases} 1 & \text{if } 0 \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad (1.4.16)$$

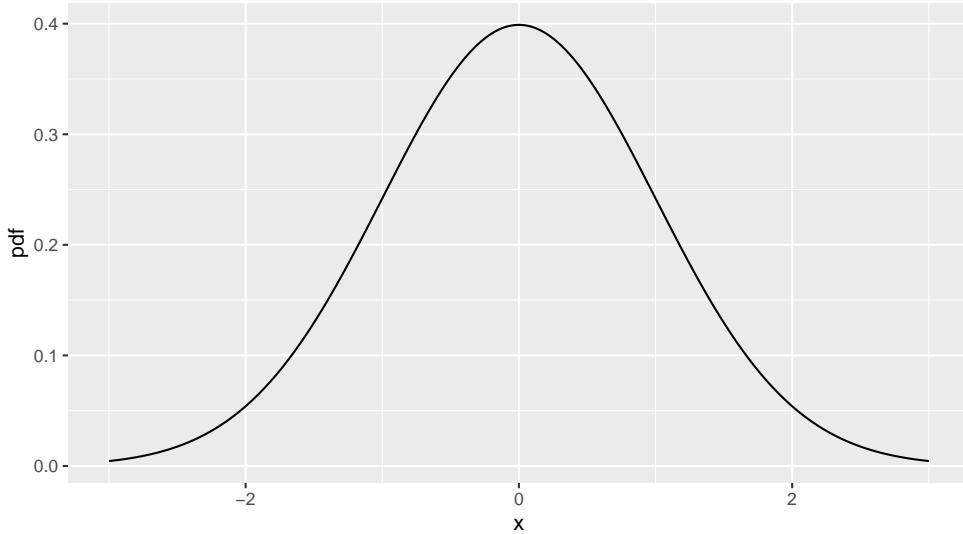
Find the theoretical quantiles $q_{0.025}$ and $q_{0.975}$.

Normal distribution

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

$$\mathbb{E} X = \mu$$

$$\text{Var } X = \sigma^2$$



Normal distribution is extremely widely used

- means and sums of a large number of values tend to be normal (Central Limit Theorem)
 - This makes normal a distribution of choice for statistical inference
 - Many natural features are close to normal
 - size of adult organisms
 - retirement age
 - temperature
 - agricultural yields
 - These are roughly equal outcomes
- TBD: normal

t-distribution

TBD: *t*

Log-normal distribution

Normal distribution has wide range of applications, but there are important classes of phenomena that are rather different from normal. Examples include price (Figure 1.10) and income (Figure 3.11 at page 161). Intuitively, it is easy to understand that these two example cannot be normally distributed—both price and income cannot be negative, but there are no clear upper limits for either. And while most income and price values tend to be “typical”, extremely rich people and super expensive prices exist. So in both cases we expect to see a right-skewed distribution with a long right tail.

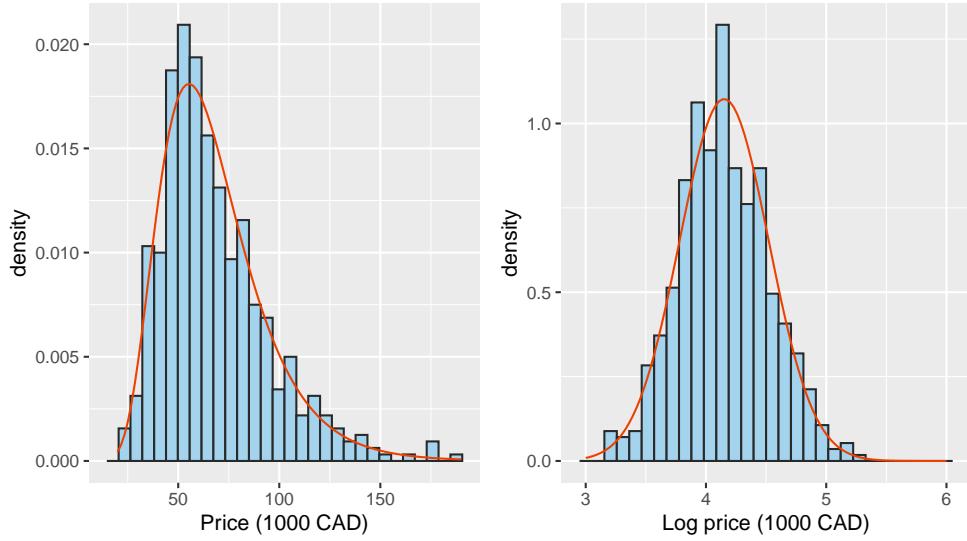


Figure 1.10: House prices in 1987 in Windsor, Canada. The left panel shows price histogram, the right panel log-price histogram. As the latter looks broadly normal, we conclude that the price distribution is approximately log-normal. The red lines show the best-fit smooth log-normal density (left) and normal density (right). Log-normal density can be imagined as a normal bell curve that is squeezed from left and stretched from right. Both red curves match data well.

It turns out that in case of price and income, if we take logarithm of these values, i.e. analyze log income instead of income, we get a distribution that looks quite close to normal. This is the idea of log-normal distribution: a RV is log-normally distributed if its logarithm is normally distributed. The p.d.f. of log-normal distribution (red line on Figure 1.10, left panel) resembles a normal curve, just its left side is compressed and its right side stretched: this is because exponentiation compresses small numbers and stretches large numbers. Its p.d.f. is given by

$$f(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2} \frac{(\log x - \mu)^2}{\sigma^2}} \quad (1.4.17)$$

p.d.f is probability density function, see [Section 1.4.2](#) page 52.

where μ and σ^2 are mean and variance of the corresponding normal distribution (distribution of $\log X$). Log-normally distributed RV-s are often denoted by $LN(\mu, \sigma^2)$. If you are using computer to work with log-normal values, these two parameters may be called “scale” and “shape”.

The expected value and variance of log-normal are

$$\mathbb{E} X = e^{\mu + \frac{1}{2}\sigma^2} \quad \text{Var } X = e^{2\mu + \sigma^2}(e^{\sigma^2} - 1). \quad (1.4.18)$$

Figure 1.11 shows the p.d.f for a few combinations of the parameters μ and σ .

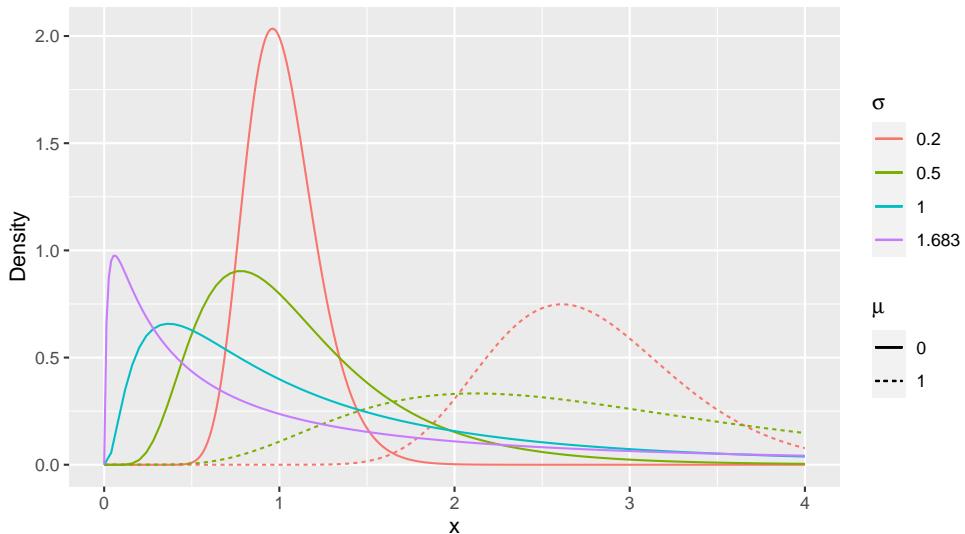


Figure 1.11: Example log-normal p.d.f-s. σ changes the concentration of the distribution: small σ corresponds to a fairly concentrated values that are distributed in a rather normal fashion, large σ in turn describes a distribution with heavy right tail. μ changes scale, the example curves with $\mu = 1$ are of similar shape as the corresponding curves with $\mu = 0$, just more stretched out. See Table 1.7 for the corresponding inequalities.

Why do some phenomena follow log-normal instead of normal distribution? There are two partial explanations:

- Neither price nor income can be negative. Hence whatever distribution these two phenomena follow, it cannot contain negative values. But normal distribution stretches to negative values.
- It appears that the processes that create income and price are not additive but multiplicative: instead of a sum of many independent random processes, these phenomena seem to be better described as a *product* of independent positive random processes. Hence log values look like normal as logarithm transforms the product to a sum.

80/20 rule: upper $x\%$ of population possesses $100 - x\%$ of all ressource, e.g. upper 20% owns 80% of all wealth. See Section [Section 1.2.2 Pareto ratio](#), page 15.

The parameter σ of log-normal distribution describes different degree of inequality: on Figure 1.10, typical houses cost \$50k, but some houses are over \$150k (CAD). This means the most expensive houses cost thrice as much as typical houses.⁹ However, the inequality in house prices are not very large: the 80/20 rule for this distribution tells that the most expensive 43% of houses in Windsor contain 57% of total housing value in that neighborhood. In this case $\sigma = 0.372$. Surprisingly, the UK income distribution in Figure 3.11 is more equal, the 20/80 ratio is 47.2/52.8 as $\sigma = 0.142$. But Titanic ticket prices are more unequal. Here $\sigma = 0.909$ and 32.5% of passengers paid 67.5% of the total ticket revenue. A few more 20/80 ratios are given in Table 1.7. We see that case of large σ , the right tail is very long and indicates the presence of super-wealthy: in case of $\sigma = 3.29$ the upper 5% of population owns 95% of the resources.

Table 1.7: Log-normal 20/80 ratios depending on σ . For instance, if $\sigma = 3.29$ then the upper 5% of population possesses 95% of total resources. See Figure 1.11 for the shape of the corresponding p.d.f-s.

σ	Top share (pct)	Owned wealth (pct)
0.20	46.02	53.98
0.50	40.13	59.87
1.00	30.85	69.15
1.68	20.00	80.00
3.29	5.00	95.00

Pareto distribution

Log-normal distribution is a good approximation for individual income. But there are phenomena that are much more extreme. For instance, human influence, web site popularity, and size of cities tend to be distributed in a much more unequal manner. In such case Pareto distribution can be a good approximation.

The p.d.f of Pareto distribution is given by

$$f(x) = \alpha x_0^\alpha x^{-\alpha-1}, \quad x > x_0. \quad (1.4.19)$$

It has two parameters: “scale” x_0 and “shape” α . As p.d.f is proportional to x risen to power $-\alpha - 1$, Pareto distribution is often called *power law*.

Figure 1.12 shows a few examples of the p.d.f with different α using linear scale (left panel) and log-log scale (right panel). Shape controls the spread of the values—Pareto values drawn from a large α distribution are fairly concentrated (green line on

⁹If looking at log-prices instead of prices, we can say that the most expensive houses cost approximately 5 (the unit is log \$1000 CAD) while typical houses cost 4 (log \$1000 CAD), i.e. 20% more. But this figure is not robust with respect of measurement units. If we measure the price not in thousands of dollars but in dollars, the most expensive house would be only 10% more expensive than the mean log-price.

This is because the price is ratio measure but log-price is only interval measure. In order to define inequality in this way we have to be able to compute ratios.

the figure). When α gets smaller, the values are more and more spread and contain larger and larger outliers and the values display increasing inequality. The expected value of Pareto RV is

$$\mathbb{E} X = \frac{\alpha}{\alpha - 1} x_0, \quad \alpha > 1. \quad (1.4.20)$$

If $\alpha \leq 1$ then the expected value does not even exist—there are too many too large outliers, so that the sample mean will not converge, and the few richest persons in the sample control almost all wealth.

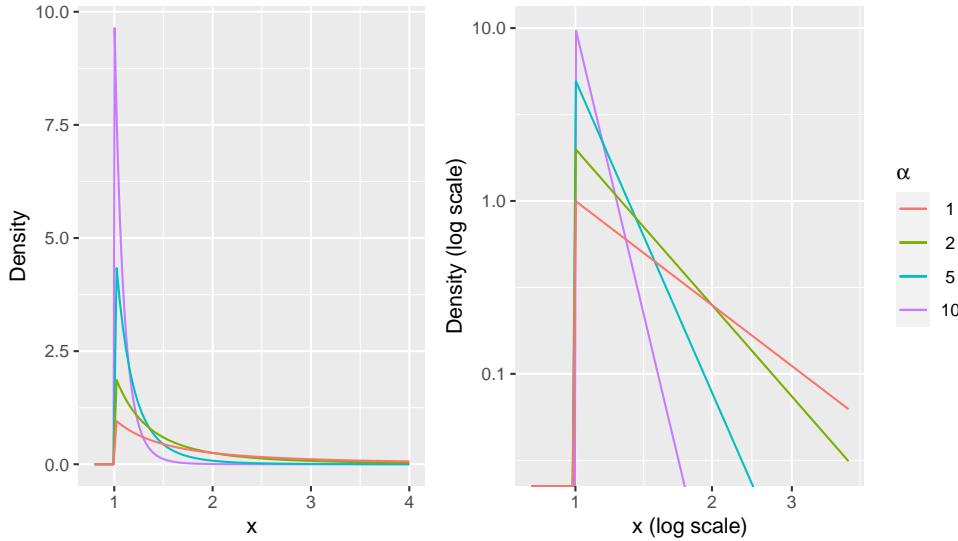


Figure 1.12: Pareto p.d.f for different shape parameter α . Linear scale (left) and the same distributions in log-log scale (right). These distributions may be hard to tell apart on linear scale, but on log-log scale all of them are just straight lines at different angle. All these examples have scale $x_0 = 1$, the cutoff point on the left hand side.

The other parameter, x_0 describes the cutoff point. Pareto distribution needs a cutoff, otherwise it would go to infinity at zero. Sometimes one uses a shifted version of Pareto where the cutoff is shifted to 0. This is called *Pareto-II* or *Lomax* distribution.

An interesting characteristic of Pareto distribution is the fact that the p.d.f looks like a straight line in log-log scale. It is easy to see by taking logarithm of p.d.f (1.4.19):

$$\log f(x) = \log(\alpha x_0^\alpha) - (\alpha + 1) \log x. \quad (1.4.21)$$

One can immediately see that $\log f(x)$ is a linear function of $\log x$ because α and x_0 are constants (see the right panel of Figure 1.12). This gives a good way to tell whether a distribution is more like log-normal or more like pareto: log-normal histogram tends to look like normal in *density*– $\log x$ scale. Pareto tends to look like a straight line in *log density*– $\log x$ scale.

As lines in log-log scale do not have any features, all places on the line always look the same, the distribution is sometimes called *scale free* distributions. In scale-free distribution, wherever you are the picture looks similar: most observations are much smaller, but there are always cases that are much larger. This may explain some of the frustration with career people have: however successful you are, there are always others who are much much more successful. And as you typically socialize with those who are at the comparable level, then you do not notice that you are quite far up in the distribution and most of others are far behind you.

The [80/20 rule](#) can be computed from the equation (given $x_0 = 1$)

$$x_*^\alpha - x_* - 1 = 0 \quad (1.4.22)$$

where x_* is the upper percentage threshold. For instance, if $\alpha = 3$ the upper 43% controls 57% of all resources.

Example solutions are:

α	x_*	upper	lower
3	1.325	43.0	57.0
2	1.618	38.2	61.8
1.5	2.148	31.8	68.2
1.2	3.506	22.2	77.8
1.1609	4.001	20.0	80.0
1.1	5.427	15.6	84.4

1.4.3 Central Limit Theorem

Central Limit Theorem (CLT) plays an extremely important role in statistical inference. It is somewhat similar to Law of Large Numbers, but unlike LLN, our intuition does not help much with CLT. While LLN describes what happens to the sample average when sample size increases, CLT describes the shape of the sum of random variables, and tells that under certain assumptions, sum of RV-s is approximately normally distributed. So if we add up (literally!) a lot of random numbers, the result will be normally distributed. Even more, its variance is proportional to the number of realizations we summed. We first explain and demonstrate CLT at work, and thereafter define it formally and discuss the assumptions behind it.

Why sum tends to be normal

We use Pareto distribution with parameter $\alpha = 5$ ([Section 1.4.2 Pareto distribution](#), page 58) to demonstrate how CLT works. Pareto(5) distribution¹⁰ (Figure 1.13 top left panel) does not resemble normal distribution much, it has its most common values near 0, and the larger values are increasingly less common. In this sense it is more similar to exponential distribution. However, when we start adding up these variables, we can see that the values near zero become increasingly less likely, and the values

¹⁰We use Pareto-II (Lomax) distribution here with the cutoff shifted to 0. Compare with Figure 1.12 at page 59.

near the mean tend to be more and more common. When summing $S = 100$ Pareto(5) RV-s (bottom right panel), the result looks very much like a normal.

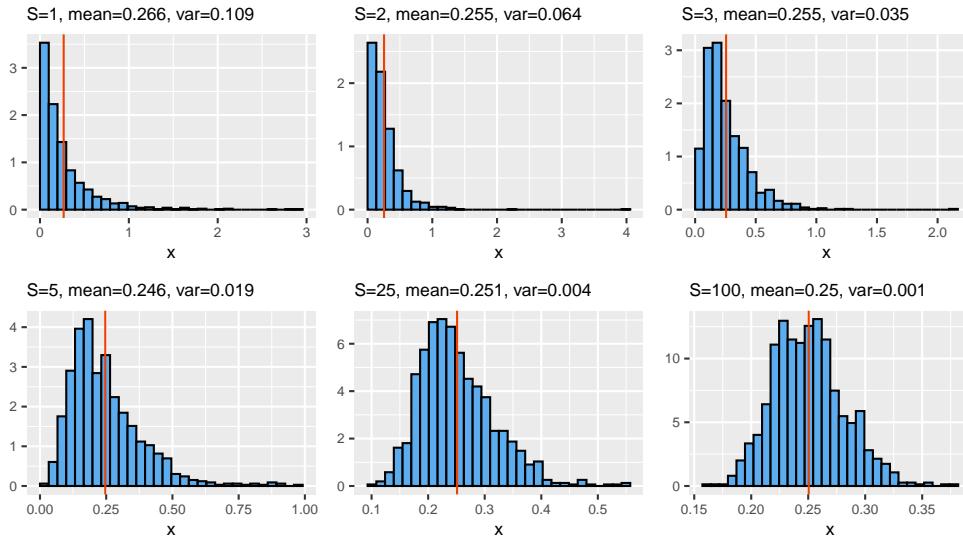


Figure 1.13: Histogram of means of Pareto(5) RV-s. The upper-left panel shows that of a single Pareto(5) RV (i.e. sample size $S = 1$) while the bottom-right panel shows the histogram of means of $S = 100$ Pareto RV-s. The shape of the distribution is getting increasingly normal as S increases. The orange line indicates the average of the sample of means. All figures depict the histogram of 1000 replications.

Before we explain why this happens, let's make clear what exactly do these histograms depict. In the upper-left image we just generate $R = 1000$ random Paretos with shape parameter $\alpha = 5$. The second panel ($S = 2$) shows $R = 1000$ replications of sum of two Paretos. First we generate two Pareto realizations to add them. But just adding two numbers will give us a single number, and we cannot say much about the distribution based on just a single number. So we replicate this for $R = 1000$ times. In the third panel we generate $S = 3$ random Paretos and sum these, and again, in order to see the distribution, we repeat this $R = 1000$ times. And so on, using $S = 5$, $S = 25$ and $S = 100$ random Paretos to sum, and in each case replicating the experiment for $R = 1000$ times. CLT tells us that the distribution will look more-and-more like normal if we add a larger number S of RV-s. The number of replications R is not related to CLT, we use a large R just to get sufficiently smooth histograms. Remember—the distribution is the property of a RV but histogram is a visualization of the sample. Sum of RV-s is also a RV, and its properties are not related to the sample size.

Expected value of Pareto-II distribution is $1/(\alpha - 1) = 0.25$ in this case, and variance is $\frac{\alpha}{(\alpha-1)^2(\alpha-2)} = 0.1042$ here. [Section 1.4.2](#).

But why does the image turn more-and-more normal when we add more RV-s? One might intuitively think that adding Paretos will still result in a long-tailed distribution with maximum near 0. After all, values near 0 are the most likely ones. True, values near 0 are the most common ones, but that is not the whole story. If we

want to take mean of two Paretos and still be close to 0, we need to get *both of these* to be close to 0. But it is more likely to get one of these close to zero and the other not close to zero, instead of getting both of these close to zero. The logic is exactly the same as in case of binomial distribution: when tossing two coins it is more likely to receive one heads and one tails, rather than receiving two heads or two tails. So instead of our average of Pareto pairs piling up at 0, it will pile up at a somewhat larger number. This is what we see on the $S = 2$ panel: the values near 0 are still fairly common, but now the peak at 0 is less prominent than in the first panel.

When averaging more than two RV-s, the same logic applies more-and-more. When computing the sum of 100 RV-s, then we can pretty much guarantee that we get a value somewhere in the middle—in average, we get an average number. Indeed, the chances to get a value near 0 for 100 times is very-very small. This is why we have more and more prominent hump in the middle of the plot—we have arrived close to the normal distribution.

Formal definition of CLT and assumptions behind it Now it is time to look at the formal definition of CLT.¹¹

Theorem 4 (Central Limit Theorem, CLT). If X_1, X_2, \dots, X_S are independent and identically distributed random variables with expected value $\mathbb{E} X$ and variance $\text{Var } X$ then, as the sample size $S \rightarrow \infty$, the distribution of their mean $\bar{X}_S = \frac{1}{S} \sum_{i=1}^S X_i$:

- a) is normally distributed $\bar{X}_S \sim N(\mu, \sigma^2)$, with expected value μ and variance σ^2 .
- b) The expected value of the sample mean, $\mathbb{E} \bar{X}_S$, equals to $\mu = \mathbb{E} X$
(Expected value of average is the same as expected value of individual RV.)
- c) Variance of the sample mean, $\text{Var } \bar{X}_S$, equals to $\sigma^2 = \frac{1}{S} \cdot \text{Var } X$
(Variance is inversely proportional to the sample size.)

For instance, if we take the example of $S = 5$ in Figure 1.13, CLT states that the distribution is approximately normal with mean $1/4 = 0.25$ and variance $0.1042/5 \approx 0.02$. One can see that the distribution somewhat resembles normal, but as $S = 5$ is still far from infinity, the distribution is also visibly different from normal distribution. But when we take $S = 100$ (the bottom-right panel), then our eyes cannot easily distinguish the result from normal distribution any more.

The result b) tells us that we can average a large number of random values and the expected value will still be the same. This is handy when computing—we do not even have to compute.

But the result c) tells us something even more important: the larger the sample size S , the smaller the variance of its mean $\text{Var } \bar{X}$. More precisely, variance of mean is inversely proportional to the sample size. If we measure precision by standard

¹¹Strictly speaking, only the result a) is part of the CLT. b) is law of large numbers, and c) is a direct result of definition of variance for independent random variables. We list here together for compactness.

deviation, the root of the variance, then the precision increases as the root of sample size. Four times larger sample gives us twice as precise results.

It is good to know that CLT is not universally true but relies on a few assumptions. In particular, it applies if

- The random variables X_1, \dots, X_S are independent. This matters for the argumentation we gave above: it must be less likely to get two small values, than one small and one large value. In case of correlated data this may not be true. For instance, when doing 1000 temperature measurements in a hot summer day we should not expect the result to reflect the yearly average temperature well. These values are highly correlated.
- Both expected value and variance of X_i must exist. Although in practice we do not encounter heavy-tailed distributions so often, it is good to know that long-tailed distribution we do encounter may converge at a much slower rate. So when we have encountered only well-behaved variables like number of children or log income, it may come as an unpleasant surprise that our experience does not carry over to an analysis of city size or twitter tweets. The errors are much larger than expected.

Why CLT is so important There are two main reasons why CLT has such a central place in statistics. First, it tells us that when doing computations on large samples, we can use the properties of the normal distribution instead of a huge number of different tailor-made rules for exponentials, binomials and Paretos. This is what we usually use for confidence intervals, t -tables, and so on.

Second, it explains why many natural values, such as human height or temperature are approximately normally distributed. For instance, height is a sum of a large number of factors, some genetic, some environmental, some pulling us taller, other pushing us shorter. But when averaging over all those factors, the typical humans tend to be of about the average height. And when we see a different distribution, e.g. that of human wealth, this indicates that some of the assumptions behind CLT are violated. In case of wealth it is probably independence—the economy seems to work in the way that the rich get richer. Factors that influence wealth are not independent, the wealthy ones seem to be more prone to encounter the opportunities that are even more favorable to wealth accumulation.

Conditional Expectation

TBD: notation, perhaps conditional variance, conditional distributions

1.4.4 Information and Entropy

Entropy is a measure of information in a RV. It is sometime called *Shannon entropy* to distinguish it from a concept of similar name in thermodynamics. Entropy tells how much information do we gain when we learn about the outcome. It can take either value 0 or a positive number. “0” means we do not get any new information—everything was already known in advance; the RV did not contain any randomness at

all.¹² Positive entropy, however, means that there is a certain amount of uncertainty in the result, and learning about the outcome helps to clarify this. The larger the entropy, the more uncertain is the outcome and the more information we gain when we learn about it. As an example, consider two “random” events: *the sun will rise tomorrow*, and *the sun will shine tomorrow*. The first of these is essentially a certain event, so when we see sunrise the following day we’ll learn nothing. Entropy of this RV is 0. The second one, however, depends on the random weather, and hence we learn something when we experience either sun or rain in the following day.

In discrete case where the RV X can take values $k = 1, 2, \dots, K$, entropy is defined as negative expected value of log probability of possible outcomes:

$$\mathbb{H}(X) = -\mathbb{E} \log \Pr(X = x_k) = -\sum_{k=1}^K \Pr(X = x_k) \cdot \log_2 \Pr(X = k). \quad (1.4.23)$$

When using binary logarithm, the entropy is measured in *bits*. When using natural logarithms, the units are called *nats*. It is easy to see that $1 \text{ nat} = \log_2 e = 1.443 \text{ bits}$. We use binary logarithms in this text but some authors prefer natural logarithms.

Note that for zero-probability states, $0 \log 0$ is undefined. However, as the corresponding $\lim_{x \rightarrow 0} x \log x = 0$ (see (A.1.3) in Section A.1.1) the contribution to entropy is 0. This means that when we expand our sample space with additional zero-probability events, the entropy value is not affected. When we are aware of additional kind of events that almost never happen will not affect the amount of information we can gain. Only events with positive probability matter in terms of information.

Example 1.10: Entropy of uniform distribution

In order to have an intuitive understanding about what entropy measures, let us analyze entropy of discrete uniform distribution. In case of K potential outcome states with equal probability $1/K$, (1.4.23) gives:

$$\mathbb{H}(X) = -K \left(\frac{1}{K} \cdot \log_2 \frac{1}{K} \right) = \log_2 K. \quad (1.4.24)$$

So entropy is just logarithm of the number of states. Figure 1.14 displays such an example. We have 8 possible states, $A-H$, some of which have probability 0 (white on figure) while the others are equally likely. The first row depicts the case where all the probability mass is concentrated in the state A with $\Pr(A) = 1$ while every other state S has $\Pr(S) = 0$. Here we cannot gain any information as we know in advance that A happens for sure. This is reflected by the corresponding $\mathbb{H} = 0$. However, the next state already contains some uncertainty: we don’t know if A or B will happen, both are equally likely. When we learn about which event happened, we gain $\log_2 2 = 1$ bit of information. Further down in the table, there are more possible states and hence more uncertainty, and we gain more information when we learn about the outcome.

¹²An example of such a “non-random” RV (degenerate RV) is *get value 0 with 100% of probability*.

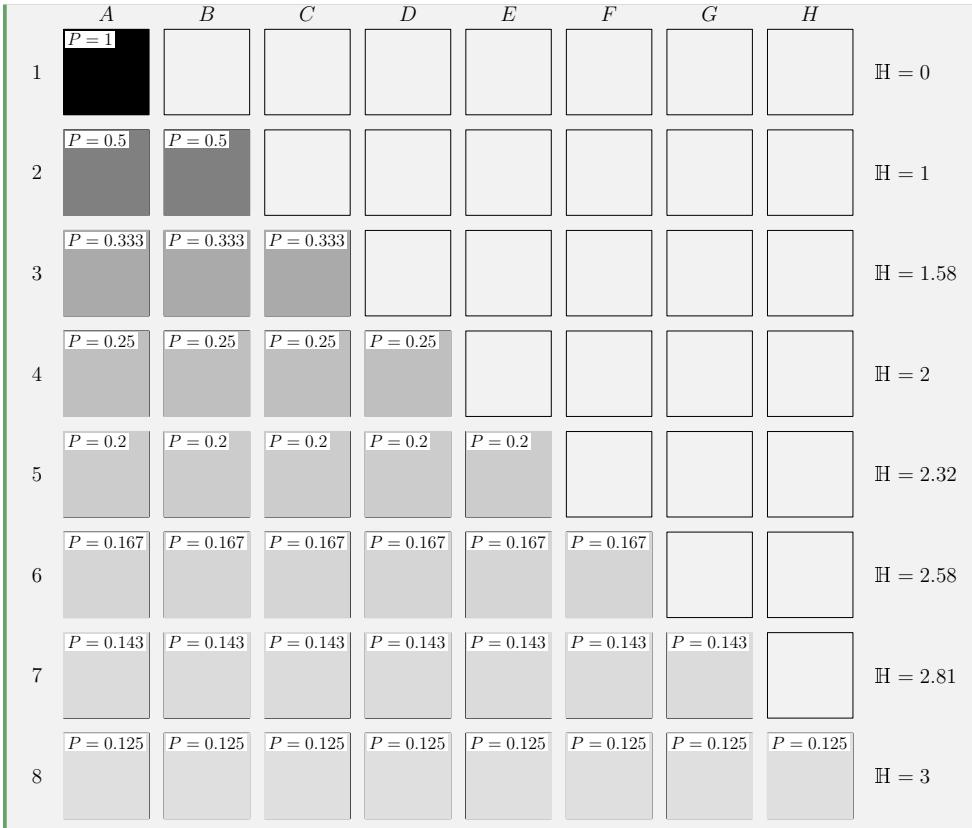


Figure 1.14: Entropy in case of different probability over states. Different shades of gray denote different probability, uniformly spread over 8 states A–H. The rightmost column is the corresponding entropy.

Exercise 1.18: Entropy of Bernoulli random variable

Look at a Bernoulli process with parameter p . Compute it's entropy as a function of p . Explain the intuition behind how the result depends on p .

Hint: consider making a plot on computer.

1.5 Statistical Inference

Statistical inference refers to statistically sound conclusions based on data that *can be generalized* to the whole population. The statistical methods we discussed in [Section 1.2](#) are sound, but do not allow generalizations. Descriptive statistics describes data using statistical tools. For instance, we may find that in our sample, mothers who smoke give birth to smaller babies. If this is all we are interested in, we can stop here. But can our data tell something about another sample of mothers and babies? Or about *all* mothers and babies? Yes, the current sample can tell something about other samples, and about the whole population. Inferential statistics does exactly that.

1.5.1 Statistical Hypotheses and Hypothesis Testing

In this section we introduce a lot of concepts: confidence intervals, confidence levels, significance levels, statistical hypothesis and hypothesis testing, and different types of errors. This section just introduces the concepts, how to actually compute the confidence intervals is discussed in Sections [1.5.2](#) and [1.5.3](#).

Hypothesis testing and confidence level

Statistical inference is typically done through statistical hypotheses and hypothesis testing. Statistical hypotheses are claims about the world, claims that may or may not be compatible with data. If data contradict the claim, we *reject the hypothesis*, if data are compatible with the claim, we do not reject the hypothesis. Unlike many other fields, statistics normally does not give definite answers. For instance, while economists typically want to say that unemployment rate is “11.2 pct”, a statistics-based answer will include a measure of uncertainty. A statistically correct answer may be “with 95% confidence we can say that unemployment rate is between 10.8 and 11.6 pct”.¹³ Such answers are pretty much the only type of results that statistics can offer.

What does such a claim mean? There are two components here:

- We don’t know what exactly is the unemployment rate, but we are “reasonably certain” it belongs to the interval [10.8, 11.6].
- The “reasonably certain” means we are 95% certain.

As one can see, understanding such somewhat fuzzy claims requires a bit of statistical literacy.

Intervals like [10.8, 11.6] are very commonly reported in statistical practice and hence they have a distinct name—*confidence intervals*. If we are 95% confident that the true value falls into this interval then we call it “95% confidence interval”. See [Section 1.5.2 Doing Statistical Inference](#), page [71](#) for how to compute confidence intervals.

A *statistical hypothesis*, often denoted as H or H_0 , is a claim, usually stated in a definitive manner. In the example above, a hypothesis might be H_0 : “unemployment

¹³You may have noticed that such interval-based claims are also quite common in sciences. Often they originate from statistics-based methods.

rate is 11.2%”, or perhaps instead H_1 : “unemployment rate is more than 10%”. The hypotheses can either be *rejected* (it means it is incompatible with data) or not rejected (if it is compatible with data). Note that hypotheses cannot be confirmed! Hypotheses can only be rejected or not rejected at certain *confidence level*. Confidence level means the probability that the rejection is the correct decision. It is based on data quality, sample size and other factors. In applications we typically look at confidence levels 95% or 99%. If a hypothesis is rejected, we can consider it “wrong”—given our data and methods was correct. If a hypothesis is not rejected, it is compatible with the data. It may be correct, or it may still be wrong if our dataset is too small or too noisy.

Hypotheses are often presented in pairs, where one is called *null-hypothesis* H_0 and the other *alternative hypothesis*, often denoted by H_1 or H_A . The null hypothesis is the original claim we are testing and potentially rejecting, H_A is the alternative that must be true when H_0 is false. For instance, when analyzing mothers’ smoking habits and babies birth weight, H_0 might be “smoking and non-smoking mothers give birth to babies of equal weight in average”. The alternative H_A in this case will be “Birth weight of babies, born to smoking and non-smoking mothers, differs in average”. Note that H_A does not claim that babies of either one or another group weigh more. If the weight is not equal, it must differ. If one wants to test if the weight differs in a certain way, that is a separate hypothesis. While certain sources always state H_0 and H_A explicitly, other studies either only discuss H_0 or leave the hypotheses implicit. In that case it is often obvious from question that is analyzed.

Example 1.11: Rejecting and not rejecting a statistical hypothesis

Assume we analyze unemployment data and conclude that with 95% confidence the true unemployment rate is between 10.8 and 11.6 pct with the best estimate being 11.2.

Now consider the government, that always prefers to paint a bit more rosy picture, claims that the rate is just 8.9%. We can treat this as a statistical hypothesis $H_0 : u = 8.9$ (where u means unemployment rate). The government’s claim is clearly incompatible with our data (and model), after all, according to our analysis, we are 95% confident that the rate is at least 10.8%. So we can reject H_0 and accept the alternative $H_A : u \neq 8.9$. But note that we were just 95% certain that $u \in [10.8, 11.6]$ and hence we cannot reject it definitively, but only with 95% confidence.

However, the politically independent Central Bank has no incentive to make the figures any better than they are and publishes its own analysis according to which $u = 11.4\%$. This can also be written as a statistical hypothesis $H_1 : u = 11.4$.^a What can we say about this? The number 11.4 fits squarely inside our confidence interval and hence the result is compatible with our data. So we cannot reject H_1 . But neither can we tell that it is correct—it is just compatible with data, maybe correct, maybe not correct.

^aHere we use H_0 and H_1 to denote different hypotheses. H_1 here is not the alternative to H_0 , the alternative to the latter is $H_A : u \neq 8.9$ above.

Which hypotheses can be rejected depends on data quality—how much relevant information there is in data, and on the analysis—how well can we extract that information. The lower the data quality, the less can we tell, the fewer hypotheses can we reject. In the extreme case where the data contains no information (or we do not have any data), we cannot reject anything.

Example 1.12: Unemployment example with bad data

Now imagine we only have access to inferior data and our results are much less precise. We are only able to conclude that unemployment must be in a range of $[7, 14]\%$. Can we now reject $H_0 : u = 8.9$ (the Government's claim) and $H_1 : u = 11.4$ (the Central Bank's claim)? As both of these fit into our interval, we cannot reject either of them. Both claims are compatible with our low quality data. But we can still say that a scaremongerer who claims $u = 30$ is wrong.

However, if we do not have any data, the only thing we can say is that $u \in [0, 100]$ (this must be true by definition of the unemployment rate). Now even the scaremongerer can go unopposed, we just have no way to evaluate that claim.

Confidence level, significance level, and p -value

Significance level is the mirror image of confidence level. It is the probability that we reject H_0 even if it is correct (type-I error, see below). We normally want this number to be small. Significance level is frequently denoted by α and often chosen to be $\alpha = 0.05$.

Significance level is *not* something computed from data. It is a choice that should be done before beginning the analysis: when are we willing to say that the hypothesis is not compatible with data and reject it? If our data and model suggest that H_0 is only 1 percent likely, are we willing to reject it? What if it is 10% likely?

Hypothesis testing is typically done by computing a *test statistic*, such as t -value or F -value. If H_0 is correct, the test statistic tends to have certain kind of values, often small values near zero. But if H_0 is not correct, the test statistic will have other, “more extreme” values. But as we are working with random processes, such extreme values may also occur even if H_0 is correct, just it is not that likely. This is the idea of p -value. p -value is probability that at a test statistic value that is at least as extreme than you see in data, is observed even if H_0 is correct. If we want to reject H_0 , we must have p -value smaller than the significance level, $p < \alpha$. So p -value is about *observing test statistic* that is as extreme as what you see in data.¹⁴

Example 1.13: Significance and p -value

Imagine we are analyzing whether smoking is related to birth weight. We collect data about mothers' smoking behavior and their babies' birth weight. We choose

¹⁴Sometimes people understand p -value as “the probability that H_0 is correct”. This is not quite right. But for someone who are just getting into statistics, it is often a good enough definition of what p -value means. So if you cannot remember the correct definition, try to remember this, and be aware that “there was something more in it”. Let not the perfect be the enemy of good!

H_0 : “mother’s smoking is not associated with birth weight”. We also have to decide a significance level, here $\alpha = 0.05$ is an appropriate choice.

A suitable test statistic is t -statistic (see [Section 1.5.3](#)). t -statistic behaves in the way as described above—if H_0 is correct, we expect to see mostly small values, and only rarely large values. Say, we find $t = 2.8$. From the t -value table we can find that the corresponding p -value is 0.006. This means that if H_0 —there is no relationship—is correct, the probability to see a t -value 2.8 or larger is just 0.006. As this probability is less than our chosen significance level $\alpha = 0.05$, $p < \alpha$, we reject H_0 and conclude that smoking and birth weight are related.

However, if we find $t = 1.8$, the table suggests $p = 0.075$ instead. This means we have 7.5% probability to observe this large number even if H_0 is correct. As now $\alpha < p$, we cannot reject H_0 , so we cannot conclude that smoking and birth weight are related.

It is important to choose significance level before the analysis. Otherwise one may inadvertently adjust the level up or down, depending on how the p -value turns out in data, and what is the desired outcome.

Type-I and Type-II errors

Our hypothesis testing can go wrong in two ways:

1. We reject H_0 even if it is correct. These errors are called *type-I errors* or *false positives*.
2. We fail to reject H_0 even if it is incorrect. Such errors are called *type-II errors* or *false negatives*.

The words “positive” and “negative” are easy familiar in the context of medical diagnosis, e.g. with COVID tests. Test being “positive” means it indicates the patient has the disease, negative test means no sign of the disease. But no test is perfect and sometimes a person who does not have COVID will receive a positive result. This is type-I error or false positive. In an analogous fashion, if the test fails to discover COVID (it comes back negative despite the patient having COVID) then we made a type-II error (false negative).

There is always a trade-off between type-I and type-II errors. If we pick very low confidence level, we immediately reject H_0 as soon as it does not look quite right, even if the reason is just random noise in our data. We do a lot of type-I errors but few type-II errors. In contrary, if we pick a very high confidence level, we often fail to reject H_0 even if it is wrong. We do many type-II errors but very few type-I errors. In the extreme case where we pick confidence level 0, we reject all H_0 -s, and if we pick confidence level 1, we never reject anything. The optimal choice, obviously, is somewhere in between. How we want to balance between false positives and false negatives depends on the associated costs. If false positives are cheap but false negatives expensive, we want to use a low confidence level to avoid false negatives. If the opposite is the case, we set confidence level very high.

TBD: Example

TBD: some sort of literature example where one chooses different confidence levels for different error costs.

Cheatsheet 1.5: Summary of the concepts

The statistical inference section above introduced a large number of concepts. Here is a summary of the most important ones.

Null hypothesis a claim about the world we want to test, usually by trying to “reject” it based on data. We reject it if it is incompatible with data. Often denoted by H_0 .

Confidence interval is the interval where the true value most likely belongs. If we are 95% certain that the true value is between 10.6 and 11.8, the we say that “95% confidence interval is [10.6, 11.8]”.

Confidence level is the probability that you do not falsely reject H_0 , often chosen to be 95%.

Significance level is the probability that you do falsely reject H_0 . It is often chosen to be 5% and denoted by α .

Test statistic a number computed from data. If H_0 is correct, then the test statistic value is typically small and it is unlikely to find large values.

p-value probability to find test statistic at least as extreme (as large) if H_0 is correct. This means if p-value is small, H_0 can be rejected.

It is sometimes understood as “probability that H_0 is correct”, this is not quite right.

Confidence interval A region where the parameter of interest lies with a certain confidence level. For instance, “with 95% confidence, the population mean is between 7 and 8”.

Type-I errors (false positives) erroneously rejecting H_0 .

Type-II errors (false negatives) erroneously not rejecting H_0 .

1.5.2 Doing Statistical Inference

In the previous section we talked a lot about confidence intervals, confidence levels, and p -values. But we did not discuss how can we actually compute those figures. The central task in this section is to do exactly that, namely to devise methods to compute the confidence intervals. We start with a somewhat trivial task, namely making statistical claims about individual random variable realizations. This helps us to build the necessary machinery for more complex problems later.

Consider a RV X . What would be a statistically sound claim about its realization x ? Imagine X is the temperature tomorrow, and tomorrow we will learn its actual value (realization) x . Obviously, today we still do not know the correct value. But we can still say something like “with 95% confidence the temperature tomorrow will be between 14 and 17 degrees”. How can we find the boundaries 14 and 17, and where is the 95% coming from?

The Simulation Approach

Let's start with the “95%” first. This is the confidence level, the mirror image of significance level. We have to decide it before the analysis. In the current case, the confidence level means the probability that our prediction is correct, so in 95% of cases, the temperature fall in the $[14, 17]$ interval. This is the 95% confidence interval for our weather forecast. We can imagine an implicit H_0 : Tomorrow we will have ... degrees. However, prediction here is not just a single number but a possible range.

Confidence level is the probability that rejecting H_0 is the correct decision. See [Section 1.5.1 Hypothesis testing and confidence level](#), page 66, and Cheatsheet 1.5.

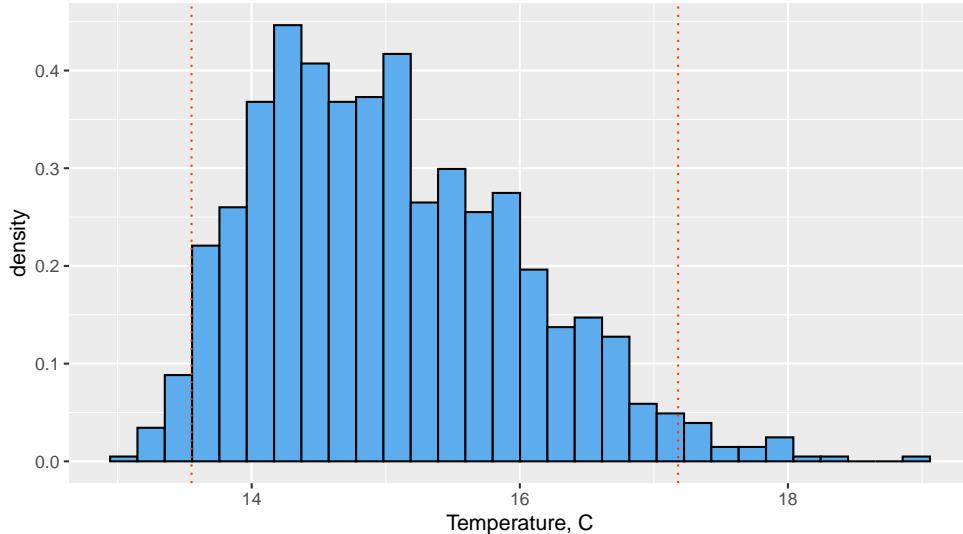


Figure 1.15: Temperature distribution from a hypothetical weather model. The expected value is 15.1, the central 95% confidence interval is between the dotted vertical red lines.

All these values are uniquely determined by the distribution of X . In case of weather forecast, the distribution is originating from the weather model we are using. Such models include random components to account for the fact that we cannot predict weather perfectly. Assume we run the model 1000 times and receive 1000 temperature predictions as in Figure 1.15. The distribution is somewhat right skewed, and predicts the temperature between 13 and 19 degrees with the expected value being 15. But other values than 15, such as 14 and 16 also look perfectly feasible. But which values do we consider feasible? For instance, 18 degrees seems to be unlikely, and even warmer weather seems even less plausible.

A common answer to this question is to look at the central 95% of the predictions. Central 95% of the observations means that we consider the leftmost (the coldest) 2.5% and the rightmost (the warmest) 2.5% predictions to be unlikely. Given 1000 predictions, we can just remove the 25 coldest and 25 warmest predictions, and we are left with the central 95% of the predictions. This is equivalent to computing the 0.025-th and 0.975-th sample quantiles. In case of this sample, the corresponding quantiles are 13.55 and 17.18 degrees (dotted vertical lines on the figure). This interval, [13.55, 17.18] is called *confidence interval* (CI), more precisely 95% confidence interval for the predicted temperature. So 95% CI is the interval that contains the actual value with 95% probability, given that our weather model is correct. The values in this range are considered likely, and those outside this range are considered unlikely. So we may say that +16 degrees will be quite likely but +18 will be unlikely. We can reject H_0 : “temperature will be over 18C” at 5% significance level.

As the 95% CI are only correct 95% of time, one may be tempted to improve on the type-I error and report 99% or 99.9% CI instead. This is fair, but unfortunately the result will be less informative. If I say that temperature tomorrow will be between -100C and +100C then I am correct 100% of time (well, at least on Earth). However, such a prediction does not help us to make any practical decisions, such as what to wear tomorrow. This is the trade-off between providing precise estimates and avoiding errors. 95% is often a good confidence level, but sometimes one may use a much higher level. But information that is sometimes incorrect is better than a claim that is always correct but devoid of any usable information. Sometimes one can compute the optimal confidence level by considering the cost of type-I (false positives) and type-II (false negatives) errors, and choosing a confidence level that leads to the smallest overall loss.

But why did we select the central 95% interval as our confidence region? Why not leave out the largest 5% and pick the smallest value till 0.95-th quantile as the confidence region? Or the other way around? And what about picking both extremes and leaving a narrow 5% gap in the middle? There are a few reasons for this, some of those more theoretical, some more practical.

- First, we are usually interested in a confidence region that is as concentrated as possible: we want the 95% of possible outcomes to have a small error margin. The narrower my temperature forecast, the better idea you have what to wear tomorrow. The obvious choice is to pick the region on the histogram with highest chances and not to leave a gap in the middle where the values are most likely.

- Second, in typical applications the distribution is symmetric and unimodal (usually close to normal). Both tails are thin and it makes sense to cut the 2.5% of observations in both tails if we want to get the most concentrated confidence region. But if it is not symmetric, we may actually choose a different percentages in different tails.

TBD: Example of asymmetric CI

- But what about cutting off the top 5% of temperature predictions instead of the middle range? This is sometimes the desired approach. If you want to know whether the temperature will be *no warmer than 17 degrees*, then you may look at *one-tailed confidence interval* and compute the probability that the temperature will be below that threshold. Or doing this the other way around, you may find the threshold temperature that our predictions will not exceed with 95% probability. But note we do not care about how cold can weather be in this case.

If the distribution is bimodal, we may actually want to leave a gap in the middle.

TBD: Example of bimodal CI

TBD: Example to compute the loss and CI based on the loss

Theoretical Confidence Intervals

Next, we discuss how to compute confidence intervals theoretically for certain important cases. In order to do it, we typically have to know the stochastic process that generates our data (the statistical model), and based on that we can often compute the theoretical quantiles. Sometimes this can be calculated easily (e.g. for uniform distribution), sometimes one has to consult tables (e.g. for normal and t -distribution).

Confidence interval for normally distributed values For standard normal RV $X \sim N(0,1)$, the lower 2.5% quantile $q_{0.025} = -1.96$ and the upper 2.5% quantile is $q_{0.975} = 1.96$ ($q_{1-\alpha} = -q_\alpha$ because standard normal is symmetric). If X follows a general normal distribution, $X \sim N(\mu, \sigma^2)$, with expectation equal to μ and variance σ^2 , the corresponding quantiles are

$$q_{0.025} = \mu - 1.96\sigma \quad \text{and} \quad q_{0.975} = \mu + 1.96\sigma. \quad (1.5.1)$$

Where is the “1.96” coming from? It is just a property of normal distribution: for standard normal, 95% of cases are between value -1.96 and 1.96 . Nowadays, many statistical software packages provide function to easily compute normal quantiles. But these can also be looked up in the tables. Table 1.8 shows a typical *t-value table*. It displays the significance level α in columns and *degrees of freedom df*. As we work with the normal distribution here, we ignore most of the table and just note that $df = \infty$ corresponds to the normal distribution quantiles in the *t-value table*. That row is also called *z-values*, so *z-value* is the same with *t-value* with infinite degrees of freedom. (See Section 1.5.2 Degrees of freedom and t -distribution, page 77 for more about degrees of freedom and *t*-values.) When you need to know the critical *z* value, you can pick it from the table from the column that corresponds to your

$X \sim N(0,1)$ means that RV X is normally distributed with expected value 0 and variance 1.
 $X \sim N(\mu, \sigma^2)$ means that RV X is normally distributed with expected value μ and variance σ^2 .

Significance level is probability that you falsely reject H_0 . See Section 1.5.1 Confidence level, significance level, and *p*-value, page 68.

desired significance level α . In case of normal distribution, this will be the lowermost line, and for 5% significance level ($\alpha = 0.05$), the critical value is 1.96. This means 95% cases fall between -1.96 and 1.96 . However, if we want to capture 99% cases, we have to pick $\alpha = 0.01$ and the corresponding $t_{cr} = 2.58$.

	Two-tailed significance level α						
df	0.2	0.1	0.05	0.01	0.005	0.001	
10	1.37	1.81	2.23	3.17	3.58	4.59	
20	1.33	1.72	2.09	2.85	3.15	3.85	
50	1.30	1.68	2.01	2.68	2.94	3.50	
100	1.29	1.66	1.98	2.63	2.87	3.39	
200	1.29	1.65	1.97	2.60	2.84	3.34	
∞	1.28	1.64	1.96	2.58	2.81	3.29	

Table 1.8: Critical t -values. The lowermost line with $df = \infty$ corresponds to z -values, the normal quantiles.

Example 1.14: Confidence intervals for human height

Look at the [fathers' and sons' height data](#), in particular sons' heights. The distribution as a histogram is shown on the figure below, overlapped with approximated normal density curve (red).

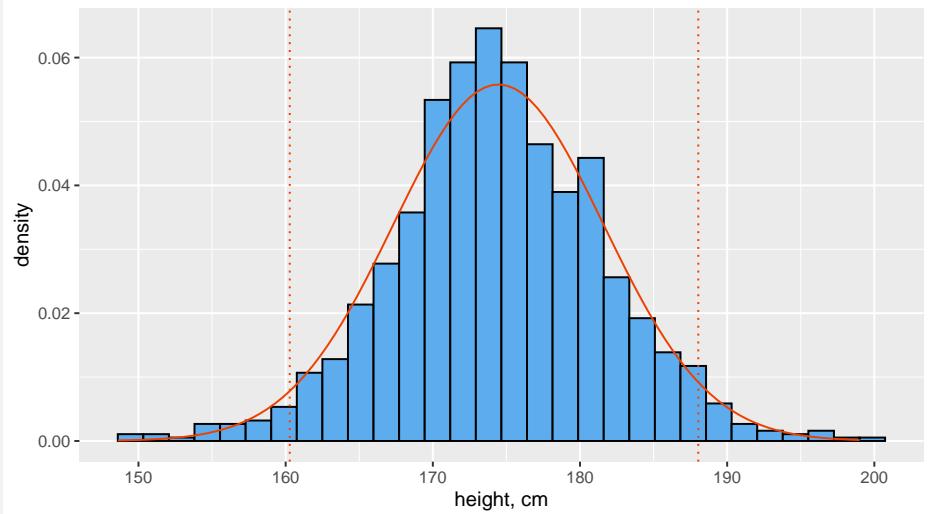


Figure 1.16: Distribution of son's height in fathers-sons data. It is well approximated by normal distribution (red line) with mean $\mu = 174.5$ and standard deviation $\sigma = 7.2$, this is common for measures of adult animals. Dotted vertical lines are the boundaries of the 95% confidence intervals.

As evident from the figure, most of observations are concentrated around the mean 174.5 cm, roughly in the interval of 160–190cm. More precisely, the lower 2.5% observations are shorter than 160.3 and the upper 2.5% of observations are taller than 188 cm. This means that the middle 95% of the observations fall into the interval [160.3, 188]. This is the 95%-confidence interval (CI) for sons' height. If data were exactly normally distributed with a similar mean and standard deviation, the corresponding theoretical quantiles were $q_{0.025} = 174.5 - 1.96 \cdot 7.2 = 160.4$ and $q_{0.975} = 174.5 + 1.96 \cdot 7.2 = 188.5$. As one can see, the deviations from the theoretical values are rather small. Normal distribution is a good approximation for human height.

Confidence interval for sample mean Now we move to a more important task, namely finding the confidence interval for the mean of N random variables. Why is this a more important task?

Remember, we use a sample of N datapoints (realizations) to learn something about the underlying process, the RV. If N is large, we can easily finds the sample quantiles—the CI. But the sample only has a single mean. True, we can easily compute it, but our primarily interest is to *learn about the RV, not about the sample*. Hence, after all, we are not interested in the sample mean but in the expected value of the underlying RV. For instance, imagine that you are conducting a poll of 1000 voters before the elections. You find that 520 of them prefer liberals and 480 prefer conservatives. We are not interested in the sample average 0.52, we are interested in who will win the elections. This is the expected value of the RV, the preferences of the whole electorate. We want to use our sample to find the CI for its expected value.

Because sample only has a single mean, we cannot compute its confidence interval through quantiles. Consider a sample of size N of RV X . Denote its mean by μ and variance by s^2 . From Central Limit Theorem we know that a) the mean \bar{X} of RV-s tend to be normally distributed; and b) the variance of sample mean is inversely proportional to sample size,

$$\text{Var } \bar{X} = \frac{1}{N} \text{Var } X \quad (1.5.2)$$

where \bar{X} denotes the mean of a sample of size N . We also know that in a large sample, the sample average tends to be close to the expected value $\mathbb{E} X$, and sample variance tends to be close to the variance $\text{Var } X$. If we put these two facts together, we have that for the sample mean

$$\bar{X} \sim N\left(\mu, \frac{s^2}{N}\right), \quad (1.5.3)$$

Accordingly, from (1.5.1), the 95% confidence intervals of a mean of normals is

$$\left[\mu - 1.96 \frac{\sigma}{\sqrt{N}}, \mu + 1.96 \frac{\sigma}{\sqrt{N}} \right]. \quad (1.5.4)$$

See [Section 1.4.3 Formal definition of CLT and assumptions behind it](#), page 62.

Example 1.15: Sample mean of sons' height

Let's look again at the sons' height data. As a reminder, the mean sons' height is 174.5 and its standard deviation is 7.2. Now we take 1000 times a random sample of 4 sons and calculate their mean height.^a In this way we get 1000 sample means, and we plot the results on a similar histogram as in Example 1.14. Note that we have to take many samples in order to compute many sample means, and be able to plot their distributions.

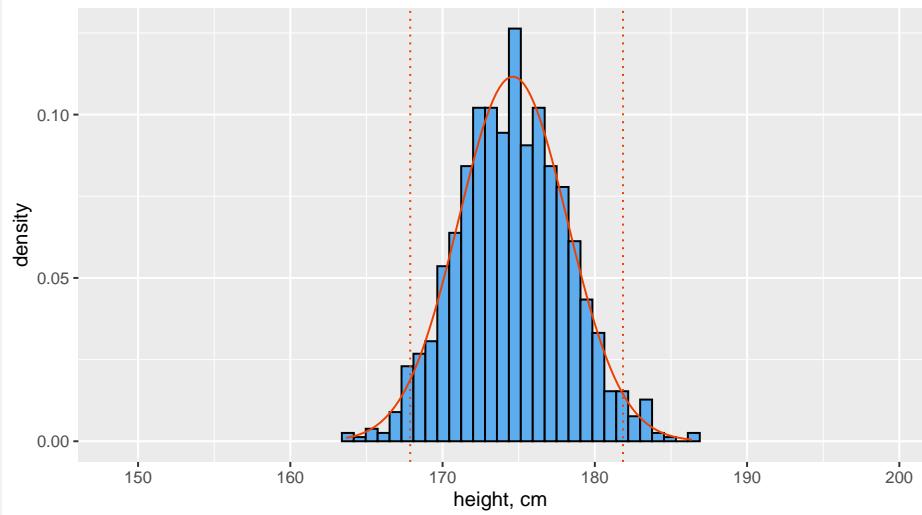


Figure 1.17: Distribution of mean height of four sons. The distribution of means are well approximated by normal (red line) with mean $\mu = 174.6$ and standard deviation $\sigma = 3.6$. Dotted vertical lines are the boundaries of the 95% confidence intervals.

The figure is plotted in a similar scale as Figure 1.16. It reveals that the result is well approximated with a normal density, however this time the spread is narrower. The sample of means has mean value $\mu_m = 174.6$, almost exactly the same as the sample of individual heights $\mu = 174.5$. Its standard deviation $\sigma_m = 3.6$ is only half of that for the heights, $\sigma = 7.2$. This is to be expected as the standard deviation of a sample of four should be $1/\sqrt{4} = 1/2$ of the standard deviation of individual values. Accordingly, both the empirical confidence intervals [167.9, 181.9] and theoretical confidence intervals are only half as wide, [167.6, 181.6].

^aCLT says that distribution of average of independent random variables of all kinds tend to be normal if $N \rightarrow \infty$. However, if X is normally distributed, the result is *exactly* normal even if N is small. So average of just four heights here will result in a distribution that is very close to normal.

Degrees of freedom and t -distribution

When adding two normal RV-s, the result is normally distributed. But here is a “but”: it is normally distributed if we *know* its expected value. This is a nice theoretical result but unfortunately it has little practical value. It is rare we know the expectation, much more likely we have to *compute it from data*.

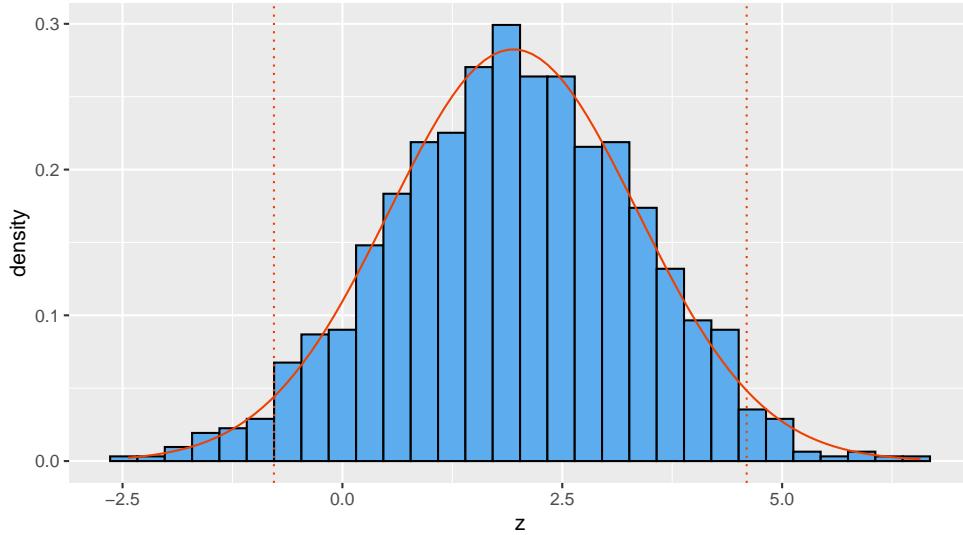


Figure 1.18: Testing $H_0: Z = X_1 - X_2 = 0$ where $X_1 \sim N(2, 1)$ and $X_2 \sim N(0, 1)$. The histogram displays the distribution of Z for 1000 replications.

Figure 1.18 exemplifies the problem. We create two random normals from different distributions: $X_1 \sim N(2, 1)$ and $X_2 \sim N(0, 1)$, and analyze their difference $Z = X_1 - X_2$.¹⁵ The histogram looks quite similar to normal, as confirmed by the orange normal density curve with mean 1.94 and standard deviation 1.41, close to the theoretical results $\mathbb{E} Z = 2$ and $\sqrt{\text{Var } Z} = 1.414$. All seems well.

But this image is somewhat misleading. It depicts the distribution of 1000 different trials while in practice we are normally left with results of a single experiment. So instead of testing the hypothesis on a sample of 1000 results, we need to test if 1000 times on a single result, and see how often we are correct.

But now it turns out the result is not normally distributed any more, but rather t -distributed. t -distribution is pretty similar to normal. We typically use “standard t ” distribution which is similar to standard normal. Standard t has a single parameter: *degrees of freedom*, usually denoted by df . Degrees of freedom is in a way a measure of information content in data, how many data points we have that actually carry information. It turns out that after computing the mean, we are left with $N - 1$ df, instead of the original number of observations N . This is because if you know the

¹⁵Sum and difference are very similar: instead of analyzing the difference, we can look at sum as well—as X_2 is symmetric around zero, the distribution of Z will be unaffected.

mean, you can always compute the value of the N -th data point, if you are given the values of all the $N - 1$ data points.

So the result of adding normals when you do not know the expected value is not normally distributed, but t -distributed with $N - 1$ degrees of freedom. As one may guess, if sample is getting large then the extra information you lose because you have to compute mean from the sample is of less and less importance, and for a large degrees of freedom (large sample) t -distribution converges to normal.

TBD: CLT, and the fact we don't have to start with a normal

1.5.3 Comparing Distributions

One of the most common tasks that leverages statistical inference is to compare distribution of certain variables across different groups, datasets or time periods. For instance, can we say that police treats African-Americans differently than white Americans? We can easily compute how often either group is stopped by the police and compare those figures, but aren't those numbers just a statistical blip, just random noise that will go the other way the next day?

In order to answer this and other similar questions we have to approach it through formal statistical hypothesis testing. We model the “treatment” as RV-s: police treating on group is one RV X and police treating the other group is another RV Y . We want to know if both groups are treated similarly, i.e if $X = Y$. But we do not have access to information about the underlying treatment, X and Y . What we can observe is just samples of realizations: how were members of one group and the other group treated. We have to work with samples.

In practice, it is hard to test if two random variables are equal, if $X = Y$. It is easier to test if certain characteristics, such as expected values, are equal. Hence we set our null hypothesis to be H_0 : $\mathbb{E} X = \mathbb{E} Y$. This can be worded as “the police treats both groups in a similar manner, in average”. Next, we can look at data (samples of realizations) and see if we can reject the H_0 . This can be done using t -test.

Obviously, if the RV-s are similar, then they should result in samples with similar average. This is often enough for applied work. The cases that are relevant for applications tend to have fairly similar distributions. However, be aware that this is just a necessary but not sufficient condition: RV-s may still differ even if both have similar expected value. For instance

$$X = \begin{cases} -1 & \text{with probability } 0.5 \\ 1 & \text{with probability } 0.5 \end{cases} \quad \text{and} \quad Y = \begin{cases} -2 & \text{with probability } 0.5 \\ 2 & \text{with probability } 0.5 \end{cases} \quad (1.5.5)$$

have both expected value $\mathbb{E} X = \mathbb{E} Y = 0$ and hence they generate samples where mean tends to be 0. But they are obviously different.

Comparing Proportions: Is Smoking Ban Associated with Less Smoking?

One simple but widely used average is sample proportion. Below, we look at the proportion of smokers in two samples. Smoking is a very simple RV—someone either

is a smoker or not.¹⁶ Hence smoking can be described with a Bernoulli RV, and the only parameter we need to compute is the proportion of smokers.

Sample proportions are rather important in the applied work. For instance, we may want to test what proportion of patients recovered depending on the care they got; whether workplaces that offer certain amenities have more female workers; or whether there are more years of major heatwaves in 21st century. All these questions can be analyzed in a similar manner as what we do below.

Bernoulli: “success” occurs with probability p , “failure” with $1 - p$. See [Section 1.4.1 Bernoulli Distribution](#), page 49

In the Western World, regulations about smoking have become increasingly restrictive over the recent decades. In particular, smoking is banned in many common indoor areas, such as restaurants or workplaces. *SmokeBan* data (AER package in R) provides data for 10,000 workers who work either on a workplace with or without smoking ban, and who are either smokers or non-smokers. A simple analysis suggests that 6098 workplaces with smoking bans are associated with a lower percentage of smokers, 21.2 percent, while 28.96 percent of workers smoke on 3902 workplaces without such a ban. So the difference is 7.76 percentage points. As we have quite a large sample—10,000, it suggests that the effect is real, not just a random fluke in our sample. But is it really the case? Maybe it is just random sampling noise that will be reversed in the next survey? Let us answer this question first by simulations, and thereafter by stock t -test.

Our task is to compare two samples: workers on jobs where there is a workplace smoking ban, and workers on jobs with no such ban. This is called *unpaired* data because there is no obvious correspondence between individuals across the samples. We can answer the question by testing H_0 : average percentage of smokers in both types of workplaces is the same.

Let us begin by introducing some notation: let S_i denote the smoking indicator of individual i (0: does not smoke, 1: smokes). Let \mathcal{B} denote the set of those individuals who are working at smoking-ban workplaces, and \mathcal{N} the set of those who work at no-ban workplaces. Define

$$\bar{S}_{\mathcal{B}} = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} S_i \quad \text{and} \quad \bar{S}_{\mathcal{N}} = \frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} S_i \quad (1.5.6)$$

For a set \mathcal{S} , the $|\mathcal{S}|$ denotes the number of elements in that set. See more in [Section 0.1 Notation](#), page vi.

for average smoking rate at the smoking-ban and the no-smoking-ban workplaces. We will test the hypothesis $H_0 : \bar{S}_{\mathcal{B}} = \bar{S}_{\mathcal{N}}$, or equivalently, $H_0 : \bar{S}_{\mathcal{B}} - \bar{S}_{\mathcal{N}} = 0$. Assume H_0 holds, i.e. there is no real difference between these two types of workplaces. In that case the expected values $\mathbb{E} S_{\mathcal{B}} = \mathbb{E} S_{\mathcal{N}} \equiv \mathbb{E} S$ where $\mathbb{E} S$ is the expected value of overall smoking probability at all workplaces. We can approximate the latter by sample mean as

$$\bar{S} = \frac{1}{N} \sum_i S_i \quad (1.5.7)$$

Remember: even if expected values are the same $\mathbb{E} S_{\mathcal{B}} = \mathbb{E} S_{\mathcal{N}}$, the sample averages $\bar{S}_{\mathcal{B}}$ and $\bar{S}_{\mathcal{N}}$ may differ. See [Theorem 1, Law of large numbers](#) on page 42.

where the sum is taken over the whole dataset, i.e. over both smoke-ban and no-ban workplaces. In *SmokeBan* dataset the corresponding value is $\bar{S} = 24.23$ percent.

¹⁶The real world, obviously, is more complicated. One may be either casual or heavy smoker, or maybe just recently quit smoking, and there are many other possibilities. But in these data, there are only two options: smoking or not smoking.

It is important to realize that data here is generated through a $Bernoulli(\bar{S})$ process: it is a discrete process with only two outcomes (smoker or non-smoker), where “smoker” occurs with probability \bar{S} . This is the best we can do with current data as *smoker/non-smoker* is the only piece of information we have for each individual. However, we may miss information about how much someone smokes and how important is smoking for them.

TBD: one-sided vs two-sided test

First let’s simulate the results. We can create a synthetic dataset by creating 3902 random workers without smoking ban and 6098 ones with smoking ban, both generated using the $Bernoulli(24.23)$ process. We stress here again that we use exactly the same probability for smoking-ban and no-ban workplaces as this is what our H_0 claims. Thereafter we compute the average smoking tendency among our synthetic workers for both types of workplaces. As both sets of workers were created through an identical process, the difference can only be attributed to the random noise. Finally, we repeat the simulations many times, and see how often we get a difference that is at least as large as we see in data, 7.76 percentage points. If such big difference is rare enough (say, it occurs less often than in 5% of cases), we can reject H_0 at the corresponding significance level (5% level in this example).

Table 1.9: Simulated smoking habits (percent of smokers) at smoking-ban/no-smoking-ban workplaces for 5 random simulations. The difference in data is 7.76 percent. We can see that simulated values in these 5 examples are much smaller than what is visible in the data.

	Ban (pct)	Non-ban (pct)	difference (pct pt)
1	24.14	23.38	0.76
2	23.50	24.37	0.87
3	24.14	24.25	0.11
4	24.01	23.70	0.32
5	24.06	23.60	0.47

Table 1.9 shows five example simulations. In all cases the the simulated difference is much smaller than the actual difference 7.76 percentage points. This supports our intuition, telling that the sample is large enough to distinguish a 7-percentage point difference. But this was only 5 simulations. Do the results still hold if we run more trials? Indeed they hold. Figure 1.19 shows a histogram of 10,000 such simulations. The maximum value obtained in these simulations was 3.3 (in absolute value), well below the observed 7.76, and the 95% confidence interval is $[-1.7, 1.75]$. For the reference, we also record that the average difference over all simulations is 0.01 and its standard deviation is 0.88. We can conclude that chances to observe such a value under H_0 are extremely low, less than 1 in 10^4 . Hence we can reject H_0 at 5% confidence level (we can also reject it at 0.01% confidence level). This suggests that H_0 is not correct. But note that, strictly speaking, we cannot claim it is *incorrect*, based on these data we can only say that it is extremely *unlikely*.

This is about as far as we can get through statistical methods. We can say that a hypothesis is “unlikely” at a given significance level, but we cannot say it is “wrong”.

TBD: Example with insignificant results, continuous data

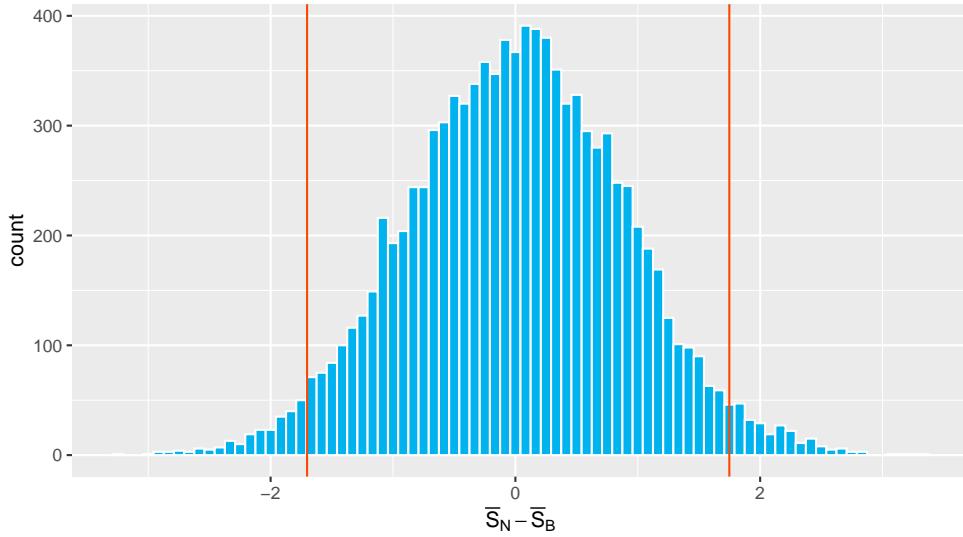


Figure 1.19: Histogram of 10^4 simulation runs. One can see that the most common values are close to 0, values above 3 are extremely rare. Orange vertical lines mark the 95% confidence intervals. The maximum observed value in these simulations is 3.3 (in absolute value), well below 7.76 in data. We conclude that encountering such a big difference under H_0 is extremely rare, and accordingly reject H_0 .

TBD: Exercise

Although the simulation approach we did above is intuitive, it is often too complex. Fortunately we can replace it with a simple formula. First, note the differences in Figure 1.19 are approximately normally distributed. This is a direct result of Central Limit Theorem, and the fact that sum (or difference) of two normal RV-s is normal:

- Both \bar{S}_B and \bar{S}_N are sums of i.i.d random values, and hence because of CLT they are both approximately normally distributed under H_0 as

$$\bar{S}_B \sim N\left(\mathbb{E} S, \frac{\text{Var } S}{N_B}\right) \quad \text{and} \quad \bar{S}_N \sim N\left(\mathbb{E} S, \frac{\text{Var } S}{N_N}\right) \quad (1.5.8)$$

Remember: $X \sim N(\mu, \sigma^2)$ means X is a normal random variable with expected value μ and variance σ^2 .

where σ^2 is variance of S . (See more [Section 1.4.3 Central Limit Theorem](#), page 60.) We do not know the expected value $\mathbb{E} S$ and variance $\text{Var } S$ but we can approximate these with sample average \bar{S} and sample variance s_S^2 .

- Moreover, as S follows the Bernoulli process, we can also use Bernoulli variance $\text{Var } S = \mathbb{E} S \cdot (1 - \mathbb{E} S)$, or rather its sample analogue $s_S^2 = \bar{S} \cdot (1 - \bar{S})$ instead of computing the sample variance of S .
- Finally, the difference $d = \bar{S}_B - \bar{S}_N$ is difference of two independent normals with equal expected value, and hence normally distributed with mean 0 and variance $\sigma_d^2 = \frac{\sigma^2}{N_B} + \frac{\sigma^2}{N_N}$.

This allows us to find the confidence interval of d under H_0 using the properties of normal distribution: the CI, related to H_0 is $[-t_{cr} \cdot \sigma_d, t_{cr} \cdot \sigma_d]$. t_{cr} is the critical value, for 95% confidence level it is 1.96. When we plug the numbers into the formula, we get variance

$$\begin{aligned}\sigma_d^2 &= \frac{\sigma^2}{N_B} + \frac{\sigma^2}{N_N} = \bar{S}(1 - \bar{S}) \left(\frac{1}{N_B} + \frac{1}{N_N} \right) = \\ &= 0.2423 \cdot (1 - 0.2423) \left(\frac{1}{6098} + \frac{1}{3902} \right) = 7.72 \times 10^{-5} \quad (1.5.9)\end{aligned}$$

or

$$\sigma_d = 0.00878 \quad \text{or} \quad 0.878 \quad \text{percentage points.} \quad (1.5.10)$$

This is almost exactly the value we received above through simulations. Now we can compute the 95% CI as

$$[-1.96 \cdot 0.878, -1.96 \cdot 0.878] = [-1.722, -1.722] \quad (\text{pct points}). \quad (1.5.11)$$

The observed difference is way outside of this range, so we can reject H_0 at significance level of 0.05.

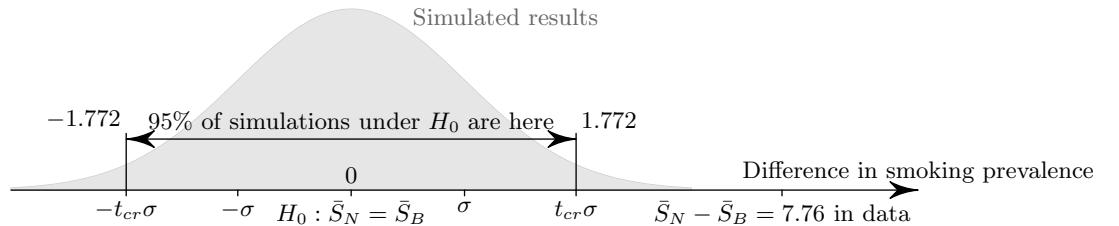


Figure 1.20: Simulated and actual data: the smoking ban example. All simulations give results near 0, but the difference in actual data is much larger.

As we can see, both approaches resulted in very similar confidence intervals and in the exactly same conclusion. This is to be expected: one approach used explicit simulations to come to the conclusion, the other approach implicitly did the same. Just instead of computing all the random numbers, we used the theoretical results from CLT, Bernoulli distribution, and sum of normals.

Finally, although we can reject H_0 : smoking behavior does not differ with the smoking ban, we cannot tell that smoking ban “causes” workers to smoke less. It is possible that the observed effect is due to reverse causality (few smokers at workplace make it feasible to introduce the ban) or confounding factors are possible (see more in [chapter 4 Causality](#), page 191). Our conclusion is pure correlational: smoking ban and smoking are “associated”.

TBD: small sample size and t -distribution

1.6 Lies, Damned Lies, and Statistics

Statistics is often colloquially accused of being unreliable, and sometime one can hear claims that “anything can be proven with statistics”. There are obviously many reasons for such unfavorable image for statistics. One broad category of problems is related to the fact that humans are just not good at understanding uncertainty. There are a number of fallacies related to probability and statistics, such as *Prosecutors fallacy* many people, including those who are highly educated in mathematics, will get trapped into.

But the fact that humans are not good in understanding uncertainty has not gone unnoticed by those who are interested in pushing their own agenda while disregarding the truth. Statistics has been widely misused by various players in order to “prove” the claims behind their dark ambitions.

1.6.1 Statistical Fallacies

Statistical language is heavy

Statistics works with uncertainty. Even more, most of the statistical results are uncertain. A typical result of statistical analysis reads like “it is more than 95% certain that the average difference between Reds and Greens is at least 1”. Such language is hard to understand and requires both statistical literacy and willingness to think for a second. Both of these are often in short supply, and the audience may prefer a simpler message “Greens are better than Reds” instead. Compare the statistical language above with other type of results, e.g. India-Iran soccer game ended with 2:1. In the former case statistics *cannot* predict precise results, while in the latter sentence tells something that is almost trivial to understand. As a result, statistical claims are often simplified into “everyday” language in a wrong way. For instance, the “layman’s version” of the claim above may be “the difference between Reds and Greens is 1”. This may be incorrect.

Probability versus plausibility

Tversky and Kahneman (.... citation) describe an experiment where people are told some information about an imaginary person, and later asked what is the person doing now, years later:

Imagine a woman named Linda, thirty-one years old, single, outspoken, and very bright. In college she majored in philosophy. While a student she was deeply concerned with discrimination and social justice and participated in antinuclear demonstrations.

What do you think, what is Linda doing now? Please assess how likely are these options from 1 (very unlikely) to 5 (very likely):

- a) Linda is active in the feminist movement
- b) Linda is a bank teller
- c) Linda is a bank teller and is active in the feminist movement.

Please evaluate the probability before you continue reading!

Kahneman and Tversky showed that people tend to consider the first option the most likely, and the second option the least likely. Linda's description just seems to fit well with someone who is feminist, and does not seem to fit too well to someone who is bank teller. But this is not the same as *how likely it is*. First, there are quite a few bank teller jobs, and there may well be many more bank tellers than active feminists. But more importantly: by construction, being bank teller *must be at least as likely* as being a bank teller and feminist. Feminist and bank teller are not perfectly related events, and hence there are bank tellers who are not feminists. If Linda is a teller and feminist, she *is also* a teller.

Kahneman and Tversky found that adding more details and explanations to a story makes it considered more plausible, even if these details make it less probable. Which story sounds more plausible: the president fires the attorney general, or the president fires the attorney general because the latter wanted to investigate the president's private businesses? The second explanation, although be less probable, sounds better and will be remembered more easily.

Our intuition fails when working with probability

Human intuition is not well suited to understand probabilities. Our brains and eyes are super good in doing image recognition and motion detection-related tasks. We are also pretty good at estimating distances, time, and average values. But in computing simple probabilities we are mediocre at best and often hopelessly wrong.

TBD: how good is the test/Bayes

Even the simple concept of sample space and possible events may take quite a bit of training and work. For instance, most people cannot understand why the solution of the two daughter problem is $1/3$. With a slight modification of the problem we can get even more crazy and counter-intuitive results. Although the problem is not hard, our intuition fails completely, and even when the solution is explained to us step-by-step, we have hard time understanding what is going on.

Example 1.16: Two daughter problem: girl has a name

Before considering this example, make sure you understand the two daughter problem as described in Exercise 1.6. This example assumes you are well familiar with the problem and understand the solution.

Consider a slightly different problem:

A family has two children. One of them is a girl, called Hina.

What is the probability that the other child is a girl too?

The modified problem sounds almost the same as the original problem, except a piece of irrelevant information, the name. However, it turns out that now we were provided different information and the answer is $1/2$, not $1/3$. How came?

Before we continue with the solution, let us introduce some notation. Denote a girl not called Hina by \bar{H} , a girl called Hina by H , and a boy by B . Also, denote the probability that a girl is called "Hina" as p_H . More specifically, assume that the probability that the *first girl* is called Hina is p_H . As parents do not call

Two daughters problem (see Exercise 1.6 page 31): A family has two kids. One of them is a girl. What is the probability that the other is a girl too?

both of their children with the same name, the second girl will be called Hina with probability p_H only if the first one was called something else. If the first girl is called Hina, the second one is never called Hina. (This rule is not central for the solution, and we might assume the name of the second child is independent of that of the first child.)

The table below displays the relevant events, and as the events are not equally likely, it also displays the corresponding probabilities. For instance, probability of (\bar{H}, \bar{H}) is a product of $\frac{1}{4}$ (the family has two girls), $(1 - p_H)$ (the first one is not called Hina) and $(1 - p_H)$ (the second one is not called Hina). But probability of (H, \bar{H}) is $\frac{1}{4}p_H$ because if the first girl is called Hina, the second one is given a different name for sure. The last column shows the corresponding event in the original two daughter problem context. One can check that the probabilities sum to unity, and the probabilities for each of the events in the original problem sum to $\frac{1}{4}$.

	event	probability	TD Event
a)	\bar{H}, \bar{H}	$\frac{1}{4}(1 - p_H)(1 - p_H)$	G, G
b)	\bar{H}, H	$\frac{1}{4}(1 - p_H)p_H$	G, G
c)	\bar{H}, B	$\frac{1}{4}(1 - p_H)$	G, B
d)	H, \bar{H}	$\frac{1}{4}p_H$	G, G
e)	H, H	0	G, G
f)	H, B	$\frac{1}{4}p_H$	G, B
g)	B, \bar{H}	$\frac{1}{4}(1 - p_H)$	B, G
h)	B, H	$\frac{1}{4}p_H$	B, G
i)	B, B	$\frac{1}{4}$	B, B

The events that correspond to the conditioning information—the family has a girl called Hina—are marked blue and below we only consider these events. The probability of interest (G, G)—the family has two girls—is made of the two events, (\bar{H}, H) and (H, \bar{H}) , out of four possible events (\bar{H}, H) , (H, \bar{H}) , (H, B) and (B, H) and hence the probability

$$p_{G,G} = \frac{(1 - p_H)p_H + p_H}{(1 - p_H)p_H + 3p_H}.$$

It is easy to see that this probability depends on p . In particular, when $p_H = 1$ then the answer is $1/3$, but in a realistic case where p_H is small, it converges to $1/2$. For instance, if $p_H = 0.01$, the answer is 0.4987, effectively $1/2$.

How on earth can we get a totally different answer by just giving the girl a name? After all, we know that all children have names, and Hina is as good as any other name...?

The crucial difference here is that there are two simple events of interest: \bar{H}, H and H, \bar{H} . Both of those correspond to G, G in the original problem. But now we distinguish between Hina and all other daughters. Even more, the probability of

Simple event: an event that cannot be decomposed into even simpler events. See Section 1.3.1, page 23.

both events is fairly similar (given p_H is small): $1/4(1 - p_H)p_H \approx 1/4p_H$. This explains the fundamental difference in this case: a family with two daughters has *twice as high* chance that one of them is called Hina! If a family has a daughter with *any name*, this is not a rare event. A family with two daughters *does not* have twice the chance that one of them has a name. If the daughter is called Hina, this is a rare event and having two daughters approximately doubles the chance that one of them has the name. Hence two-daughter families have twice the chance to remain in the sample after conditioning on the name. Our conditional sample space now looks different than in case the name was not considered.

TBD: Exercise with example number of families in each group

Contrast this problem with another one: you observe a runner in a park going behind a bush. Can you estimate where and when will she re-appear? This is an easy task to our brain. Unless the runner turns or does other unexpected moves, we are fairly good at estimating the where and when we can see her again although from the mathematical point of view, this is an incredibly more complex exercise than the two daughters problem.

Incomplete/Missing Data

We seldom enjoy working with high-quality data that describes the problem we are interested in very well. Data is often collected for another purpose, omits or under-represents certain population groups (non-homogeneous or unknown sampling), and only contains proxies for what we want to know (low or unknown validity). It is easy to come to wrong conclusions when ignoring these issues.

Sometimes the analysts forget to include the features that are *not* in data when they do the model or interpret the results. This is often a problem when the origin of the dataset is unclear, or the analysts simply do not think on the full picture. For instance, in order to analyze the efficacy of tourniquet to save life in case of severe injuries, a study looked at its usage in Iraq war.

TBD: find citation

The data, collected by hospitals, contained information about injuries, treatment (using tourniquet or not), and whether the injured soldiers survived. The authors did not find any difference related to tourniquet usage. But what was missing in the analysis? When using hospital data, they were only able to include information on soldiers who reached hospitals. Those who died before reaching the hospital were not included. Hence, even if tourniquet has no effect on those who reach hospital, we cannot conclude that this is true for the first stage, between the battlefield and the hospital.

Exercise 1.19: Damage in Allied bombers

You are a statistician, attached to the Allied air force during the WW2. Your task is analyze the damage, done by German anti-aircraft fire to the bombers that return from missions over the continent. Based on the damage pattern you will make recommendations about where to place armor on the airplanes. As armor is heavy, one cannot just armor the whole airplane, but it is feasible to

put armor on certain vulnerable places. Your analysis reveals that the planes tend to have a lot of damage in wings and in the fuselage. You don't see many damaged engines and cockpits.

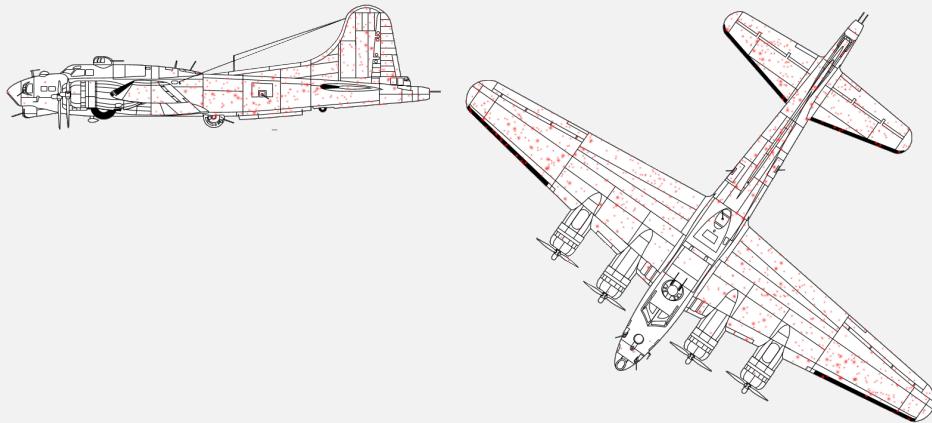


Figure 1.21: Hypothetical damage in allied bombers. Red dots denote damage in any of the thousands of bombers, marked here on a single figure. Original image: Emoscopes CC BY-SA.

What is your recommendation: which parts of the airplanes should be armored?

Solution on page [421](#).

Ecological fallacy

In applied research and analysis, it is common to use group averages as proxies for individual characteristics. However, this approach has clear limitations people (including researchers) are sometimes not aware of.

For instance, imagine we analyze the relationship between crime and wealth. Police reports crime rate by neighborhoods, and if we do not have data about the wealth of criminals, we may use average wealth across neighborhoods as a convenient proxy.

Suppose we find that there is more crime in poorer neighborhoods. It is tempting to conclude that the poor are more likely to become criminals. However, we cannot do this based on these data!

There are several problems with this conclusion. The fact that there are more criminals in poor neighborhoods does not mean that criminals are poor as well. The average data does not let us to investigate who are the criminals. Such conclusion is *ecological fallacy*, the tendency to assume that all group members share similar characteristics to the group averages. Perhaps these are the richest people in the poor neighborhoods instead? As long as the subgroup of interest (here criminals) is small and selective, we cannot assume that what is true for the average is also true for the small subgroup.

There are other problems with this conclusion too, for instance, it is not obvious that criminals commit their misdeeds in the neighborhood they live in.

Wrong Assumptions/Wrong Models

As in any other analysis, not just our data but also our models must be correct to produce correct results.

A common fallacy is to forget about confounding factors and claim causality when just observing correlation. For instance, when observing that taller children are smarter, one might conclude that height somehow causes cognitive skills. However, the reason may just be that taller kids are older (and we know that skills develop rapidly over time).

Just Errors Finally, it is also possible that statistical analysis contain just simple computation errors or other similar errors. If the methods are feasible and well-known, these are easy to correct.

Too imprecise questions Sometimes a bad answer starts with a vague question. Both our language and our thoughts tend to be imprecise, and we may not realize that a question cannot be answered, or can be answered in many different ways.

Consider a question: which country is more dangerous in terms of shark attacks on people—China or Indonesia? What would be the answer to this question? Just count of shark attacks in those countries? But populations differ, so perhaps number of attacks per person? Or perhaps the number of attacks per swimmer, as the inland population may not matter much in terms of sharks? But both of these countries contain a large number of different beaches, some which virtually never see any sharks? So do we want to compute a probability to be attacked on an “average” beach if you swim there? Is this number useful for anything?

Note that if policymakers want to consider installing shark nets or banning swimming on certain beaches, they need information about particular beaches, not country averages.



Raja Ampat islands in Indonesia have some of the most beautiful beaches on this planet, but some of those are also frequently visited by sharks. Does this make *Indonesia* dangerous?

Rolandandika, CC BY-SA 4.0, via Wikimedia Commons.

1.6.2 Misusing Statistics

Misleading Presentation

Another common issue is to present correct results in a misleading manner. This often includes mixing everyday language where words typically have more vague meaning, and statistical or logical language where the words have slightly different meaning. As an example, one may claim that certain food makes it “twice more likely” to get cancer, and the difference is “highly significant”. However, even if both claims are true, this alone does not give enough information for policymakers. We need somewhat different information: given someone eats this food, how much more likely it will be, in *absolute terms* (percentage points), to get cancer? For instance, if the cancer rate grows by 1 percentage point, from 1% to 2%, this will amount to 1% of those who eat the food. But twice as large may also mean an increase from 0.0001% to 0.0002%, and increase of 0.0001 pct points, or one in million. In the latter case the problem is much less urgent, and there are probably much easier ways to improve public health. The problem here is that the words “twice as likely” and “significant”

in everyday language suggest the presence of an important effect. But this may not be true if we interpret these words in the strict statistical sense.

This kind of misleading presentation is sometimes related to lack of statistical literacy, but sometimes it is also a deliberate strategy to advance a different agenda.

Exercise 1.20: How to multiply wealth of all Icelanders

Misleading presentation is sometimes deliberately used in politics. Imagine a politician running for an office in Iceland with a slogan *Vote for me! I'll multiply the wealth of all Icelanders!* How can she fulfill her campaign promise if elected? What exactly is misleading in this claim? For reference, the population of Iceland is 360,000 and its total wealth is \$38 billion.

Example 1.17: Are hospitals unsafe during the weekends?

[Freemantle et al. \(2016\)](#) show, based on UK data, that those who are admitted to hospitals over weekend have more serious conditions and are more likely to die within 30 days of the hospital admission. The excess death rate, associated with Saturday admissions is approximately 10%, and that for the Sunday admissions is 15%. They also show that those who are admitted during weekends have higher predicted mortality risk to begin with, and discuss the implications on hospital staffing and weekend schedules. They are very open that their study is observational and cannot tell much about causality.

However, in March 2015 the UK Conservatives made “truly seven day NHS”^a a political slogan ([Godlee, 2016](#)). In particular, the health secretary Jeremy Hunt claimed that these extra deaths are caused by poor staffing over the weekends. He also upset doctors with muddled claims about pay ([Craven, 2015](#)). As a result of the political games, a poll in 2016 found that 53% of patients believe that hospitals are unsafe at weekends ([Iacobucci, 2016](#)).

These claims are far removed from the original study. The authors never claimed that hospitals are “unsafe”, in case of a serious condition one should still get help at a hospital, even during the weekends.

^aNHS—National Health Service, the British health care system.

Chapter 2

Linear Algebra

In these notes we use vectors and matrices for two main purposes: to hold data, and to simplify algebra—linear algebra (LA) formalism tremendously simplifies algebra and computations for certain types of tasks.

This section covers the basic LA concepts we need. As our usage of LA is heavily matrix-oriented, we cover vectors only superficially. Later we typically assume that vectors are just special matrices with only a single column (or a single row). In a similar fashion we do not use inner and outer product concepts, we treat both these products as just matrix products between row- or column matrices.

Contents

2.1	Why Linear Algebra in Machine Learning	91
2.2	Vectors and Vector Spaces	92
2.2.1	Vectors	92
2.2.2	Norm and Distance	99
2.3	Matrices	103
2.3.1	What are matrices	103
2.3.2	Matrix operations	106
2.3.3	Inverse Matrix	114
2.3.4	Eigenvalues	116
2.4	Application: wireframe images	117

2.1 Why Linear Algebra in Machine Learning

The concepts “vector” and “matrix” have (at least) two related meanings. One is a type of data storage, for data that is arranged in one dimension (vector) or in two dimensions (matrix or data frame). The other meaning is vectors and matrices in mathematical, in linear algebra sense. These are numbers, stored in an 1-D or 2-D structure, exactly like the storage structures. However, linear algebra defines a large number of certain mathematical operations on these structures. Here we are interested

in the mathematical properties of these objects, but both types are closely related, for instance, many popular computer libraries that support vectors and matrices also implement the corresponding mathematical operations.

As it turns out, a large number of operations we do with ordinary numbers, such as addition, multiplication and inverse generalize easily to matrices as matrix addition, matrix multiplication, and inverse matrix. More importantly, many statistical¹ problems generalize from univariate to multivariate versions using exactly these operations (and we stress that machine learning is in many ways a branch of statistics). For instance, instead of univariate normal distribution, we can use multivariate normal to describe distribution of correlated values. This allows us to handle multivariate problems in a dimension-agnostic way, just deriving, writing, and coding formulas for general N-dimensional case.

For instance, the way to solve linear regression models in any dimensions can be written as

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T \cdot \mathbf{X})^{-1} \cdot \mathbf{X}^T \cdot \mathbf{y}. \quad (2.1.1)$$

Even if one does not know what do these symbols and operations mean, one can see that the formula is rather simple. As efficient software implementations of matrix operations are widely available, this formula can almost literally converted into computer code.

As linear algebra is ubiquitous in science and engineering, there exist dedicated well-optimized libraries, and even dedicated hardware to speed up certain linear algebra processes. For instance, ordinary graphics cards with thousand of simple computing cores, originally designed for computer games, are optimized for matrix multiplication because this is how one rotates 3-D scenery. This makes linear algebra a method of choice when implementing and using related methods.

Linear algebra is also the method of choice when presenting statistical methods. Every source, besides the very beginner-oriented texts uses linear algebra, and assumes the reader is familiar enough with the basic concepts. In this sense it is a central component of machine learning language.

Below we walk through the basics of vectors and matrices with the focus on matrix multiplication, inverse matrix, and metric distance.

2.2 Vectors and Vector Spaces

This section briefly introduces vectors, vector spaces, and a few related concepts (in particular *norm* and *distance*) in a non-matrix way. Although later we rely heavily on matrix notations, the concepts in this section do not require matrix formalism.

2.2.1 Vectors

What Are Vectors

Vectors are ordered collections of elements, for our purpose they are just sequences of numbers. Normally we denote vectors by bold lower case letters, so an example vector

¹Here we are mainly concerned with statistics, but the same is also true for many physics and engineering problems.

is $\mathbf{x} = (1, 2, 3)$. Here the vector \mathbf{x} contains three *elements* (also called *components*), 1, 2, and 3, in this order. Order matters, $(1, 2, 3) \neq (2, 1, 3)$. We denote vector components with the same letter as the vector itself, just not in bold, and supplied with the element index. For instance, given the vector \mathbf{x} above, we have $x_1 = 1$ and $x_3 = 3$. We can also use symbols to denote vector elements so another vector example is $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3)$. Here we only work with numeric vectors, i.e. with vectors where all elements are numbers. But the components do not have to be numeric, they may be all kind of objects, including letters, texts, images, functions and other vectors.

An important property of vector is its number of elements, called *dimension*.² Our example vectors \mathbf{x} and $\boldsymbol{\beta}$ are 3-dimensional. 3-D vectors are widely used to describe coordinates in our 3-space, e.g. in 3-D computer games. But our vectors can be of any (positive) dimension, including 1-dimensional (just single objects like individual numbers). They may also have very high dimensionality, for instance color images can be stored as vectors of millions of elements. In theoretical applications we can also work with infinite-dimensional vectors.

From our perspective, one of the most important roles of vectors is to hold data. For instance, consider the dataset about 50 U.S. States (R dataset *state.x77*). A few first observations of it look like:

Population	Income	Illiteracy	Life Exp	Murder	HS Grad	Frost	Area
3615	3624	2.1	69.05	15.1	41.3	20	50708
365	6315	1.5	69.31	11.3	66.7	152	566432
2212	4530	1.8	70.55	7.8	58.1	15	113417
2110	3378	1.9	70.66	10.1	39.9	65	51945

We can describe the data points (observations) of this dataset as

$$\begin{aligned}\mathbf{x}_1 &= (3615, 3624, 2.1, 69.05, 15.1, 41.3, 20, 5.0708 \times 10^4) \\ \mathbf{x}_2 &= (365, 6315, 1.5, 69.31, 11.3, 66.7, 152, 5.66432 \times 10^5) \\ \mathbf{x}_3 &= (2212, 4530, 1.8, 70.55, 7.8, 58.1, 15, 1.13417 \times 10^5).\end{aligned}\tag{2.2.1}$$

When stacking these data vectors horizontally on top of each other, we get a data matrix (design matrix). Alternatively, we can look at individual variables as vectors, in that case we have

$$\begin{aligned}\mathbf{v}_1 &= (3615, 365, 2212, 2110, 21198, \dots) \\ \mathbf{v}_2 &= (3624, 6315, 4530, 3378, 5114, \dots) \\ \mathbf{v}_3 &= (2.1, 1.5, 1.8, 1.9, 1.1, \dots) \\ &\dots\end{aligned}\tag{2.2.2}$$

²We encounter the concept *dimension* in two different meanings. Here it is the dimension of the underlying vector space, or the number of elements in the vector. But often one refers to all vectors as 1-D objects, contrary to matrices that are 2-D objects. This is because vectors are like a 1-D string of numbers and have only a single length, while matrices resemble 2-D rectangle of numbers and have both length and width. One has to understand which is meant by dimension a particular case.

Another comment about the notation: here we are using subscript index not to refer to individual components, but to refer to different vectors. If we refer to an individual component, we can add another index, e.g. $v_{11} = 3615$ and $x_{23} = 1.5$ in the example above. So the first component refers to the vector, and the last one to the component. Note that we denote components (just numbers) with ordinary font while vectors with index are in bold! However, there is a variety of notation used in the literature.

Exercise 2.1: Vector dimension

What is dimension of vectors \mathbf{x}_i in (2.2.1) and \mathbf{v}_i in (2.2.2)?

Vector Addition and Scalar Multiplication

If we use vectors just to store data, we may not really need to do any computations with these. Most of the linear algebra is based on two simple operations: addition and multiplication by scalar. With *scalar* we mean here a single number that is not a vector. We denote sum of two vectors by $\mathbf{x} + \mathbf{y}$, and multiplication by scalar α as $\alpha\mathbf{x}$.

When talking about addition and scalar multiplication, we normally mean just the ordinary mathematical operations. But these do not have to be the common addition and multiplication, these can be all kind of operations as long as they satisfy a few axioms, including

1. there is a special element, null vector $\mathbf{0}$, so that $\mathbf{x} + \mathbf{0} = \mathbf{x}$ for all \mathbf{x} .
2. multiplying any vector with scalar 0 will result in null vector: $0\mathbf{x} = \mathbf{0}$ for all \mathbf{x} .
3. multiplying any vector with scalar 1 will retain the original vector: $1\mathbf{x} = \mathbf{x}$ for all \mathbf{x} .
4. the operations follow certain distributive laws: $\alpha(\mathbf{x} + \mathbf{y}) = \alpha\mathbf{x} + \alpha\mathbf{y}$.

For the vector operations we look here, scalar multiplication is performed by multiplying all vector components by the scalar:

$$\alpha\mathbf{x} = \alpha(x_1, x_2, \dots, x_K) = (\alpha x_1, \alpha x_2, \dots, \alpha x_K). \quad (2.2.3)$$

In a similar fashion, vector addition is performed by adding the corresponding components of the vectors:

$$\begin{aligned} \mathbf{x} + \mathbf{y} &= (x_1, x_2, \dots, x_K) + (y_1, y_2, \dots, y_K) = \\ &= (x_1 + y_1, x_2 + y_2, \dots, x_K + y_K). \end{aligned} \quad (2.2.4)$$

As is obvious from this definition, the vectors must have same dimension (here K) to be possible to add those.

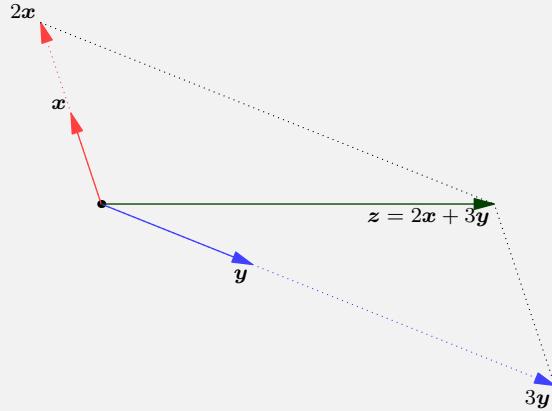
Example 2.1: Graphical way to add vectors

Consider two vectors, $\mathbf{x} = (-1, 3)$ and $\mathbf{y} = (5, -2)$. Let's compute $\mathbf{z} = 2\mathbf{x} + 3\mathbf{y}$.

When we just multiply and sum the components we get

$$\mathbf{c} = 2 \cdot (-1, 3) + 3 \cdot (5, -2) = (13, 0).$$

This operation can be represented graphically as



The solid red and blue arrows depict the original vectors \mathbf{x} and \mathbf{y} , the corresponding dotted arrows are $2\mathbf{x}$ and $3\mathbf{y}$, and long green solid arrow is their sum $\mathbf{z} = 2\mathbf{x} + 3\mathbf{y}$.

Exercise 2.2: What is the capital of France?

Word embeddings (see [Section 7.3.5 Word embeddings](#), page 329) is a way to describe words as numeric low-dimensional vectors (typically 100-300 dimensions).^a So in 100-component example the embedding for word *Berlin* (see below in the table) looks like $\mathbf{e}(\text{Berlin}) = (-0.562, 0.630, -0.453, -0.299, -0.006, \dots)$. All these numbers correspond to different components, but unfortunately the components are not interpretable in general. Embeddings are computed based on words' co-occurrence in texts. As similar words tend to occur in similar contexts, they tend to have similar embedding vectors. More interestingly, one can also do certain mathematical operations with embedding vectors. For instance, it is well known that $\mathbf{e}(\text{king}) - \mathbf{e}(\text{man}) + \mathbf{e}(\text{woman}) \approx \mathbf{e}(\text{queen})$ where $\mathbf{e}(\text{word})$ is the embedding vector of *word*.

Below is the first five components (out of 100) for *Berlin*, *Germany*, *France*, and *Paris*^b (the rest of 95 components are not shown).

word	1	2	3	4	5
Berlin	-0.562	0.630	-0.453	-0.299	-0.006
Germany	0.194	0.507	0.287	0.132	-0.281
France	0.605	-0.678	-0.436	-0.019	-0.291
Paris	-0.074	-0.855	-0.689	-0.057	-0.139

Compute $e(\text{Berlin}) - e(\text{Germany}) + e(\text{France})$ and check how close do you get to $e(\text{Paris})$.

^aWhile a 100-D vector may not sound like low-dimensional, typical vocabularies contain between 10,000 and one million different words. Using one-hot encoding would result in the corresponding number of dimensions, so a few hundred components is much less than that.

^bThe data, *glove.twitter.27B.100d.txt*, based on 2 billion tweets, can be downloaded from [Stanford NLP project](#) website.

All vectors that can be formed from certain elementary vectors using these two operations form a *vector space*. X . For our purpose it is simply a set of all relevant vectors. For a set to be valid vector space, it must be *closed* with respect to these operations. This means that whatever elements of X and real numbers we take, all their sums and products must also be in X . Formally,

- if $\mathbf{x} \in X$, $\mathbf{y} \in X$ and $\mathbf{z} = \mathbf{x} + \mathbf{y}$, then $\mathbf{z} \in X$.
- if $\mathbf{x} \in X$ and $\mathbf{z} = \alpha\mathbf{x}$, then $\mathbf{z} \in X$ for each $\alpha \in \mathbb{R}$.

An intuitive and easy-to-understand example of vector space is \mathbb{R}^2 . In \mathbb{R}^2 vectors are just pairs of real numbers, addition is defined as adding the corresponding components of vectors, and scalar multiplication is defined as multiplying all vector components with the scalar. In this case the scalar multiplication is equivalent to stretching (or squeezing) the vectors while retaining their direction, and vector addition is equivalent to parallel shift of one vector to the “end” of the other one. As a special case, negative of the vector \mathbf{x} , $-\mathbf{x}$, is just the original vector pointing in the opposite way.

Example 2.2: Application of 2-D vector space \mathbb{Z}^2

Imagine you are designing a 2-D computer game. We can choose the coordinates in different way, a natural choice is to take the origin $(0, 0)$ to be the bottom-left corner of the screen, and count the horizontal coordinates right, and vertical coordinates up. We also specify the vectors as (horizontal, vertical),^a or (x, y) . As the objects on our screen can only be at certain pixels, we are only interested in integer coordinates, so $x, y \in \mathbb{Z}$ where \mathbb{Z} is the set of all integers, or $(x, y) \in \mathbb{Z}^2$ where \mathbb{Z}^2 is the set of all pairs of integers.

Your player is located at coordinates $\mathbf{p} = (194, 33)$. She shoots an arrow upward that moves 10 vertical pixels per frame. Where is the arrow after 10 frames?

We can write the arrow’s 2-D speed vector as $\mathbf{v} = (0, 10)$. $v_1 = 0$ as the arrow does not move horizontally at all, and $v_2 = 10$ means that it moves up by 10 pixels in one time unit (here the rendering frame). In 10 frames the arrow moves

$10\mathbf{v} = (0, 100)$ and hence is located at $\mathbf{p} + 10\mathbf{v} = (194, 133)$. The location at arbitrary frame $t > 0$ can be written as $\mathbf{a}(t) = \mathbf{p} + t \cdot \mathbf{v}$. Here \mathbf{a} , \mathbf{p} , and \mathbf{v} are vectors, locations on screen, and speed on screen respectively; and t is scalar, the frame count from the moment arrow was released.

^aAlghouth *horizontal, vertical* may sound an obvious and trivial choice, it conflicts with the traditional way of displaying matrix indices, namely vertical, horizontal. Even more, vertical elements are traditionally counted from top-left corner down.

One can also easily visualize and understand the 3-D vector space \mathbb{R}^3 . Higher-dimensional spaces \mathbb{R}^K are still straightforward, but cannot be visualized.

Vector Space: Base and Linear Independence

The definition of vector space above—closedness with respect to scalar multiplication and vector addition—suggests that all vectors we can form from certain *base vectors* using only vector addition and scalar multiplication form a vector space. So if the base vectors are \mathbf{a} and \mathbf{b} , the vector space is a set of all possible vectors

$$\mathbf{c} = \alpha \cdot \mathbf{a} + \beta \cdot \mathbf{b} \quad \alpha, \beta \in \mathbb{R} \quad (2.2.5)$$

Figure 2.1 depicts on such example. Two base vectors \mathbf{a} (red) and \mathbf{b} (blue) can be used to compose \mathbf{c} and \mathbf{d} . See also Exercise 2.6 about how to compute the corresponding α and β . Such constructs, sums of vectors, multiplied by scalars, are called *linear combinations*. So vector space is made of all possible linear combinations of the base vectors.

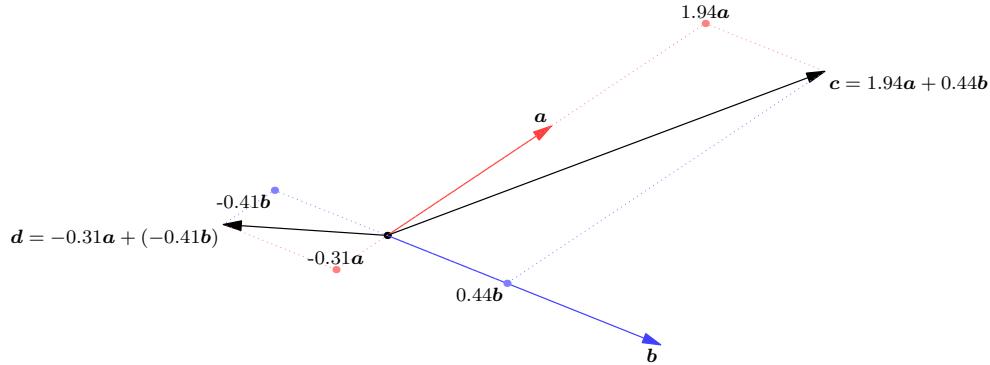


Figure 2.1: Vector space: all vectors on a plane can be computed as a linear combination of two base vectors, here \mathbf{a} (red) and \mathbf{b} (blue). The dotted red and blue arrows show which linear combinations are needed to create vectors \mathbf{c} and \mathbf{d} (black).

The figure shows only two base vectors. We can easily add another one but it turns out to be unnecessary—on the plane depicted on Figure 2.1, two base vectors are sufficient. However, removing either \mathbf{a} or \mathbf{b} will collapse the plane: there is no way to cover a plane using a single vector only. This property of a vector space—how many base vectors are needed to cover the space—is called *dimension* of vector space.

Obviously, a single base vector can only cover a line. We can multiply \mathbf{b} with any number but the result will always stay in the line defined by \mathbf{b} . So vector space made of a single base vector, 1-dimensional space, is a line. In an analogous fashion, every linear combination of vector \mathbf{a} and \mathbf{b} will stay on their plane, there is no way to describe a point outside of the plane using only these two vectors. We need a third one that points out of the plane. That would result in a 3-D vector space. These examples are easy to visualize and understand as our space is 3-D. Mathematically we can easily describe higher dimensional spaces but our imagination fails as soon as we move from three to four dimensions. But even when we cannot imagine high-dimensional space, it serves as an useful tool when working with high-dimensional vectors.

Base vectors and the dimension of vector space are closely related to linear independence. A set of vectors is *linearly independent* if one cannot compute one of these vectors from the others (by using only scalar multiplication and vector addition). Formally, we say that vectors $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n$ are linearly independent if and only if

$$\begin{aligned} \alpha_1 \mathbf{a}_1 + \alpha_2 \mathbf{a}_2 + \cdots + \alpha_K \mathbf{a}_K &= 0 \\ \Updownarrow \\ \alpha_1 = \alpha_2 = \cdots = \alpha_K &= 0 \end{aligned} \tag{2.2.6}$$

It is easy to see that this formal definition is equivalent to the informal claim above. We can easily express one vector in (2.2.6), for instance \mathbf{a}_1 , using other vectors as

$$\mathbf{a}_1 = \frac{\alpha_2}{\alpha_1} \mathbf{a}_2 + \frac{\alpha_3}{\alpha_1} \mathbf{a}_3 + \cdots + \frac{\alpha_K}{\alpha_1} \mathbf{a}_K. \tag{2.2.7}$$

But this is only possible if $\alpha_1 \neq 0$. Hence for us to be able to express at least a single vector in this way, we need at least one α to be non-zero. And by definition, this means our vectors are not linearly independent. In that case it is often said they are *linearly dependent*.

Example 2.3: Are these vectors linearly independent?

Let us test if vectors $\begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ 4 \end{pmatrix}, \begin{pmatrix} 5 \\ 6 \end{pmatrix}$ are linearly independent.

We can express one vector as a linear combination of others

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix} = 2 \cdot \begin{pmatrix} 3 \\ 4 \end{pmatrix} - \begin{pmatrix} 5 \\ 6 \end{pmatrix},$$

or alternatively write

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix} - 2 \cdot \begin{pmatrix} 3 \\ 4 \end{pmatrix} + \begin{pmatrix} 5 \\ 6 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{2.2.8}$$

Hence these vectors are not linearly independent.

Exercise 2.3: Are these vectors linearly independent?

Are vectors $\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$, $\begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix}$, $\begin{pmatrix} 7 \\ 8 \\ 9 \end{pmatrix}$ linearly independent?

2.2.2 Norm and Distance

Many machine learning methods need to compute distance between data points. For instance, nearest-neighbors method is concerned with the “closest” data points to the one we want to analyze, while clustering methods make clusters out of observations that are “close”. As we typically think about data points as vectors, all such methods need to compute distance between vectors. We first discuss a generalization of vector length,³ called *norm*. Thereafter we define distance between two vectors by just computing the norm of their difference.

Norm

Let us start with the ordinary geometry. Take the example of a 2-dimensional vector on plane. If the vector is given as $(1, 1)$, what is its length? The answer is $\sqrt{2} \approx 1.414$ (see Figure 2.2). The length of the diagonal of a square with unit length sides is 1.414. More generally, we can use the Pythagorean theorem and write the length of vector (x_1, x_2) as $\sqrt{x_1^2 + x_2^2}$. This formula can be generalized to a 3-D space \mathbb{R}^3 , the length of 3-vector $\mathbf{x} = (x_1, x_2, x_3)$ is

$$\sqrt{x_1^2 + x_2^2 + x_3^2}. \quad (2.2.9)$$

This is our most obvious understanding of length in 3-space. For instance, the box with side lengths of 1, 2 and 2 has diagonal of length $\sqrt{1^2 + 2^2 + 2^2} = 3$. We can generalize the same concept of length further into K -dimensional space \mathbb{R}^K as

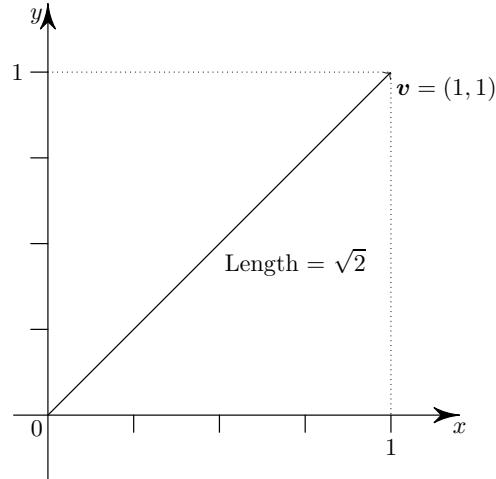


Figure 2.2: Vector \mathbf{v} has both components, v_x and v_y equal to one. From Pythagorean theorem, its length is $\sqrt{2}$. Generalized “length” of a vector is called norm and denoted by $\|\mathbf{v}\|$, in this case $\|\mathbf{v}\| = \sqrt{2}$.

$$\sqrt{\sum_{i=1}^K x_i^2} \quad (2.2.10)$$

³As *length*, here we mean length as length in space, not the number of components (we call the latter the vector’s *dimension*).

but unfortunately our imagination cannot keep up when we move above three dimensions.

This “obvious” concept of length is called *Euclidean norm*. We intuitively think in Euclidean terms as this is how the 3-D space we live in is “made”. However, there are many other ways to define length, and sometimes the conventional approach is not the best one. Those more general concepts of are called “norm”, this is why we call the Euclidean length Euclidean *norm*.

Norm of vector \mathbf{x} is typically denoted by $\|\cdot\|$, is a generalization of the concept of length: it is a function that assigns a non-negative real number to every vector. But one cannot just assign each vector an arbitrary number. Valid norm must satisfy three conditions:

1. $\|\mathbf{x}\| \geq 0$; $\|\mathbf{x}\| = 0$ if and only if $\mathbf{x} = \mathbf{0}$: norm must be positive, only null-vector has zero norm.
2. $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ (triangle inequality): the direct way is the shortest way.
3. $\|\alpha\mathbf{x}\| = |\alpha| \cdot \|\mathbf{x}\|$ (multiplication by scalar).

In machine learning applications we are often much sloppier, and use measures of “length” that are not valid metric norms. For instance, if our task is to rank texts based on the vocabulary similarity then we can easily violate the assumption 1 (see more in Section [6.1.2 Cosine Similarity](#)).

L_p norm

A rather straightforward generalization of Euclidean norm is L_p norm, also called *Minkowski norm*. It is defined by replacing “2” in the formula for Euclidean norm by a positive parameter p , and the norm is often denoted by adding a small p -subscript to the norm symbol:

$$\|\mathbf{x}\|_p = \left[\sum_{i=1}^K |x_i|^p \right]^{1/p}, \quad p > 0. \quad (2.2.11)$$

Example 2.4: L_3 norm of vector $(1,1)$

Let us compute L_3 norm of $\mathbf{v} = (1,1)$, depicted in Figure 2.2. Remember, its Euclidean, L_2 , norm is $\sqrt{2} \approx 1.414$. Its L_3 norm is

$$\|\mathbf{v}\|_3 = \left[\sum_{i=1}^K |x_i|^3 \right]^{1/3} = (|1|^3 + |1|^3)^{1/3} = \sqrt[3]{2} \approx 1.26.$$

Obviously, in terms of L_p norms, Euclidean norm is just L_2 norm. There are two other interesting and popular special cases: Manhattan norm and Chessboard norm.

Manhattan norm (also *taxicab norm*) is L_1 norm, defined as

$$\|\mathbf{x}\|_1 = \sum_{i=1}^K |x_i|. \quad (2.2.12)$$

So Manhattan norm is the sum of absolute values of the vector components, or from the geometric viewpoint it is just the sum of the vector's "sides".

Example 2.5: Manhattan norm of vector (1,1)

The Manhattan norm on the same (1,1) vector that we analyzed above is

$$\|\mathbf{v}\|_1 = \left[\sum_{i=1}^K |x_i|^1 \right]^{1/1} = (|1|^1 + |1|^1)^{1/1} = 1 + 1 = 2$$

It is easy to see why Manhattan norm is useful, and why is it called taxicab norm. Imagine you are taking cab in a city where streets are laid out in a rectangular grid, for instance in Manhattan. If your destination is 10 blocks east and 10 blocks north, then the cab driver has to drive at least 20 blocks, not matter which route she chooses. The "Manhattan-length" of your 10-block-by-10-block ride is 20 blocks. You can also imagine that the driver will not be impressed if you tell her that you are only willing to pay for a 14 blocks trip because that is the "correct" Euclidean distance.

Chessboard norm also *Chebyshev norm* is L_∞ norm. It can be computed as

$$\|\mathbf{x}\|_\infty = \lim_{p \rightarrow \infty} \left[\sum_{i=1}^K |x_i|^p \right]^{1/p} = \max_i |x_i|. \quad (2.2.13)$$

Although we cannot directly compute L_∞ distance by substituting infinity in the L_p formula (2.2.11), the fact that it amounts to maximum individual component is intuitively fairly obvious to understand. Namely, when we take numbers to p -th power as in $|x_i|^p$, the larger numbers "gain" more from this operation if p is large. At the limit where $p \rightarrow \infty$, all other components are negligible next to the largest one.

The name *chessboard norm* refers to the fact that in chess, it measures the number of moves king needs in order to move to the given number of squares in each direction (Figure 2.3).

Metric distance

Closely related to norm is *metric distance*. We can always define distance between vectors \mathbf{x} and \mathbf{y} as

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|. \quad (2.2.14)$$

So a “distance” between two vectors is the “length” of their difference. For suitable “nice” metrics we can also define the opposite

$$\|\mathbf{x}\| = d(\mathbf{x}, \mathbf{0}). \quad (2.2.15)$$

This is possible with L_p norm but not with certain other similarity measures, such as cosine similarity where one cannot define distance from null-vector, $d(\mathbf{x}, \mathbf{0})$, in a consistent manner.

In order for a distance measure $d(\cdot, \cdot)$ to be a proper metric distance, it has to have these three properties:

1. $d(\mathbf{x}, \mathbf{y}) = 0$ if and only if $\mathbf{x} = \mathbf{y}$ (identity of indiscernibles). Zero distance means the vectors are equal.
2. $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$ (symmetry). Distance is the same, whichever way you measure it.
3. $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$ (triangle inequality). There is no shorter way than the direct route.

Among the distance measure we encounter in these notes, L_p is a proper metric distance but cosine similarity is not.

We illustrate L_p distances by the corresponding unit circles (Figure 2.4). Unit circle is a set of points that are at distance 1 from the origin. In case of Euclidean metric, the unit circle is the familiar circle with radius 1, centered at the origin. In case of the other metrics, the unit circles look different. Some of these have practical applications, for instance walksheds in neighborhoods with grid-like street layout are L_1 circles.

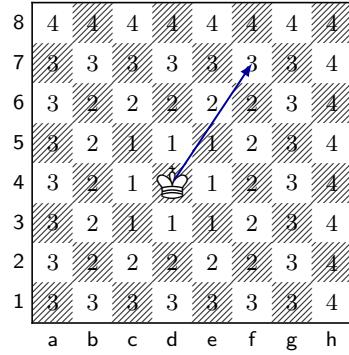


Figure 2.3: King in chess can move one field in every direction. The numbers on the chessboard denote the number of moves king at d4 needs to reach that position. For instance, it needs three moves to get to f7, and hence the vector from d4 to f7 has chessboard norm 3.

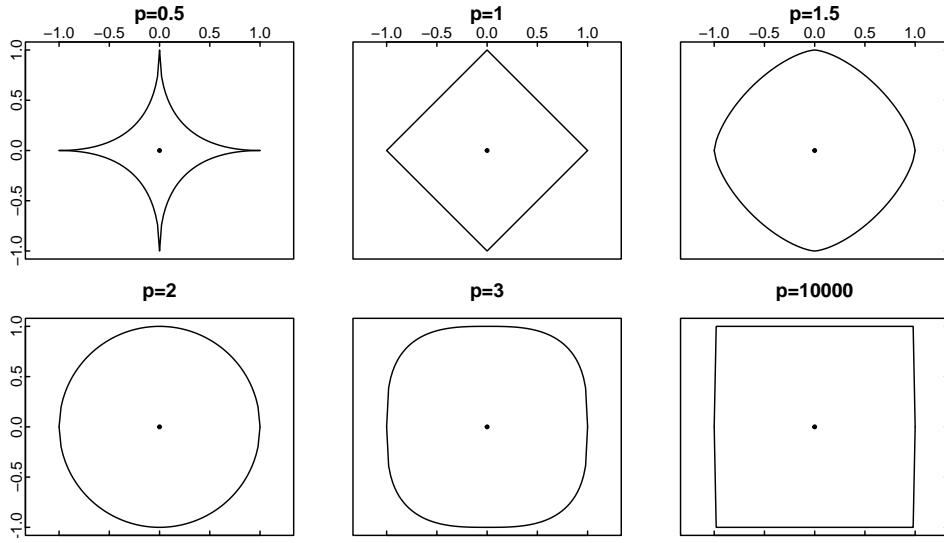


Figure 2.4: Unit circles – points sets of distance 1 from the origin (0,0) (the central dot) in different 2-D L_p spaces. If $p < 2$, the circle looks more like a star, with the Manhattan distance, $p = 1$, being diamond-shaped. If $p > 2$, the circles are more and more box-shaped.

2.3 Matrices

2.3.1 What are matrices

Matrices are some of the central objects in linear algebra. You may imagine matrices as rectangles of numbers, in many ways similar to data frames, that can be indexed based on their rows and columns. As is the case with vectors, the concept “matrix” means two different things, one is data storage on computer, and the other is a mathematical object that has certain mathematical properties.

Here are two examples of matrices:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 0 & 0 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}. \quad (2.3.1)$$

There is no universal way to denote matrices, here we follow one common tradition and use upper case letters in upright sans-serif font like X or Σ . Unlike vectors, we do not use bold symbols for matrices. Stacking numbers into rectangles is not of much interest by itself, but it turns out that these rectangles—matrices—make it possible to represent various kinds of data and related operations in a much simpler and more efficient manner.⁴ This is the main reason why linear algebra is ubiquitous in statistics and sciences.

⁴Matrices can be generalized into objects that have 3 or more dimensions, called *tensors*. These are widely used in physics, but also in advanced ML methods, such as neural networks. Modern

The first matrix A has 3 *rows* and 3 columns, the second matrix B has 3 rows but only a single column. This is *matrix dimension*. Matrix dimension is normally denoted by *rows* \times *columns* and hence matrix A is of dimension 3×3 and B is of 3×1 . But confusingly, *dimension* also means another closely related concept. Namely, sometimes we say that matrices are 2-dimensional while vectors are 1-dimensional objects. This “object dimension” is not to be confused with matrix dimension. Dimension of A is 3×3 while at the same time A is a 2-D object... Normally it is clear from the context what kind of dimension we are talking about.

The individual numbers⁵ the matrices are made of are called *matrix elements* or *matrix components*. These are often denoted by the corresponding lower case letter, supplemented with two *indices*, one for row and the other for column—by convention, matrix elements are indexed first by row and thereafter by column.⁶ For instance, the matrix A above can be written in more abstract form as

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \quad (2.3.2)$$

where the a_{11} , the element in the first row and the first column is 1, a_{12} , the element in the first row and the second column is 2, and so on. Sometimes matrix is written by the corresponding elements as $A = \{a_{ij}\}$ where $i = 1 \dots N$ and $j = 1 \dots M$. This must be understood as we take all the individual elements (numbers) a_{ij} and arrange those into the matrix.

A central role of matrices in machine learning contexts is to hold and manipulate data. For instance, the US States data on page 93 is technically a data frame, but could as well be a matrix. Holding data in matrix form makes it possible to use linear algebra methods, and as we discussed above, this is an excellent option when we are doing multivariate statistics.

A note about matrices and data frames. Both structures look similar, they are both rectangles of data. But while matrices are linear algebra objects, data frames are not. Data frames are a convenient way to store and display heterogeneous data, data where columns can be of different type, not necessarily numbers. Matrices are rectangles of only numbers (as far as these notes are concerned). While matrices in the abstract sense are not related to storage concerns, the computer implementations are. They are normally stored in a different way than data frames to facilitate operations as blocks, while data frames are often designed for easy acces by columns in mind.

Matrix Components

Certain combinations of matrix elements have their own names. As these are widely used when discussing matrices, we introduce the most important ones here.

software, such as *tensorflow* library relies heavily on tensor operations and can employ dedicated hardware, such as GPU or *tensor processing unit* (TPU) for speeding up tensor operations. We do not cover tensors in these notes.

⁵In these notes we only consider numeric matrices. But matrix elements do not have to be just numbers.

⁶Note that this tradition—rows first and columns second—contradicts with the most common 2-D image data representation: horizontal first and vertical second. This is a frequent source of confusing errors when describing graphical data in matrix form.

Matrix *diagonal* (also *main diagonal*) are the elements in the form a_{ii} . Consider matrix \mathbf{A} in (2.3.3). Its main diagonal, (a_{11}, a_{22}, a_{33}) , is left white. We usually talk about diagonal in case of square matrices only, but note that it is also defined for non-square matrices. All elements above the main diagonal, i.e. elements a_{ij} where $i < j$, are called *upper triangle* (red in (2.3.3)) and below the diagonal, i.e. elements a_{ij} where $i > j$, are *lower triangle* (blue in (2.3.3)).

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} b_{11} \\ b_{21} \\ b_{31} \end{pmatrix} \quad \mathbf{C} = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \end{pmatrix}. \quad (2.3.3)$$

Special matrices

There are a number of matrices of special form that are important enough to have a special name. It is important to know a few of those that are used most frequently in the literature.

Square matrix is a matrix with equal number of rows and columns. The matrix \mathbf{A} in 2.3.3 is a square matrix while \mathbf{B} and \mathbf{C} are not.

Symmetric matrix is a matrix where the upper and lower triangle are identical (but mirrored). For instance

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 9 \\ 3 & 9 & 16 \end{pmatrix} \quad (2.3.4)$$

is a symmetric matrix but

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 9 \\ 9 & 3 & 16 \end{pmatrix} \quad (2.3.5)$$

is not.

Formally, \mathbf{A} is a symmetric matrix if $a_{ij} = a_{ji}$ for all i, j . Obviously, only square matrices can be symmetric.

Diagonal matrix is a matrix where all elements outside of the main diagonal are zeros. Here is an example of two diagonal matrices:

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} b_{11} & 0 & 0 \\ 0 & b_{22} & 0 \end{pmatrix}. \quad (2.3.6)$$

Note that diagonal matrix may be a non-square matrices as \mathbf{B} in the example above, but one almost always refers to square matrices (as matrix \mathbf{A} above) when talking about diagonal or non-diagonal matrices.

All square diagonal matrices are symmetric.

Unit matrix (aka *identity matrix*). This is a diagonal square matrix where there are ones on the main diagonal and zeros elsewhere. It is conventionally denoted by I , or I_n for $n \times n$ identity matrix in cases where the dimension is not obvious from the context. Here are two examples:

$$I_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad I_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The importance of unit matrix is related to its properties in [matrix multiplication](#) where it is the neutral element, exactly like number one is the neutral element when multiplying numbers. Matrix-multiplying every compatible matrix A with the unit matrix I results

$$I \cdot A = A \cdot I = A \tag{2.3.7}$$

(see more in [Section 2.3.2](#) below). In particular, this means $I \cdot I = I$.

Vectors as matrices

When working with matrices it is common to treat vectors just as a special kind of matrices, 2-dimensional objects where one dimension is equal to 1. So unlike “true” vectors, vectors-as-matrices have additional properties, namely number of rows and number of columns. However, despite treating vectors as matrices, vectors are typically denoted by lower case letters in slanted bold font. We follow this habit here.

Vector-as-matrix approach gives us two types of vectors: *column vectors* are of shape $N \times 1$ and *row vectors* are of $1 \times N$. For instance, $\mathbf{a} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ and $\mathbf{b} = \begin{pmatrix} 3 \\ 4 \\ 5 \end{pmatrix}$ are

column vectors while $\mathbf{c} = (-1 \ -2)$ and $\mathbf{d} = (-1 \ 0 \ 1)$ are row vectors. Normally, if no extra explanation is given, the vectors are assumed to be column vectors. So if we talk about vector \mathbf{x} , we mean a column vector, unless we explicitly state that it is a row vector. Its transpose (see below) however, \mathbf{x}^\top , is a row vector. In order to save space, it is also customary to use row vectors and transposition operator to denote

column vector. For instance, to denote a column vector $\mathbf{z} = \begin{pmatrix} -1 \\ 1 \\ -1 \\ 1 \end{pmatrix}$ we often write

$$\mathbf{z} = (-1 \ 1 \ -1 \ 1)^\top.$$

2.3.2 Matrix operations

Matrix Transposition

A widely used operation, *matrix transposition* is “mirroring” matrix on its main diagonal. We denote transposed matrix by superscript $^\top$ in these notes.⁷ The transposes

⁷The other widely used notation for matrix transposition is apostrophe like A' .

of the matrices above in (2.3.3) are

$$\mathbf{A}^\top = \begin{pmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \end{pmatrix} \quad \mathbf{B}^\top = (b_{11} \ b_{21} \ b_{31}) \quad \mathbf{C}^\top = \begin{pmatrix} c_{11} & c_{21} \\ c_{12} & c_{22} \\ c_{13} & c_{23} \end{pmatrix}. \quad (2.3.8)$$

Note that transposition swaps the number of rows and columns.

Formally, if $\mathbf{A} = \{a_{ij}\}$ where $i = 1 \dots N$ and $j = 1 \dots M$, then $\mathbf{A}^\top = \{a_{ji}\}$.

It is easy to see that the transpose of a symmetric matrix is identical to the matrix itself.

Scalar Multiplication

Matrix multiplication with a scalar (a number) is defined exactly as in case of vectors, by multiplying every matrix element with that scalar. If $\mathbf{A} = \{a_{ij}\}$, then $\lambda\mathbf{A} = \{\lambda a_{ij}\}$. For instance,

$$3 \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 3 & 6 & 9 \\ 9 & 6 & 3 \end{pmatrix} \quad -1 \begin{pmatrix} 0 \\ 10 \\ 20 \end{pmatrix} = \begin{pmatrix} 0 \\ -10 \\ -20 \end{pmatrix}. \quad (2.3.9)$$

It is common to denote scalar multiplication either by dot like $\lambda \cdot \mathbf{A}$, or by just $\lambda\mathbf{A}$. We'll use both notations in these notes.

Matrix Addition (and Subtraction)

Matrix addition is defined as elementwise operations: if $\mathbf{A} = \{a_{ij}\}$ and $\mathbf{B} = \{b_{ij}\}$, then $\mathbf{A} + \mathbf{B} = \{a_{ij} + b_{ij}\}$. For instance,

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} + \begin{pmatrix} 3 & 2 & 1 \\ 4 & 3 & 2 \\ 5 & 4 & 3 \end{pmatrix} = \begin{pmatrix} 4 & 4 & 4 \\ 8 & 8 & 8 \\ 12 & 12 & 12 \end{pmatrix} \quad (2.3.10)$$

and

$$(1 \ 0 \ -1) - (2 \ 1 \ 0) = (-1 \ -1 \ -1). \quad (2.3.11)$$

Obviously, only matrices with similar dimensions can be added and subtracted.

Matrix Multiplication

Matrix multiplication is among the most important matrix operations, and one of the prime tools for data manipulation. Matrix product can be done manually, and although we almost always use computers in practice, it is important to have the basic understanding of it. In particular, understanding how matrix dimensions play in multiplications, and being able to compute simple products manually are invaluable skills for both coding, debugging, and devising easier and faster ways to solve data problems.

Matrix product is defined in a way that we take a row from the first matrix, column from the second matrix, multiply the corresponding elements, and sum these

products. This will be the element of the product matrix at the row (the row number) that was taken from the first matrix, and column (the column number) that was taken from the second matrix. This process must be repeated for every row in the first and for every column in the second matrix. Note that for this to be possible, the number of columns in the first matrix must equal to the number of rows in the second matrix. If this is not the case, the matrices cannot be multiplied.

This definition can be understood as a visual rule: all rows of the first matrix must be multiplied by all columns of the second matrix. Lets multiply $C = A \cdot B = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \cdot \begin{pmatrix} 4 & 3 \\ 2 & 1 \end{pmatrix}$. You can imagine a process like this, where rows of A are denoted with blue and columns of B with pink:

$$\begin{pmatrix} c_{11} & \cdot \\ \cdot & \cdot \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ \cdot & \cdot \end{pmatrix} \cdot \begin{pmatrix} 4 & \cdot \\ 2 & \cdot \end{pmatrix} = 1 \cdot 4 + 2 \cdot 2 = 8 \quad (2.3.12)$$

$$\begin{pmatrix} \cdot & c_{12} \\ \cdot & \cdot \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ \cdot & \cdot \end{pmatrix} \cdot \begin{pmatrix} \cdot & 3 \\ \cdot & 1 \end{pmatrix} = 1 \cdot 3 + 2 \cdot 1 = 5 \quad (2.3.13)$$

$$\begin{pmatrix} \cdot & \cdot \\ c_{21} & \cdot \end{pmatrix} = \begin{pmatrix} \cdot & \cdot \\ 3 & 4 \end{pmatrix} \cdot \begin{pmatrix} 4 & \cdot \\ 2 & \cdot \end{pmatrix} = 3 \cdot 4 + 4 \cdot 2 = 20 \quad (2.3.14)$$

$$\begin{pmatrix} \cdot & \cdot \\ \cdot & c_{22} \end{pmatrix} = \begin{pmatrix} \cdot & \cdot \\ 3 & 4 \end{pmatrix} \cdot \begin{pmatrix} \cdot & 3 \\ \cdot & 1 \end{pmatrix} = 3 \cdot 3 + 4 \cdot 1 = 13. \quad (2.3.15)$$

and hence $C = \begin{pmatrix} 8 & 5 \\ 20 & 13 \end{pmatrix}$. Note how c_{11} is calculated from the first row of A and the first column of B , c_{12} from the first row of A and the second column of B , and so on. In general, c_{ij} is “made” of i -th row of A and j -th column of B .

More formally, let A be $N \times K$ matrix and B be $K \times M$ matrix. We define the product as

$$C = A \cdot B \quad (2.3.16)$$

where c_{ij} , the element of C at the row i and column j , is defined as

$$c_{ij} = \sum_{k=1}^K a_{ik} b_{kj}. \quad (2.3.17)$$

C dimensions are determined by the number of rows in A and number of columns in B , hence C is $N \times M$.

Let's repeat the example from above using the formal rule. We have $C = A \cdot B = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \cdot \begin{pmatrix} 4 & 3 \\ 2 & 1 \end{pmatrix}$. Here $N = K = M = 2$, hence the product will be a 2×2 matrix. We can take the definition directly and compute c_{11} :

$$c_{11} = \sum_{k=1}^2 a_{1k} b_{k1} = 1 \cdot 4 + 2 \cdot 2 = 8. \quad (2.3.18)$$

Analogously we can do all the other elements:

$$c_{12} = \sum_{k=1}^2 a_{1k} b_{k2} = 1 \cdot 3 + 2 \cdot 1 = 5 \quad (2.3.19)$$

$$c_{21} = \sum_{k=1}^2 a_{2k} b_{k1} = 3 \cdot 4 + 4 \cdot 2 = 20 \quad (2.3.20)$$

$$c_{22} = \sum_{k=1}^2 a_{2k} b_{k2} = 3 \cdot 3 + 4 \cdot 1 = 13 \quad (2.3.21)$$

(2.3.22)

and hence, as above, $C = \begin{pmatrix} 8 & 5 \\ 20 & 13 \end{pmatrix}$. In practice, when multiplying matrices manually, it is easier to follow the visual rule we described above.

Example 2.6: Product of non-square matrices

Multiply the following non-quadratic matrices: $C = \begin{pmatrix} -1 & 1 \\ 1 & -1 \\ -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 2 \end{pmatrix}$. First, note the first matrix has 3 rows and the second has 1 column, hence the result will be a 3×1 matrix. Use the visual rule:

$$\begin{pmatrix} c_{11} \\ \cdot \\ \cdot \end{pmatrix} = \begin{pmatrix} -1 & 1 \\ \cdot & \cdot \\ \cdot & \cdot \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 2 \end{pmatrix} = -1 \cdot 1 + 1 \cdot 2 = 1$$

$$\begin{pmatrix} c_{21} \\ \cdot \\ \cdot \end{pmatrix} = \begin{pmatrix} \cdot & \cdot \\ 1 & -1 \\ \cdot & \cdot \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 2 \end{pmatrix} = 1 \cdot 1 - 1 \cdot 2 = -1 \quad (2.3.23)$$

$$\begin{pmatrix} \cdot \\ \cdot \\ c_{31} \end{pmatrix} = \begin{pmatrix} \cdot & \cdot \\ \cdot & \cdot \\ -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 2 \end{pmatrix} = -1 \cdot 1 + 1 \cdot 2 = 1$$

and hence the answer is $\begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$.

Properties of Matrix Product

Matrix product has a number of useful properties, several of which it shares with product of real numbers. Most of these can be proven by following the definition of the corresponding operations.

Multiplication by scalar

$$(\lambda A) \cdot B = A \cdot (\lambda B) = \lambda(A \cdot B). \quad (2.3.24)$$

In case of numbers, this is analogous to the fact that the product does not depend on the order of factors.

Associative property

$$A \cdot (B \cdot C) = (A \cdot B) \cdot C \quad (2.3.25)$$

given the corresponding products exist. This property is shared with ordinary numbers.

Exercise 2.4: Test the associative property

Compute the products

$$\left[\begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} \cdot \begin{pmatrix} -1 & 1 \\ 1 & -1 \\ -1 & 1 \end{pmatrix} \right] \cdot \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

and

$$\begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} \cdot \left[\begin{pmatrix} -1 & 1 \\ 1 & -1 \\ -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \right]$$

Distributive property For matrices A, B, C we have

$$A \cdot (B + C) = A \cdot B + A \cdot C. \quad (2.3.26)$$

This is also the behavior of ordinary numbers.

Non-commutative Unlike numbers, matrix product is not commutative:

$$A \cdot B \neq B \cdot A. \quad (2.3.27)$$

There are many special cases though where the matrix product is commutative. For instance, multiplication by null matrix, and by unit matrix are commutative. Also multiplication by 1×1 matrix, essentially just a number, is commutative.

As matrix product is non-commutative, we have to distinguish *left-multiplication* (*pre-multiplication*) and *right-multiplication* (*post-multiplication*). For instance, in the expression $A \cdot B$, B is pre-multiplied by A and A is post-multiplied by B . As the product is not commutative, we have to preserve the multiplication type when doing algebra.

Example 2.7: Matrix product is not commutative

Consider matrices

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 2 & 0 \\ 0 & -1 \end{pmatrix}.$$

The products are

$$A \cdot B = \begin{pmatrix} 0 & -1 \\ 2 & 0 \end{pmatrix} \quad \text{and} \quad B \cdot A = \begin{pmatrix} 0 & 2 \\ -1 & 0 \end{pmatrix}.$$

These are, obviously, not equal, here we have $A \cdot B = (B \cdot A)^\top$. This is, however, not always the case.

Transpose of product Transpose of matrix product

$$(A \cdot B)^\top = B^\top \cdot A^\top. \quad (2.3.28)$$

So a product can be transposed by a) transposing the factors; and b) switching their order.

This property has no real analogue with numbers as transposition of numbers carries little meaning.

Example 2.8: Transpose of matrix product

Lets take $A = (3 \ 2 \ 1)$ and $B = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$. Compute $(A \cdot B)^\top$. First, lets compute the product and transpose it:

$$A \cdot B = (3 \ 2 \ 1) \cdot \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = 3 \cdot 1 + 2 \cdot 2 + 1 \cdot 3 = 10. \quad (2.3.29)$$

As this is just a number, it's transpose is 10 as well. By using the transposition formula, we have

$$B^\top \cdot A^\top = (1 \ 2 \ 3) \cdot \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix} = 1 \cdot 3 + 2 \cdot 2 + 3 \cdot 1 = 10. \quad (2.3.30)$$

Note that the products $B \cdot A$ and $A^\top \cdot B^\top$ would result in a 3×3 matrix instead.

Exercise 2.5: Explain why Example 2.7 works

(2.3.28) shows that when we swap the order of factors in the matrix product, we have to first transpose both factors, and then transpose the result:

$$\mathbf{A} \cdot \mathbf{B} = (\mathbf{B}^\top \cdot \mathbf{A}^\top)^\top.$$

Explain why in Example 2.7 we do not have to transpose the factors, i.e. why

$$\mathbf{A} \cdot \mathbf{B} = (\mathbf{B} \cdot \mathbf{A})^\top.$$

Vector multiplication as matrix product

There are two ways to multiply vectors: inner (dot) product and outer product. When treating vectors as matrices, we have two types of vectors instead — column vectors and row vectors — and we can treat both types of multiplication as just the matrix product of different types of vectors. Namely, product of a row vector and column vector is equivalent to inner product, while column vector multiplied by row vector is will give the outer product.

Let $\mathbf{a} = (a_1 \ a_2 \ \dots \ a_N)^\top$ and $\mathbf{b} = (b_1 \ b_2 \ \dots \ b_N)^\top$. Now

$$\mathbf{a}^\top \cdot \mathbf{b} = \mathbf{b}^\top \cdot \mathbf{a} = a_1 b_1 + a_2 b_2 + \dots + a_N b_N \quad (2.3.31)$$

is the inner product of \mathbf{a} and \mathbf{b} while

$$\mathbf{a} \cdot \mathbf{b}^\top = (\mathbf{b} \cdot \mathbf{a}^\top)^\top = \begin{pmatrix} a_1 b_1 & a_1 b_2 & \cdots & a_1 b_N \\ a_2 b_1 & a_2 b_2 & \cdots & a_2 b_N \\ \vdots & \vdots & \ddots & \vdots \\ a_N b_1 & a_N b_2 & \cdots & a_N b_N \end{pmatrix}. \quad (2.3.32)$$

is their outer product. It is useful to keep in mind that $\mathbf{a}^\top \cdot \mathbf{b}$ (row times column) is just a single number but $\mathbf{a} \cdot \mathbf{b}^\top$ (column times row) is a $N \times N$ matrix.

Example 2.9: Inner and outer product

Let's multiply length-3 vectors of ones: $\mathbf{a} = \mathbf{b} = \mathbf{1}_3$. Their inner product is

$$\mathbf{a}^\top \cdot \mathbf{b} = (1 \ 1 \ 1) \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = 1 \cdot 1 + 1 \cdot 1 + 1 \cdot 1 = 3, \quad (2.3.33)$$

just a number. The outer product is

$$\mathbf{a} \cdot \mathbf{b}^\top = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \cdot (1 \ 1 \ 1) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad (2.3.34)$$

a 3×3 matrix.

Other Matrix Operations

Trace of Matrix For a square matrix, its *trace* is the sum of its diagonal elements:

$$\text{Tr } A = \sum_{i=1}^N a_{ii}. \quad (2.3.35)$$

Example 2.10: Matrix trace

$$\text{Tr} \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} = 1 + 5 + 9 = 15. \quad (2.3.36)$$

Determinant TBD

2.3.3 Inverse Matrix

Column space vector space, generated by the column vectors of the matrix

Column rank dimension of the column space

Row space vector space, generated by the column vectors of the matrix

Row rank dimension of the row space

Full column rank column rank equals to the number of columns

Full rank rank equals to the smallest of either number of rows or number of columns

Theorem 5. Row and column rank are equal (and are called *matrix rank*)

is a scalar function of matrix elements

- Useful descriptor of matrix properties in many contexts

- For 2×2 matrix $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$

$$\det A \equiv |A| = a_{11}a_{22} - a_{12}a_{21}$$

- for 3×3 matrix $B = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}$

$$\begin{aligned} \det B \equiv |B| = & b_{11}b_{22}b_{33} + b_{12}b_{23}b_{31} + b_{21}b_{32}b_{13} - \\ & - b_{31}b_{22}b_{13} - b_{21}b_{12}b_{33} - b_{11}b_{23}b_{32} \end{aligned}$$

Theorem 6. determinant is non-zero \Leftrightarrow matrix is full rank

Easy to see for a diagonal matrix

- A : square matrix

B is *inverse* A iff

$$BA = I,$$

and is denoted by A^{-1} .

Properties:

- $A^{-1} \cdot A = A \cdot A^{-1} = I$
- A^{-1} is also a square matrix
- A^{-1} is unique

For 2×2 matrix

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

the inverse is

$$\mathbf{A}^{-1} = \frac{1}{|\mathbf{A}|} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}$$

Definition 2. Matrix is *non-singular* \Leftrightarrow inverse exists

Theorem 7. Matrix is *non-singular* \Leftrightarrow it is full rank

Hence non-singular matrix

- has non-zero determinant
- is full-rank
- does not contain linearly dependent columns or rows (whichever is smaller)
- How does the size of ϵ affect the precision?

Exercise 2.6: Find base vector multiplier for a given vector

Equation (2.2.5) shows how all vectors in the vector space can be made of base vectors:

$$\mathbf{c} = \alpha \cdot \mathbf{a} + \beta \cdot \mathbf{b} \quad \alpha, \beta \in \mathbb{R}.$$

Given vector \mathbf{c} and the base vectors \mathbf{a} and \mathbf{b} and, compute the multipliers α and β .

Hint: attempt to transform (2.2.5) in such a way that vectors \mathbf{a} and \mathbf{b} are combined in a matrix and α and β in a vector, and use matrix multiplication instead of addition. Note also that we are in a 2-D space and hence all vectors only have two components.

Characteristic roots (eigenvalues) are solutions (λ) of the equation

$$\mathbf{A}\mathbf{c} = \lambda\mathbf{c}$$

Characteristic vectors (eigenvectors) are corresponding \mathbf{c} -s.

Intuition:

- Multiplication by \mathbf{A} does not change \mathbf{c}
- Only scales by λ .

Rewrite:

$$(\mathbf{A} - \lambda\mathbf{I})\mathbf{c} = \mathbf{0}$$

$\Rightarrow \mathbf{A} - \lambda\mathbf{I}$ must be singular $\Rightarrow |\mathbf{A} - \lambda\mathbf{I}| = 0$.

Matrix:

$$\mathbf{A} = \begin{pmatrix} 30 & 28 \\ 28 & 30 \end{pmatrix}$$

Solve for eigenvalues (using the $\det \mathbf{A} = 0$ condition $|\mathbf{A} - \lambda\mathbf{I}| = 0$):

$$|\mathbf{A}| = (30 - \lambda)(30 - \lambda) - 28^2 = 0$$

The solution:

$$\lambda_1 = 58 \quad \lambda_2 = 2$$

The corresponding eigenvectors:

$$\begin{pmatrix} 30 & 28 \\ 28 & 30 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \lambda \begin{pmatrix} x \\ y \end{pmatrix}$$

and we have

$$\mathbf{c}_1 = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \quad \mathbf{c}_2 = \begin{pmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}$$

Find eigenvalues, eigenvectors of the unit matrix

$$\mathbf{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

2.3.4 Eigenvalues

TBD: eigenvalue decomposition

Condition number

While matrix rank is a very convenient mathematical concept—matrix either is full rank or it is not—this is not as clear cut in practice. The problem arises from imprecise data and numerical errors, and all matrix manipulation methods give numerical errors, the larger the matrix, the larger the errors. This means, in practice, that we cannot rely on simple yes/no answer to the full rank question. We need another measure, a continuous measure that tells us how close we are to singularity. Condition number offers such a measure.

Condition number is defined as ratio of the largest and smallest eigenvalues (in absolute value):

$$\kappa \equiv \frac{|\lambda|_{\max}}{|\lambda|_{\min}} \tag{2.3.37}$$

where λ -s are the eigenvalues, $|\lambda|$ -s are the absolute values of eigenvalues (moduli in case of complex eigenvalues), and $|\lambda|_{\max}$ and $|\lambda|_{\min}$ are the largest and the smallest eigenvalue in terms of the absolute value.⁸ We know that singular matrices have (at least) one eigenvalue equal to zero, and hence the condition number is infinite (unless it is a zero matrix). On the other hand, if all the eigenvalues are equal, $\kappa = 1$. So κ constitutes a continuous measure of “how close to singularity” a matrix is.

⁸Greene (2003) defines condition number as square root of this expression. This is more convenient in practice as one has to work with smaller values.

Example 2.11: Condition numbers

As all eigenvalues of the unit matrix

$$\mathbf{I} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}$$

are equal to 1, its condition number is 1 as well.

The singular matrix

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

has eigenvalues 16.12, -1.117 and 0. As 0 is the smallest of those (in absolute value), the condition number $\kappa = 16.12/0$ is infinite.

2.4 Application: wireframe images

Matrices are complex structures that are often hard to understand intuitively. But there are exceptions. One of these is matrix representation of images. Here we only discuss line art images (wireframe images), photos (bitmap images) are discussed below in [Section 7.2 Images](#), page 316.

Let's make an image of \mathbb{B} , the runic letter for “b”. The line art images can be represented in vertices, the “corners” of images, that are connected to other vertices. Let's put the bottom of \mathbb{B} in the origin, and call it A . Let us also make the letter's height equal to two. The vertex coordinates may be

name	x	y	explanation
A	0	0	bottom
B	0	2	top
C	0.6	1.5	upper triangle
D	0	1	middle
E	0.6	0.5	lower triangle
A	0	0	back to bottom

The actual data is the columns x and y . We can put these in matrix \mathbf{B} :

$$\mathbf{B} = \begin{pmatrix} 0 & 0 \\ 0 & 2 \\ 0.6 & 1.5 \\ 0 & 1 \\ 0.6 & 0.5 \\ 0 & 0 \end{pmatrix}. \quad (2.4.1)$$

We can plot these vertices, and connect them by lines (Figure 2.5). This is a convenient way to represent wireframe images. For more complex images we may add additional data, e.g. color of the vertices, and a list of vertex pairs that are connected (this is called *edgelist*). This image representation format can be easily generalized to higher dimensions, e.g. for 3-D objects.

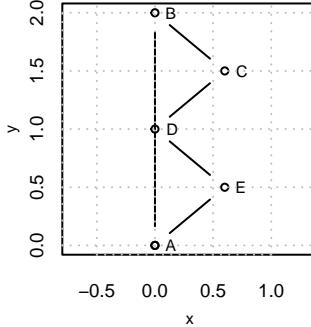


Figure 2.5: Wireframe image of the β -rune defined by matrix B in (2.4.1). All vertices are plotted and thereafter sequentially connected.

Image Rotation Prerequisites: Matrix multiplication, Basic trigonometrics

Matrix form is a convenient data representation for various image transformations that can be expressed through linear operators. These include image rotation, scaling and projection on lower-dimensional hyperplanes.

Image rotation can be done by multiplying the object vertex data by *rotation matrix*. We define 2-D rotation matrix as

$$R(\alpha) = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix}. \quad (2.4.2)$$

This matrix will rotate the image (as defined above) *clockwise* by angle α if *post-multiplied* by $R(\alpha)$.⁹

$$B^\alpha = B \cdot R(\alpha). \quad (2.4.3)$$

Note that the 2-D rotation matrix must 2×2 square matrix: two rows are needed to be compatible with 2-column data matrices, and two columns ensure that rotated data still has two columns, one for (rotated) x and one for y .

Let's rotate the vertices of β -rune, defined above, by 30° . The corresponding rotation matrix will look like

$$R(30^\circ) = \begin{pmatrix} \cos 30^\circ & -\sin 30^\circ \\ \sin 30^\circ & \cos 30^\circ \end{pmatrix} = \begin{pmatrix} 0.866 & -0.5 \\ 0.5 & 0.866 \end{pmatrix} \quad (2.4.4)$$

⁹We defined the image by stacking the x and y coordinates of vertices in *columns*. Alternatively, x and y can be stacked in rows. This is equivalent to transposing the image data matrix (B in (2.4.1)) as defined here. Accordingly, the corresponding formula will look like transpose of (2.4.3): $B^T \alpha = R(\alpha)^T \cdot B^T$.

and the rotated vertices are

$$\mathbf{B}^{30} = \mathbf{B} \cdot \mathbf{R}(30^\circ) = \begin{pmatrix} 0 & 0 \\ 0 & 2 \\ 0.6 & 1.5 \\ 0 & 1 \\ 0.6 & 0.5 \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0.866 & -0.5 \\ 0.5 & 0.866 \end{pmatrix} = \begin{pmatrix} 0.00 & 0.00 \\ 1.00 & 1.73 \\ 1.27 & 1.00 \\ 0.50 & 0.87 \\ 0.77 & 0.13 \\ 0.00 & 0.00 \end{pmatrix}. \quad (2.4.5)$$

The rotated matrix is given in Figure 2.6. Note that the vertex metadata is not affected by rotation. Here these are just vertex labels and the connection rule (we just connect all vertices to the next vertex), but these may also include vertex colors and more complex edgelists.

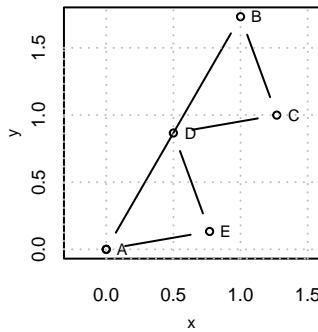


Figure 2.6: The same object as in Figure 2.5 but rotated 30 degrees by multiplication with the corresponding rotation matrix. The rotated vertices are given as \mathbf{B}^{30} in (2.4.5).

Exercise 2.7: Inverse of rotation matrix

Intuitively, the inverse of a rotation matrix $\mathbf{R}(\alpha)$, $\mathbf{R}(\alpha)^{-1}$, must be rotation in the opposite direction by a similar amount, i.e. $\mathbf{R}(-\alpha)$. Show that for 2-D rotation matrices this is indeed the case, i.e. $\mathbf{R}(\alpha) \cdot \mathbf{R}(-\alpha) = \mathbf{I}$.

It is easy to generalize the rotation matrix into higher dimensions. In the 3-D space, we can rotate the object around each of the three axes, x , y , and z , and hence we have three rotation matrices. For *right-handed* coordinate system these are

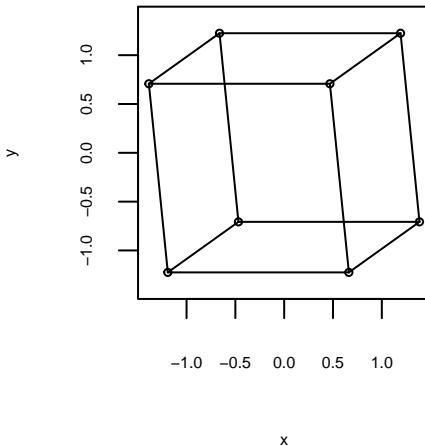
$$\mathbf{R}^{xy}(\alpha) = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \mathbf{R}^{zx}(\alpha) = \begin{pmatrix} \cos \alpha & 0 & \sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{pmatrix}$$

and $\mathbf{R}^{yz}(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{pmatrix}. \quad (2.4.6)$

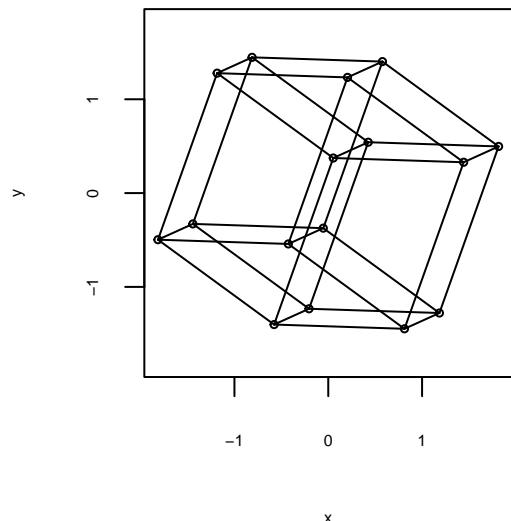
Arbitrary rotations in space can be achieved by sequential application of these matrices. Although our imagination stops here, it is easy to continue into even higher dimensions. However, to display three or higher dimensional objects, we have to project these on the 2-D plane.

Projection Projection (more specifically parallel projection) can be defined as removing some of the coordinates, e.g. dropping z and keeping only x and y to make a 2-D projection of a 3-D object. If we want to project the object onto another plane besides the $x - y$ plane, we may start by rotating it first. As dropping a coordinate is equivalent of deleting the corresponding column, the projection matrix is similar to rotation matrix with the dropped column removed.

Example: cube, rotated and projected on 2-D:



This is easy to understand. But we can make two-dimensional projections not just 3-D spatial objects—we can go into higher dimensions. Here is a similar image of 4-D cube, *tesseract*. This is impossible to understand as our brain has virtually no experience with 4-D world.



Chapter 3

Regression Models

This chapter discusses regression models, in particular linear and logistic regression. These are perhaps the most important models for both inferential and predictive modeling, both on their own but also as components of more complex models.

Contents

3.1	Linear Regression	123
3.1.1	The Problem: Why We Want Linear Regression	123
3.1.2	Simple Regression	125
3.1.3	Interpreting regression results	134
3.1.4	Formal Definition of Linear Regression	141
3.1.5	Model evaluation: MSE, RMSE, R^2	143
3.1.6	Multiple Regression	149
3.1.7	Categorical Variables	154
3.1.8	Feature Transformation	160
3.1.9	Interactions Effects	165
3.1.10	Theoretical considerations	171
3.2	Logistic Regression	179
3.2.1	What Is Logistic Regression And What Is It Good For?	179
3.2.2	Interpreting logistic regression results	185
3.2.3	Solving logistic regression model	190
3.3	Linear probability model	190

3.1 Linear Regression

3.1.1 The Problem: Why We Want Linear Regression

In both research and applied analysis we are often interested in relationships—are larger values of x associated with larger or smaller values of y ? Or maybe we already know that the relationship exists but we may want to quantify it—by how much are y larger for those cases where x values are larger by one unit?

As a example, let's analyze the length and width of iris sepals (from the well-known [iris data](#)).¹ Figure 3.1 (right panel) displays the length and width of the sepals for the iris species *setosa*. We see an upward trending point cloud where each point denotes a setosa flower. The fact that it trends upward is no surprise—we expect that the longer leaves are also wider. However, the exact form of the relationship may not be obvious—how much wider are falls that are 10mm longer? And does the same relationship hold for different fall lengths? Do very long leaves get wider at an increasing pace and become more rounded? Or perhaps the way around—very long leaves are more elongated? And sometimes we do not have any particular reason to expect an increasing or decreasing relationship. But we can always plot data like here and check what kind of relationship do we see.

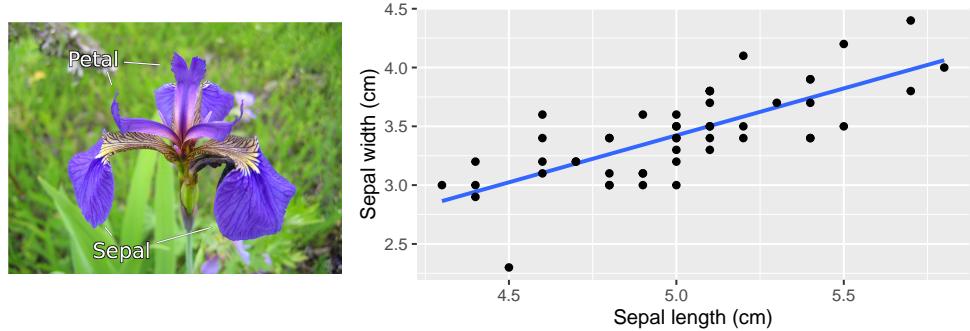


Figure 3.1: Flower of iris setosa (left). Sepals are the big purple falls spreading downward, petals are thinner and grow upward. The right panel displays the relationship between the width and length of sepals. Not surprisingly, longer petals tend to be wider. The blue trend line is computed using linear regression.

Original image by Denis Anasimov, [wikimedia commons](#).

More formally, we sometimes want to test if two variables are related. And if they are, then how strong is the relationship? For instance, are variables x and y more closely related than x and z ? Another time we may know the value of one variable and want to use this knowledge to predict the value of another one. For instance, what is “typical” width of setosa sepals that are 5cm long? What is the “typical” price of a house that is 200 m² large? But there are many ways how two variables can be related. Figure 3.2 shows a few different options. Which of these curves is “correct”? Which of these is “better”?

Obviously, there is no general answer to the “correct” and to the “better” question. It depends on the process, data and the problem. But we want to construct a tool that can be used to answer the following questions:

- are these two variables related? Yes or no?

¹Iris flower dataset was introduced by Ronald Fisher in 1936. It contains measurements for 150 flowers of species *setosa*, *versicolor* and *virginica*. It is one of the most widely used statistical dataset. ([Wikipedia entry](#))

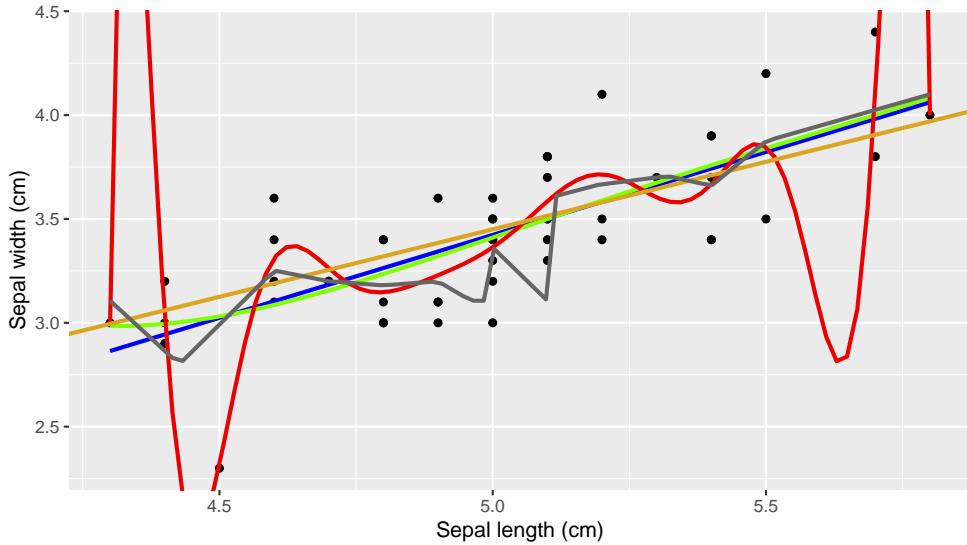


Figure 3.2: The same data as in Figure 3.1, right panel. Besides the original trend line (dark blue), this figure indicates a number of other possible relationships. Which one of these is better? Just by looking at the figure, we can say that the red line looks too wobbly while we may not like the kinks of the gray line. But the blue, green, and yellow line look fairly similar. In typical problems we prefer the simplest option with certain favorable properties, here the original regression line (blue).

- how strong is the relationship? For many real world problems, such as prediction, it is not just enough to say that there is a relationship, we need a numeric value.
- is the number statistically significant? Maybe it is just a random blip that the numbers look related?
- how does the relationship in one dataset compare to that in the other dataset? Is it stronger or weaker?
- and finally, we want the tool to be intuitive and easy to use.

We stress here that in order to answer the questions above we need a mathematical tool. Just eyeballing the data and deciding which curve is the “best” is not precise enough and does not scale (but it is a very important starting point!). The tool should have clear mathematical formulation and clear mathematical assumptions so we can judge in each case if it is appropriate to use it. It should also be flexible, allowing various tweaks to be incorporated to address problems of different flavor. And finally, it should be simple to implement and use on computer.

3.1.2 Simple Regression

TBD: History

Introduction

Linear regression is perhaps the most popular tool to answer these questions. It checks all the boxes in the list above offering both yes/no-style answers and quantitative answers. It is also simple, intuitive, and easy to use.

Linear regression is a statistical model. Here *model* means a specification how the “outcome variable” y is related to the “explanatory variable” x . A model is a necessary tool if we want to “ask data” about the true relationship. In typical applications we consider here, we need a *statistical model*, a model that contains both a deterministic and a stochastic part. The deterministic part is what we are typically interested in, the part of the relationship we can reliably describe and use for inference and prediction. The stochastic part is rarely used beyond evaluating model’s performance, but it is a necessary component that takes care of the stochastic nature of data. In many-many common applications we simply cannot reliably predict the outcome based on the information we have. The stochastic component of the model helps to handle such unreliability in a consistent and precise manner.

In regression models we describe the value of the *outcome variable* y using *explanatory variable* x . In Figure 3.1 above we treated that data in a way that sepal length is the explanatory variable and sepal width is the outcome variable. Sometimes, but not here, we can interpret it in a causal sense, i.e. the regression model tells us what happens to the outcome if we manipulate the explanatory variable in a certain manner (e.g. make a leaf 1cm longer). There are many other way to call these variables, e.g. *endogenous variable*, *dependent variable*, *target*, or just “y” for the outcome, and *exogenous variables*, *independent variables*, *features*, *predictors*, or just “x” for x . See Cheatsheet 3.1 on page 132.

Hence linear regression treats data in a fundamentally asymmetric way—data is partitioned into explanatory variables and the endogenous variable. Sometimes this is a natural approach, for instance if our task is to predict salary (y) based on education (x), or if we are interested how drug dosis (x) affects the patients’ health (y). In other cases, it may be less relevant. For instance, it is not obvious why we should treat width and length of leaves in an asymmetric manner.²

Despite of being an old (over 200 years) method, it is still immensely popular, and it is hard to see it being replaced any time soon. Linear regression is definitely not everything a data scientist has to know, there are just too many problems (for instance, natural language processing or image analysis) that cannot be tackled with linear regression. But linear regression wins almost always in terms of simplicity and interpretability. It is also a handy benchmark and a building block for many more complex statistical models, such as neural networks.

Setup

Linear regression model is traditionally written as

$$y_i = \beta_0 + \beta_1 \cdot x_i + \epsilon_i. \quad (3.1.1)$$

²Such asymmetric treatment of data is common to a large class of models, commonly called *supervised learning* methods in machine learning literature. Certain other methods (*unsupervised learning*), , such as clustering or principal component analysis, do treat all data in a similar manner.

Here y is the outcome variable, x is the explanatory variable, and ϵ is the *error term* (also called *disturbance term*, or *noise term*), see Cheatsheet 3.1 on page 132 for summary of the terms. Index i indicates individual observations, typically rows in the data frame. Note that x , y and ϵ have index i but β_0 and β_1 do not—it indicates that each observation i has a different value for y , x and ϵ , but they all share the same *parameters* (aka *coefficients* or *betas*) β_0 and β_1 . Coefficients are a property of the model, not a property of individual observations.

Let us explain the role of all the symbols using a small example. Consider the dataset below:

i	x	y
1	0	1
2	1	1
3	2	2

It describes a data frame with three rows, $i = 1, 2, 3$ and two variables, x and y . The linear regression equation (3.1.1) for these data can be written as

$$y_i = \frac{5}{6} + x_i \cdot \frac{1}{2} + \epsilon_i, \quad (3.1.2)$$

i.e. the parameters $\beta_0 = \frac{5}{6}$ and $\beta_1 = \frac{1}{2}$. (see Section 3.1.4 below how β_0 and β_1 are defined.) This is simply a shorthand notation for three equations

$$\begin{aligned} y_1 &= \frac{5}{6} + x_1 \cdot \frac{1}{2} + \epsilon_1 \\ y_2 &= \frac{5}{6} + x_2 \cdot \frac{1}{2} + \epsilon_2 \\ y_3 &= \frac{5}{6} + x_3 \cdot \frac{1}{2} + \epsilon_3. \end{aligned} \quad (3.1.3)$$

Exercise 3.1: Compute ϵ

Use the 3-line dataset and the parameter values $\beta_0 = 5/6$ and $\beta_1 = 1/2$, given above, to compute ϵ_1 , ϵ_2 and ϵ_3 .

Solution on page 423.

So the error terms ϵ are just terms that take care of the difference between the computed and the actual values. In practice it is almost never possible for a model to capture the actual y values precisely, and hence we need some tools to account for the discrepancy. Disturbance terms are these tools.

Let us demonstrate linear regression using Iris data, the same dataset that was used in Figure 3.1. The first few lines of the data are shown in Table 3.1. Let's pick sepal width as the outcome y and sepal length as the explanatory variable x , as in Figure 3.1. This is what we know. We know all x and y values and we know our model, but we don't know β_0 , β_1 and ϵ .

Now we can put (3.1.1) in a more specific form as

$$\text{Sepal width}_i = \beta_0 + \beta_1 \cdot \text{Sepal length}_i + \epsilon_i. \quad (3.1.4)$$

Table 3.1: Example cases from Iris dataset

Sepal length	Sepal width
5.1	3.5
4.9	3.0
4.7	3.2
4.6	3.1
5.0	3.6

When estimating the model we find the best values of β_0 and β_1 where *best* means the best in the linear regression sense. The solution here is $\hat{\beta}_0 = -0.569$ and $\hat{\beta}_1 = 0.799$ (see more in [3.1.4 Formal definition of linear regression](#) on page [141](#)). The “hat” on top of $\hat{\beta}$ stresses that these are not “true” values but our best estimates based on data. So we can rewrite the definition (3.1.1) for the first few observations in the data as

$$\begin{aligned} 3.5 &= -0.569 + 0.799 \cdot 5.1 + e_1 \\ 3 &= -0.569 + 0.799 \cdot 4.9 + e_2 \\ 3.2 &= -0.569 + 0.799 \cdot 4.7 + e_3 \\ &\dots \end{aligned} \tag{3.1.5}$$

Note that the parameter values (β_0 and β_1) are the same for all observations i , but the variable values differ for each i . We have replaced ϵ by e to stress that we don’t know the correct ϵ , but we can compute e from (3.1.5).

The two essential parts of the model is the deterministic part $\beta_0 + \beta_1 x$, and the stochastic part ϵ . In most applications we are primarily interested in the deterministic part. By itself it describes y as a linear function of x because as we have $y = \beta_0 + \beta_1 \cdot x$, the modeled x - y relationship will form a straight line on graph. Parameter β_0 is called *intercept* (also *constant*), parameter β_1 is commonly called *slope*, but often one refers to both parameters together as “betas” or “coefficients”.

In the situation where we typically use linear regression (and other statistical models) we usually do not know the “correct” values of β_0 and β_1 but we know our data, i.e. all the explanatory and outcome variable pairs (y_i, x_i) for $i = 1 \dots N$. Hence our first task is to find β_0 and β_1 (see more in [3.1.4 Formal definition of linear regression](#) on page [141](#)) before we can use the model for anything else. Sometimes we are interested in the parameter values itself as these may carry policy-relevant meaning (see more in [Interpretation](#)). In other cases we do not care much about the betas, but want to use those to predict other interesting outcomes, such as y values (see more in [Prediction](#)).

Example 3.1: How fast does the universe expand?

By early 20th century, it was clear that certain nebulae in sky are outside of our Milky Way galaxy and astronomers attempted to use those to determine the Solar motion in space. By late 1920s, there was already data for both velocity and distance for 24 “extragalactic nebulae”, i.e. galaxies. In 1929, Edwin Hubble

published a paper where he plotted velocity versus distance for those 24 objects (Hubble, 1929).

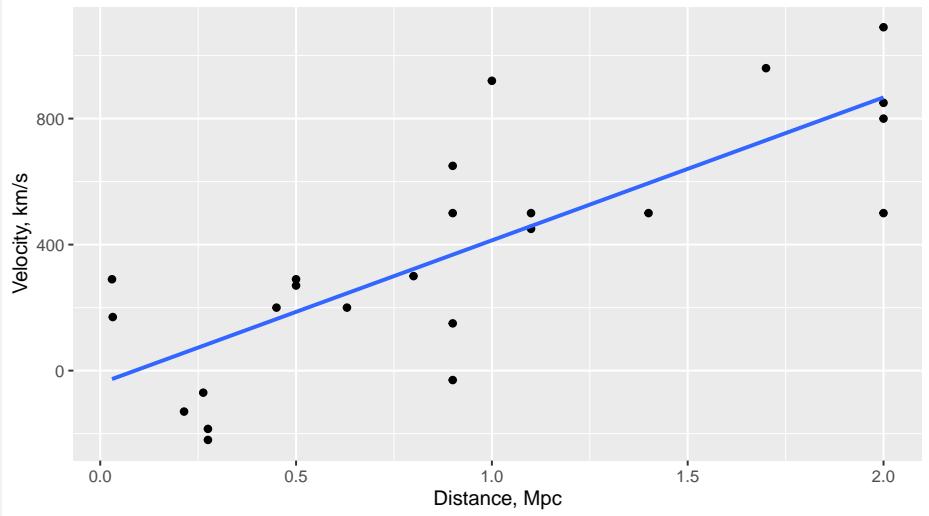


Figure 3.3: The original Hubble diagram. Hubble estimated the slope to be approximately $500 \text{ km s}^{-1} \text{ Mpc}^{-1}$. Despite the less-than-impressive data, and the fact that he badly overestimated the slope, it is considered one of the most important cosmological discoveries of all time.

On the figure, the more distant galaxies are clearly moving faster away from us. This suggests that the universe is expanding. The expansion rate can be estimated from the same data using linear regression

$$\text{velocity}_i = \beta_0 + \beta_1 \cdot \text{distance}_i + \epsilon_i. \quad (3.1.6)$$

The estimation results are $\beta_0 = -40.4$ and $\beta_1 = 453.9$. The estimated value of β_1 indicates that 1 Mpc^a more distant galaxies move 453.9 km/sec faster away from us (see Section 3.1.3 Interpreting Regression Results below). The expansion rate β_1 is nowadays called ‘‘Hubble constant’’ and its modern estimates are approximately $72 \text{ km s}^{-1} \text{ Mpc}^{-1}$. We can reverse this rate and ask ‘‘how long time it takes for a galaxy, moving 454 km/sec, to reach to 1 Mpc = 3.09×10^{19} km distance? This gives us roughly 2 billion years, the number of years since the Big Bang (modern estimates are 13.8 billion years). This was an important piece of evidence supporting the idea that the universe is young.

^aparsec (pc = 3.09×10^{13} km or 3.26 light years) is a distance where the Earth orbit’s radius is visible as 1'' arc. The closest stars are 1.3pc away from us, Milky Way disk is 50,000pc in diameter. Mpc = 1 000 000 parsec.

Prediction

Prerequisites: Section 1.3.5 Expectation, expectation as a linear operator

(3.1.1): $y_i = \beta_0 + x_i \cdot \beta_1 + \epsilon_i$

Imagine we have somehow figured out the “right” values of β_0 and β_1 . Now we can immediately use the model for predicting the results. This amounts to answering the questions like “what will be the outcome value y that corresponds to the explanatory variable value x ? Let’s look at the model definition (3.1.1) again. We somehow know the values of β_0 and β_1 (we just figured these out). We also know $x_i, i = 1 \dots N$ (this is our data). But we don’t know ϵ_i , as that is an unobserved stochastic error. True, we can compute it as in Exercise 3.1, but this is only possible in case we know y . But as we try to *predict* y , then we probably don’t know it to begin with... So we cannot compute the predicted y .

Instead, what we can do is to compute its expected value $\mathbb{E}y$ instead. Let’s take expected value of (3.1.1):

$$\mathbb{E}y = \mathbb{E}[\beta_0 + \beta_1 \cdot x + \epsilon] = \mathbb{E}[\beta_0 + \beta_1 \cdot x] + \mathbb{E}\epsilon. \quad (3.1.7)$$

As β_0 , β_1 and x are known values, their expectations are just these values. We just have to find $\mathbb{E}\epsilon$. Normally we just use an assumption

$$\mathbb{E}\epsilon = 0. \quad (3.1.8)$$

(see Section 3.1.10 Assumptions in OLS Models.) It means that its expectation is exactly zero, and hence its mean in a finite sample (like our data) tends also to be close to zero. This may sound like a strong assumption but it is actually pretty harmless in most cases. If we assume something else, say $\mathbb{E}\epsilon = a$ for some constant a , this would amount of shifting y values up by a . But we already have the intercept term, β_0 that plays a similar role and shifts y values up and down. As a result, the β_0 would decrease by amount a , so that $\beta_0 + a$ will remain constant. Our predictions would not change.³

Note that typically we predict $\mathbb{E}y$, the expected value of y . We may instead predict something else, e.g. median or other quantiles of y , its minimum value, or probability that y is positive. Such predictions may need different assumptions instead of (3.1.8).

With these assumptions in place, we can just write our predictions as

$$\hat{y}(x) = \mathbb{E}y = \beta_0 + \beta_1 \cdot x. \quad (3.1.9)$$

(One often uses “hat” like in \hat{y} to denote estimated or predicted values for y .) It may be written in different forms, for instance

$$\hat{y}(x_i) \quad \text{or} \quad \hat{y}_i \quad \text{or} \quad \hat{y}(x_i; \beta_0, \beta_1) \quad (3.1.10)$$

where the first form makes the dependency on x explicit, the second form uses index i for a short-hand notation, and the third version also indicates that the prediction

³Here we assume that the model in fact includes the constant term. If this is not the case, the assumption may have major implications. See

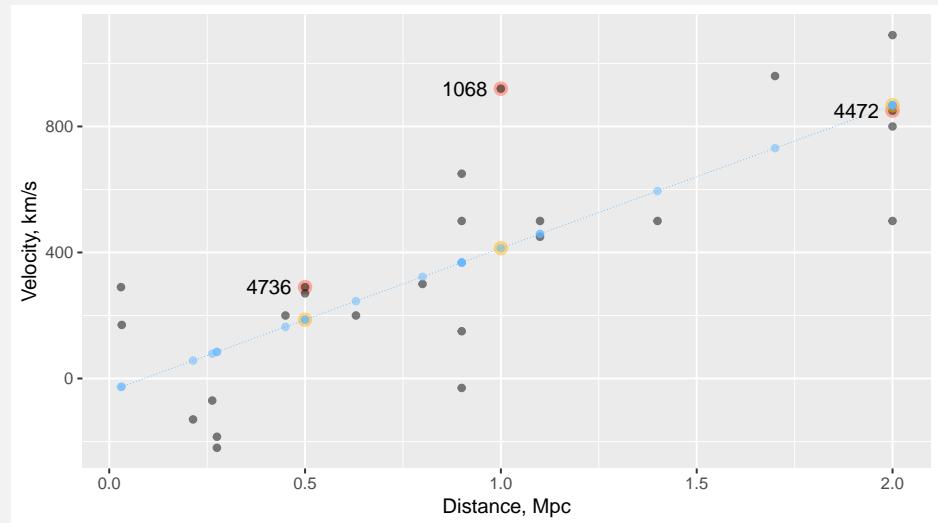
TBD: reference to the example

depends on the model parameters β_0 and β_1 . Note that \hat{y} is a *linear function* in x —this is why linear regression is called *linear* regression. Hence \hat{y} will be a line on the x - y plane.

Example 3.2: Predicted Velocity of Galaxies

Let us use the results from Example 3.1 to predict the speed of galaxies in the Hubble data. We found that the model estimates are $\beta_0 = -40$ and $\beta_1 = 454$. We focus on three galaxies: NGC 4736 ($R = 0.5$), NGC 1068 ($R = 1.0$) and NGC 4472 ($R = 2.0$ Mpc). Using the estimated β -s, we can find the predicted speed from (3.1.9) as 186.5, 413.4 and 876.3 km s $^{-1}$.

The figure below shows the measured velocity in data (black) and the predictions (light blue).



NGC 4736, a galaxy in Canes Venatici. The modern estimate of its distance is 4.9 Mpc, almost 10 times more than at Hubble's time. R Jay Gabany (Blackbird Obs.), CC BY-SA 3.0, via Wikimedia Commons.

Figure 3.4: Hubble data. The black dots represent the actual distance-velocity combinations of galaxies as known to Hubble in 1929. Light blue dots are the predicted velocities, corresponding to the model in Example 3.1. NGC 4736, 1068 and 4472 are marked with orange/yellow halo.

All the predicted values are on a straight line because (3.1.9) represents a linear function. We can see that the prediction error (residual) for NGC 1068 is rather large, but in case of NGC 4472 we have predicted almost exactly the correct value.

When we know the true value y (for instance, on training data), we can also compute the prediction errors, typically called *residual terms*, *deviations*, or *residual errors*:⁴

$$e = y - \hat{y} = y - (\beta_0 + \beta_1 \cdot x). \quad (3.1.11)$$

Exercise 3.2: Predict using linear regression

Demo dataset:

i	x	y
1	0	1
2	1	1
3	2	2

Consider the demo dataset on page 127. The parameter estimates are $\beta_0 = \frac{5}{6}$ and $\beta_1 = \frac{1}{2}$. Compute the predicted values \hat{y}_1 , \hat{y}_2 and \hat{y}_3 .

Solution on page 423.

Obviously, the better the model, the smaller are the residual terms. But in general we face trade-offs—we cannot make residual error for one observation smaller without making it larger for another observation at the same time. But errors do not necessarily mean the model is imperfect. The errors in Hubble estimate originate from three sources: incorrect speed measurements; incorrect distance measurements; and the fact that galaxies are not just fixed to the expanding space but are also moving relative to their co-moving space. None of these problems makes Hubble's model bad. It still captures the expansion factor, and it had been fairly close to the modern estimates if astronomers in 1920-s had had access to the modern methods for distance estimation. After all, Hubble's greatest achievement was not to accurately predict the velocity of “extragalactic nebulae” but to realize that the universe is expanding. The error terms played only a minor role in that discovery.

Cheatsheet 3.1: Simple Regression: Definition

Model

$$y_i = \beta_0 + \beta_1 \cdot x_i + \epsilon_i$$

- y : *outcome* (also *target*, *endogenous variable*, *left-hand variable*, y)
- β_0, β_1 : *parameters* (also *coefficients*, *betas*)
- β_0 : *intercept* (also *constant*)
- β_1 : *slope* (also *effect*)
- x : *explanatory variable* (also *feature*, *exogeneous variable*, *right-hand variable*, *feature*, *predictor*, *attribute*, x). This is your data.
- ϵ : *error term* (also *disturbance term*)
- i counts *observations* (also *cases*)

The deterministic part of the model $y_i = \beta_0 + \beta_1 \cdot x_i$ is *linear* in x , i.e. depicts a straight line on x - y plane. The error terms takes care of the fact that the data points may be off that line.

Prediction When we know β_0 , β_1 and x , we predict y as

$$\hat{y}_i = \beta_0 + \beta_1 \cdot x_i.$$

⁴There is an important conceptual difference between residuals e and disturbance terms ϵ (but fortunately it does not matter much in practice). Namely, if we know the *correct values* of β_0 , β_1 and y , then we can actually recover the *disturbance* ϵ . However, when β -s are estimated from data (as they almost always are), similar exercise will give us the *residual* e instead.

The statistical problem is to find a good combination of β_0 and β_1 so that the prediction line fits the existing data well.

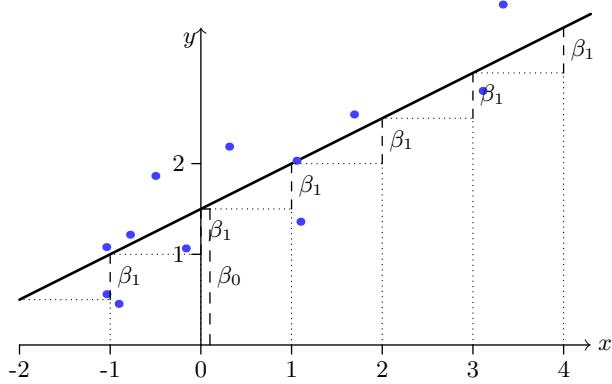


Figure 3.5: Interpretation of regression coefficients. The blue dots represent the data points and the thick line is the regression line. Intercept β_0 represents the vertical intercept of the thick regression line, i.e. the predicted y value at the point where $x = 0$. Slope β_1 corresponds to the “climb” of the line when x increases by one unit. This figure is made using random data.

3.1.3 Interpreting regression results

Interpretation

One of the big advantages of linear regression is its interpretability. There are other interpretable models, such as logistic regression, but none can compete with linear regression in terms of ease and simplicity. In many situations we are less interested in the predicted values and more interested in understanding the underlying process, and in such cases linear regression is often the obvious choice.

To interpret a model means to “understand” the parameter values and being able to tell a story what do these values mean. Simple regression has two parameters, intercept β_0 and slope β_1 . The meaning of these parameters can easily be understood when analyzing the predicted values. From (3.1.9) we see that if $x = 0$, the predicted $\hat{y}(0) = \beta_0$. Hence intercept indicates the expected (predicted) value of y if $x = 0$ (See Figure 3.5). To understand what does slope, β_1 , describe, we can compute the difference

$$\hat{y}(x+1) - \hat{y}(x) = [\beta_0 + \beta_1 \cdot (x+1)] - [\beta_0 + \beta_1 \cdot x] = \beta_1. \quad (3.1.12)$$

Hence slope tells us how much larger is the prediction \hat{y} when x larger by 1 unit.

(3.1.9):

$$\hat{y}(x) = \beta_0 + \beta_1 \cdot x.$$

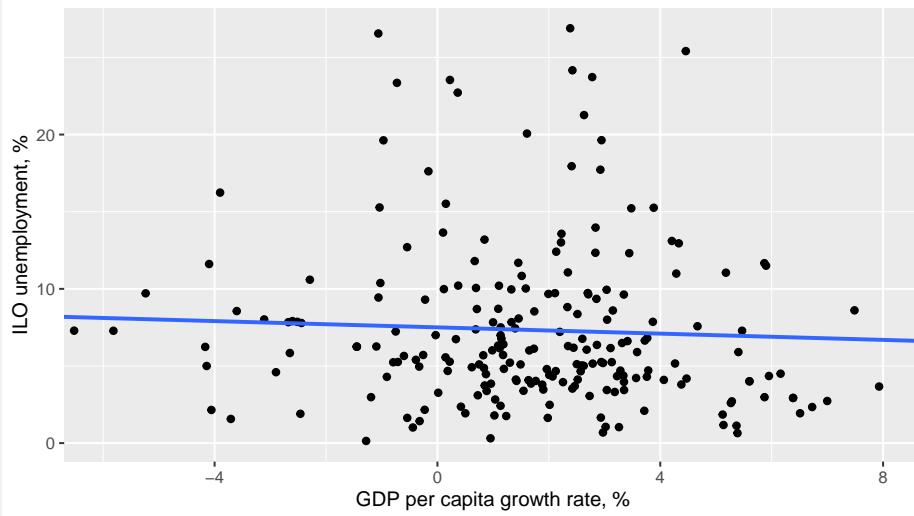
Example 3.3: Unemployment versus GDP growth

Figure 3.6: Relationship between unemployment and GDP growth across countries in 2016, and the corresponding regression line. World Bank data.

Figure 3.6 shows the relationship across countries between unemployment and GDP growth where unemployment is measured as percentage of labor force, and GDP growth is measured in percentages. We can model the relationship as

$$\text{unemployment}_i = \beta_0 + \beta_1 \cdot \text{GDP growth}_i + \epsilon_i \quad (3.1.13)$$

where i denotes different countries. The corresponding linear regression estimates are $\hat{\beta}_0 = 7.5$ and $\hat{\beta}_1 = -0.1$. Here “intercept” means that expected unemployment for a zero-growth country is 7.5 percent. For each additional percent of growth, that number falls by 0.1 pct points. For instance, if economic growth is 2%, the model predicts the unemployment rate to be 7.4%. The estimate for *growth* seems surprisingly small (and is not statistically significant), but remember the data describes a cross-section of countries in 2016, a period of rather robust growth, and not relationship over time for an individual economy.

Note that linear regression (nor any other statistical model) does not allow to make causal claims. The *growth* estimate -0.1 cannot be interpreted that more growth *causes* less unemployment, at least not based on this data.

Exercise 3.3: Income and education

You estimate the relationship between income and education in the form

$$\text{income}_i = \beta_0 + \beta_1 \cdot \text{education}_i + \epsilon_i$$

where *education* is measured in years and income in dollars. You find $\beta_0 = 1000$,

$\beta_2 = 5000$. What does β_0 tell you? Is it an interesting number? What does β_1 tell?

Solution on page 424.

Note that while these interpretations are always correct from the mathematical perspective, they may sometimes carry little real world meaning. For instance, the regression line in Figure 3.1 is given by parameters $\beta_0 = -0.569$ and $\beta_1 = 0.799$. The intercept means that zero-length sepals are -0.569 cm wide. This does not make any sense, but as none of our flowers have sepal length less than 4cm, it does no harm when we use our model for the actual 4-6cm long flowers. But extrapolation for small flowers may be very misleading as this example suggests. The slope parameter β_1 means that for each unit (i.e. centimeter) sepals are longer, they are 0.799 units (i.e. centimeters) wider in average. This number is reasonable and tells us something about the shape of the flowers.

Exercise 3.4: How is sons' height related to fathers' height?

The father-son dataset (see Example 1.14) contains 1078 fathers' and sons' height. An example of the data looks like

Father	Son
178.4	180.4
167.3	175.4
166.7	179.1
164.4	162.5

where "Father" and "Son" are the corresponding heights in centimeters. When we estimate the regression model

$$\text{Son}_i = \beta_0 + \beta_1 \cdot \text{Father}_i + \epsilon_i$$

we get the following results: $\beta_0 = 86.1$ and $\beta_1 = 0.51$.

Interpret these results. Are any of these interpretations misleading?

Solution on page 424.

Correlation and causation

One has to keep in mind that regression coefficients cannot be interpreted causally. The regression parameter that connects sepal width and length cannot be interpreted as for every centimeter the sepal grows in length, it grows 0.799 centimeters in width. Linear regression⁵ only computes the average relationship: in our data, longer leaves are also wider. Data alone do not tell why. These traps are sometimes easy to avoid. For instance, the results of Exercise 3.4 would read that "if fathers grow by 1cm then sons grow by 0.5cm". This is obviously nonsense—even if you were able to make someone's father taller, this will in not affect the height of their children...

⁵This issue is not specific to linear regression only but it is a common problem with all statistical models. In order to establish causality based on statistical analysis, we need very specific information that is typically not present in what we call "data". See more in Chapter 4.

But other times a causal interpretation sounds perfectly natural. Interpretations like “if we increase schooling by one year then income will grow by 6 percent” or “if 1 pct point more people wear masks the infection rate will fall by 1.2 pct” sound perfectly plausible claims. The problem is that the common datasets do not tell if such an interpretation is correct or misleading. Humans easily slip into semi-causal interpretation, and the fact that the correct language sounds clumsy and non-natural does not help here. Humans are also prone to interpret relationships causally even when explicitly stated that this may not be true. It is better to re-phrase the two previous examples as “those who attended school for one more year earn 6 pct more income” and “regions where 1 pct point more people wear masks see 1.2 pct lower infection rate”.

The causality-agnostic language also has a special phrase, *associated with*, to denote the correlational relationship like what is computed in linear regression. So our sepal results may be phrased as *1 cm longer leafs are associated with 0.799 centimeters more width* and we can say that “one year more of schooling is associated with 6 pct higher income”.

Interpreting the regression table

The statistical software we use for linear regression typically outputs not just coefficient values but a complete table of results. Table 3.2 shows an example of such a table, computed for the *setosa* sepal length–sepal width regression

$$\text{Sepal Width}_i = \beta_0 + \text{Sepal Length}_i + \epsilon_i \quad (3.1.14)$$

(See Figure 3.1).

Typical software output uses the variable names to label the estimates instead of β_0 and β_1 . Here β_0 is called “intercept” and β_1 is “sepal length” as this is the variable that β_1 is multiplied by. The column “Estimate” presents the same estimated coefficients we discussed above. Here we discuss the other columns in this table. As it turns out, all these columns are very important.

Table 3.2: Software output table from sepal length–sepal width regression. Different software package may provide slightly different output, but the main information is very much the same.

	Estimate	Std. Error	t-value	Pr(> t)
Intercept	-0.569	0.522	-1.091	0.281
Sepal length	0.799	0.104	7.681	0.000

Next column in the table is labelled “Std. Error”. This is the standard error of the estimate. As the points do not line up exactly, we need to include certain randomness in the model (this is term ϵ in (3.1.1)). Intuitively, depending which data points we sample, our regression coefficients will be slightly larger or slightly smaller.

Under mild assumptions, these coefficients follow t -distribution (see Section 1.4.2 [t-distribution](#)) and this column provides their standard error. You can imagine that we collect different *setosa* flowers many-many times. Each time we get a slightly different sample, and hence we get slightly different estimated values. Standard error describes the variability of the estimates obtained in this way. But in practice we do not want to do many samples (usually we even cannot do it), so “Std. Error” is computed using the mathematical properties and underlying assumptions instead.

In this table we can see that the intercept’s standard error is 0.522 while the sepal length coefficient’s error is 0.104. Hence the latter is much more precisely determined by our data than the former.

The next column is labelled “ t -value”. This is the t -value for the coefficient:

$$t = \frac{\text{Estimate}}{\text{Std. Error}}. \quad (3.1.15)$$

It is a number computed just from the two previous columns. This is related to the most common hypothesis test that is done in context of linear regression: H_0 : Estimate = 0. You can imagine the data where x and y (i.e. sepal length and sepal width) are not related. But just because randomness in data, we always see some kind of relationship. One can show that if H_0 is correct and certain conditions hold, then t value as defined here is t -distributed (that’s why it is called t -value) and large t values are unlikely under H_0 . In the table the intercept’s t -value is -1.091 and that for sepal length is 7.681. Hence it is much more likely to see such intercept value than such “sepal length” value just by random chance. We can compare t values here with the critical t values from the t -value table (Table 1.8 is an example of such a table). For instance, for two-tailed test at 5% significance level at 50 degrees of freedom⁶ the critical value is $t_{cr} = 2.01$. The Table 1.8 does not have an entry for $df = 48$, but $df = 50$ is close enough.

Finally, the last column “ $\text{Pr}(>|t|)$ ” is p -value, how likely it is to get such a t -value if H_0 is correct. It is essentially the significance number we get if we use a t -table to look up the t -value of the previous column. The probabilities are 0.281 and 0, so just by playing with random data, we can get an intercept of similar size in more than 25% of cases. However, the chances of getting similar value for sepal length coefficient are much smaller and essentially 0.

Example 3.4: Interpreting Regression Table

Consider the Hubble dataset of 24 observations. When we estimate the model

$$\text{velocity}_i = \beta_0 + \beta_1 \cdot \text{Distance}_i + \epsilon_i$$

we get the results:

	Estimate	Std. Error	t value	$\text{Pr}(> t)$
Intercept	-40.436	83.448	-0.485	0.633
Distance	453.860	75.246	6.032	0.000

(see Example 3.1).

⁶Degrees of freedom for linear regression model is number of observations minus the number of parameters to be estimated, here $50 - 2 = 48$.

We can compute t -values by dividing estimates and standard deviations as

$$\text{Intercept: } -40.436/83.448 = -0.485$$

$$\text{Distance: } 453.86/75.246 = 6.032.$$

These numbers are exactly the same as in the table above, so usually there is little need to compute t -values.

We can find the p -values using Table 1.8. First, we need to find the degrees of freedom. It is the number of observations minus the number of estimated parameters, $df = 24 - 2 = 22$. As the table does not have an entry for $df = 22$, we pick the closest value, $df = 20$ (2nd line). The t -value of the intercept, 0.485,^a is smaller than any value in that row. In particular, it is smaller than 1.33, the critical t -value that corresponds to the significance level of 20%. Hence we can conclude that even if the true intercept is 0, there is more than 20% probability to see that big value (-40.436) just by chance. This is considered too large, and hence we cannot reject $H_0 : \beta_0 = 0$. Intercept is not statistically significant.

However, the t -value of *Distance* is 6.032. This, in turn, is larger than any number in that row. In particular, it is larger than 3.85, the critical value that corresponds to the significance level 0.1%. We can conclude that if the true parameter is 0, there are very small probability to see such large β value as 453.86 just by chance. We cannot say how large is the probability exactly based on the table alone, but we can say it is less than 0.1%. In most circumstance such a level is considered more than enough to reject $H_0 : \beta_1 = 0$ and hence β_1 is statistically significant.

Nowadays, statistical software typically also provides p -values, so tables as 1.8 are less important. The software output may also be accompanied with additional information, such as significance markers or confidence intervals.

^aRemember: the sign of t -value does not play a role when computing the p -value. It just shows the sign of the corresponding coefficient.

Exercise 3.5: Interpreting regression table

1. You estimate your model and find $\beta = 4$ while its standard error is 1.6. Compute t value.
2. You have 105 datapoints and 5 variables in your dataset. Find the p -value from Table 1.8.
3. Is your estimate statistically significant at 5% level?
4. Is it significant at 1% level?
5. What does it mean in regression context: β is significant at 5% level?

Solution on page 424.

TBD: Example with intercept 0

Cheatsheet 3.2: Simple Regression: Interpretation

Interpreting coefficients

- Intercept β_0 : the y value at $x = 0$ (in average)
- Slope β_1 : the cases where x is larger by one unit have y larger by β_1 units (in average).

It is not correct to say that *if we increase x by one unit, y will increase by β_1 units*. This claim implies causality but normally we cannot establish causality.

Interpreting regression table

Consider Hubble regression

$$\text{velocity}_i = \beta_0 + \text{distance}_i \cdot \beta_1 + \epsilon_i.$$

Software regression output looks something like this:

	Estimate	Std. Error	t-value	Pr(> t)
Intercept	-40.436	83.448	-0.485	0.633
Distance	453.860	75.246	6.032	0.000

first column (no name here): parameter names. Intercept is β_0 and the name of the x -variable, slope β_1 of which is presented.

Estimate estimated value of the parameter

Std. Error estimated standard error of the parameter

t-value $t = \frac{\text{Estimate}}{\text{Std. Error}}$, t-value for testing $H_0 : \text{Estimate} = 0$.

Pr(>|t|) p -value of the t -test, the probability that we observe estimate of such size if H_0 is correct.

Interpretation:

- Intercept: galaxies at distance 0 Mpc move at speed -40 km/sec.
- Slope: Galaxies that are 1 Mpc further away move 454 km/sec faster.

3.1.4 Formal Definition of Linear Regression

Now we have done all the preparatory work to define the linear regression model.

Let us revisit the definition of residual term (3.1.11). We noted above (see Section 3.1.2 Prediction) that better models tend to have lower prediction errors, but we cannot drive all of them down to zero at the same time. Instead, we somehow have to address the trade-offs we face. In case of linear regression, we define the “best” model as the one that minimizes the sum of squared residuals. This is why linear regression is often called “least squares”—the word “least” refers to minimization and “squares” refers to the fact that we minimize squared errors.⁷

We can minimize the sum of squared errors (SSE) by selecting good values for β_0 and β_1 . This gives us an informal definition of linear regression: it is the linear model (3.1.1) where parameters β_0 and β_1 are chosen in a way that the models residual sum of squares, SSE, is minimized.

Formally, we can write the sum of squared errors as

$$\begin{aligned} SSE(\beta_0, \beta_1) &= \sum_{i=1}^N e_i^2 = \\ &= \sum_{i=1}^N (y_i - \hat{y}_i)^2 = \\ &= \sum_{i=1}^n [y_i - (\beta_0 + \beta_1 \cdot x_i)]^2. \end{aligned} \tag{3.1.16}$$

Here we write SSE as $SSE(\beta_0, \beta_1)$ to stress that its value depends on β -s. The first line of (3.1.16) is the definition of SSE. Note that we minimize $\sum_i e_i^2$, not $\sum_i \epsilon_i^2$. This means that we do not know the true values of e , but for whatever β_0 and β_1 we pick, we can always compute e . The others two expressions follow from the definition of residuals and from the definition of predicted value. The “correct”, i.e. optimal β -s are those that minimize $SSE(\beta_0, \beta_1)$, formally written as

$$(\hat{\beta}_0, \hat{\beta}_1) = \arg \min_{(\beta_0, \beta_1)} \sum_{i=1}^N e_i^2 = \arg \min_{(\beta_0, \beta_1)} \sum_{i=1}^n [y_i - (\beta_0 + \beta_1 \cdot x_i)]^2. \tag{3.1.17}$$

Here we denote the optimal values for β_0 and β_1 by $\hat{\beta}_0$ and $\hat{\beta}_1$. This can be understood as we play around with β_0 and β_1 until we have achieved the smallest possible SSE, and then we call the corresponding values $\hat{\beta}_0$ and $\hat{\beta}_1$. Note that these values are our *estimates*, not necessarily the “true” values of β_0 and β_1 (and we follow habit by denote estimated value of β by $\hat{\beta}$). The true values are unknown, the estimates are the best we can do based on data. In case of simple regression, one can get fairly far with computing SSE manually by just trying different β values. Alternatively,

⁷The term “linear regression” and “linear least squares” are usually treated as synonyms. However, we do not necessarily have to minimize sum of squared errors. We may choose to minimize other functions of the residual terms, for instance sum of absolute values of the errors. The result is still linear, and still has the property of regression to mean (Galton, 1886), but it is usually called “median regression”, not linear regression.

one can rely on non-linear optimization (see [Section 8.2 Gradient Ascent](#)). Linear regression turns out to be even simpler, as here we can solve the best β -s analytically (see [Section 3.1.10 Solving Linear Regression Models](#)). This is the only statistical model where it is possible and no doubt, this has also contributed to its popularity.

Example 3.5: SSE for the iris sepals regression

Let's compute a few SSE values for *setosa* sepals, for the same data we used in [Figure 3.1](#). Pick first $\beta_0 = 0$ and $\beta_1 = 1$, i.e. we assert that in average sepals are as wide as they are long. [Table 3.3](#) shows the relevant calculations. The first 4 rows show the first four lines of data. The two first columns are data, sepal length and sepal width. The third column, \hat{y} is the predicted width, and given our choice of β -s, it is exactly equal to sepal length. The fourth column, $e = \hat{y} - \text{Sepal width}$ is the residual. As we are predicting way too large width, the residuals are all positive. The final column, e^2 , contains the squared values of the corresponding residuals. The last row is the sum of the corresponding columns. Here we are only interested in the last number, the sum of squared errors.

Table 3.3: Computing SSE for setosa data. Sepal length and Sepal width are the actual datapoints. \hat{y} is the predicted width, given $\beta_0 = 0$ and $\beta_1 = 1$. e is the corresponding deviance and e^2 is squared deviance, “squared error”. The last line gives the sum of all rows.

	Sepal length	Sepal width	\hat{y}	e	e^2
1	5.10	3.50	5.10	1.60	2.56
2	4.90	3.00	4.90	1.90	3.61
3	4.70	3.20	4.70	1.50	2.25
4	4.60	3.10	4.60	1.50	2.25
...
sum	250.30	171.40	250.30	78.90	127.91

In this example we have $SSE = 127.91$, much more than 3.159 we get when picking optimal values for β_0 and β_1 . See [Example 3.6](#).

3.1.5 Model evaluation: MSE, RMSE, R^2

One of the first tasks after estimating the model is to understand how good a job does it do. Is this model actually better than just predicting the average value for everyone? Just how much better is the model? A natural answer to this comes from the least squares model definition: how big is it's SSE?

But SSE alone is not a good answer. There are several problems when using just SSE as a model goodness indicator:

- SSE grows as we add more datapoints to the model. So a large value of SSE may either mean that the model does not describe the data well, or that we just have a lot of data.
- SSE is measured in squared units, so for instance if y is measured in dollars, MSE will be in dollars-squared. This is hard to interpret.
- It is also hard to compare models on different kind of data. If units of measurement are different, then SSE will also be different even on the same dataset. And sometimes the units are inherently different. For instance, income and temperature cannot be measured in the same units, and hence we cannot tell which one is modeled better—at least not based on SSE alone.

Fortunately, all three issues have fairly simple solutions.

In order to fix the first issue—SSE grows with dataset size—we can use not sum of squared errors but *mean squared error* (MSE) instead. MSE is just average of SSE over datapoints:

$$MSE = \frac{1}{N} \sum_{i=1}^N e_i^2 = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2. \quad (3.1.18)$$

You may notice that the formula for MSE resembles that of sample variance (1.2.2), just instead of the average value \bar{y} , we use the predicted value \hat{y}_i to compute the deviations.

(1.2.2): $s^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x}_i)^2$.

The solution to the second problem is easy too: instead of MSE, we can use its square root, called root-mean-squared-error (RMSE):

$$RMSE = \sqrt{MSE}. \quad (3.1.19)$$

RMSE is measured in the same units as y and hence easily interpretable. If MSE resembles variance, its square root resembles standard deviation and we can say something like “typically, our predictions are off by RMSE”. The wording—“typically...”—is deliberately vague. As you can see from (3.1.18) and (3.1.19), RMSE is a sort of average prediction error. However, we do not call it “average” because people may then think we are talking about the arithmetic average. But it is not the arithmetic average. Obviously, both MSE and RMSE can also be used to define linear regression in analogous fashion as SSE in (3.1.17). A set of betas that minimizes SSE, will also minimise MSE and RMSE.

The solution to the last issue is a little bit more involved but it leads us to the well known R^2 , perhaps the most popular measure of goodness in regression models.

We start with the observation that SSE, the sum of errors “left over” by the model, does not tell much about the model’s performance unless we know how spread-out are the observations (y -s) to begin with. So we define *total sum of squares (TSS)*

$$TSS = \sum_{i=1}^N (y_i - \bar{y})^2 \quad (3.1.20)$$

where \bar{y} is the average of y . Note the difference between TSS and SSE as defined in (3.1.16): while SSE computes the error terms as the difference between the true y_i and the model prediction \hat{y}_i , TSS computes it as the difference between y_i and the average, \bar{y} . So TSS is a convenient measure of the total spread in the data. It is very much equivalent to variance (1.2.2), just multiplied by N .

This gives us a measure of model goodness: if the “leftover variance” SSE/TSS is small, the model “explains away” most of the variation in the data. For instance, if $SSE/TSS = 0.2$, the model only “leaves behind” 20% of the original variation. Traditionally, one looks at the reverted version of this measure: R^2 is defined as

$$R^2 = 1 - \frac{SSE}{TSS}. \quad (3.1.21)$$

In this hypothetical example $R^2 = 0.8$, and the model explains 80% of the variation in data. Let’s think a second what this means. In one extreme case where our model is completely useless, and our predictions are no better than just predicting the mean value for every data point, we have $SSE = TSS$ and hence $R^2 = 0$. In another extreme case where our model is able to predict every single observation exactly, we do not have any prediction errors and hence $SSE = 0$ and hence $R^2 = 1$. So R^2 gives us a convenient and easy-to-interpret measure of prediction goodness: which percentage of the total variation is explained by the model. Small R^2 indicate the model is not good (from the predictive perspective) and high R^2 shows that it predicts well.

Unfortunately, the squared deviations SSE and TSS that R^2 is based on is not easy to visualize, but one can construct a similar measure based on range to make it more intuitive (Figure 3.7). In this example, the “total range” in data (call it TR) is 4 and the “error range” ER is 3. We can define a range-based R^2 as $R_r^2 = 1 - 3/4 = 0.25$.

Example 3.6: R^2 for setosa sepals regression

We follow up Example 3.5 and compute R^2 of the corresponding regression. However, instead of picking arbitrary parameter values (0 and 1 in Example 3.5), we compute the regression estimates. These are $\beta_0 = -0.569$ and $\beta_1 = 0.799$ (see page 3.1.2 Section 3.1.2). First we present a similar table to compute deviations and SSE as in Example 3.5:

Table 3.4: Computing R^2 for setosa data. The table is analogous to the table in Example 3.5, just this time using the actual regression coefficient values instead of 0 and 1.

A similar measure as R^2 , but based on range. The four data points \mathbf{x}_1 - \mathbf{x}_4 (colored) have total range, the vertical difference between the topmost and the lowermost data point, $TR = 4$. However, the corresponding residuals, e_1 - e_4 have range (“error range”) of $ER = 3$ only. Hence the model decreases the range in data from 4 to 3 and $R_r^2 = 1 - 3/4 = 0.25$. But note that this measure is not the “true” R^2 because it is defined based on range, not based on variance. This is why it is denoted by R_r^2 , not R^2 .

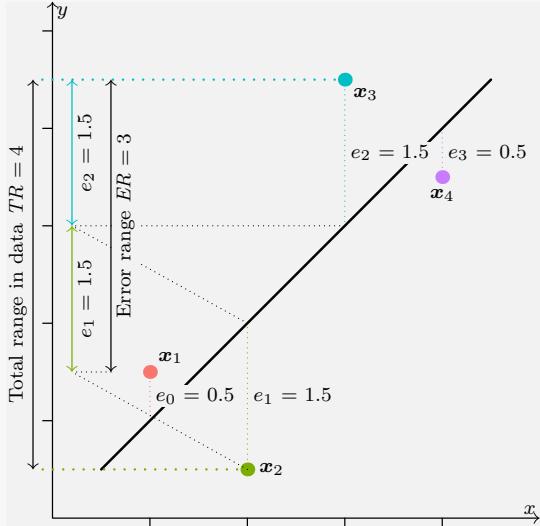


Figure 3.7: Range-based construction of R^2

	Sepal length	Sepal width	\hat{y}	e	e^2
1	5.10	3.50	3.50	0.00	0.00
2	4.90	3.00	3.34	0.34	0.12
3	4.70	3.20	3.18	-0.02	0.00
4	4.60	3.10	3.10	0.00	0.00
...
sum	250.30	171.40	171.40	0.00	3.16

As we have picked the regression estimates for β_0 and β_1 now, the deviations e are small, and we see both positive and negative values now. The table shows that $SSE = 3.159$, a much smaller value than 127.91 we got in the previous example.

In order to compute R^2 , we also need TSS (3.1.20):

$$TSE = \sum_i (\text{Sepal width}_i - \overline{\text{Sepal width}})^2 \quad (3.1.22)$$

where $\overline{\text{Sepal width}}$ is the average value of sepal width. Plugging in the data, we find $TSS = 7.041$, and hence

$$R^2 = 1 - \frac{3.159}{7.041} = 0.551 \quad (3.1.23)$$

In this example the simple regression model explains 55.1 percent of the total variation.

Exercise 3.6: Compute TSS, SSE, R^2

Consider data $\mathbf{x} = (0,0,2,2)$ and $\mathbf{y} = (1, -1, 3, 1)$. When you fit the regression line $y_i = \beta_0 + \beta_1 x_i + \epsilon_i$, you find that $\beta_0 = 0$ and $\beta_1 = 1$.

Compute i) TSS;ii) SSE;iii) R^2 .

Solution on page [424](#)

R^2 is not an universal measure. It is computed from prediction errors e and hence it focuses on prediction. If our task is to predict, we should strive to get as high R^2 as possible. But if our primary focus is inference, interpretation of β -s, the high R^2 value is of less importance. For instance, in the Hubble regression, the most important result is that $\beta_1 > 0$ —the universe is expanding. The errors are related to the measurement errors and to the fact that galaxies are also moving in space, not just with space. R^2 describes the ratio of these factors—expansion of universe, measurement errors, and the proper motion of galaxies; and this is much less interesting than the fact that the universe is expanding.

Different type of data lead to different R^2 values. In social sciences, it is common to observe R^2 in a range of 0.2...0.3 for ordinary regressions—this just means that human behavior is hard to predict. Accessible data just do not have the information needed to tell what humans are up to, something that everyone who has lived together with a partner has probably noticed ☺. If we are interested in changes over time, we often find R^2 less than 0.05, and in contrary, if we are predicting future behavior based on the current behavior, R^2 may exceed 0.9.

Finally, note that when defining SSE, TSS and R^2 we did not make use of the fact that we are working with *linear* regression. In fact, all these measures are well defined for all supervised learning models with continuous outcomes. This includes nearest neighbors, trees and related methods, and neural networks, as long as the variable of interest is continuous.

Which of these measures—RMSE, R^2 , β -s or t -values should one focus on when evaluating a regression model? It depends:

- β -s, in particular the slop parameter β_1 , tell us something about how x and y are related. For instance, if we analyze income and education, then it may tell that an additional year of education is associated with \$8,000 more of yearly income. But it tells little about how good is the model, or if this figure is reliable.
- t -values focus on the reliability part. High t -values mean that the association between x and y is not just a random blip, but is indeed there in the dataset. However, it is just about the statistical reliability, not the size of the association.
- R^2 describes the overall model goodness from prediction perspective. High R^2 (close to 1) means that the model can capture most of the variability in data, low R^2 (close to 0) means that there is a lot of variation that the model does not capture. This is important for predictive modeling, but if our task is just to compute β , then it matters much less.
- RMSE describes the prediction errors. It is silent about the overall model performance, it also does not tell anything about the association between x

and y . It just gives and estimate how much off are predictions made by this model.

So typically β -s and t -values are more important for inferential modeling, and R^2 and RMSE for predictive modeling. We should add here that even if all four values look reasonable, the model may still be off—this is just a part of the diagnostics one ought to do when using linear regression.

Example 3.7: R^2 of Hubble diagram: 100 years later

When Hubble published paper in 1929, the cosmological data was very primitive from the 21st century viewpoint. We can replicate his results on modern data and compare the models. The figure below compares the original Hubble diagram (left) with the one that is based on modern data (right). Already a visual inspection suggests that the modern data is better aligned with a line.

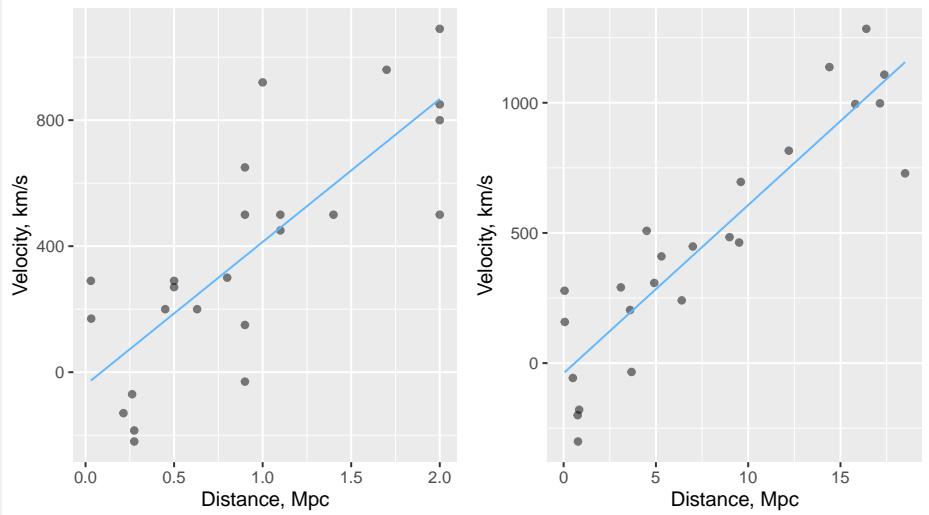


Figure 3.8: The original (1929) Hubble diagram (left) and its replication using the modern data for the same galaxies (right). A visual inspection suggests that a line fits the modern data better than the original data. One can also see that the modern distance estimates are up to 10 times larger than the original ones, the speed estimates have not changed that much.

Linear regression results for both data are in the table below:

	Original	Modern
Intercept	-40.44	-38.66
Distance	453.86	64.57
R^2	0.62	0.82

We can see that the same model using modern data provides noticeably better R^2 by explaining 82% of variation instead of 62% in case of the original data. But is it a *better model*? Do we want to improve R^2 even more?

These questions are a bit vague, but one may argue that the model is the same, just in the modern case we have better data (smaller measurement errors). So R^2 here is more of a data quality measure than the model goodness measure. But can we improve R^2 even more? Before improving it, we should understand why is the measured $R^2 < 1$ in the first place. As space is expanding uniformly (as far as we know), the fact that galaxies are not aligned perfectly with the line is due to two factors: measurement errors, and motion of galaxies in space (called *proper motion*). We would like to improve measurement precision, but extending the model to take into account the proper motion would require modeling the proper motion of galaxies, something that has little to do with the overall expansion. After all, this model is made to show that the universe is expanding, and less than perfect R^2 is not obscuring this message. Here a large R^2 is a nice-to-have feature, but not an essential one.

Cheatsheet 3.3: SSE and related terms

There are many acronyms related to sum of squared errors:

- *SSE*: Sum of squared errors $SSE = \sum_i e_i^2 = \sum_i (y_i - \hat{y}_i)^2$
- *MSE*: Mean squared error $MSE = \frac{1}{N} SSE$
- *RMSE*: Root mean squared error $RMSE = \sqrt{MSE}$
- *TSS*: Total sum of squares $TSS = \sum_{i=1}^N (y_i - \bar{y})^2$
- *R^2* : how much of total variation in data does the model expand: $R^2 = 1 - SSE/TSS$

3.1.6 Multiple Regression

Prerequisites: simple regression; linear algebra: vectors, matrices, Section 2.3.1 Vectors as matrices, page 106, matrix multiplication

What is Multiple Regression

In case of simple regression we are concerned with how a single explanatory variable x is associated with outcome y . But often we are interested in more than a single explanatory variable. For instance, in order to predict income, we may want to include education, but also age, gender and place of residence (rural or urban). So we do not have just a single explanatory variable x but more than one of those. This is the idea of *multiple regression*.

Technically, multiple regression is very similar to simple regression, just we allow several explanatory variables to influence the outcome y at the same time. Say we are interested in the effect of K explanatory variables. Now instead of (3.1.1) we write

$$y_i = \beta_0 + \beta_1 \cdot x_{1i} + \beta_2 \cdot x_{2i} + \cdots + \beta_K \cdot x_{Ki} + \epsilon_i. \quad (3.1.24) \quad (3.1.1): y_i = \beta_0 + x_i \cdot \beta_1 + \epsilon_i$$

For instance, in the income–education example above, we may have $K = 4$: x_1 is education, x_2 is age, x_3 is gender, and x_4 is place of residence. The outcome y is income. Exactly as in case of simple regression, we call y the outcome variable, x_k are explanatory variables and ϵ is the error term. The unknown parameters β_k are sometimes called slopes but more often just “betas”. And finally, index i stresses that each observation i has a different value for y , x_1 , x_2 , ..., x_K and ϵ , but they all share the same parameters $\beta_0 \dots \beta_K$. In a similar fashion we also generalize the expression for prediction (3.1.9) to multivariate case

$$\hat{y}(x) = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \dots \quad (3.1.25) \quad \begin{matrix} (3.1.9): \\ \hat{y}(x_1, x_2, \dots) = \beta_0 + \beta_1 \cdot x_1 \end{matrix}$$

While in case of the simple regression the predicted values form a line on $x - y$ plane, in multiple regression case it forms a K -dimensional *hyperplane* in $K + 1$ dimensional x and y -space.

Example 3.8: How is income related to education and literacy?

Let's analyze the relationship between income, education and illiteracy by U.S. states (R dataset *state.x77*, see also the example in Section 2.2.1). A sample of the data looks like

Income	HS Grad	Illiteracy
4449.00	50.20	1.00
5149.00	65.20	0.50
5237.00	52.50	1.10
5087.00	50.30	0.80

where *income* is in 1977 dollars, *HS Grad* is high-school graduation rate (pct), and *Illiteracy* is illiteracy rate (pct of population). We estimate multiple regression

model

$$\text{Income}_s = \beta_0 + \beta_1 \cdot \text{HSGrad}_s + \beta_2 \cdot \text{Illiteracy}_s + \epsilon_s$$

where s are states. The estimates are $\beta_0 = 2131.33$, $\beta_1 = 44.55$ and $\beta_2 = -52.64$.

Note that the estimates are not the same as when estimating two separate simple regression models. For instance, a model

$$\text{Income}_s = \beta_0 + \beta_1 \cdot \text{HSGrad}_s + \epsilon_s$$

would lead to estimates $\beta_0 = 1931.1$ and $\beta_1 = 47.16$ instead. See the page 150 below for the explanations related to direct and indirect effects.

Example 3.8 can also be visualized as that only contains two explanatory variables ($K = 2$) and hence the prediction hyperplane is a 2-D plane in 3-D space (Figure 3.9). The image depicts two of the explanatory variables, *HS Grad* and *Illiteracy* on the horizontal plane, and income on the vertical axis. The gray plane represents the model-predicted values—the *regression plane*. In a similar fashion as the linear model in two dimensions represents a line, in three dimensions it represents a plane $\hat{y} = \beta_0 \cdot \text{HSGrad} + \beta_2 \cdot \text{Illiteracy}$. The figure indicates that the plane is sloping upward toward higher HS graduation rate, the slope along the illiteracy axis is almost invisible. The large blue and yellow dots represent the actual income values with those below the regression plane barely visible, small dots are the corresponding model-predicted values. The vertical lines that connect the small dots of predicted values with large dots of actual values are the residual errors.

We can see that the regression plane splits the data points in space through the middle with roughly a half of the actual points above it and another half below it. This is similar to the 2-D picture (see e.g. Figure 3.1) where the line splits the point cloud on a plane in a similar fashion.

When incorporating three explanatory variables, the figure should contain a 3-D hyperplane in a 4-D hyperspace, but unfortunately neither our tools nor our brains can handle 4-D visualizations. In higher dimensions we can only visualize similar regression planes that represent a higher-dimensional model where some of the variables are held constant. However, such visualizations may be quite misleading.

Interpreting multiple regression effects

Interpretation of multiple regression coefficients is conceptually similar to that of simple regression. However, multiple regression allows to eliminate indirect effects and look at only direct effects. Imagine we are interested of the effect of education on income.⁸ We estimate a model of form

$$\text{Education}_i = \beta_0 + \text{Income}_i \cdot \beta_1 + \epsilon_i \quad (3.1.26)$$

But education and income may be related through different mechanisms. One, and the most intuitive one is the “direct effect” where education directly influences the

⁸As “effect” we mean association, all sorts of relationships, including the causal effect. When not doing causal inference we usually talk about just “effects”. When analyzing causal influence we talk about “causal effects”.

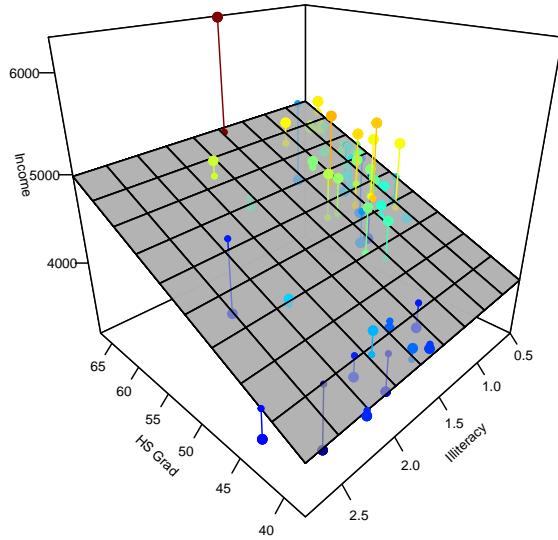


Figure 3.9: Regression plane with two explanatory variables (HS Grad and Illiteracy). The gray plane represents the 2-D regression plane, the large dots are the actual income values, the small dots are the predicted values on the regression plane, and the vertical lines that connect those values are the corresponding residual errors. Colors correspond to the actual income values.

income (e.g. if the employer pays higher salary for those with diploma). The direct effect may also go the other way around, e.g. if income determines what level of education one can afford. Both of these are direct effects (Figure 3.10, left panel). But this is not the only way these two variables are related. For instance, education also influences one's choice of where to live, e.g. in urban or rural area, and income differs by location. This is an indirect effect: education influences location, and location in turn influences income. The opposite causality is plausible as well where income determines where to live, and location determines the educational choice. It is the same indirect effect. When working with simple regression, we allow the location choice to change when education changes and hence what we measure is a sum of direct and indirect effect. (Obviously, there are more factors than just location choice that influence education and income, so it may be better to talk about indirect effects in plural.) This is manifested by the fact that we do not include any information about location in the model. The only explanatory variable is education.

But this is not the case of multiple regression where we estimate a model of a form

$$\text{Education}_i = \beta_0 + \text{Income}_i \cdot \beta_1 + \text{Location}_i \cdot \beta_2 + \epsilon_i. \quad (3.1.27)$$

Here we include location as an additional explanatory variable and hence it cannot just change as education changes—now it is determined by data. As a result, the first indirect effect between Education and Urban/rural choice (Figure 3.10, right panel) is broken. Education is not allowed to influence the location choice in an arbitrary

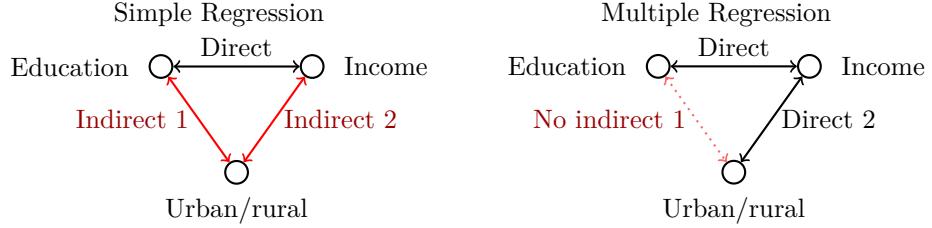


Figure 3.10: Analyzing the effect of education on income using simple regression (left) and multiple regression (right). Simple regression includes indirect effect, the red line from education over urban/rural location to income. Multiple regression fixes the urban/rural choice through data and in this way breaks the line between education and location. Only direct effect (black line) is left. Two-way arrows stress that the causality may run in both directions across the links.

way inside the model—all influence is captured by data. What is left are two direct effects—from education to income, and from location choice to income (or the other way around as we cannot tell whether these factors cause income or income causes these factors). More realistically, there are always more variables we cannot control, so we just remove some of the indirect effects but still leave others in the model. This process, including explanatory variables that remove the respective indirect effects from the model, is called *controlling for* these variables. So in the example above we analyze the relationship between education and income, while *controlling for* geographic location.

Now back to interpreting the numerical values of β -s. Multiple regression interpretation is fairly similar to that of simple regression. First, we immediately see from (3.1.24) that intercept corresponds to the expected outcome value *given all explanatory variables have value 0*. It often refers to an unrealistic, or even impossible case, e.g. income where age and education are 0. We rarely find the intercept to be an interesting parameter.

However, the other coefficients are typically interesting. As visible from (3.1.24), if x_1 is larger by one unit, then predicted y is larger by β_1 units, and if x_2 is larger by one unit then y is larger by β_2 units, and so on. However, note that for this to be true we have to keep all other x -s fixed while increasing x_1 , or while increasing any particular x . So the interpretation is the following: β_k shows how many units larger y corresponds to one unit larger x_k (in average) *while other explanatory variables remain at the same level*.

Let's return to the income-education-location example. In case of simple regression, the coefficient means “how much more will those workers earn who have one more year of education. We compare more and less educated workers, and compute the difference in their earnings. In case of multiple regression, the coefficient means “how much more will those workers earn who have one more year of education, *given their place of residence is the same*”. Hence we compare more and less educated workers *in the same place*, and compute the difference in their earnings. These are different questions and typically lead to different answers. In the simple regression case we include location choice as one potential way how more educated workers can

increase their income. In multiple regression case we compare workers in the same location and hence exclude that mechanism.

Example 3.9: Income, education and literacy: interpretation

Let's now interpret the results of Example 3.8. The results were $\beta_0 = 2131.3$, $\beta_1 = 44.6$ and $\beta_2 = -52.6$. The interpretation of intercept β_0 —income in a hypothetical state with no high-school graduates but also no illiterates is \$2131.3—is not particularly interesting. But β_1 tells us that in case of two states that have similar illiteracy levels, the one with 1 pct pt higher HS graduation rate has \$44.6 larger income. This is an interesting outcome. In a similar fashion, β_2 tells that among two states with similar HS graduation rate, the state with 1 pct pt larger illiteracy has \$52.6 lower income. This is clearly relevant as well.

However, if we estimate a simple regression model that only contains *HS Grad* but no *Illiteracy*, then the estimated value is a bit larger, $\beta_1 = 47.2$ (see Example 3.8). The difference is related to indirect effects: states with high HS graduation rates tend to have low levels of illiteracy, and low illiteracy adds to the income. This is an indirect effect of HS graduation rate. This path is blocked when we control for illiteracy and hence we get a lower estimate for HS graduation rate.

Note that we are not talking about causality here. Low HS graduation rates tend to be associated with high illiteracy rates, but that does not mean that more easily accessible high schools would cause illiteracy to be low.

When is it advantageous to use multiple regression? Direct effects, identified in multiple regression, are easier to interpret and it is easier to base policy implications on these. For instance, imagine that we conduct two simple regression analyses and find that better income is associated with better income, as does living in a certain geographic area. What should we recommend to do? Improve education, or build more homes in that region?⁹ In both cases the message is clear but different. In multiple regression case we can actually disentangle these two effects and tell which one is more important. It is not possible using just simple regression. But simple regression is more appropriate in other cases. For instance, if you want to know whether better educated individuals earn more then location does not matter. What you want is just the simple regression analysis.

Example 3.8: regression model

$$\text{Income}_s = \beta_0 + \beta_1 \cdot \text{HSGrad}_s + \beta_2 \cdot \text{Illiteracy}_s + \epsilon_s$$

using U.S. states' data

⁹Here we talk about causal effects. It is rare in practice that we can identify causal effects although these are usually what policymakers need to make decisions.

$\mathbf{x} = (x_1, x_2, \dots, x_K)^\top$ is a shorthand for $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_K \end{pmatrix}$. See Section 2.3.1, page 106.

$\boldsymbol{\beta}^\top \cdot \mathbf{x}_i = \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_K x_{iK}$, see (2.3.31), page 112.

Formal Definition of Multiple Regression

From now on we follow vector-based formalism that makes notation, mathematics, and numerical computations substantially simpler. We stack all explanatory variables x_i into a vector \mathbf{x} , a shortcut for $\mathbf{x} = (x_1, x_2, \dots, x_K)^\top$. Now the multiple regression definition 3.1.24 can be written as

$$y_i = \beta_0 + \boldsymbol{\beta}^\top \cdot \mathbf{x}_i + \epsilon_i. \quad (3.1.28)$$

To simplify the notation further, the constant “1” is often included as the first (0-th) component of \mathbf{x} and hence \mathbf{x} is defined as $\mathbf{x} = (1, x_1, x_2, \dots, x_K)^\top$. Note that, strictly speaking, it is not correct to refer \mathbf{x} now as “data” or “variables”, unless you are willing to refer to a constant as “data”. But this trick helps us to simplify notation even further, and we still call it somewhat sloppily “data”. So the regression model in its final vector form is written as

$$y_i = \boldsymbol{\beta}^\top \cdot \mathbf{x}_i + \epsilon_i. \quad (3.1.29)$$

Note that whatever is the number of variables K , the vector form (3.1.29) remains the same. Vectors allow us to abstract away from K , both in notation and in computer code. Using vector notation we can write the predicted values analogously, as

$$\hat{y}_i = \hat{\boldsymbol{\beta}}^\top \cdot \mathbf{x}_i \quad (3.1.30)$$

where $\hat{\boldsymbol{\beta}}$ is the vector of estimated parameter values.

Now we can generalize the definition of simple regression (3.1.16) to multiple regression. We just use the multiple regression predictions (3.1.30) to define the sum-of-squared-errors (SSE):

$$\begin{aligned} SSE(\boldsymbol{\beta}) &= \sum_{i=1}^N e_i^2 = \\ &= \sum_{i=1}^N (y_i - \hat{y}_i)^2 = \\ &= \sum_{i=1}^n (y_i - \mathbf{x}_i^\top \cdot \boldsymbol{\beta})^2. \end{aligned} \quad (3.1.31)$$

The notation we use stresses that SSE depends on the parameter vector $\boldsymbol{\beta}$. The solution $\hat{\boldsymbol{\beta}}$ is just the parameter vector that minimizes $SSE(\boldsymbol{\beta})$:

$$\hat{\boldsymbol{\beta}} = \arg \min_{\boldsymbol{\beta}} SSE(\boldsymbol{\beta}). \quad (3.1.32)$$

This minimization problem can be solved analytically, see Section 3.1.10, page 176.

3.1.7 Categorical Variables

So far we have assumed that all our variables are numeric and hence the multiplication $\boldsymbol{\beta} \cdot \mathbf{x}$ is possible. But there are many types of data that are not numeric. For instance,

gender is often recorded as dichotomous label “male” or “female”. Home type may be either “rental apartment”, “condo”, “single-family home” or “other”. And some variables, although coded as numbers, are not really numbers. For instance, family status may be coded as 1–single, 2–married, 3–divorced, etc. Such variables cannot be directly included into regression models, and even if done so (we *can include* the numerical categories for the marital status variable above), the results are probably wrong and misleading. The problem stems from the measure type—we can only do multiplication and addition with interval or ratio measures. But house type and family status are nominal measures, even if coded as numbers.

Consider the *Males* dataset (see page 413) that contains wages of 545 young men in 1980s. We are going to describe the wage as a function of marital status, and ethnicity. In the dataset we have marital status (variable *married*) coded as “yes” and “no” for married and non-married men respectively. Ethnicity (variable *ethn*) is coded as “black”, “hispanic”, and “other”. To give you better idea of the data, Table 3.5 left part shows a small sample of it. *wage* refers to log hourly wage.

We would like to estimate a regression model along the lines:

$$\log \text{wage}_i = \beta_0 + \beta_m \cdot \text{married}_i + \beta_e \cdot \text{ethn}_i + \epsilon_i \quad (3.1.33)$$

However, we cannot use the existing variables in a model like this as both marital status and ethnicity are not numbers but categories. Hence we have to somehow convert these variables into numeric ones. The most popular approach to transform categorical variables into numbers is by creating *dummy variables (dummies)*. Dummies are called so because they are “dummy”, simple variables that can only take two values: 0 and 1.

Let’s start with *married*. This is a two-category variable with two possible values, “yes” and “no”. An obvious choice is to convert it to binary 0/1 variable where “0” refers to “no” and “1” refers to yes. Let’s call the variable *m* (Table 3.5 middle column).

Table 3.5: Sample of Males data (left), binary (dummy) variable *m* denoting status “married” (center). Dummies for three possible ethnic categories are in the rightmost three columns.

wage	married	ethn	<i>m</i>	<i>e_b</i>	<i>e_h</i>	<i>e_o</i>
1.20	no	other	0	0	0	1
1.52	yes	other	1	0	0	1
1.46	no	black	0	1	0	0
1.69	yes	black	1	1	0	0
1.12	no	hisp	0	0	1	0
2.22	yes	hisp	1	0	1	0

As *m* is numeric, we can use it directly in the regression model like

$$\log \text{wage}_i = \beta_0 + \beta_m \cdot m_i + \epsilon_i. \quad (3.1.34)$$

This amounts to fitting just β_0 for non-married men (as their $m = 0$) and $\beta_0 + \beta_1$ for the married men. If we do this, we get the following results:

Interval measure: difference is defined; ratio measure: origin (zero) defined; nominal measure: only equality defined. See Section 1.1.1 page 2.

	Estimate	Std. Error	t-value	Pr(> t)
Intercept	1.5524	0.0105	147.28	0.0000
m	0.2203	0.0159	13.85	0.0000

The basic interpretation of the result is the same as in case of ordinary regression (see Section 3.1.3):

- “Intercept” gives the average outcome value where all other explanatory variables are 0. In this case this means intercept corresponds to the average log-wage where $m = 0$, i.e. average log-wage for those who are not married. Non-married men earn 1.55 log units in average.
- “m” describes extra log wage of those who have one unit larger m . So men who have $m = 1$ have average log-income larger by 0.22 units compared to men with $m = 0$. Or in plain language, married men earn more by 0.22 (in log terms), in total 1.77.

Interpretation can also be understood from the fact that we are fitting β_0 for the unmarried and $\beta_0 + \beta_1$ for the married men, hence β_0 must describe the unmarried and β_1 the difference between married and unmarried men.¹⁰

So we managed to include a categorical variable into our model. The interpretation tells us how much do the corresponding categories’ outcome differ, in average. It was rather easy in case of two categories.

Exercise 3.7: Do union members earn more?

The Males data also includes union membership (either “yes” or “no”). We can create analogous dummy $u = 1$ for union members and 0 for non-members. When running a simple regression

$$\log(wage_i) = \beta_0 + \beta_1 \cdot u_i + \epsilon_i \quad (3.1.35)$$

we get the following results:

	Estimate	Std. Error	t-value	Pr(> t)
Intercept	1.605	0.009	174.87	0.000
u	0.179	0.019	9.65	0.000

Use this table to answer the following questions:

1. What is the log-wage for non-union members? (in average)
2. What is the log-wage for union members? (in average)?
3. How big is the difference in favor of the union members?

Solution on page 425

¹⁰While this is the most common way of introducing dummies, there are other options. For instance, it is possible to specify the model in a way that β_0 is the average wage for unmarried and β_1 is that for the married men. Different specifications are suited for different questions. For instance, the original specification where β_1 captures the difference between married and non-married men is well suited to answer “Do married men earn more”?

The variable *married* has only two categories and hence we managed to transform it into a single dummy m . But how to handle *ethn* that has three possible nominal values? In this case we need to create *two dummies*. In order to understand the process better, let's start by creating three dummies, e_b , e_h and e_o in a way that if $\text{ethn} = \text{black}$ then $e_b = 1$, $e_h = 0$ and $e_o = 0$; if $\text{ethn} = \text{hisp}$ then $e_b = 0$, $e_h = 1$ and $e_o = 0$; and if $\text{ethn} = \text{other}$ then $e_b = 0$ and the other two e -dummies are both 0. So we have converted one column with three different values into three columns with two values each. (This is sometimes called *one-hot encoding*.) The resulting dummies are given in Table 3.5 in the three rightmost columns. Intuitively, one might now want to estimate a model as

$$\log(wage_i) = \beta_0 + \beta_b \cdot e_{bi} + \beta_h \cdot e_{hi} + \beta_o \cdot e_{oi} + \epsilon_i \quad (3.1.36)$$

but this will not work. To see why, let's look what are the coefficients describing. For blacks, the estimated log wage would be $\beta_0 + \beta_b$, for hispanics $\beta_0 + \beta_h$ and for others it will be $\beta_0 + \beta_o$. We have four β -s but only three groups, and hence we cannot determine all four β -s at the same time. For instance, if we add 1 to β_0 while subtracting 1 from β_b , β_h and β_o at the same time, the predictions will remain exactly the same. We cannot identify all β -s.

As a solution, it is customary to leave out one category, called *reference category*. Statistical software typically picks the first category as the reference category, we follow this habit here. So instead of (3.1.36) we estimate the model

$$\log(wage_i) = \beta_0 + \beta_h \cdot e_{hi} + \beta_o \cdot e_{oi} + \epsilon_i. \quad (3.1.37)$$

The estimation results are below:

	Estimate	Std. Error	t-value	Pr(> t)
Intercept	1.5231	0.0236	64.46	0.0000
e_h	0.0983	0.0312	3.15	0.0016
e_o	0.1521	0.0254	5.98	0.0000

The multi-category dummies are slightly harder to interpret, although the basics are the same:

- “Intercept” describes the log-wage in case where all explanatory variables are zero. Here we have just two explanatory variables, e_h and e_o as we left e_b out as reference. Because of how the dummies are constructed, if both $e_h = 0$ and $e_o = 0$, we must have $e_b = 1$. This means when all explanatory variables are zero, we are looking at the reference category, blacks (as e_b is not included in the model, it does not count as an explanatory variable). Hence *Intercept describes the outcome for the reference category!* In principle we could add an extra line to the table:

	Estimate	Std. Error	t-value	Pr(> t)
e_b	0.0000	0.0000	0.0000	0.0000

i.e. we can imagine the dummy for the reference category is included in the table, just its value is exactly zero. This is sometimes done, in fact, to make the reference category more explicit.

- The other dummies have the ordinary meaning. e_h describes additional salary for men who have $e_h = 1$ instead of $e_h = 0$ while keeping e_o constant, i.e. it describes the extra salary for hispanics compared to blacks (remember: if $e_h = 1$ then e_b must be 0). The interpretation for e_o is similar.

In summary, in case of multi-category dummies, intercept describes the reference category and estimates for the other dummies describe the difference between the corresponding groups and the reference group. It is crucial to know what is the reference category in order to understand the results. Note that we can describe two-category dummies in exactly the same way: we create two categories (married and non-married) and left the non-married out as the reference category.

Exercise 3.8: Interpret multi-category dummies

The Males dataset also includes a variable *residence* that describes the geographic location. These are *rural area*, *north east*, *northern central* and *south*. When estimating the model where we explain the log wage with the geographic location, we get the following results:

	Estimate	Std. Error	t-value	Pr(> t)
Intercept	1.584	0.057	27.6	0.000
north east	0.164	0.061	2.7	0.007
nothern central	0.047	0.060	0.8	0.431
south	0.032	0.059	0.5	0.591

1. What is the reference group for variable *residence*?
2. What is the predicted log wage in Northern Central?
3. What is the predicted log wage in rural areas?
4. How much larger (or smaller) is log wage in South compared to rural areas?
5. How much larger (or smaller) is log wage in North East compared to South?

Answer on page [425](#)

Exercise 3.9: Why a single race only?

Consider the example with income and ethnicity above. We repeatedly stressed that the ethnicity dummies are mutually exclusive, e.g. if $e_h = 1$ then e_b must be 0. Why this? Why cannot we allow multi-racial individuals?

Answer on page [425](#)

Cheatsheet 3.4: Categorical variables in linear regression

Introducing and interpreting categorical variables in linear regression goes like this:

1. Convert categorical variables to dummies. You need one dummy for each category, e.g. in case of 10 cities you get 10 different dummies. The dummies are coded are mutually exclusive, for each observation one and only one dummy has value “1” while all others have value “0”.
2. Leave one dummy out as the reference category.
3. Interpretation:
 - Intercept: predicted value for the reference category
 - β_c : predicted difference between the category c and the reference category.

Always report what is your reference category!

3.1.8 Feature Transformation

Prerequisites: [Log-normal distribution, page 56](#)

Standardization TBD: standardized features

Log-transformation Many types of data, such as income or price, have a well-defined lower bound but no obvious upper bound. The corresponding distributions tend not to look normal but are more similar to log-normal (Figure 3.11) and hence violate the assumptions we need to compute standard errors (see Section 3.1.10, page 171). An obvious remedy is to analyze log-income instead of income and in empirical literature income analysis is almost universally done in log form. In such case, transforming your outcome variable into log outcome has two main advantages, one theoretical and one data-driven.

1. If distribution of log-income is more similar to normal, the issue of violating the normality assumption is likely small. This is the theoretical advantage.
2. Second advantage is data driven, and is typically correct in this type of data. Namely, log transformation improves the predictive power of the model (increases R^2), often by a substantial amount. This, in turn, is related to the fact that this type of data is often created not by additive processes but by multiplicative processes (see below).

Let's analyze the effect of log-transform in context of simple regression. When transforming the outcome to $\log y$, we can write the model as

$$\log y_i = \beta_0 + \beta_1 \cdot x_i + \epsilon_i. \quad (3.1.38)$$

Taking exponent of both sides we can transform it back to non-log form:

$$y_i = e^{\beta_0} \cdot e^{\beta_1 \cdot x_i} \cdot e^{\epsilon_i}. \quad (3.1.39)$$

This is not a linear model but a multiplicative model: y is not a sum but a product of three different terms:

1. e^{β_0} is the value of y in case both $x = 0$ and $\epsilon = 0$. This is the analogue of intercept.
2. e^{β_1} describes the relationship between y and one unit larger x : the cases that have one unit larger x have outcome y larger by e^{β_1} times.
3. and finally, e^{ϵ_i} is the (multiplicative!) error term.

The second advantage, in other words, is an empirical regularity: it appears that fat-tailed outcome can typically be better explained by multiplicative models instead of additive models.

Finally, we also discuss the interpretation. The basic interpretation of the model is always the same but as our outcome now is $\log y$, we now have that one unit larger

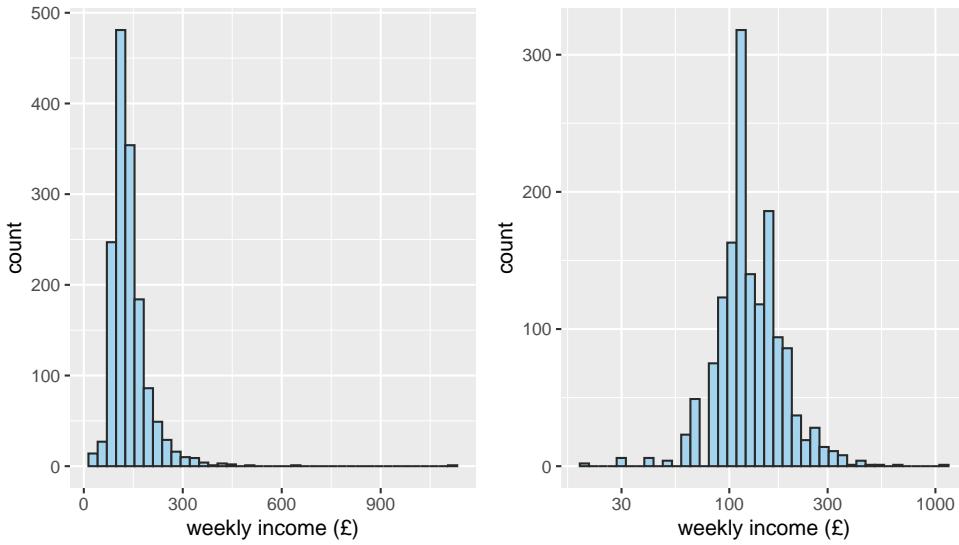


Figure 3.11: Distribution of UK household income in early 1980-s. Income distribution (left panel) does not look normal, it has a long thin tail of high-income households reaching up to weekly income 1000£. Log income (right panel) is fairly close to normal as logarithm spreads low-income observations out and squeezes the high-income ones closer together. Ecdat package data.

x corresponds to β_1 units larger $\log y$ (not y !). When transforming the model back into non-log form (outcome is y , not $\log y$), we can restate the interpretation as *one unit larger x is associated with e^{β_1} times larger y* . If β_1 is small, then $e^{\beta_1} \approx 1 + \beta_1$ and we can say that it describes how many percent larger y we tend to observe when x is larger by one unit.¹¹ For instance, if $\beta = 0.1$, $e^\beta = 1.105 \approx 1.1$. Remember, this is a multiplicative effect and hence we can say that one unit larger x is associated with 10 percent larger y .

Example 3.10: How does income depend on age?

Let us use the same UK budget dataset as in Figure 3.11 above. The data include age of the household head (between 19 and 60). We convert this to four age categories (-29, 30-39, 40-49, 50-) and estimate the regression model in the form

$$\log y_i = \beta_0 + \boldsymbol{\beta}^\top \mathbf{a}_i + \epsilon_i$$

where \mathbf{a} is a vector of the corresponding age category dummies. The results are

¹¹Note that this interpretation does not hold if one uses decimal logarithm instead of natural logarithm because $10^\beta \not\approx 1 + \beta$ even if β is small.

	Estimate	Std. Error	t-value	Pr(> t)
Intercept	4.669	0.018	258.26	0.000
30-39	0.213	0.022	9.49	0.000
40-49	0.297	0.028	10.69	0.000
50-	0.174	0.046	3.79	0.000

Interpretation of the results is as follows:

- The reference category is the “-29”, the one that is missing in the table.
- *Intercept* indicates that the expected log-income for households in the reference category is 4.669.
- *30-39* indicates that households where the head is 30-39 years old earn 0.213 more in log-units (in average). This means they earn $e^{0.213} = 1.237$ times more, or 23.7 percent more than the reference category.
- Analogously, *40-49* year old households earn 0.297 more in log-units, i.e. $e^{0.297} = 1.345$ times more, or 34.5 percent more than the reference category.
- Finally, the over-50 households earn more than the reference category but less than the middle-aged households.

R^2 of the model is 0.082. For comparison, R^2 of linear model, without log-transforming income, is 0.063, indicating that log-transform improves the model. This is not an impressive number, but realistically, we should not expect to be able to predict household income well based just age of its head.

Log-log transformation In certain type of data, it may be advantageous to log-transform not just y but also x and hence to look at the model

$$\log y_i = \beta_0 + \beta_1 \log x_i + \epsilon_i. \quad (3.1.40)$$

The standard interpretation sounds like “ $\log y$ is larger by β_1 units in observations that have $\log x$ larger by one unit”. In order to find the interpretation of β_1 , we can again take exponent of both sides. We get

$$y_i = e^{\beta_0} \cdot x_i^{\beta_1} \cdot e^{\epsilon_i}. \quad (3.1.41)$$

This models suggests it is worthwhile to look at a case where x is larger by a certain proportion, say by α percent. In that case y will be larger $(1 + \alpha)^{\beta_1}$ times. If we choose a small α (for example, 1 percent, i.e. $\alpha = 0.01$), this is approximately equal to $(1 + \alpha)^{\beta_1} \approx 1 + \alpha\beta_1$. Hence we can interpret it as *how many percent is y larger when x is larger by one percent*. This figure is often called *elasticity*. Compare the interpretation of log-transformed and log-log transformed data. In the former case we find *percentage increase per unit increase in x*, in the latter *percentage increase per one percent increase in x*. Cheatseet 3.5 summarizes the interpretation of the regression coefficients. As multiple regression model can include both log-transformed and not

log transformed predictors, different model estimates may have to be interpreted in different ways.

Example 3.11: Linear, log-linear, and log-log transformations

We use linear regression to analyze the relationship between price and mass of diamonds (data from R package *ggplot2*). Figure 3.12 shows the relationship for no transformation, log-transformation, and log-log transformation. Just a visual impression suggests that the latter fits best to a line.

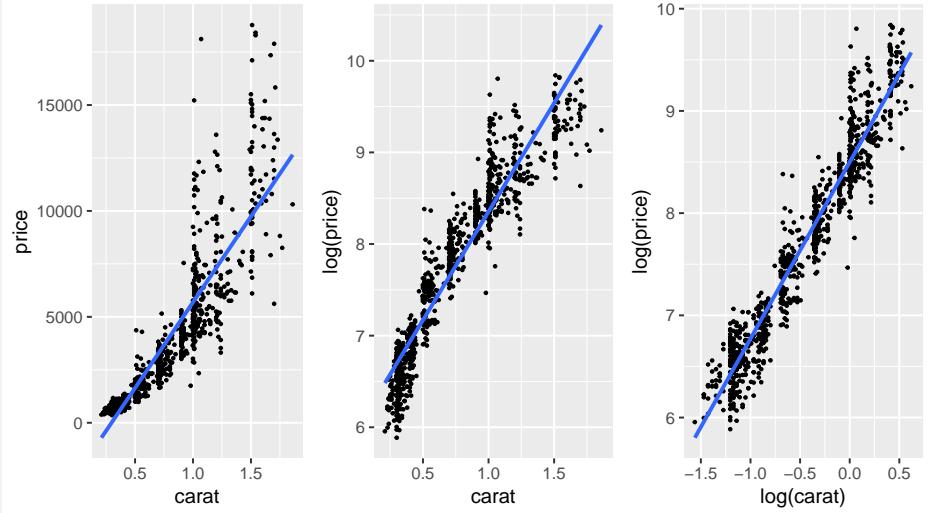


Figure 3.12: Diamond mass ($\text{carat}=0.2$ gram) and price data, including the corresponding regression lines. Left panel shows the linear model in price and carat. One can see that the line does not capture the convex pattern in data. Middle panel shows a model that is linear in log price and carat. Now the data pattern is concave and again the line fails to capture it well. On the right panel we log-transform both variables, and the result looks very good visually.

Next we analyze how do the corresponding linear regression models look like:

Table 3.6: Results of three different regression models: linear-linear, log-linear, and log-log.

.	object.
Intercept	-2406.107*** 112.179	5.986*** 0.023	8.500*** 0.011
carat	8096.442*** 134.733	2.368*** 0.028	
log(carat)			1.728*** 0.016
# obs	964	964	964
R^2	0.7896	0.8850	0.9254

We can see that the log-log model has the best predictive power (highest R^2) while linear-linear has the worst R^2 . The corresponding regression coefficients can be interpreted as follows: for linear-linear model, β_1 means that one carat heavier diamonds are 8096.442 dollars more expensive. In log-linear model, 1 carat heavier diamonds are $e^{2.368} = 10.674$ times more expensive. Finally, log-log model suggests that 1 percent heavier diamonds are 1.728 percent more expensive.

TBD: other kind of feature engineering

Cheatsheet 3.5: Log transformations in linear regression

The table below summarizes interpretation of linear, log-linear and log-log models

Type	Interpretation of β_1
linear-linear ($y \sim x$)	one unit larger x is associated with β_1 unit larger y
els. log-linear ($\log y \sim x$)	one unit larger x is associated with β_1 percent larger y (only holds for small β_1 values)
log-log ($\log y \sim \log x$)	one percent larger x is associated with β_1 percent larger y

3.1.9 Interactions Effects

Interaction effects (also *cross-effects*) is a way to build regression model that does not just handle variables independently, but allows different outcomes for certain joint combinations of variables. This is one of the most widely used methods to add flexibility to regression models.

Artificial example

Let's look at an artificial example.¹² Consider an analysis where we are interested in income as a function of cognitive skills and social skills.¹³ Assume we have collected data on personal income, performed a test for cognitive skills (such as IQ test), and assessed the social skills too. Let us measure both skills in a binary fashion: low (0) and high (1). Take a look at the four individuals (*a*, *b*, *c*, and *d*) in Table 3.7.

Table 3.7: Example skill-income data.

	1	2	3	4	5	6
id	Annual income, \$	Social	Skills Cognitive	Interaction Social × Cognitive	Captured by	
a	40,000	0	0	0	β_0	
b	60,000	0	1	0	$\beta_0 + \beta_c$	
c	50,000	1	0	0	$\beta_0 + \beta_s$	
d	100,000	1	1	1	$\beta_0 + \beta_s + \beta_c + \beta_{sc}$	

We focus on the first four columns for now. The baseline individual *a*, the one with low social and low cognitive skills, earns \$40,000 a year. The next one, individual *b*, has low social skills but high cognitive skills and makes \$60,000, i.e. \$20,000 more than individual *a*. This suggest that the effect of cognitive skills is \$20,000. No surprise, cognitive skills are valuable. However, when we compare individuals *c* and *d*, we see that adding cognitive skills for someone who already has high level of social skills improves her income by \$50,000. Cognitive skills are even more valuable for someone who has more social skills.

This effect cannot be captured by the baseline multiple regression model. If we were to estimate the data by a model like

$$\text{income}_i = \beta_0 + \beta_s \cdot \text{social skills}_i + \beta_c \cdot \text{cognitive skills}_i + \epsilon_i, \quad (3.1.42)$$

we will interpret β_c as the effect of cognitive skills, no matter what is the level of social skills. If we run such a linear regression on these data, we get $\beta_s = \$25000$ and

¹²See Deming (2017).

¹³Cognitive skills are skills that required for conscious mental work, such as reading, learning, math. These can be measured with standard tests, such as IQ or AFQT. Social skills are skills we use in human communication and persuasion, and include a plethora of small-scale behavioral habits that are hard to assess and train consciously.

TBD: find a few good papers.

$\beta_c = \$35000$. The latter figure corresponds to the average effect of cognitive skills effect for low- and high social-skilled individuals (i.e. the average of 20 and 50). But what if we want to capture the fact that higher social skills are related to a larger effect of cognitive skills?

This can be done by amending the model (3.1.42) with an *interaction term*, $\beta_{sc} \cdot \text{social skills}_i \times \text{cognitive skills}_i$. From practical perspective, the interaction term is equivalent to creating a new variable, social skills \times cognitive skills (see column 5 in Table 3.7), and adding it into the regression model as just another feature. Modern software typically has handy shortcuts for this operation, so usually you do not need to create such additional variables explicitly. So the corresponding linear regression model with an interaction effect will look like

$$\begin{aligned} \text{income}_i = & \beta_0 + \beta_s \cdot \text{social skills}_i + \beta_c \cdot \text{cognitive skills}_i + \\ & + \beta_{sc} \cdot \text{social skills}_i \times \text{cognitive skills}_i + \epsilon_i. \end{aligned} \quad (3.1.43)$$

When we estimate this regression model, we get $\beta_0 = \$40000$, $\beta_s = \$10000$, $\beta_c = \$20000$ and $\beta_{sc} = \$30000$.

Unfortunately, interaction effects make regression models harder to interpret. The basic interpretation remains the same: β tells how much larger is the expected outcome for those who have the variable's value larger by one unit. However, now the variable values are not independent any more. We cannot have social skills \times cognitive skills = 1 if the person has social skills = 0. So we cannot just conclude that “those with high social skills earn 10000 more than those with low social skills” as β_1 suggests. Now the effect size depends on the level of cognitive skills.

Interpreting the Interaction Effects

In order to interpret the results, let us start by predicting the income for everyone in data. As even experienced researchers get confused by the interaction effects, it is helpful to write down the table of dummies for each four individuals (Table 3.7, columns 3-5). Importantly, we have also marked the interaction effect here (column 5). Each of the dummy columns corresponds to one variable in the regression model (3.1.43) and hence to the respective β .

Consider the first individual a who has both low social and low cognitive skills. She has all the explanatory variables equal to 0, so her predicted income will just be $y_a = \beta_0 = \$40000$ (see the last column in the table). Next, for the individual b who has low social skills but high cognitive skills, we have $y_b = \beta_0 + \beta_c = \60000 as b has cognitive skills dummy equal to unity. Individual c has low cognitive skills but high social skills and hence her income is $y_c = \beta_0 + \beta_s = \50000 . Finally, d has both high social and high cognitive skills, and hence her $\text{social} \times \text{cognitive} = 1$ as well. Her income is accordingly $y_d = \beta_0 + \beta_s + \beta_c + \beta_{sc} = \100000 . The summary of modeled effects are in the last column in Table 3.7.

In order to interpret the interaction effect β_{sc} , we compute the income differences. Individuals b and a have low social skills and their income difference is only due to the effect of cognitive skills $\beta_c = \$20000$. Individuals c and d have high social skills and their income difference is captured by sum of two coefficients, $\beta_c + \beta_{sc} = \$50000$ as individual d has both cognitive skills and the interaction effect non-zero.

So in conclusion we can interpret the interaction effect β_{sc} as the *additional effect of cognitive skills* for those individuals who have high social skills.

Sometimes it is worthwhile to present the effect in a graphical form (Figure 3.13). The figure depicts two lines: the blue line describes the relationship between cognitive skills and income for low-social skill individuals, and the red line depicts the relationship for high-social skill individuals. Red line is steeper than the blue line, indicating that high-social skill individuals gain more from cognitive skills. If there were no interaction effects, the high-social skilled individuals would be represented just by an upward shift of the blue line (marked by dots). However, because high skills in both dimensions complement each other, the red line is steeper, and the “extra steepness” is captured by β_{sc} .

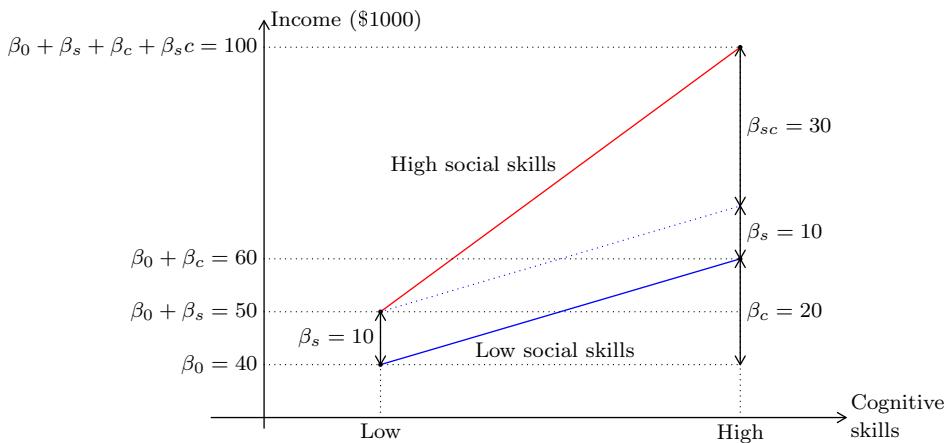


Figure 3.13: Interpretation of interaction effects. The blue line depicts the relationship between income and social skills for low-social-skill individuals, and red line that for the high-social-skills individuals.

Interaction effects are a popular way to add flexibility to the linear regression and other similar models. The result is not linear any more in the original features ($social \times cognitive$ is not a linear term!) but it is still a linear function in the extended feature set where $social \times cognitive$ forms a separate feature. But the added flexibility comes with a cost—more complex interpretation. While the model (3.1.43) is not hard to interpret, we have lost the beauty of the original model: β_1 and β_2 are not the universal effects of social and cognitive skills any more. The effect of one factor depends on the level of another factor.¹⁴

One can easily extend the interaction effects to multi-category variables, and to continuous variables. It is also easy to introduce 3-way interactions but those are substantially more demanding to interpret. However, if we are only interested in prediction then interpretation is not a major concern.

¹⁴In certain literature, in particular in psychology, the sentence is often phrased as “the effect of one variable is *moderated* by another one”.

Example 3.12: Importance of social skills

Deming (2017) analyzes the effect of cognitive and social skills on wage. He uses NLSY data^a to establish cognitive skills, and workplace occupational requirements to associate jobs with social skills. Both skills variables are **standardized**, i.e. their average value is zero. He uses a linear regression model of the form

$$\begin{aligned} \text{log wage}_i = & \beta_0 + \beta_1 \cdot \text{cognitive skills}_i + \beta_2 \cdot \text{social skills}_i + \\ & + \beta_3 \cdot \text{cognitive skills}_i \times \text{social skills}_i + \beta_4 \cdot \mathbf{X}_i + \epsilon_i \end{aligned} \quad (3.1.44)$$

where \mathbf{X} are the other individual characteristics besides of the skills. His results are

variable	effect	std.error
cognitive skills	0.206***	0.007
social skills	0.049***	0.006
cognitive \times social	0.019***	0.006
R^2	0.344	

where *** means the estimate is significant at 1% confidence level. These outcomes have the following interpretation (see Section 3.1.6 on page 150):

- One unit larger cognitive skills^b are related to 0.206 units larger log income (i.e. $e^{0.206} = 1.220$ times larger wage, see [log-transformation](#)) for those with social skills equal to zero (i.e. average social skills).
- One unit larger social skills are associated with 0.049 units larger log wage (i.e. $e^{0.049} = 1.05$ times larger wage) for those with cognitive skills zero (i.e. average cognitive skills).
- There is an additional log wage premium 0.019 (i.e. $e^{0.019} = 1.019$ times larger wage) for workers with both social and cognitive skills one unit above the mean. If both skills are two units above the mean, the log-premium is four times as large and the wage is $e^{4 \cdot 0.019} = e^{0.076} = 1.079$ times larger.

This is the central results of the study: cognitive skills are more valuable for workers with high social skills. Equivalently, this can be put in the other way around: social skills are more valuable for workers with high cognitive skills.

^aNational Longitudinal Survey of Youth

^b“Unit” in case of standardized features is their standard deviation.

When to use interactions?

When do we want to include interaction effects to the model? And what kind of interaction effects? This is something we have to decide, as the number of possible interaction effects will easily get out of hand. For instance, in case of three explanatory

variables, x_1 , x_2 and x_3 , the model with full interaction effects will be

$$\begin{aligned} y_i = & \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \\ & + \beta_{12} x_1 x_2 + \beta_{23} x_2 x_3 + \beta_{31} x_3 x_1 + \\ & + \beta_{123} x_1 x_2 x_3 + \epsilon_i. \end{aligned} \quad (3.1.45)$$

Note that this also includes a 3-way interaction effect $\beta_{123} x_1 x_2 x_3$. Interpreting such a model is rather complicated, it also has more stringent data requirements than a model without any interaction effects. So we cannot just include all kinds of interaction effects in all models.

There are a few good reasons to include these effects.

First, it may help to in terms of allowing the effect of interest to be more flexible. As the [artificial example](#) above shows, we may want to allow the effect of cognitive skills to depend on the level of social skills. Interaction effects is one convenient way of achieving this.

This usually only makes sense for the effect of interest, not for other variables. For instance, we might also include age and education interaction in the same model, but that would make the model more complicated, without necessarily given us any more insight. After all, we are interested in cognitive and social skills, not age and education. However, we might also include an interaction term between cognitive skills and age—this will add even more flexibility, and allow the skills to have different effect at different age. In contrary, interaction between skills and education may be hard to interpret as education is very closely related to skills anyway.

Another reason to include interaction effects is related to predictive modeling. In predictive modeling, we are typically not interested in interpretation, and hence the model complexity is not of major concern. However, when we introduce too much flexibility to the model, then we may run into overfitting (see [Section 5.3 Overfitting and Validation](#), page 245). So even in that case it is better not to introduce too many interaction effects. One can use standard model selection tools, such as forward selection, to assess whether the particular terms are worth including into the model.

see [Section 5.1 Predictive modeling](#), page 231

So in case of inferential modeling, one typically only includes interactions of the effect of interest, and a few other variables where we either expect to see a strong relationship, or which' relationship we are particularly interested. Here are a few potential examples:

- Effect of a vaccine: we may want to know how does effect of this drug depends on other medications the patients are taking. Such side effects may be critical in determining the cure. Hence we may include multiple interaction effect in the form

$$\begin{aligned} \cdots + & \beta_v \times \text{vaccine} + \beta_i \times \text{ibuprofen} + \beta_a \times \text{antidepressant} + \\ & + \beta_{vi} \times \text{vaccine} \times \text{ibuprofen} + \beta_{va} \times \text{vaccine} \times \text{antidepressant} + \dots \end{aligned}$$

We may include an interaction effect gender, but only if we have good reasons to believe that male and female bodies may react differently to the particular

vaccine (e.g. because it affects certain hormones). We probably do not want to include interactions with time if there is little reason to think that the effect will change from year-to-year.

- Effect of free-trade agreement on businesses. We may want to allow the effect to depend on the business sector as, e.g. firms in easily tradable manufacturing goods sector may face different opportunities than much less tradable service sector. We may also want to include interaction with time, as the new developments, spurred by the agreement, may take several years to materialize. However, gender of the CEO is probably irrelevant (unless this is our research question), so we do not want to include the corresponding interaction effect.

TBD: Examples from the literature

Finally, interaction effects are not the only way to introduce more flexibility to the model—instead of differences, captured by the interaction effect, one may estimate the levels for all groups separately. We do not discuss this approach in this book.

Interaction Effects and Intersectionality

Interaction effects is the linear regression way to assess *intersectionality*. The concept of intersectionality refers to the fact that many important experiences by individuals who belong to multiple groups cannot be described as only a sum of experiences by members of one and only one of those groups. The concept of intersectionality is typically used in context of discrimination. For instance, it may not be correct to describe the experience of a black women as a sum of experience of black (men) and (white) women. A workplace that treats black men equally to white men, and white woman equally to white men, may still treat black women in a unfair fashion: the trait of being black and the trait of being woman “intersect”.

We can transform this example into a regression model. Assume “treatment” here means wage the workers of the particular group receive (we can as well use other “treatments”, such as promotion, hiring, or harassment). We can write a linear regression model for wage as

$$w_i = \beta_0 + \beta_r \cdot race_i + \beta_s \cdot sex_i + \epsilon_i \quad (3.1.46)$$

where w_i is wage of individual i . If we model income in this way, the “treatment” (i.e. wage) is just sum of the treatment of the corresponding race parameter β_r and sex parameter β_s . There is no intersectionality. However, if we add the interaction effect

$$w_i = \beta_0 + \beta_r \cdot race_i + \beta_s \cdot sex_i + \beta_{rs} \cdot race_i \times sex_i + \epsilon_i \quad (3.1.47)$$

then the treatment is “made of” three components: treatment of the corresponding race group, the corresponding gender, and the “intesctional effect” β_{rs} . The members of particular race and gender may receive wage that differs from the sum of just race effect and just gender effect.

Note also that in linear regression we are always working with average values, e.g. looking for average salaries of whites and non-whites, and of men and women. Such aggregated approach has certain parallels with group prejudices. It is easy to look at,

say, β_s only, and claim that this number describes *all* women. This is not correct, the number describes the difference between male *average* and female *average* for *all* man and women *in this sample*. Also, it is important to understand that models (3.1.46) and (3.1.47) only address the size of male-female difference, not its cause.

Non-linear regression Linear regression assumes a linear relationship between y and x , and extended features, not necessarily between y and x .

TBD: What it is and why OLS is called linear

TBD: Polynomial regression

3.1.10 Theoretical considerations

Assumptions in OLS Models

Linear regression is not universally correct. In order for the coefficients to be interpretable, the standard errors and t -values to be correct, and predictions to be reliable we need a number of assumptions. Fortunately, as we defined the model in a rather rigorous way, we also have precise assumptions. This is fortunately, because we can now analyze each particular model, dataset, and process we are modeling, and analyze how likely it is that the assumptions are satisfied, and what happens if they are not.

Here we list just the most relevant assumptions we use in these notes:

1. **The model is correctly specified.** This means that the process we are analyzing is actually well described by a linear relationship, and not with something else, e.g. a curve. This is obviously important in order to talk about “correct” β -s, if the model is wrong to begin with then there is not such thing as correct β -s.
This is typically not a problem for noisy data (human behavior-related data tends to be noisy), and it is also good fit for many other type of relationships. But not for every relationship. If the underlying process is not well approximated with a linear model, then the regression estimates describe some sort of average relationship, which may or may not be good for our purpose.
2. **Mean-zero error term** $\mathbb{E}\epsilon = 0$. This is effectively normalization. It is almost always a harmless assumption, unless we are interested in the exact value of the intercept. But as we rarely are, so this assumption is rarely a problem.
3. **Normal errors** $\epsilon \sim N(0, \sigma^2)$. Normally distributed errors are needed for correct t -values. However, if the normality is not violated too much then we can still rely on the z -values in large samples through central limit theorem. But if deviations from normality are large, then the errors can be misleading even in large samples. Large deviations usually mean some sort of fat-tailed distributions, e.g. when analyzing a sample with many large outliers. Often a remedy is to take a log of the original variable.
4. **Independent error terms.** Error term of one observation must not influence the error of another observation. If it does, our standard errors may be very misleading. This is typically a problem in two types of data:

- (a) temporally or spatially related data, e.g. time series or geographic data. Stock price yesterday influences stock prices today, and house prices in a neighboring town influence house prices in this town.
- (b) clustered data, i.e. in stratified samples. For instance, drug use by high school students is not independent but affected by their peers, many of whom also attend the same schools.

These problems can be corrected through fairly straightforward methods, but you have to choose an correction method that is appropriate to the nature of the data.

5. **Explanatory variables and error term are independent:** $\mathbf{x} \perp\!\!\!\perp \epsilon$. (note: the previous assumption was about error terms of different observations, this is about x and ϵ of the same observation.) This is needed to get correct estimates of β . It is fairly harmless if we are only interested in association (i.e. non-causal relationship)—we just report that those who have more education also earn more. However, this is the crucial problem for causal inference, i.e. if we want to tell how much will someone's income improve if she were to take a college degree.

One should test the assumptions as needed when working with linear regression models. What and how do you test depends on the nature of the problem and data as some of the violations may be harmless.

Linear Regression: The matrix approach

In both theoretical and practical applications the regression models are often presented in matrix form. It may make the presentation more clear, but more importantly, storing and handling data in a matrix form tremendously simplifies solving the multiple regression model, both analytically and on computer. In matrix form there is no real difference between simple and multiple regression, neither in the way it is presented nor how it is solved by computer.

Let's look again at the linear regression model in vector form, (3.1.29):

$$y_i = \mathbf{x}_i^\top \cdot \boldsymbol{\beta} + \epsilon_i$$

or, more explicitly,

$$y_i = (1 \quad x_1^i \quad x_2^i \quad \dots \quad x_K^i) \cdot \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_K \end{pmatrix} + \epsilon_i \quad \text{for } i = 1 \dots N \quad (3.1.48)$$

(Remember that the first component of the \mathbf{x} is normally taken to be number 1 corresponding to the intercept β_0 , see Section 3.1.6 on page 154). Note we assume we have N observations indexed by i and $K + 1$ unknown parameters $\beta_0, \beta_1, \dots, \beta_K$. Alternatively we may say we have a single unknown parameter vector $\boldsymbol{\beta}$. $K + 1$ scalar parameters correspond to K explanatory variables and the constant.

Note that the first term of the matrix product in (3.1.48) is just a row vector of the data for the first observation. Hence we can take all the N rows corresponding to each observation in the data, and arrange these underneath each other like this:

$$\underbrace{\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}}_{N \times 1} = \underbrace{\begin{pmatrix} 1 & x_1^1 & x_2^1 & \dots & x_K^1 \\ 1 & x_1^2 & x_2^2 & \dots & x_K^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^N & x_2^N & \dots & x_K^N \end{pmatrix}}_{N \times (K+1)} \cdot \underbrace{\begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_K \end{pmatrix}}_{(K+1) \times 1} + \underbrace{\begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_N \end{pmatrix}}_{N \times 1}. \quad (3.1.49)$$

This is the matrix form. It can be written in a more compact manner as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \quad (3.1.50)$$

where \mathbf{y} is typically called *outcome vector*, \mathbf{X} is *design matrix*, $\boldsymbol{\beta}$ is the same parameter vector as in case of vector-form formula (3.1.29), and $\boldsymbol{\epsilon}$ is *disturbance vector*.

A few comments may be helpful:

- \mathbf{X} is the design matrix, the matrix that incorporates all data we use in the model, already converted into the numeric form and potentially also normalized, re-scaled, log-transformed and so on. If interaction effects are included in the model, they must have been incorporated in \mathbf{X} . So it is in a way a data matrix, but it is usually not a matrix of unmodified original data.
- we have the design matrix \mathbf{X} not transposed in the matrix notation, unlike in the vector form where \mathbf{x}_i^\top is transposed. This is because we arrange the data for one observation *horizontally* in the design matrix and we normally denote horizontal vectors with transposition sign.

Exercise 3.10: Matrix form

Show that (3.1.49) and (3.1.48) are equivalent.

Solution on page 425

In a similar fashion, we can stack all the prediction vectors in (3.1.30) as

$$\underbrace{\begin{pmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_N \end{pmatrix}}_{N \times 1} = \underbrace{\begin{pmatrix} 1 & x_1^1 & x_2^1 & \dots & x_K^1 \\ 1 & x_1^2 & x_2^2 & \dots & x_K^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^N & x_2^N & \dots & x_K^N \end{pmatrix}}_{N \times (K+1)} \cdot \underbrace{\begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_K \end{pmatrix}}_{(K+1) \times 1} \quad (3.1.51)$$

(3.1.30): $\hat{y}_i = \hat{\boldsymbol{\beta}}^\top \cdot \mathbf{x}_i$

and when writing this in the matrix form we get just

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}. \quad (3.1.52)$$

We can further compute the disturbance terms vector as

$$\mathbf{e} = \mathbf{y} - \mathbf{X}\boldsymbol{\beta} \quad (3.1.53)$$

and SSE as

SSE: sum of squared errors, see
Cheatsheet 3.3, page 148

Remember:
 $\mathbf{e}^T \cdot \mathbf{e} = e_1^2 + e_2^2 + \dots + e_K^2$,
 see Section 2.3.2 Vector
 multiplication as matrix product,
 page 112.

In order to use the matrix approach, we have to transform data into matrix form—create the *design matrix*. Design matrix is our final data in a numeric matrix form. It is “final” in the sense that it only contains the data that is actually used in the model, and it must be in a form that can be directly plugged into the formulas. In particular, this means all the non-numeric features must have been transformed to numeric ones; if feature normalization is desired this must be done and so forth. Not all models gain from construction of the design matrix, and there are models that contain more than one design matrix. But design matrix is the primary form of feeding data into linear regression and many other models.

Broadly, creating the design matrix proceeds through the following steps:

1. select only the relevant explanatory variables (i.e. no outcome variable) from your original data. Add information from different datasets if needed.
2. convert all these variables into a desired numeric form. This may include converting non-numeric variables into numeric (e.g. instead of $gender \in \{M,F\}$ we may use $female \in \{0, 1\}$). It also includes converting numbers into another form if needed, e.g. continuous age into age groups or income into log income.
3. You may have to add additional columns, in particular constant—a column of number ones—as this is not normally included in your data; interaction effects, and other engineered features.
4. stack the observations on top of each other as a matrix. The result looks much like a data frame but it must be a matrix in the mathematical sense of the word.

Example 3.13: Convert data to design matrix

Assume we have a dataset that looks like

name	age	position	salary
Liu Bei	28	manager	77,000
Sun Ren	23	employee	55,000
Thorgerd Egilsdóttir	26	employee	66,000
Freydís Eiríksdóttir	30	senior manager	123,000

We want to describe salary as a function of position and age using a linear model

$$\text{salary}_i = \beta_0 + \beta_a \cdot \text{age}_i + \beta_m \cdot \text{manager}_i + \epsilon_i. \quad (3.1.55)$$

Hence we want to estimate the effect of $K = 2$ variables (age and manager status) using $N = 4$ observations. We need to learn $K + 1 = 3$ unknown parameters.

Before moving any further we have to decide how to code the variables. While we can keep *age* as it is—it is already a numeric variable—we have to change *position* to a numeric form. We may just convert it into a dummy *manager* = 1(person is manager) and the modified data will now be

name	age	manager	salary
Liu Bei	28	1	77,000
Sun Ren	23	0	55,000
Thorgerd Egilsdóttir	26	0	66,000
Freydíð Eiríksdóttir	30	1	123,000

In vector form the same equation would look

$$\text{salary}_i = (1 \quad \text{age}_i \quad \text{manager}_i) \cdot \begin{pmatrix} \beta_0 \\ \beta_a \\ \beta_m \end{pmatrix} + \epsilon_i \quad (3.1.56)$$

where $i \in \{1, 2, 3, 4\}$.

The corresponding design matrix will look like

$$\mathbf{X} = \begin{bmatrix} 1 & 28 & 1 \\ 1 & 23 & 0 \\ 1 & 26 & 0 \\ 1 & 30 & 1 \end{bmatrix} \quad (3.1.57)$$

The first column, \mathbf{x}_0 , is the constant. This is something normally needed in a linear model and here we add it to the data. The second column is *age*, left unchanged. The third column is *manager*, equal to unity if the person is manager and zero otherwise. Note we have removed *name* as we don't use this in our model, and *salary* as this is our outcome variable:

$$\mathbf{y} = \begin{bmatrix} 77,000 \\ 55,000 \\ 66,000 \\ 123,000 \end{bmatrix}. \quad (3.1.58)$$

Now we can write the model (3.1.55) in matrix form as

$$\begin{bmatrix} 77,000 \\ 55,000 \\ 66,000 \\ 123,000 \end{bmatrix} = \begin{bmatrix} 1 & 28 & 1 \\ 1 & 23 & 0 \\ 1 & 26 & 0 \\ 1 & 30 & 1 \end{bmatrix} \cdot \begin{bmatrix} \beta_0 \\ \beta_a \\ \beta_m \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \end{bmatrix}. \quad (3.1.59)$$

This is the expression (3.1.49) written for these particular data. Now we can solve this for β .

Solving the Linear Regression Model

Prerequisites: Matrix product (Section 2.3.2), matrix inverse (Section 2.3.3), Linear regression in matrix form (Section 3.1.10). Be able to follow matrix calculus rules but not necessarily understand all of it. You know what is gradient and its notation (Section A.2.1 Gradient, page 407).

Linear regression is the only statistical model where an analytic solution exists. Here we demonstrate how to derive the solution. The proof is easy for those who know matrix calculus. But before we get to the general matrix form, let's find the solution of the scalar version of the linear regression model. This serves as a template that helps to understand the matrix version of the same problem.

We start with a model

$$y_i = \beta \cdot x_i + \epsilon_i, \quad (3.1.60)$$

i.e. a linear regression model that does not contain the intercept β_0 . This is because we want to derive the solution formula using only scalar algebra, and this is only possible if we have a single unknown parameter (here labeled as just β).

We start with the definition of linear regression (3.1.17) as the parameter value that minimizes SSE. As the model here does not include intercept β_0 , this is

$$(3.1.17): (\hat{\beta}_0, \hat{\beta}_1) = \arg \min_{(\beta_0, \beta_1)} \sum_{i=1}^N e_i^2$$

$$\hat{\beta} = \arg \min_{\beta} SSE(\beta) = \arg \min_{\beta} \sum_{i=1}^n (y_i - \beta \cdot x_i)^2. \quad (3.1.61)$$

As we typically do when computing a minimum of a function, we take a derivative of it w.r.t. β and set it to 0:

$$\frac{\partial}{\partial \beta} SSE(\beta) = -2 \sum_{i=1}^N (y_i - \beta \cdot x_i) \cdot x_i. \quad (3.1.62)$$

When we set this to 0 then we get

$$\begin{aligned} -2 \sum_{i=1}^N (y_i - \beta \cdot x_i) \cdot x_i &= 0 \quad \Rightarrow \\ \sum_{i=1}^N (y_i - \beta \cdot x_i) \cdot x_i &= 0 \quad \Rightarrow \\ \sum_{i=1}^N y_i \cdot x_i - \sum_{i=1}^N \beta \cdot x_i^2 &= 0 \quad \Rightarrow \\ \hat{\beta} &= \frac{1}{\sum_{i=1}^N x_i^2} \cdot \sum_{i=1}^N x_i \cdot y_i. \end{aligned} \quad (3.1.63)$$

(The last line is derived using the fact that β does not depend on i and hence it can be moved out of the sum sign.) So the optimal beta can be computed as a product of a) sum of $x_i \cdot y_i$ and b) inverse of sum of x_i^2 .¹⁵

¹⁵Advanced pocket calculators in 1980-s allowed to compute regression coefficient using this method. In one memory register they stored $\sum x_i^2$ and in another $\sum x_i \cdot y_i$, and hence it was possible to estimate a scalar regression model using just two memory positions.

Now it is time to replicate the exact same approach in matrix form. We start our solution by writing the least squares conditions (3.1.31) and (3.1.32) in matrix form using the *SSE* definition (3.1.54):

$$\begin{aligned}\hat{\beta} &= \arg \min_{\beta} SSE(\beta) = \\ &= \arg \min_{\beta} \mathbf{e}^T \cdot \mathbf{e} = \\ &= \arg \min_{\beta} (\hat{\mathbf{y}} - \mathbf{X}\beta)^T (\hat{\mathbf{y}} - \mathbf{X}\beta).\end{aligned}\tag{3.1.64}$$

Remember:
 $\mathbf{y}^T \cdot \mathbf{y} = y_1^2 + y_2^2 + \dots$
 See [Section 2.3.2 Vector multiplication as matrix product](#), page 112.

This is the matrix equivalent of (3.1.61).

Next, we proceed as we did above when finding a minimum of *SSE*: we take derivative of it with respect to β . Note that while *SSE* is a scalar, β is now a vector. Hence we need vector calculus rules to proceed. In particular, we have to keep track and preserve left- and right multiplication. In the scalar case (3.1.62) we just used the chain rule but let's here open the parenthesis instead of introducing the matrix calculus chain rule. Opening the parenthesis is straightforward, but because matrix multiplication is not commutative, we have to keep track of left and right multiplication:

$$\begin{aligned}(\hat{\mathbf{y}} - \mathbf{X}\beta)^T (\hat{\mathbf{y}} - \mathbf{X}\beta) &= \\ &= \mathbf{y}^T \mathbf{y} - \mathbf{y}^T \mathbf{X}\beta - \beta^T \mathbf{X}^T \mathbf{y} + \beta^T \mathbf{X}^T \mathbf{X}\beta = \\ &= \mathbf{y}^T \mathbf{y} - 2\mathbf{y}^T \mathbf{X}\beta - \beta^T \mathbf{X}^T \mathbf{X}\beta.\end{aligned}\tag{3.1.65}$$

The last line uses the fact that $\mathbf{y}^T \mathbf{X}\beta$ is a scalar and hence $\mathbf{y}^T \mathbf{X}\beta = \beta^T \mathbf{X}^T \mathbf{y}$.

In order to find the optimum, we again compute gradient of (3.1.65) and set it to zero. Just in case of matrices, the derivative is called gradient and instead of being equal to scalar (number) 0 it must equal to zero vector $\mathbf{0}$. As we are optimizing over a vector value β now, we have to use the rules of matrix calculus. As *SSE* is a scalar, the result will just be a column vector.

The gradient can be computed as follows:

$$\begin{aligned}\frac{\partial}{\partial \beta} SSE(\beta) &= \frac{\partial}{\partial \beta} [\mathbf{y}^T \mathbf{y} - 2\mathbf{y}^T \mathbf{X}\beta + \beta^T \mathbf{X}^T \mathbf{X}\beta] = \\ &= -2\mathbf{X}^T \mathbf{y} + (\mathbf{X}^T \mathbf{X} + (\mathbf{X}^T \mathbf{X})^T) \beta = \\ &= -2\mathbf{X}^T \mathbf{y} + 2\mathbf{X}^T \mathbf{X}\beta = -2\mathbf{X}^T (\mathbf{y} - \mathbf{X}\beta).\end{aligned}\tag{3.1.66}$$

This is the matrix analogue of (3.1.62). We used the following considerations to compute the individual components:

- the first term, $\mathbf{y}^T \mathbf{y}$, does not depend on β and hence we drop it.
- we use (A.2.4) for the term $\frac{\partial}{\partial \beta} (\mathbf{y}^T \mathbf{X}\beta)$;

- $\frac{\partial}{\partial \beta} (\beta^\top X^\top X \beta)$ is computed using (A.2.11) with $X^\top X$ in place of A .

Using the optimality condition $\frac{\partial}{\partial \beta} SSE(\beta) = \mathbf{0}$ we get

$$-2X^\top (X\beta - y) = \mathbf{0} \quad \text{or} \quad (3.1.67)$$

$$X^\top X \beta = X^\top y \quad (3.1.68)$$

By left-multiplying both sides by $(X^\top X)^{-1}$, we get

$$\hat{\beta} = (X^\top X)^{-1} X^\top y. \quad (3.1.69)$$

$$(3.1.63): \hat{\beta}_1 = \frac{1}{\sum_{i=1}^N x_i^2} \cdot \sum_{i=1}^N x_i \cdot y_i.$$

This is the matrix analogue of (3.1.63). Note that the matrix formula uses matrix products instead of sums of products, and that the term $\frac{1}{\sum_{i=1}^N x_i^2}$ has been replaced by its matrix analogue $(X^\top X)^{-1}$.

So we derived a simple analytic solution to the linear regression problem. It is the only statistical model where we do not have to rely on non-linear optimization but can just plug the numbers (matrices) into a formula and get the result right away.¹⁶ The solution involves three matrix multiplications (cheap operations) and one matrix inverse (expensive operation). In practice, the analytic solution is the preferred way only if the dimension of $X^\top X$ is small, i.e. we have no more than thousands of variables. If we have more variables then, Gradient Descent (see Section 8.2 Gradient Ascent, page 341) may be faster as that approach only requires repeated computation of gradient (3.1.66) but not the expensive inverse $(X^\top X)^{-1}$.

Note that in order for the solution to exist, the inverse $(X^\top X)^{-1}$ must exist. In particular this means that X must be full rank.

¹⁶Although the formula is very simple, the actual computations may be quite complicated and imprecise. In particular, inverting the $K \times K$ matrix $X^\top \cdot X$ may be time-consuming and imprecise if the number of variables K is large. However, normally we leave these tasks for dedicated libraries.

3.2 Logistic Regression

The previous section introduced linear regression, one of the central workhorses for inferential analysis. The main requirement for the linear regression is that the outcome variable, y , is continuous, or at least close to continuous. This was the case with both galaxies and income.

However, for a large class of problems, this is not the true. For instance, the question whether someone survived the shipwreck, whether a tweet will be retweeted, and whether an oil drill gets stuck in the drillhole cannot be described with continuous outcome. The passenger either survived or not, and a tweet was retweeted or not. Even if we describe these outcomes with numbers (e.g. survival as “1” and death as “0”), the result is not a continuous problem. We need different tools for this type of tasks.

3.2.1 What Is Logistic Regression And What Is It Good For?

Consider policymakers during economically challenging times. Unemployment is large and work is nowhere to be found. Government is spending lot of money on benefits and the voices that are concerned about the effect on workers’ motivation and governments coffers are growing in strength. But actually—it is not just that work is nowhere to be found. There are plenty of jobs available. But unfortunately those jobs require different skills, skills that most unemployed do not possess. So government comes up with idea to upskill the unemployed instead of just paying benefits. It announces a subsidized training program where all unemployed are welcome to participate. But who will actually end up joining this program?

Table 3.8: An example of “Treatment” data. treat is treatment, “T” mean the person participated and “F” means they did not participate in a training program. re denotes real income (in USD) and u unemployment in years 1974-1978.

treat	age	educ	ethn	married	re74	re75	re78	u74	u75
F	50	17	other	T	33308	30436	25860	F	F
F	26	12	other	T	37226	25960	26599	F	F
T	31	9	hispanic	F	0	0	26818	T	T
F	30	11	hispanic	T	29389	26855	36943	F	F
F	48	11	other	T	22179	19694	16255	F	F

Let us analyze this question using “Treatment” dataset (R package *Ecdat*). The dataset describes various labor market-relevant variables, such as education, income and unemployment, but for now, let’s focus only on age (see Table 3.8 for an example). Are the participants more likely young or old? Figure 3.14 displays the relationship between participation and age. The graph looks a bit weird, this is because there are only two possible values for participation—either 1 (participated) or 0 (did not participate), and only integer values for age. In order to avoid too much overlap, we have knocked the points a bit off from their true location so we get a small point cloud for each age and participation combination. The figure reveals that most

participants ($\text{Participation} = 1$) are young. It is hard to see the age distribution of non-participants—the black dots overlap quite a bit, but it seems the most common age range in this dataset is 20-30.

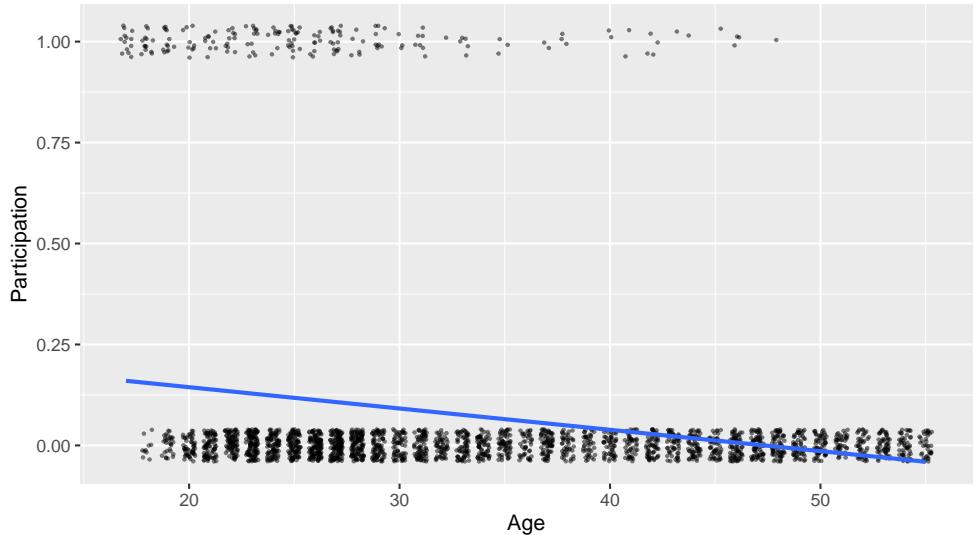


Figure 3.14: Participation as a function of age. Treatment data. In order to avoid overlap on the plot, the points are moved slightly off from their true location. The blue line is the linear trend line.

The model also displays trend line (regression line, blue). But unlike in case of Hubble diagram (Figure 3.8) where the dots were aligned with the regression line, more-or-less, the line here seems to miss the dots almost completely. Why does linear regression behave so miserably?

The main culprit is the fact that the outcome is a binary variable. Treatment status can only be “0” or “1” and nothing in between. But a line cannot touch just one or another of these values, a line also connects everything in between. So we necessarily see values like “0.1” and “0.5”, numbers that do not make any sense in terms of treatment. The way to overcome this problem is to interpret the outcome not as the treatment value, but *probability* that the individual is treated. So a value “0.5” would mean fifty-fifty probability that someone is treated while “0.99” would mean that the person almost certainly participated. Taking this view, the trend line suggests that the probability for a 20-year old to participate is approximately 15%, but for a 40-year old the probability is more like 5%.

In fact, this approach is widely used and a linear regression model that describes probability is called *linear probability model* (LPM, see Section 3.3). But LPM-s also have another problem. You can see that the line falls below zero around age 46. Should we interpret it as the 50-year olds have a negative probability to participate? That is obviously nonsense. In a similar fashion, the line will exceed probability 1

somewhere (the age where this happens will be negative in this case, so it is not a problem here). We can obviously hack the model in a way that we set probability to zero if the predicted probability is negative. But what should we do with the 48-year old participant then (the oldest participant in Figure 3.14 is 48 years old)? If the participation probability at that age is zero then we should not see even a single participant in that age category. But we see a few, so we need to set the probability not to zero but to a small positive number... If you are still with me then you probably agree that making linear regression to work with probabilities needs a lot of hacking, and the model is not a nice and intuitive any more. So we need another model that a) models probability $\Pr(\text{outcome} = 1)$, not the value of outcome; and b) ensures that the probability is in $[0,1]$ interval.

There is a wide range of applications with binary outcomes where such a model is handy. For instance, if someone attends college, gets a job, defaults a loan, that an email is spam, or that an image depicts a cat are all binary-outcome questions. And linear regression is not well suited to answer such questions.

Logistic regression (aka *logit*) is the most popular model designed for exactly this type of tasks, the tasks with *binary outcome*. “Binary outcome” means these questions can only have two answers—“0” or “1”, “true” or “false”, “cat” or “not a cat”. This makes it distinct from linear regression that is designed to measure *continuous outcomes*, i.e. outcomes that can take all sorts of *numeric* values. Whether the outcome is numeric or something else plays almost no role for logistic regression, we can always transform two possible outcomes into “0” and “1”. This is what we did with treated and non-treated above.

Exercise 3.11: Linear or logistic regression?

Would you use logistic or linear regression to analyze these questions:

1. How long will cancer patients survive after treatment?
2. How good is students’ GPA?
3. Who gets admitted to an elite school?
4. Will the tweet be retweeted?
5. How many people will read the tweet?
6. Who survived a shipwreck?

Solution on page 426

Mathematically, it is essentially a transformation of linear regression model that is interpreted as probability. The transformation is done using *logistic function* (aka *sigmoid function*)

$$\Lambda(\eta) = \frac{e^\eta}{e^\eta + 1} = \frac{1}{1 + e^{-\eta}}. \quad (3.2.1)$$

Here one can understand η as the “output” of linear regression, and $\Lambda(\eta)$ is *logistic transformation* of η , (see Figure 3.15). It has two properties that make it a perfect fit for probability modeling:

- It is monotonically increasing, i.e. a larger η always corresponds to a larger $\Lambda(\eta)$. This makes it a good choice to model the fact that we typically see smooth transitions in data, such as older workers are less likely to participate as in Figure 3.14.

Logistic transformation and *log-transformation* (see [Section 3.1.8 Feature Transformation](#), page 160) are different concepts!

- Its values are strictly in the interval $(0,1)$. So these are directly interpretable as probabilities and we do not need any further hacks.

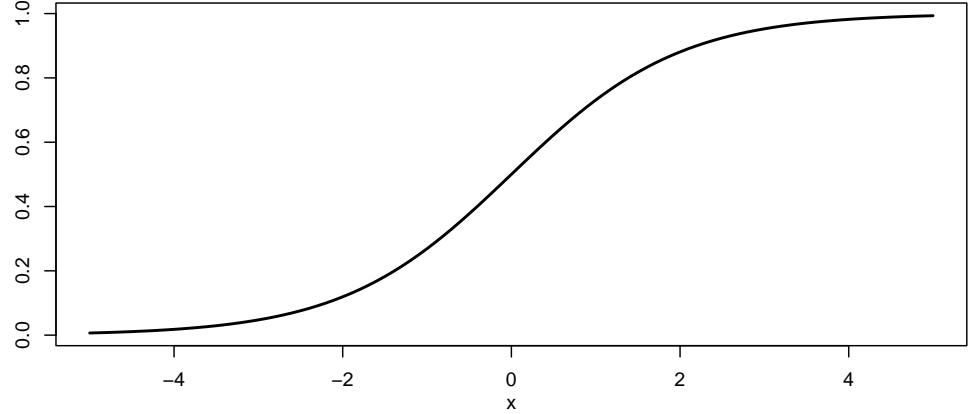


Figure 3.15: Logistic function (logistic transformation). While the input variable η can have any value in $(-\infty, \infty)$, $\Lambda(\eta)$ is limited to interval $(0,1)$. This is what makes it suitable for modeling probability.

$\boldsymbol{\beta}^T \cdot \mathbf{x} = \beta_0 + \beta_1 \cdot x_1 + \dots + \beta_K \cdot x_K$
See Section 2.3.2 Vector multiplication as matrix product, page 112 and Section 3.1.6 Formal Definition of Multiple Regression, page 154.

More specifically, η in the logistic regression formula (3.2.1) is not called “linear regression output” but *link*, *linear predictor* or *linear index*. But it is calculated in exactly the same way as the predicted value for linear regression: $\eta_i = \beta_0 + \beta_1 \cdot x_i$ in case of a single explanatory variable, or in vectorized form as $\theta_i = \boldsymbol{\beta}^T \cdot \mathbf{x}_i$ in case of multiple explanatory variables. So we can write the logistic probability in a slightly longer form as

$$\Lambda(\mathbf{x}) = \frac{1}{1 + e^{-\boldsymbol{\beta}^T \cdot \mathbf{x}}}. \quad (3.2.2)$$

This is the expression for the probability that the outcome is “1” for given values of \mathbf{x} . For completeness, we state it once again:

Logistic regression model

$$\Pr(Y = 1 | \mathbf{x}) = \frac{1}{1 + e^{-\boldsymbol{\beta}^T \cdot \mathbf{x}}}. \quad (3.2.3)$$

This must be understood as the rule to compute the probability that the outcome $Y = 1$ if the value of the explanatory variable is \mathbf{x} . Exactly as in case of linear regression, we have to find such parameter vector $\boldsymbol{\beta}$ that gives the “best” fit with data.

(3.1.1): $y_i = \beta_0 + x_i \cdot \beta_1 + \epsilon_i$

Note another important difference between logistic and linear regression models. Namely, the logistic regression (3.2.3) does not contain an error term while the linear regression (3.1.1) does. This is because in case of linear regression we are modeling outcome value, and we need an error term to take into account the fact that the

modeled and actual values almost always differ. But in case of logistic regression, we model probability, which means that the event may happen or not happen. Probability describes a process that is already stochastic, so we do not need an additional error term.

Let us demonstrate these calculations using treatment data above (Figure 3.14). But before we can even calculate anything, we have to specify which event are we modeling—are we modeling probability of treatment or non-treatment? In this case it seems more natural to model probability of treatment, $\Pr(T = 1)$ instead of $\Pr(T = 0)$. It is often useful to model probability of the “rare” events, or probability of “interventions”. Treatment checks both boxes here as only $\sim 7\%$ of cases in data are treated, and treatment is more “active” process than non-treatment. But both approaches are equally valid, one has to make a decision and stick with that.

As we look at how the treatment probability depends on age, we have a single explanatory variable x , namely age, and we can write

$$\begin{aligned}\eta_i &= \beta_0 + \beta_1 \cdot \text{age}_i \\ \Pr(T_i = 1) &= \frac{1}{1 + e^{-\eta_i}} = \frac{1}{1 + e^{-\beta_0 - \beta_1 \cdot \text{age}_i}}.\end{aligned}\tag{3.2.4}$$

In order to actually calculate the probability of treatment, we have to pick β_0 and β_1 values.

For instance, let’s just guess that the values 0 and -0.1 for β_0 and β_1 respectively, and compute the participation probability for a 30-year old person. We have

$$\Pr(T = 1 | \text{age} = 30) = \frac{1}{1 + e^{-\beta_0 - \beta_1 \cdot \text{age}}} = \frac{1}{1 + e^{-0 - 0.1 \cdot 30}} = \frac{1}{1 + e^3} \approx 0.047.\tag{3.2.5}$$

So our model, given the choice of parameters, predicts that roughly 5% of 30-year olds will participate. The actual number in data is 0.043. Figure 3.16 shows how the modeled participation probability depends on age for three different sets of parameters. The figure suggests that out of the three combinations displayed there, the one we calculated above $(0, -0.1)$ (blue curve) is close to actual data. The red curve $(0, 0.05)$ gets age dependency completely wrong, and the green curve $(0, -0.05)$ suggests participation probabilities that are too high. But it is hard to select good combination of parameters just by visual inspection even for this simple case with a single explanatory variable only. The best set of parameters for logistic regression is usually computed using Maximum Likelihood method (see [Section 3.2.3 Solving logistic regression model](#), page 190). The corresponding probability is shown by the dashed black curve.

When we compute the best possible coefficients (the dashed black line in Figure 3.16), we get the following results:

	Estimate	Std. Error	z value	$\Pr(> z)$
Intercept	1.0343	0.3300	3.13	0.0017
age	-0.1229	0.0122	-10.05	0.0000

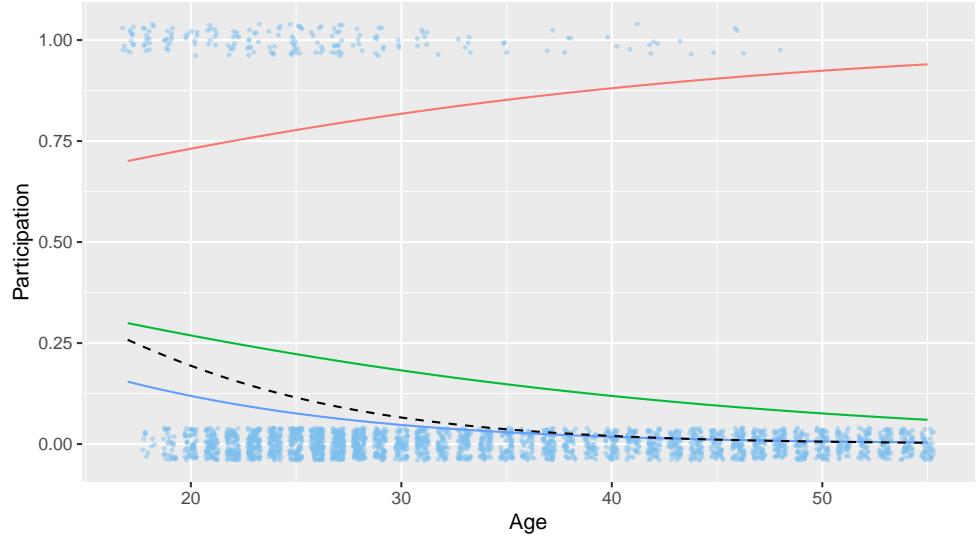


Figure 3.16: The same participation data as in Figure 3.14. The lines depict parameter combinations $(\beta_0, \beta_1) = (0, 0.05)$ (red), $(0, -0.05)$ (green) and $(0, -0.1)$ (blue), the black dashed line is the Maximum Likelihood estimate $(1.034, -0.123)$. The red line clearly misses the data, green line captures the pattern of falling participation in age, while the blue line seems to fit well and also capture the fact that participation rate is very low for over 40 year olds.

The results table, as provided by common software packages, looks rather similar to the linear regression table (see Table 3.2). We see similar columns for estimates, standard error, z -value and p -value (obviously, different software packages provide somewhat different output). The meaning of the parameters is rather similar to that of linear regression with two main differences: first, the interpretation of logistic coefficients is quite different from that of the linear regression coefficients, so it is explained in the next section ([Section 3.2.2 Interpreting logistic regression results](#), page 185).

See [Section 3.1.3 Interpreting the regression table](#), page 137 above for how to interpret linear regression table values where $df = \infty$. See [Section 1.5.2](#).

Second, instead of t -values, logistic regression estimates are typically reported with z -values. From practical standpoint, these are fairly similar. Just instead of critical t value, we are concerned with critical z -values (for 5%-significance level it is 1.96, see [Table 1.8](#)). In a similar fashion, z -value measures distance between the estimated coefficient and H_0 value, and in exactly the same way, the software normally assumes $H_0 : \beta = 0$. The difference between z and t values is primarily in the assumptions. In case of linear regression, for the t values to be correct, the error term ϵ must be normally distributed. In logistic regression, for z values to be correct, the sample size must be large.

Exercise 3.12: Which values are statistically significant?

Imagine you estimate a logistic regression model in the form

$$\Pr(\text{finds job}_i) = \Lambda(\beta_0 + \beta_1 \cdot \text{education}_i + \beta_2 \cdot \text{big city}_i + \beta_3 \cdot \text{age}_i)$$

You'll get the following results:

	Coef	Std.err	<i>z</i>
Education	0.120	0.03	4.00
Big city	0.150	0.10	1.50
Age	0.002	0.10	0.02

Which coefficients are statistically significant (at 5% level)?

Hint: consider *z*-value table.

Solution at page [426](#)

3.2.2 Interpreting logistic regression results

Prerequisites: [Section 3.1.3 Interpretation](#), page 134, [Section 3.1.6 Interpreting multiple regression effects](#), page 150.

Logistic regression is in many ways similar to linear regression, including by being an interpretable model. Unfortunately, interpretation of logistic regression results is more complicated than in case of linear regression. There are two related reasons for that. First, logistic regression is a non-linear model, and hence the slope depends on the values of the explanatory variables (see Figure 3.17). And second, because the slope depends on the explanatory variables, we cannot just interpret the parameters β_0 and β_1 directly in terms of probability.

There are two popular ways to overcome these limitations: *marginal effects* and *odds ratios*.

Marginal effects

Marginal effect (ME) is slope of the logistic function on the figure where we have probability on the *y*-axis and the explanatory variable *x* (not the link function η) on the *x*-axis. Marginal effect answers the same question as slope β_1 in case of linear regression: *How much more likely is the outcome if *x* is larger by one unit*. In the example above, ME will answer the question *How much more likely is that someone will participate given she is one year older*. In case of multiple logistic regression we should also add the phrase *given all other explanatory variables are the same*. So, in this sense marginal effects are very similar to linear regression coefficients. However, there are two major differences, both of these related to the fact that we now have a non-linear model:

- Marginal effects must be calculated from β -s, and the calculation is not obvious. Fortunately, modern software will do it with a simple function call.

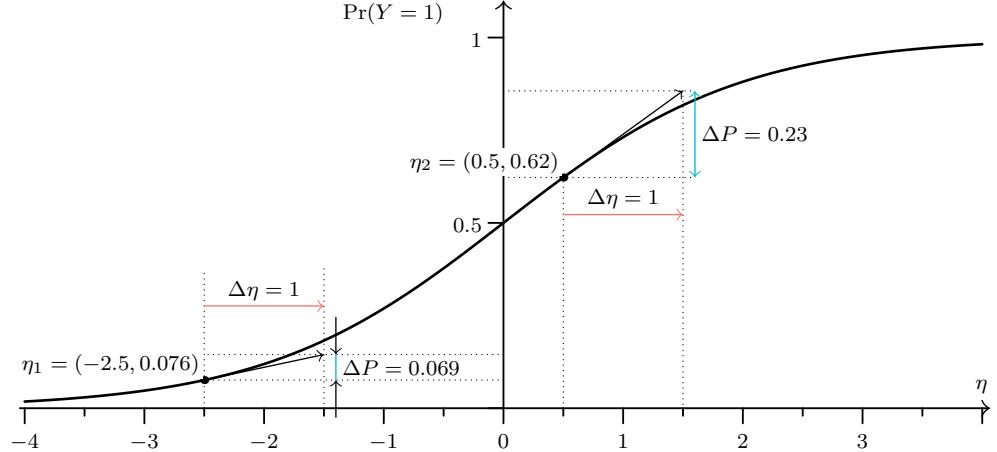


Figure 3.17: Interpretation of logistic regression results. How much larger is $\Pr(Y = 1)$ differ when the link value η is larger by one unit, depends on the η value. The probability grows at rate 0.069 per unit of η at $\eta_1 = -2.5$, and at rate 0.23 per unit of η at $\eta_2 = 0.5$.

- Marginal effects depend on the values of x . So different observations with different x values will have different marginal effects. Hence we must always decide what kind of cases we are interested in. The effect differs case-by-case.

As marginal effect is just slope, we can compute it by taking the derivative of the logistic probability. For instance, in order to compute the marginal effect of age in the example above, we take derivative of the treatment probability (3.2.4):

$$\frac{\partial}{\partial \text{age}} \frac{1}{1 + e^{-\eta}} = -\frac{e^{-\eta}}{(1 + e^{-\eta})^2} \beta_1 \quad (3.2.6)$$

where $\eta = \beta_0 + \beta_1 \cdot \text{age}$. This is straightforward to compute, but normally we let statistical software do the work.

Figure 3.17 demonstrates the meaning of marginal effects. The thick black curve is the logistic curve as a function of the link η . Its slope differs at different points, here we have marked $\eta_1 = -2.5$ where the slope is 0.069, and $\eta = 0.5$ where the slope is 0.23. These numbers—0.069 and 0.23—are the *marginal effects of η* . But we are interested in *marginal effect of age* instead—how much more likely it is to participate for those who are one year older. Now we have to take into account that η depends on x as $\eta = \beta_0 + \beta_1 x$. Hence one unit larger x means β_1 units larger η and hence the marginal effect of x is just the marginal effect of η , multiplied by β_1 .

As marginal effects depend on x , we cannot just provide marginal effects that apply universally. Obviously, in case η is very small or very large, the effect will also be very small, while the η values near 0 are associated with larger effects. Typically, one of these three options is reported: a) marginal effect at the mean x value; b) compute all individual marginal effects and takes the average; or c) marginal effect for certain specific interesting cases. Example of marginal effect output is in the table below:

factor	AME	SE	z	p	lower	upper
age	-0.0075	0.0008	-9.1811	0.0000	-0.0090	-0.0059

The basics of this table are quite similar to that of the logistic coefficients table above. *AME* is average marginal effect, software computes the marginal effects for every individual in these data and takes the average. *SE* stands for the standard error of *AME*, *z* and *p* are the corresponding *z* and *p* values, and the two last columns are CI for *AME*.

AME is directly interpretable in a similar fashion like the β -s in linear regression. The number -0.0075 means that:

One year older individuals are 0.0075 less likely to participate in average.

This can be phrased somewhat better using percentage points:

One year older individuals are 0.75 percentage points less likely to participate in average.

Percentage point: difference in values that are given in percentages (see [Section 1.1.1 Ratio measures](#), page 4).

And as explained above, if we are working with multiple logistic regression, we should add “... *if all other explanatory variables are the same*” to the sentence above.

Odds ratios

Another popular way to interpret logistic regression results is by using *odds ratios*. Odds ratio is simply the ratio of the one group to the other, in the example above it will be the probability of participation over the probability of non-participation,

$$r = \frac{\Pr(Y = 1|\boldsymbol{x})}{\Pr(Y = 0|\boldsymbol{x})} \quad (3.2.7)$$

If we compute the probabilities using the sample averages, we get

$$r = \frac{N_{y=1}}{N_{y=0}} = \frac{185}{2490} = 0.074. \quad (3.2.8)$$

Odds ratios are popular to describe the probabilities of certain kind of events, such winning chances in certain horse races. But unfortunately, these ratios are not used widely, and hence people tend not to understand the values well.

It turns out that logit coefficients are directly interpretable as effects on logarithms of odds ratios, *log-odds*. From (3.2.3) we can express $e^{\boldsymbol{\beta}^\top \cdot \boldsymbol{x}_i}$ as

$$e^{\boldsymbol{\beta}^\top \cdot \boldsymbol{x}_i} = \frac{\Pr(Y = 1|\boldsymbol{x})}{1 - \Pr(Y = 1|\boldsymbol{x})} = \frac{\Pr(Y = 1|\boldsymbol{x})}{\Pr(Y = 0|\boldsymbol{x})}. \quad (3.2.9)$$

This is exactly odds ratio.

Exercise 3.13: Prove (3.2.9)

Use the logistic regression definition (3.2.3) to derive (3.2.9).

We can use this idea to find the effect on the odds ratio. Consider two vectors of explanatory variables, \mathbf{x}_1 and \mathbf{x}_2 . The latter is otherwise equal to the former, except one of \mathbf{x}_2 components, x_{2i} , is larger by one unit compared to x_{1i} . So while $\mathbf{x}_1 = (1, x_{11}, x_{12}, \dots, x_{1i}, \dots, x_{1K})$, the $\mathbf{x}_2 = (1, x_{11}, x_{12}, \dots, (x_{1i} + 1), \dots, x_{1K})$. Hence the odds ratio for case \mathbf{x}_2 is

$$\begin{aligned}\frac{\Pr(Y = 1|\mathbf{x}_2)}{\Pr(Y = 0|\mathbf{x}_2)} &= e^{\beta_0 + \beta_1 x_{11} + \beta_2 x_{12} + \dots + \beta_i (x_{1i} + 1) + \dots + \beta_K x_{1K}} = \\ &= e^{\beta_0 + \beta_1 x_{11} + \beta_2 x_{12} + \dots + \beta_i x_{1i} + \dots + \beta_K x_{1K}} e^{\beta_i} = \frac{\Pr(Y = 1|\mathbf{x}_1)}{\Pr(Y = 0|\mathbf{x}_1)} e^{\beta_i}. \quad (3.2.10)\end{aligned}$$

So e^β describes the multiplicative effect on odds ratio: if x is larger by one unit, the odds ratio is larger by e^β units.

For instance, the age effect in the model (3.2.4) above is -0.123. Hence the odds ratio effect is

$$e^{-0.123} = 0.884.$$

This means that odds of one year older individuals is 88 times that of younger individuals. Or alternatively, one year older individuals have 12% lower odds to participate. Note that unlike in case of marginal effects, this number—12%—is measured in percentages (of the baseline rate), not percentage points.

Odds ratios have two advantages over marginal effects: they are easier to compute (you only need to take exponent) and they are stable—odds ratios are constant and independent of personal characteristics. This contrasts to marginal effects that depend on the other parameters. But as odds ratios are harder to understand, and as nowadays the software to compute marginal effects is easily available, the odds ratios have become less popular.

Cheatsheet 3.6: Linear regression vs logistic regression

Here we list the main differences between linear versus logistic regression:

Model Linear regression models the outcome value:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

Logistic regression models the outcome probability:

$$\Pr(y_i = 1) = \Lambda(\beta_0 + \beta_1 x_i)$$

Here x_i is the predictor, y_i is outcome, and β -s are unknown parameters to be estimated; $\Lambda(x) = 1/(1 + \exp(-x))$ is the *logistic function* (sigmoid function).

Usage Linear regression can be used where the outcome y is *continuous variable* (e.g. height, income, duration).

Logistic regression can be used where outcome is *binary variable* (e.g. found a job, survived shipwreck, earthquake occurs).

Interpretation Linear regression: β_1 means *one unit larger x is associated with β_1 unit larger y (if other predictors the same)*.

Logistic regression: cannot easily interpret β_1 as this is a non-linear model. Need to compute marginal effects (or odds ratios instead).

Prediction Linear regression: predict outcome value

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

Logistic regression: predict outcome probability

$$\widehat{\Pr}(y_i = 1) = \Lambda(\hat{\beta}_0 + \hat{\beta}_1 x_i)$$

Predict outcome category (classification/categorization):

$$\hat{y}_i = \begin{cases} 1 & \text{if } \widehat{\Pr}(y_i = 1) > 0.5 \\ 0 & \text{if } \widehat{\Pr}(y_i = 1) < 0.5 \end{cases}$$

3.2.3 Solving logistic regression model

TBD: Talk about loglik function

3.3 Linear probability model

Chapter 4

Causality

Humans want to manipulate their environment. We want to avoid going hungry, we want to cure illness and we want to achieve a successful career. And we know well what to do in order to achieve these goals—just eat, take a drug, and maybe go and study economics. But how did we learn that eating and studying help to achieve these goals? The relationship between eating and hunger is probably implanted in our brains—after all, only those animals that figured it out were able to survive and breed. But knowledge about drugs and illnesses, or economics and career, is based on data, experiments and theoretical considerations. This chapter discusses the ways one can obtain such knowledge from data. As we will see, the knowledge about how two variables (e.g. your college major and future success) are associated is not enough to tell how *manipulating one of these* (e.g. choosing to major in economics) is associated with changes in the other (e.g. making a more successful career). We need to know more to answer this question.

Contents

4.1	Introduction	192
4.2	What is cause?	193
4.2.1	Sufficiency and Necessity	193
4.2.2	Measuring the Amount of Cause and Effect	193
4.3	Causality with data: three explanations	194
4.4	Strategies for Causal Inference	199
4.4.1	We Know Which Model is Feasible	199
4.4.2	Randomized Controlled Trials	200
4.4.3	Natural Experiments	202
4.4.4	Case-Control Study	204
4.4.5	Controlling for Confounding Factors	205
4.4.6	Explicit Modeling of Selection Process	205
4.5	Causal inference in linear regression framework	205
4.5.1	Counterfactual and Identifying Assumption	206
4.5.2	More about identifying assumptions: mean independence	209
4.5.3	A Few Popular Estimators	213

4.6	Cognitive Illusions in Causal Inference	228
4.7	Causality and complex social problems	229
4.7.1	Effect of bike helmet laws	229

4.1 Introduction

Causal inference means using data to gather just this kind of knowledge, knowledge that helps us to manipulate the world in our liking. We use such knowledge extensively in our everyday lives, both instinctively (you pull your hand away from hot pan when you get burned) and deliberately (you flip the switch to turn on lights). Such act can only be successful if we know that touching hot surfaces causes burns, and flipping the switch causes the lamp to turn on. In these two example we know and understand the process very well, or at least well enough to successfully employ it. But there are many important situations where we do not understand the results of our acts well, or where our understanding is just wrong.

It also turns out that causality is more complex than just flipping a switch. Sometimes the cause is just a binary on-off event, such as a switch, or dropping a glass so it breaks. Other times there the cause can come in different quantity, *dose*, for example when we are interested in the amount of training airline pilots receive. Sometimes we want to measure the size of the outcome, for instance when we are interested in the effect of college education on salary. In other cases the size of outcome carries little meaning (in case of breaking a glass it matters little how many pieces it breaks into) but we may be interested in the probability of the outcome—how likely it is that the glass breaks in the first place, and how does it depend on the dose of the cause. Sometimes the cause is not a single event but multiple events linked in chains. For instance, in case of plane crash this may include weak training of pilots, combined with management’s reluctance to address technical problems, and a bad weather on landing.

In this section we are mainly concerned with dose of a single cause embedded in such chains. For instance, *how much less likely are airplane crashes if pilots have $x\%$ more training?* In this example, we are not just interested in “pilot training” but in certain doses of pilot training. The question itself is often clear and well defined and can in principle be answered from data. However, as it turns out, the data with necessary structure is extremely rare. The best answers come from randomized experiments, but these are often expensive, unethical, or just impossible. Airplane accidents are a good example of important causal questions where we cannot conduct experiments.

Because suitable experimental data is hard to find, we have to resort on other sources of information. Unfortunately, this leads to both more complex econometric methods, and less reliable answers.

4.2 What is cause?

People normally have a pretty good intuitive understanding of what is cause. Two events A and B are causally related if the latter at least partly depends on the former. The dependency can be understood that if you remove the cause A , then B will not occur, or even if it occurs, it will be somewhat different. The *cause A* also has to precede the *effect B* in time, at least in the physical world. But the intuitive understanding is in many ways limited and does not cover several different ways how one event may influence another.

4.2.1 Sufficiency and Necessity

The intuitive concept of causality applies well if A is both necessary and sufficient for B to occur.

But if this is not the case then the concept of cause gets murkier. For instance, in 2000 the supersonic airliner Concorde crashed because another airplane (Continental DC-10) dropped a large piece of metal on the runway. The accelerating Concorde ran over the debris a few minutes later. This caused its tire to rupture, a piece of rubber hit the wing and broke the fuel tank there. What was the cause of the accident? Improper maintenance of DC-10 or a dangerous design of Concorde? Planes should not leave debris on runway, no doubt. But airliners should also be able survive tire ruptures (this was not the first time Concorde's tire broke at high speed). Both of these factors were necessary but individually they were not sufficient for the crash. Such factors are often referred to as *contributing causes*. In a similar fashion, when counting the death from a certain disease, how should one count a case where someone had more than one medical condition, including the disease of interest? For instance, would someone who dies of lung malfunctioning while having both heart attack and acute COVID-19, count as COVID-19 death? The person might have survived a single condition, either just heart attack or just COVID-19.

Alternatively, an event may be sufficient but not necessary. If two kids are throwing rocks to a window almost instantaneously, the first rock will break the glass and the second one will just go through the already broken pane. Will the second rock still be the cause? After all, when we say that only the first rock is the cause, then when we remove the cause, the outcome will still be the same—a broken window. So should we blame the second child as much as we blame the first one?

4.2.2 Measuring the Amount of Cause and Effect

In many cases the cause is just a binary “it is there/it is not there” quantity. Dropping an unboiled egg on floor causes it to break. We can say that one unit of cause (dropping the egg) causes one unit of outcome (smashed egg). Here the concept of quantity and quantitative effect is rather useless. We can just say that “egg will break if you drop it”, simple words that everyone will understand.

In other cases the quantity carries an important meaning. For instance, a vaccine is made of a number of different ingredients, the most important of which is called the “active ingredient”. This is the substance that actually helps to fight the infection. For instance, the Pfizer coronavirus vaccine contains 30 µg of viral RNA, the substance

that actually makes body to build up resistance. As this mount would be a tiny grain, barely visible to human eye, the vaccines normally contain a lot of “fillers”, such as salts, sugar, and water. However, from the medical viewpoint the important question is *how much less likely it will be to contract the disease if one takes x µg of the active ingredient?* This question involves two quantitites: the quantity of the active ingredient, *dose* of treatment, measured in µg; and quantity of the effect, here measured in terms of probability difference. In this case we expect to see a negative relationship: more micrograms of the active ingredient will make it less likely to contract the disease. Unfortunately, the language is now more complicated than before.

Sometimes we are only interested in quantity of the outcome but not in the dose of the cause. E.g. we may ask *how much more (or less) likely are hurricanes now because of global warming?* In this case we take the dose of global warming as given and only ask how it affects the probability of hurricanes. For instance, a valid answer might be “10 percent”, i.e. the hurricanes are 10% more likely now than in the past due to global warming.

Such questions get harder to understand if we add the uncertainty measures because we rarely know the exact answers. Instead of a simple “10%”, one may now give the confidence intervals: *we are 95% certain that global warming has increased the probability of hurricanes between 5 and 15%*. Note that this claim contains two unrelated probability measures: confidence of our results (95%), and the probability of hurricanes (growth between 5 and 15%). Such double use of probability needs some probability literacy, and even for the literate it needs a second or two to understand the sentence. This is the language of science, this is very much the only type of results science can produce, but undoubtedly, complexity of claims like this has contributed to the wide-spread skepticism of global warming and scientific results in general. (See more in [Section 1.6.1 Statistical language is heavy](#), page 83.)

4.3 Causality with data: three explanations

Let us now leave (fortunately) rare hurricanes and air disasters aside and return to situations where we can collect “data”, i.e. we observe a multitude of similar cases where we measure various factors. For example, assume we collect data about patients’ vaccination status (whether they got flu shot) and health (whether they got flu) in a large hospital. The data may look like in Table 4.1. We are interested in the effect of *treatment* (here flu shot) on *outcome* (here getting sick with flu). This example only contains four observations but we can imagine a similar dataset of thousands of lines. Here we are interested in the flu shot as a binary on-off event, either someone got it, or did not get it. We are not interested in the dose (the amount of the active substance), timing or type of the flu shot. In a similar fashion, we record outcome as a binary variable: flu or no flu. We do not measure severity of the illness, and we do not distinguish between different strains of the virus. But in the population, we do not just look at the binary flu or no-flu event, but compute the probability to get flu.

Whatever the size of the table, for our purpose it can be summarized in just two numbers: average flu for those who got flu shot (0 according in Table 4.1) and for

Id	Flu shot S	Flu F
1	0	1
2	0	0
3	1	0
4	1	0

Table 4.1: Example flu shot data. Id is the patient id, Flu shot is a dummy variable denoting whether the person got ($S = 1$) or did not get ($S = 0$) a flu shot, and Flu denotes whether they got flu ($F = 1$) or not ($F = 0$). The table shows four observations only, but there can be many more.

those who did not get flu shot (0.5 in the table). Formally, we can write

$$\mathbb{E}[F|S = 0] = 0.5 \quad \text{and} \quad \mathbb{E}[F|S = 1] = 0.0. \quad (4.3.1)$$

If we compute the difference between these two groups we get

$$\mathbb{E}[F|S = 1] - \mathbb{E}[F|S = 0] = -0.5 \quad (4.3.2)$$

Those who got the flu shot are 50 percentage points less likely to contract flu, at least in average based on these data. As this problem is framed, the data tends to make people to believe that flu shots are indeed effective. If we want to generalize from these 4 observations alone, we measure the difference in the flu rate for the no-flu shot group and the flu shot group. In this example it is 50 percentage points, so one may want to conclude that flu shots make the flu risk 50 percentage points lower.

However, this conclusion is premature. This empirical regularity—flu shot is associated with 50 pct points lower probability to get flu—can be explained in three fundamentally different ways, each involving very different reasoning and very different implications.

Note the specific choice of words—*associated with*. This is a common way to say what the table tells while avoiding any misleading causal claims. It literally means that those with flu shot have lower probability to contract flu. That is all it means. In particular, it does not mean that the lower probability is *because* of the flu shot. It does not mean that expanding the flu shot program to more people would lower the incidence rate. It does not mean that if *you* get flu shot then *you* will be less likely to get flu. (See also [Section 3.1.3 Correlation and causation](#), page 136.) Choosing an appropriate vocabulary, and being able to understand and correct the common misconceptions is extremely important when working with causal inference.

Next, we discuss the three possible ways to explain data in Table 4.1.

Percentage point: difference in values that are given in percentages (see [Section 1.1.1 Ratio measures](#), page 4).

Model 1: Flu shot causes (no) flu To start with, it is possible that flu shot has a direct impact on the flu, in particular on the probability to get flu. Schematically, we can write it as a *causal diagram*



Empirical observations may show that those who got flu shot are less likely to contract flu. Such a regularity is easy to measure in widely available datasets. Unfortunately, it does not mean that flu shot is effective—it does not mean that if more people will get flu shot, then less people will get sick. Neither does it mean that if you get flu shot then you are less likely to get flu. In order to address these claims, we need very specific data that, unfortunately, is much harder to collect.

Yuemin Cao, CC0 1.0



Figure 4.1: Does flu shot help to avoid flu?

The example data above suggests that if this causal interpretation is correct, flu shot is highly effective by lowering the flu probability by 50pt points. Hence the policymakers should encourage more people to get a flu shot.

This is the easy-to-understand explanation we discussed above, and it is something people intuitively tend to do if the problem is framed as above. While not necessarily true, this is definitely a strong candidate explanation for the effect we see in data.

Model 2: Flu causes (no) flu shot Alternatively, the exact same data can be generated if it is flu instead that has an effect on flu shot. For instance, people who do not feel well may avoid flu shot because they do not want to go to pharmacy to get it, while the healthy ones will get it. The causal diagram will run the opposite way:



The result, in terms of data, will look exactly the same as in Table 4.1. The example explanation above—only healthy people will go out to get the shot—is often referred as *self-selection*, the case where people select into treatment depending on the outcome. In our example, the flu shot may be completely worthless, even harmful, but because these are healthy people who get it (self-select into treatment), the previous causal model still indicates as if it were highly effective. Hence in the current example, self-selection biases the estimate upward—makes flu shot to look more effective than it actually is. If the upward bias is large enough, then even a harmful treatment may appear effective. And this is not just a question for academic research but of immediate policy relevance. If flu shot is worthless, there is no reason to recommend it to more people. If it is harmful, we should abolish the program completely!

What is the reason we may get a completely wrong result? The problem here is neither data collection nor analysis but the causal model. If we use Model 1 to

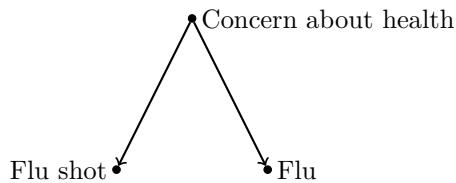
analyze data generated by Model 2, we get wrong results.

Exercise 4.1: Self-selection and downward bias

The above example described a mechanism (sick people avoid going out) that causes the flu shot to seem more effective than it actually is (upward bias). Give an example of *downward bias* through self-selection—a way for the flu shot to seem less effective than it actually is.

Solution on page [426](#).

Model 3: A third factor causes both flu and flu shot As a third possibility, there may be other factors that explain why some people get flu shot and do not get flu. For instance, those who are more concerned about their health may take flu shot, but they also wash hands, dress appropriately according to weather and have a more healthy diet. As a result they do not get flu even if the flu shot itself is worthless. The causal diagram will look like



This is another example of self-selection where people who are less likely to get flu self-select into treatment, and those who are more likely to contract it will select into no-treatment. As a result, the estimated effect will be upward biased.

What distinguishes model 3 from model 2 is the fact that here the self-selection is not based on outcome but on *confounding factors*, other factors that explain whether someone takes flu shot. If we can incorporate confounding factors into the model, we can eliminate the problem. But when working with complex questions, such as human behavior, we rarely have information about all relevant factors. In the example above, while data about flu and flu shots may be abundant in medical records, information on general health behavior (such as how often someone washes hands) is fragmentary at best and more likely completely missing.

If Model 3 turns out to be the correct causal model then there is again little reason to suggest that more people should get a flu shot. The health authorities should instead recommend washing hands and eating more vegetables.

Exercise 4.2: Confounding factors and downward bias

The example above, again, argued that confounding factors (concern about one's health) can cause an upward bias—flu shot seeming more effective than it actually is. Can you come up with different confounding factors, ones that can make flu shot seem less effective than it actually is? Can you tell which of these processes are more likely?

Solution on page [426](#).

So the exact same dataset gives us different results, depending on which causal model we use. Unfortunately, typical data, such as in Table 4.1, does not provide any guidance on which causal model is correct. Even more, in complex cases (and human behavior is complex) they can all be correct at the same time and influence our results together in different ways. Data in the table is not enough to provide causal explanation. There are a number of strategies one may follow to establish causality. Randomized Controlled Trials are considered the best option, followed by natural experiments, case-control studies, and other methods.

Exercise 4.3: Does smoking cause lung cancer?

Lung cancer was historically a very rare disease. However, by 1960-s, it had become the most common cancer type in the West, and it was clearly correlated with smoking. But does smoking cause cancer?

Explain the correlation between smoking and cancer using all three causal models: smoking causes cancer, cancer causes smoking, and confounding factors cause both smoking and cancer.

Solution: see Example 4.1 below.

Example 4.1: Smoking and lung cancer

By 1960s, cigarettes were the dominant way of consuming tobacco and one could easily see that the rapid growth of tobacco smoking was accompanied with an explosive growth of lung cancer with a roughly 20-year lag. But a lot else had also changed by 1960, including urbanization, transportation and the chemical environment in our homes. So the correlation was not a proof of smoking being harmful.

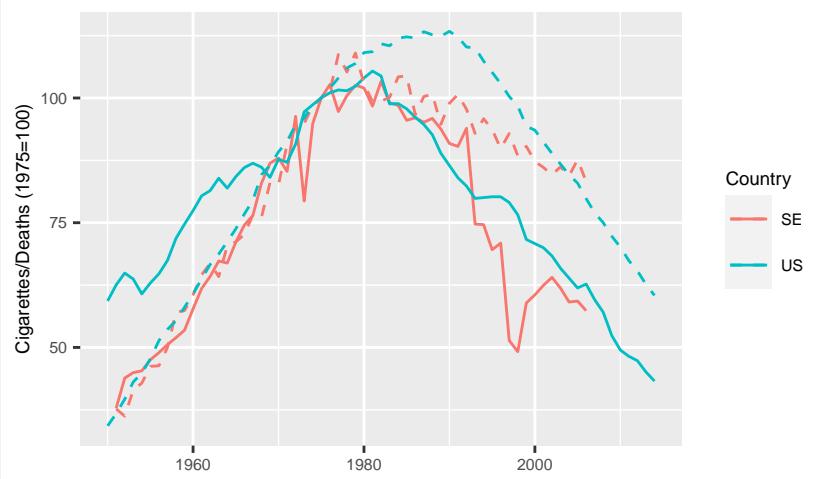


Figure 4.2: Time trends of smoking and lung cancer in Sweden (red) and the U.S. (blue), 1975=100. Solid lines depict the cigarette sales, and dashed lines are age-normalized death rates in lung cancer. We can see cigarette sales in both countries peaking around 1980. This is followed by a fall in death rate somewhat later.

Source: [International Smoking Statistics, WHO Cancer Mortality Database](#).

Already back then there were many doctors who suggested that smoking causes cancer. In particular, they argued that tars and nicotine in the tobacco smoke disturbs the growth-control mechanism in lung cells. Thus they argued that the first causal model is the correct one.

But doctors' opinion is a weak argument. Doctors have got it wrong many times, for instance, even in late 19th century many doctors insisted that scurvy is caused by tainted food, "damp air", salty meat and various other factors, despite that already in 16th century it had become evident that fresh food and citrus fruits will rapidly cure the disease. In particular the tobacco industry insisted that the explanation is "smoking gene". They argued that smoking itself is harmless, but people who smoke tend to have "smoking gene", certain set of genes that makes them likely to smoke, but also likely to get cancer. This is the third model with smoking gene being the confounding factor.

Finally, it is also possible that cancer causes smoking: in particular, cancer in very early stages before it is diagnosed, or before it even can be diagnosed, makes people to itch for a cigarette. This is model 2, reverse causality explanation.

4.4 Strategies for Causal Inference

The previous section explained why do we need to know which causal model is the correct one in order to establish the causal effect from data. We also explained that data alone is not enough to decide which model is correct, we also need knowledge of the possible selection mechanisms. This section discusses a few ways to acquire such knowledge.

4.4.1 We Know Which Model is Feasible

Sometimes it is possible to eliminate one or two explanations based on different type of information. Often we know that the decision to take or not to take treatment preceded the outcome in time, and hence Model 2 is infeasible. We may also know plausible physical or physiological mechanisms that can carry influence from treatment to outcome while there are no plausible confounding factors.

Unfortunately, in many important applications this is not true. In case of social processes and human behavior, one can often provide multiple plausible explanations supporting all three causal models.

Example 4.2: Do parachutes help to survive a "gravitational challenge"?

[Smith and Pell \(2003\)](#) take an absurd example and discuss the effect of parachutes on survival for jumping from aircraft. As explained above, we have three possible causal models:

1. Parachutes cause survival. This is the obvious explanation no-one (except [Smith and Pell, 2003](#)) can argue with. There is also plenty of medical evidence about how our bodies react to rapid acceleration.

2. Survival causes parachute use. Here we can eliminate this potential mechanism because the decision to use or not to use parachute must have preceded survival—parachutists were alive when leaving the airplane. Hence causality cannot flow this way.
3. A third factor can cause both parachute use and survival. Although in principle this is possible, it is hard to come up with any plausible mechanism that might cause such an observed link ([Smith and Pell \(2003\)](#) suggest this is mental health).

So in this case we are able to eliminate both model 2 and model 3, and hence model 1 must be correct. Note that the elimination was not based on data (a table of observations of parachute use and survival) but on more general knowledge about parachutes, decisions, and how our bodies work.

4.4.2 Randomized Controlled Trials

Randomized Controlled Trials (RCT)-s are considered the “gold standard” of causal inference. The idea of RCT is to randomly assign individuals into treatment group and control group. Taking the flu shot example from above, all members of the treatment group receive the flu shot, while none of the control group members will get it. Most importantly, RCT assigns treatment through a random mechanism. This means that treatment status depends on a random event, such as a random number generated by computer, and hence it cannot depend on the outcome (as stipulated by model 2) or by confounding factors (as suggested by model 3). Hence we can immediately eliminate models 2 and 3. Only model 1 will remain as a feasible explanation.

For instance, we can imagine conducting a RCT regarding the flu shot efficiency. We need a large number of participants who are willing to give their explicit consent to join the experiment. Next, we randomize all participants into the treatment and control group. If possible, the trials are *double-blind*, i.e. neither the volunteer who receives a shot, nor the nurse who administers it knows whether the syringe contains placebo or vaccine (syringes may be labeled, and the information about what each label contains is not released before the experiment is over). Those who receive the actual vaccine will form the treatment group and those who received placebo will form the control group. Placebo may be a similar injection as the vaccine, just without the active substance (the injection is mainly water, salt, and other unrelated substances). Later one collects the participants’ health information through the flu season, and when this is done, the treatment/control group information is released. Now we can analyze whether the flu shot was effective.

So the idea of RCT-s is very simple and the results are convincing. Unfortunately, RCT-s are not without their downsides.

- Most importantly, there are many questions where conducting RCT-s would be unethical, illegal, too expensive, or completely infeasible. For instance, it may be considered illegal to pay different workers different wage for similar work, even if it allows us to get valuable information about how work motivation depends on income. Alternatively, if we want to analyze how does tax rate influence macroeconomic performance then we have to randomize the countries

into low-tax and high-tax regime, and to ensure the governments are conforming with this protocol for over a decade or more. This is clearly impossible.

- Humans may react to the fact that they are in either treatment or control group. In simple medical experiment this can be addressed by placebos, but and in many cases we cannot design a convincing placebo. For instance, when analyzing the effect of content of education, it is impossible to design a “placebo education” program in a way that the participants do not understand if they are taught “real knowledge” or “placebo knowledge”.
- RCT-s may also be infeasible if cases of interest are rare and the delay in creating a suitable sample may be unacceptable. A similar situation may occur even while the cases are fairly common but the set-up is considered too high-risk (such as suicide attempts). In such cases we want to act immediately and not follow the data collection protocol.
- Randomization attempts to ensure that the treatment and control groups are similar in all respects, except that the former group receives treatment. However, because of the random nature of randomization, this may not be true in small samples.
- It is hard to experiment with humans who may drop out of study and otherwise violate the assigned protocols.
- RCT analysis assumes the treatment is pre-determined and does not depend on outcome. But in many applications, such as psychotherapy, the standard practice is to adjust treatment depending on the outcome.

Example 4.3: RCT—how to determine the effect of pneumonia vaccine

Bonten *et al.* (2015) analyze the efficacy of polysaccharide conjugate vaccine against pneumococcal pneumonia (effect of a specific vaccine on a particular type of pneumonia). This is a good example how a relatively straightforward RCT application—to determine the efficacy of a new drug—is quite hard to conduct in practice.

The authors enrolled 84,496 elderly (65 year old or older) into the study. The participants must have had no previous pneumococcal vaccinations and no immuno-compromising conditions. The randomization was done by randomizing syringes in the shipment box with either vaccine or placebo. No participant (including medical workers) knew the randomization status. The final sample included 42,240 vaccinated persons and 42,256 placebos. The pneumonia data was collected 2008–2013 in different medical centers. Participants who received other related vaccines, or developed other diseases, such as lung cancer, were excluded from the analysis. The participants were followed by home visit for the next two years to detect any side effects.

Pneumonia was suspected in 3232 cases, out of which the analysts detected 89 relevant pneumonia cases among the vaccinated and 178 among the placebo group. There were too few pneumonia-related deaths to make any conclusions if the vaccine helps to prevent deaths, the study did not find any evidence about adverse effects like chronic medical conditions.

The main difficulty for the study was the small number of relevant pneumonia cases that necessitated both the enormous sample and a long study period.

4.4.3 Natural Experiments

(Sometimes called *quasi-experiment*). Sometimes either nature or human institutions may provide a situation that is similar to a randomized experiment. From research perspective it is particularly valuable if it happens in a context where RCT is not possible.

Below we list a few examples:

- Resettlement of Karelians in Finland at the end of WW2. In WW2, Soviet Union captured a large swath of Finnish territory. The Finnish inhabitants, mostly farmers, were rapidly resettled in various places in Finland where land was available. This created essentially a random experiment where the population of various villages was increased in a rapid and random manner.
- WW2 German missiles destroying city blocks in London. The precision of Nazi V2 missiles was good enough to hit London, but inside of the city, they exploded essentially in random locations. This creates an experiment where certain city blocks were randomly destroyed. How does such destruction affect urban development?
- Collapse of bridge. This is an abrupt change in commuting options. Importantly, only causal model 1—bridge collapse leads to change in commuting—is possible, model 2 (commuting change causes the bridge to collapse) and model 3 (confounding factors cause both commuting change and bridge collapse) are

not feasible.

- Opening a new college that attracts new type of students. This allows to analyze the effect of college education on this group of students. Obviously, the new college causes the new students to attend, not the other way around. We may also be able to eliminate confounding factors if the opening did not coincide with a sudden improvement of economic fortunes of the same group.
- Curriculum changes from one year to another. This is analogous to the previous example. Curriculum change causes students to change the subjects they learn, not the other way around; and we may also be able to eliminate confounding factors.
- [Correia et al. \(2020\)](#) analyze the effect of 1918 influenza pandemic on regional mortality and post-epidemic economic development. See Example 4.4. They use the fact that cities and states in the US implemented the interventions—closing businesses, banning gatherings and promoting hygiene—at different point of time. The authors argue that timing of these measures is as good as random, i.e. not related to the unobserved mortality and economic trends in any systematic way so we can consider this as an experiment.

Example 4.4: Do more extensive public health measures during pandemic help economy? [Correia et al. \(2020\)](#)

The 1918 flu epidemic was perhaps the largest pandemic in the 20th century, killing approximately 50 million people in slightly over a year.^a In the US, the public health response was largely left to the individual cities to decide, and typically included school closures, public gathering bans, and isolation and quarantine, and may also contained altered work schedules, business closures, face mask ordinances and other measures ([Markel et al., 2007](#)). What makes the response to a natural experiment is the fact that cities implemented these responses (non-pharmaceutical interventions, NPI-s) in different time.

[Correia et al. \(2020\)](#). analyze two relationships:

1. How does 1918 flu mortality influence the subsequent economic recovery and development?
2. How do NPI-s, implemented during the pandemic, influence economy? Note that NPI-s have potentially two effects: first through their effect on mortality, and second through direct influence on economy of certain NPI-s, such as bans on public gatherings or businesses closures.

As natural experiment is not a RCT, we do not have well-defined control and treatment groups. Instead, we have a number of cities that implemented different NPI-s at different point of time. As the study analyses economic recovery *after* the pandemic, the causality cannot go from economy to pandemic. Model 2 is eliminated. Regarding model 3 the authors argue that mortality was not related to economic shocks. Hence there were no hidden confounding factors that determined both mortality and the economic development later. The only effect from mortality to economy was the direct effect: mortality influenced behavior, economic decisions, and hence economic growth. This leaves only model 1, the direct causal effect, to explain the findings regarding the question 1.

For the second question, authors employ the variation of type and timing of

NPI-s. In a similar fashion, they argue that “variation across cities is unrelated to economic fundamentals”, i.e. there were no hidden confounders that determined both timing of NPI-s and economic recovery later. The only effect from NPI-s was through their influence on mortality, morbidity, human behavior and hence economy (model 1). These two arguments form the identifying assumptions for the models.

Formally, they estimate models of the form (see more in Section [Section 4.5 Causal inference in linear regression framework](#), page 205)

$$\begin{aligned} y_{st} &= \alpha_s + \tau_t + \beta_t M_{s,1918} + \mathbf{X}_s \gamma_t + \epsilon_{st}^M \quad t \neq 1918 \\ y_{st} &= \alpha_s + \tau_t + \beta_t NPI_{s,1918} + \mathbf{X}_s \gamma_t + \epsilon_{st}^{NPI} \quad t \neq 1918 \end{aligned} \quad (4.4.1)$$

where y is an economic development indicator for city s in year t , α and τ are constants, M is mortality, and \mathbf{X} are all other city-specific covariates. Formally, the identifying assumptions are

$$M_{1918} \perp\!\!\!\perp \epsilon^M \quad \text{and} \quad NPI_{1918} \perp\!\!\!\perp \epsilon^M, \quad (4.4.2)$$

The former assumes there were no confounding factors between economic outcomes and mortality, and the latter assumes no confounding factors between economic outcomes and NPI .

The authors find substantial effects in both models. Increased mortality has substantial negative economic effects, including fall in employment, manufacturing output, bank assets and investments in durable goods. In contrary, earlier and more forceful public health interventions do not lead to worse economic outcomes but the way around, more employment, output and assets. (Some of the results are not statistically significant though).

^aIn comparison, the First World War that ended in the same year killed approximately 10 million in over four years.

4.4.4 Case-Control Study

In many contexts where it is not possible to conduct a RCT, one may compare different cases with different outcomes, and see if there are more “treated” cases in one group. For instance, one can compare patients with diagnosed lung cancer and patients with no cancer in a certain age group, and compare the percentage of smokers in these groups.

Unlike RCT, case-control studies cannot unambiguously establish causality. They are similar to other observational studies that establish correlation, but in order to eliminate other causal models we still need additional information.

Example 4.5: Flu Vaccine Efficacy: a Case-Control Study

Ferdinands *et al.* (2014) conduct a case-control study to analyze influenza vaccine efficacy for children. They enroll 216 children (6 month to 17 year olds) who are admitted in intensive care units with acute respiratory problems in selected

hospitals. 44 of these children are diagnosed flu and 172 are not. Those with flu are “cases” while those without flu are “controls”. As both groups are selected from children who are in a similar situation, admitted into intensive care with respiratory problems, they are broadly similar.

The authors analyze what proportion of cases and controls have been vaccinated against influenza, and find that complete flu vaccination is much less prevalent among cases (odds ratio 0.26). They conclude that vaccination is “associated with a three-quarters reduction in the risk of life-threatening influenza illness in children”.

The study shows convincingly that vaccination is associated with less flu among seriously ill children. However, they cannot eliminate reverse causality and confounding factors, such as healthy lifestyle. See [Section 4.3 Causality with data: three explanations](#), page 194.

4.4.5 Controlling for Confounding Factors

One of the most obvious solution is to explicitly control for all available confounding factors. Unfortunately, all relevant information is rarely present.

But there are examples where researchers can access complete relevant information. Consider college admission. The procedures differ, but in some colleges students are admitted based on limited information only. This may include test score (e.g. SAT test), high school GPA, and essay that is graded from 1 to 5. Importantly, we know that these three variables is everything that decides college admission, and the researchers may get access to these data. If the is the case then they will be able to completely control for confounding factors.

4.4.6 Explicit Modeling of Selection Process

Sometimes we can model the selection process based on theoretical considerations.

TBD: Heckman’s method.

4.5 Causal inference in linear regression framework

Prerequisites: [Section 3.1.2 Simple Regression](#), page 125, [Section 4.3 Causality with data: three explanations](#), page 194, [Section 4.4 Strategies for Causal Inference](#), page 199

This section discusses some of the causality aspects more formally in a linear regression framework. Linear regression is just a simple and popular framework but the central ideas here carry over to all other statistical models. In particular, the fundamental problem, “curse of counterfactual”, is always there, no matter which model we are using. A formal statistical presentation is useful because this helps to identify the exact technical requirements for the data. Thereafter we can analyze how the particular dataset is collected and discuss whether these requirements are satisfied.

Many important causal questions, for instance

- does the drug cure illness?
- how will getting a college degree affect my income?
- does the advertisement work?

can be written as linear regression problems in the form

$$y_i = \beta_0 + \beta_1 T_i + \epsilon_i \quad (4.5.1)$$

where y is the outcome (illness, income, or whether someone buys the product), T is *treatment*, the indicator whether the person attended a college, took the drug, or was shown the advertisement. β_0 and β_1 are parameters we want to calculate. Here we discuss the case where T can be measured as a binary 0/1 indicator variable (for instance, whether the person took the drug or received placebo instead). The central parameter of interest in (4.5.1) is β_1 . This tells us how much larger (or smaller) y would be if $T = 1$ instead of $T = 0$. This is exactly the causal effect we are interested in, the effect we can use for policy design. The disturbance term ϵ captures the individual-specific effects, e.g. individual responsiveness to drugs and illnesses, or learning ability. These individual specific effects do not depend on T .

4.5.1 Counterfactual and Identifying Assumption

(3.1.1): $y_i = \beta_0 + x_i \cdot \beta_1 + \epsilon_i$

Let us start with (4.5.1). Why does β_1 answer the causal question here? What is the difference between (3.1.1) and (4.5.1)? Why did we warn with the former that the β -s cannot be interpreted causally and why don't we do it here? After all, both models are almost the same!

In fact, the difference is not in the models as written above. The models are practically the same. The difference is in what are we analyzing. Figure 4.3 shows both of these cases. It depicts education and pay for two persons: Xuande and Zhang Fei. The left panel shows the causal effect: it is the additional income for *the same person*, here Xuande, given he has college degree, compared to the case where he does not have the degree. So if we give Xuande more education, he will receive more pay. These manipulations are depicted by arrows. We stress here that causality is *related to manipulations*, causal effect is the answer to exactly such questions: what happens if we manipulate education? This case corresponds to (4.5.1), it is also the essence of the causal Model 1.

The right panel shows the other case, the case that answers the question of correlation, but not the causal effect. Instead of seeing Xuande in two states, with and without college degree, we see two different people. One of them is Xuande without the degree and the other is Zhang Fei with the degree. We see that besides of having more education, Zhang Fei is also paid better, the differences are denoted by dotted lines. But based on this figure alone we do not know if Xuande will receive a similar pay if we could manipulate his education to be similar to what Zhang Fei has. This is the case of (3.1.1). Unless we know more, we cannot tell which causal model is behind these data.¹ So the difference is not in the linear regression models, but in

¹Here we denote the causal effect by β_1 and the correlation by α_1 , however, in the regression models we usually do not make such distinction in notation.

Model 1: 

the types of data we are analyzing: does the data contain information about such manipulation or not?

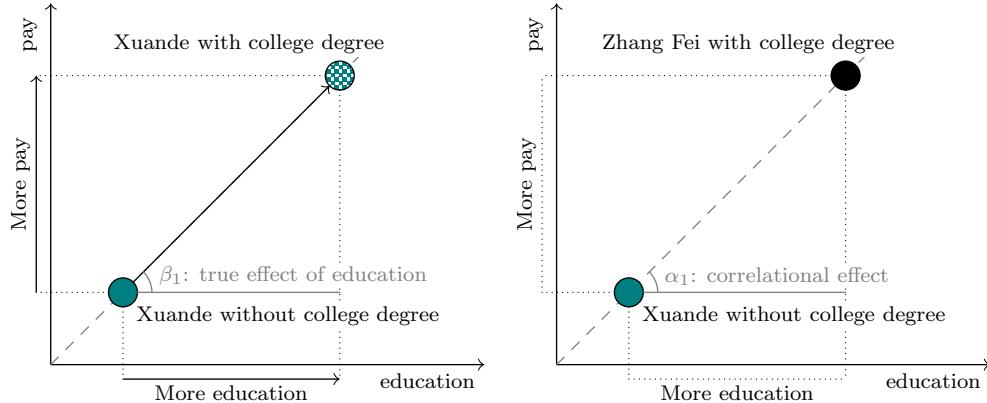


Figure 4.3: Causal versus correlational data. The left panel answers the causal question: what will happen to Xuande, if he gets more education? The corresponding shifts are denoted by arrows. The right panel compares the education and pay of two different persons, Xuande and Zhang Fei, their differences are denoted by dotted lines. The left figure is based on the causal Model 1. The right image can be based on any causal model, and unless we know more, we cannot tell what will be Xuande's pay if he would obtain a similar education as Zhang Fei. On these figures we have made the causal effect β_1 and the correlational effect α_1 to be equal, but it does not to be so.

Next, let's look at (4.5.1) more formally. In this case we know that data is generated by Model 1. Now we can just replace $T = 0$ with $T = 1$ in order to compute the outcome when treated, or the way around. So the outcome of individual i is

$$y_i = \begin{cases} \beta_0 + \epsilon_i & \text{if not treated; denote this by } y_i(0) \\ \beta_0 + \beta_1 + \epsilon_i & \text{if treated; denote this by } y_i(1) \end{cases} \quad (4.5.2)$$

where we have denoted the outcome y_i as a function of the treatment status T by $y_i(T)$. Importantly, the disturbance term ϵ_i is the same for both $T_i = 0$ and $T_i = 1$. The disturbance term must not depend on treatment.²

How can we compute β_1 ? From (4.5.2) we see immediately that $\beta_1 = y_i(1) - y_i(0)$, i.e. the effect is just the difference between the outcome when treated, $y_i(1)$, and when non-treated, $y_i(0)$, and this is true for every single individual i . This sounds almost like a trivial thing to compute. And indeed, it were trivial if only we could measure both $y_i(1)$ and $y_i(0)$. But unfortunately we can never, never ever, observe $y_i(1)$ and $y_i(0)$ at the same time. Someone (or something) is either treated or not treated. Nothing can be in two treatment states at the same time.³ In our observed world the

²This is actually a rather harmless assumption here where we know the correct causal model. If, in fact, ϵ does depend on T , then the change of the disturbance term will just be a part of the treatment effect. But it is a crucial problem in case where we do not know the model.

³We stay in the macroscopic world and do not discuss quantum superposition, Schrödinger's cat and related topics here.

treatment status is either $T_i = 0$ and only $T_i = 0$, or $T_i = 1$ and only $T_i = 1$. The corresponding outcome is called *actual outcome*, and the other, the one we cannot observe, is *counterfactual outcome* (or just *counterfactual*). The actual outcome is easy to handle: this is what you observe. Just measure it. All the trouble with causal inference is to come up with a suitable proxy for the counterfactual. We stress here that the only way to “measure” counterfactual is to find a good proxy. It is fundamentally impossible to measure the counterfactual value. This is the “curse of counterfactual”. Hence, and unfortunately, it is impossible to observe data that corresponds to the left panel of Figure 4.3. We are stuck with the correlational right panel.

Example 4.6: Former outcome as counterfactual

It may be tempting to use pre-treatment observations as proxies for post-treatment counterfactuals. This may or may not be correct but in any case, it requires additional justification.

For instance, returning to the question of effect of college degree on income, we might consider taking pre-college income as counterfactual for post-college income. Obviously, this is absurd. At the time students start college they are fresh high school graduates with little to no work experience and often with no job. Had they not attended college, they would be working in most cases by now, and have 4 years of work experience by now. A 18-year old person without work experience will not form a valid proxy for a 23-year old with 4 years of experience.

See more in [Section 4.5.3 Before-after estimator](#), page 216.

As counterfactual is not observed, it is hard to say anything about it based on data, or at least based on data alone. Coming up with a convincing counterfactual always includes certain assumptions, usually referred to as *identifying assumptions* or *counterfactual assumptions*. For instance, such an assumption may state that in average, the treatment group outcome would be the same as the control group outcome, if the members of the treatment group had not received treatment. In practice, these assumptions must always be backed up with knowledge about the data—not *by data* but *about data*, knowledge about how the data has been generated. The assumption we just cited seems credible if we performed a randomized controlled trial (RCT). After all, randomization is done for this exact purpose, to make the treatment and control group look exactly the same in all known and unknown dimensions. However, it will not be credible at all if we are comparing salaries of college graduates and non-graduates. College graduates and non-graduates differ in many aspects, not just by the fact that one group has spent several years of their life as students. Note that the knowledge about selection process is not usually called data. Just obtaining more “data”, i.e. observations about treated and non-treated outcomes, does not allow us to make more credible conclusions about the causal effect. We need *both* data *and* knowledge about the selection process. If we are lucky then the latter will allow us to come up with convincing identifying assumptions.

4.5.2 More about identifying assumptions: mean independence

What happens if the identifying assumptions are wrong? Below we explain that this is similar to using a wrong causal model.

Take the linear regression model (4.5.1) as the point of departure. For simplicity, that model only contains a single explanatory variable, the treatment status T , but one can easily add more. Data we collect consists of tuples in the form (T, y) , i.e. every case is a pair of two numbers, the treatment status and the corresponding outcome. How can we estimate β_1 ? Intuitively, in a large sample, the average outcome for the treated, $\bar{y}(1)$, and for the untreated, $\bar{y}(0)$, should tell us something about the effect. In particular, we are tempted to interpret their difference $\bar{y}(1) - \bar{y}(0)$ as the treatment effect. (We talk about large samples in order to avoid issues with sampling noise, those issues are not related to causality.)

The intuitive concept of “average over a large sample” corresponds to mathematical concept of expected value, so we can replace the large sample average with the corresponding expected value. We denote the expected values by $\mathbb{E}[y|T = 1]$ (for the treated) and $\mathbb{E}[y|T = 0]$ (for the non-treated). Next, we can use model (4.5.1) to compute these values as

$$\mathbb{E}[y|T = 1] = \mathbb{E}[\beta_0 + \beta_1 T + \epsilon|T = 1] = \beta_0 + \beta_1 + \mathbb{E}[\epsilon|T = 1] \quad (4.5.3)$$

and

$$\mathbb{E}[y|T = 0] = \mathbb{E}[\beta_0 + \beta_1 T + \epsilon|T = 0] = \beta_0 + \mathbb{E}[\epsilon|T = 0]. \quad (4.5.4)$$

We used the following facts when calculating the results above: expected value of constants β_0 and β_0 are just these two constants, and β_1 drops out from the second equation because in that case it is multiplied by $T = 0$. But the last terms, the conditional expectations of ϵ , are critical. $\mathbb{E}[\epsilon|T = 1]$ is the expected value of the error term for the treated individuals, $\mathbb{E}[\epsilon|T = 0]$ is the same for the non-treated individuals. In general, these two differ, i.e. $\mathbb{E}[\epsilon|T = 1] \neq \mathbb{E}[\epsilon|T = 0]$.

When we now compute the difference between the two expected values (4.5.3) and (4.5.4), we get

$$\mathbb{E}[y|T = 1] - \mathbb{E}[y|T = 0] = \beta_1 + \mathbb{E}[\epsilon|T = 1] - \mathbb{E}[\epsilon|T = 0]. \quad (4.5.5)$$

Obviously, this equals to the correct value β_1 only in case of

$$\mathbb{E}[\epsilon|T = 1] = \mathbb{E}[\epsilon|T = 0].^4 \quad (4.5.6)$$

This condition is known as *mean independence assumption*, often denoted by $\mathbb{E}[\epsilon \perp\!\!\!\perp T]$. This is the technical way to state the identifying assumption for cross-sectional estimator.

Figure 4.4 explains the role of mean independence and how it affects the effect estimation. It displays the outcomes for four people, Xuande (blue), Guan Yu (black), Zhang Fei (green) and Cao Cao (red). The top left panel displays the causal effect

⁴There is an important difference regarding ϵ here, and in (4.5.2). In the latter we assume that we have access to both $y_i(1)$ and $y_i(0)$ for *the same cases*, i.e. we know the counterfactual value. This is not the case here, and hence we need the assumption (4.5.6).

(4.5.1): $y_i = \beta_0 + \beta_1 T_i + \epsilon_i$

Mean of a large sample converges to expected value, $\bar{X}_N \rightarrow \mathbb{E} X$ as $N \rightarrow \infty$. See [Theorem 1 Law of large numbers, LLN](#), page 42

we are interested in—what will be the outcome of Guan Yu and Xuande if they were treated ($T = 1$) instead of non-treated ($T = 0$). We measure the effect as the difference between the average of the counterfactual outcomes, $\mathbb{E}[y|T = 1]$ (dotted circles) and the average actual outcome, $\mathbb{E}[y|T = 0]$ (solid circles). The observed average $\mathbb{E}[y|T = 0]$ is marked with a solid gray circle and the counterfactual average $\mathbb{E}[y|T = 1]$ is a dotted gray circle. The difference is the causal effect β_1 . Note that both the actual and the counterfactual outcome of Guan Yu and Xuande differ, this is because Xuande has positive ϵ and Guan Yu has negative ϵ . But importantly, their respective ϵ is the same in both treated and non-treated state.

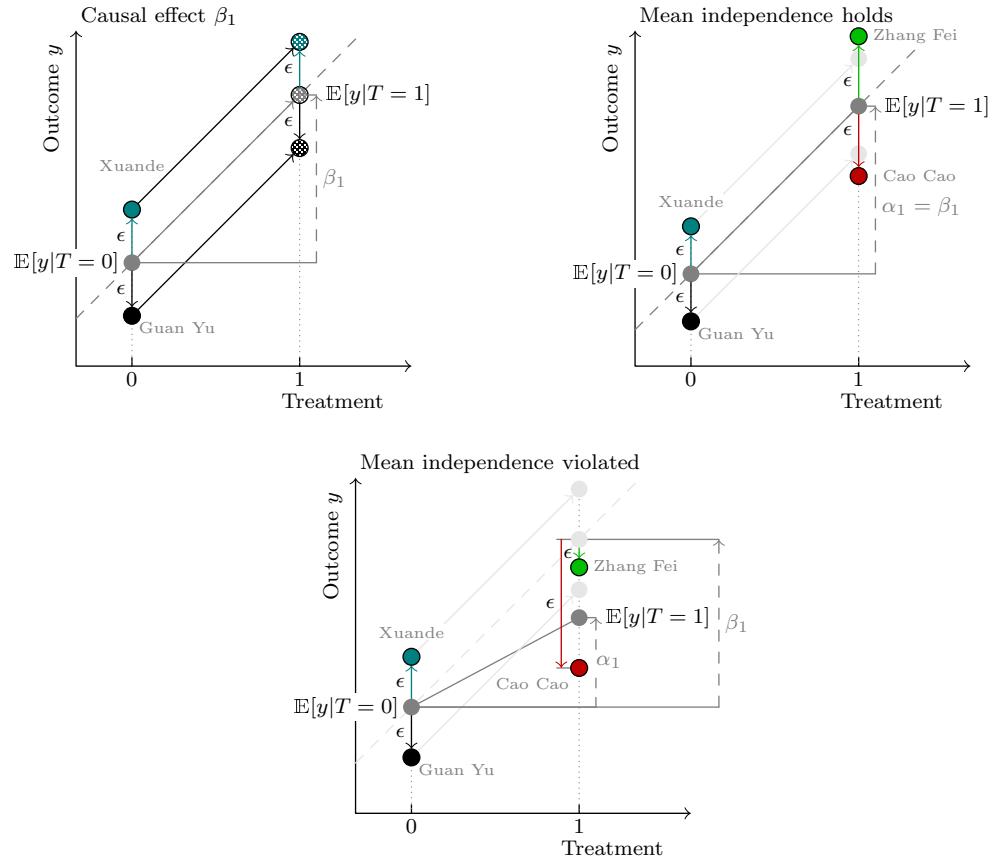


Figure 4.4: The role of mean independence assumption. The top-left panel shows the true causal effect $\beta_1 = \mathbb{E}[y|T = 1] - \mathbb{E}[y|T = 0]$ for two persons—Xuande and Guan Yu. On the top-right panel, the mean independence assumption holds, and the treated group forms a valid counterfactual for the non-treated group. The measured correlational relationship α_1 equals to the true causal effect β_1 . At the bottom panel, the assumption is violated, $\mathbb{E}[\epsilon|T = 0] = 0$ while $\mathbb{E}[\epsilon|T = 1] < 0$, and the correlation effect α_1 underestimates the causal effect. Explanations in text.

However, we cannot measure β_1 of the top-left panel, because the counterfactual

outcome (dotted circles) cannot be observed. What we can do instead is displayed on top-right panel. We compare two untreated persons (Xuande and Guan Yu) with two treated persons (Zhang Fei and Cao Cao). As before, we compute the effect as the difference between average outcome of the treated ($\mathbb{E}[y|T = 1]$) and the untreated ($\mathbb{E}[y|T = 0]$). All four people in our sample have different ϵ , but what is important—both the untreated and the treated have average $\mathbb{E}\epsilon = 0$. Hence the average of treated Zhang Fei and Cao Cao (gray circle at $T = 1$) is the same as the average of counterfactuals for untreated Xuande and Guan Yu (light gray circles at $T = 1$). Hence the difference between the average for the treated and for the untreated, α_1 , is equal to the correct causal effect β_1 . The mean independence assumption is satisfied.

Finally, the bottom panel shows the case where the mean independence assumption is violated. The average value of the error term for untreated Xuande and Guan Yu, $\mathbb{E}[\epsilon|T = 0] = 0$, but for treated Zhang Fei and Cao Cao $\mathbb{E}[\epsilon|T = 1] < 0$ as both of these ϵ -s are negative. Hence the average for the treated group (dark gray circle) is below the average of the counterfactuals (middle light-gray circle), and the measured difference α_1 is less than the true causal effect β_1 .

How is the mean independence assumption related to causal models? It turns out that if mean independence assumption is equivalent to causal Model 1. Let us discuss this from an intuitive viewpoint first, and show it formally thereafter. It is better to visualize the model where treatment is continuous so assume we have a model

$$y_i = \beta_0 + \beta_1 \cdot d_i + \epsilon_i \quad (4.5.7)$$

where d is a continuous measure, the dose of treatment. Further, assume that all the standard linear regression assumptions (see [Section 3.1.10 Assumptions in OLS Models](#), page 171) are satisfied, in particular assumption 5. If this is the case, then data about treatment dose d and outcome y will give us correct, unbiased, estimate of β_1 . This is a direct consequence of the assumptions in [Section 3.1.10](#). Intuitively, $d \perp\!\!\!\perp \epsilon$ means that observations with all kind of x values may have large ϵ values and small ϵ values; it is not that large d tends to have large ϵ values, or the way around. This is why we do get a correct result. This setup corresponds to the causal model 1: what happens to d will influence y , but y does not influence d . True, at any given d value y may be larger or smaller, but that is “taken care of” by the disturbance term ϵ , not by d .

Assumption 5: error term and explanatory variables are independent: $x \perp\!\!\!\perp \epsilon$.

Model 1: $d \rightarrow y$. See [Section 4.3 Causality with data: three explanations](#), page 194.

Model 2: $y \rightarrow d$.

But now imagine the correct model is not the model 1 but model 2. The first problem is that in that case we cannot change y by just manipulating d . Second, we cannot even recover the inverse of the correct estimator. Instead of (4.5.7) we have now

$$d_i = \alpha_0 + \alpha_1 \cdot y_i + \eta_i. \quad (4.5.8)$$

Again, assume all the standard assumptions are satisfied, but because now y is the explanatory variable, the independence assumption means $y \perp\!\!\!\perp \eta$, not $d \perp\!\!\!\perp \eta$! It turns out that d and η are not independent. This is intuitively obvious: cases with large η value also tend to have a large d value and the way around— d and η are correlated. Hence we will not recover the correct relationship (the correct relationship between y and d is $1/\alpha_1$).

TBD: Figure, show formally

Model 3: $z \rightarrow x, y$.

Model 3 causes similar problems, the logic is broadly similar but more complex and we do not discuss it here.

So the mean independence assumption is needed to recover the correct relationship.

Besides of the technical assumptions—which causal model is behind the data, the identifying assumptions always have the intuitive side—what do these technical requirements mean in terms of data, and what do they mean in terms of institutions. For instance, imagine a randomized medical experiment to test a new drug, where the participants are randomized into the treatment group (they receive the drug) and control group (they receive placebo).

- In terms of data, we expect all control and treatment group individuals to be similar to each other in terms of all characteristics, including age and pre-existing conditions. This is because they were assigned to groups by random. If we have additional information about the participants then we can test this—do all the observable variables, e.g. age, gender, family status, education, and so forth, look similar between the groups? If yes then this convinces us that the characteristics we do not know (e.g. sleep behavior and type of diet) may also be similar.
- In terms of institutions we have to ask if researchers were able to correctly follow the randomization. Maybe someone involved in the experiment told the participants what they get and let them choose? Maybe the participants signed up into multiple similar experiments in the hope that at least in one of those they will get the “real thing”? Maybe they found each other through social media and split and shared their pills so that everyone at least “got something”? For the results to be convincing we need to know that nothing similar happened. However, we cannot easily test this based on data.

So (4.5.6) states the technical side of the identifying assumption. If possible, then one should always test the data side. The institutional side cannot typically be tested on data, but on either our general knowledge, or knowledge about the specific situation related to these data. It is the researchers’ responsibility to know and explain the relevant institutions.

Example 4.7: Expected value of unobserved characteristics

Take again the example of college degree and income. The unobserved factors ϵ that influence wage may include socio-economic background, cognitive and non-cognitive skills, health, geographic location (such as country and rural/urban location) and so on. So $\mathbb{E}[\epsilon|T = 1]$ is the expected value of such factors for college graduates and $\mathbb{E}[\epsilon|T = 0]$ is the expected value of the same factors for those who did not attend college.

We know that college graduates tend to have higher socio-economic status, they are more likely urban and living in high-income regions, and they possess more cognitive skills. Both higher-status background and innate skills help to get well-paid jobs later in life too. So there are a lot of observable characteristics that differ between these groups. So it is hard to argue that all other factors are similar. Hence most likely $\mathbb{E}[\epsilon|T = 1] \neq \mathbb{E}[\epsilon|T = 0]$, the mean independence assumption is violated. We can still compare the income of graduates and non-

graduates, but we cannot interpret the result as the causal effect of college degree.

4.5.3 A Few Popular Estimators

Prerequisites: Section 3.1.2 Simple Regression, page 125, Section 3.1.9 Interactions Effects, page 165, independent random variables 1.3.4, conditional expectations 1.4.3.

There are many ways to estimate causal effect β_1 . Here we introduce a few simple and popular methods: cross-sectional estimator, before-after estimator, and differences-in-differences estimator. While the two former methods are simple and popular in media, the latter one is based on slightly more credible assumptions, and is often used in research. These are all based on *fixed effects* approach and assume that certain values or trends are invariant.

Cross-Sectional Estimator

TBD: just difference in means versus OLS

The idea with cross-sectional estimator is very simple: we assume that the difference between treated and non-treated outcomes is due to the treatment, and only due to the treatment, at least in average. If this is the case then untreated cases (controls) form a valid counterfactual, and we get correct estimates by just computing the average difference between the treated and the controls. Formally, we assume $\mathbb{E}[\epsilon|T = 1] = \mathbb{E}[\epsilon|T = 0]$ and hence the effect of interest is

$$\beta_1 = \mathbb{E}[y|T = 1] - \mathbb{E}[y|T = 0] \quad (4.5.9)$$

(see (4.5.5)). As discussed above, this assumption seems a credible one in case of RCTs, and a lot less credible where different type of people can freely decide whether to get treatment. For instance, it is extremely hard to justify that college graduates and non-graduates are similar in every way except graduation. We have many good reasons to think that ϵ and T are systematically related.

Example 4.8: Cross-sectional estimator of college effect is biased

Let us continue Example 4.7. The arguments there suggest that college graduates are drawn from more favorable ends of distribution of ϵ and hence T is positively correlated with ϵ . This means $\mathbb{E}[\epsilon|T = 1] > \mathbb{E}[\epsilon|T = 0]$. As a result the estimator (4.5.5) is upward biased:

$$\mathbb{E}[y|T = 1] - \mathbb{E}[y|T = 0] = \beta_1 + \mathbb{E}[\epsilon|T = 1] - \mathbb{E}[\epsilon|T = 0] > \beta_1 \quad (4.5.10)$$

The bad news is that we don't know by how much is the estimate biased, and based on data alone we cannot tell. Here "data" means a table of college graduation status and income for a large number of individuals. Education is one of the many unfortunate examples where it is hard to find plausible information to break the curse of counterfactual and actually compute the effect.

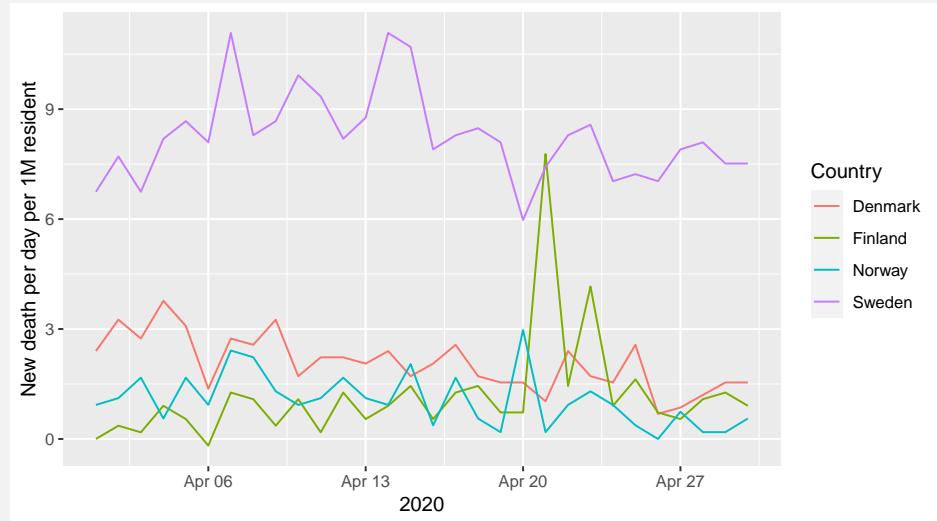
The intuitive side of the assumption was already stated above but we repeat it here: for the cross-sectional estimator to be valid, the average unobserved characteristics of the treated and the controls must be similar. If this is the case, then controls make valid counterfactuals for the treated as they are otherwise similar, except that they received the treatment.

Example 4.9: COVID-19 stay-at-home orders in Nordic countries

Nordic countries^a are small but highly developed countries in Northern Europe. They are rather similar in terms of their institutions, featuring extensive social safety net, high taxes, effective governance and little corruption. The population of Denmark, Finland and Norway is approximately 5 million, that of Sweden is 10 million. During the COVID-19 pandemic in 2020, most European countries issued stay-at-home orders, banned public gatherings and closed non-essential businesses. However, almost nothing was done in Sweden where only gatherings of more than 500 were forbidden, and face masks remained rare.

This suggest that we can estimate the effect of COVID-19 measures by using a cross-sectional estimator where we compare Sweden to other Nordic countries. As Swedish policy was clearly exceptional, we can define it as treatment. So in this case treatment means “not introducing stay-at-home orders”. Non-treatment would then be what other countries did, namely to issue such orders.

We focus here on the first wave of COVID in April 2020.^b The figure below shows the daily number of new deaths in all these three countries.



The figure clearly indicates that the death rate per million residents was much higher in Sweden fluctuating between 6 and 9 over this period. In the comparison countries, it stayed below 3 for most of the time, with the exception of a short peak in Finland.

The CS estimate of the effect is just the difference between the corresponding average values. The average death rate in Sweden is 8.25, in other Nordic

countries it is 1.43 and hence their difference, the effect, is 6.82 (deaths per day per million residents).

^aHere we consider Finland, Denmark, Norway and Sweden.

^bThe data we use originates from <https://raw.githubusercontent.com/datasets/covid-19/master/data/time-series-19-covid-combined.csv>, the prepared dataset is in <https://bitbucket.org/otoomet/lecturenotes/raw/master/data/covid-scandinavia.csv.bz2>.

In practice, it is often useful to use linear regression in the form of (4.5.1) instead of (4.5.9). Linear regression approach has two advantages:

- We can easily include other covariates, e.g. demographic variables such as age distribution in case of COVID death rate. This allows to take into account that the treatment and control group may differ along certain ways we can observe (e.g. patients may be of different age). Adding more covariates when just comparing means can be done only in a very limited fashion. One has to split data not just along the treatment/control group but also along other covariates, and we will rapidly run into curse of dimensionality.
- Linear regression software provides the confidence intervals and statistical significance figures with no additional work. True, we can get the same result when performing *t*-test on the treatment and control samples directly, so this is just a minor convenience.

Example 4.10: COVID-19 stay-at-home orders in Nordic countries: regression approach

Here we replicate the results of Example 4.9 with linear regression. As we define the treatment to be “no lockdowns”, it is equivalent to being “Sweden”, so we can write the model (based on (4.5.1)) as

$$deaths_i = \beta_0 + \beta_1 Sweden_i + \epsilon_i. \quad (4.5.11)$$

The dummy $Sweden_i$ must be understood as for every observation we use a 0/1 dummy that tells if this is an observation about Sweden. The results are in the table below:

	Estimate	Std. Error	t-value	Pr(> t)
Intercept	1.432	0.121	11.845	0.000
Sweden	6.820	0.242	28.210	0.000

The results must be interpreted as follows: *Intercept* is the average daily death rate in case $Sweden = 0$, i.e. the death rate outside Sweden (compare with the results in Example 4.9). *Sweden* is the effect of treatment, i.e. not introducing lockdowns. As the table indicates, the effect is very large compared to the intercept (6.82 versus 1.432) and highly significant. So the model suggests that the “no-lockdown” treatment resulted 5.8-fold increase in death rate. This seems like a very large effect.

Can we conclude that lockdowns elsewhere were a very good idea? Not so fast. First, is the identifying assumption credible? In this case it is $\mathbb{E}[\epsilon | Sweden = 1] =$

$\mathbb{E}[\epsilon | Sweden = 0]$, i.e. the omitted variables for Sweden are similar to those in the other Nordic countries (in average). While there are good reasons to believe that Sweden is somewhat different from its Nordic neighbors, it is hard to believe the difference in death rate should be that big. After all, the standard deviation of the error term is just 1.14^a. This is much smaller than the effect 6.82. So even if the mean independence assumption may not be completely correct, it seems that any bias here is dwarfed by the effect size.

But before we offer any policy conclusions, note two caveats. First, we are talking about a large difference in a small rate (a few cases per million). Maybe it does not matter that much. And second, we do not know what did Sweden benefit from this policy. Did its economy perform better? Did the population maintained better mental health? We cannot give solid policy advice before answering those questions too.

^aSoftware packages typically report the estimated standard error of the residuals.

Before-after estimator

Before-after estimator (BA) is similar to cross-sectional estimator, but instead of comparing two different groups at the same time (the treated and the non-treated), we only look at the same group (treatment group) and compare their outcomes before and after the treatment. If the disturbance term does not change over time (in average) then the difference is the causal effect.

While the main approach remains very similar to CS estimator, it is useful to introduce slightly different notation. Assume there are two time periods: $t = 0$ is time before treatment, and $t = 1$ is time after treatment. The corresponding treatment indicator for individual i is T_{it} with $T_{i0} = 0$ before treatment, and $T_{i1} = 1$ after treatment. So we may write

$$y_{it} = \beta_0 + \beta_1 T_{it} + \epsilon_{it} \quad (4.5.12)$$

Here y_{it} is the outcome of individual i at time t , and two indices for ϵ_{it} indicates that the error term may be different for period 0 and period 1. Because treatment occurs after period 0 and before period 1, we have $T_{i0} = 0$ and $T_{i1} = 1$ for all individuals i . The expected outcome after the treatment, $\mathbb{E}[y|t = 1]$, is now

$$\mathbb{E}[y|t = 1] = \mathbb{E}[y|T = 1] = \beta_0 + \beta_1 + \mathbb{E}[\epsilon|t = 1] \quad (4.5.13)$$

where we use the fact that “after”, at $t = 1$, everyone is treated, i.e. $T = 1$. In a similar fashion, the expected outcome before the treatment is

$$\mathbb{E}[y|t = 0] = \mathbb{E}[y|T = 0] = \beta_0 + \mathbb{E}[\epsilon|t = 0] \quad (4.5.14)$$

and the estimated effect

$$\mathbb{E}[y|T = 1] - \mathbb{E}[y|T = 0] = \beta_1 + \mathbb{E}[\epsilon|t = 1] - \mathbb{E}[\epsilon|t = 0]. \quad (4.5.15)$$

This captures the correct value, the causal effect β_1 , only if $\mathbb{E}[\epsilon|t = 1] - \mathbb{E}[\epsilon|t = 0] = 0$. To put it in words, this means that the expected disturbance term before and

after treatment is similar. Or more plainly—there is no unobserved trend. This is the identifying assumption for the before-after estimator. The requirement is pretty obvious—the BA estimator is just the outcome difference over time, and this is the causal effect only if there is no other time differences interfering with the effect. For instance, if we are using before-after estimator to assess the effect of college degree, we have to assume that if a person had not attended college, her income would have stayed the same. (Note: we only look at those who attended college in this estimator.) This is a completely unrealistic assumption, similar to the claim that high-school students and young workers without college degree in their mid-20s would earn exactly the same. Effect of college attendance is an unfortunate example where it is very hard to find data and institutions that enable causal inference. But at the same time it is also a number that both high-school graduates and policymakers would like to know. However, in other cases before-after estimator may be justified.

Example 4.11: President's approval: before and after September 11th

September 11th terror attacks were a major shock for the U.S. society, and the effect was immediately reflected in the president's approval ratings. The figure below depicts the Presidents (G. W. Bush) approval rating from July till November, 2001. The dashed vertical line is the September 11th terror attacks. Tremendous support to president is immediately obvious from the huge increase in the approval rate.

TBD: Explain data, data source, also in the repo

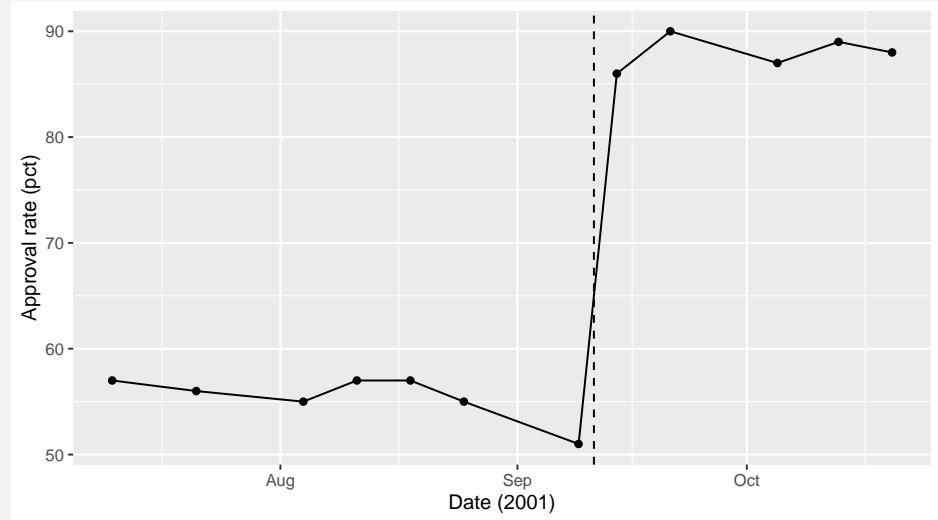


Figure 4.5: U.S. President G. W. Bush approval rating through summer and fall 2001. The dashed vertical line corresponds to September 11 terrorist attacks.

The president's approval rate was hovering around 55% in August and early September, and between 85 and 90% in late September and October. The Sept 7-10 polls indicated that 51% of the respondents approved his performance but

just a week later, September 14-15, it was at 86%. The sharp jump corresponds exactly to the terror attacks, and is missing anywhere else in this data.

G. W. Bush's average approval rate from early July till early September was 55.4% and from mid-September till end of October it was 88%. The difference, 32.6 pct points, is the BA estimate of the effect.

Instead of period means, we can also just look at the last pre-attack poll (support 51%) and the first post-attack poll (support 86%). Using this approach the effect is 35 pct points.

Note that the identifying assumption—no trends besides the effect of treatment—seems to be violated here as the approval rate seems to be heading downhill through the summer. But we consider this to be a minor issue as the 9/11 effect clearly dwarfs the trend.

But this analysis has another important problem. Namely applicability. The analysis seems convincing and interesting, however, the result (35 pct points) can hardly be used for any policy-relevant decisions. This is the opposite to the case of college attendance: we can answer the question, but unfortunately it is not particularly relevant question.

In a fashion, similar to what we did in case of cross-sectional estimator, we can also use linear regression to compute the BA estimate. This can be done by using (4.5.12), typically one also has to create the auxiliary treatment indicator T based on time of the observations.

Example 4.12: Presidents approval: before and after September 11th, the regression approach

Let us revisit the George W. Bush's approval ratings in 2001. First we compute the treatment indicator T . Here treatment is related to time, $T = \mathbb{1}(date > 2001-09-11)$, i.e. it equals to one for all observations after September 11th, and to zero for all observations before that date. However, in order to stress the fact that the treatment is related to observation after the event (and in order to distinguish between treatment group and post-even observations for differences-in-differences estimator), we label it *After* instead. Thereafter we use (4.5.12) as the regression model. The complete data including *After*, 12 observations, is in the table below:

date	Approval, pct	After
2001-07-10	57	0
2001-07-21	56	0
2001-08-04	55	0
2001-08-11	57	0
2001-08-18	57	0
2001-08-25	55	0
2001-09-09	51	0
2001-09-14	86	1
2001-09-21	90	1
2001-10-05	87	1
2001-10-13	89	1
2001-10-20	88	1

Table 4.2: G.W.Bush approval ratings through the first fall of his presidency. The last column, After, is the post September-11 indicator.

Now we adapt model (4.5.12). We use *After* in place of T and drop the index i as the data is just about a single person, so we have

$$y_t = \beta_0 + \beta_1 \cdot \text{After}_t + \epsilon_t. \quad (4.5.16)$$

When we estimate this model, we get the following results:

.	object	..
Intercept	55.429	<i>0.734***</i>
After	32.571	<i>1.137***</i>
# obs	12	
R^2	0.9880	

Table 4.3: DiD regression estimate for the effect of 9/11 terror attacks on presidents approval rating. Standard errors in italics.

The model shows that before the attacks, the approval rate was $\beta_0 = 55.429\%$, and after the attacks it was larger by $\beta_1 = 32.571$ pct points. This is the BA estimate, it is easy to see that it has extremely large t value. The approval level after the attacks is predicted to be $\beta_0 + \beta_1 = 88\%$. These are exactly the same numbers we have in Example 4.11.

Note that credibility of the estimator, the credibility of the identifying assumption, relies on our knowledge of the events through the last decades of 20th and the first decades of 21st century. We know that no other president has seen such a boost in the approval rate, and there has been no unexpected events comparable to September 11th attacks.

Differences-in-differences estimator

Differences-in-differences (also diff-in-diff or DiD) estimator combines the cross-sectional and before-after estimators. The former is biased if the treatment and control groups differ in a way we cannot take into account (i.e. $\mathbb{E}[\epsilon|T = 1] \neq \mathbb{E}[\epsilon|T = 0]$), and the latter if there is an uncontrolled trend in the treated group (i.e. $\mathbb{E}[\epsilon|t = 1] \neq \mathbb{E}[\epsilon|t = 0]$). DiD compares the time trend for the treated group and non-treated group, or what amounts to the same, it compares treated-nontreated differences before and after the treatment. This relaxes the assumptions behind the cross-sectional and before-after estimators and replaces these with a different identifying assumption: time trends for the treated and non-treated groups are the same. However, we pay for the more relaxed assumptions with more stringent data requirement: now we need four data points, two for treated and for non-treated, one before and one after the treatment for each.

Let us first take a hypothetical example. Imagine there is a federal country that contains a number of provinces. In year 2015 certain provinces decided to substantially boost the public education by investing in schools, teachers and outreach. We consider these additional investments to be treatment T , so some provinces were in the treatment group $T = 1$ while others that did not invest are in the control group $T = 0$. According to survey data from 2014, before the treatment began, the average schooling level in the treatment provinces $\bar{y}(T = 1, t = 2014) = 9$ years and in control provinces $\bar{y}(T = 0, t = 2014) = 8$ years. Another survey from 2020, five years into the treatment, found that $\bar{y}(T = 1, t = 2020) = 11$ and $\bar{y}(T = 0, t = 2020) = 9$. (See Figure 4.6.)

We can immediately see that the data does not support the CS and BA identifying assumptions. As the treatment and control groups differ already in 2014, before the treatment even began, it is hard to argue that they would not differ in 2020 if some of the provinces were not investing in education. In a similar fashion, in case of BA estimator, the assumption that without such an investment, the education level of 2020 would be the same as in 2014 for the treatment provinces is not convincing. After all, in control provinces the level is increasing with no treatment whatsoever! What DiD method assumes is that the *trend difference* is due to treatment. So without the treatment, the treatment provinces would have followed the dashed trajectory on the figure, leading up to the counterfactual of 10 years by 2020. However, as the actual outcome was 11 years, the difference, 1 year of extra schooling, is the DiD effect.

Table 4.4: Four datapoints

Time	Control ($T = 0$)	Treatment ($T = 1$)	Difference
Before ($t = 0$)	a	b	$b - a$
After ($t = 1$)	c	d	$d - c$
Trend	$c - a$	$d - b$	
Difference in trend	$(d - b) - (c - a)$	$(d - c) - (b - a)$	

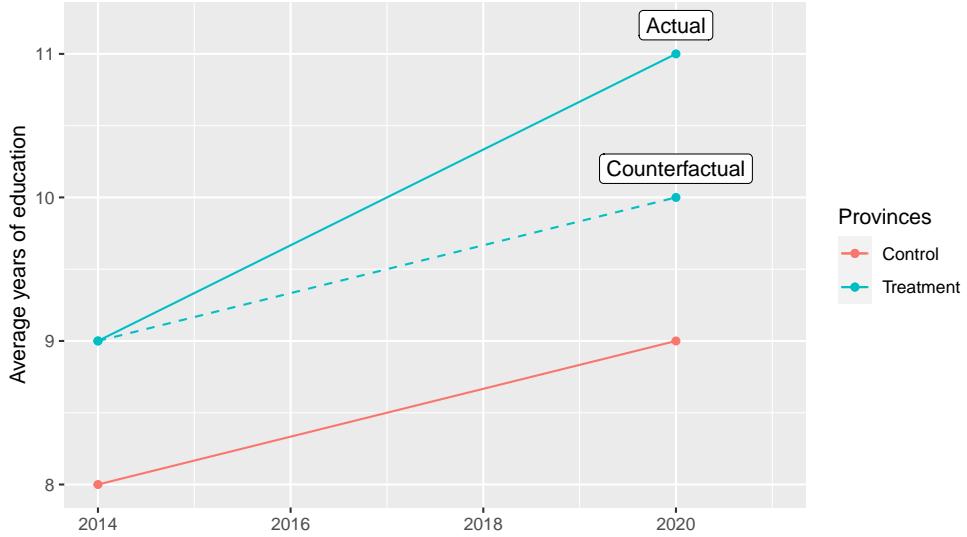


Figure 4.6: Hypothetical education data. Both the levels and trends differ for the treatment and control provinces. Dashed line denotes the counterfactual assumption, the difference between the actual and counterfactual value is the DiD estimate, here 1 year of extra schooling.

Let us now look at this idea more formally. We have two groups, control $T = 0$ and treatment $T = 1$ and two time periods: before $t = 0$ and after $t = 1$. Imagine we have measured these four data points as a (control before), b (treatment before), c (control after) and d (treatment after) (See Table 4.4). So a and b are measured before anyone was treated, and if we want to use cross-sectional assumption, we would expect those to be equal.

The values c and d describe the control and treatment group outcomes after treatment, at $t = 1$. Their their difference, $d - c$, is both due to the treatment effect and other, unobserved, difference. But we assume that the unobserved difference after the treatment equals to that before treatment, $b - a$. This is the identifying assumption. Hence the difference what is left over when we subtract the pre-treatment difference $b - a$ is the treatment effect:

$$\beta = \text{pre-treatment difference} - \text{post-treatment difference} = (d - c) - (b - a). \quad (4.5.17)$$

Alternatively, we can look at the difference over time. The values in the column *control*, a and c , describe the control group outcomes before and after the treatment. In case of before-after estimator they should be equal⁵ but now we allow a time trend $c - a$. The next column, *Treatment*, shows the outcomes for the treatment group where the time trend is $d - b$. This time trend is caused by both treatment and other,

⁵ As above, as no-one in the control group is ever treated, both $Y(0|T = 0, t = 0)$ and $Y(0|T = 1, t = 1)$ are observable so no counterfactual assumption is needed here.

unobserved factors. But we assume that the unobserved trends for the control and treatment group are the same, $c - a$. Hence the difference what is left over when we subtract the control group time trend from the treatment group time trend is the treatment effect:

$$\begin{aligned}\beta &= \text{treatment group trend} - \text{control group trend} = \\ &= (d - b) - (c - a) = (d - c) - (b - a).\end{aligned}\quad (4.5.18)$$

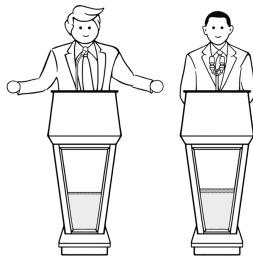
As both of these approaches gave us the same estimate, we can conclude that both assumptions are equivalent. So the identifying assumption for the DiD model can be summarized as:

Unobserved differences between the treatment and control groups are similar before and after treatment (in average)

or

Unobserved time trends for the treatment and control groups are the same.

Example 4.13: COVID-19 Epidemic and Presidents Approval



Elected leaders care about their approval ratings. The numbers are regularly provided by polls. Chesie Yu, CC BY-NC-SA 4.0

Political leaders often enjoy a strong support during the time of crisis. Did the same also apply to the US president Donald Trump in spring 2020, during the COVID-19 epidemic? Let's answer this question with polling data. But as presidents' rating ebbs and flows over time, we compare Trump with Barack Obama using differences-in-differences approach. As spring 2020 was Trump's fourth year in office, we compare his approval trend with that of Barack Obama in 2016, fourth year of Obama's second term in office. The identifying assumption here is that the approval rate trends for Trump in 2020 were similar to those of Obama in 2016, had the COVID epidemic not happened.

A sample of the data is in the table below:

Table 4.5: An excerpt of approval ratings data for presidents Obama and Trump during their fourth year in office. Polling data from [RealClearPolitics](#). The displayed period, from mid-January to mid-April centers on mid-March, the weeks in 2020 where the world, including the US, rapidly realized the magnitude of the unfolding health crisis.

poll	date	approve	president
The Economist/YouGov	2016-01-17	43	Obama
Bloomberg	2016-03-21	50	Obama
NBC News/Wall St. Jnl	2016-05-17	51	Obama
ABC News/Wash Post	2020-01-22	47	Trump
Economist/YouGov	2020-02-10	45	Trump
Reuters/Ipsos	2020-04-13	46	Trump

We choose a single day, March 15th as the day of "treatment". By March 15th 2020 the coronavirus epidemic had become the leading issue in US media and

politics. The number of infected and dead was increasing rapidly and within a week California ordered the first state-wide lockdown. Hence “before” are polls conducted before March 15th, and “after” are later polls. The treatment group is made of Trump, as the pandemic occurred on his watch. Obama did not experience anything similar in 2016 and hence he forms the control group. When we compute the group/time period averages, we get an analogue to the Table 4.4:

Table 4.6: The effect of COVID-19 pandemic on president’s approval rate

Time	Approval rate (pct)		Difference (pct pt) (Trump - Obama)
	Control (Obama)	Treatment (Trump)	
Before (before March 15)	45.9	45.1	-0.86
After (after March 15)	48.1	46	-2.11
Trend (After–Before), pct pt	2.21	0.96	-1.25

We can see that the average approval rate for both presidents between mid-January and mid-March was fairly similar around 45% while Obama was enjoying a small lead of 0.86 pct. However, by end of March–early April Obama’s lead had increased to 2.11 pct points. The difference of these two figures is the effect estimate, –2.15 pct points. Alternatively, we can look at the growth of the popularity of both presidents over the same time period. During this time, Obama gained 2.21 pct points of approval while Trump 0.96 pct points. By construction, the difference is exactly the same number, –1.25.

We may depict this estimator graphically by plotting two lines, one for Obama and one for Trump, for two time points, “before” and “after”:

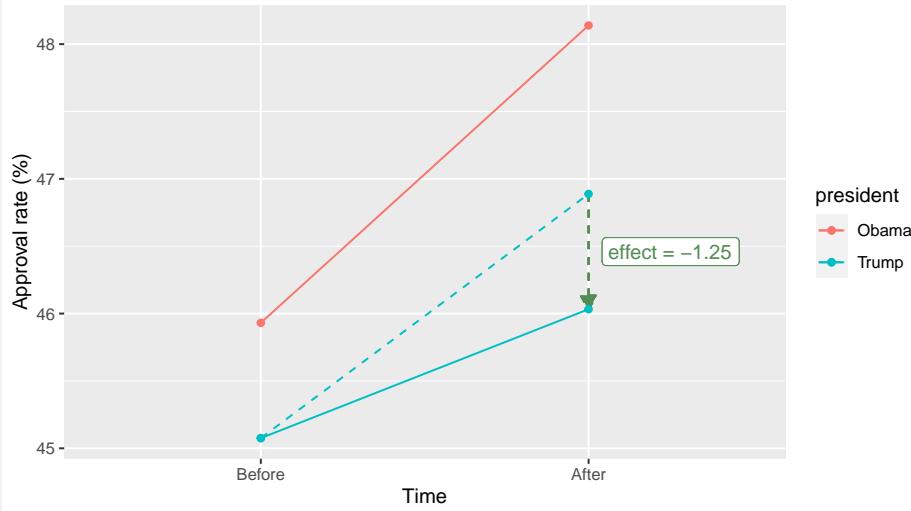


Figure 4.7: Presidents' average approval rate before and after March 15th of their 4th year in office. We can see that Obama's approval increased by more than two percentage points over this period while that of Trump grow by slightly less than one point. The dashed blue line depicts the counterfactual—the path of Trump approval rate, if it had been similar to that of the Obama's. The difference between the counterfactual and the actual approval (green dashed line), -1.25 points, is the effect.

This analysis suggests the epidemic was actually hurting the president's standing. Do we believe the result is correct? It is certainly plausible—unlike Korean president Jae In Moon for instance, Trump was not a leading figure in driving the nation's response to the virus. However, our belief should fundamentally depend on whether we believe in the identifying assumption: without the virus, Trump's approval in 2020 had followed a similar trend as that of Obama in 2016.

Table 4.4 treats the data as if we have just a single observation in each 4 cells of the table. But we may have more data, and we may have additional variables we may want to control for, for instance political preferences, age, place of residence, and other characteristics of the respondents. In this case we can use linear (or other type) regression instead of the tabulation. As in the examples with presidents' approval, we may observe multiple polls for both before and after period. Linear regression can easily capture the trend with a term $\beta_1 \cdot \text{after}$, and difference between the treatment and control groups by $\beta_2 \cdot \text{treatment}$. However, if we use these two terms only, we assume the trends are equal for both groups and hence by construction the effect is zero. So we also need an interaction term of the form $\beta_3 \cdot \text{after} \times \text{treatment}$ (see Section 3.1.9) to allow the trends between the groups. So the regression model will look like

$$y_{it} = \beta_0 + \beta_1 \cdot \text{after}_{it} + \beta_2 \cdot \text{treatment}_{it} + \beta_3 \cdot \text{after}_{it} \times \text{treatment}_{it} + \epsilon_{it}. \quad (4.5.19)$$

Here β_0 captures the baseline effect, the average outcome for the control group before treatment; β_1 captures the difference in the baseline trend, the outcome growth for the control group from “before” to “after”; β_2 captures the baseline difference between treatment and control groups (before treatment); and finally, β_3 is the estimated difference in time trends for the treatment and control group. The last figure, β_3 , is exactly the DiD estimate we are looking for. Hence, in order to estimate DiD using linear regression, you have to include:

- a) intercept β_0 ,
- b) a term for after-treatment time period: $\beta_1 \cdot \text{after}$,
- c) a term for the treatment group: $\beta_2 \cdot \text{treatment}$,
- d) an interaction effect for treatment group after the treatment: $\beta_3 \cdot \text{after} \times \text{treatment}$.

The latter is the estimate of interest. We may add additional controls as needed.

Example 4.14: President's approval rating: the regression approach

Let's return to the example of Obama's and Trump's approval rating. We select the time period from mid-January to mid-April of the fourth year of their presidency, as in Example 4.13. If fact, our dataset contains 149 polls for this period (See Table 4.5), so we have many observations for each table cell. We estimate

the following model:

$$y_{it} = \beta_0 + \beta_1 \cdot \text{after}_{it} + \beta_2 \cdot \text{Trump}_{it} + \beta_3 \cdot \text{after}_{it} \times \text{Trump}_{it} + \epsilon_{it}. \quad (4.5.20)$$

We get the following results:

.	object	..
Intercept	45.931	<i>0.433***</i>
after	2.208	<i>0.582***</i>
president Trump	-0.856	<i>0.539</i>
after×president Trump	-1.251	<i>0.785</i>
# obs	149	
R^2	0.2065	

Table 4.7: DiD regression estimate for the effect of COVID-19 epidemic on the US president's approval rating. Standard errors in italics.

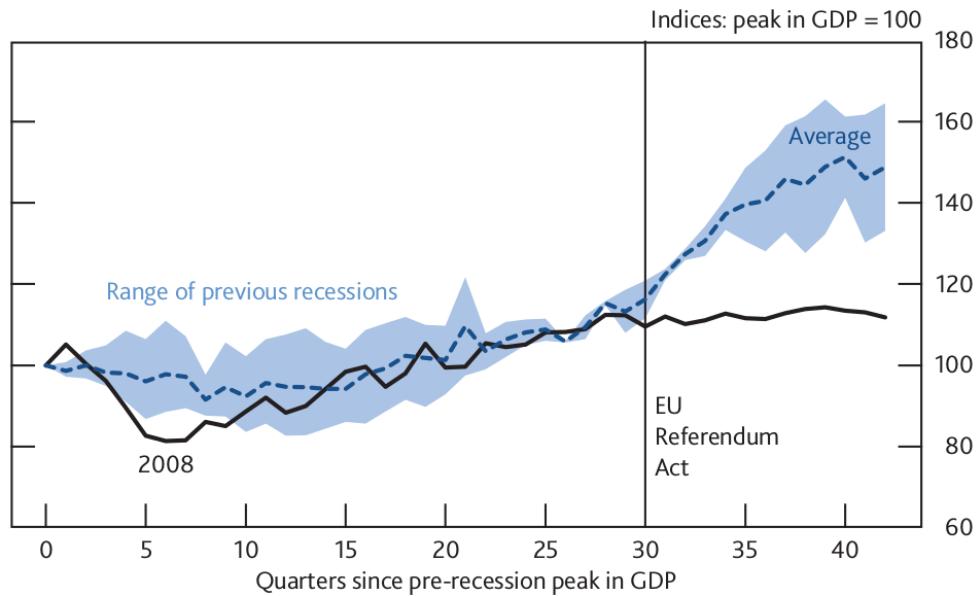
As expected, the regression approach gave us exactly the same numbers, including the main effect, $\text{after} \times \text{Trump} = -1.25$, as the table-based approach. Unless we introduce additional controls, linear regression just compares the averages. But unlike the table above, we now also have standard errors. These suggest that *after*, the spring-2016 trend for Obama, is indeed statistically significant. However, none of the Trump-related effects is statistically significant. The polling average for Trump is a little bit less than that for Obama, and his polling numbers have been lagging even more over the spring, but both effects are small and may well be a sampling noise. Hence, we can conclude that the epidemic did not give Trump any noticeably boost, and may instead have hurt him slightly.

To recap, let's list here all the predicted values:

- The baseline approval rate, for Obama, before mid-March, was $\beta_0 = 45.93\%$.
- For Trump, the approval rate was slightly lower, $\beta_0 + \beta_2 = 45.075\%$.
- After mid-March, Obama experienced a mild increase of $\beta_1 = 2.208$ pct points.
- After mid-March, Trump's growth was somewhat smaller than Obama's, $\beta_1 + \beta_3 = 0.957$ pct points, leading the average approval rate to $\beta_0 + \beta_1 + \beta_2 + \beta_3 = 46.032\%$.
- The main effect of interest here, the difference in springtime growth, is $\beta_3 = -1.251$ pct points. However, it is not statistically significant.

DiD estimators are sometimes used in a less formal context. Figure 4.8, taken from [England \(2019\)](#), p 14), compares the 2009 recession and the following recovery with “previous recessions”. One can see that while the previous recessions were followed by a substantial investment growth, this has not been the case since the Great Recession. After the Brexit referendum decision was made in 2015 (marked as the *EU Referendum Act* on the figure), the investment level has remained essentially flat. The authors conclude that “weak investment appears to primarily reflect Brexit and associated uncertainty”, a conclusion that also receives support from investor surveys.

Business investment after previous recessions^(a)



Sources: ONS and Bank calculations.

- (a) Chained-volume measure. Recessions are defined as at least two consecutive quarters of negative GDP growth. Previous recessions include those beginning in 1973, 1975, 1980 and 1990. A recovery ends if a second recession occurs in the period shown.

Figure 4.8: Effect on Brexit referendum on the business investments in UK: an example of graphical DiD approach.

Cheatsheet 4.1: OLS Estimators for causal inference

- T : treatment
- t : time
- \bar{y} : average outcome

Cross-sectional (CS) estimator

- **Control group:** other subjects
- **Identifying assumption:** the treatment group, if untreated, is similar to the control group (no unknown group differences)
- **Group average estimator:**

$$\beta_T = \bar{y}(T = 1) - \bar{y}(T = 0)$$

- **Regression model:**

$$y_i = \beta_0 + \beta_T \cdot T_i + \epsilon_i$$

Before-after (BA) estimator

- **Control group:** the same subjects before treatment
- **Identifying assumption:** the subjects, if left untreated, are similar to what they were before treatment (no unknown time trend)
- **Group average estimator:**

$$\beta_T = \bar{y}(t = 1) - \bar{y}(t = 0)$$

- **Regression model:**

$$y_{it} = \beta_0 + \beta_T \cdot After_{it} + \epsilon_{it}$$

Differences-in-differences (DiD) estimator

- **Control group:** the same subjects, but applying the time trend of the control group.
- **Identifying assumption:** the subjects, if left untreated, show similar trend as the control group subjects
- **Group average estimator:**

$$\begin{aligned} \beta_T = & \bar{y}(T = 1, t = 1) - \\ & - \left[\bar{y}(T = 1, t = 0) + (\bar{y}(T = 0, t = 1) - \bar{y}(T = 0, t = 0)) \right] \end{aligned}$$

- **Regression model:**

$$y_{it} = \beta_0 + \beta_1 \cdot After_{it} + \beta_2 \cdot Treatment_{it} + \beta_T \cdot After_{it} \times Treatment_{it} + \epsilon_{it}$$

4.6 Cognitive Illusions in Causal Inference

Humans brains are developed to serve us well in typical everyday situations we encountered through the past hundreds of thousands of years. However, accurately establishing causality based on observational or experimental data has apparently not been an important survival task for our ancestors. This manifests in cognitive biases related to causal inference.

Consider a binary treatment-binary outcome data, such as the flu shot-flu example in Section 4.3. This data can always be displayed as a four-cell contingency table (Table 4.8). It depicts four potential outcomes, for instance a denotes the count of cases where both the treatment and outcome are absent. This is typical data humans observe, it is much more rare to be able to directly manipulate the treatment in order to conduct something that resembles a RCT. Evolution has taught us to deduce causality from such case counts.

Table 4.8: Four potential outcomes in binary treatment/binary outcome data. “0” and “1” denote presence and absence of treatment and outcome, the letters in cells are the corresponding case counts.

Treatment	Outcome	
	0	1
0	a	b
1	c	d

We are inclined to believe “treatment” causes “outcome” if we see many cases in cells a and d while c and b remain relatively empty, and there are no obvious confounding factors. It should be clear to the reader by now that such data alone is not enough to establish causality. However, very often this is all we have, and we have to use such information to successfully live in the environment we live in. Remember—we are talking about a time frame of hundreds of thousands of years, most of which our ancestors spent as hunter-gatherers in African savannas.

It turns out that humans are more likely to believe treatment causes outcome ([Matute et al., 2015](#)) if

1. The outcome is very likely, say, 75% or more. This is called *outcome-density bias*.
2. The treatment is very likely (*cause-density bias*).

Both biases are related to the cell d in the table being well populated compared to the other cells, and hence reinforce each other. People are likely to believe in bogus causal relationship if both treatment and outcome are very likely, for instance when they take a homeopathic pill every few hours while the ailment goes away rapidly. A simple remedy to counter this bias is to recommend people to lower the frequency of treatment. It makes the d -cell case count smaller and hence the bias will be less strong.

4.7 Causality and complex social problems

Sometimes the causal chains are much more complex and harder to predict than is apparent when someone first encounters the problem. This is often the case if the question are related to humans and social problems. Below we walk through an example, namely usage of mandatory bicycle helmet laws.

4.7.1 Effect of bike helmet laws

It is well established that bicycle helmets substantially reduce head injuries during certain type of crashes. Is this evidence enough to justify mandatory helmet laws (MHL-s)? It turns out we need much more evidence.

The fact that helmets help to prevent head injuries is best established through mechanical experiments where model heads, with and without helmet, are dropped to hard surface. One can find information for both about frontal impact ([Cripton et al., 2014](#)) and for oblique impact ([Mills and Gilchrist, 2008](#)). In such experiments the researchers have full control over the environment, such as impact speed, type of helmet, hair and skin properties and so on, and in this sense they answer the exact question they are designed for very well. The question in the above-cited papers is about head injury when hitting hard surface at given speed.

Obviously, head injury is not a random process that only depends on the presence of helmet—there are many more decisions involved. For a start, one has to decide whether to cycle or not. Thereafter one chooses the route, speed, and makes other decisions about biking like distance from curb, whether to pass someone, etc. Also the other road users choose their behavior, such as motorists must decide speed and distance when passing a cyclist. So there are many reasons to believe that such studies do not give the complete picture.

1. Typical crashes occur at different angles and surfaces, and not necessarily in conditions similar to that of the laboratory environment. But as the experiments get better, we can assume the laboratory models get increasingly close to the real cases.
2. in certain circumstances the helmet may get stuck and hurt the wearer more than would be the case without helmet (rotational injuries). So far, the non-experimental evidence from actual accidents tends to indicate that helmets help to prevent head injuries by a substantial degree ([Amoros et al., 2012](#)), so the cases where helmets hurt are probably rare in practice. However, we are outside of controlled experiment realm now.
3. cyclists may act differently depending on whether they wear or do not wear helmets. In particular, wearing helmet can lead to more risky behavior (risk compensation). [Fyhri et al. \(2018\)](#) does not find any effect of wearing helmet on cyclists' speed in a field experiment. However, the study was limited in terms of number of participants (31) and situations encountered on the road. More research is needed here but it is much harder to simulate realistic situations here.

4. MHL-s may discourage cycling. As crashes are rare, the net health effect may be negative if, in order to avoid rare crashes, people avoid the healthy exercise in the first place. However, we may debate if authorities should strive toward fewer crashes, or more healthy population.

The discouragement may occur through several mechanisms:

- (a) helmets are considered inconvenient, either to wear, or to carry around
- (b) the authorities are using scaring tactics to make the point for helmets.
This may make people afraid of biking in first place instead of choosing helmets.
- (c) helmet requirement makes bike shares less harder to implement.

These effects are very hard to pinpoint in experiments.

5. If MHL discourages cycling, it also makes cycling less safe through following mechanisms:

- (a) “safety in numbers”: the less bikes there are on street, the less the motorists expect to encounter them, the less prepared they are to notice cyclists in traffic and hence the more likely are the accidents.
- (b) some people may choose driving over cycling increasing the amount of motorized traffic.
- (c) fewer cyclists also means less political will to invest in cycling infrastructure.

6. helmets may also cause drivers to behave differently and behave more (or less) risky with respect to cyclists. For instance, [Walker \(2007\)](#) finds that cars passed cyclists with helmets significantly closer in average.

Note that despite of the large number of potential mechanisms, the net effect may be dominated by just one or two major ones while all the others are of very little importance. The problem is that we don't know how strong are each of these effects, and hence the policy will remain largely uninformed.

Chapter 5

Predictive modeling and model goodness

Contents

5.1	Predictive modeling	231
5.2	Categorization	231
5.2.1	Confusion matrix and related concepts	232
5.3	Overfitting and Validation	245
5.3.1	What is overfitting	245
5.3.2	Validation: which model is the best	249
5.3.3	Cross-validation	250
5.3.4	Training-validation-testing approach	250

5.1 Predictive modeling

TBD:

5.2 Categorization

We discussed model evaluation in the context of linear regression above in Section 3.1.5, including how figures like $RMSE$ and R^2 describe different sides of the model performance. Here we focus on evaluating categorization.

It turns out that $RMSE$ and R^2 are not appropriate indicators for model goodness in case of categorization. In case of continuous outcomes, such as income or intensity of light, a good model will be the one that predicts values very close to the true observed ones. Hence the measure should be based on the difference between the predicted and actual value, $\hat{y}_i - y_i$. But this approach does not really work for categorization for several reasons:

Nominal measures: can only be compared as equal/not equal; ordinal measures: can only be compared as equal, larger, smaller. See see [Section 1.1.1](#).

1. First, categories are nominal or ordinal measures and hence the difference $\hat{y}_i - y_i$ is typically not defined. For instance, when predicting treatment status then the whole concept *treated – nontreated* does not make much sense.
2. Second, our predictions can be wrong in two ways: either we predict 1 instead of 0 (false positives) or 0 instead of 1 (false negatives). Neither *RMSE* nor R^2 distinguishes between these types of errors. There are more possible errors if we have more categories.
3. Finally, even if we define the difference $\hat{y}_i - y_i$, e.g. as 0 if our prediction is correct and 1 if it is not correct, we have lost all information about how “far” off the prediction was from the correct one.

Solutions to the first two issues are based on confusion matrix. In order to address the third issues, we have to look not just the predicted categories but the predicted probabilities of the categories.

5.2.1 Confusion matrix and related concepts

Confusion matrix is a popular way to assess the performance of categorical models. Instead of attempting to measure distance between the predicted and the true values, we just tabulate and count all types classification errors. This simple approach allows to avoid the first and second problem listed above.

Confusion matrix

Confusion matrix is in essence just a cross-tabulation of the actual and predicted classes. It is a central tool that many categorization-related goodness measures are based on. Here we discuss confusion matrix in case of two categories only but it easily generalizes to a larger number of classes.

Let’s start with an example. Imagine we work in a hospital and have ten patients who all do a medical test. This is a quick test that shows if the patient has a certain condition, such as asthma. As is the tradition in medicine, we call the test “positive” (+) if the patient has asthma, and “negative” (-), if they have not. But the test is imprecise, and only over time we will learn the actual value. The actual and test values of all patients are in Table 5.1 (left panel). The last column in the table, *Type*, shows the correctness of the test results: *TP* (true positives) are patients who had asthma and were tested positive, *TN* are patients who do not asthma and were tested negative. These are the correct results. But in a number of cases, the test was wrong: *FN* (false negatives) are asthma cases that were tested negative and *FP* are the opposite, healthy patients who were tested positive. As the test can only have two possible outcomes, positive and negative, these four types are the only possible correctness results.

The right panel shows a summary table of the table at left, just the counts of all four possible types. In these data we have two true negatives, one false positive, three false negatives and four true positives. This is the idea of a confusion matrix. Next, let’s discuss it in a more formal fashion.

Table 5.1: Example diagnosis data (left) and the corresponding confusion matrix (right).

Case#	Actual	Test	Type		Actual	Test		
					-	+	Total	
1	+	+	TP			2	1	3
2	-	-	TN			3	4	7
3	+	-	FN					
4	-	-	TN					
5	-	+	FP					
6	+	+	TP					
7	+	-	FN					
8	+	-	FN					
9	+	+	TP					
10	+	+	TP					
				Total	5	5	10	

Assume we have in total T cases from two categories: P positives denoted by “+”, and N negatives denoted by “-”. This can be a similar medical diagnosis problem as in the example above, but may also be something completely different. For instance, we in case of weather forecast, we can label rain as positive and dry days as negative. These are the “actual categories”, determined either through expensive testing or diagnosis, or maybe the correct results will be apparent over time (as in case of weather forecast). But we know that the actual categories are correct. Now we use a model to predict the category for each case. We would like the model to predict every single case correctly as positive or negative but most models are not that good. Let’s say that in total, the model predicts \hat{P} cases as positive and \hat{N} cases as negative. For an overview, we create a similar 2×2 cross-table as above, where we present the counts for actual and predicted classes (Table 5.2). The table indicates how many actual positive cases were predicted as positive, how many as negative, and so on. This is confusion matrix.

Table 5.2: Confusion matrix for two categories, labeled here as “-” and “+”. The table entries are counts: TP, true positives, refers to positive cases that were also predicted to be positive, P is the number of actual positive cases. See explanations in the text.

Actual	Predicted		
	-	+	Total
-	TN	FP	N
+	FN	TP	P
Total	\hat{N}	\hat{P}	T

In case of two categories, the core of confusion matrix contains four cells:

- *True positives (TP)* are cases that are actually positive, and are correctly predicted as positive. We like TP to be large.

- *True negatives (TN)* are actually negative and are predicted as negative. We like TN to be large.
- *False positives (FP)*, also *type-I errors*, are cases that are actually negative but were incorrectly predicted as positive. We would like FP to be zero.
- *False negatives (FN)*, also *type-II errors*, are cases that are actually positive but were predicted as negative. We would like FN to be zero.

In case of confusion matrix, these concepts often refer to the corresponding counts, e.g. FP is the number of cases we incorrectly predict as positive. However, these may also refer to probabilities or percentages, e.g. FP may be a probability that we predict a case incorrectly as positive, or percentage of such cases. Obviously, in case of a good model we have high values of TP and TN while the counts of FP and FN are small. Table 5.2 also includes one-way counts: P is the number of actual positives, N is the number of actual negatives, \hat{P} is the number of predicted positives and \hat{N} is that of predicted negatives. Finally, T denotes the total number of cases.

Example 5.1: Confusion Matrix

Dataset *Treatment* contains information about individual participation in a labor market training program, and background information, such as age, previous unemployment, and income. Here we use that information to estimate a logistic regression model to predict the participation status based on age, previous real income and previous unemployment:

$$\Pr(\text{Participated}_i) = \Lambda(\beta_a \cdot \text{age}_i + \beta_{r75} \cdot \mathbb{1}(re75_i > 0) + \beta_{u75} \cdot u75_i)$$

The original data has 185 participants out of 2675 individuals in total, while our model predicts 134 as participants and 2541 as non-participants. When we create confusion matrix, a cross-table of actual and predicted values, the results looks like this:

Actual	Predicted		Total
	Non-Participants	Participants	
Non-Participants	2452	38	2490
Participants	89	96	185
Total	2541	134	2675

Let's consider participants as positives below. So for $TN = 2452$ individuals, our model correctly predicts that they did not participate in the program. For an additional $TP = 96$ cases it correctly predicted that they participated. TN is rather large, these are good news for our model. But unfortunately TP is not much larger than $FN = 89$, the number of individuals who participated but were incorrectly predicted as non-participants. Finally, the count of type-1 errors, false positives, is smaller, $FP = 38$, indicating that the model does not mis-categorize many non-participants as participants.

Although a 2×2 table seems simple, confusion matrix is actually surprisingly confusing. So it is not surprising it is called *confusion* matrix \circledcirc . It is partly related to the notation and language. In particular, *true positives* refer to cases that are actually positive, and are predicted as positive; not to the “ground truth”, the cases that are actually positive as one may think. This is why we introduce the “actual” status here, to distinguish between the actual positives P and “true” positives TP . In addition, N typically denotes the total number of cases, not just the number of actual negatives. Here we denote the total number of cases by T .

Moreover, you can see the confusion matrix defined in slightly different way in the literature, e.g. putting actual values in columns and predictions in rows, and putting positives first and negatives second. So each time you see a confusion matrix in the literature, you need to understand how exactly it is defined. All these definitions are correct, but mixing them up is not! Here we consistently use the definition above: actual values in rows, predicted values in columns; negatives first and positives second.

Exercise 5.1: Compute the confusion matrix

Consider a variable that can be of two categories: “0” and “1”. First, you ask an expert for her opinion, and later the actual values also become evident. The values are as follows:

case:	1	2	3	4	5	6	7	8	9	10
Actual	1	0	0	1	1	0	0	0	1	0
Expert	0	0	0	1	0	0	1	0	1	1

Construct the confusion matrix.

Solution on page [427](#).

Exercise 5.2: Confusion matrix for the naive model

Consider the data in Example 5.1. Let us construct a naive model that predicts all observations to the majority category—to the category that is more common (non-participants in that case). How will the corresponding confusion matrix look like, if you consider the participants as positives?

Solution on page [427](#).

Based on these numbers, we define a number of model goodness measures:

- *Accuracy*: percentage of correct answers

$$\text{Accuracy} = \frac{TP + TN}{T}$$

Accuracy is an easy and intuitive summary measure: what percentage of our predictions turn out to be correct. However, it is not very informative in case of very inequally sized categories as even a naive model that always predicts the most common class can achieve high accuracy (see Exercise 5.2 and Example 5.2).

- *Recall* is the percentage of actual positives that are correctly identified (recalled) as positives

$$\text{Recall} = \frac{TP}{P} = \frac{TP}{TP + FN}.$$

If our main concern is to capture all positives, recall may be a good measure. However, it is easy to fool: if we predict every case to be positive, then we get $\text{Recall} = 1$ but the model is hardly of any use.

- *Precision* is a sort of mirror image of recall: percentage of predicted positives that turn out to be correct.

$$\text{Precision} = \frac{TP}{\hat{P}} = \frac{TP}{TP + FP}.$$

Precision may be a good measure if avoiding false positives is a major concern. As the other measures, this can also be fooled easily: if we ensure that only the most likely cases are labelled as positive, we can ensure that precision is high.

- *F score* is an attempt to find a balance between recall and precision. It is just the harmonic mean of these measures

$$F = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}}$$

Example 5.2: Accuracy, Precision, Recall, F-score

Consider the confusion matrix in 5.1. From the matrix we can compute all four model performance measures:

$$\text{Accuracy } A = \frac{TP + TN}{T} = \frac{96 + 2452}{2675} = 0.953 \quad (5.2.1)$$

$$\text{Recall } R = \frac{TP}{P} = \frac{96}{185} = 0.519 \quad (5.2.2)$$

$$\text{Precision } P = \frac{TP}{\hat{P}} = \frac{96}{134} = 0.716 \quad (5.2.3)$$

$$\text{F-score } F = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} = \frac{2}{\frac{1}{0.716} + \frac{1}{0.519}} = 0.602 \quad (5.2.4)$$

So our model is highly accurate but not impressive in terms of recall and precision. This is because the groups are of very different size: only 185 (6.9 percent) of individuals participated in the program, and hence if we would predict “non-participant” for everyone, we would still get accuracy 93.1 percent. So despite of the impressive accuracy, the model does not do actually much better than a naive guess. $R = 0.519$ tells that we only catch slightly over 50% of the participants correctly, and $P = 0.716$ shows that only around 70% of predicted participants are correct. Finally, F -score is predictably between R and P .

Exercise 5.3: Accuracy, Precision, Recall

Look at categorizing cases into two color categories: Red and Yellow. Consider the confusion matrix:

Actual	Predicted	
	Red	Yellow
Red	10	20
Yellow	10	60

Assuming Yellow is the positive, compute A , P , R and F -score.

[Solution](#) on page 428.

These model goodness measures have multiple names, and they are closely related to other similar measures. A number of examples are listed here:

- Accuracy-based measures

Misclassification rate is just the opposite of accuracy:

$$\text{Misclassification rate} = 1 - A \quad (5.2.5)$$

- Recall-based measures

True positive rate and **sensitivity** are just another names for recall

Specificity is recall for negative outcomes:

$$\text{Specificity} = \frac{TN}{N} = \frac{TN}{TN + FP} \quad (5.2.6)$$

False positive rate measures the percentage of negatives that is falsely categorized as positives. It is also 1 minus specificity:

$$\text{False positive rate} = 1 - \text{specificity} = \frac{FP}{N} = \frac{FP}{TN + FP} \quad (5.2.7)$$

False negative rate is the opposite of FPR , it measures the percentage of positives that are falsely categorized as negatives:

$$\text{False negative rate} = \frac{FN}{P} = \frac{FN}{TP + FN} = 1 - \text{Recall} \quad (5.2.8)$$

Note that for each model we can only compute a single accuracy measure but two P and R measures: one for positive and one for negative cases (and even more if we have more than two categories). It is often clear from the problem which cases we should analyze and which measures we should focus. For instance, in case of medical diagnosis, the “positive” typically means the illness, and we may be concerned about catching as many cases as possible (we need a high recall). Alternatively, if the treatment is expensive and potentially harmful, we may be interested to ensure all cases we identify are actually correct (we look for high precision).

Exercise 5.4: Flipping positives and negatives

Consider the treatment data in Example 5.1 (the same as in Exercise 5.2).

- Assume participants are positives. Construct the confusion matrix and compute accuracy, precision and recall.
- Assume non-participants are positives. Construct the confusion matrix and compute accuracy, precision and recall.
- What do you think, which of these options is better?

Solution on page [428](#).

Exercise 5.5: COVID test sensitivity

[Ferté et al. \(2021\)](#) analyze specificity and sensitivity of rapid covid tests (Abbott Panbio SARS-CoV-2 Ag rapid test) on students at Bordeaux University. They find specificity to be 100% and sensitivity 63.5% in the overall population, in the asymptomatic group the numbers are 100% and 35%.

Construct example confusion matrices that have corresponding sensitivity and specificity. What do you think about the quality of the test?

Solution on page [428](#).

ROC curve

Categorization models normally do not just predict the class, but the *probability* that the observation belongs to each class. It is customary to take probability 0.5 as the threshold between the categories. If the predicted probability is less than the threshold, it belongs to one, if it is above the threshold, it belongs to the other category. Say, probability 0.25 corresponds to “spam” and 0.6 to “no-spam” category.

While value 0.5 is intuitive and exactly in the middle of the range, we do not have to pick this value. If different type errors are associated with different costs, we may be much more willing to err in one side than another and pick a threshold noticeably different from 0.5. For instance, a judge may consider 0.9 as a too low confidence to sentence someone for a felony, because such decision, if wrong, will have serious consequences for the defendant. Obviously, our predictions change if we pick a different threshold, and different models may show different behavior here. ROC curve (*receiver operating characteristics*) is a way to make the type-I/type-II error trade-off explicit, and help the user to choose between different models and thresholds.

ROC curve is also based on confusion matrix related concepts, true positive rate, *TPR*, (i.e. recall) and *false positive rate*, *FPR*, defined as

$$\text{False positive rate} = \frac{FP}{N}.$$

While *TPR* measures the percentage actual positives that are identified correctly, *FPR* measures the percentage of actual negatives, incorrectly identified as positives. Obviously, we want our model to show high TPR (ideally 1) and low FPR (ideally 0).

ROC curve makes these tradeoffs explicit. It plots TPR against FPR for different thresholds. A typical ROC curve is shown in Figure 5.1. The figure shows to mod-

els, linear probability model and logistic regression, addressing labor market training participation as a function of age, experience unemployment, and other individual characteristics. Typically all models offer two extreme choices: $FPR = TPR = 0$ and $FPR = TPR = 1$. The first corresponds to the case where all observations are predicted to be negative, the other to the cases where these are predicted to be positive. (Here it is not the case for LPM as the predicted probabilities may be outside $[0,1]$ interval.) Obviously, we are interested in the cases in the middle where FPR is low but TPR approaches to one. The figure suggests that logistic regression clearly outperforms LPM at low FPR values. For instance, if $FPR = 0.1$, TPR for LPM is approximately 0.9, but for logit 0.95.

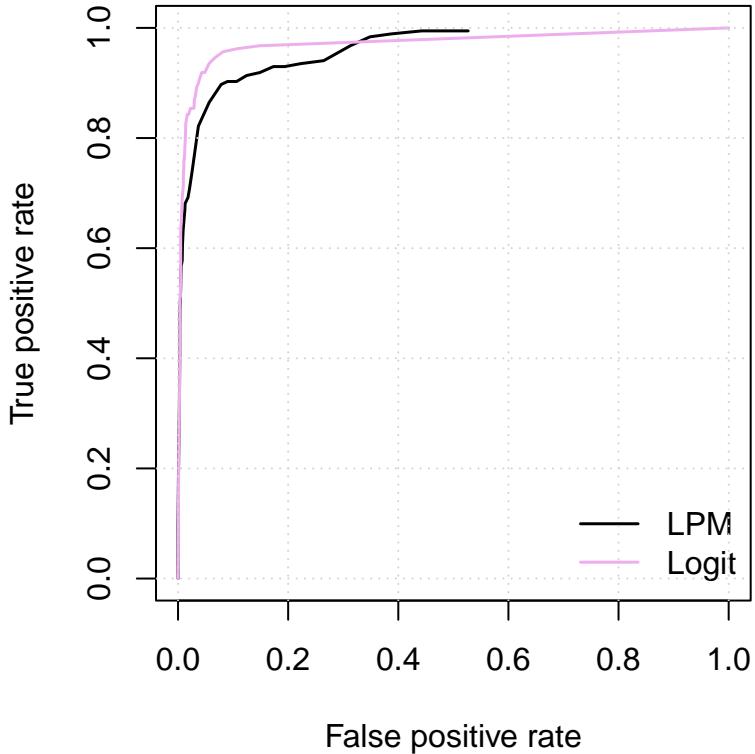


Figure 5.1: Example ROC curve for linear probability model (black) and logistic regression (pink). The figure suggest that in most cases logit outperforms LPM as it is able to achieve higher TPR over TPR range 0 to 0.3.

Example 5.3: Computing ROC curve

Let's look at a case where we have two classes: "0" (negative) and "1" (positive). We are working with four cases only. We run our model and the algorithm predicts the following probabilities:

case	Pr(positive)	true value
1	0.3	0
2	0.4	1
3	0.6	0
4	0.7	1

Denote the probability threshold value by θ , i.e. if $\text{Pr}(\text{positive}) > \theta$, we predict the case to be positive, otherwise it is assigned to the negative category. If we pick $\theta = 0.5$, our algorithm predicts the cases 3,4 to be positive and 1,2 to be negative. This is the most intuitive approach, but may not be the best if type-I/type-II errors have very different price.

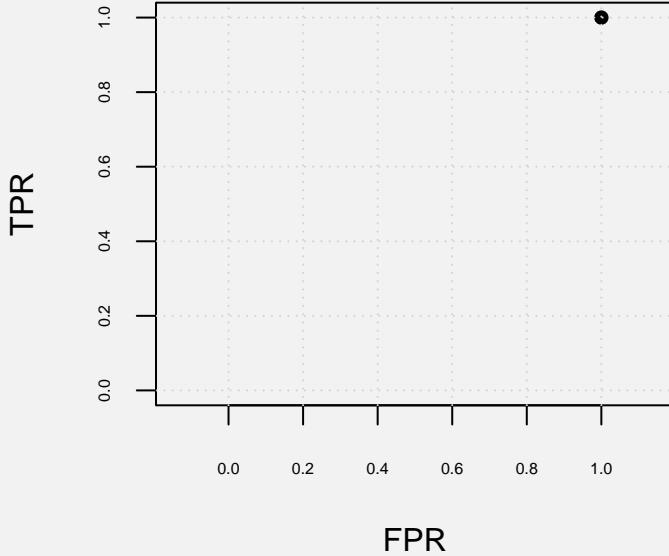
ROC curve is what makes these tradeoffs explicit. Let's start with an extreme threshold, $\theta = 0$. Now the algorithm predicts everything to be positive^a and the confusion matrix will look like

		Predicted	
		1	0
Actual	1	2	0
	0	2	0

Note that here $P = 2$ and $N = 2$. Obviously, we get all the actual positives, but we get all actual negatives wrong. This corresponds to the true positive and false positive rates

$$\text{TPR} = \frac{TP}{P} = \frac{2}{2} = 1 \quad \text{FPR} = \frac{FP}{N} = \frac{2}{2} = 1 \quad (5.2.9)$$

and will give us a data point on the ROC curve:



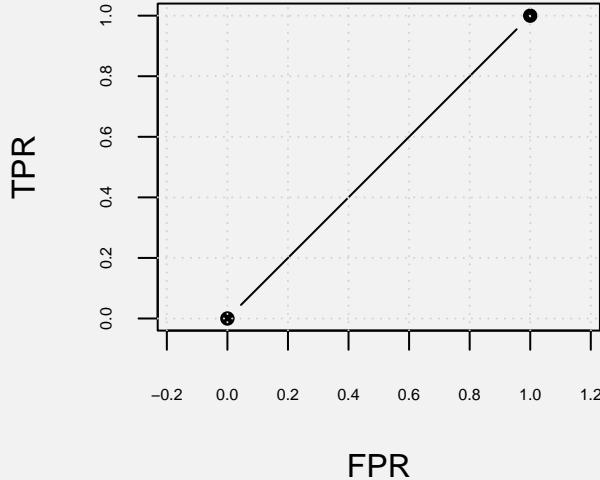
On the other extreme, we can take threshold $\theta = 1$. Now none of the cases is predicted positive and the confusion matrix will be

		Predicted	
		1	0
Actual	1	0	2
	0	0	2

All the negatives are correct but all positives are wrong, and the true positive and false positive rates are accordingly

$$\text{TPR} = \frac{2}{2} = 0 \quad \text{FPR} = \frac{2}{2} = 0. \quad (5.2.10)$$

This will correspond to the lower-left corner of the ROC curve:



Note that now the ROC curve is already a curve, not just a single point. Such two extreme cases are always possible with a naive model that predicts either all cases positive or negative, and hence do not tell us much about the underlying model.

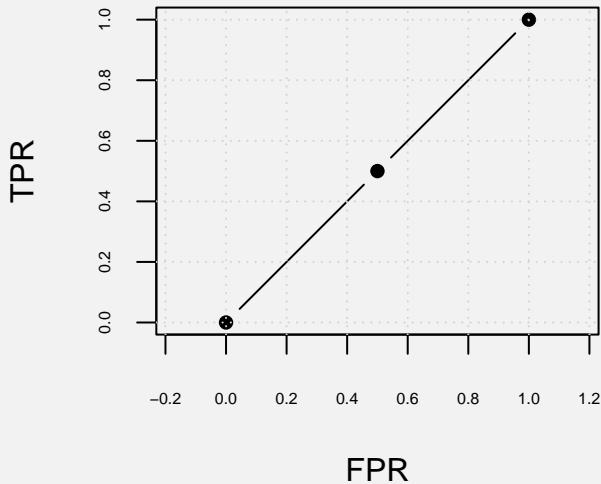
As a third example, take the threshold $\theta = 0.5$ and hence cases 1 and 2 will be predicted negative and cases 3, 4 positive. The confusion matrix is

		Predicted	
		1	0
Actual	1	1	1
	0	1	1

Now half of the predictions are correct and a half are wrong:

$$\text{TPR} = \frac{1}{2} = 0.5 \quad \text{FPR} = \frac{1}{2} = 0.5. \quad (5.2.11)$$

We get another point in the middle of the same ROC curve:



As above, we got a point on the same line that denotes random outcomes. This indicates that our model performs exactly as good as a naive model that randomly predicts half of the cases negative and the other half positive.

^aIt predicts literally *everything* to be positive only for such models, like logistic regression, where predicted probabilities are in the interval (0,1). This may not be the case for e.g. *k*-NN (predicted probability may be 0) or for LPM (predicted probability may be negative).

Limitations of the confusion matrix approach

While confusion matrix offers us a large number of intuitive indicators for the model performance, it is oblivious about the confidence of our estimators. As long as the predictions do not change, the increased confidence in the predictions is not reflected in the results. This makes it hard to compare models on small datasets as small changes in categorization results may obscure more important underlying confidence effects.

Cheatsheet 5.1: Confusion matrix and related measures

Confusion matrix:

Actual	Predicted			Total
	-	+		
-	TN	FP		N
+	FN	TP		P
Total	\hat{N}	\hat{P}		T

where

TP true positives

P actual positives

TN true negatives

N actual negatives

FP false positives

\hat{P} predicted negatives

FN false negatives

\hat{N} predicted negatives

T total cases

Model goodness measures:

Accuracy percentage of correct predictions $A = \frac{TN+TP}{T}$

Precision percentage of predicted positives that are correct $Pr = \frac{TP+FP}{\hat{P}}$

Recall percentage of actual positives detected $R = \frac{TP+FN}{P}$

F-score balanced mean of Pr and R : $F = \frac{2}{\frac{1}{Pr} + \frac{1}{R}}$

True positive rate same as recall

False positive rate percentage of negatives that are predicted incorrectly as positives $FPR = \frac{FP}{N}$

5.3 Overfitting and Validation

Prerequisites: [Section 3.1 Linear Regression](#), page 123

5.3.1 What is overfitting

When working with machine learning models, we typically start with the “learning” part, i.e. model training. Training makes the model to “learn” patterns in data (typically by computing certain parameters) and later we can use the same patterns for predictions. But model can only learn about patterns it actually sees, i.e. patterns that are there in data we are using for training (called *training data*). Later, when we use the model for making predictions, we typically want to make predictions for values that are not in training data. This is the typical workflow for supervised learning.

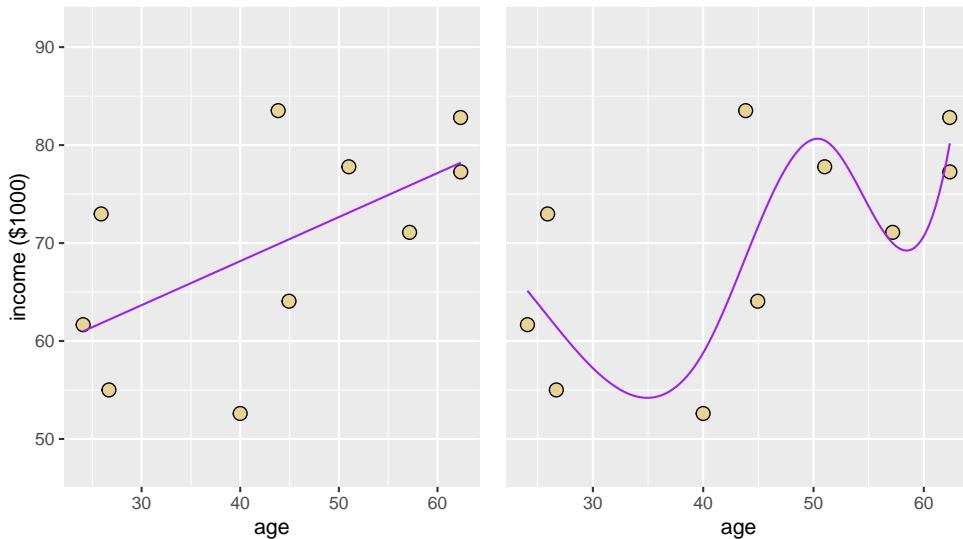


Figure 5.2: Two possible patterns to explain the same data

However, it turns out that powerful and flexible models may learn patterns that are not useful. Consider Figure 5.2. It depicts a fictional income-age relationship of $N = 10$ individuals, depicted as golden dots. The left and the right hand panels show two possible patterns—a simple linear trend (linear regression) at left, and a more complex wavy pattern at right. Which of these patterns is the “correct” representation of data? Just visual inspection suggests that a line may be good enough, but the complex curve at right is definitely getting closer to the dots. But be careful here. We do not want to capture *data* (patterns in this particular sample). What we want to do is to use data to learn the patterns in the underlying RV where the data is sampled from. In this case, we are not very much interested in the relationship in

this sample (you can see it on the figure anyway) but in this society where the data is coming from.

The problem is that the flexible curve may be too much taylor-made for this particular sample. It captures not so much the trends in the underlying RV but those in this particular sample. When we get another sample then a too closely targeted pattern may not hold any more. This typically leads to plummeting performance when predicting new data points. The model performs unrealistically well on training data, but on everything else it is mediocre at best.

This is often not a major concern as many simple models, such as linear and logistic regression. After all, it is hard to argue that the line at left on Figure 5.2 is capturing data too well. But if some regions of data space are not well represented in training data, then more flexible models may produce estimations that are wildly off. A less flexible model may, in contrast, still provide meaningful predictions. This phenomenon is referred as *overfitting*.¹

Overfitting is a pervasive problem in most ML models, and more so in more flexible model. As linear regression is rather rigid (we describe the relationship as a hyperplane), it is less of a problem here, but that does not mean regression models are immune to overfitting. Below we provide two artificial examples of overfitting in linear regression context.

Consider the example in Figure 5.3. The left panel shows a fictitious dataset that represents how income depends on age. We can see an upward trending relationship. The right panel shows a number of different polynomial regression models, and their corresponding predictions using the same data. All models contain polynomials of age in the form²

$$y_i = \beta_0 + \beta_1 \cdot \text{age}_i + \beta_2 \cdot \text{age}_i^2 + \beta_3 \cdot \text{age}_i^3 + \dots + \epsilon_i. \quad (5.3.1)$$

The first model, of degree 0, contains just the constant term β_0 , essentially assuming that income is independent of age. Degree 1 is the linear model $y_i = \beta_0 + \beta_1 \cdot \text{age}_i + \epsilon_i$, degree 2 is a quadratic relationship $y_i = \beta_0 + \beta_1 \cdot \text{age}_i + \beta_2 \cdot \text{age}_i^2 + \epsilon_i$, and so on. The higher the polynomial degree, the more flexible the model—the more complex curves it can represent. This is because higher degree models contain more base functions, and by combining more functions we are able to approximate data better.

Which model is the best one? Just by looking at the image, one may suggest that degree 0 (horizontal red line) is too inflexible, data seem to follow an increasing trend and a constant does not capture it. Both linear and quadratic model (degree 1 and 2) both capture the trend well. The linear (degree 1) model obviously shows a constant trend while the quadratic model also captures the steady increase of trend after age 35. The 4th-degree polynomial seems somewhat too wobbly, and the 7th-degree model fluctuates even more. The most flexible model displayed here, the 9th-degree polynomial, has gone completely wild and jumps up and down way outside of what fits to the image the image. But despite of its wild behavior, it manages to capture all

¹In a high-dimensional feature space the datapoints are always sparse, and there are always large uncovered regions.

²These example models are created using orthogonal polynomials, not just powers of age. In case of ordinary polynomials, high-order *age* terms will introduce a lot of multicollinearity and the numeric precision of calculations will be insufficient. These issues are not related to overfitting.

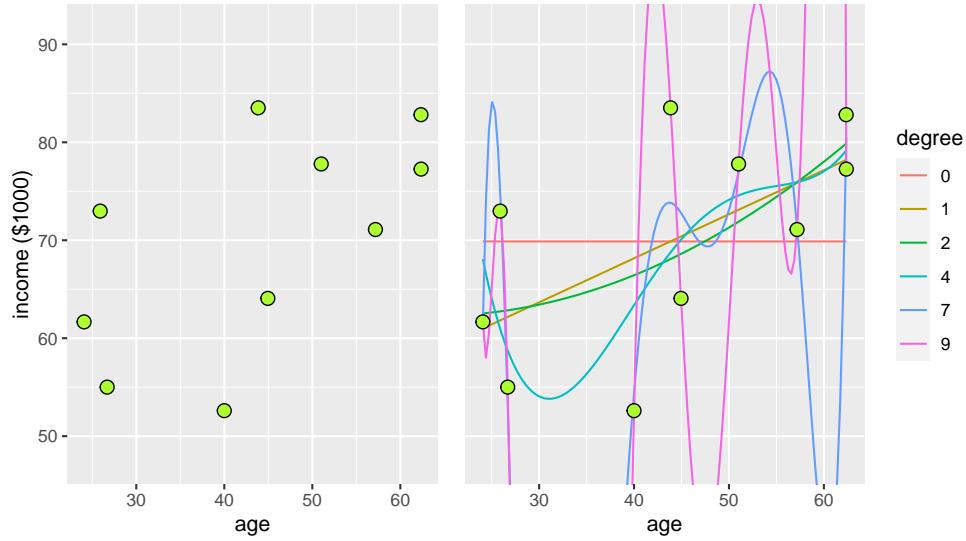


Figure 5.3: Artificial age-income data. Left panel shows just the data points, the right panel displays the same data and a number of polynomial regression models with various polynomial degrees.

data points *exactly!* This is a general pattern—the higher the degree, the closer the model to the actual data points. This can be confirmed by computing the corresponding $RMSE$ -s or R^2 -s:

degree	0	1	2	4	7	9
RMSE	10.50	8.43	8.31	7.87	4.50	0.00
R^2	0.10	0.42	0.44	0.49	0.83	1.00

This is easy to understand intuitively: a more flexible model is more able to fit the actual data points. You can imagine the polynomial degree as some sort of inverse “rigidity”, where higher degree means more flexible line. 9th-degree polynomial can fit 10 data points perfectly. But the price is paid in the gaps between data points. There is nothing that limits the model’s wobbles in those gaps, and hence flexible models can jump wildly. (In case of less flexible models, it is the “rigidity” of the curve that does not let it to jump too much.) But what matters more—the better precision at the data points where we know the answer, or unrealistic behaviour between those data points? And why do we claim that the behaviour inbetween the known data points is unrealistic if we, per definition, do not know what is going on in those regions? Note that often we are interested in model performance exactly in the gaps—after all, there is little need to make predictions where we already know the answer.

But in order to formally evaluate the effect of the apparent misbehavior between data points, we have to know more. In case of this example, the 9th-degree model predictions for a 57-year old seem completely out of touch with the reality, suggesting

the income at that age is deep in negative territory. Nothing we know about human lifecycle suggests this is the case. A simple linear (degree-1) model feels much more appropriate here, suggesting income around \$75,000. But in other applications we may want to trust the more flexible model instead. And often we just have no idea what to expect. How can we still get a good idea of which models is the best?

Example 5.4: Overfitting in case of categorization

Consider a two-dimensional example in Figure 5.4. The figure depicts a categorization problem where the coordinate pairs (x_1, x_2) are classified as red or blue. The dots are the observations we know, they seem to indicate that the lower left of the figure is red-dominated, and the upper right part is blue-dominated. The decision boundary between these two areas follows a wavy pattern.

The pale red and blue background are our predictions—the model predicts that all dots in the red region are red and blue region are blue. The left panel does the prediction using a logistic regression. This results in a simple linear decision boundary. This seems to be a too rigid approach, it does not capture the blue and red waves (the yin-yang pattern) that is clearly systematic and not just random noise, and that trespasses over the decision boundary line. The right panel uses single nearest neighbor to predict the colors. It results in a complex elaborate boundary that carves out every single red and blue dot. This seems to be a too complex boundary and suggests we are overfitting.

Nearest neighbors: each dot is predicted to be of the color as the nearest known colored dot. See more in [Section 6.2 k-Nearest Neighbors](#), page 263.

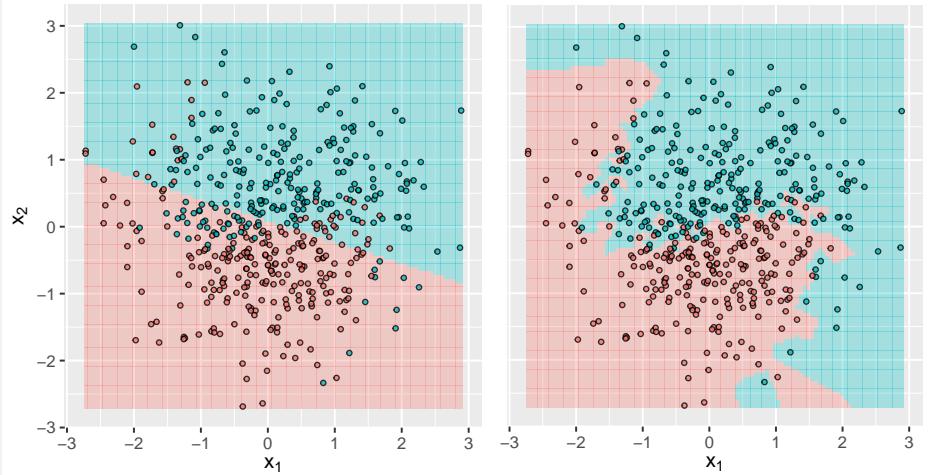


Figure 5.4: Categorization task with blue and red dots. Logistic regression (left panel) does not capture the red and blue “peninsulas” extending into the other color area. This is underfitting. Nearest neighbors (with $k = 1$) carves out a separate island for every single dot. The decision boundary seems too complex. This is overfitting.

5.3.2 Validation: which model is the best

A look at the Figure 5.3 gives a hint about how to assess the model goodness—if we just had an additional data point that is in the middle of the observed values, say at age 45, then we could compute all models’ predictions at that age, and compare that with the actual value. But we do not have any more data. What should we do?

Fortunately, there is a very easy solution to this problem. We just split the data into two parts—*training data*, the one we will actually use for fitting the model, and *validation data*, the part of the data we use later to compute the predictions in the gaps.³ From the model’s perspective, validation data is exactly the additional data point—this is a data point the model hasn’t seen, and hence the model has not had a chance to squeeze a wobbly line through those datapoints. The split of data into training and validation sets is typically done randomly, and validation data is often chosen to be 20% of the original size. This leaves most of the data, 80%, for training.

For instance, we can keep ages 26.7 and 57.2 as validation data and use everything else for training (Figure 5.5). The figure shows the same data as Figure 5.3, but now indicating training observations as green dots and validation observations as red dots. Because we now only have 8 training data points instead of 10, we can only fit polynomial regressions up to degree 7, and now 7th-degree curve perfectly fits all green training data points. However, a simple visual inspection suggests that the 7th-degree curve at red validation points is much farther off compared to the other, lower-order polynomials.

The prediction errors for both validation age, and *RMSE* are shown in Table 5.3. We can see that the linear model (1st-degree polynomial) achieves the best results—the lowest *RMSE*—here. The 7th-degree model that is able to fit all training data perfectly, produces enormous error at age 60. The constant, 0-degree model appears to be too rigid and worse in terms of *RMSE*. This is called *underfitting*, a situation where the model is too rigid and would gain from more flexibility. Models of degree 2 and more are overfitting—they follow the training data too well, while the performance on validation data suffers. This is overfitting, a situation where less flexibility would be better. Such a pattern—the performance initially improves with added flexibility, and thereafter deteriorates, is very common in practice. On one side, we are underfitting, on the other side overfitting. The best place is in the middle where the validation performance achieves its maximum.

Figure 5.6 offers another look at the same results. The left panel displays two *RMSE*-s, one for 8 training data points (red) and another for two validation data points (blue). Here we have selected a different pair of observations for validation than in Table 5.3, and the results are somewhat different. But in a similar fashion as in the table, the 0-th degree polynomial is worse than the 1st degree polynomial for validation data, while all higher degrees are worse. On training data, however, larger degree always gives better prediction, as manifested by the steadily falling training *RMSE* curve.

Obviously, the results differ if we choose different two observations for validation.

³Here we refer to the part that is used for model tuning as *validation data*. It is very common to call it *testing data* instead. However, because of conceptual similarity with cross-validation (see Section 5.3.3 Cross-validation, page 250), we call it *validation data* here. We reserve *testing data* to denote a different concept (see Section 5.3.4 Training-validation-testing approach, page 250).

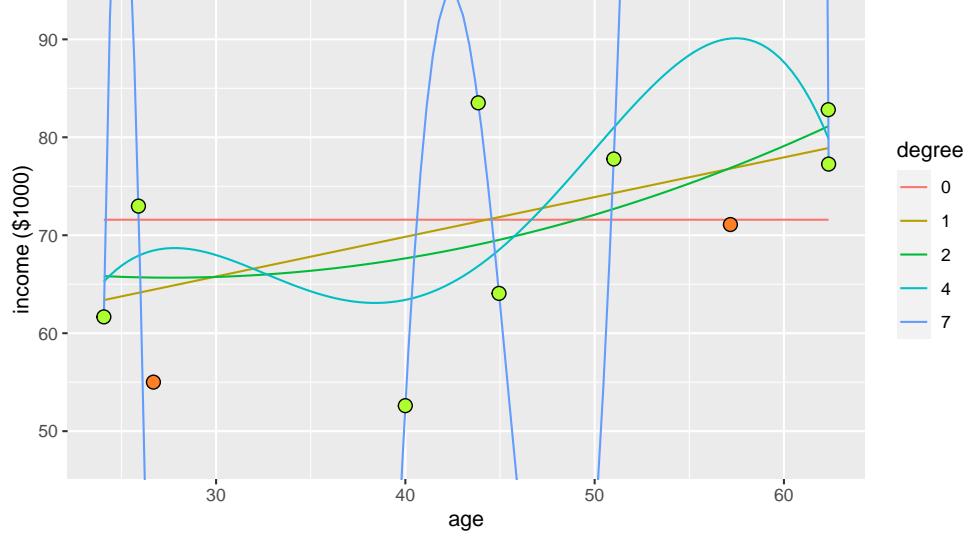


Figure 5.5: The same artificial age-income data as on Figure 5.3. The training data points are denoted with green, validation points with red. The polynomial regression curves up to degree 7 are fitted through the data training data. One can see the 7-th degree polynomial that fits all training data perfectly, predicts values that are far off from the actual validation values.

The right panel of Figure 5.6 shows a set of such curves with five different choices of validation observations. The overall picture is broadly similar—at small degree, the validation RMSE tends to be small, and it rapidly grows at a larger degree. But details differ—sometimes it is degree 0, sometimes 1, and twice degree 4 that gives the best validation RMSE. This is one of the problem with trainin/validation approach, and one of the reasons why cross-validation (see below) is preferred.

5.3.3 Cross-validation

TBD: figure

5.3.4 Training-validation-testing approach

The idea with training-validation split is to separate model fitting from model validation, and to use the latter step to improve the model. However, this essentially amounts to fitting in two steps on the complete dataset, and we are still prone to overfitting.

A possible solution is to do a three-fold split: to split data into training, validation, and testing chunks.

The training chunk is used for training individual models. This is what is normally called “training”, and typically involves computing a number of model parameters.

Table 5.3: Prediction errors from polynomial regression on validation data as shown in Figure 5.5. The linear model (1st-degree polynomial) achieves the smallest RMSE on validation data.

degree	Error at age		RMSE
	26.7	57.2	
0	16.58	0.49	11.73
1	9.44	5.71	7.80
2	10.67	5.77	8.58
4	13.43	18.99	16.45
7	-54.23	491.95	349.97

Validation chunk is used to select between different trained models. This is essentially training as well, but now we are not training model parameters, but instead *hyperparameters* by checking which models performs best. Hyperparameters are conceptually similar to the parameters, just traditionally not called like that. For instance, in case of linear regression, the parameter vector β is called “parameters”. But which features to include in the model is not called a parameter. We can call this a hyperparameter instead.

Finally, there is also a dedicated testing (hold-out) chunk. This is only used when all the model fitting and testing is done. It will indicate what is the final model performance on unseen dataset. After this figure has been revealed, one should not go back to model tuning. In actual applications, testing data is sometimes separated physically and organizationally from the main work. It may set up in a way that the research group does not even have access to that data. Instead, they submit their final model to a separate organizational entity who then computes the final performance on the testing data.

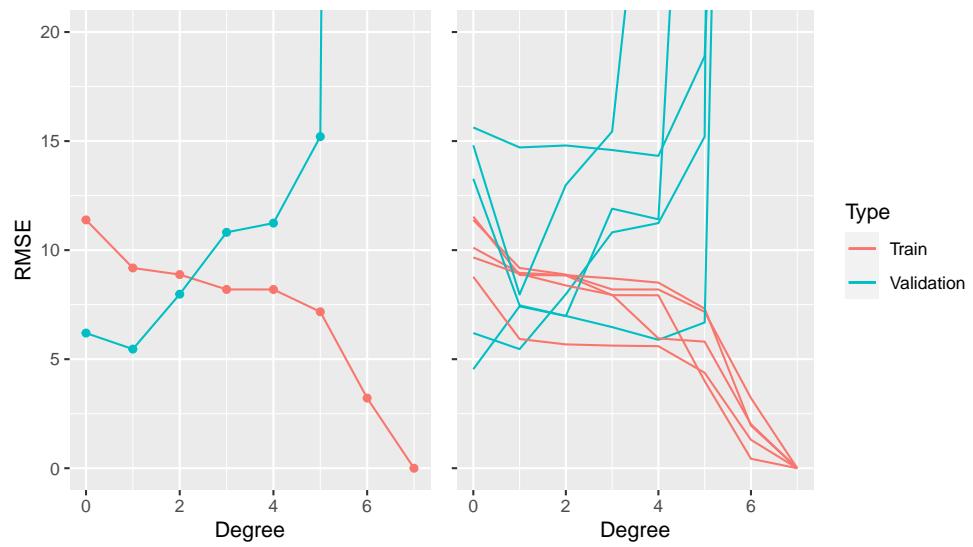


Figure 5.6: Training and Validation RMSE. Left panel displays a single training-validation split, right panel displays five different splits. Higher polynomial degree will always result in a lower RMSE when computed on training data (red), but on validation data it first falls a little bit and thereafter rapidly increases when the model gets too “wobbly” at higher degrees.

Chapter 6

Machine Learning Models

This chapter discusses several supervised learning methods and some mathematical tools that are used in some of these models.

Contents

6.1	Metric Distance: A Revisit	254
6.1.1	Data-Driven Metrics	254
6.1.2	Cosine similarity and angular distance	260
6.2	<i>k</i> -Nearest Neighbors	263
6.2.1	Introductory Example	263
6.2.2	What is Distance	264
6.2.3	Instance-based learning	265
6.3	Trees and tree-based methods	267
6.3.1	Decision trees: introduction	267
6.3.2	How trees work: two examples	270
6.3.3	Building trees: recursive binary splitting	272
6.3.4	Finding the best split: entropy	275
6.3.5	Ensemble Methods	279
6.4	Naïve Bayes	281
6.4.1	Before We Begin: A Single Word-Based Bayesian Classifier	281
6.4.2	Smoothing: how to compute probabilities with too few data	283
6.4.3	Naive Bayes Classifier	285
6.5	Support Vector Machines	296
6.6	Neural Networks	298
6.6.1	Feed-Forward Networks	298
6.6.2	Convolutional Neural Networks	307
6.7	Comparison and Review	313

Table 6.1: Recent house sales

id	price (\$ 1000)	m ²	crime (per 1000)
a	800	200	2
b	1500	400	1
c	?	200	1

6.1 Metric Distance: A Revisit

[Section 2.2.2 Metric distance](#), page 102 introduced the concept of *metric*. We mainly discussed Euclidean and other L_p -related metrics. However, in machine learning applications it is often useful to let data decide the way we measure distance. This gives rise to various data-based transformations, including feature normalization and Mahalanobis distance.

Many ML applications also permit violating the strict assumptions behind the distance metric. We may not be particularly concerned if triangle inequality holds, or if distance is zero only for identical vectors. We want a simple and good enough method to rank vectors instead. This is how we can use the popular cosine similarity measure. Below we discuss a number of more popular approaches.

6.1.1 Data-Driven Metrics

Imagine you are using the nearest neighbors method to predict house prices. Your dataset contains two training examples (a and b) and you want to predict the price of c (See Table 6.1). Nearest neighbors (see [Section 6.2 k-Nearest Neighbors](#), page 263) predicts the house c to have the same price as the more “similar” one of the training examples a and b .

But which house is more similar? Clearly, house a is of the same size while b is in a similar neighborhood. Obviously, we can choose a distance metric and compute distance. For instance, the Euclidean distance $d_E(c, a) = 1$ and $d_E(c, b) = 200$ and hence house c is more similar to house a than to house b . But does this way of deciding similarity make sense? If we measure house size in km² and crime rate per million instead of per 1000 residents, we would come to the opposite conclusion. So our similarity ranking depends on the measurement units. This looks like a false start to begin with.

We can identify two separate issues here:

1. Ranking according to Euclidean metric (as well as other L_p metrics) is not robust with respect to measurement units.
2. We don’t know how we should weight difference in size relative to the difference in crime rate.

The first problem is actually a specific manifestation of the second problem. If we were able to address the weighting, the first problem would also vanish, as the weights would presumably make the ranking unit-invariant.

One popular approach to address this problem is to use metrics that are derived from data. There are several popular approaches, all of these normalize the units with respect to certain variation in data. However, despite that these distance metrics are “data driven”, they are not necessarily more correct than other units. Sometimes the preferred metric can be deduced from the nature of the problem, but other times one has just to experiment and find the best approach.

TBD: Example with an island

Feature normalization

Perhaps the most popular such data-driven metric is *feature normalization*: transforming the features into mean-zero and variance-one version. This would constitute an answer to the house-price-problem above along these lines: “we think that customers value one standard deviation difference in house size about the same as one standard deviation difference in the neighborhood crime rate”.

Technically, normalized features can be computed like this. Consider a feature vector of length N , $\mathbf{x} = (x_1, x_2, \dots, x_N)$. It can be transformed into the normalized vector $\tilde{\mathbf{x}}$ by first subtracting its mean, and thereafter dividing it by standard deviation:

$$\tilde{\mathbf{x}} = \frac{\mathbf{x} - \bar{x}}{\text{sd } \mathbf{x}} \quad \forall i \in 1 \dots N \quad (6.1.1)$$

where $\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$ is the average value of \mathbf{x} and $\text{sd } \mathbf{x} = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}$ is the standard deviation of \mathbf{x} . The normalized vector $\tilde{\mathbf{x}}$ has mean zero and standard deviation 1. Note that normalization is done for each feature vector independently, the other features do not play any role here. However, we must know the values for all observations for that vector to compute \bar{x} and $\text{sd } \mathbf{x}$. The result, obviously, does not depend on the units any more because $\text{sd } \mathbf{x}$ in 6.1.1 is measured in the same units as \mathbf{x} . We can say that we are introducing a new unit of measurement, the standard deviation of \mathbf{x} .

Example 6.1: Data normalization

Consider the matrix \mathbf{X} below. It contains three columns, $X_{\bullet 1}$, $X_{\bullet 2}$ and $X_{\bullet 3}$, and three rows a , b and c . The first two columns have similar spread (2) but different means (2 and 12 respectively), while the third column also has a different scale.

	$X_{\bullet 1}$	$X_{\bullet 2}$	$X_{\bullet 3}$
a	1	11	10
b	2	12	20
c	3	13	30
mean $\bar{X}_{\bullet j}$	2	12	20
std.dev $\text{sd } X_{\bullet j}$	1	1	10

$x_{\bullet 1}$ stresses that index “1” is the column index, the bullet • is a placeholder for rows.
See [Section 0.1 Scalars, vectors, matrices](#), page vii.

The table also lists the mean value for all columns $\bar{X}_{\bullet j}$ for $j = 1, 2, 3$, and (sample size corrected) standard deviations. When normalizing data, we simply subtract the corresponding mean from each variable in data (i.e. “2” from values of $X_{\bullet 1}$,

“12” from $\mathbf{X}_{\bullet 2}$ etc), and divide the resulting differences by the corresponding standard deviation (i.e. “1” for $\mathbf{X}_{\bullet 1}$ and $\mathbf{X}_{\bullet 2}$, and “10” from $\mathbf{X}_{\bullet 3}$). We get

	$\tilde{\mathbf{X}}_{\bullet 1}$	$\tilde{\mathbf{X}}_{\bullet 2}$	$\tilde{\mathbf{X}}_{\bullet 3}$
1	-1	-1	-1
2	0	0	0
3	1	1	1

One can see that all three variables are now equal—they are all $(-1, 0, 1)$. This is rather intuitive: they all have one observation in the middle, and two other at each side and equally far from it.

Figure 6.1 shows a graphical comparison of normalized and non-normalized features. The left panel depicts the data points in the original feature space where spread of x_2 is much larger than spread of x_1 . The dotted circle denotes a set of equidistant points from the dark blue point in its center (using Euclidean distance in \mathbb{R}^2). One can see that the circle encompasses the green dot but not the yellow dot—hence the green dot is closer to the dark blue dot than the yellow dot. On the right panel we see the normalized version of the same data. The visual impression confirms that both features are now spread roughly equally. The solid circle depicts a set of equidistant points from the dark blue dot in this feature space. In this space, the yellow dot is closer to the dark blue dot than to the green dot. Feature normalization reverses the distance ranking. Both panels also show the circles in the other feature space, those are now transformed to ellipses.

Figure 6.2 gives a similar example using Boston housing data. Both the left and the right panel depict the same data, neighborhood crime rate versus the average number of rooms. On the left panel we use the original features while on the right panel we use the normalized features. Unlike in Figure 6.1, we do not force equal aspect ratio here and hence both panels look exactly the same, only the values on the axes differ. The Euclidean distances differ too. For instance, the distance between the green and the orange dot is 1.349, and between the green and the blue dot 5.666 in the original features (left panel). These distances are 1.92 and 0.666 in normalized features (right panel). Hence the closest colored neighbor to the green dot is orange in the original features and blue in the normalized features.

From the technical point of view, normalization is a good option if the features are roughly independent, and their distribution is roughly symmetric and does not have fat tails. This assures that the variance is stable and the mean is in the middle of the observations.

More conceptually, normalization is justified in such cases where standard deviation is a relevant scale unit. In case of the house price example above, this is true if people consider both house size and neighborhood security a relevant measure, and standard deviation of the respective variables is a good proxy for how people value these two factors. However, if the customers never care about crime, except for the worst few neighborhoods, feature normalization may not be a good approach.

Another common reason to use feature normalization is to transform values that are measured in arbitrary and hard-to-understand units into more easily understand-

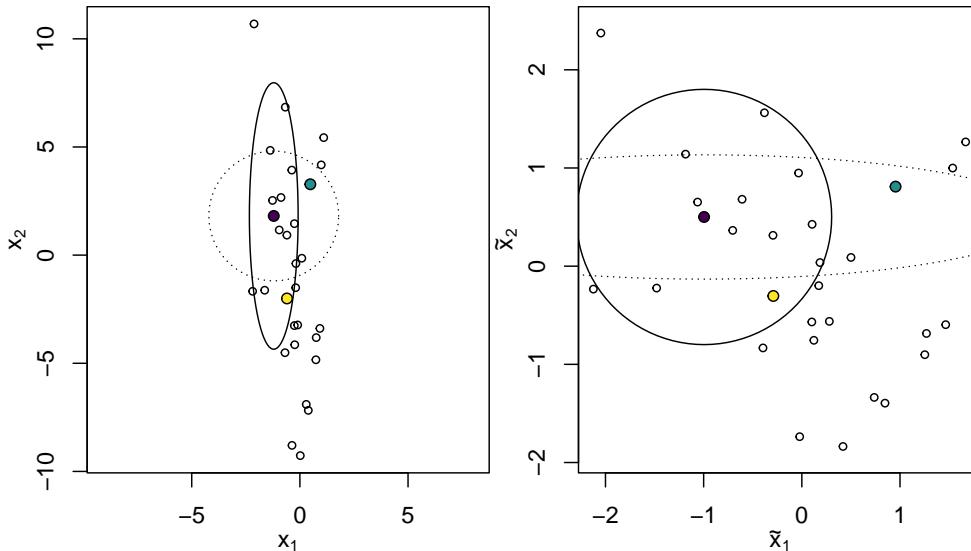


Figure 6.1: Non-normalized features (left) and normalized features (right). Dark blue, green and yellow mark the same three datapoints on both images. The dotted line depicts a circle in the original feature space, the solid line is a circle in the normalized feature space. Note how the relative distance between dark blue and green, and dark blue and yellow dots differ in the original and in the normalized features.

able (and comparable) ones. For instance, we may survey the support for a government policy on a scale from 1 (very much against it) to 5 (very much in favor of it). One unit in this scale is hard to understand while a sentence like “those whose support it one standard deviation above the mean...” carries more meaning.

There is one more technical reason why it is advisable to normalize features—if the design matrix contains columns of very different scale then its condition number will be high and hence the numeric properties may suffer.

Min-max scaling An easy alternative to normalization is min-max scaling. It is conceptually similar to normalization, just instead of dividing the centered values by the standard deviation, we divide those by the range. For example, we can create a new set of features that are all in equal range as follows:

$$\tilde{x}_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \quad \forall i \in 1 \dots N \quad (6.1.2)$$

where x_{\min} and x_{\max} are the corresponding minimum and maximum values of \mathbf{x} . In this example all the features will be converted into the $[0,1]$ interval, but we can shift and scale these into another interval instead, say $[-0.5, 0.5]$. Min-max scaling works well if the features are independent and have uniform-like distribution with no tails—all values end abruptly at the boundary. This ensures that the minima and maxima are stable.

Matrix condition number is the ratio of the largest and the smallest eigenvalue, $\kappa \equiv \frac{|\lambda|_{\max}}{|\lambda|_{\min}}$. See [Section 2.3.4 Condition number](#), page 116.

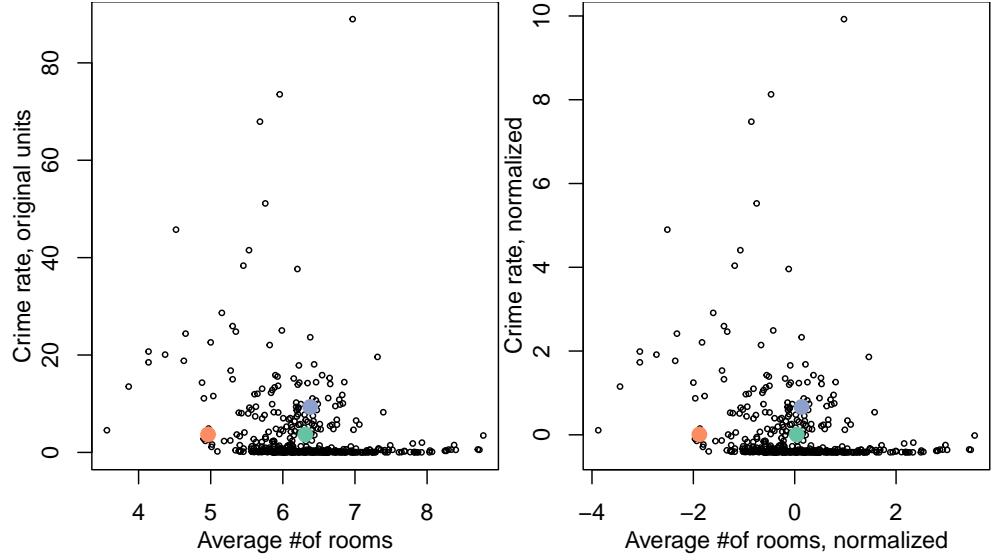


Figure 6.2: Boston housing data: neighborhood crime rate (crim) versus average number of rooms (rm). Non-normalized (left) versus normalized features (right). While the images look exactly the same, the Euclidean distance rankings are different: the nearest (colored) neighbor the green dot is the orange on the left panel, and the blue dot on the right panel.

As min-max scaling is very similar to feature normalization, its advantages and disadvantages are similar too.

Mahalanobis distance **Prerequisites:** [Section 2.2.2 Norm and Distance](#), page [99](#), [Eigenvalues and eigenvalue decomposition](#) [2.3.4](#), [feature normalization](#) [6.1.1](#), covariation matrix.

This is a generalization of feature normalization in case where the features may be correlated. Consider Figure 6.3. Here the two features \mathbf{x}_1 and \mathbf{x}_2 do not just have a different variance, they are also clearly correlated. If we use feature normalization we discussed above, we will change the picture somewhat, but we cannot address the fact that the data points are clearly clustered around the diagonal line.

Mahalanobis transformation, in contrast, stretches and rotates the data in a way that is aligned with the axis of the data. In the left panel of Figure 6.3, the solid ellipse depicts the equidistant points from the central dark blue dot. The ellipse is elongated along the long axis of the correlated data, and compressed along its short axis. So Mahalanobis distance measures distance with respect to the extent of the point cloud in each particular direction, not just along the coordinate axes as is the case with feature normalization.

Mahalanobis distance can be done and understood easily using matrix notation and eigenvalue decomposition. Consider \mathbf{X} to be a $N \times K$ data matrix and $\mathbf{x}_{i\bullet}$ and $\mathbf{x}_{j\bullet}$ to be two rows (observations) from that data. The Mahalanobis distance between

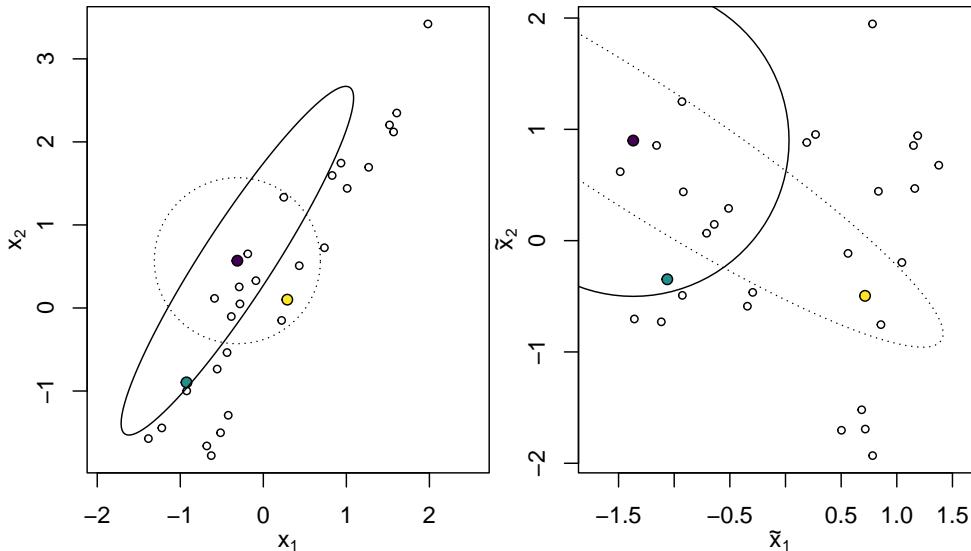


Figure 6.3: Original features (left) and Mahalanobis-transformed features (right). The same three cases are marked with different colors on both images. The dotted line depicts a circle in the original feature space, the solid line is circle in Mahalanobis feature space.

the observations $\mathbf{x}_{i\bullet}$ and $\mathbf{x}_{j\bullet}$ is defined as

$$d_E(\mathbf{x}_{i\bullet}, \mathbf{x}_{j\bullet}) = \sqrt{(\mathbf{x}_{i\bullet} - \mathbf{y}_{x\bullet})^\top \Sigma^{-1} (\mathbf{x}_{i\bullet} - \mathbf{x}_{j\bullet})} \quad (6.1.3)$$

where Σ is the covariance matrix of \mathbf{X} .

Mahalanobis distance is equivalent to transforming the data matrix into

$$\tilde{\mathbf{X}} = \left(\mathbf{X} - \mathbf{1}_N \cdot \bar{\mathbf{x}}^\top \right) \Sigma^{-\frac{1}{2}} \quad (6.1.4)$$

where $\bar{\mathbf{x}}$ is the vector of column means and $\mathbf{1}_N \cdot \bar{\mathbf{x}}^\top$ is accordingly a matrix of column means.

Mahalanobis transformation is essentially the same as transforming data to [principal components](#) (see Section 9.3). Thereafter it measures Euclidean distance in such a rotated and stretched feature space. In case the features are uncorrelated, Mahalanobis distance is equivalent to Euclidean distance in normalized data.

Mahalanobis distance is a good measure for data where the data variation is a meaningful distance measure, and not just along the features as in case of normalization, but also along the axes of variation in data.

Example 6.2: Mahalanobis transformation of iris data

We demonstrate the Mahalanobis transformation using Iris data. Figure 6.4 shows an analogous transformation as in Figure 6.3 for 2-D slice of iris data. The different species, denoted by different colors, are reasonably well separated

on both figures. However, in the original coordinates (petal length and width, left panel) the data points form an elongated cloud where the different species cluster at different location in this cloud. In transformed coordinates, the points are stretched out along the minor axis, increasing the distance between dots for similar species.

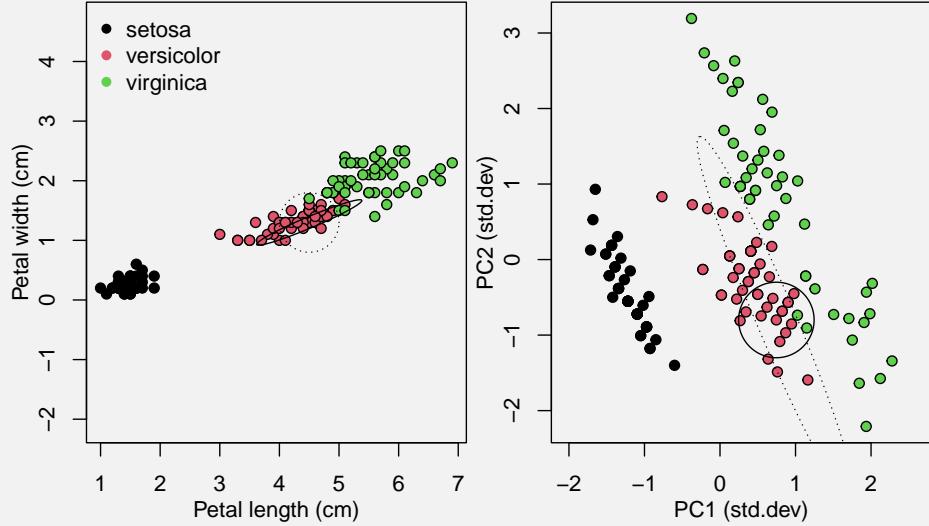


Figure 6.4: Iris data: petal width versus petal length in the original coordinates (left panel) and in the corresponding Mahalanobis-transformed coordinates (right panel).

In transformed coordinates the species do not form as tight clusters any more as in the original coordinates, making categorization more difficult. This fact is also visible from the example circles, the circle that centers on a red observation in Mahalanobis coordinates (solid line) includes two green points while the circle in the original coordinates (dotted line) includes only a single green dot while capturing most of the red ones. For k -NN to work well, it should be possible to draw such circles around most datapoints that contain many dots of the correct color and few of other colors. This is easier in the original coordinates.

Note that one of the major reason for data transformation, features measured in incomparable units, does not hold here as both features are measured in centimeters.

6.1.2 Cosine similarity and angular distance

Prerequisites: [Vector Norm 2.2.2](#)

Cosine similarity is a similarity measure that is not based on L_p distance. It is widely used when assessing similarity in features that are not numeric, such as when comparing texts.

Cosine similarity between vectors \mathbf{x} and \mathbf{y} is defined as

$$c(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^\top \cdot \mathbf{y}}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|} \quad \mathbf{x} \neq \mathbf{0}, \mathbf{y} \neq \mathbf{0}, \quad (6.1.5)$$

where $\|\mathbf{x}\| = \sqrt{\mathbf{x}^\top \cdot \mathbf{x}}$ is the Euclidean norm. It is easy to see that $c(\mathbf{x}, \mathbf{x}) = 1$.

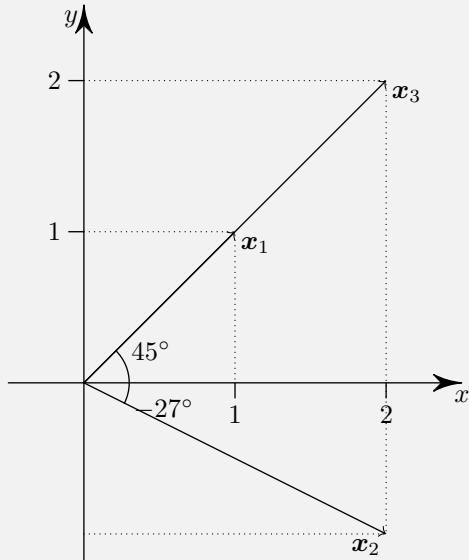
It's name, *cosine* similarity, originates from the fact that inner product of vectors equals to the product of their norms, multiplied by the cosine of the angle between them:

$$\mathbf{x}^\top \cdot \mathbf{y} = \|\mathbf{x}\| \cdot \|\mathbf{y}\| \cdot \cos \phi, \quad (6.1.6)$$

where ϕ is the angle between vectors \mathbf{x} and \mathbf{y} . Hence cosine distance equals just to the cosine of the angle between the vectors. Note that it is *solely the angle* between the vectors. Cosine distance is agnostic to the length ([norm](#)) of the vectors (as long as this is positive). It is a measure in similarity in direction the vectors point to, and not a measure of the length of the vectors. For instance, when analyzing texts using bag-of-words (see [Section 7.3.3](#)), this amounts to comparing word frequencies in the texts. The number of words (text size) is irrelevant. Such an approach may be very well suited when we try to understand the topic of the text while the texts itself may be of very different size.

Example 6.3: Cosine similarity in \mathbb{R}^2

The easiest way to understand cosine similarity is to analyze it in \mathbb{R}^2 plane. Look at the vectors \mathbf{x}_1 and \mathbf{x}_2 on the figure below. $\mathbf{x}_1 = (1, 1)$ and hence it points 45° upward. $\mathbf{x}_2 = (2, -1)$ and accordingly points 27° downward, and hence the angle between the two vectors is $45 + 27 = 72^\circ$.



Now calculate cosine similarity. First, the Euclidean norms are

$$\begin{aligned}\|\mathbf{x}_1\| &= \left\| \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\| = \sqrt{\begin{pmatrix} 1 \\ 1 \end{pmatrix}^\top \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix}} = \sqrt{1+1} \approx 1.414 \\ \|\mathbf{x}_2\| &= \left\| \begin{pmatrix} 2 \\ -1 \end{pmatrix} \right\| = \sqrt{\begin{pmatrix} 2 \\ -1 \end{pmatrix}^\top \cdot \begin{pmatrix} 2 \\ -1 \end{pmatrix}} = \sqrt{4+1} \approx 2.236.\end{aligned}\quad (6.1.7)$$

Now we can plug the numbers into the cosine similarity definition (6.1.5):

$$c(\mathbf{x}_1, \mathbf{x}_2) = \frac{\mathbf{x}_1^\top \cdot \mathbf{x}_2}{\|\mathbf{x}_1\| \cdot \|\mathbf{x}_2\|} \approx \frac{\begin{pmatrix} 1 \\ 1 \end{pmatrix}^\top \cdot \begin{pmatrix} 2 \\ -1 \end{pmatrix}}{1.414 \cdot 2.236} = \frac{2-1}{3.162} = 0.316. \quad (6.1.8)$$

We can easily check that 0.316 is cosine of 71.6° . Hence the computed cosine similarity is equal to the cosine of the angle between \mathbf{x}_1 and \mathbf{x}_2 . (The difference is related to rounding errors.)

TBD: Example where two vectors of different size are at same similarity with a third one

Cosine similarity has a few very favorable properties, in particular it is easy to compute, involving just multiplications, additions, and one division. In case of sparse matrices, only non-zero components need to be considered. All this makes it very well suitable for analyzing high-dimensional data, such as words in texts.

Unlike the distance measures above, cosine similarity is not a metric distance as larger value means not more distant but more similar data vectors. The maximum similarity, distance between identical vectors is 1 while the minimum similarity, distance between opposite vectors, is -1. This is sufficient to order vectors according to their similarity, and often this is all we need.

In case one needs a difference measure instead of similarity measure, one can use cosine distance $d_{cos}(\mathbf{x}, \mathbf{y}) = 1 - c(\mathbf{x}, \mathbf{y})$. Cosine distance is zero in case of vectors that point in the same direction, the maximal possible distance is 2 when two vectors point in exactly opposite direction. Another option is to use *angular distance*, defined as

$$d_a(\mathbf{x}, \mathbf{y}) = \frac{\cos^{-1} c(\mathbf{x}, \mathbf{y})}{\pi}, \quad (6.1.9)$$

instead of cosine distance. However, there is little gain from selecting a more computationally demanding metric if our task is just to rank vectors according to similarity.

Exercise 6.1: Cosine similarity

Consider vectors $\mathbf{x}_1 = (1, 2, 3)$, $\mathbf{x}_2 = (3, 2, 1)$ and $\mathbf{x}_3 = (1, 1, 1)$.

1. Compute the (Euclidean) norms $\|\mathbf{x}_1\|$, $\|\mathbf{x}_2\|$ and $\|\mathbf{x}_3\|$.
2. Compute the normalized vectors $\mathbf{x}_1^n = \mathbf{x}_1 / \|\mathbf{x}_1\|$, $\mathbf{x}_2^n = \mathbf{x}_2 / \|\mathbf{x}_2\|$ and $\mathbf{x}_3^n = \mathbf{x}_3 / \|\mathbf{x}_3\|$.
3. Compute cosine similarity between \mathbf{x}_1 and \mathbf{x}_2 , and between \mathbf{x}_1 and \mathbf{x}_3 .

Hint: use the normalized vectors to compute similarity.

Solution on page [429](#).

Exercise 6.2: Cosine, angular distance are not proper metric distances

Show that neither cosine nor angular distance are proper [metric distances](#) as defined in Section [2.2.2](#).

6.2 k -Nearest Neighbors

Prerequisites: [Metric distance](#)

Nearest neighbors is one of the simplest and most intuitive machine learning methods. We predict the value, or a class, of a new observation as the class of the most similar observation in the training data set.

6.2.1 Introductory Example

Imagine we have data as depicted on Figure [6.5](#), left panel. It contains yellow and violet training observations, and our task is to categorize the empty unknown data points into one of these color categories. Intuitively, it is reasonable to assume that points that are “close” on the image should have similar color. So if an empty circle is fairly close to a violet training observation and far from everything yellow, we should consider violet as a good prediction for the unknown class.

This intuitive approach is the basis for *nearest neighbor* classification: we just categorize an unknown data point into the category that corresponds to the category of its closest neighbor. This has been done on the right panel of Figure [6.5](#): it divides the figure into tiny squares (101×101 squares) and categorizes the center of each square into either yellow or violet by looking at its closest neighbor’s color. The squares that coincide with the unknown data will tell us how the model will categorize these data points.

This baseline approach is very sensitive to individual outliers in the data. If we have a violet point sitting deep inside the yellow territory, we would immediately think that everything in the close neighborhood of the outlier also belongs to the violet class. Nearest neighbors does not allow for reasonable smoothing. Fortunately, a remedy here is very easy. Instead of using the category of the nearest neighbor as the predicted class, we can smooth the picture somewhat by using, say, 5 nearest neighbors, and finding the category that is preferred in this group (often referred to as *majority voting*). If 3 out of the 5 closest neighbors are yellow and 2 are violet, we will pick yellow. This results in a noticeably smoother pictures (Figure [6.6](#) shows exactly the same data categorized using 5 and 25 nearest neighbors).

This is the essence of k -nearest neighbors (k -NN). In case of only two categories, k is often chosen to be an odd number in order to avoid ties in majority voting.

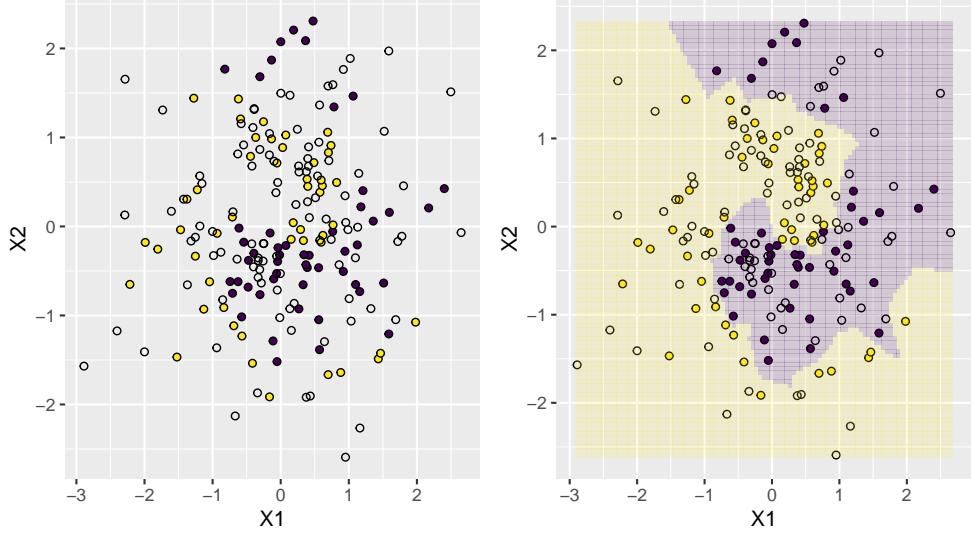


Figure 6.5: Example data: some of the datapoints are categorized into yellow and violet, but some are not (left panel). Intuitively, the empty circles should be classified according to a colored one nearby. This is the intuition of the nearest neighbor method. On the right panel, all the points that are closer to a violet one are painted violet and those that are closer to a yellow one are colored yellow. All the empty circles now lie in one of these areas of solid color and can be categorized either as yellow or violet.

6.2.2 What is Distance

However, the simple and intuitive method is not without its issues. As soon as we leave the 1-dimensional world, it may not be clear any more which observations are closer to each other. The nice example in Figure 6.5 is somewhat deceiving, by making you to believe that what looks close on the image is also close in the data space. But take a simple example. Assume you are predicting house prices and you have data of recent sales like in Table 6.2, including the price (in \$1000), size (m^2), and neighborhood crime rate (incidents per 1000 residents). Your task is to predict the price of the house c that is similar to a in terms of neighborhood crime rate, and similar to the house b in terms of size. Which one is more similar? Your prediction will be very different depending on which one you choose as the nearest neighbor. Obviously, there is no correct way to tell. We have to weight the different features somehow by using an appropriate distance metric.

Moreover, the previous example used simple numeric features, but this may not always be so. How can you tell which text is closer to another one? Which customer is more similar to a third one? In these cases we don't even have numeric measures to start with, and our decisions about creating those add an additional layer of assumptions to the model.

Obviously, one can always choose a pre-determined distance metric, either Eu-

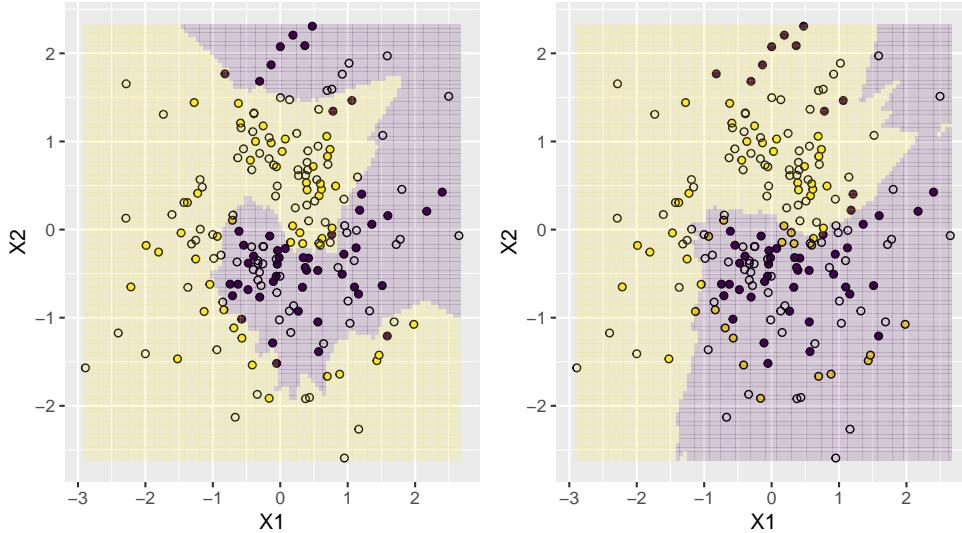


Figure 6.6: The same data points as in Figure 6.5, but now categorized based on 5 (left) and 25 (right) nearest neighbors. We can see that in the latter case, there are several groups of points that are embedded in the area of different color.

Table 6.2: Recent house sales

id	price (\$ 1000)	m ²	crime (per 1000)
a	800	200	0.2
b	1500	400	0.1
c	?	200	0.1

clidean or another one. Even more, k -NN does not require the metric to be a valid metric in the sense of vector spaces, it is enough if it allows us to order the observations by “closeness” in a consistent way. This opens the option for cosine similarity (more about it later).

6.2.3 Instance-based learning

k -NN is somewhat different from many other machine learning models, such as linear regression, decision trees or neural networks in the sense of what does model training mean. In case of linear regression, *training* the model means computing the best coefficient vector β . Neural networks are similar, just we call the parameters “weights” and “biases”. The case of trees is broadly similar, but the parameters are not just numbers, but lists of splitting variables and locations. In all these cases, “training” means computing the parameters or deciding the split locations, and the “trained model” is just set of such parameters. Normally the set of parameters is much smaller

than the original data, e.g. we may have to compute 100 parameters out of 100,000 rows of data. In a way, model training is a way to compress data, this is hard work and you may notice that training complex models on large datasets is slow.

But this is not true for nearest neighbors. Plain k -NN does not compute any parameters or other model features. After all, predictions are made by finding the closest neighbors to the point of interest, and this cannot be done if we do not have access to the original data. So k -NN “learns” by just memorizing data. Obviously, just storing data is in no way a compression algorithm, and hence “trained” k -NN models are large, as large as the dataset (or more precisely, as large as the design matrix).

This is also a reason why k -NN models are not interpretable. It does not help to explain the relationship between variables, it is just a description: this point of interest is more similar to one outcome, another point of interest is more similar to another outcome. This is why we predict the outcomes to be different. But sometimes such a description may be enough to *explain* the outcome to others.

6.3 Trees and tree-based methods

Decision trees is a popular method for predictive modeling, both for regression and categorization problems. Trees are easy to implement and easy understand. Trees is one of the most explainable machine learning method, a method where it is possible to explain someone how and why certain decisions were made. Our mental decision-making is often based on tree-like way of thinking.

Trees are also a popular way to “grow forests”, a large collection different tree-based methods combined into a single ensemble method.

First, we describe what exactly are the decision trees and what are the main related concepts. Afterwards we look at algorithms to create trees; and finally tree-based ensemble methods (forests).

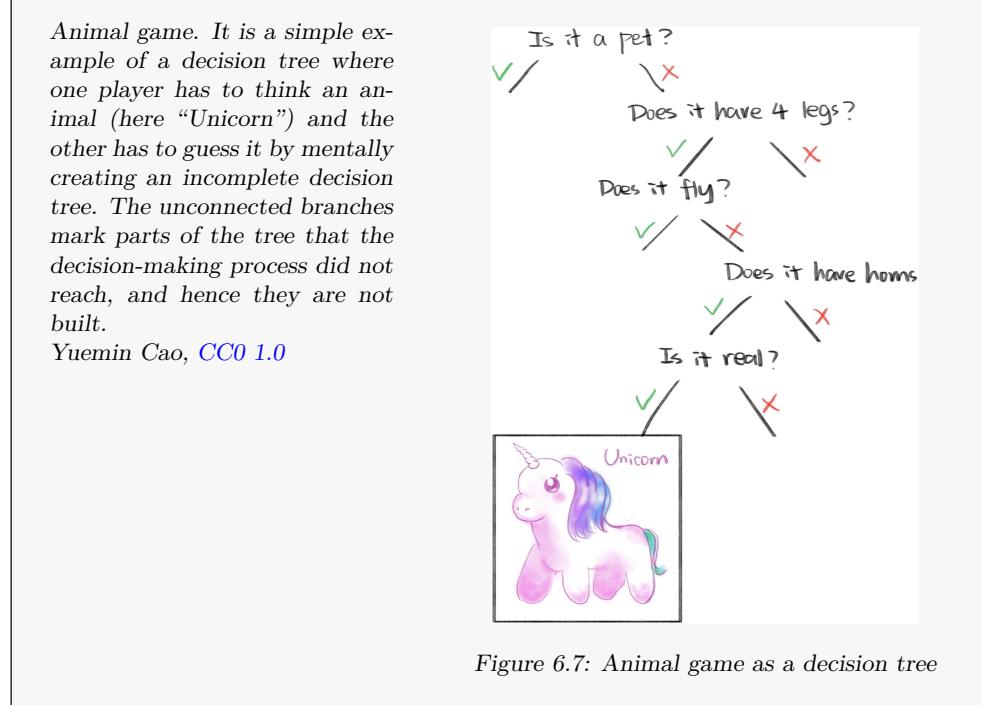
6.3.1 Decision trees: introduction

A popular implementation of tree is the “animal game”, a children game where one player thinks an animal, and the other player have to guess it by asking questions regarding the animal. But the questions must be worded in a way that the answer is always “yes” or “no” (Figure 6.7). Based on the answers, the player who is guessing can narrow down the set of possible animals until it contains just the correct one. One can write down the questions and yes-no answers, although no-one does it in this game. The result will look like a tree, or more precisely like a single path from the trunk (the first question) till a leaf (the answer). The other branches—questions not asked—will remain incomplete. Note that the tree is depicted “upside-down”, decision trees are traditionally depicted with the trunk at top and branches leading downward.

However, the decision trees in this game are based on our pre-existing knowledge, not on any data. They are also created on-the-fly, depending on the answers to the previous questions. As a result, they may be quite inefficient. Further below we discuss how to build decision trees based on data and how to do it in an efficient manner.

Figure 6.8 demonstrates a decision tree that is built on data. It describes survival rate on Titanic, based on three variables—sex (male or female), passenger class (1st, 2nd or 3rd), and age. This example demonstrates all the basic concepts of decision trees (although the exact details may differ).

- The tree consists of nodes, branches, and leafs. Nodes are points where the tree makes decision based on values in data. In the figure they are depicted as blue ovals with two numbers inside—the survival rate, and percentage of the total observations.
- The tree starts with the trunk node (at top). In Figure 6.8, it contains the full (100%) dataset with the average survival rate of 0.38 (printed inside the blue node, and also reflected in the blueness of it). Each node involves a decision, printed underneath the blue node. The first decision is about gender. If the corresponding individual is male, we take the left branch, if not male, then the right branch. Each node has a single question with only two possible answers: “yes” (left) and “no” (right).



- Branches lead to new nodes (new questions) or *leafs*. These are points where we do not ask any more questions but output the predicted value instead. In this figure, these are the survival rates (first numbers inside of the node). However, these may also be predicted categories, probabilities of the predicted categories, counts in the dataset, or all the above.

There is a number of reasons why decision trees are popular:

- Trees are easy to understand, also be those who have no formal training in machine learning methods. They are *explainable*.
- They reflect logic of human decision-making.
- Trees can be built with arbitrary complexity, they can be made very simple (trunk and two branches), or extremely complex.
- There are well-established methods how to simplify too complex trees (pruning) in order to avoid overfitting.
- Trees can be used both for both regression and classification tasks. There are little conceptual differences between trees built for these two outcomes. Classification trees can handle any number of categories with no additional tricks.
- It is easy to modify the tree growing process in different ways. This makes it easy to grow many different trees based on the same data and in this way to create ensemble methods.

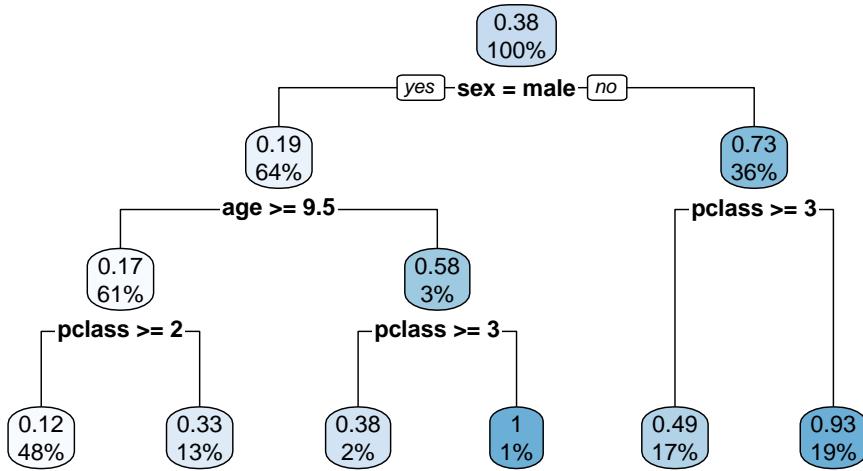


Figure 6.8: Decision tree to predict Titanic survival. We start at the topmost root node (survival rate 0.38 for 100% of the cases). The first decision is based on “sex”: if it is male, we move to the left branch (survival rate 0.19 for 64% of cases), if not male, then to the right branch (survival rate 0.73 for 36% of cases). The next decision is based on “age” for men, and on “pclass” for women. Finally we reach a leaf node, e.g. for the 1st and 2nd class women (the rightmost leaf) it tells that their survival rate was 0.93.

Decision trees have the following downsides:

- While decision trees are easy to explain, they may be hard to interpret. It may be easy to explain *how* a decision was made, but hard to understand what does this tell about the problem, and why the decision is done in this way and not another way.
- Decision trees may be rather unstable, small variations in data may lead to totally different trees. This makes it harder to understand and interpret trees, even if the predictions remain similar. For comparison, think about linear regression—small changes in data change the regression line just a little bit.

Unlike the animal game example above, typical decision trees are made “of data” using popular algorithms. The popular recursive binary splitting (see [Section 6.3.3 Building trees: recursive binary splitting](#), page 272) works broadly in this way:

- Pick a feature and split your data into two parts based on this feature.
- Now take these two parts, and repeat the process.
- When it is time to stop, for instance, when there is too few cases left in the splits, then predict as the mean value (for regression) or as the majority category (for classification) in each split.

- Each time, when deciding how to split your data, choose such a split that minimizes the total variance.

6.3.2 How trees work: two examples

We begin with a simple classification task (Figure 6.9). The left panel displays the decision boundary plot—a plane with red and blue dots. The tree is attempting to categorize the regions according to the dominant color of the dots. The right panel displays the corresponding decision tree. The first question the tree asks—the condition at the root node—is if $x_2 < 0.14$, i.e. if the point lies below the horizontal line approximately in the middle of the figure (orange horizontal dotted line). All 100 training observations are going through that test. If the condition is true (this is the case for 51 observations), then we move to the next node left of the root. That node does not do any further tests but classifies the result as red (category “0”) as 38 out of the 51 observations there are red. This is a leaf node. But if $x_2 \not< 0.14$, then we move right of the root to the first node there. This tests if $x_1 < -0.47$. The condition corresponds to the orange vertical dotted line. If true, we are at left of this line (and we are also above the $x_2 = 0.14$ line, and we again categorize the data points as red as 9 out of 16 observations here are red. However, if $x_1 \not< -0.47$ (top-right region), then the predicted category is blue (category “1”).

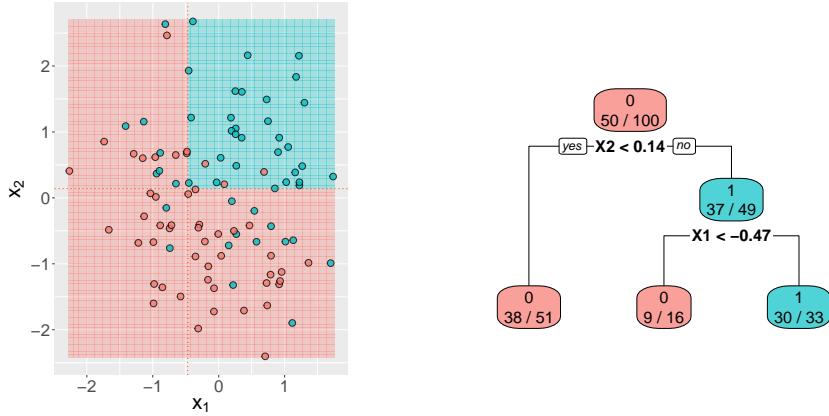


Figure 6.9: A simple decision tree solving a 2-D classification task. The decision boundary plot (left) and the corresponding decision tree. Dotted orange lines denote the node conditions.

So this tree splits the feature space into three rectangles (the three leafs in the right panel). Two of these leafs predict red, one predicts blue. Because the conditions are testing if x_1 and x_2 are below certain thresholds, the corresponding decision boundaries are vertical or horizontal lines. Hence the corresponding decision regions are rectangles. One can easily see that if we add a third feature to the data, the rectangles will transform into 3-D boxes, and in case of more dimensions, these will be hyperrectangles of the corresponding dimension.

Traditionally, there are two limitations put onto the conditions and branching. First, the conditions are made in this way that they always only have two possible answers (True or False). This is not really a limitation, because we can always describe a multi-branch node as a series of two-branch nodes. For instance, if gender is coded as “male”, “female” and “not specified”, we could make a node that leads to three branches, one for each possible gender values. However, instead of this we can also make two two-branch nodes, the first of these may distinguish between “male” and not “male”, and the second one between “female” and not “female”. In both cases we end with three branches that lead to further conditions (or leafs). This approach has the advantage that the corresponding algorithms are simpler.

Second, the conditions are traditionally simple: they only involve a single feature and only greater-than/less-than comparisons. One can imagine doing conditions like $x_1 + x_2 > 0$ to achieve diagonal decision boundaries, but that is not traditionally done in decision trees. But this is a real limitation and it means that decision boundaries for diagonal regions are complex patterns of horizontal and vertical lines. Trees are better in capturing vertical than diagonal structures.

The second example considers an 1-D regression task. We use Boston Housing data to predict the median house values across neighborhoods, based on the average number of rooms. The left panel of Figure 6.10 shows data (gray circles) and predicted values (red line), and the right panel shows the corresponding regression tree. As we can see, the tree splits the data into four leafs, with the corresponding predicted values

$$\widehat{medv} = \begin{cases} 19 & \text{if } rm < 6.5 \\ 25 & \text{if } 6.5 \leq rm < 6.9 \\ 32 & \text{if } 6.9 \leq rm < 7.4 \\ 45 & \text{if } rm \geq 7.4. \end{cases}$$

On the figure, each leaf corresponds to a horizontal stretch of the blue prediction line, the decision boundaries between leafs are depicted as orange vertical lines.

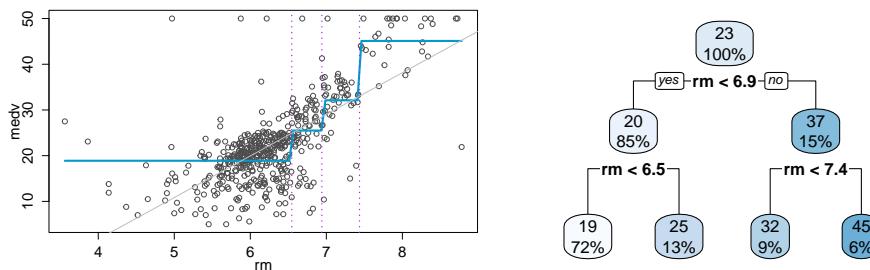


Figure 6.10: Regression tree solving an 1-D problem. Predicting house values based on number of rooms, Boston Housing data. On these data, the tree achieves RMSE = 5.93, while linear regression (gray line) has RMSE = 6.60.

6.3.3 Building trees: recursive binary splitting

Let's now discuss how to make trees based on data.

We have predictors \mathbf{X} , outcome \mathbf{y} :

- $\mathbf{y}: N \times 1$
- $\mathbf{X}: N \times K$

Split the feature space into M rectangular regions \mathcal{R}_m , $m = 1, 2, \dots, M$.

Predict a constant value in each region:

$$\hat{y}(x) = \sum_m c_m \cdot \mathbb{1}(\mathbf{x}_i \in \mathcal{R}_m)$$

Where c_m is the predicted value for region m .

c_m is the average in region m

$$c_m = \frac{1}{n_m} \sum_{i: x_i \in \mathcal{R}_m} y_i$$

To put it simple, trees split the feature space into rectangular blocks (leafs), and in each leaf, they predict the majority outcome. We would like to find the best possible way to slice the feature space – to create as pure leafs as possible, but in general, this cannot be done. There are just too many possible ways to split a high-dimensional dataset. We need a simpler way that is good enough.

The most popular, and feasible, algorithm is called *recursive binary splitting*. The idea of this algorithm is fairly simple. You just split your dataset into two halves (these do not have to be of equal size). Thereafter you treat the two halves as two different datasets, and you just continue splitting until some sort of stopping condition is reached (Figure 6.11). Obviously, you should not just split the data in an arbitrary manner, you should find the best split before you do it (see [Section 6.3.4 Finding the best split: entropy](#), page 275).

- Data:

$$(\mathbf{X}, \mathbf{y}) = \left(\begin{pmatrix} \mathbf{x}'_{1\bullet} \\ \mathbf{x}'_{2\bullet} \\ \vdots \\ \mathbf{x}'_{N\bullet} \end{pmatrix}, \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} \right) = \left((\mathbf{x}_{\bullet 1}, \mathbf{x}_{\bullet 2}, \dots, \mathbf{x}_{\bullet K}), \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} \right)$$

K attributes of length N , a single endogenous variable y of length N .

Assume binary features $x_{ij} \in \{0,1\}$.

```

1  function growTree((X, y))
2      let N = number of cases in X
3      if y_i = 0 for all i = 1, 2, ..., N:
4          return leaf(0)
5      if y_i = 1 for all i = 1, 2, ..., N:
6          return leaf(1)
7      let j = bestAttribute((X, y))

```

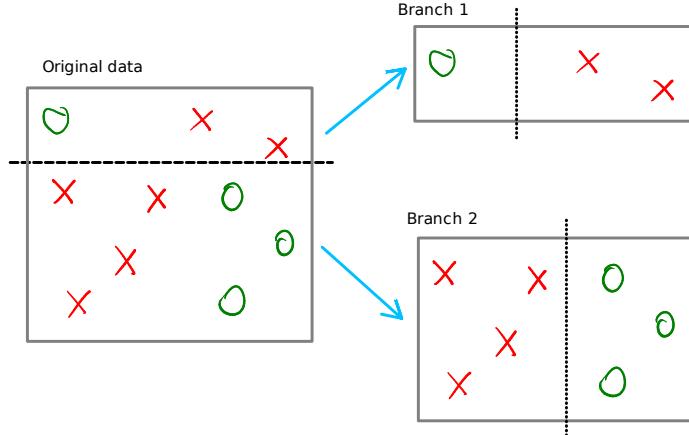
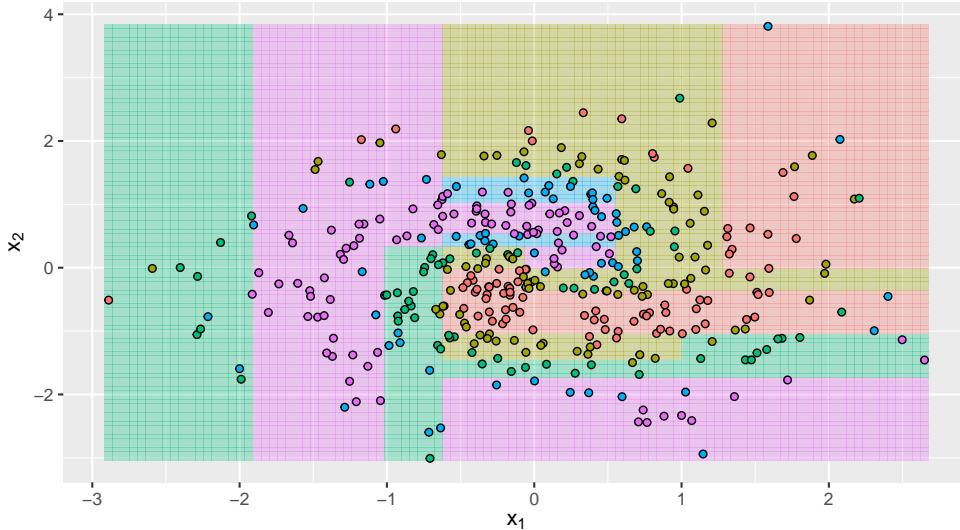


Figure 6.11: Recursive binary split. Binary refers to the fact that the algorithm always splits the original data (left) into two subsets “Branch 1” and “Branch 2” (right). Thereafter, both of these subsets are treated as new datasets, and split again. The process can be repeated many times until some kind of stopping condition is reached. This is why it is called recursive.

```

8          # the best way to split ( $X, y$ ) into two parts
9          # one corresponding to  $x_j = 0$ , the other to  $x_j = 1$ 
10         let D =  $X_{-j}$ 
11         # remove feature  $j$  from data
12         return node(j, growTree(( $D_i, y_i$ ) for  $i : x_{ij} = 0$ ),
13                      growTree(( $D_i, y_i$ ) for  $i : x_{ij} = 1$ ))

```



Example 6.4: Splitting data for decision trees: income and education

Here we use [males dataset](#) about personal characteristics and income. An example of the relevant variables in the dataset look like

school	union	ethn	maried	residence	wage
11	no	other	no	south	-0.04
14	no	other	yes	south	1.61
12	no	other	no	south	1.77
10	yes	other	yes	nothern_central	2.45

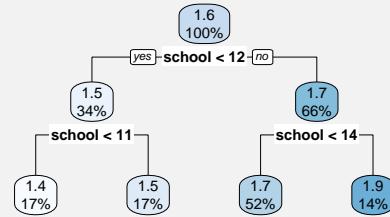
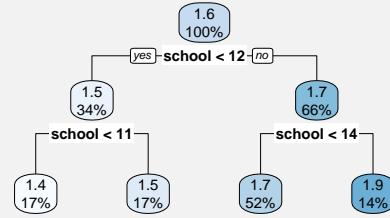
It contains 4360 observations in total. The task is to predict the *wage* (the log hourly wage), based on the other variables; hence we are talking about a regression problem. We make the trees shallow (maximum depth 2) in order to make them easy to understand.

If we only include the variable *school* (years of schooling), then we get the tree at right. The root node asks if *school* < 12, i.e. if the person has not graduated from high school. If true, the predicted salary will be \$1.5 per hour. If not, the right branch now asks if *school* < 14, i.e. if the person has taken some college education, but not even graduated from a 2-year college.

The *RMSE* of the model is 0.512.

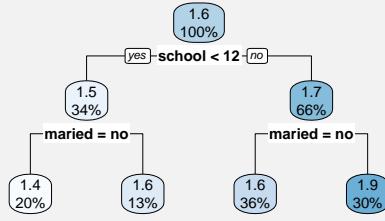
Next, we also include variable *residence*. The corresponding tree is visible at right. Perhaps surprisingly, the tree turns out to be *exactly the same!* Obviously, the *RMSE* of the model is 0.512, exactly the same value.

This is because for the first three splits (the three nodes visible on the figure) it turns out that *school* gives better splits than *residence*. More information is available, but it turns out not to be that useful.



Finally, we include all available variables. Now the tree turns out slightly different—the root node still asks about schooling, but both second-level nodes now test the marital status instead (and assign lower wage to non-married people). The splitting algorithm decides that this is more useful than to test schooling again, and indeed, *RMSE* is now slightly smaller, 0.507.

This example shows how trees work—they use the best information available (or at least what the recursive splitting thinks is the best information available). In the example above, the best information turns out to be embedded in the *school* variable. Even what is left in this variable after using some of this information (asking if someone has HS degree) is more useful than what is in *residence*. But *married* turns out to be more useful than the “depleted” *school* variable.



6.3.4 Finding the best split: entropy

In order to come up with good predictions (leaves), the leaves should be as “pure” as possible. In regression problems, it usually means that the leaves should have small variance (or RMSE).

Minimize quadratic loss inside of regions:

$$\hat{y}_j = \arg \min_{\hat{y}} \sum_{i: \mathbf{x}_i \in R_j} (\hat{y} - y_i)^2$$

First split in 1 dimension:

$$s_1 = \arg \min_s \left[\min_{\hat{y}_1} \sum_{i: x_i < s} (\hat{y} - y_i)^2 + \min_{\hat{y}_2} \sum_{i: x_i \geq s} (\hat{y} - y_i)^2 \right]$$

n -th split among d dimensions: choose, s, d to minimize

$$\min_{\hat{y}_1} \sum_{i: x_{id} < s} (\hat{y} - y_i)^2 + \min_{\hat{y}_2} \sum_{i: x_{id} \geq s} (\hat{y} - y_i)^2$$

Establishing leaf purity for categorization tasks is a little more complex. There are several methods to compute the leaf purity, below we discuss entropy-based purity.

Consider the same data as in Figure 6.11 (Figure 6.12). The left side of the figure depicts the same split as in Figure 6.11 above. In one of the branches we have one circle and two crosses, the other has three circles and four crosses. The right-hand side depicts another possible split: in the first branch we have a single circle and five crosses, and the other branch as one cross and three circles. Which of these splits is more useful?

Entropy: a measure of information in distribution.
See [Section 1.4.4 Information and Entropy](#), page 63.

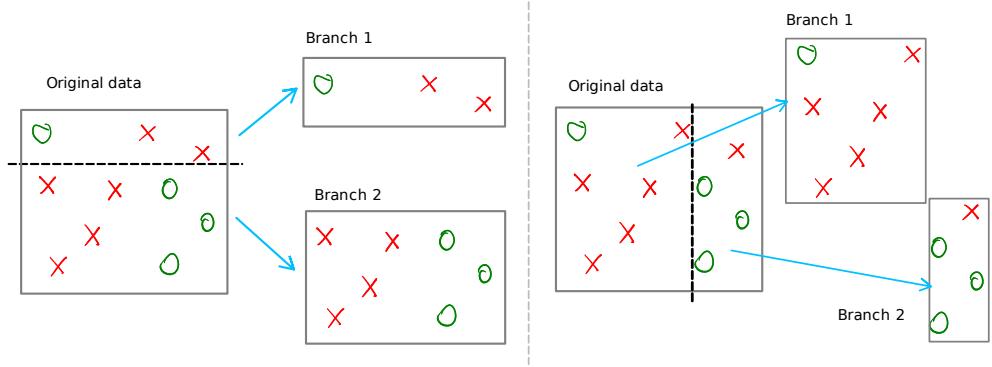


Figure 6.12: Comparing two possible splits of the same data. The entropy of the original node (6 crosses, 4 circles) is 0.971. The split at left results in entropy 0.965 and hence the gain is 0.006. The split at right gives a much larger entropy gain, see Exercise 6.3.

In this figure, it is fairly obvious that the right-hand split is better: both leaves are now fairly pure (83% and 75% respectively), while the left-hand split results in leaves with 67% and 63% of purity. However, such intuitive purity measure does not generalize well to more complex cases, including more than two categories. Instead, we should compute entropy gain of both of these splits.

As a reminder, information is defined as

$$(1.4.23): \quad H(x) = -\sum_x \Pr(x) \log_2(\Pr(x)), \quad (6.3.1)$$

and entropy is defined in (1.4.23). In order to compute the entropy gain, we first need to find the entropy in the original data. It consists of 10 data points, 4 of which are circles and 6 of which are crosses. Hence entropy is

$$\begin{aligned} H_0 &= -\Pr(circle) \log_2 \Pr(circle) - \Pr(cross) \log_2 \Pr(cross) = \\ &= -0.4 \cdot \log_2(0.4) - 0.6 \log_2(0.6) \approx 0.971. \end{aligned} \quad (6.3.2)$$

The first possible split consists of two branches: the first one of size 3 with one circle and two crosses; and the other one of size 7 with three circles and four crosses. Their corresponding entropies are

$$\begin{aligned} H_1 &= -1/3 \cdot \log_2(1/3) - 2/3 \log_2(2/3) \approx 0.918 \\ H_2 &= -3/7 \cdot \log_2(3/7) - 4/7 \log_2(4/7) \approx 0.985. \end{aligned} \quad (6.3.3)$$

The final entropy would be weighted average over these two values, where weights are the corresponding branch sizes:

$$H_{left} = 3/10 \cdot H_1 + 7/10 \cdot H_2 \approx 0.965. \quad (6.3.4)$$

Hence the split decreased entropy from 0.971 to 0.965, a gain of 0.006.

Exercise 6.3: Compute entropy gain

Compute the entropy gain at the right split of Figure 6.12.

Solution on page 430

Downsides of trees

- Trees are unstable
- Not invariant with respect to rotating data
- Capturing certain patterns requires overly complex trees

Regularizing Trees

Decision trees are easy to overfit. This manifests in too deep and complex trees that can easily achieve 100% accuracy on training data. There are three common solutions for overfitting.

1. Set maximum depth or another simple parameter. This is perhaps the simplest and most straightforward way to avoid overfitting—it only allows the tree to grow until it reaches a given depth. Besides depth, one may want to set minimum number of data points to be split, minimum leaf size, or other parameters. Setting such limits is simple and may improve computation speed. However, the resulting trees may turn out to be less efficient, as some potentially useful branches will not be considered. See Figure 6.13 for an example how setting maximum depth may cause either underfitting or overfitting.
2. Stop growing trees if split not significant. This requires a certain threshold value, in terms of MSE or entropy improvement, so when the best split will not improve the overall goodness of the model by at least this much, the tree will stop.
This approach is too shortsighted though, as a mediocre split now may make it possible to achieve a very good split later.
3. *Pruning* is a more far-sighted (albeit more complex) alternative to decisions based on split significance. The idea of *const complexity pruning* is similar to lasso regression, instead of minimizing MSE of the tree, we choose the loss function that penalizes the number of terminal nodes \mathcal{T} :

$$L(\lambda) = \sum_{m=1}^{\|\mathcal{T}\|} \sum_{i:i \in \mathcal{R}_m} (y_i - \hat{y}_i)^2 + \alpha \|\mathcal{T}\| \quad (6.3.5)$$

Thereafter we can build large trees for different α and cross-validate for the best α .

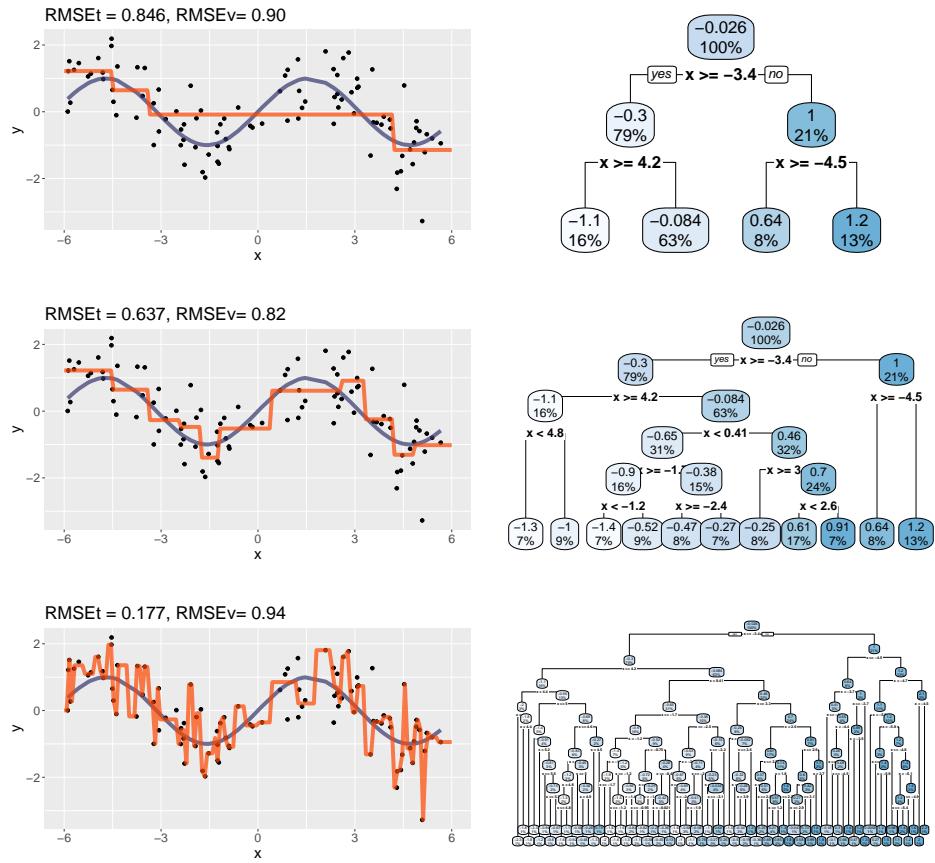


Figure 6.13: Overfitting in regression trees. The true relationship is marked with the purple line, observed values with black dots. The upper panel fits a tree of depth 2. This clearly underfits and cannot capture the wavy pattern in data. The middle panel displays a tree of depth 5. This seems about right—the tree clearly gets the main pattern but does not jump to grab every single datapoint. The lower panel shows a tree of depth 8. This one clearly overfits and attempts to catch individual datapoints. Right-hand side depicts the corresponding tree structure.

The depth-5 model has the lowest RMSE on validation data, 0.82, the deepest tree achieves the lowest RMSE on training data (0.18), but that is deception—overfitting.

6.3.5 Ensemble Methods

So far we discussed just trees–decision-making based on a single decision tree. However, it turns out one can get better estimators when combining multiple trees into “forests”. In this way we build *ensemble methods*. The idea of ensemble methods is very simple: instead of relying on a single model, we use a number of models and see what do they all predict. If they agree, this is good news. If they disagree, we just pick the solution that most of the models agree on (we do “majority voting”). In case of regression outcome we just aggregate the predictions of different models.

While ensemble methods are often based on various kinds of trees, one can do ensembles of other types of models as well, e.g. by combining different neural network models, different data sources, or different pre-processing. A major reason why ensemble methods work well with trees is that trees are typically created by the greedy splitting algorithm. As this is suboptimal (it is shortsighted), it leaves a lot of potential information behind. Ensemble methods introduce more variation in how the trees are built, and in this way help to gain some of that information back.

Bagging

Bagging (bootstrap aggregating) is basically averaging predictions from a large number of bootstrapped samples (random samples taken from the training data). The basic reason why bagging works is as follows: as our data is a random sample for the population, the trees (or other models) built based on it are random too and hence predict random results. But we can make random results to be more stable if we average a large number of them.

If we have B training sets, we can build B different trees, one for each training set, and get B different predictors $\hat{y}^1, \hat{y}^2, \dots, \hat{y}^B$. The final predictor is just the average of these, $\hat{y} = \frac{1}{B} \sum_i \hat{y}^i$ (or the majority category in case of classification). However, as we normally only have a single training set, we can rely on bootstrapping to build more training sets. In case of bagging there is no need to prune the trees, the aggregation functions as regularization.

The number of trees, B , is a hyperparameter that should be tuned, a good value may be around 100.

Random Forests

The downside with bagging is that it tends to rely on the same main features for all trees and hence gets too little variation in the model. Random forests solve this issue by adding random feature selection. Each time the bagging algorithm considers a split, it only considers K' features to use for splitting, instead of the full set of K features. This forces the individual trees to use different features. In practice, random forests are typically more precise method than bagging.

Random forests have two hyperparameters (in addition to the individual tree parameters): number of trees, and number of features K' to include into individual splits. A good choice tends to be $K' = \sqrt{K}$ where K is the original number of features in the data. In case of $K \approx 1000$ features in the original data, each split is done on a

subset of $K' \approx 30$ features only. This leaves the majority of features out, and forces the trees to use information that may otherwise not be used.

Boosting

Boosting methods are in many ways similar to bagging and random forests. Below we describe *AdaBoost*, one of the most popular boosting algorithms.

Imagine a two-class categorization task where the categories are “-1” and “1”. We can create different models (e.g. different types of trees like in case of bagging or random forest) that produce predictions $\hat{y} \in \{-1, 1\}$. For each of the model we can compute the error rate

$$e^m = \frac{1}{N} \sum_{i=1}^N \mathbb{1}(\hat{y}_i^m \neq y_i) \quad (6.3.6)$$

where m denotes the model that predicts outcome \hat{y}_i^m for observation i . Some of these are *weak classifiers* where the error rate is not much above that of a random guess, while others work much better.

Next, we combine all these predictions together not by majority voting but by a weighted sum:

$$\hat{y}_i = \text{sign} \left(\sum_{m=1}^M \alpha_m \hat{y}_i^m \right). \quad (6.3.7)$$

Importantly, α_m are model weights. Initially we can start by weighting all models in an equal fashion, but afterwards we want to give much more weight to good classifiers than to weak classifiers.

Algorithm:

1. build a simple tree
 - may just be a stump
2. predict
3. you get correct and incorrect results
4. compute weights:
 - (a) weights for this tree: better accuracy, higher weights
 - (b) weights for each observation: wrong get more weight by α .
5. repeat many times
6. your prediction is the weighted average of all trees.

Unfortunately, as is the case with other models, ensemble improve prediction accuracy at the expense of interpretability. We cannot present a bagging or random forest model as a decision tree any more.

6.4 Naïve Bayes

Naive Bayes is a classification method that is based on very simplistic (naive) independence assumption, and on the Bayes theorem. The independence assumption is unrealistic in many cases, but tremendously simplifies the computations and makes it able to handle high-dimensional cases. It turns out this tradeoff–computational simplicity against unrealistic assumptions–pays off in many types of problems.

6.4.1 Before We Begin: A Single Word–Based Bayesian Classifier

Prerequisites: [Bayes theorem 1.3.3](#)

Imagine you open your mailbox and see the following email:

I have very urgent and confidential business proposition for you. On 26th December 2004 an Oil Consultant/Contractor with the Myanmar National Petroleum Corporation, Mr. A Y Mustafa ...

Mr. A Y Mustafa died from an automobile crash ...

I am looking for a foreigner who will stand in as the next of kin to Mr. Mustafa ...

*Yours truly,
Mr. M. Lwin*

Further below it is explained how you can get \$4 million in a few days if you agree with Mr. Lwin’s “proposition”. This would make your day! (Or maybe even your life?)

But before you reply to this urgent proposal, maybe you should check if this email is spam instead? After all, the phrase “confidential business proposition” may catch your eyes as somewhat weird. Will these words help us to build a spam filter? Our final task is to build such a spam filter based on the Naive Bayes model, but before we get there, let’s build a Bayesian spam filter based on a single phrase only.

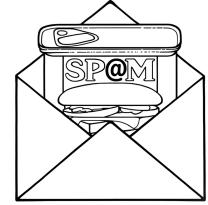
We can never be 100% certain about the correct category in common applications, so instead of directly modeling a spam/non-spam decision, we model the probability that an email that contains the phrase “confidential business proposition” (*CBP*) is spam. Formally, we are interested in the conditional probability

$$\Pr(S|CBP) \tag{6.4.1}$$

where S is the spam status ($S = 1$ for spam and $S = 0$ for no spam), and CBP is the CBP status ($CBP = 1$ if the email contains the phrase and $CBP = 0$ if it does not). As both S and CBP can have two values, we have for four probabilities in total:

$$\begin{array}{ll} \Pr(S = 0|CBP = 0) & \Pr(S = 0|CBP = 1) \\ \Pr(S = 1|CBP = 0) & \Pr(S = 1|CBP = 1). \end{array} \tag{6.4.2}$$

The first row represents the probabilities that the email is not spam ($S = 0$) while not containing the phrase ($CBP = 0$) and while containing the phrase ($CBP = 1$).



Many emails turn out to be spam.
Chesie Yu, [CC BY-NC-SA 4.0](#)

In the second row we have the corresponding probabilities for spam ($S = 1$). Exactly as in the upper row, the first probability refers to the case where the email is spam while not containing the phrase, while the second probability represents the case that an email that contains the phrase is spam. Obviously, as every email is either spam or non-spam, the corresponding probabilities must sum to unity: for emails without the phrase $\Pr(S = 0|CBP = 0) + \Pr(S = 1|CBP = 0) = 1$ and the same for emails with the phrase, $\Pr(S = 0|CBP = 1) + \Pr(S = 1|CBP = 1) = 1$. For simplicity, we often use the shorter notation (6.4.1) to represent all four probabilities.

Next, we can use Bayes theorem (1.3.4) to express the probabilities in (6.4.1):

$$\Pr(S|CBP) = \frac{\Pr(CBP|S) \cdot \Pr(S)}{\Pr(CBP)}. \quad (6.4.3)$$

As above, CBP represent the phrase status (email does contain or does not contain CBP) and S represents the spam status (email is spam or not spam), so (6.4.3) actually represents four different probabilities, exactly as does (6.4.1) above. To fix the ideas, let's focus on $\Pr(S = 1|CBP = 1)$, the probability that an email containing “confidential business proposal” is spam. This version of (6.4.3) is

$$\Pr(S = 1|CBP = 1) = \frac{\Pr(CBP = 1|S = 1) \cdot \Pr(S = 1)}{\Pr(CBP = 1)}. \quad (6.4.4)$$

Let us now go over of all the probabilities on the right-hand-side of (6.4.4).

- $\Pr(S = 1)$ is the prior, the unconditional probability of the email being spam. It is just the percentage of spam emails in our data. Note that while computing the spam percentage is simple, it requires a training dataset, a manually labeled set of emails.
- $\Pr(CBP = 1|S = 1)$ is the percentage of spam emails that contain the phrase. This is the main source of information that includes both spam and content data and allows us to improve our predictions. It is straightforward to calculate this probability by simply selecting all the spam emails and computing the percentage that contain “confidential business proposal”.
- Finally, the normalizer $\Pr(CBP)$ is just the unconditional probability that emails contain the phrase, be they spam or not. This is the simplest probability to compute, we don't even need labeled training data.

So all these probabilities can be computed easily if we have training data, a dataset of suitable size where the emails are already labeled into spam and no-spam ones. The rest is essentially just tabulating, and using the Bayes theorem as the final step.

Example 6.5: Bayesian spam filter based on a single word

Assume you have labeled training data of 1000 emails, 400 of these are spam and 600 are not spam. We focus on a single word, “viagra” in these emails. Denote the “viagra” status by $V = \mathbb{1}(\text{email contains “viagra”})$. The table below shows the counts of all types of emails:

	$V = 0$	$V = 1$	Total
$S = 0$	500	100	600
$S = 1$	150	250	400
Total	650	350	1000

Based on this table, let us compute $\Pr(S = 1|V = 1)$, the probability that an email is spam, given it contains “viagra”. When using (6.4.4), we first have to find the following probabilities:

- $\Pr(V = 1|S = 1)$, probability of “viagra” in spam emails. From the table we can see that it is $250/400 = 5/8 = 0.625$.
- The prior, $\Pr(S = 1)$, the proportion of spam emails. It is $400/1000 = 2/5 = 0.4$.
- The normalizer, $\Pr(V = 1)$, the probability to see “viagra” in emails. It is $350/1000 = 7/20 = 0.35$.

Based on these numbers we can easily compute

$$\begin{aligned} \Pr(S = 1|V = 1) &= \frac{\Pr(V = 1|S = 1) \cdot \Pr(S = 1)}{\Pr(V = 1)} = \\ &= \frac{\frac{5}{8} \cdot \frac{2}{5}}{\frac{7}{20}} = \frac{5}{7} \approx 0.714. \quad (6.4.5) \end{aligned}$$

The Bayesian update based on a single word made the final probability 0.714, close to 2-fold increase over the prior $\Pr(S) = 0.4$. Such substantial improvement was only possible because in this example data the word “viagra” is very common in spam emails. If a word is rare, it can only identify a small number of spam emails. If it is rare but very spam-specific, it gives us large precision but the recall will remain low as most of the spam emails are left unidentified.

Exercise 6.4: Probability of spam given no “viagra”

Use the data as in the Example 6.5. Compute the probability $\Pr(S = 0|V = 0)$, probability that the email is not spam if it does not contain “viagra”. How much larger is the posterior compared to the prior?

Solution on page 430.

6.4.2 Smoothing: how to compute probabilities with too few data

In the examples above we did not talk much about how to compute the probabilities, such as $\Pr(W = 1|S = 1)$. Intuitively, one may want just to use the proportion of

documents where the word is present (as we did above):

$$\Pr(W = 1|S = 1) = \frac{N_{W=1|S=1}}{N_{S=1}} \quad (6.4.6)$$

where $N_{W=1|S=1}$ is the count of spam-emails where the word is present, and $N_{S=1}$ is the total number of spam emails. This intuitive approach is justified if the counts are large. But if we observe just a few cases of the word, these probabilities may be far from the truth. A common manifestation of this problem is the case where we observe only a single instance of the word. Obviously, this means it only belongs to a single class, say spam. Now every new email that contains that word will have probability (from (6.4.4))

$$\Pr(S = 1|W = 1) = \frac{\Pr(W = 1|S = 1) \cdot \Pr(S = 1)}{\Pr(W = 1)} = \frac{\frac{1}{N_{S=1}} \cdot \frac{N_{S=1}}{N}}{\frac{1}{N}} = 1 \quad (6.4.7)$$

where we denote the total number of emails by N , and spam emails by $N_{S=1}$. In a similar fashion, the probability that the email containing the word is non-spam is

$$\Pr(S = 0|W = 1) = \frac{\Pr(W = 1|S = 0) \cdot \Pr(S = 0)}{\Pr(W = 1)} = \frac{\frac{0}{N_{S=0}} \cdot \frac{N_{S=0}}{N}}{\frac{1}{N}} = 0 \quad (6.4.8)$$

where $N_{S=0}$ is the number of non-spam emails. For instance, imagine we have seen the word “viagra” just once, in a spam email. Hence every new email that contains this word will be categorized as spam because $\Pr(viagra = 1|S = 0) = 0$. No buts, no ifs.

But typically it is not just a single word that occurs in our corpus only once. Imagine the word “conference” also appears only once, in a valid email. So now we categorize every message containing “conference” unambiguously as valid, and every message containing the word “viagra” as spam. But what should we do with an email that contains both “viagra” and “conference”? One word will unambiguously say it is spam, and the other word will say it is valid.

Obviously, it is problematic to rely on a single rare value for categorization. Rare words are, by definition, rare, and hence may occur in one or another category just by chance. The problem arises because we are drawing too strong conclusions from too little data. Clearly, a single “viagra” and a single “conference” is not enough to claim we have 100% certainty to categorize the email. As this certainty originates from the probability calculation (6.4.6), that approach must be incomplete. Intuitively, it is easy to see what is wrong with that formula—it does not take into account the sample size. If we find 0 valid ones out of total $N = 1$ emails that contain “viagra” then (6.4.6) will give the corresponding probability being 0. If we find 0 valid emails out of $N = 1000$ such emails then (6.4.6) will still result in probability 0. But in the latter case we clearly have much more reliable result.

A popular solution to this problem is called smoothing. Smoothing is equivalent to Bayesian estimation¹ of the probabilities. Instead of taking the strictly frequentist

¹“Bayesian” here refers to Bayesian statistics, not to the Naive Bayes estimator. Naive Bayes is based on Bayesian theorem but Bayesian statistics includes much more than this method.

approach (6.4.6),² we should take a Bayesian approach where we include a prior for the probability of interest. A Bayesian prior (beta-prior) is equivalent to adding a small positive number to the counts.

Take the spam example. Let's the prior for $\Pr(W = 1|S = 1) = 0.5$, i.e. we assume the word W is equally likely to be present or absent in spam emails. We can assume that for every word we analyze, we have two additional spam emails: one that contains the word, and one that does not contain the word. Hence we have seen $N_{W=1|S=1} + 1$ spam emails with the word, out of $N_S + 2$ in total, and instead of (6.4.6) we have the probability is accordingly

$$\Pr(W = 1|S = 1) = \frac{N_{W=1|S=1} + 1}{N_{S=1} + 2}. \quad (6.4.9)$$

Now if none of the spam emails contained the word “deadline”, then we have

$$\Pr(\text{deadline} = 1|S = 1) = \frac{1}{N_{S=1} + 2} > 0. \quad (6.4.10)$$

This is not zero, and hence we cannot say that every single mention of “deadline” unambiguously shows that it is a valid email.

But we do not have to assume we have seen exactly one email that contains and does not contain the word. We can instead assume we have seen α emails where $\alpha > 0$ does not have to be an integer. So we generalize (6.4.9) as

$$\Pr(W = 1|S = 1) = \frac{N_{W=1|S=1} + \alpha}{N_{S=1} + 2\alpha}. \quad (6.4.11)$$

Now in case we have not seen the word in spam, the corresponding probability will not be 0, but $\alpha/(N_{S=1} + 2\alpha)$ instead. α describes our confidence in the prior, small α mean little confidence and large α means a lot of confidence. Obviously, the more observations we collect (the larger N), the less α matters and the result is close to the pure frequentist probability $N_{W=1|S=1}/N_S$. Data overrides the prior. But it is never exactly 0 and hence does not corrupt the estimator. Note that the term 2α in denominator that takes into account that our prior was 1/2: if we haven't seen any example from spam, i.e. $N_{S=1} = 0$, then $\Pr(W = 1|S = 1) = \alpha/(2\alpha) = 1/2$, i.e. the prior. As the denominator in (6.4.11) is typically large (in thousands), the term 2α in denominator plays a little role. The term that matters is α in the numerator.

Finally, we do not have to use prior 1/2, one can use different priors for different words and classes, and also have priors for categories. See [Murphy \(2012, p. 87\)](#) for more information.

6.4.3 Naive Bayes Classifier

Prerequisites: Bayes theorem, independent events: [Section 1.3.3 Conditional Probability and Bayes Theorem](#), page [25](#)

²This claim is a bit misleading. While frequentists may be happy with (6.4.6), they are not happy with how we use this probability afterwards. In particular, ignoring uncertainty of computed values is not a correct way of doing frequentist statistics.

Obviously we should not base our spam filter just on a single word. It would not catch much spam, and it may remove good emails that for some reason contain a similar expression. A much better approach would be to look at many words, potentially at all the words we have learned in the training data. But let's start with adding another phrase, *lottery winner* (*LW*), to our spam filter. So now our model contains two indicators, *CBP* and *LW*, both of which can be 0 or 1, and the probability for spam can now be expressed as

$$\Pr(S = 1 | CBP, LW) = \frac{\Pr(CBP, LW | S = 1) \cdot \Pr(S = 1)}{\Pr(CBP, LW)}. \quad (6.4.12)$$

Here we have simplified the notation a little bit:

- The prior, $\Pr(S = 1)$, is unchanged. This is just the unconditional probability that an email is spam.
- $\Pr(CBP, LW | S = 1)$ can be any of these four probabilities, depending on which phrases the email contains:
 1. $\Pr(CBP = 0, LW = 0 | S = 1)$: probability that a spam email does contain neither “confidential business proposition” nor “lottery winner”.
 2. $\Pr(CBP = 0, LW = 1 | S = 1)$: probability that a spam email does not contain “confidential business proposition” but contains “lottery winner”.
 3. $\Pr(CBP = 1, LW = 0 | S = 1)$: probability that a spam email contains “confidential business proposition” but does not contain “lottery winner”.
 4. $\Pr(CBP = 1, LW = 1 | S = 1)$: probability that a spam email contains both “confidential business proposition” and “lottery winner”.
- Similar reasoning also applies to the normalizer $\Pr(CBP, LW)$. We have to pick one of the four possible normalizers depending on which of these expression occur in the email.
- And finally, the question of interest itself, $\Pr(S = 1 | CBP, LW)$, also contains four possibilities: it is the probability that the email is spam, given it contains or does not contain any combination of “confidential business proposition” and “lottery winner”.

So in order to use the Bayesian approach with two phrases, we need 9 probabilities in all: one for the prior, four for the conditional probabilities, and four for the normalizers. All these should be computed from the labeled training data. This is straightforward to do, given we have a labeled training data set of a suitable size.

Example 6.6: Bayesian spam filter with two phrases

Assume we have data about 1000 emails that may or may not contain phrases *confidential business proposition* (*CBP*) and *lottery winner* (*LW*). We count the spam/non-spam emails that contain or do not contain either of these phrases and get:

	$CBP = 0$	$CBP = 0$	$CBP = 1$	$CBP = 1$	
	$LW = 0$	$LW = 1$	$LW = 0$	$LW = 1$	Total
$S = 0$	400	100	60	40	600
$S = 1$	50	100	140	110	400
Total	450	200	200	150	1000

Let's compute the probability that an email that contains CBP but does not contain LW is spam, $\Pr(S = 1 | CBP = 1, LW = 0)$.

From Bayes' theorem,

$$\Pr(S = 1 | CBP = 1, LW = 0) = \frac{\Pr(CBP = 1, LW = 0 | S = 1) \Pr(S = 1)}{\Pr(CBP = 1, LW = 0)}.$$

From the table, we can find that

$$\Pr(CBP = 1, LW = 0 | S = 1) = 60/600 = 1/10$$

$$\Pr(S = 1) = 600/1000 = 3/5$$

$$\Pr(CBP = 1, LW = 0) = 200/1000 = 2/10$$

Hence the probability of interest

$$\Pr(S = 1 | CBP = 1, LW = 0) = \frac{\frac{1}{10} \cdot \frac{3}{5}}{\frac{2}{10}} = 3/10 = 0.3$$

Again, this can be calculated directly as $60/200$, but that approach will not work for Naive Bayes below.

But unfortunately the story does not stop here. If we use three phrases instead of two (e.g. we add “million dollars”, we have 8 combinations for both the conditional probability and for the normalizer: one set of four where “million dollars” is present and another set of four where it is absent. In case of four phrases we have 16 combinations, and so on. It is easy to see that when we analyze K phrases or words, we have 2^K different combinations. In case of a realistic text, K may easily exceed 10,000. So a complete Bayesian approach will involve $\sim 2^{10,000}$ different combinations. This is infeasible to do.³ We run into the curse of dimensionality.

But it is easy to avoid the curse of dimensionality by introducing additional assumptions. The most popular one, and the one that is the basis of the Naive Bayes method, assumes that the presence of words in data is independent, given the email is spam or no spam. So in case of our two phrases we can write the conditional

In case of independent events, the joint probability is the product of their individual probabilities:
 $\Pr(A, B) = \Pr(A) \cdot \Pr(B)$.
 See [Section 1.3.3 Independent Events](#), page 35.

³It is often not appreciated just how incredibly big are these numbers. For comparison, the age of Universe is approximately $10^{18} \approx 2^{54}$ seconds, and the visible universe contains $\sim 10^{90} \approx 2^{300}$ elementary particles. $2^{10,000}$ is just way way beyond of what fits into our universe, so there is no way we can ever collect and analyze this much data. (As before, we do not talk about quantum computing).

probability as

$$\Pr(CBP, LW|S = 1) = \Pr(CBP|S = 1) \cdot \Pr(LW|S = 1) \quad (6.4.13)$$

for spam emails and

$$\Pr(CBP, LW|S = 0) = \Pr(CBP|S = 0) \cdot \Pr(LW|S = 0) \quad (6.4.14)$$

for valid emails. As explained above, this formula represents four different probabilities for all four different combinations of presence of these two phrases in spam emails. To fix the ideas, let's look at $\Pr(CBP = 1, LW = 1|S = 1)$, i.e. probability that a spam emails contains both these phrases. The assumption means that if e.g. $\Pr(CBP = 1|S = 1) = 0.1$ and $\Pr(LW = 1|S = 1) = 0.1$, then $\Pr(CBP = 1, LW = 1|S = 1) = 0.01$. If 10% of spam emails contain *CBP* and 10% of spam emails contain *LW*, then 1% of spam emails contain both phrases. Our assumption claims that this is true for spam emails, and that it is also true for non-spam emails, but not necessarily for all emails when we combine both spam and valid ones. This is what the conditioning on spam status S does in (6.4.13).

Let's first look at the good news. Where we formerly had to calculate four different probabilities, one for each combination of *CBP* and *LW*, now we only need two: one for *CBP* and one for *LW*. And even better, when we add another feature, we only need one additional probability. So in case of 10,000-word vocabulary, we only need 10,000 different probabilities. This is fast and trivially fits into the computer memory. So we have circumvented the curse of dimensionality in case of the conditional probability in (6.4.12). The normalizer still has too many combinations but we'll discuss what to do with it below.

But what does this assumption mean and why is it called “naive”? Independence of random variables means that realization of one RV does not contain information about realizations of the other one. If we learn about the first phrase, “confidential business proposition”, this does not tell us anything about the presence of the other phrase, “lottery winner” in case we are just looking at the spam emails. So the method ignores the fact that the words may be correlated, and not just correlated but the correlation may carry different meaning than the individual words. For instance, when tokenizing “New York” into individual words, we treat the resulting “new” and “york” as separate independent identities. The model does not understand that “New York” carries a distinct meaning, very different from what these two single words carry. These are the bad news, and this is why the approach is called “naive”. But here the good news outweigh the bad ones, as we now have a model that can actually be computed.

Let us now build a full two-class Naive Bayes model. Assume we have a vocabulary of size K of words W_1, W_2, \dots, W_K where W_i denotes the presence (if $W_i = 1$) or absence (if $W_i = 0$) of word i in the document. We are looking for a category S where $S \in \{0,1\}$ denotes whether the document is spam or not. We can write the independence (naive) assumption as

$$W_i \perp\!\!\!\perp W_j | S. \quad (6.4.15)$$

This assumption means we describe the words as picked randomly from different distributions, one for $S = 0$ and one for $S = 1$. As we discussed above, the assumption

is unrealistic as it ignores the relationship between words but it resolves the curse of dimensionality problem. In fact, the Naïve Bayes method scales surprisingly well.

Remember that by definition of [independent events](#) their joint probability is equal to the product of individual probabilities. For two words we have $\Pr(W_1, W_2) = \Pr(W_1) \cdot \Pr(W_2)$ (see [1.3.7](#)). The same applies for more than two probabilities, $\Pr(W_1, W_2, \dots, W_K) = \prod_{j=1}^K \Pr(W_j)$ (see [\(1.3.8\)](#)). It also applies for conditionally independent probabilities as conditional probabilities are just word probabilities either in spam or non-spam category, so we can write

$$\Pr(W_1, W_2, \dots, W_K | S) = \prod_{j=1}^K \Pr(W_j | S). \quad (6.4.16)$$

This is the conditional probability we need in the Bayesian approach if we include all K words, and above we discussed that when using the independence assumption, we only have to find $2K$ probabilities from the training data, $\Pr(W_j = 1 | S)$ and $\Pr(W_j = 0 | S)$ for $j = 1, \dots, K$, in order to compute the joint probability.

Let us now solve the spam email problem by using the independence assumption [\(6.4.16\)](#). We want to estimate $\Pr(S = 1 | W_1, W_2, \dots, W_K)$, the probability of email being spam, given the words it contains or does not contain. We start by expressing this probability through Bayes theorem:

$$\begin{aligned} \Pr(S = 1 | W_1, W_2, \dots, W_K) &= \frac{\Pr(W_1, W_2, \dots, W_K | S = 1) \cdot \Pr(S = 1)}{\Pr(W_1, W_2, \dots, W_K)} = \\ &= (\text{here we use the independence assumption (6.4.16)}) = \\ &= \frac{\Pr(W_1 | S = 1) \cdot \Pr(W_2 | S = 1) \cdots \Pr(W_K | S = 1) \cdot \Pr(S = 1)}{\Pr(W_1, W_2, \dots, W_K)} = \\ &= \frac{\Pr(S = 1) \cdot \prod_{j=1}^K \Pr(W_j | S = 1)}{\Pr(W_1, W_2, \dots, W_K)}. \end{aligned} \quad (6.4.17)$$

Now the numerator is factorized into a product of $K + 1$ factors in the form of $\Pr(W_k | S)$ and $\Pr(S = 1)$. These are just K conditional probabilities in the form *probability the word k exists/does not exist in spam emails*. So we need just K numbers, as probability of absence of the word is just $1 - \text{probability of presence of it}$. These conditional probabilities can easily be calculated based on word counts in spam/non-spam documents.

But this only solves one part of our problem: the normalizer $\Pr(W_1, W_2, \dots, W_K)$ is still there, and it is still intractable.⁴ But fortunately we can eliminate the normalizer in a simple way. Let's also compute the probability that the email is not spam. Using the same approach as in [\(6.4.17\)](#), we get

$$\Pr(S = 0 | W_1, W_2, \dots, W_K) = \frac{\Pr(S = 0) \cdot \prod_{j=1}^K \Pr(W_j | S = 0)}{\Pr(W_1, W_2, \dots, W_K)}. \quad (6.4.18)$$

⁴Note that we assume independence of *conditional* distribution $W_i \perp\!\!\!\perp W_j | S$, not independence of unconditional distribution which would be written $W_i \perp\!\!\!\perp W_j$. The former means that words are independent in each class, spam and non-spam; the latter means that the words are independent when combining both classes. The latter is not a result of the former.

As above, we have a numerator, a product of K word frequencies in non-spam emails and the prior $\Pr(S = 0)$, and an intractable normalizer in the denominator. But note that these two expressions, (6.4.17) and (6.4.18), contain exactly the same normalizer. Even more, as the normalizer is a probability, it must be positive (between zero and one). So we can just leave the normalizer out and still compare these probabilities! However, if we leave the normalizer out in expressions (6.4.17) and (6.4.18) then the results are not probabilities any more. These are now called *likelihoods*. Likelihoods are not valid probabilities, in particular, they do not sum to unity, and they may exceed 1. So we cannot interpret the results as probabilities but we can still say that the largest likelihood corresponds to the largest probability. If likelihood for spam is larger, we call this email spam, if the likelihood for non-spam is larger we say it is non-spam.

So we can now follow these steps to predict whether the email is spam. All of it can be done on training data.

1. Compute the priors $\Pr(S = 0)$ and $\Pr(S = 1)$.
2. For each word W in the vocabulary, compute the conditional probabilities $\Pr(W = 1|S = 1)$ and $\Pr(W = 1|S = 0)$.
3. Compute the numerators (likelihoods) of the naive Bayes formula (6.4.18)

$$\mathcal{L}(S = 1|W_1, W_2, \dots, W_K) = \Pr(S = 1) \cdot \prod_{j=1}^K \Pr(W_j|S = 1) \quad \text{and} \quad (6.4.19)$$

$$\mathcal{L}(S = 0|W_1, W_2, \dots, W_K) = \Pr(S = 0) \cdot \prod_{j=1}^K \Pr(W_j|S = 0)$$

where $\mathcal{L}(\cdot)$ denotes the corresponding likelihood.

4. Which likelihood is larger, $\mathcal{L}(S = 1|\mathbf{W})$ or $\mathcal{L}(S = 0|\mathbf{W})$? (\mathbf{W} denotes a vector of all vocabulary words, $\mathbf{W} = (W_1, W_2, \dots, W_K)^T$.) This is the prediction. If likelihood for spam is larger we have a spam email, if non-spam likelihood is larger, it is not a spam.

In this example we only consider the presence of words $\Pr(W_j = 1|S = 1)$, but one may also add information about absence of words $\Pr(W_j = 0|S = 1) = 1 - \Pr(W_j = 1|S = 1)$. Obviously, these probabilities differ for spam and non-spam cases.

In practice, it is better to work with logarithms of likelihoods (*log-likelihoods*) instead of the likelihoods as the latter typically contain too small numbers. The corresponding log-likelihoods are

$$\begin{aligned} \ell(S = 1|\mathbf{W}) &\equiv \log \mathcal{L}(S = 1|\mathbf{W}) = \log \Pr(S = 1) + \sum_{j=1}^K \log \Pr(W_j|S = 1) \\ \ell(S = 0|\mathbf{W}) &\equiv \log \mathcal{L}(S = 0|\mathbf{W}) = \log \Pr(S = 0) + \sum_{j=1}^K \log \Pr(W_j|S = 0). \end{aligned} \quad (6.4.20)$$

The probability $\Pr(W_j|S = 1)$ in the sum above denotes two probabilities: $\Pr(W_j = 1|S = 1)$ and $\Pr(W_j = 0|S = 1)$ and we have to pick the one that corresponds to the message, the former if the word is there and the latter if it is not there. However, it is often easier to just look at the presence of words and ignore the information contained in their absence. In that case (6.4.20) transforms to

$$\ell(S = 1|\mathbf{W}) = \log \Pr(S = 1) + \sum_{j=1}^K \log \Pr(W_j = 1|S = 1) \cdot \mathbb{1}(W_j = 1)$$

and

$$\ell(S = 0|\mathbf{W}) = \log \Pr(S = 0) + \sum_{j=1}^K \log \Pr(W_j = 1|S = 0) \cdot \mathbb{1}(W_j = 1) \quad (6.4.21)$$

where $\mathbb{1}(W_j = 1)$ is the indicator function. So we only include the log-probabilities for words that are actually present in the message. We follow this approach below.

Normally the prediction will be the category that corresponds to the largest probability, and the category with the largest probability is the one with the largest log-likelihood. So we can write both equations in (6.4.20) as

$$\hat{S} = \arg \max_S \ell(S) = \log \Pr(S) + \sum_{j=1}^K \log \Pr(W_j = 1|S) \cdot \mathbb{1}(W_j = 1) \quad (6.4.22)$$

where \hat{S} means the predicted category.

Let us repeat the algorithm above for log-likelihood. To compute Naive Bayes, we need:

1. Compute the log prior probabilities $\log \Pr(S = 0)$ and $\log \Pr(S = 1)$.

This amounts to two probabilities.

2. For each word W in the vocabulary, compute the log conditional probabilities $\log \Pr(W = 1|S = 1)$ and $\log \Pr(W = 1|S = 0)$.

This is two probabilities for each word, or $2 \cdot K$ probabilities in total.

3. Compute the Naive Bayes log-likelihoods (6.4.20)

$$\ell(S = 1|\mathbf{W}) = \log \Pr(S = 1) + \sum_{j=1}^K \log \Pr(W_j = 1|S = 1) \cdot \mathbb{1}(W_j = 1)$$

and

$$\ell(S = 0|\mathbf{W}) = \log \Pr(S = 0) + \sum_{j=1}^K \log \Pr(W_j = 1|S = 0) \cdot \mathbb{1}(W_j = 1) \quad (6.4.23)$$

where $\ell(\cdot)$ denotes the corresponding log-likelihood.

4. Which log-likelihood is larger, $\ell(S = 1|\mathbf{W})$ or $\ell(S = 0|\mathbf{W})$? This is the prediction.

In case of typical texts where the vocabulary size is $\sim 10,000$ we have to compute tens of thousand of log-probabilities. This is an easy task for modern computers given we have suitable training data.

Example 6.7: Email classification

Assume we have categorized the following emails as spam/no-spam:

1. *viagra is good in life*: spam
2. *life is good*: no spam
3. *viagra in life*: no spam

Hence the training data tells us that the unconditional probability of spam $\Pr(S = 1) = 1/3$ and the unconditional probability of non-spam $\Pr(S = 0) = 2/3$.

These three emails contain vocabulary (in alphabetical order) “good”, “in”, “is”, “life”, “viagra”; and the corresponding [document-term-matrix](#) (see Section 7.3.3) is in table below:

	good	in	is	life	viagra
\mathbf{x}_1	1.0	1.0	1.0	1.0	1.0
\mathbf{x}_2	1.0	0.0	1.0	1.0	0.0
\mathbf{x}_3	0.0	1.0	0.0	1.0	1.0
N_W	2.0	2.0	2.0	3.0	2.0
$\Pr(W = 1 S = 1)$	1.0	1.0	1.0	1.0	1.0
$\Pr(W = 1 S = 0)$	0.5	0.5	0.5	1.0	0.5

Table 6.3: DTM of the three example emails (rows \mathbf{x}_1 , \mathbf{x}_2 and \mathbf{x}_3), the corresponding word counts (N_W), and conditional probabilities of word in spam ($\Pr(W = 1|S = 1)$) and no-spam ($\Pr(W = 1|S = 0)$) emails.

Now we receive a new email: “no viagra no life”. We want to categorize it based on our training data above. We convert it into BOW representation using the existing vocabulary:

	good	in	is	life	viagra
\mathbf{x}_4	0	0	0	1	1

Table 6.4: The new email as BOW. Note that the word “no” is missing in the training vocabulary. We ignore it here as we have no way of telling what would the corresponding probabilities be.

These tables together are sufficient to compute the likelihoods. We also only analyze the presence of words ($\Pr(W = 1|S)$ for “life” and “viagra”) and leave out the word absence-related information ($\Pr(W = 0|S)$ for “good”, “in” and “is”). Although we normally prefer log-likelihood, let’s first do the likelihoods.

The likelihood for spam is:

$$\begin{aligned}\mathcal{L}(S = 1|\mathbf{x}_4) &= \\ &= \Pr(S = 1) \times \Pr(\text{life} = 1|S = 1) \times \Pr(\text{viagra} = 1|S = 1) = \\ &= 0.333 \times 1 \times 1 = 0.333.\end{aligned}\quad (6.4.24)$$

There are only two known words present in the new message, hence we have only three terms in (6.4.24): the prior $\Pr(S = 1)$ and the conditional probabilities for each of these words. Next we compute the likelihood for non spam:

$$\begin{aligned}\mathcal{L}(S = 0|\mathbf{x}_4) &= \\ &= \Pr(S = 0) \times \Pr(\text{life} = 1|S = 0) \times \Pr(\text{viagra} = 1|S = 0) = \\ &= 0.667 \times 1 \times 0.5 = 0.333.\end{aligned}\quad (6.4.25)$$

Our method gives a tie as both likelihoods are $1/3$.

We can repeat the exercise with log-likelihood, this is the method we normally use. First we compute log likelihood for spam:

$$\begin{aligned}\ell(S = 1|\mathbf{x}_4) &= \\ &= \log \Pr(S = 1) + \log \Pr(\text{life} = 1|S = 1) + \log \Pr(\text{viagra} = 1|S = 1) = \\ &= -1.099 + 0 + 0 = -1.099,\end{aligned}\quad (6.4.26)$$

and the likelihood for no spam:

$$\begin{aligned}\ell(S = 0|\mathbf{x}_4) &= \\ &= \log \Pr(S = 0) + \log \Pr(\text{life} = 1|S = 0) + \log \Pr(\text{viagra} = 1|S = 0) = \\ &= -0.4050 - 0.693 = -1.099.\end{aligned}\quad (6.4.27)$$

The conclusion is the same as in case of likelihoods, it is a no surprise as we are just looking at the logs of the same numbers.

Exercise 6.5: Categorize using Naïve Bayes

Use the training data in Example 6.7, and categorize the sentence “life is life”.

Solution on page 431.

There is an additional advantage of using log-likelihood instead of likelihood. Namely, the log-likelihood in (6.4.22) can be computed using a matrix product. The log-likelihood for a single email can be expressed as

$$\ell(S) = \log \Pr(S) + \sum_{j=1}^K \mathbb{1}(W_j = 1) \cdot \log \Pr(W_j = 1|S).\quad (6.4.28)$$

Denote by \mathbf{W} the vector of presence (1)/absence (0) of each word in the vocabulary, and by \mathbf{P}_S the vector of log probabilities $\log \Pr(W = 1|S)$. Now we can write the

Remember: $\mathbf{W}^\top \cdot \mathbf{P}_S = W_1 \cdot P_{S1} + W_2 \cdot P_{S2} + \dots + W_K \cdot P_{SK}$.
See [Section 2.3.2 Vector multiplication as matrix product](#), page 112.

log-likelihood as

$$\ell(S) = \log \Pr(S) + \mathbf{W}^\top \cdot \log \mathbf{P}_S. \quad (6.4.29)$$

Here we compute the likelihood using a single addition and a vector product. But note that it is only the vector \mathbf{W} that depends on the email, the word log-probabilities \mathbf{lP}_S do not. Hence we can compute all log-likelihoods by a single matrix product

$$\ell(\mathbf{S}) = \log \Pr(S) + \mathbf{W} \cdot \log \mathbf{P}_S \quad (6.4.30)$$

where \mathbf{W} is just the document-term matrix.

It is very easy to generalize Naive Bayes to more than two classes—you just need to compute the log-likelihoods for each class, and then to pick the class with the largest log-likelihood.

Naive Bayes almost always requires smoothing. In typical applications there are many rare words that only occur in one or another class. Let's imagine “viagra” only occurs in a spam email, and “deadline” only in a valid email. If we calculate the word probabilities without smoothing using (6.4.6), then we have $\Pr(viagra = 1|S = 1) > 0$ and $\Pr(viagra = 1|S = 0) = 0$, and $\Pr(deadline = 1|S = 1) = 0$ and $\Pr(deadline = 1|S = 0) > 0$. Now whenever you encounter a document that contains “viagra”, we have $\mathcal{L}(S = 0) = 0$ (see (6.4.19)) and hence it will be unambiguously categorized as spam; the opposite is true for “deadline”. The other words in the text play no role because zero remains zero even when multiplied by many other positive probabilities. But intuitively, how can we base our prediction on a single observation of a single word, while ignoring everything else in the message, and claim we are 100% certain in our conclusion? Note that replacing likelihood by log-likelihood only shifts the problem from zero-likelihood to minus-infinity-log-likelihood and does not offer any solution.

Additionally, the NB classifier fails completely if the text to be categorized contains two rare words, each pointing to a different class, like both “viagra” and “deadline” in the example above. In that case all likelihoods will be zero and the prediction will either be arbitrary, or undefined. Note that more data is not a solution to the rare word problem: with more data, one also includes words that are increasingly rare, so we always have many-many words that are represented just 1-2 times in the corpus.

Smoothing addresses both of these issues by effectively replacing zero probability with a small positive number. In practical applications, the smoothing parameter α (see (6.4.11)) is a hyperparameter that should be tuned using cross-validation or another similar approach. A good value for α tends to be small, typically much smaller than 1.

Example 6.8: Email classification with smoothing

Let us revisit the example 6.7 and add smoothing to that model. Let's pick $\alpha = 0.2$. We have the following data

	good	in	is	life	viagra
x_1	1.00	1.00	1.00	1.00	1.00
x_2	1.00	0.00	1.00	1.00	0.00
x_3	0.00	1.00	0.00	1.00	1.00

Table 6.5: DTM of the three example emails from example 6.7. The first row (the first email) is spam, the following two are not spam.

However, now we have to adjust the way we compute the probability by using 6.4.11. For *good* we get

$$\begin{aligned} \Pr(good = 0|S = 0) &= \frac{1 + \alpha}{2 + 2\alpha} = 1.2/2.4 = 0.5 \\ \Pr(good = 1|S = 0) &= \frac{1 + \alpha}{2 + 2\alpha} = 1.2/2.4 = 0.5 \\ \Pr(good = 0|S = 1) &= \frac{0 + \alpha}{1 + 2\alpha} = 0.2/1.4 \approx 0.143 \\ \Pr(good = 1|S = 1) &= \frac{1 + \alpha}{1 + 2\alpha} = 1.2/1.4 \approx 0.857. \end{aligned} \tag{6.4.31}$$

Remember that in the case of no smoothing in example 6.7 the probabilities were $\Pr(good = 1|S = 0) = 0.5$ and $\Pr(good = 1|S = 1) = 0.5$. Now the former, 0.5, remains unchanged while the latter is moved away from the original extreme value 0. It is easy to see that this type of smoothing moves the calculated probabilities toward 0.5, the prior, the uninformative “middle ground”.^a The fewer observations we have, the larger is the shift. This process makes intuitively sense: the fewer observations we have the less certain we are in the calculated probabilities, and the more we are willing to admit that we don’t know well what do these words mean. “We don’t know” is a way to say that we should pick probabilities close to 0.5.

It is also easy to see why did we write 2α in the denominator. One α applies to the presence, and another to the absence of the word. This ensures that the probability of presence and absence of the word sum to unity.

The next table shows the smoothed probabilities:

	good	in	is	life	viagra
$\Pr(W = 1 S = 0)$	0.50	0.50	0.50	0.92	0.50
$\Pr(W = 1 S = 1)$	0.86	0.86	0.86	0.86	0.86

Table 6.6: Smoothed probabilities for the DTM above for $\alpha = 0.2$.

Now we continue the process in exactly the same way as earlier, just using the smoothed probabilities instead of the original ones.

^aYou can see this by taking a very large α value, say 100.

6.5 Support Vector Machines

Support Vector Machines (SVM-s) are simple models that can capture a complex decision boundary. Figure 6.14 shows dots of two colors, arranged in a yin-yang pattern. The linear decision boundary of logistic regression fails to capture the wavy boundary between the gold and purple dots (see Figure 5.4). SVM with liner kernel closely resembles the logistic regression. But SVM can represent it reasonably well, given one choose a more powerful kernel, in this figure both polynomial with degree 3 and radial kernel will do.

Note that it is, strictly speaking, not correct to say that logistic regression cannot capture such a complex boundary. It can, given we introduce suitable functions of the features, e.g. a series of polynomials or splines. However, this is not what common logistic regression implementation and applications do.

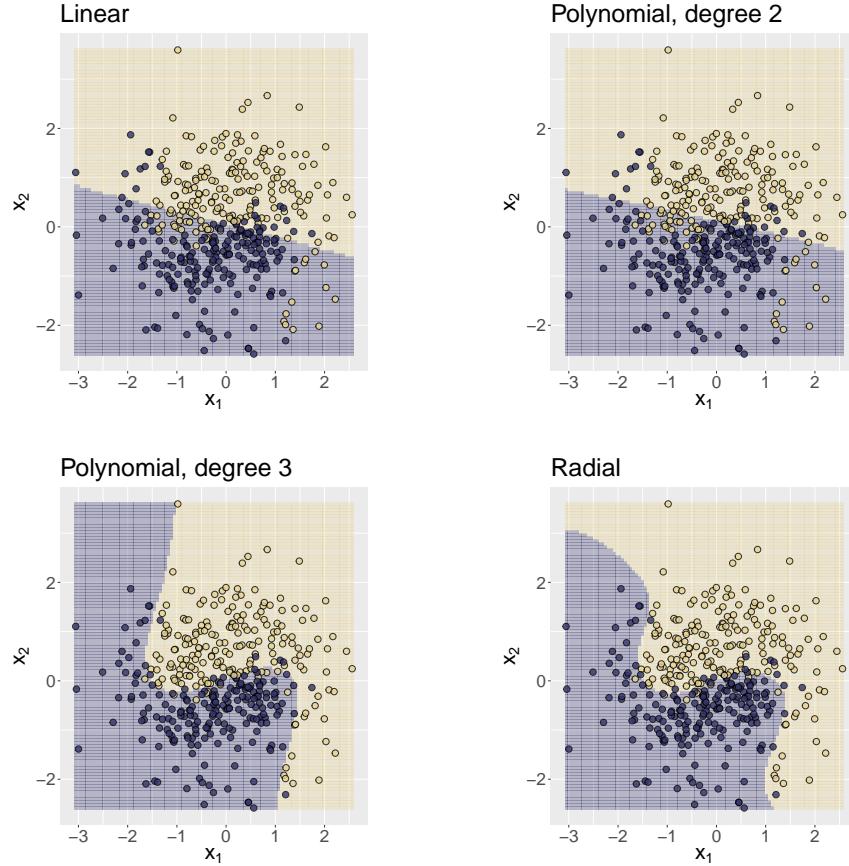


Figure 6.14: SVM decision boundary using different kernels. Linear kernel results in a picture that is very similar to logistic regression (see Figure 5.4). In a similar fashion, quadratic kernel (top right) is unable to replicate the wavy pattern of color dots. But both 3rd degree polynomial and radial kernels can capture the main aspects of the decision boundary.

6.6 Neural Networks

Neural networks are one of the fastest developing classes of machine learning models. These have achieved tremendous progress in a variety of fields, in particular image and natural language processing. Modern neural networks can recognize images, faces, fingerprints, can convert pictures to text and text to pictures, understand spoken language and answer question.

While they are definitely exciting methods, these are also some of the most demanding ones with the number of trainable parameters in millions or even billions.⁵ This makes training and using such models slow and demanding, not only do such models require a lot of computing resources, they are also complicated to train. One also needs a large amount of training data which may be challenging to obtain. Another major downside of neural networks is lack of interpretability. As ML models are increasingly employed as decision-making aides, there is an increasing interest for model transparency. Neural networks and other complex models are essentially black boxes even for data scientists who train them.

We start the discussion with feed-forward neural networks, and introduce convolutional networks, the ones used for image processing, thereafter.

6.6.1 Feed-Forward Networks

Biological Origins

Humans have been interested in how brain works since ancient times. The first step in solving the puzzle was understanding the basic working mechanisms of *neurons*, the brain cells (Figure 6.15). In a very simplistic view, each neuron receives signals from other neurons through its dendrites, and outputs a signal to the dendrites of other neurons through its axon. Normally the axon is quiet and does not send any signal, but if large enough number of the neuron's input dendrites become active, then the neuron "fires", i.e. sends a signal through its axon to another neuron.

Next, we model this simple neuron using mathematical tools and create an artificial neuron. These artificial neurons are the fundamental building blocks of all computational neural networks. The simplest networks, made of such artificial neurons, are called *perceptrons*. We start by introducing *and-perceptron* and *or-perceptron*, both are essentially made of a single neuron. Thereafter we move to more complex perceptrons with *hidden layers*.

Perceptron

Perceptron (feed-forward neural network) is a simple neural network that can be built using such artificial neurons. Perceptrons can perform various prediction tasks but cannot compete with more specialized networks, such as convolutional networks for image processing or LSTM blocks for natural language processing. However, the more specialized models almost always contain embedded perceptrons-feed-forward layers.

⁵The largest language model as of 2021, *Switch-C*, contains $1.57 \cdot 10^{12}$ parameters.

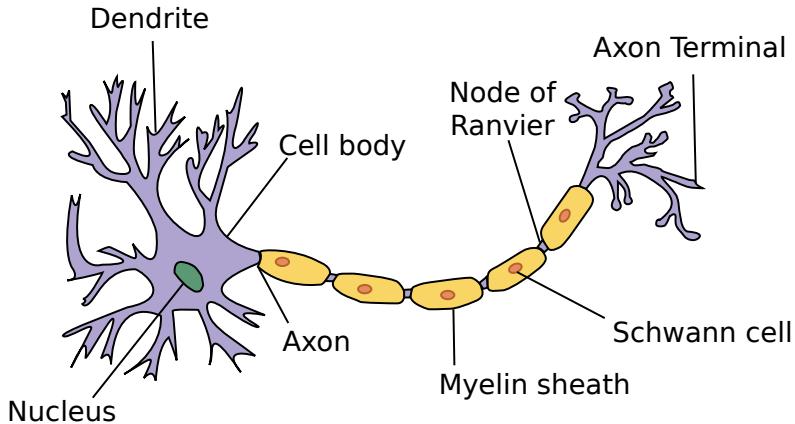


Figure 6.15: Schematic look of neuron. The cell has a long “tail”, axon, that is connected to the dendrites of other neurons. If a neuron receives enough electrical signals to its dendrites, it will “fire”, i.e. send a signal out of its axon.

By Dhp1080, CC BY-SA 3.0, from [Wikimedia Commons](#)

AND and OR Perceptron Imagine we have a neuron that gets two inputs (e.g. from other neurons) and outputs a signal only if both of these inputs are “high”. We can model the neuron activity by the following way:

1. Label the inputs x_1 and x_2 . These are the signals it receives through “dendrites”, and these can be either “low”–0, or “high”–1.
2. Let the neuron add both signals x_1 and x_2 . We are a little more general here and compute the weighted sum $z = w_1x_1 + w_2x_2$ where w_1 and w_2 are weights, so the neuron can assign different importance to different signals.
3. Let the neuron fire (output 1) if $z > \bar{z}$ where \bar{z} is a threshold, otherwise it will be quiet (output 0). So we can write output $y = \mathbb{1}(z > \bar{z})$.

Such a process activates the neuron if the input values are large enough (given w_1 and w_2 are positive), and it outputs 1 when active. We can write it formally as

$\mathbb{1}(z > \bar{z})$ is indicator function, see [Section 0.1 Functions](#), page viii

$$y = \mathbb{1}(w_1x_1 + w_2x_2 > \bar{z}). \quad (6.6.1)$$

Figure 6.16 depicts this perceptron as a simple neural network. The figure indicates some of the most important components of neural networks. The inputs are entering through *input layer*. The input nodes do not really do any operations, these are simply the feature vectors that are fed into the model. In this example we only have two inputs, but in complex networks, such as image processing, the input layer may contain millions of nodes corresponding to the input pixels.

In this example, input layer feeds its data directly to the output layer. Output layer contains a single node (neuron) that does two operations: first the linear transformation $z = w_1x_1 + w_2x_2$ and thereafter activation $y = \mathbb{1}(z > \bar{z})$. w_1 and w_2 are typically called *weights* and \bar{z} is called *bias*. Both of these operations are done almost

universally by all nodes (except input nodes) in all networks. However, in practice it is rare to use indicator function (step function) for activation. The most popular function in practice is ReLU, [see below](#).

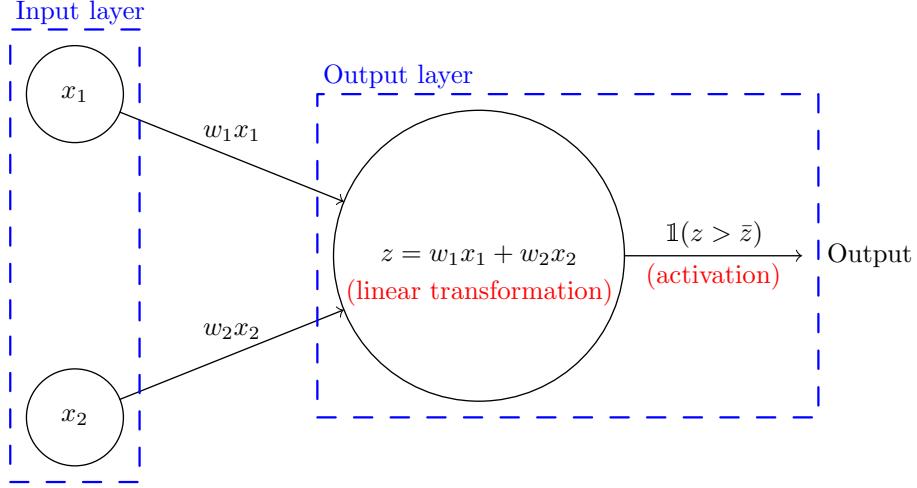


Figure 6.16: And Perceptron. The inputs x_1 and x_2 form the input layer, the single computing node forms the output layer. While the input layer only provides output to the node, the output node itself performs two operations: linear transformation $z = w_1x_1 + w_2x_2$, and activation, $y = \mathbb{1}(z > \bar{z})$.

This simple perceptron can perform logical *AND* and *OR* operations when choosing the weights and the bias accordingly. *AND* and *OR* operations are binary logical operations that take two inputs x_1 and x_2 , both of which may only have two values, 1 or 0, and always return either 1 or 0, depending on the inputs. See Table 6.7. Logical *AND* is 1 only if both inputs are 1, logical *OR* is 1 if any of the inputs is 1, and logical *XOR* (exclusive or) is 1 if exactly one of the inputs is one. This table can be understood as a dataset where we observe two features (inputs) x_1 and x_2 , and predict any of the outputs *AND*, *OR* or *XOR* (or maybe all three outputs at the same time). Unlike traditional datasets, this small table is comprehensive data, i.e. there are no more possible combinations of inputs, and the outputs are always exactly the same given inputs. No stochastic noise is possible here.

Table 6.7: AND, OR and XOR operations

inputs		outputs		
x_1	x_2	AND	OR	XOR
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

In order to perform the listed operations, we need to find suitable parameters, a triple of numbers (w_1, w_2, \bar{z}) . A possible solution for *AND*-perceptron is to choose $w_1 = w_2 = 1$ and $\bar{z} = 1.5$. For instance, if $x_1 = 1$ and $x_2 = 0$, then $z = 1 \cdot x_1 + 1 \cdot x_2 = 1 < \bar{z}$ and hence $y = 0$. However, if both $x_1 = x_2 = 1$, then $z = 2 > \bar{z}$ and so $y = 1$. Obviously, there are infinitely many possible solutions, for example, when keeping $w_1 = w_2 = 1$, every $\bar{z} \in (1, 2)$ will produce correct results.

Exercise 6.6: *OR*-perceptron

Use the same perceptron model as in Figure 6.16. Construct *OR*-perceptron: find a suitable set of (w_1, w_2, \bar{z}) that performs *OR*-operation.

[Solution](#) on page 431.

The neural networks and network operations are normally presented in vector form (and very often in matrix or even tensor form). In these examples, the input layer is feeding a vector $\mathbf{x} = (x_1, x_2)^\top$ to the output layer, and output layer is performing and operation

$$y = \mathbb{1}(\mathbf{x}^\top \cdot \mathbf{w} > \bar{z}) \quad (6.6.2)$$

where $\mathbf{w} = (w_1, w_2)^\top$ is the single neuron's vector of weights.

XOR-perceptron and hidden layers Can we use the same perceptron structure to perform *XOR*-operation? For *XOR*-perceptron, using the same model, we need values $\mathbf{w} = (w_1, w_2)$ and \bar{z} that represent the *XOR* column in Table 6.7. Using the vector notation (6.6.2), we can write four equations, one for each row of data in the table:

$$\begin{aligned} (0 \ 0) \cdot \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} &= 0 < \bar{z} & (1 \ 0) \cdot \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} &= w_1 > \bar{z} \\ (0 \ 1) \cdot \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} &= w_2 > \bar{z} & (1 \ 1) \cdot \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} &= w_1 + w_2 < \bar{z}. \end{aligned} \quad (6.6.3)$$

The first expression shows us that \bar{z} is positive; the second and third equation show that both w_1 and w_2 are larger than \bar{z} ; but the fourth equation, corresponding the $x_1 = 1$ and $x_2 = 1$ case contradicts the previous results by requiring that $w_1 + w_2 < \bar{z}$. Hence *XOR*-perceptron, using the model of Figure 6.16, is not possible.

A solution is to use a more complex network by introducing a *hidden layer* between the input and output layers (Figure 6.17). Now instead of connecting inputs \mathbf{x} to the output node y , we connect inputs to hidden layer nodes. In *XOR*-perceptron, the hidden layer contains two nodes, h_1 and h_2 . Both of these nodes work in a similar fashion as the output node in the *AND*-perceptron: they read inputs, perform a linear transformation in the form $\chi = \mathbf{x}^\top \mathbf{w}$, and activation in the form $h = \mathbb{1}(\chi > b)$. However, their outputs h are not the network final outputs, but are instead fed to the output layer as inputs. The output layer works exactly the same way as in case of *AND*-perceptrons above, just it takes its inputs not from the input layer but from the hidden layer.

Why is hidden layer called “hidden”? This is because we cannot observe these values in our data. Both inputs \mathbf{x} and outputs y are observable in labeled training

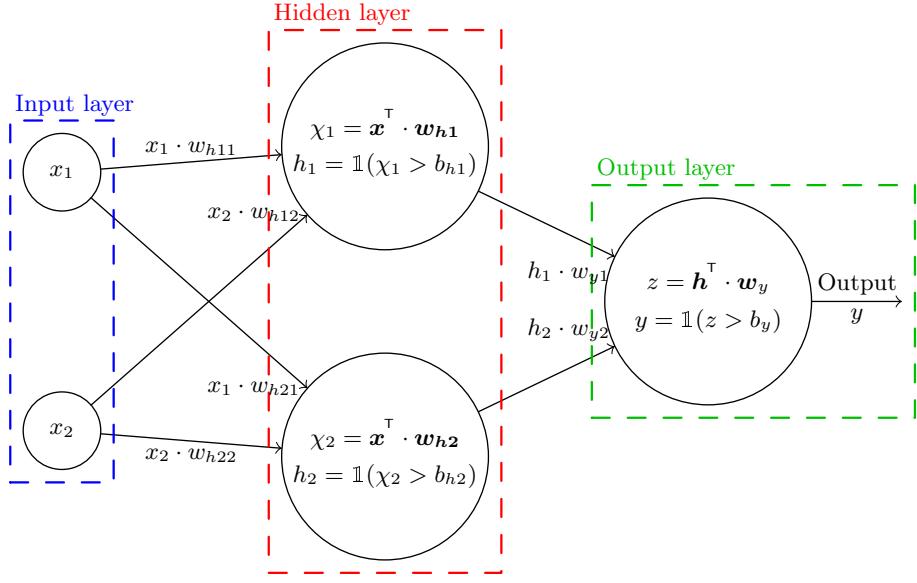


Figure 6.17: XOR Perceptron. The inputs x_1 and x_2 form the input layer, but now both input layer nodes are connected to both hidden layer nodes h_1 and h_2 , and not to the output layer. Both hidden layer nodes perform linear transformation and activation, using different weights \mathbf{w}_h and biases b_h . The single output layer node behaves exactly like in case of AND-perceptron, just it gets its inputs from the hidden layer, not from the input layer.

data. But we usually have no information about the “correct” values of the hidden layer values. Even more, there may be no such thing as hidden layers in the actual data generating process, our hidden layer nodes are just convenient mathematical tools that do not correspond to anything in reality. However, in certain cases hidden layers may represent concepts that are interpretable. For instance, one has found that in image processing tasks, some layers and nodes tend to recognize lines, arcs, bright regions on the image, and similar basic image features.

Let us now finish the *XOR* perceptron. There are many ways to set the parameters in a way that our network performs *XOR* operation. One particular way is to notice that $x_1 \text{ } XOR \text{ } x_2 = (x_1 \text{ } OR \text{ } x_2) - (x_1 \text{ } AND \text{ } x_2)$. Can we somehow, using the network in Figure 6.17, perform this subtraction? Yes we can. We can take the input nodes x_1 , x_2 and the hidden layer node h_1 and convert these into an *AND*-perceptron by setting $\mathbf{w}_{h1} = (1 \ 1)^\top$ and $b_{h1} = 1.5$. Thereafter we can take x_1 , x_2 , and the second hidden layer node h_2 and make an *OR*-perceptron by setting $\mathbf{w}_{h2} = (1 \ 1)^\top$ and $b_{h2} = 0.5$. And finally we can make the output layer node y to subtract *AND* from *OR* by setting $\mathbf{w}_y = (-1 \ 1)^\top$. The activation process must leave both values “1” and “0” unchanged, so we can pick $b_y = 0.5$.

Table 6.8 lists all the parameters for the perceptron in Figure 6.17. All in all we have 9 parameters. Obviously there is an infinite number of possibilities to choose the parameters, e.g. if we multiply all the weights and biases by a (positive) constant, the results are unaffected. We can also swap the role of h_1 and h_2 , and introduce

many other changes without affecting the predictions of our network. Although such flexibility may seem advantageous, it also suggests that neural networks are prone to overfitting and should normally be used with some form of regularization. It is also obvious that *XOR* binary logical operation does not have anything like “hidden layer”. There is no way to measure the “true values” of χ_1 and χ_2 , these values are just a trick to make the perceptron be able to perform *XOR*. So at least in this perceptron, the hidden values remains “hidden”, or perhaps even non-existent.

Table 6.8: XOR-perceptron parameters

node	weights	bias
h_1	1, 1	1.5
h_2	1, 1	0.5
y	-1, 1	0.5

Exercise 6.7: Use the perceptron for *XOR*

Show that the perceptron with parameters as given in Table 6.8 can perform *XOR*. In particular, show that $0 \text{ } XOR \text{ } 1 = 1$.

Hint: set $x_1 = 0$, $x_2 = 1$, and compute all the hidden values χ_1 , h_1 , χ_2 , h_2 and z . Do you get $y = 1$?

Computations the network nodes do are usually written (and coded) in matrix (or tensor) form. We can explain this by re-visiting the hidden layer of *XOR* perceptron example. The first hidden node χ_1 does the linear transform $\chi_1 = \mathbf{x}^\top \cdot \mathbf{w}_{h1}$ and the second hidden node $\chi_2 = \mathbf{x}^\top \cdot \mathbf{w}_{h2}$. We can combine these two transformations into

$$\begin{pmatrix} \chi_1 \\ \chi_2 \end{pmatrix} = \begin{pmatrix} \mathbf{w}_{h1}^\top \cdot \mathbf{x} + b_{h1} \\ \mathbf{w}_{h2}^\top \cdot \mathbf{x} + b_{h2} \end{pmatrix}. \quad (6.6.4)$$

And simplify this into a matrix form

$$\boldsymbol{\chi} = \mathbf{W}_h \cdot \mathbf{x} + \mathbf{b}_h. \quad (6.6.5)$$

Here $\boldsymbol{\chi}$ is the vector of linear terms inside of the hidden layer nodes, \mathbf{W}_h is matrix of hidden layer weights, where rows are the nodes and columns are the elements that correspond to the input vector. Finally, \mathbf{b}_h is a vector of hidden layer biases.

Multi-layer perceptrons

The *XOR*-perceptron above (Figure 6.17) is a small *multi-layer perceptron*. Multi-layer perceptrons are neural networks, made of the same building blocks as *XOR*-perceptron. Figure 6.18 show a larger similar perceptron:

- The network has an input layer with four nodes. This means we use four features for predicting the outcomes y_1 and y_2 .

- The middle of the network is made of two hidden layers, the first one with five and the second layer with four nodes.
- Finally, it has an output layer with two nodes. Two nodes is a common feature for binary outcomes, in those cases one node predicts the probability of the first outcome, and the other node the probability of the second outcome.

This network is densely connected: all nodes of a given layer get inputs from all nodes of the previous layer, and send their output to all nodes of the following layer.

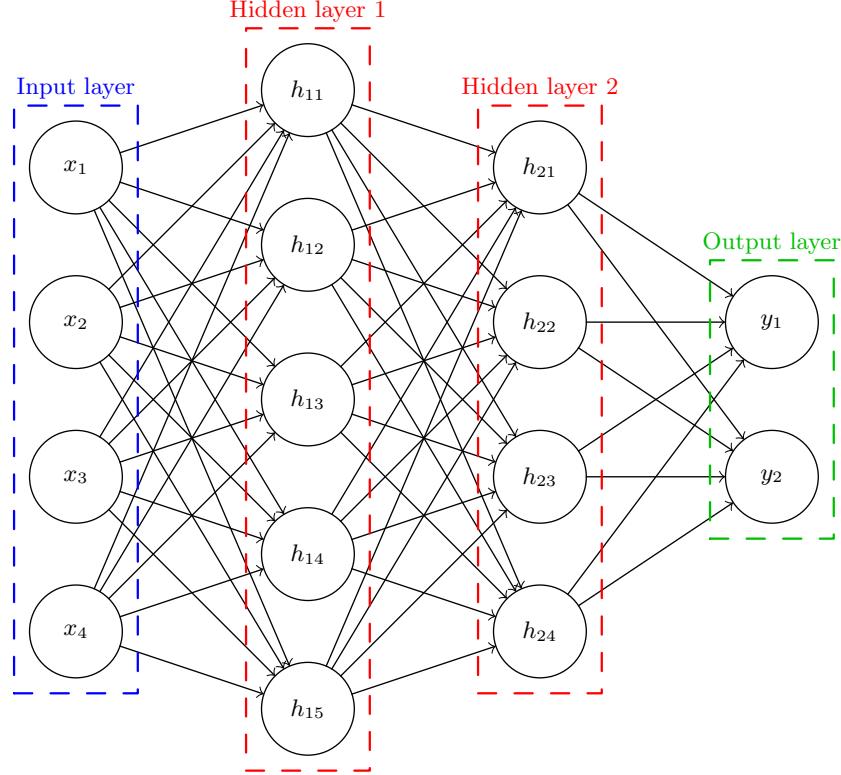


Figure 6.18: Multi-layer perceptron: this is a dense network with four inputs, two outputs, and with two hidden layers, the first one with five and the second one with four nodes. It is a dense network in a sense that all nodes in the previous layer are connected to all nodes in the following layer.

Each node in the network works in a similar way as in the XOR-perceptron. For instance, the node h_{12} computes the linear transform $\chi_{12} = \mathbf{w}_{h12}^\top \cdot \mathbf{x}$ and thereafter activation $\mathbb{1}(\chi_{12} > \bar{\chi}_{12})$. Here \mathbf{w}_{h12} is the weight vector for the node χ_{12} .

Activation

Above we explained that all nodes do both the linear operations (that can be expressed in matrix form), and activation. There are several popular activation functions, below we discuss a few of these.

ReLU (rectified linear unit) is perhaps the most popular activation function for neural networks (except in the output layer). It is essentially a line with a kink at 0, and defined as

$$\text{ReLU}(x) = \max(x, 0) = \begin{cases} 0 & x < 0 \\ x & x \geq 0 \end{cases} \quad (6.6.6)$$

It has the advantage of being simple and being piecewise linear, and hence easy to compute. Its derivative is either 0 or 1, although is not differentiable at 0. It seems to matter little in practice, although most theoretical optimization literature seem to focus only on differentiable (Lipschitz continuous) functions (see [Bottou et al., 2018](#)).

The downside of ReLU is that it is constant on the whole negative domain. Hence the gradient is just zero on a whole range of parameter values and training performance may suffer.

Leaky ReLU is a version of relu where the function value is not identically zero at the negative domain, just it has a small slope there (see Figure 6.19)

$$\text{ReLU}(x) = \max(x, 0) = \begin{cases} \alpha \cdot x & x < 0 \\ x & x \geq 0 \end{cases}. \quad (6.6.7)$$

$0 < \alpha < 1$ is a hyperparameter that must be specified, not trained.

Leaky ReLU is used in contexts where one needs non-zero gradient in the negative domain.

Softmax or Logistic (multinomial logit) is defined as logistic transformation of input vector \mathbf{x} . It's i -th component is

$$\Lambda(\mathbf{x})_i = \frac{e^{x_i}}{\sum_j e^{x_j}} \quad (6.6.8)$$

Its one-dimensional version is the same as logistic function , and is often called *sigmoid* function in ML literature. It is often denoted by $\sigma(x)$. See Figure 6.19

Softmax has a very useful property: namely, whatever the value of x_i , e^{x_i} is always positive. And hence $\sum_j e^{x_j}$ is always positive, and hence the components of $\Lambda(\mathbf{x})$ sum to unity. So softmax is a transformation that converts all kind of inputs into valid probabilities, this is why it is the most popular output layer activation function for categorization problems. Whatever comes to the output layer, it always outputs a valid probability vector.

Example 6.9: Softmax outputs

Below is a small table of sample inputs, exponents, and softmax outputs for a three-valued softmax output layer.

Inputs x_i			Exponents e^{x_i}			$\sum_i e^{x_i}$	Probabilities $\frac{e^{x_i}}{\sum_i e^{x_i}}$		
0.10	0.20	0.30	1.11	1.22	1.35	3.68	0.30	0.33	0.37
1.00	2.00	3.00	2.72	7.39	20.09	30.19	0.09	0.24	0.67
0.10	-0.10	3.00	1.11	0.90	20.09	22.10	0.05	0.04	0.91

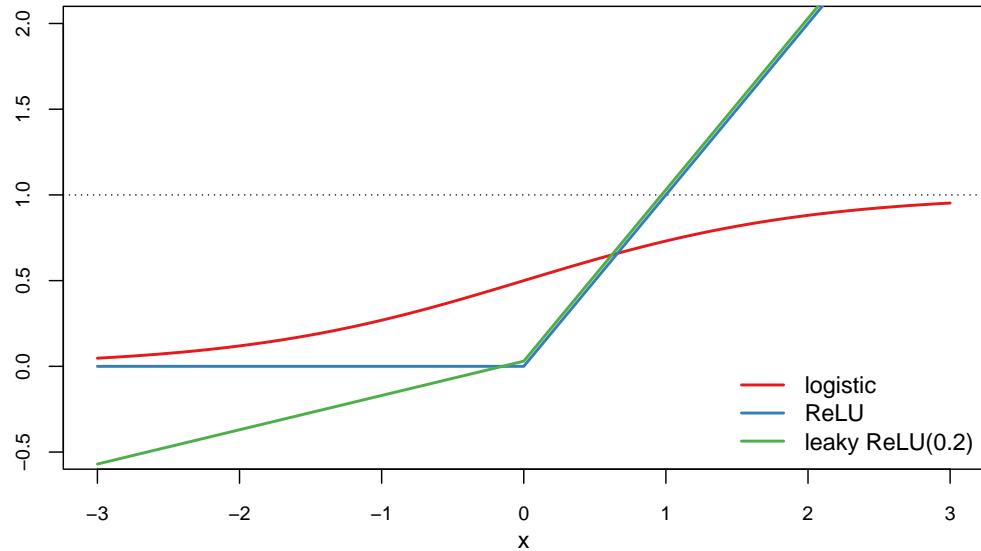


Figure 6.19: A few popular activation functions for neural networks. Leaky ReLU with $\alpha = 0.2$ is shifted slightly up for clarity.

The table shows three sets of inputs: $(0.1, 0.2, 0.3)$, $(-1, 0, 5)$ and $(0.1, -0.1, 2)$. The first of these consists of rather similar probabilities, $(0.30, 0.33, 0.37)$. In the second case, the third probability is noticeably larger, while the first and second are small but positive. In the third case, the first two probabilities are rather similar, but the third one is much larger again.

Exercise 6.8: Softmax property

a) Compute $\text{softmax}(1,2,3)$ and $\text{softmax}(4,5,6)$.

b) Prove that

$$\text{softmax}(\mathbf{x}) = \text{softmax}(\lambda + \mathbf{x}) \quad (6.6.9)$$

where \mathbf{x} is the vector of inputs, and $\lambda \in \mathbb{R}$.

Solution on page [432](#)

No activation Finally, one can also leave activation out (use identity function as the activation function). This is useful for regression models. In fact, neural network with no hidden layers and identity activation is equivalent to linear regression.

TBD: multiple outputs

TBD: outputs: prediction vs regression

6.6.2 Convolutional Neural Networks

Convolutional neural networks are networks that incorporate *convolutions*, certain kind of weighted sums over spatially arranged data. Convolutional filters are well-known methods to manipulate and enhance images or other signals, and have been widely used long before the neural networks became popular.

In the following sections, we explain what are convolutions, how they can be used to detect image elements, and how they can be incorporated into neural networks.

Convolutions and convolutional filters

Convolution is essentially a weighted sum or average over certain spatial region of the data. It is a function of two sources of information—data and weights. *Spatial* in this context means not necessarily space, but all other kind of arrangements where it makes sense to talk about distance between data points. This includes images where we can talk about neighboring pixels, or pixels that are farther apart, or time series, where we can talk about observations that are taken in next day, versus two days ago.

For instance, a convolution of series of daily temperatures T_t may involve sum of temperature T of the given day t , plus two days before and after that day. The weights may be larger for the given day and smaller for the other days, for instance $w_{-2} = w_2 = 0.25$, $w_{-1} = w_1 = 0.5$ and $w_0 = 1$. Here the index for weights denotes how many days off we are from the central day of interest. Convolution is often denoted by $*$, and for the daily temperatures we may write the convolution as $T * w$. As we can choose an arbitrary day t as the day of interest, the result depends on t and we write $(T * w)(t)$:

$$(T * w)(t) = w_{-1}T_{t-2} + w_{-1}T_{t-1} + w_0T_t + w_1T_{t+1} + w_2T_{t+2}. \quad (6.6.10)$$

The vector of weights $w = (w_{-2}, w_{-1}, w_0, w_1, w_2)$ is called *kernel* or *filter*. Note that as we approach the “edge” of our data, we cannot compute the convolution any more: for instance, for the last data point we do not know the values of T_{t+1} and T_{t+2} . So convolution either “loses” some data at the edge, or alternatively, the “over-the-edge” data must be filled in (*padded*) somehow (see [Section 6.6.2 Padding](#), page 313 below).

Example 6.10: 1-D convolution

Imagine we are measuring temperature on Pluto. But it is far away and we only get time at the expensive telescope once a month to do the measurements. We measure the temperature in five consecutive months and get $T_1 = 40K$ (-233C), $T_2 = 44K$, $T_3 = 44K$, $T_4 = 52K$, and $T_5 = 44K$. As our measurements are imprecise, we may want to smooth the individual observations over time (compute moving average). But we also want to put more weight on the current month’s measurement and less weight on the neighboring months, as the measured differences may reflect true processes on Pluto. So we choose weights $w_{-1} = 0.25$, $w_0 = 0.5$, and $w_1 = 0.25$. The convoluted (averaged) temperatures are:

$$\begin{aligned}(T * w)(2) &= w_{-1}T_1 + w_0T_2 + w_1T_3 = 43K \\ (T * w)(3) &= w_{-1}T_2 + w_0T_3 + w_1T_4 = 45K \\ (T * w)(4) &= w_{-1}T_3 + w_0T_4 + w_1T_5 = 48K.\end{aligned}$$

In this way we can use convolutions for smoothing noisy observations. This is also why we want the weight to sum to unity in this case—we do not want our smoothed temperature values to be systematically biased.

For 2-D case, convolutions are defined in a similar fashion. For instance, we may convolve surface temperature over certain geographic area by averaging measurements in this area while giving more distant measurements lower weight. The main difference between 1-D and 2-D case is that now we need a 2-D weight structure. So instead of a simple sum, now we need a double sum

$$(T * W)(k,l) = \sum_i \sum_j w_{ij} T_{k+i,l+j}, \quad (6.6.11)$$

where the weight matrix $W = \{w_{ij}\}$. For instance, when doing a similar temperature measurements, but now over a rectangular grid, we may set the weight matrix to

$$W = \begin{pmatrix} \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \\ \frac{2}{16} & \frac{4}{16} & \frac{2}{16} \\ \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \end{pmatrix}. \quad (6.6.12)$$

As in case of time series smoothing, we chose the weights to sum to unity in order to avoid systematic bias in temperature. But in other applications, the weights do not have to sum to one. We also do not have to choose a 3×3 kernel, it may be of any dimension, including even numbers like 2×2 or 4×4 , or we can use non-square kernels, such as 2×3 .

In neural networks, 2-D convolutions are one of the main workhorses for image processing. Let us demonstrate 2-D convolution by constructing a filter that detects vertical edges in images. We pick a very simple 3×4 image, that include a 2×2 black box in the top-right corner (see Figure 6.20 left). For simplicity, the image only has two possible pixel values, 0 (white) and 1 (black). We choose a 2×2 kernel with weights as

$$W = \begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix}. \quad (6.6.13)$$

One can easily imagine convolutions as moving this kernel window over the image, pixel-by-pixel. At each location one has to multiply the pixel values with the corresponding kernel weights and add the results.

The image has one vertical edge in the upper-middle of the image, the one that separates the black and white area. We start moving the filter across the image from the top-left corner (green frame). This results in zero value, as all the pixel values are 0, and multiplying those with the corresponding filter values does not change the matters. Next, we move the kernel window right by one pixel (red frame). In that position the negative filter weights overlap with 0-values pixels while positive weights overlap with 1-value pixels. As a result, the filter returns 2. Finally, in the rightmost position (blue frame), both positive and negative weights overlap with equal pixel values (1) and hence the output is again 0. The filter only “fires” if there is a vertical edge on the image. The figure only depicts convolution along the upper row of the

image. If we lower the filter down by one pixel, then the all-zero lowermost row does not contribute to the output, and we get values 0, 1, and 0. The filter still “fires” for the vertical edge in the middle, but now it fires only “partially” because the vertical edge is only partially in the filter’s window.

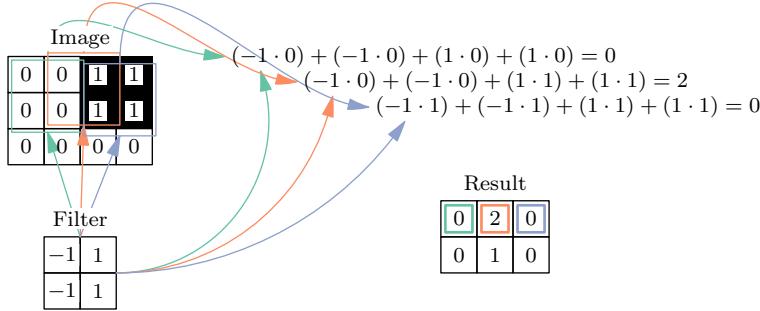


Figure 6.20: Vertical edge detection using 2D convolution filter. Moving the convolution filter across the image, the top-left corner (green frame) results in zero, as all the pixel values are 0. In the next position (red frame) the negative filter weights overlap with 0-values pixels while positive weights overlap with 1-value pixels. As a result, the filter returns 2. Finally, in the rightmost position (blue frame), both positive and negative weights overlap with equal pixel values (1) and hence the output is again 0. The filter only “fires” if there is a vertical edge on the image.

The result of moving over the image (ther “Result” box in the figure) is a new image where brightness corresponds tof “vertical edgeness” of the original image. In this example, the largest edgeness value is in the middle-upper position, the position where the window exactly overlaps the vertical edge. Values 0 correspond to the case where the image does not contain any vertical edges, and value 1 corresponds to the case where the filter window only partially contains a vertical edge. Now if we want to know if any particular location contains a vertical edge, we have to see what are the values of the result layer in that location. Large value represent to a vertical edge.⁶

⁶Large *positive* values represent an edge between white pixels at left and black pixels at right. Large *negative* values correspond to the opposite case.

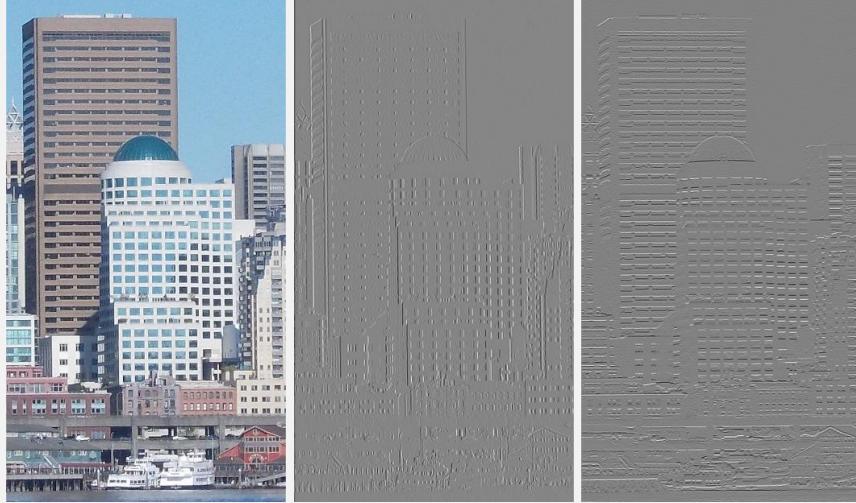
Example 6.11: Edges on image

Figure 6.21: Edge detection: original image (left), vertical edges (center), and horizontal edges (right). Unlike Figure 6.20, this is a color image, and the edge detection filters contains three similar layers, one for each color channel. The filters used are $\begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix}$ for the vertical edges, and $\begin{pmatrix} -1 & -1 \\ 1 & 1 \end{pmatrix}$ for the horizontal edges.

Exercise 6.9: Corner detection with convolutions

Consider image $M = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$ and filter $F = \begin{pmatrix} -1 & -1 \\ -1 & 3 \end{pmatrix}$. What is the convolved image $M * F$? Explain where/which kind of corners does the filter detect.

Solution on page [432](#).

From convolutional layers to convolutional networks

In the previous example, we just moved a single convolutional filter around over the image, and got another image where high pixel values denote “edginess” on the original image.

In case of convolutional networks, this process is embedded in a neural network, and we typically use more than just a single filter. Figure 6.22 shows what happens when using four filters (kernels) on a color image. At left, we see the three color channels (red, green, blue) of the image. The filter is currently located at the lower-right corner of the image. But now it is not a 2×2 filter but $2 \times 2 \times 3$ filter instead— 2×2

is its spatial dimension, and the last dimension encompasses all image channels. Note that in the spatial sense it is still a 2×2 filter. This is because the pixels in each channel are aligned—they have well-defined position with respect to each other. But the color channels are not aligned in this way, there is no inherent reason to say that the green channel is located underneath the red one, or the way around. Colors are unordered, non-spatial features. Hence we usually just talk about 2×2 filters, even if they encompass all the channels. But be aware that in reality they are of dimension $2 \times 2 \times 3$ and hence contain 12 weights, not 4 weights. Weights for the filter's 3 layers will differ, in general.

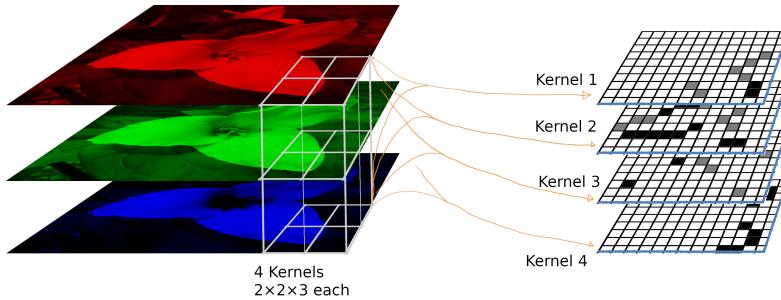


Figure 6.22: Color image and multiple filters (kernels). The input image contains three channels: red, green and blue (left). All kernels work on all layers, hence in some sense they are not 2×2 but $2 \times 2 \times 3$ kernels with 12 weights each. Each kernel, when moved across image, creates a new convoluted channel (right). In this example, we have four different kernels, so together they produce a 4-channel convolution image. The new channels do not represent colors but certain other properties of the image, hence here they are represented as black-and-white only. We can take these four channels as inputs for a second layer of convolutions.

Each filter, when moved across the image, produces a new channel, the image convoluted with this particular kernel. These are shown at right on the figure. If the original image was of size 100×100 pixels, the convoluted image has 99×99 pixels, because we lose one pixel at the edge. But instead of a single convolutional filter, we use four filters in this example, labeled as “kernel 1”, “kernel 2” and so on. Accordingly, the convolutional layer contains $4 \times 12 = 48$ weights, 12 weights for each kernel. Each of these kernels creates a convoluted image. These are the four black-and-white images at right on the figure. So the size of the final output of this convolutional layer is $99 \times 99 \times 4$. These are depicted as black-and-white, because they are not interpretable as color channels. They may capture edges, or bright spots, or corners, or other details instead, depending on what exactly the kernels are.

Also, while the dots on the resulting channels (black-and-white layers at right on the Figure) have clear spatial position, this is not the case for ordering of the channels. There is no reason to think that, for instance, corners should be underneath edges or the way around. Hence one set of convolutional filters gave us a multi-channel convoluted image that is in some ways similar to the original image—its width and height have clear spatial meaning, but the channels are unordered. Hence we can add another set of convolutional filters that take the first convoluted image as input,

and perform another set on convolutions on that image. The input image now has as many channels as how many filters we had in the first layer of convolutions, four in this example, and hence the second layer convolutional filters would be of size $2 \times 2 \times 4$ and contain 16 weight each.

It is possible to hand-craft all these filters. For instance, we can add a filter for horizontal edges, a filter for corners, a filter for diagonal lines and so on. However, in case of neural networks, we normally do not hand-craft the filters but let the networks learn what is a good combination of kernels. For example, the network may learn that many vertical edges are associated with buildings, while human face may be better visible when using curved lines. This is partly because hand-crafting filters is a rather laborious task, but more importantly—in typical image processing tasks we do not know what a good set of filters might be. It is not hard to manually design kernels that distinguish between lines and curves, but what kind of kernels can distinguish between cats and dogs? Instead, in typical image processing tasks we allow the network to learn a large number (e.g. 64) filters in multiple layers, so the network does not have to rely on just vertical edges, but can find various details that may allow to distinguish complex images.

Note that filters are not limited to 2×2 size, they may be a lot larger, and they do not have to be of square shape.

Above we discussed just convolutions—the linear part of the convolutional layers. The actual layers in neural networks also include activation. So a single convolutional filter in a convolutional layer outputs

$$f(b + (\mathbf{T} * \mathbf{w})(x, y)) \quad (6.6.14)$$

where \mathbf{T} is the input image tensor ($100 \times 100 \times 3$ in the example above), \mathbf{w} is the tensor of weights ($2 \times 2 \times 3$ in this example), x and y are pixel coordinates, b is bias, and $f(\cdot)$ is the activation function. Hence a single convolutional filter in this example contains $w \times h \times l + 1$ parameters, where w is the kernel width, h is kernel height and l is the number of layers. “+1” is because of the bias in the activation function. In the example above it is $2 \times 2 \times 3 + 1 = 13$ parameters for each filter, and $4 \times 13 = 52$ parameters for a convolutional layer.

Padding, Pooling, and Strides

Pooling After running the data through a convolutional filter, we have another matrix that tells how well did each place in the image correspond to what the filter captures. In the example in Figure 6.20 we can see that the edges of the image do not correspond to vertical edges, the lower-middle of the image corresponds somewhat, and top-middle corresponds the best. Often we are not interested in such a detailed knowledge. For instance, we may be interested in the location of the cleanest vertical edge while willing to ignore the other less-clean representation nearby.

In this case we may run the resulting data through a *pooling layer*. Popular max pooling finds the maximum value in a small area, e.g. in a 2×2 square on the layer. In this example, max pooling will result in value 2, indicating that there is a clear vertical edge somewhere in that region. It will however ignore 0-s and 1-s, so we do not learn that there is also places that do not represent vertical edges. If we are

interested in the latter, we may choose average pooling instead. This will correspond to average “edgeness” of that region on the image.

Padding It is obvious from Figure 6.20 that the resulting layer is smaller than the original image layer. We have to fit the whole filter onto the image, and as soon as its dimension is more than 1, we have fewer points in the output than in the input. We can proceed in different ways:

- We can just accept that this is the case, and that the layers get smaller and smaller as the data proceeds through successive convolutional layers.
- Alternatively we can “pad” the layers with certain values, e.g. some pre-determined values, or values from nearby pixels. This is called *padding*.

Strides Finally, we do not have to move the convolution window by a single pixel each time. We may choose another step size, called *stride*. A large stride may be useful if the image is fuzzy, or if the features do not change rapidly from pixel to pixel. Large strides are a way to lower the resolution in the middle of the network. One may also choose different strides for horizontal/vertical movement.

6.7 Comparison and Review

ML methods are powerful tools, but as with other tools, none of them is a universal jack-of-all trades. There are many considerations when picking a suitable models. Below we discuss a number of example cases.

Interpretability If interpretability—understanding what do the results mean—is a major goal, then linear or logistic will be the first choice. No other method can be understood in such a clean fashion.

Explainability In an analogous, if explainability—being able to explain someone why such decisions were made—is desired, one should start with decision trees. Decision trees can easily be explained to people with limited statistical literacy.

Predictive performance Typically, the models that offer the best predictive performance are k -NN, random forests and other ensemble methods, SVM-s, and neural networks. All of these have their strong and weak sides.

Neural networks are unmatched in their performance to identify complex patterns. They beat all other methods in image or speech recognition and text processing. However, that does not mean that neural networks are always the way to go. First, they only help in cases where there actually is a complex pattern in data. Figure 6.23 shows an example with a complex pattern (left), where one might benefit from powerful and flexible models, such as random forests or neural networks. The RHS figure, however, shows a simple gradient from bottom left to top right. Here just a plain

logistic regression performs adequately, and what is more, no more advanced model can perform any better. There is just no information in data that logistic regression cannot use. Advanced models will perform equally well at best, and will overfit in the worst case.

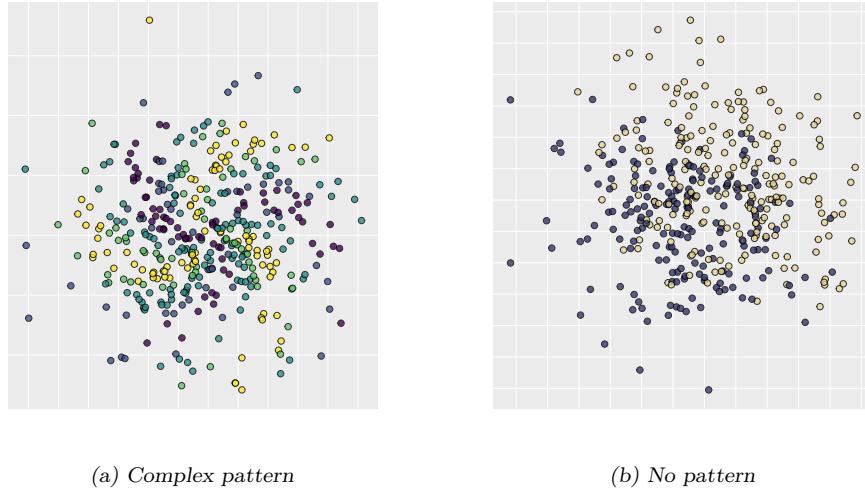


Figure 6.23: 2-D example of complex pattern (left) and a simple pattern (right). While advanced models, such as neural networks, can pick up the spiral pattern at left, even a simple logistic regression is capable of identifying the left-right gradient at the right. No more advanced model can beat it here as data just do not contain any complex patterns.

Unfortunately, it may not be obvious at all if such patterns exist in data. In certain cases, e.g. in case of images, our brain can evaluate this very well. But in other kind of data it is almost impossible to know. Experience is your best friend here.

Computational and data considerations While simple models on small datasets are computed almost instantaneously, training complex models on large dataset can easily take days. Even if that is desirable from performance perspective, the associated cost may render such models infeasible.

In a similar fashion, more flexible models typically require much more labeled training data to be able to learn to generalize correctly. Again, this it may not be feasible to acquire enough labeled data of suitable quality. This is one of the reasons AI applications sometimes fail unexpectedly, when confronted with a dark-skinned face or female voice. The developers were just using training data that they found easiest to get, and that happened to be about white males.

Chapter 7

Different Types of Data

Introductory machine learning problems are often presented using well-behaved numeric-only datasets. Here we look at some of the different data types and explain how to use these for ML models.

Contents

7.1	Numeric Data	315
7.2	Images	316
7.2.1	Black-and-white images	316
7.2.2	Color images	317
7.2.3	Image transformations	317
7.3	Text as Data	323
7.3.1	Preprocessing: Tokenization, Stemming, and Lemmatization	323
7.3.2	n -grams	324
7.3.3	Bag of Words and Document-Term-Matrix	325
7.3.4	TF-IDF	327
7.3.5	Word embeddings	329

7.1 Numeric Data

This is one of the most common forms of data, and in a way the easiest one to work with. Most machine learning and other analytical methods are designed for numerical data, even more, typical mathematical operations we want to do, such as multiplication and addition, can only be done using numeric data. However, numeric data is not just numbers. It can come in various forms, and not all forms of numeric data works with all methods.

7.2 Images

One of the distinct and valuable data source is images. Images are relatively straightforward to process and store as these are normally represented as pixel arrays where each array element represents one pixel on the image. In this section we only discuss bitmap images, wireframe images were discussed above in [Section 2.4 Application: wireframe images](#), page 117. We also do not discuss the compressed image formats, such as *jpeg* or *png* that allow to compress and store such bitmaps in a more efficient way.

We start with black-and-white images, as these are stored in somewhat easier way, and talk about color images thereafter.

7.2.1 Black-and-white images

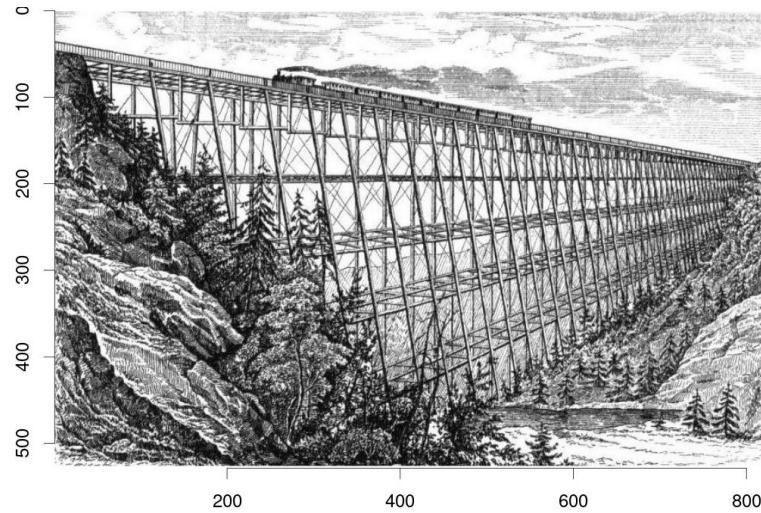


Figure 7.1: Lyman Trestle in Connecticut, around 1876 (from [Wikimedia Commons](#)). It is stored in memory as 820×526 array of gray values. The axes depict the corresponding pixel coordinates.

Black-and-white images are normally stored as matrices of gray values. Figure 7.1 depicts one such grayscale image and Figure 7.2 shows a closer view on a 10×10 pixel detail, centered at the locomotive's smokestack. The image is made of *pixels*, small squares of different shade of gray, these are made clearly visible in Figure 7.2. The numerical values of the shades of gray for each pixel is what is stored in the image matrix. The 10×10 matrix itself, it's size corresponding to the image size, is shown in the lower panel. Here the uppermost line on the detail, where the shade transfers from light of the sky to dark gray of the smoke, corresponds to the first (uppermost) row of the matrix. The gray values range from 0.98 (light sky) to 0.43 (dark smoke in the top-right pixel). One can also see that the darkest areas of the smokestack are of value close to zero (the few lowermost lines) while the lightest points are of value

1.00 (the perfect white).

In practice, it is important to keep in mind that matrices are typically stored as *rows-by-columns*, while images (and plotting coordinates) are typically presented as *width-by-height*. Also, high values may correspond to either dark or low pixel intensity. The gray values are sometimes coded as real numbers in [0,1], and sometimes as integers from 1 to 255. All this is obviously software-specific, but causes quite a bit of confusion when working with images for the first time.

7.2.2 Color images

Color images are constructed in broadly similar way as black-and-white images, just these contain three separate layers for different colors, normally red, green and blue.

Figure 7.3 shows such a color image. The top panel is the original high-res image (at left) and a [low resolution](#) (5×8 -pixels) version of it to facilitate the display of data matrices. The lower panels depict the three color channels, R, G, and B. Small numbers close to “0” indicate little intensity (black), and high values close to “1” indicate high intensity (either red, green or blue, depending on the channel). For instance, the columns 4 and 5 in the first row of the R channel have values 0.00 indicating that these two pixels contain no red. The same pixels have value 0.37 in G and 0.72 in B channel, indicating that the flag’s blue contains about 1/3 green and the 2/3 blue, “pure blue” that is produced by the computer screen. The middle pixels (row 3 and column 5) however are of value 1.00 in all channels. This means that pixel is “pure white”, displaying the maximum color intensity in all channels.

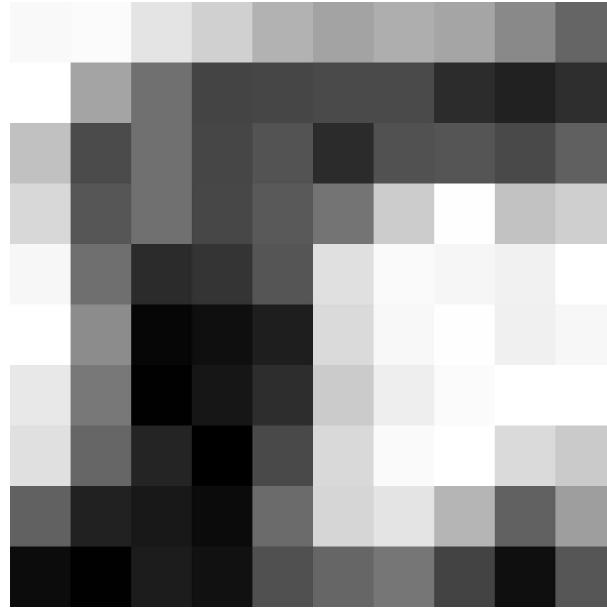
Such 3-channel layout of images is very common for color images, for instance all jpeg images are made of three channels. When working with image data in memory, then all these layers are put “on top of each other”. So the Scottish flag may be stored as a $5 \times 8 \times 3$ or a $8 \times 5 \times 3$ array, a tensor.

Other images, such as some png-s also contain a fourth layer, representing transparency. In fact, the [original png image](#) contains such a transparency layer. But as the image is completely oblique, the fourth layer has all pixels marked as “1.0”. There may be even more layers, e.g. one that indicates the pixel’s distance (depth), but that is not common.

7.2.3 Image transformations

Transforming Bitmap Images into Coordinate Matrix Form

Wireframe image data we discussed above contain vertex coordinates, while color, a vertex attribute, is a secondary consideration. In contrast, bitmaps images only contain the color values in a regular grid but do not explicitly contain the pixel coordinates. In this sense wireframe images are similar to sparse matrices and bitmaps to dense matrices. So in order to rotate a bitmap as matrix, we first have to create its coordinate matrix. This can be done by creating an (x, y) coordinate pair for each pixel so that the pixels can be described by a triple $(x, y, value)$. As the pixels are arranged in the matrix in a regular matrix, x and y correspond to either matrix columns and rows, or the way around, depending on the software. Are vertex coordinates and colors will be treated differently, we put the image information into two



(a) Detail (locomotive's smokestack) from Figure 7.1.

	1	2	3	4	5	6	7	8	9	10
1	0.98	0.98	0.90	0.82	0.71	0.66	0.70	0.67	0.56	0.43
2	1.00	0.66	0.47	0.31	0.31	0.33	0.33	0.22	0.17	0.22
3	0.77	0.33	0.47	0.31	0.36	0.21	0.35	0.37	0.32	0.41
4	0.85	0.37	0.47	0.31	0.38	0.48	0.81	1.00	0.77	0.82
5	0.97	0.46	0.21	0.24	0.37	0.89	0.98	0.96	0.95	1.00
6	1.00	0.57	0.07	0.11	0.16	0.86	0.97	1.00	0.95	0.97
7	0.91	0.50	0.05	0.13	0.22	0.81	0.94	0.98	1.00	1.00
8	0.89	0.43	0.18	0.05	0.32	0.86	0.98	1.00	0.86	0.80
9	0.41	0.17	0.14	0.09	0.45	0.85	0.90	0.73	0.41	0.64
10	0.09	0.05	0.16	0.11	0.35	0.43	0.49	0.30	0.11	0.37

(b) The gray level values corresponding to the detail in the upper panel. The high values (near 1.0) correspond to white and low values (near 0.0) correspond to black. One can see that the darkest details of the smokestack are of value 0.05 and the lighest sky is of value 1.00.

Figure 7.2: Detail from image 7.1, and the corresponding gray values.

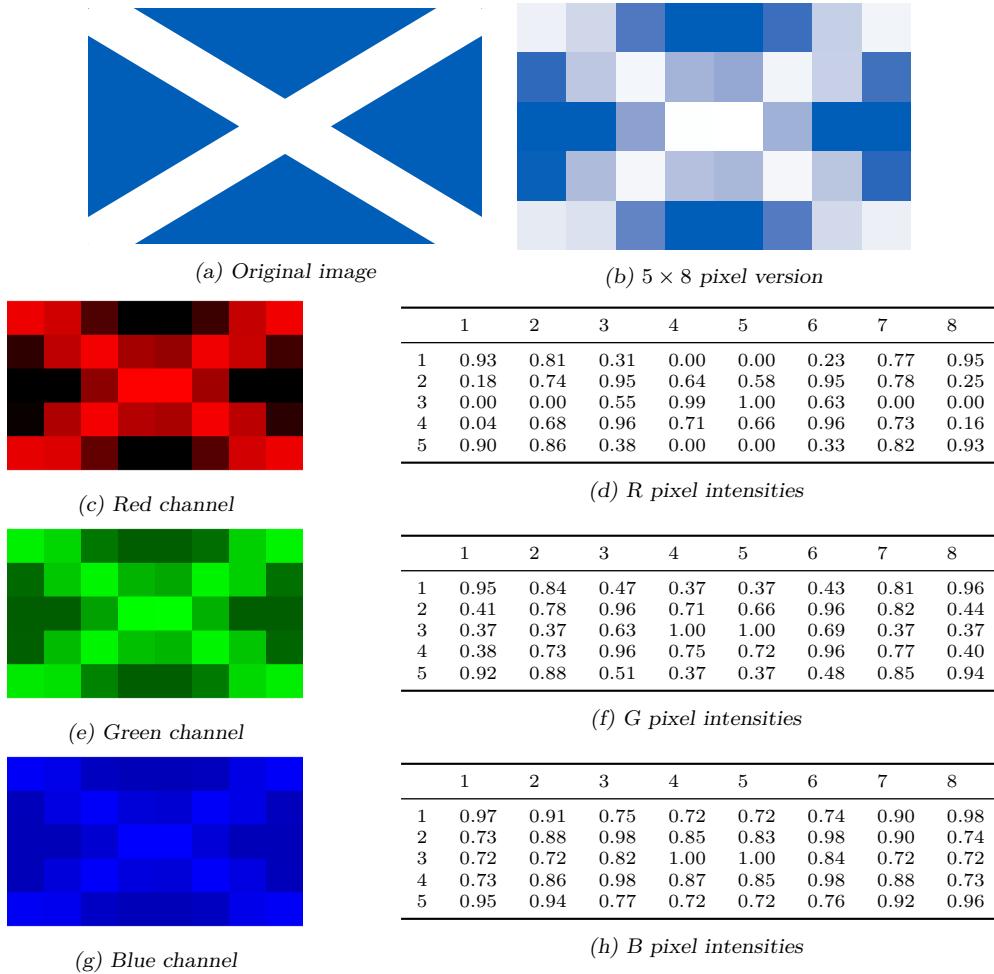


Figure 7.3: Flag of Scotland. The upper left picture shows the original flag, the upper right picture shows the same image in low-resolution for better display of data. The lower panels depict the corresponding color channels, R, G and B; images at left and the pixel intensities at right. The pixel intensities are close to “1” in the white cross as the white color is made of all three color channels at full intensity. However, the blue areas have little red (values close to 0) color, some green (values around 0.37), while blue channel remains at high intensity (values 0.72). So in blue, the flag has low contrast.

matrices: a $N \times 2$ coordinate matrix \mathbf{X} , and a $N \times 1$ pixel gray value matrix \mathbf{G} . Note that N is not the height but $height \times width$ of the image because each row in these matrices correspond to a single pixel, not to a single row. For instance, the image detail from figure 7.2 will be stored in two matrices,

$$\mathbf{X} = \begin{pmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ \vdots & \\ 1 & 2 \\ 2 & 2 \\ 3 & 2 \\ \vdots & \end{pmatrix} \quad \text{and} \quad \mathbf{G} = \begin{pmatrix} 0.98 \\ 0.98 \\ 0.90 \\ \vdots \\ 1.00 \\ 0.66 \\ 0.47 \\ \vdots \end{pmatrix}. \quad (7.2.1)$$

The first column of \mathbf{X} denotes the horizontal position, the column of the original image matrix, and it runs from 1 to the image width for each row. The second column is the vertical position, the image row, it is 1 for each pixel in the first row, 2 for each pixel in the second row and so forth. \mathbf{G} contains the same gray values as the data matrix in Figure 7.2. The pixel coordinates \mathbf{X} are conceptually the same as vertex coordinates for wireframe images, just we are not connecting vertices by lines but instead we use the gray value as the vertex color. The gray values will not change with rotation, just the pixels must be plotted in a different place as the image rotates.

Accordingly the image rotation will consist of two steps:

1. rotate (or otherwise transform) the coordinates \mathbf{X} into \mathbf{X}' , and
2. paint a dot at coordinates (x'_{i1}, x'_{i2}) with the gray value G_i .

Figure 7.4 illustrates this approach with the original image rotated 10 degrees counterclockwise:

Projecting bitmap images

As we can rotate bitmap images, we can also project these on 1-D line. Figure 7.5 depicts an image of a page of text that is rotated by 24 degrees. Suppose we want to detect its degree of rotation for further processing. One approach is to rotate the image and project the result onto the vertical axis. If the angle is correct, we should see a clear pattern of dark (text lines) and white (interline gaps). If the angle is wrong, the gaps will be unclear.

We can proceed in a similar fashion as above. First we translate the image into the regular grid coordinate matrix \mathbf{X} and the gray intensity levels \mathbf{G} , and thereafter we project \mathbf{X} onto the vertical line (by discarding the first coordinate component). Thereafter we can either plot the density on the margin, or better, compute the sum of gray levels in narrow intervals.

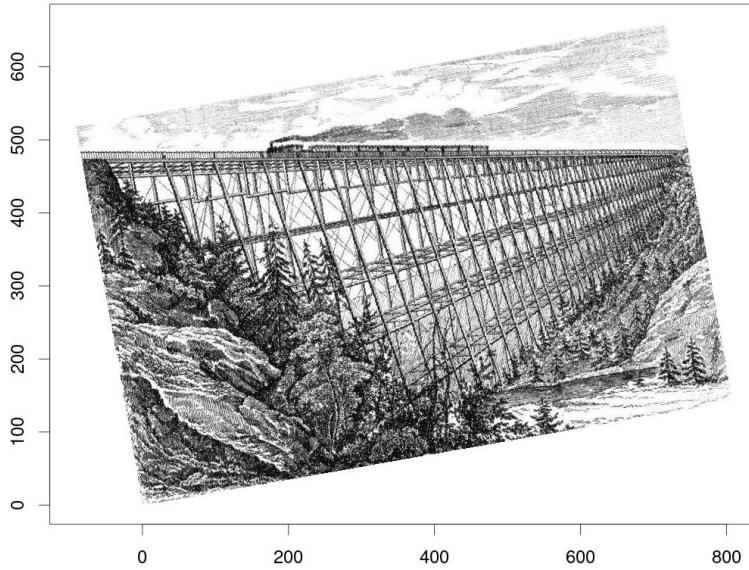


Figure 7.4: The same image rotated 10 degrees.

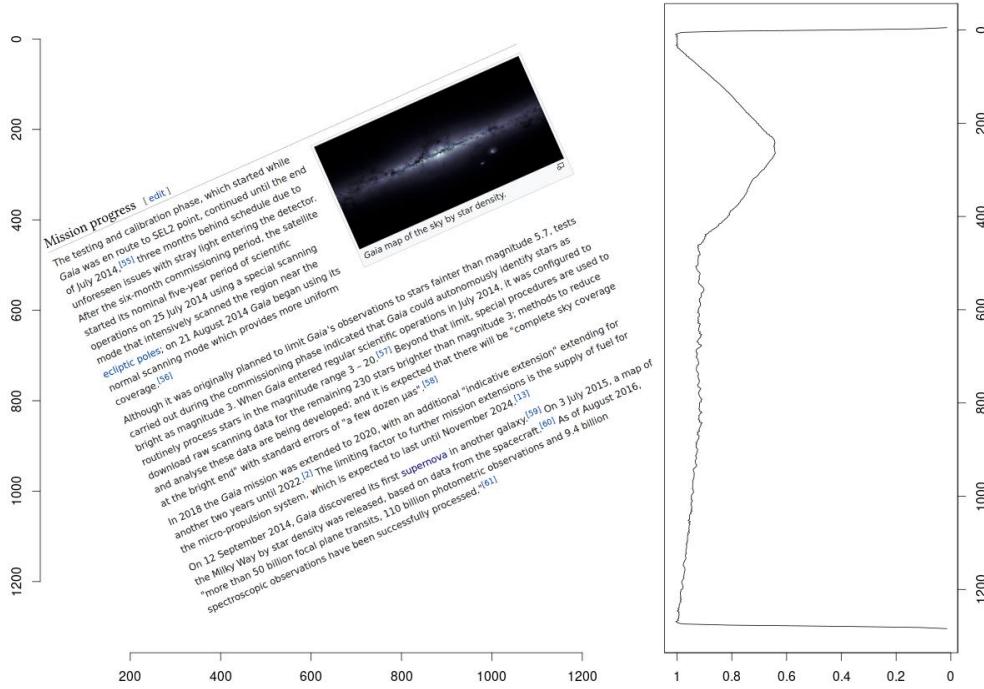


Figure 7.5: Image of a text page (left panel). It is rotated 24 degrees counterclockwise. Right panel depicts the gray value density along the vertical axis. The galaxy image in the form of a triangular dip, centered at row 200, is clearly visible. However, the text lines cannot be distinguished in the plot.

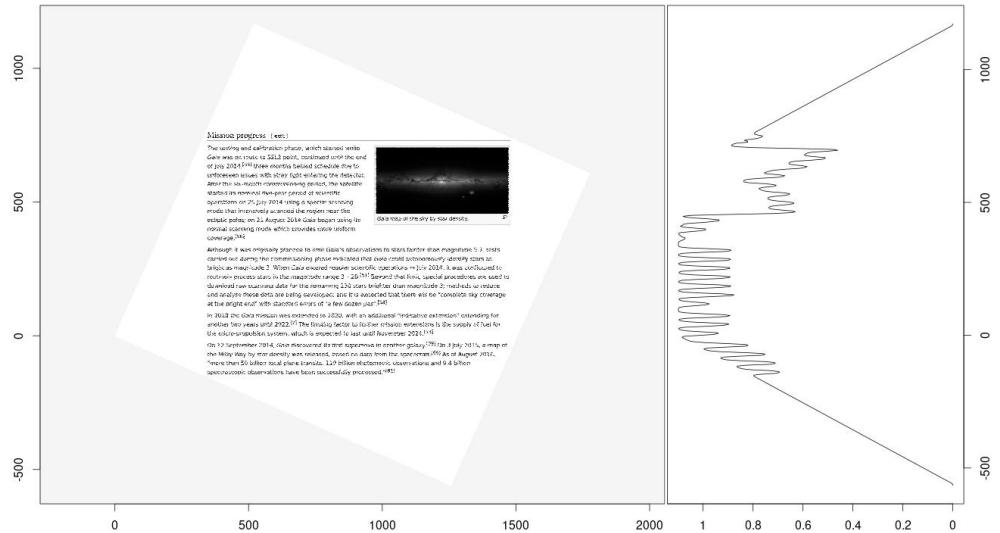


Figure 7.6: The same image as in Figure 7.5 but now rotated into correct position (left panel). The image is now visible as the rectangular dip with vertical sides. Now also the text lines are represented by a regular wavy pattern of lighter and darker stripes. Smooth slopes on both sides of the true image are related to the tilted white background embedded in the image.

7.3 Text as Data

As literate humans, we produce and read quite a lot of written text (we do not discuss voice here). So text *is* data and it contains a lot of information. But to process text on computer poses a number of challenges. To start with, text is non-numeric, and unlike images where we can easily see the pixel color values that are arranged in neat rows and columns, it is not obvious what might be the features in case of text.

Below, we discuss one simple way of using text, *document-term-matrix* (DTM). DTM can be done in different way, e.g. just by counting the occurrence of different words, or by calculating *term-frequency-inverse-document-frequency* (TF-IDF). But before we get to text itself, we have to talk about pre-processing, such as tokenization and stemming because raw text is often not the best choice for processing.

7.3.1 Preprocessing: Tokenization, Stemming, and Lemmatization

Text processing typically starts with simple methods to simplify the text by removing various features that are not relevant for current task. For instance, we may consider case of the word irrelevant, and we may want to consider different grammatical forms of the same word, e.g. *speaking* and *speaks* to be the same. This may be a good choice when we are doing topic modeling. But case may be very important for other task, e.g. when we want to extract proper names.

Tokenization

Text is normally analyzed at word level. This means we have to split it into words. This is relatively easy with English and other European languages where space and certain punctuation symbols are reliable word boundary markers. But other languages may not contain similar markers and hence the process is much more complex.

Even in English, we have a number of corner cases. For instance, what should we do with *don't*? Should we retain it as “don't”, convert it to “do not”, or just remove the apostrophe and make it into “dонт”? In contrary, what about names like *Kuala Lumpur*? Should it be retained as a single word containing space, or split into two words? We have to make such decisions depending on the task at hand. As the result may deviate from what we commonly call “words”, this process is usually called *tokenization* instead, and the results are *tokens*, not “words”.

But we do not *have* to use words. We may break written text down to individual characters, character pairs, or syllables instead. The smaller units may be useful where we have to guess the meaning of words from how they are written, from prefixes and suffixes they contain, or if they contain syllables that also occur in other, known words.

TBD: names to ethnic background

Stemming

Stemming is a process where common prefixes and suffixes are removed from the words. For instance, when encountering the word *studying*, we may remove the suffix *-ing* leaving the stem *study*. More advanced stemming algorithms may leave only the

part of stem that never changes, in this example, this would be *stud*. Note also that the never-changing-part of a word may differ between written and spoken language, or be completely missing (like in case of *go* and *went*).

Stemming helps us to see the common stems of related words, and we can for instance build a search engine that finds both *study* and *studying* when the user enters either of them. This is often what the users want when they search.

Lemmatization

However, stemming is a simplistic method that often fail to produce consistent stems for closely related words. For instance, a simple stemming algorithm may turn word *studying* into stem *study*, but the word *studies* will produce *studi*. As these stems are not identical, the search engine may not understand that the corresponding words are similar.

Lemmatization is a method that is in principle similar to stemming, but instead of just removing the standard suffixes, it uses morphological analysis of words to deduce the *lemma*, the standard base form of all the related words. Lemma is the standard form of words that is listed in dictionary. In the example above, the lemma of both *study* and *studies* is *study*. Needless to say, lemmatization is much more complex to implement than stemming and needs some sort of dictionary lookups.

Stopwords and Other Simplification

We often want to start by simplifying text. Typically one converts all words to lower case, removes punctuation, perhaps normalizes the grammatical constructs, and removes *stopwords*, common words like *and*, *not*, *but* that carry little information.

Obviously, whether capitalization, punctuation and stopwords are just a noise or actually helpful depends on the task. If you are predicting the topic of a news article, the common stopwords carry little information. However, when deducing authorship of a text, the subtle differences in stopword usage or the exact grammatical form of words used may turn out to be quite important.

7.3.2 *n*-grams

n-grams are just ordered sequences of *n* words (or other objects, such as letters or sentences). For instance, in case of document “The waiter opened the gate a little and looked out”, we can create the following bigrams (2-grams): (*the, waiter*), (*waiter, opened*), (*opened, the*), (*the, gate*), (*gate, a*), (*a, little*), (*little, and*), (*and, looked*), (*looked, out*). *n*-grams can be used in a similar fashion as tokens, their main advantage is that they preserve the order of the words. For instance, it is harder to deduce meaning of two tokens (unigrams) “do” and “not”, while a bigram (do, not) is has much more distinct meaning.

However, texts contain many more different *n*-grams than unigrams, and hence working with *n*-grams typically needs more resources and training data.

7.3.3 Bag of Words and Document-Term-Matrix

Prerequisites: Metric distance, vector norm 2.2.2, cosine similarity 6.1.2.

Statistical methods only work on numerical data, so before we can apply any common ML method on text we have to convert it into a numeric representation. *Bag of words* (BOW) is a simple and popular approach to transform texts into a numeric vector form, in essence just a frequency table of words in the text. An essential property of BOW is that it does not preserve the order of words. One can imagine throwing all the words into a bag, so for each document it will just contain the counts of the words but not their order. We obviously lose a lot of information by such “bagging”, sometimes it is useful, sometimes it is undesirable. Normally we work on many documents which may be short (like tweets) or long (like books). We can construct a BOW for each of these and stack them into a matrix, called *Document-Term-Matrix* (DTM).

DTM can be constructed in different ways, here we explain how to create it based on word counts (or more precisely, token counts), but one can choose to remember just the presence of words, not their counts. One can also construct bag-of-characters or bag-of-bigrams instead of bag-of-words.

In this form we represent documents as vectors of word counts in the vocabulary: first we collect all words in all the documents we analyze. This collection is called *vocabulary*. Say there are V words in the vocabulary. Now we can represent each document as a vector of length V , $\mathbf{x} = (x_1, x_2, \dots, x_V)^\top$, where each component x_j equals to the count of that word j in the text. If the word is not present, we set $x_j = 0$.

BOW-s are numeric vectors we can use for various mathematical operations. For instance, we can compute both Euclidean distance or cosine similarity between BOW-s. When stacking BOW-s horizontally underneath each other, we get a DTM, essentially a design matrix where features are the words, and feature values are the word counts in each document (observation).

Example 7.1: DTM of Laozi quotes

Let us create a DTM of two Laozi quotes: “*Knowing others is wisdom, knowing yourself is Enlightenment*”, and “*Mastering others is strength. Mastering yourself is true power*”.

These quotes together form vocabulary of size 10 (in alphabetical order) *enlightenment, is, knowing, mastering, others, power, strength, true, wisdom, yourself*. The first quote only contains words *enlightenment, is, knowing, others, wisdom, yourself* and hence the corresponding BOW is $\mathbf{x}_1 = (1, 2, 2, 0, 1, 0, 0, 0, 1, 1)^\top$. We must keep the word counts in a consistent order, normally in the same order as the words are in the vocabulary. The number “1” in the first position indicates that the word *enlightenment* is present 1 times, but for instance the words *is, knowing* are there two times, and words *mastering, power, strength, true* are not present at all. The second quote contains *enlightenment, is, knowing, others, wisdom, yourself* and hence its BOW is $\mathbf{x}_2 = (0, 2, 0, 2, 1, 1, 1, 1, 0, 1)^\top$. The vocabulary and both BOW-s are shown in the table below.

Table 7.1: Two BOW-s \mathbf{x}_1 and \mathbf{x}_2 , corresponding to the two Laozi quotes in the text. Both BOW-s, stacked horizontally underneath each other as in this table, form a numeric DTM that can be used in various machine learning models.

	enlightenment	is	knowing	mastering	others	power	strength	true	wisdom	yourself
\mathbf{x}_1	1	2	2	0	1	0	0	0	1	1
\mathbf{x}_2	0	2	0	2	1	1	1	1	0	1

Let us also compute Euclidean distance and cosine similarity between these two quotes. For the Euclidean distance we need their difference $\mathbf{x}_1 - \mathbf{x}_2 = (1, 0, 2, -2, 0, -1, -1, -1, 1, 0)^\top$ and hence $d_e(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{(\mathbf{x}_1 - \mathbf{x}_2)^\top \cdot (\mathbf{x}_1 - \mathbf{x}_2)} = \sqrt{13} = 3.606$. For cosine similarity, we need to compute the inner product $\mathbf{x}_1^\top \cdot \mathbf{x}_2 = 6$ and both norms, $\|\mathbf{x}_1\| = \sqrt{12} = 3.464$ and $\|\mathbf{x}_2\| = \sqrt{13} = 3.606$, and hence $c(\mathbf{x}_1, \mathbf{x}_2) = 6 / (\sqrt{12} \cdot \sqrt{13}) = 0.48$. In some applications we may also want to remove the stopword *is*.

The advantage of DTM is it's simplicity. Creating a BOW for given task requires no training data and can be done fast and easily. Sparseness of typical DTM helps to increase the algorithm speed and decrease the memory requirements, as typical vocabularies contain 10,000 to 100,000 words while most of the documents do not contain most of the words. Hence the DTM-s are in practice mostly filled with zeros, and we can use sparse matrices as the underlying data structures.

DTM-s have two major disadvantages. First, they are large. Typical word-based DTM-s contain 10,000–100,000 words, and hence every document, even just a tweet of a single word, contains this many numbers. This may make large DTM-s slow and sluggish.¹ Second, by construction they do not store the order of words. Whatever the order of words, the data looks identical. A potential way to address this issue is to use a bag of bigrams instead of BOW. As bigrams retain the order of words, such a bag will contain much of the information of the original word order. However, as there are many more bigrams compared to words, the dimensionality problem will get worse.

A potential solution to the dimensionality problem stems from the typical word distribution in natural languages. While there is a core of very frequent words, most of the words in a vocabulary are rare. Note that in practice there is always a large number of rare words. If we increase the sample size (the number or size of documents), we sample a larger number of formerly rare words so those are not rare in our BOW any more. But in actual applications, larger documents will always contain many even less common words, misspellings, names and acronyms, so the problem of a large number of infrequent words does not go away. But often we can just disregard

¹This sounds like contradicting the praise of simplicity and sparsity in the previous paragraph, but it is not. Sparsity often helps, but it is not a cure against all inefficiencies. Small dense matrices are still much more efficient than large sparse matrices.

such words—seeing a word only a few times is arguably too little to make any inference about its role for language models. Hence a common strategy is to exclude all words that are less frequent than a given threshold.

7.3.4 TF-IDF

As DTM is effectively a numeric design matrix, we can use it with a plethora of traditional ML methods. For instance, we may categorize texts using k -NN and cosine similarity. Unfortunately, this measure may not perform very well in practice. First, often the words that are common in both texts are popular ones that carry little distinctive power. Even if we remove obvious stopwords, there are still many common words that occur repeatedly in most of the texts. Second, less popular words tend to be used in a “bursty” way, so if one word is already used in a document, it will be very likely used again. Just word counts will put too much weight on a few words that are used many times in the texts. As a remedy, one may use term-frequency inverse document frequency (*TF-IDF*) transformation. There are many different specific definitions of TF-IDF in the literature, here we follow the approach of [Murphy \(2012, p 482\)](#).

First, we transform the word counts into *tf* form as

$$\text{tf}(x_{ij}) = \log(1 + x_{ij}) \quad (7.3.1)$$

where x_{ij} is the count of word j for the text i . This transformation suppresses large counts of single words, but is still more informative than just binary contains/does not contain features. By adding 1 to x_{ij} we ensure TF of a missing word is zero, and TF for every word present in document is positive.

Second, for each word j in the vocabulary, we define *idf* as logarithm of the inverse of number of documents that contain the word j . Normally we adjust the inverse a bit, e.g. we take inverse of 1 plus the number of documents in order to avoid cases where a vocabulary word is not found in any document.² Formally, we may define *idf* as

$$\text{idf}(j) = \log \frac{N}{1 + \sum_{i=1}^N \mathbb{1}(x_{ij} > 0)}. \quad (7.3.2)$$

$\mathbb{1}(x_{ij} > 0)$ is the indicator function that equal to one if the document i contains word j , and hence $\sum_{i=1}^N \mathbb{1}(x_{ij} > 0)$ is the count of documents that contain the word j . We also use the total number of documents N as numerator. This is a form of normalization where the words that are present in all documents will have the fraction of $N/(1 + N) \approx 1$ and hence its logarithm $\text{idf} \approx 0$. In the opposite end, IDF for a word that is found in none of the documents is $\log N$ and for a word in a single document only, $\text{idf} = \log N/2$. Hence *idf* assigns to words that are present in many documents low weight, and words that are present in only a few documents high weight. Note that *idf* assigns a single value for each word across all documents. IDF is a word-specific value while TF-vectors are different for each BOW.

²There may be several reasons that a word that is in no document finds its way to the vocabulary. For instance, one may use a standard vocabulary, derived from other documents. Also, training data typically contains words that are in no validation document.

Finally, the TF-IDF transformation for word j is defined as

$$\text{tf-idf}(x_{ij}) = \text{tf}(x_{ij}) \cdot \text{idf}(j) \quad j \in 1 \dots K. \quad (7.3.3)$$

Note that TF-IDF is not made of single document alone—it is a transformation of the complete DTM \mathbf{X} . While each single BOW (a row in the original data matrix \mathbf{X}) has been transformed into a row of TF-IDF matrix $\tilde{\mathbf{X}}$, the transformation needs information about how common are the words across all documents. In some ways it is similar to [feature normalization](#).

Example 7.2: TF-IDF of Laozi quotes

Let us TF-IDF transform the DTM of the Laozi quotes, *Knowing others is wisdom, knowing yourself is Enlightenment* and *Mastering others is strength. Mastering yourself is true power*, presented in Table 7.1.

Table 7.2 shows the results. The columns (features) are the $V = 10$ vocabulary words and two first lines represent the DTM as in Table 7.1. The following two lines, tf_1 and tf_2 show the corresponding tf -terms, essentially just log-transforms of the DTM. The row idf is the IDF term. It is $\log 2/(1+2) \approx -0.41$ for words that are in both documents and $\log 2/(1+1) = 0$ for words that are in a single document only. Although we do not have any such example, it would be $\log 2 = 0.69$ for words that are in none of the quotes. Finally, the last rows tf-idf_1 and tf-idf_2 are just the corresponding tf -terms multiplied by idf . These two rows form the TF-IDF-transformed data matrix $\tilde{\mathbf{X}}$.

Table 7.2: Example vocabulary, bag-of-word vectors, and TF-IDF transformation for quotes: “Knowing others is wisdom, knowing yourself is Enlightenment” and “Mastering others is strength. Mastering yourself is true power”.

	enlightenment	is	knowing	mastering	others	power	strength	true	wisdom	yourself
\mathbf{x}_1	1.00	2.00	2.00	0.00	1.00	0.00	0.00	0.00	1.00	1.00
\mathbf{x}_2	0.00	2.00	0.00	2.00	1.00	1.00	1.00	1.00	0.00	1.00
tf_1	0.69	1.10	1.10	0.00	0.69	0.00	0.00	0.00	0.69	0.69
tf_2	0.00	1.10	0.00	1.10	0.69	0.69	0.69	0.69	0.00	0.69
idf	0.00	-0.41	0.00	0.00	-0.41	0.00	0.00	0.00	0.00	-0.41
tf-idf_1	0.00	-0.45	0.00	0.00	-0.28	0.00	0.00	0.00	0.00	-0.28
tf-idf_2	0.00	-0.45	0.00	0.00	-0.28	0.00	0.00	0.00	0.00	-0.28

The TF-IDF-transformed data $\tilde{\mathbf{X}}$ is a similar numeric data matrix like DTM and can be used in different ML models as any other numeric data. It gives sometimes quite a substantial improvement in the modeling accuracy. It is also easy and fast to perform, involving just a few operations that can be easily vectorized.

7.3.5 Word embeddings

In Section 7.3.3 we introduced *bag of words* (BOW), a way to encode text as vectors. BOW is essentially a sum of one-hot-encoded word vectors. These vectors are based on the vocabulary, a list of all tokens in the text, typically arranged in an alphabetical order. But such one-hot encoded vectors have a few downsides: first, their dimension is very large (typically in tens of thousands), and second—they do not convey any information about the meaning of the words. For instance, if you compare the one-hot representation of “man”, “bro” and “dude”, then there is no way to tell that these words refer to closely related concepts. All these are long vectors of zeros, with a single “1” in an apparently random location.

Word embeddings is a method to address these two shortcomings. Typical word embeddings are vectors of about 100 components for each word, and similarity of the embedding vectors means that the underlying concepts are similar too. The idea of embeddings is that the word vectors are not based on some kind of arbitrary encoding (and alphabetical order is arbitrary when talking about word meaning), but are based on the *context*, other words that are located nearby. Hence the words that are used in a similar context will have similar embeddings.

Below, we discuss a few ways to construct word embeddings. We start with long embeddings that only address the concept similarity but leave us with similar long vectors like one-hot encoding. Thereafter we discuss the two popular approaches: *word2vec* and *glove*.

Term co-occurrence matrix

Embeddings are based on *term co-occurrence matrices* (TCM-s). This is essentially a set of frequency tables, where for every term (word) in the text, we list how many times any other word occurs in the same context.

In order to compute TCM, one has to decide what *context* to analyze. In the broad sense, context means the other words that occur near the given word but in practice it must be defined in a precise manner. For instance, context may be three words before and three words after the current word, all weighted equally. Alternatively, we may define it as the five preceding words, weighted inversely to the distance from the current word. There are many other options.

We may denote the TCM as \mathbf{X} where the elements, x_{ij} denote how many times the term j occurs in the context of term i . If the context is symmetric (we are including both the preceding and trailing words with similar weight), the TCM is symmetric: $x_{ij} = x_{ji}$.

As an example, consider the sentence “of the United Nations”. If we look at the symmetric context—one word before and after, then the context of “of” is *the* and the context of “the” is (*of*, *United*). “the” occurs once in the context of “of” and the way around. However, if our context only contains a single preceding word, then the context of “of” is empty while that of “the” contains just *of*. Hence TCM will not be symmetric.

Example 7.3: TCM of Laozi quotes

Let's use the same Laozi quotes and in Example 7.1 to compute the corresponding term co-occurrence matrix. As a reminder, the quotes are “*Knowing others is wisdom, knowing yourself is Enlightenment*”, and “*Mastering others is strength. Mastering yourself is true power*”. The corresponding long word vectors are

	enlightenment	is	knowing	mastering	others	power	strength	true	wisdom	yourself
enlightenment	1	0	0	0	0	0	0	0	0	0
is	0	1	0	0	0	0	0	0	0	0
knowing	0	0	1	0	0	0	0	0	0	0
mastering	0	0	0	1	0	0	0	0	0	0
others	0	0	0	0	1	0	0	0	0	0
power	0	0	0	0	0	1	0	0	0	0
strength	0	0	0	0	0	0	1	0	0	0
true	0	0	0	0	0	0	0	1	0	0
wisdom	0	0	0	0	0	0	0	0	1	0
yourself	0	0	0	0	0	0	0	0	0	1

Note that these are long vectors of all vocabulary words, not the encoded text itself. The first quote, encoded using these long vectors, is

	enlightenment	is	knowing	mastering	others	power	strength	true	wisdom	yourself
knowing	0	0	1	0	0	0	0	0	0	0
others	0	0	0	0	1	0	0	0	0	0
is	0	1	0	0	0	0	0	0	0	0
wisdom	0	0	0	0	0	0	0	0	1	0
knowing	0	0	1	0	0	0	0	0	0	0
yourself	0	0	0	0	0	0	0	0	0	1
is	0	1	0	0	0	0	0	0	0	0
enlightenment	1	0	0	0	0	0	0	0	0	0

Here we chose to put the words in the quote in successive rows, the columns correspond to the vocabulary entries.

Let's choose a simple symmetric context, one word before and one word after the current term. The word “is” is included twice in the first quote. The first “is” is associated with context (*others, wisdom*), and the second one with (*yourself, enlightenment*). The BOW of the first context c_1 is

enlightenment	is	knowing	mastering	others	power	strength	true	wisdom	yourself
0	0	0	0	1	0	0	0	1	0

and for the second context, \mathbf{c}_2 is

enlightenment	is	knowing	mastering	others	power	strength	true	wisdom	yourself
1	0	0	0	0	0	0	0	0	1

The combined context vector for the first quote is just a sum of these two, $\mathbf{c} = \mathbf{c}_1 + \mathbf{c}_2$ (ignoring the component names):

$$\begin{array}{cccccccccc} \hline & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ \hline \end{array}$$

We can compute the context vector for the other quote in an analogous fashion. When repeating the process for all words for both quotes, we get the complete TCM

	enlightenment	is	knowing	mastering	others	power	strength	true	wisdom	yourself
enlightenment	0	1	0	0	0	0	0	0	0	0
is	1	0	0	0	2	0	1	1	1	2
knowing	0	0	0	0	1	0	0	0	1	1
mastering	0	0	0	0	1	0	1	0	0	1
others	0	2	1	1	0	0	0	0	0	0
power	0	0	0	0	0	0	0	1	0	0
strength	0	1	0	1	0	0	0	0	0	0
true	0	1	0	0	0	1	0	0	0	0
wisdom	0	1	1	0	0	0	0	0	0	0
yourself	0	2	1	1	0	0	0	0	0	0

Simple embeddings: long vectors

The simplest and the most intuitive way of constructing word embeddings is to use the bag-of-words of the context of the words as their embeddings. This proceeds broadly in the following manner:

1. Construct the vocabulary. All common considerations hold here, e.g. it may contain all tokens, or only tokens that are frequent enough, and one may choose to keep or remove numbers.
2. For each word (token) in the text, find its *context*, the closest words around it. Note that we talk about each individual occurrence in the text, i.e. for a particular token, we have as many contexts as many times it occurs in the text.
3. Construct bag-of-words of the context. Again, there is a multitude of options, e.g. one may just look for presence of the words in the context, or one may count these words (and weight by distance).
4. Aggregate the bag-of-words for all similar tokens. Again, there are multitude of ways, e.g. one can just add them (as vectors). These aggregated BOW-s are essentially embeddings.
5. Finally, in order to make the embeddings comparable, we need to normalize those somehow—remove the differences caused by the fact that some words are used more frequently in more different context. Usually it is achieved by setting its Euclidean norm to 1.

It is fairly obvious that if certain words tend to be used in a similar context, then their embeddings, computed in this manner, will also be similar. The more different are the contexts, the more different are the embeddings too.

However, such embeddings have a few downsides. First, it is obvious that the dimension of the embeddings is the same as the vocabulary size, and hence the vectors are long. Second, and more importantly, such tables suffer from the fact that the most frequent words will have disproportionate impact of the vectors. The most frequent words like *the*, *and* and *to* may occur thousands of times in the contexts, but they carry little information about the words' usage. So we'll discuss the other methods to compute the embeddings below.

Example 7.4: Long embedding vector of Laozi quotes

Now let's transform the TCM entry for “is” to the corresponding normalized embeddings vector. As a reminder, the TCM entry was (see Example 7.3): $\mathbf{c} = (1, 0, 0, 0, 2, 0, 1, 1, 1, 2)$. This vector has Euclidean norm $\|\mathbf{c}\| = 3.46$, and hence the normalized embedding vector for *is* is

$$\|\mathbf{e}\| = \mathbf{c}/\|\mathbf{c}\| = (0.29, 0, 0, 0.58, 0, 0.29, 0.29, 0.29, 0.58).$$

Short embedding vectors: word2vec and GloVe

Creating long embeddings vectors, frequency tables as we did above, is easy and intuitive, but unfortunately the results are less than perfect. First, such embedding vectors are long, as long as the size of the input vocabulary. But more importantly—the vectors are much more dependent on the more frequent context words, rare words that are very important for the context may be almost ignored when calculating the resulting similarity. In practice, one computes the embeddings in a different way. The most popular approaches are *word2vec* and *GloVe*.

GloVe GloVe (Pennington *et al.*, 2014) is computed from TCM in a somewhat similar way as the long embedding vectors (see [Section 7.3.5 Simple embeddings: long vectors](#), page 332) are based on TCM.

Table 7.3 shows the most similar words for a selection of words, and the corresponding cosine similarities underneath. These are computed from the small *Common Crawl* dataset, an internet scraping project from 2012.³ The similar words are often strikingly obvious. For instance, for common words like *woman* and *hiking*, the most similar words are of no surprise. The word embeddings also recognize geography—the *Thailand*–related words are all from South-east Asia, and the most similar of these, *Bangkok*, is the Thai capital. Embeddings also recognize local geography: *Bainbridge* is a small island outside Seattle, of the closest tokens *Bremerton*, *Poulsbo*, and *Bellevue* are cities nearby, *Kitsap* is the county where the island is located, and *Vashon* is another nearby island. Further down are numbers and weekdays, not only are the other numbers and weekdays the most similar ones, but they are also broadly in correct order. It can also relate top US-politicians (Hillary Clinton was the U.S. foreign secretary) to other politicians, Russian women names to other Russian women names, and smileys to smileys.

A slightly different result, based on the larger Common Crawl, is presented in Figure 7.7. The upper panel represents the word “trade” and the lower panel the word “regime”. As one can see, “China” was the most common country that was mentioned in the same context as “trade”. The lower panel shows that “regime” is mainly associated with Iran, but also with Libya and Syria, reflecting the Arab Spring events of 2011.

Obviously, the embeddings reflect the texts they are based on, and include all the representation problems and biases that we see in actual data.

TCM: term-co-occurrence matrix counts the number of times one word occurs in the context of the other word. See [Section 7.3.5 Term co-occurrence matrix](#), page 329.

TBD: word2vec

TBD: Converting text to features. BOW/embeddings + n-grams + part-of-speech + other kind of features

³There are two Common Crawl datasets: the [smaller one](#) includes 42B tokens, the embeddings matrix’ (uncased) size is 1.9M. The [larger one](#) contains 840B tokens, the embedding matrix’ (cased) size is 2.03GB. See more at <https://nlp.stanford.edu/projects/glove/>.

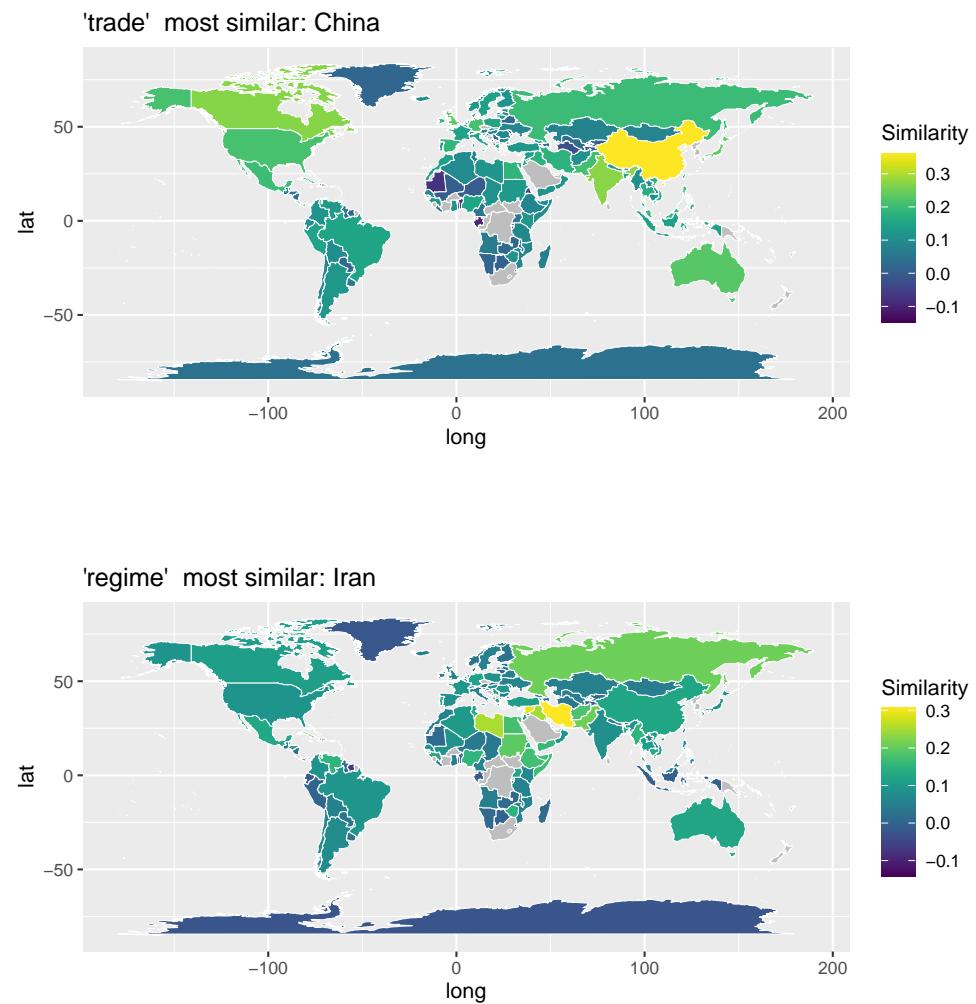


Figure 7.7: Similarity between words “trade” (top), “regime” (bottom), and the country names. Multi-word country names, such as “Saudi Arabia” cannot be used here, so these countries are left gray.

Word	Most similar words and the corresponding cosine similarity					
woman	man 0.80	girl 0.76	women 0.71	she 0.70	lady 0.69	mother 0.68
hiking	biking 0.79	backpacking 0.75	trekking 0.72	camping 0.71	climbing 0.70	kayaking 0.69
thailand	bangkok 0.75	cambodia 0.72	malaysia 0.70	laos 0.69	asia 0.67	singapore 0.67
bainbridge	bremerton 0.59	poulsbo 0.57	kitsap 0.51	bellevue 0.50	vashon 0.50	marietta 0.50
tuesday	wednesday 0.98	thursday 0.98	monday 0.97	friday 0.94	saturday 0.86	sunday 0.85
seven	eight 0.94	nine 0.94	six 0.94	five 0.94	four 0.91	three 0.89
clinton	hillary 0.82	obama 0.79	barack 0.76	mccain 0.75	bush 0.75	biden 0.69
olga	maria 0.57	elena 0.57	irina 0.56	tatiana 0.56	kurylenko 0.55	alexandra 0.53
:)	;) 0.94	:-) 0.93	:d 0.91	=) 0.88	;-) 0.85	!! 0.84

Table 7.3: A selection of words and the corresponding similarities from Common Crawl data (42B token, 300D vectors).

Chapter 8

Machine Learning Techniques

8.1 Loss Function and Non-Linear Optimization

Statistical problems typically require estimation of certain parameters (fitting the model) based on data. The parameters may be interesting itself, or these may be needed for predictions, hypothesis testing or other reasons. For instance, in case of linear regression, these are parameter β , in case of regression trees these are the splitting and stopping rules for each branch. More complex models, such as neural-network based image recognition tasks can be imagined as many layers of linear regression models on top of each other and can contain millions or hundreds of millions of parameters.

If we move beyond the simplest cases, it is completely infeasible to find the best parameter values manually. We need methods to do this on computer, to do it fast, and in a reliable fashion. Typically this proceeds through *non-linear optimization*, a technique where a certain function (called *loss function* or *objective function*) is minimized or maximized by manipulating the parameters. The parameter value that results in the smallest loss value is the best parameter, the solution.¹

Next we will look at some details of non-linear optimization. These notes will only give a brief overview of the methods in order to prepare you for applications.

8.1.1 Loss Function

A large class of statistical models involves a function, *loss function*, that describes how “bad” is the model. For instance, linear regression is defined through sum of squared errors as

$$SSE(\beta) = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta). \quad (3.1.64)$$

¹Not all models require such non-linear optimization. For instance, k -nearest neighbors do not contain any parameters, and Naive Bayes parameters (conditional probabilities) can be computed directly from the data without any optimization.

SSE is a measure of model “badness” and hence $SSE(\beta)$ can be understood as the loss function. It depends on the parameter vector β and hence we can manipulate β to get larger or smaller losses. Non-linear optimization is a technique (more precisely, a set of many different techniques) that systematically manipulate the parameter in order to find the smallest loss.²

Why do we want the smallest loss? If we are interested in prediction, then SSE is one of the most obvious measures of the model predictive power (or rather lack of it as large SSE corresponds to low power). So in this case it is almost trivially true that small loss is equivalent to good model. If we are interested in inference, we may need additional assumptions regarding the “true” model and data.

Let us illustrate the loss function with a figure. Figure 8.1 shows the Hubble regression (see Example 3.1 on page 128). The regression is in a form

$$v_i = \beta_0 + \beta_1 R + \epsilon_i \quad (8.1.1)$$

where v is velocity of galaxies (km/s) and R is distance (Mpc). So this model only has two parameters and hence it can be visualized. On one axis of the figure we display β_0 , on the other β_1 and the vertical axis describes SSE . The loss function describes an elongated parabolic surface with minimum at $\beta^* = (-40.4, 453.9)$. This is the solution to the linear regression problem shown in Example 3.1 above.

Sometimes we want to maximize a function instead of minimizing it, in that case it is often referred to as *objective function* instead of *loss function*. More strictly speaking, objective function is a more general term, it is a function that should be either minimized or maximized (i.e. *optimized*) in order to find the solution. So loss function is also an objective function.

8.1.2 Maximum Likelihood (ML)

Maximum Likelihood is a M -estimator where the parameter estimate is established by maximizing its *likelihood* (in practice, almost always, log-likelihood). Likelihood is essentially the probability to observe the data, given its parameter values. This assumes we have established the data generating process (DGP) for the observations. For instance, in case of a sample of random variables (RV), DGP is essentially the distribution we assume the data is originating from. In case of a regression model we may assume that our independent variables (features) are given and exogenous, while the response variable is calculated based on the features and a random disturbance term. Note that with *data*, we mean all data that goes into the model, including the explanatory and response variables.

A few examples

Below are a few examples of the probability to observe the data, equivalent to the likelihood function.

²As a side note—we do not actually need non-linear optimization for linear regression. This is the only statistical model where we can do the optimization analytically and directly compute the solution.

Coin Toss We toss a coin 4 times. We get H twice and T twice. We know it is a fair coin. This is a binomial process. The probability to observe k heads in n coin tosses where probability of a head is p is

$$\Pr(X = k) = C_k^n p^k (1 - p)^{n-k}. \quad (8.1.2)$$

(Note: if you are doing numeric computations, you may prefer the R function `pbinom()` instead.) The multiplier C_k^n counts how many different ways there are to observe k heads in n tosses. This is the binomial probability mass function (pmf).

The probability to observe two heads and two tails is accordingly

$$\Pr(H,H,T,T) = C_4^2 0.5^2 (1 - 0.5)^2 = 0.375. \quad (8.1.3)$$

Independent Normals We are given a sample X_1, X_2, \dots, X_n of independent draws from standard normal distribution. Note the normality of the data must either be assumed, or in rare cases somehow learned (e.g. through theoretical considerations).

Note that as we are dealing with a continuous distribution, we cannot directly write down the probability of the data. However, as the density function (pdf) gives us the probability per unit interval, we can write the probability of data per unit interval:

$$\Pr(X_1, X_2, \dots, X_n) = \phi(X_1) \cdot \phi(X_2) \cdots \phi(X_n) \quad (8.1.4)$$

where $\phi(\cdot)$ is the normal pdf. Note that the assumption of the independence allows us to write the probability as a product of individual probabilities.

Examples Involving Unknown Parameters

The previous examples did not contain any unknown parameters, so there was little left to be estimated. We can make these examples more interesting by including a parameter.

Coin Toss We toss a coin 4 times. We get H twice and T twice. We don't know whether it is a fair coin. What is the best estimate for p , the probability to receive a head?

Let's start with the probability to observe the data. As in (8.1.2) and (8.1.3) we have:

$$\Pr(H,H,T,T) = C_4^2 p^2 (1 - p)^2 = 6 p^2 (1 - p)^2 \equiv \mathcal{L}(p). \quad (8.1.5)$$

$\mathcal{L}(p)$ is the *likelihood function*. It is essentially the same probability, just we stress that the main arguments of interest are the parameters. Binomial process only has a single parameter p .

Which value of p will give the highest $\mathcal{L}(p)$ value? Let's start with a grid search (Figure 8.2). As one can see, the optimal value is 0.5. This is not surprising, as intuitively the best estimate for p is simply the sample mean.

In practice, one almost always works with log-likelihood instead of likelihood. This is for two reasons, first the likelihood values are in realistic settings often too small for current computers to represent; and also log-likelihood has a number of attractive

statistical properties. Obviously, as log is a monotonic function, the optimum parameter value will remain the same. In this simple example, we can analytically solve for the optimal value:

$$\ell(p) \equiv \log \mathcal{L}(p) = \log 6 + 2 \log p + 2 \log(1 - p). \quad (8.1.6)$$

The optimum can be solved through a simple calculus. The optimum condition is

$$\frac{\partial}{\partial p} \ell(p) = \frac{2}{p} - \frac{2}{1-p} = 0 \quad (8.1.7)$$

from where follows that the optimal $\hat{p} = 0.5$.

Mean of Normals We are given a sample X_1, X_2, \dots, X_n of independent draws from a normal distribution with variance one. Unlike in the case above, we don't know what is the mean of the distribution. Let's estimate it by ML.

For a single observation we have

$$\Pr(X = x; \mu) = \phi(x - \mu) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(x - \mu)^2\right). \quad (8.1.8)$$

As before, for the numeric analysis you may prefer the R function `dnorm` instead of this expression.

For n independent normals

$$\begin{aligned} \Pr(X_1 = x_1, X_2 = x_2, \dots; \mu) &= \\ &= \phi(x_1 - \mu) \cdot \phi(x_2 - \mu) \cdot \dots \cdot \phi(x_n - \mu) = \\ &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(x_1 - \mu)^2\right) \times \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(x_2 - \mu)^2\right) \times \dots \\ &\quad \times \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(x_n - \mu)^2\right) \end{aligned} \quad (8.1.9)$$

This is essentially the likelihood function $\mathcal{L}(\mu)$. For log-likelihood we have

$$\begin{aligned} \ell(\mu) &= \\ &= -\frac{1}{2}(\log 2 + \log \pi) - \frac{1}{2}(x_1 - \mu)^2 - \frac{1}{2}(\log 2 + \log \pi) - \frac{1}{2}(x_2 - \mu)^2 - \\ &\quad \dots - \frac{1}{2}(\log 2 + \log \pi) - \frac{1}{2}(x_n - \mu)^2 = \\ &= -\frac{n}{2}(\log 2 + \log \pi) - \frac{1}{2} \sum_{i=1}^n (x_i - \mu)^2. \end{aligned} \quad (8.1.10)$$

Our next problem is to maximize this log-likelihood. First, note that the first term in the last row in (8.1.10) does not contain the parameter μ . Hence it drops out when we take derivative of it with respect to μ . Second, note that what is left is maximizing $-\frac{1}{2} \sum_{i=1}^n (x_i - \mu)^2$ which is equivalent to minimizing sum of squared deviations. Hence least squares estimator is equivalent to ML estimator in case of normal disturbances.

This problem is easy to be solved analytically. From the optimum condition we know:

$$\frac{\partial}{\partial \mu} \ell(\mu) = -\frac{\partial}{\partial \mu} \left(\frac{1}{2} \sum_{i=1}^n (x_i - \mu)^2 \right) = \sum_{i=1}^n (x_i - \mu) = 0 \quad (8.1.11)$$

and hence ML estimator $\hat{\mu} = \sum_i x_i / n$. Not surprisingly, the intuitive estimator of distribution mean, the sample mean, also turns out to be it's ML estimator.

TBD: elliptical curves, overshooting, flat areas, local minima

TBD: learning rate

TBD: feature scaling

8.2 Gradient Ascent

Prerequisites: Vectors and matrices, gradient

Gradient Ascent (GA) and it's mirror image Gradient Descent (GD), is a popular method to find maxima and minima (collectively called *optima*) of functions. Gradient ascent is widely used in various machine learning applications, it also serves as a basis for many more complex methods, such as Newton-Raphson. While one can easily find analytic solutions for the optimum of simple functions, such as quadratic function, this is not possible for more complex cases. The “more complex” includes almost all objective functions we encounter in machine learning practice, the linear regression being the only notable exception. So we have to rely on numerical computations, usually referred as *non-linear optimization*, to find the solution. GA is one of the most popular of these methods.

The GA idea in a 2-D case is depicted in Figure 8.3.

1. Start with an initial guess \mathbf{x}^0 of the location of the maximum. Note: I denote by superscript 0 the initial vector, it's components are denoted by subscripts: $\mathbf{x}^0 = (x_1^0, x_2^0)$.
2. Compute the gradient $\nabla f(\mathbf{x}^0)$.
3. Now take a step at the direction of the gradient. This leads to a new location $\mathbf{x}^1 = \mathbf{x}^0 + R \cdot \nabla f(\mathbf{x}^0)$. Scalar R , *learning rate*, determines the length of the step. As the gradient is pointing uphill, the function value at \mathbf{x}^1 is larger than at \mathbf{x}^0 .
4. Now repeat the process choosing \mathbf{x}^1 as the starting point. This gives you the next approximation \mathbf{x}^2 .
5. Repeat until gradient is close to zero and the function value does not improve any more.

GA is similar to climbing a hill in [whiteout conditions](#) (or in total darkness). The ground is white, the sky is white, and you cannot see more than a step or two around you. How will you get to the top of the hill? Using GA! You start wherever you are (this is your \mathbf{x}^0). You feel which way the ground is rising (this is your gradient). Now

you take a few steps in that direction (this is your \mathbf{x}^1). You repeat the process until you have reached flat ground. This is the hilltop.

Let's take a concrete 2-dimensional example. Let's make one step of GA for the function $f(\mathbf{x}) = x_1 \cdot \log x_2$. You can easily see it's gradient is $\mathbf{g}(\mathbf{x}) = (\log x_2, -x_1/x_2)'$.

1. Pick a starting point. Let's choose $\mathbf{x}^0 = (1, 1)'$. The function value at that point is $f(\mathbf{x}^0) = 0$. See Figure 8.4.
2. The gradient at this point is obviously $\nabla f(\mathbf{x})|_{\mathbf{x}=(1,1)'} = (\log 1, -1/1)' = (0, 1)'$.
3. Now take a step from \mathbf{x}^0 along the gradient. But first we have to choose the learning rate R . Let's pick $R = 0.5$. Now we have

$$\mathbf{x}^1 = \mathbf{x}^0 + R \cdot \nabla f(\mathbf{x}^1) = (1, 1)' + 0.5 \cdot (0, 1)' = (1, 1.5)' \quad (8.2.1)$$

The new function value is $f((1, 1.5)') = 1 \cdot \log 1.5 = 0.405$. Indeed, we moved uphill.

4. Now we can repeat the process until reach the maximum – although note that this particular function does not possess a maximum. So we stop here \odot

This particular example is illustrated on Figure 8.4 by contour plot. Note that at \mathbf{x}^0 , the gradient points straight up as the function is constant along x_1 at our initial point $(1, 1)'$, and hence we move along the direction of steepest ascent at that point. However, as R is relatively large, the steepest ascent direction at \mathbf{x}^1 is slightly different.

Now let's describe the algorithm in more detail for the general n dimensional case.

1. Pick an initial value of the parameter $\mathbf{x}^0 = (x_1^0, x_2^0, \dots, x_n^0)'$. It is always good idea to choose parameters as close to the actual maximum as you can as this may have a huge impact on speed. (Hint: the current optimum is most likely close the place where it was in your previous run!) Often though you have little guidance about how to choose good starting values.
2. Compute the gradient of your objective function $\nabla f(\mathbf{x})$ at $\mathbf{x} = \mathbf{x}^0$.
3. Take a step from \mathbf{x}^0 in the gradient direction. The step should be neither too long (you may overshoot and land on the other side of the hill) nor too short (it takes too long time to find the maximum). So we employ the learning rate R and take the step of length $R \cdot \nabla f(\mathbf{x})|_{\mathbf{x}=\mathbf{x}^0}$. This will land you in a new place we call \mathbf{x}^1 :

$$\mathbf{x}^1 = \mathbf{x}^0 + R \cdot \nabla f(\mathbf{x})|_{\mathbf{x}=\mathbf{x}^0} \quad (8.2.2)$$

This is our new best bet for the location of maximum. It is probably not a perfect place, but unless your code is wrong (or the function constant at \mathbf{x}^0), it is a better place than \mathbf{x}^0 .

Note: here is the only point of difference between GA and GD algorithms. If we want to move downhill, we have to take a step to the direction of the steepest descent, i.e. a step to the *opposite to the gradient*, and hence the updating rule is $\mathbf{x}^1 = \mathbf{x}^0 - R \cdot \nabla f(\mathbf{x})|_{\mathbf{x}=\mathbf{x}^0}$.

Choice of a good R value is important. This is a parameter of your model, although not one that is directly related to the process you are modeling. Such parameters are typically called *hyperparameters*. Unlike many other hyperparameters, such as choice of k for k -nearest neighbors, this one will probably not affect your final results, just the speed of convergence (and it may determine if your model will converge in the first place).

4. Are we in the correct place? There are several ways to check this:

- (a) Gradient is very small. Say, $\|\nabla f(\mathbf{x}^1)\| < \epsilon^g$, where $\|\cdot\|$ is norm (length) of the vector, and ϵ^g is a small number, say 10^{-6} . Small gradient indicates that the function is flat at that point, and differentiable functions are flat at maximum.
- (b) Function value does not grow much any more. Say, $|f(\mathbf{x}^1) - f(\mathbf{x}^0)| < \epsilon^f$ where ϵ^f is another small number.
- (c) One may suggest more conditions, for instance about relative size of gradient.

Typically we only need one of the criteria above: if gradient is close to zero, the function stops growing, and vice versa.

These conditions are called *stopping criteria*. In practice, you always need an additional, bail-out criterion: stop if the process has been repeated too many times already. This is because we too often choose too small learning rate, run into numerical problems, or have coding errors.

5. If we are in correct place, stop here. If not, set $\mathbf{x}^0 \leftarrow \mathbf{x}^1$ and repeat from step 2.

Finally, a note about Gradient Ascent and Gradient Descent. Depending on the task, you may need to go downhill instead of uphill, for instance when finding the place of minimal loss. The GD algorithm is almost exactly the same as GA. The only exception is that you have to take the step toward steepest descent, not steepest ascent. This is just the opposite direction of gradient, and hence the update step would look like

$$\mathbf{x}^1 = \mathbf{x}^0 - R \cdot \nabla f(\mathbf{x})|_{\mathbf{x}=\mathbf{x}^0} \quad (8.2.3)$$

(note the minus sign). There are two trivial ways to turn the GA problem into a GD problem or the way around:

1. flip the sign of your objective function: instead of minimizing $f(\mathbf{x})$, maximize $-f(\mathbf{x})$.
2. change the algorithm in a way to take a step into the opposite direction of the gradient. If you don't want to change your code, just use negative learning rate R .

TBD: SGD

8.3 OLS Example

Linear Regression “Least Squares” means

$$\min_{\beta} L(\beta) = \sum_i (\hat{y}_i - y_i)^2$$

where

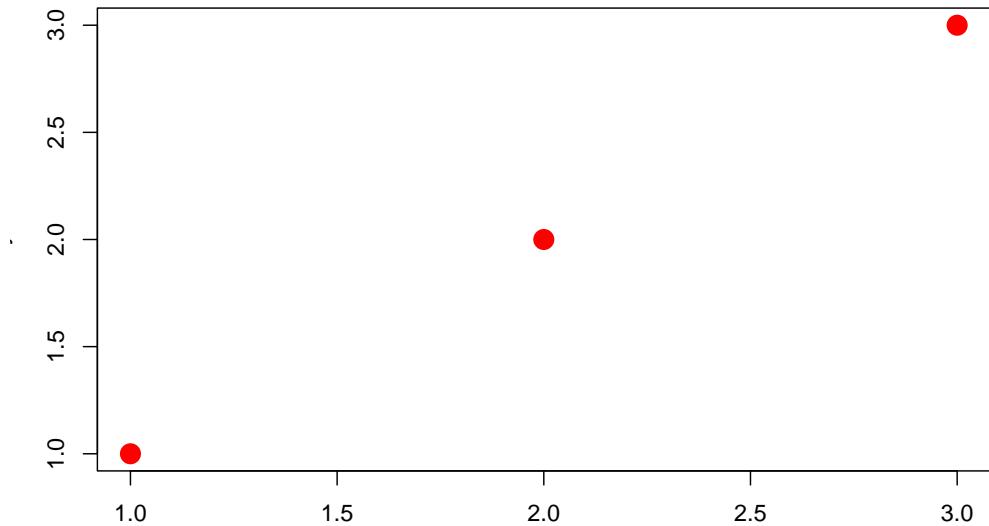
$$\hat{y}_i = \hat{\beta}' \mathbf{x}_i$$

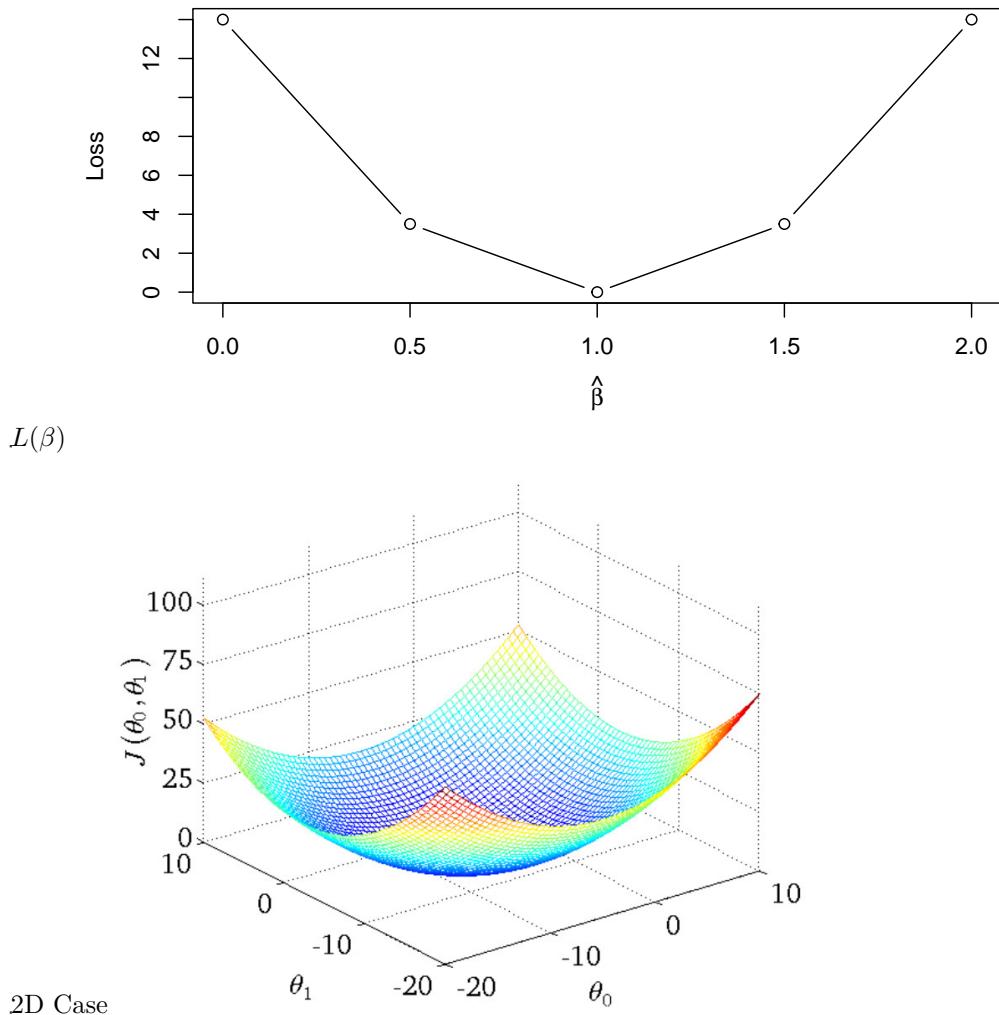
- find β that minimizes $L(\beta)$
 - L is “loss function” (objective function, cost function)
 - how?

Example: Predict September Arctic Sea Ice by March Extent
 Trial-and-Error Exercise
 (Grid Search)

- OLS model $y_i = \beta x_i + \epsilon_i$
- Use the data at right
- Find the optimal β
 - Calculate $L(0), L(0.5), L(1), L(1.5), L(2)$.
- Plot $L(\beta)$ versus β

data:





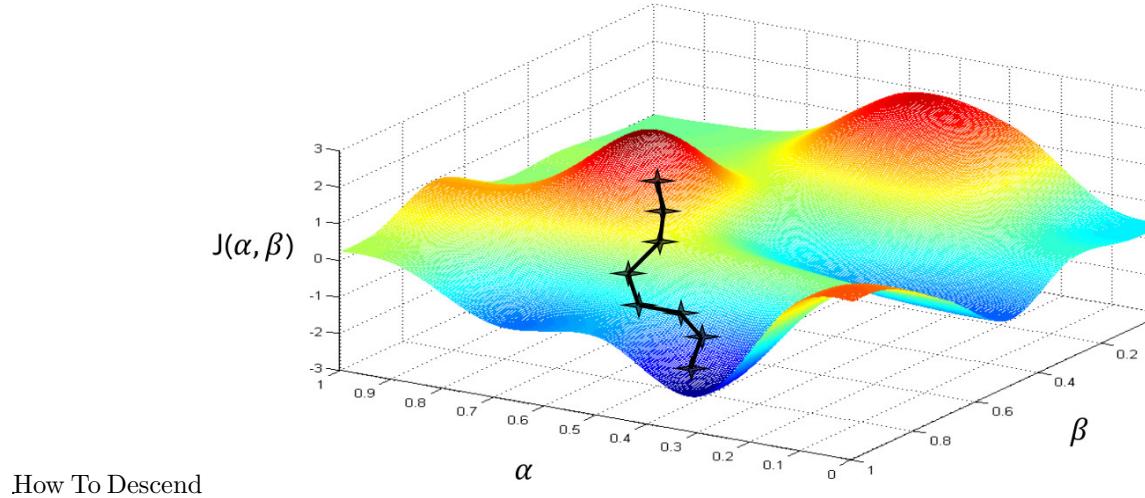
8.4 Gradient Descent

Prerequisites: Gradient (Section A.2.1).

How To Minimize $L(\beta)$?

1. Start with a β value
2. Calculate $L(\beta)$.
3. Change β so it decreases L

4. Repeat 2-4 until we are at minimum



How To Descend

8.4.1 What Is Gradient

What is Gradient Vector of first derivatives of the function with respect to it's arguments

- Direction where the function's growth is steepest
- “Speed” of growth
 - Per unit interval

Example:

$$f(\beta) = \beta^2 \Rightarrow \nabla f(\beta) = 2\beta$$

- positive if $\beta > 0$
 - $f(\cdot)$ grows when β grows
- negative if $\beta < 0$
 - $f(\cdot)$ grows when β decreases
- zero if $\beta = 0$
 - We are in a (local) optimum

Two-Dimensional Example

$$\begin{aligned} f(\beta_1, \beta_2) &= e^{-\beta_1^2 - \beta_2^2} \\ \frac{\partial f(\beta_1, \beta_2)}{\partial \beta_1} &= -2f(\beta_1, \beta_2)\beta_1 \\ \frac{\partial f(\beta_1, \beta_2)}{\partial \beta_2} &= -2f(\beta_1, \beta_2)\beta_2 \end{aligned}$$

In vector form

$$\frac{\partial f(\beta)}{\partial \beta'} = -2f(\beta)\beta$$

Linear Regression Example Example of Linear Regression:

- In non-matrix form

$$\begin{aligned} L(\beta) &= \sum_i (y_i - \beta_1 x_{1i} - \beta_2 x_{2i})^2 \\ \frac{\partial}{\partial \beta_1} L(\beta) &= 2 \sum_i (y_i - \beta_1 x_{1i} - \beta_2 x_{2i}) \cdot x_{1i} \\ \frac{\partial}{\partial \beta_2} L(\beta) &= 2 \sum_i (y_i - \beta_1 x_{1i} - \beta_2 x_{2i}) \cdot x_{2i} \end{aligned}$$

- In matrix form

$$\begin{aligned} L(\beta) &= (\mathbf{y} - \beta \mathbf{X})' (\mathbf{y} - \beta \mathbf{X}) \\ \frac{\partial}{\partial \beta} L(\beta) &= 2(\mathbf{y} - \beta \mathbf{X})' \mathbf{X} \end{aligned}$$

8.4.2 How to Optimize

How To Improve β Move in (the opposite) direction of gradient $\nabla f(\beta)$

- Gradient: in which direction the function grows most
- Climbing a snowy mountain in fog
- $-\nabla f(\beta)$: direction the function decreases most

$$\beta_{n+1} = \beta_n - R \frac{\partial}{\partial \beta'} L(\beta)$$

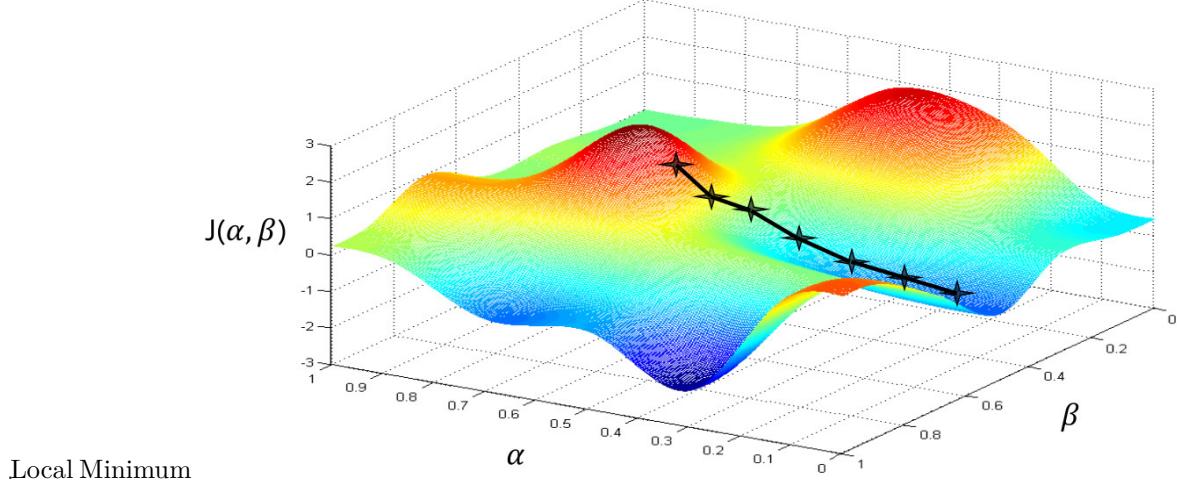
- R : step size (learning rate)
 - Should be small
 - Can be made adaptive
 - Can be calculated
- Stop when gradient close to zero ...
- or when the objective function does not decrease any more
- or when too many iterations

Algorithm

1. Set $n = 0$ and β^0 to a value

- $\beta^0 = \mathbf{0}$ is sometimes a good choice
- 2. Choose R (a small number)
- 3. Calculate $L(\beta^n)$
- 4. Calculate gradient $\nabla L(\beta^n)$
- 5. Is gradient close to $\mathbf{0}$?
 - Yes – stop
- 6. Calculate $\beta^{n+1} = \beta^n - R \cdot \nabla L(\beta)$
- 7. Calculate $L(\beta^{n+1})$
- 8. Did $L(\beta)$ decrease substantially?
 - No – stop
- 9. Is n too large?
 - Yes – stop
- 10. set $n := n + 1$, repeat 4

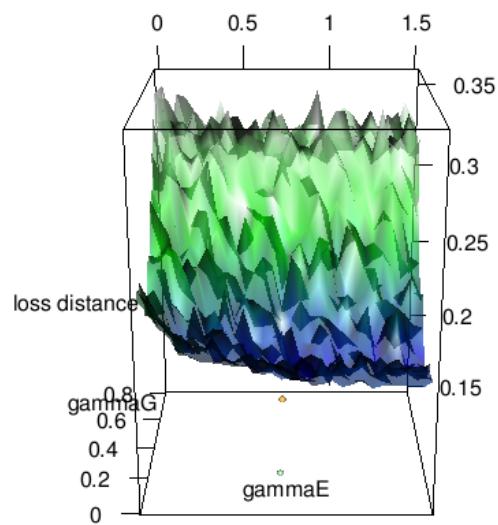
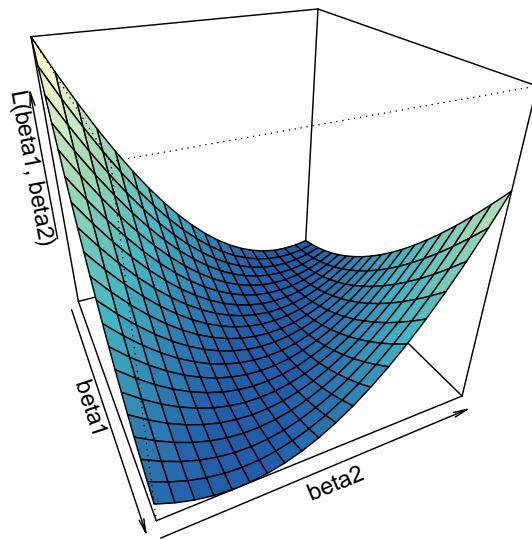
8.4.3 Problems



.Convexity Function $f(\mathbf{x})$ is convex iff:

$$\forall \mathbf{x}_1, \mathbf{x}_2, \quad t \in (0,1)$$

$$\begin{aligned} f(t\mathbf{x}_1 + (1-t)\mathbf{x}_2) &< \\ &< tf(\mathbf{x}_1) + (1-t)f(\mathbf{x}_2) \end{aligned}$$



Noisy Objective Function

8.5 Key Concepts

Key Concepts

- Cost Function (Loss Function)
- Non-Linear Optimization
- Gradient Descent
- Local/Global minima
- Convexity
- Learning Rate
- Feature Scaling

8.5.1 Resources

- Khan Academy's "Why gradient is the direction of steepest ascent" <https://www.khanacademy.org/math/multivariable-calculus/multivariable-derivatives/gradient-and-directional-derivatives/v/why-the-gradient-is-the-direction-of-steepest-ascent>

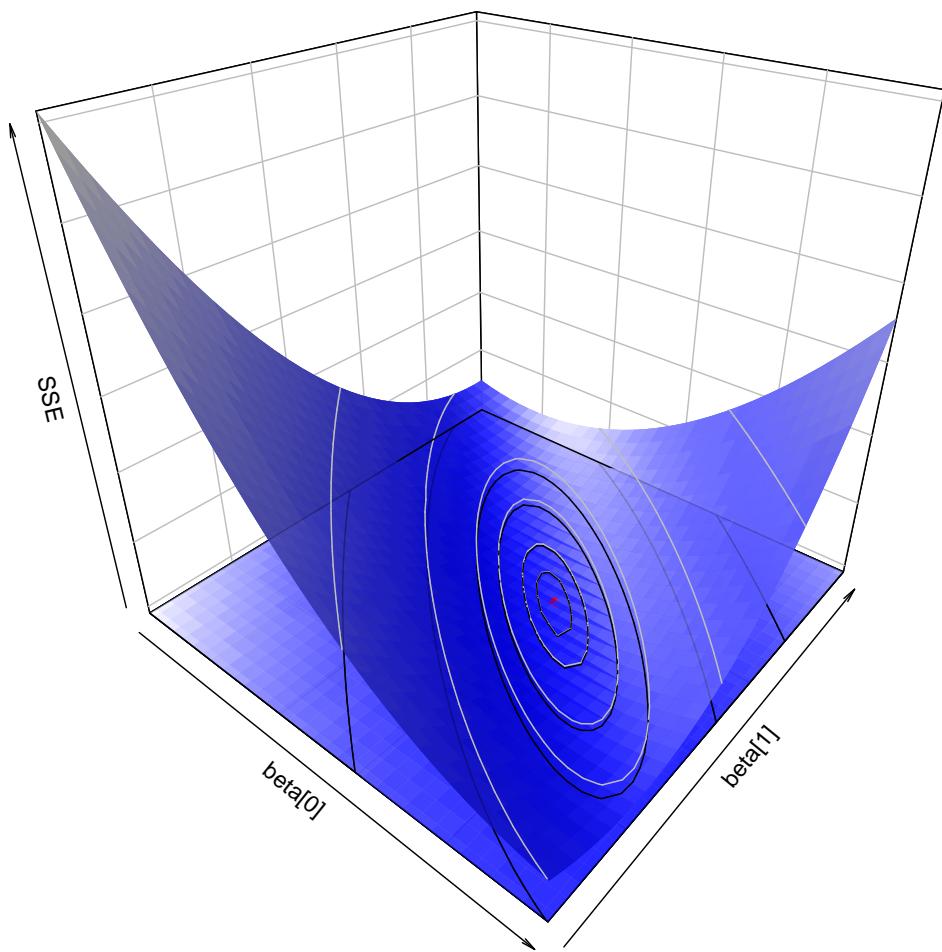


Figure 8.1: SSE as a function of β_0 and β_1 .

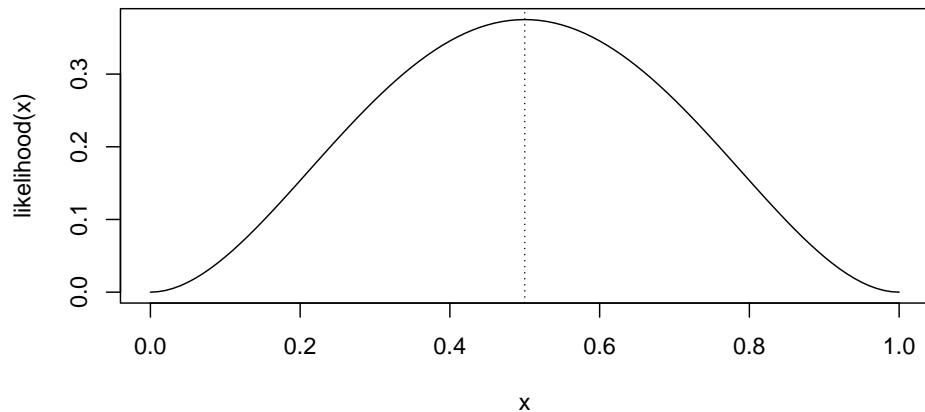


Figure 8.2: Likelihood value for the coin toss, depending on the head probability p

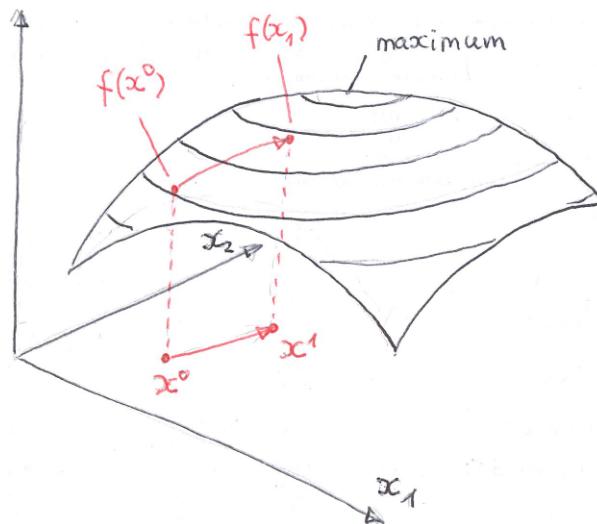


Figure 8.3: One step of Gradient Ascent. We start from an initial guess x^0 and take a step along the gradient. This moves us uphill to x^1 . The function $f(\mathbf{x})$ is depicted by the surface overlaid by (rather circular) level sets.

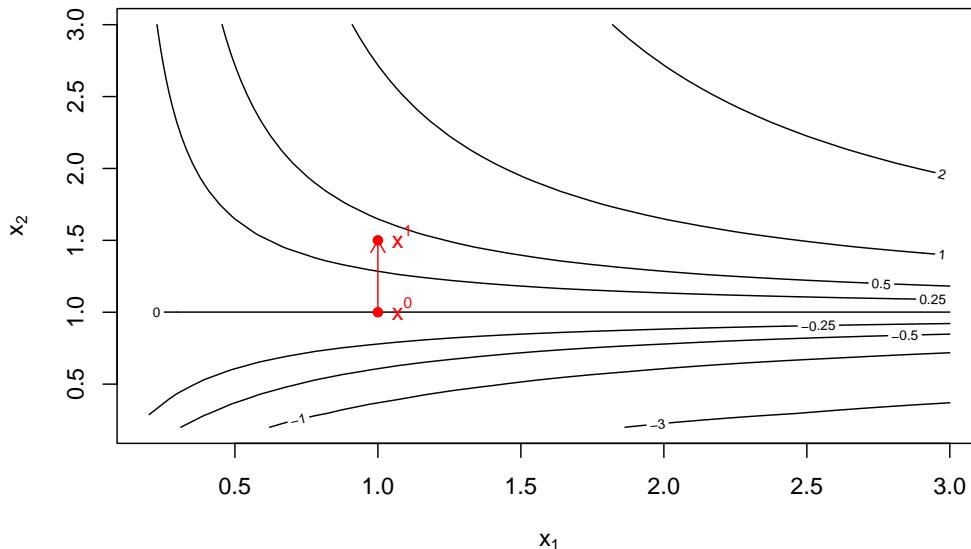


Figure 8.4: One Gradient Ascent step for function $f(\mathbf{x}) = x_1 \cdot \log x_2$. At the initial point $\mathbf{x}^0 = (1,1)'$, the gradient $(0, 1)'$ points straight up. We move in that direction by the amount (learning rate) $R = 0.5$. This leads us to $(1,1.5)'$, our next approximation for the maximum.

8.6 Feature Selection and Regularization

8.6.1 Feature Selection

In applied analysis it is quite common to have datasets with a large number of features. Often we have little knowledge about the importance of many of these. Many of these may be highly correlated and many can show certain importance in our models. For instance, when using a large international survey data, such as World Value Survey, one may find the variable *country* is highly correlated with *domestic language*, language used to fill out the survey, and the id of the team member. It may also be somewhat unexpectedly related to certain lifestyle questions, e.g. there are probably very few affirmative answers to the question “do you have a boat” in an arid landlocked area. Such closely correlated variables may cause various problems with data modeling. To name a few

- Large and unstable parameter values and large standard errors. This is mainly a problem for inferential modeling and may obscures the interesting effects that are in fact there.
- Overfitting. This is how the same issue manifests in predictive modeling.
- Model does not converge, or converges into a sub-optimal solution. While linear regression (almost) always works well, more complex model are much more demanding in terms of data properties. Even if data looks good globally, we may run into a trouble in a region where the correlation is high.

In case of a smaller well-documented dataset, it may be possible to manually select the interesting features. But this approach does not scale to larger datasets where we have thousands of similar variables we do not understand well. For instance, imagine analyzing urban movements using millions of cellphone calls, or doing sports analytics with thousands of datapoints about athletes’ movements. In such cases it is not obvious what to include or exclude in the model.

Feature selection is a method (more like a set of several methods) that helps to include only the “best” features in the model. It is in some ways similar to *regularization*, a method that does not directly select features but manipulates the model parameters and may achieve a comparable effect.

Consider the following example. We generate one variable $\mathbf{x} \sim N(0,1)$ and form a number of other highly correlated variables:

$$\mathbf{x} \sim N(0,1)$$

$$\mathbf{x}_1 = \mathbf{x} + \boldsymbol{\epsilon}$$

$$\mathbf{x}_2 = \mathbf{x} + \boldsymbol{\epsilon}$$

...

where $\boldsymbol{\epsilon} \sim_{i.i.d} N(0, 0.1)$. We generate the outcome y as $y = \mathbf{x} + \mathbf{u}$, where $\mathbf{u} \sim N(0, 0.3)$. Thereafter we estimate linear regression model in the form

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + e_i$$

i.e. we model \mathbf{y} using a number of highly correlated variables. Importantly, we keep the number of cases very low, the example below is made for $N = 12$ training data observations and 7 different \mathbf{x} vectors.

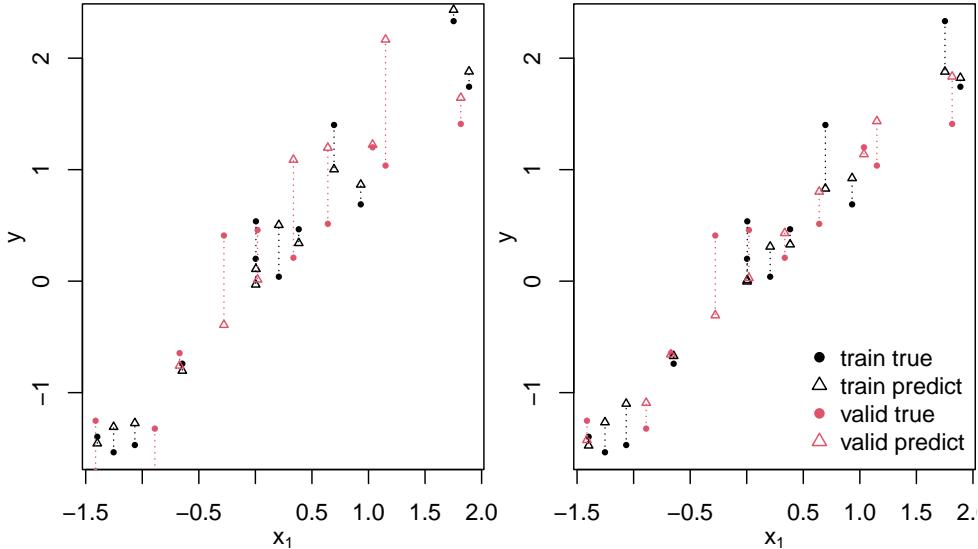


Figure 8.5: Linear regression (left panel) versus ridge regression (right panel). Solid dots represent data and empty triangles are predictions. Black is training and red testing data. Linear regression make much more noisy predictions than regularized ridge regression. There are 6 other highly correlated features not visible in this figure.

Figure 8.5 shows an artificial example with 7 highly correlated predictors. Solid black are the training data and red are validation data, solid dots represent training and empty triangles testing data, and the dotted lines between those are residual errors. The left figure depicts the linear regression results. One can see that errors in training data are mostly small but for testing data the errors are much larger, in particular for data points that are far from training observations. The fact that the model behaves much better on training than on testing data is also confirmed by the corresponding R^2 values, 0.957 and 0.541 respectively.

The right panel show results for a penalized ridge regression. Ridge suppresses the noise from highly correlated variables and correctly finds that all predictors contain essentially the same information. Hence the predictions (triangles) form almost perfect line on the figure. The corresponding R^2 values are 0.934 and 0.853. The flexibility in the linear model largely captured the noise, making the model less flexible forced it to focus on the signal instead.

8.6.2 Regularization

More complex models we use on large datasets are often very flexible. This flexibility may easily lead to enormous overfitting, and technical problems, such as lack of

convergence.

Regularization is a simple way to force flexible models to be less flexible in order to improve their performance on unseen data. It can be done in various ways like

- by adding a penalty term to the objective function
- by using a Bayesian prior over the parameter values
- by terminating iterative optimization (such as stochastic gradient descent) early

One can show that under certain assumptions, all these methods produce similar results.

Chapter 9

Unsupervised Learning

Contents

9.1	Introduction	357
9.2	Cluster Analysis	358
9.2.1	Idea	359
9.2.2	Cluster Analysis More Generally	360
9.2.3	k -Means Clustering	362
9.2.4	Hierarchical clustering	366
9.2.5	Discriminant analysis	366
9.3	Principal Component Analysis	370
9.3.1	Motivation	370
9.3.2	Principal Components: The Idea	372
9.3.3	Explained Variance	373
9.3.4	Data Rotation	376
9.3.5	Principal Component Regression	379
9.4	Comparison of Clustering and PCA	383

9.1 Introduction

In the previous sections we were working with supervised learning. In case of supervised learning, our task is to predict outcome y based on features \boldsymbol{x} . As we have labeled training data, we can tell the algorithm for each case how “far off” was the prediction from the true value. Later we use the trained model to do similar predictions on unlabeled data. As we have analyzed the prediction errors on training data, we have some confidence to assume the errors are similar on unknown data. Formally, our task is to estimate the function $f : X \rightarrow Y$ where X is the feature space and Y is the target space. A good result is such a function where the predicted value $\hat{y} = f(x_i)$ is close to the true value y_i .

Unsupervised learning is the case where we don't know the "correct" labels y_i and hence we cannot tell if our predictions are close or not to the true values. Even more, there is often no such things as true labels, cases can be categorized in many different ways and all of these may be correct in some sense.

Instead of trying to predict "correct" values that we do not know or that may even not exist, unsupervised learning is used to discover and exploit various traits in data. The common examples include:

- Cluster analysis: we group data into clusters, groups of cases that are reasonably similar to each other while being different from cases in other clusters. A small number of such clusters can thereafter be used as data simplification. We may use a single "representative" in each cluster instead of individual values and in this way to tremendously reduce the complexity of data.

For instance, the consumers can be categorized into a small number of clusters, and afterwards one may design a different marketing strategy for each cluster. It may be infeasible to have a large number of such strategies but unsupervised learning helps us to reduce the complexity to a manageable number.

- Principal component analysis: we analyze which kind of values tend to occur together in the data. This allows us to find certain combination of features, principal components, that carry most of the information. The principal components may give us novel insight into the problem, but it also allows to remove the less important traits and simplify the data in this way.
- Market basket analysis: as in PCA, we attempt to find values that tend to occur together. However, our task will be to construct claims like "consumers who bought x usually also buy y ".
- Also usually not considered as unsupervised learning, various descriptive graphs and tables play a similar role. They help us to discover the traits in the data, their limits, and structure.

9.2 Cluster Analysis

Prerequisites: Metric distance, vector norm: [Section 2.2.2 Norm and Distance](#), page 99

Cluster analysis is one of the most widely used unsupervised learning. It is a way to partition data points into a number of groups, "clusters". We want the data in cluster to be similar in some sense while data in different clusters may differ. This typically serves as a tool for simplifying and understanding data, e.g. for designing a small number of manageable strategies, or just for understanding the main traits and processes we are encountering.

First we discuss the basic idea of cluster analysis and thereafter introduce perhaps the most popular clustering algorithm, k -means.

9.2.1 Idea

Cluster analysis attempts to find natural groupings in the data. As it is an unsupervised learning method, we do not normally provide it with any pre-defined grouping information. All this will be derived directly from data. This also applies to the number of clusters—we normally do not know what is the correct number. Even more, there may not be anything like the “correct number”, one can look at the data in different ways, just some of the approaches may be more insightful regarding the current problem. The key of deciding which observation goes to which cluster is similarity—observations in the same cluster should be more similar than observations in different clusters. There are many ways of deciding which cases are more similar, these are associated with different clustering methods.

Figure 9.1 displays an example 2-D dataset (left panel) and the result of the corresponding cluster analysis (right panel). This dataset is exceptionally well suited for cluster analysis, and we can immediately see that five clusters are just a right choice, with distance within clusters being very small compared to inter-cluster distance. Such luxury is usually not there, first data may not contain such well-defined clusters; and second, in higher dimensions one cannot visualize the data space in a similar fashion. We have to rely on mathematical methods when trying to analyze the value of the particular clusters.

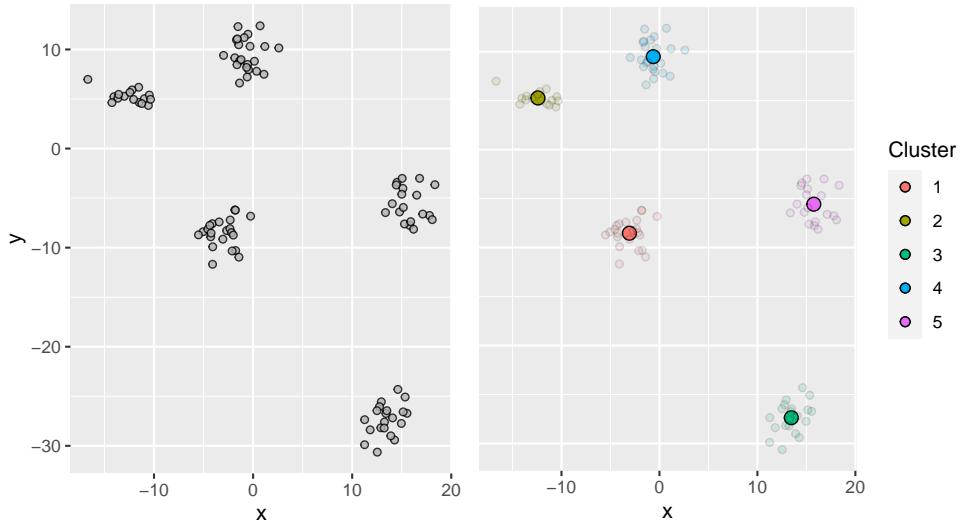


Figure 9.1: Original data (left) and cluster centers (right). This artificial dataset contains five clearly separated groups and hence it is exceptionally well suited for cluster analysis. The left panel shows the original data, the right panel the five cluster centers as computed by k-means. Colors denote which cluster each data point belongs to. Note that the ordering of clusters is different.

Clustering has many practical applications. Here are a few examples:

- Market segmentation. There are thousands of customers with each of their different interests, but marketing department cannot handle thousands of different strategies. We want to apply a small number of strategies only, and hence we need to partition customers into a small number of groups, “clusters”.
- Land use analysis in for urban planning. We want to compare land use of different cities, but the land is split into a tremendous number smaller lots, parks, streets, industrial areas, and so on. We may want to group cities into a small number of different “types” based on the land use.
- Police targeting crime: different neighborhoods may have different crime patterns, and hence police should target those neighborhoods differently. Instead of designing policing guidelines for every single neighborhood, we can cluster neighborhoods into a small number of “groups” and design different policing methods for these groups.
- Medical diagnosis. Different patients show a very large variety of symptoms. We may want to combine those into a small number of “types”, and for each type have further rules how to either treat those, or maybe do further diagnostics.

Sometimes we are interested in *hierarchical clustering*, i.e. not just clusters of data points but also of clusters of clusters.

All these examples are in some sense about simplifying and compressing data. Instead of looking at thousands of different individual cases, we replace this unfathomable diversity with a small number of different options (clusters). This may help to design a manageable number of strategies, or serve as a simplification for understanding the problem.

9.2.2 Cluster Analysis More Generally

In order to split data into clusters we need four things: suitable data, distance metric, loss function, and an algorithm that can actually compute the clusters.

First, we obviously need data. In the example below we imagine data as points on the 2-D x - y -plane, but in general these are in high-dimensional space and cannot be easily visualized. Normally we imagine data in a form of a numeric design matrix but in certain cases it may also be in a different format. For instance, one can compute string distance between words, and in that case the data may be in the form of character strings.

Second, in order to measure similarity, we need something like distance metric (see [Section 2.2.2 Metric distance](#), page 102). If the design matrix is numeric, we may rely on Euclidean or other L_p type metrics, but we may also carve out our own dedicated metric. For instance, when comparing portraits, we may want to design a metric that only looks at the faces and ignores the background. In case of more than a single feature, we also have to weight the features somehow, i.e. feature scaling matters (see [Data-Driven Metrics](#) in [Section 6.1.1](#)).

Third, we need a way to decide if a particular data point is a good fit for one or another cluster. We may do this by defining a per-cluster loss function $L(C)$ where C is the cluster, a set of data points that belong to it. The loss function typically penalizes intra-cluster distance (dissimilarity) as we would prefer all member points to be close to each other (similar to each other).

Loss function shows “badness” of a particular cluster. A large loss value means the cluster is bad. See [Section 8.1 Loss Function and Non-Linear Optimization](#), page 337.

Finally, we also want to compute the set of clusters, not just the “badness” (loss) of the result. Hence we also need an algorithm that can find a good set of clusters based on data, distance metric, and loss we selected above. The algorithm should consider different ways to put data into clusters C_1, C_2, \dots and pick such an arrangement that produces minimal loss (it should minimize the loss function). Ideally it should find the smallest possible loss but if this is not feasible, a good enough solution may do. The algorithm should return the partition—which observations go to which cluster. Formally:

$$\{C_1, C_2, \dots, C_K\} = \arg \min_{C_1, C_2, \dots, C_K} \sum_{k=1}^K L(C_k).$$

Unfortunately, in typical problems there are way too many possibilities how to partition data into clusters, so it is in general not possible to find the best way. But there are many algorithms that work well enough.

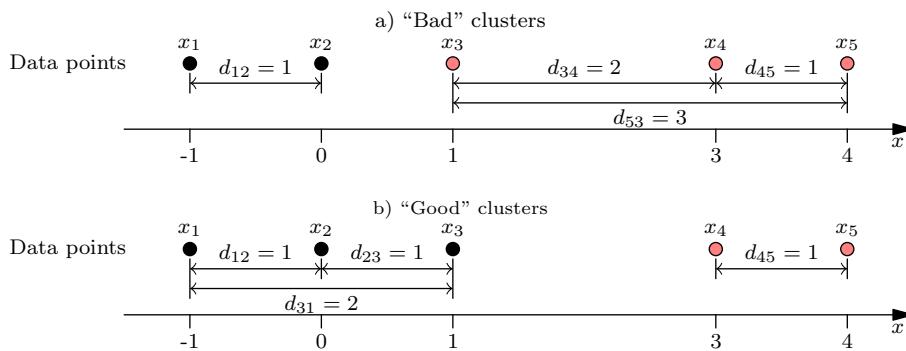


Figure 9.2: Two ways to partition data x_1, x_2, \dots, x_5 into clusters. The upper panel assigns x_3 to the red cluster, and hence the intra-cluster distances in the red cluster are 1, 2, and 3. This is more than in the good case (lower panel) where the black cluster contains intra-cluster distances 1, 1 and 2. A simple loss function that just adds the intra-cluster distances would prefer the good way over the bad way.

Figure 9.2 demonstrates two ways to partition five data points x_1, x_2, \dots, x_5 into two clusters in an 1-D case. These five dots are the data points. In 1-D case we can measure similarity as Euclidean distance, just as the (absolute value of the) difference between the data points. Third, we can compare the two clusters using a loss function. We can pick a loss function for cluster C as sum of intra-cluster squared distances:

$$L(C) = \sum_{i,j \in C} d_{ij}^2 = \sum_{i,j \in C} |x_i - x_j|^2 \quad (9.2.1)$$

(d_{ij} is just distance between data points i and j). Now let's compute the black and

red loss when partitioned in the “bad” way:

$$\begin{aligned} L(C_{black}) &= \sum_{i,j \in \{1,2\}} d_{ij}^2 = d_{12}^2 + d_{21}^2 = 1^2 + 1^2 = 2 \\ L(C_{red}) &= \sum_{i,j \in \{3,4,5\}} d_{ij}^2 = d_{34}^2 + d_{43}^2 + d_{45}^2 + d_{54}^2 + d_{53}^2 + d_{35}^2 = \\ &= 2^2 + 2^2 + 1^2 + 1^2 + 3^2 + 3^2 = 28 \end{aligned} \quad (9.2.2)$$

(we defined the loss function in a way that we add both distance from i to j and from j to i , but we can drop one of these). We can conclude that the black cluster is pretty good (loss is 2) but the red cluster is worse (loss 28). The overall loss, $L(\mathbf{C}) = L(C_{black}) + L(C_{red})$, is 30. However, in the “good” case we have

$$\begin{aligned} L(C_{black}) &= \sum_{i,j \in \{1,2,3\}} d_{ij}^2 = d_{12}^2 + d_{21}^2 + d_{23}^2 + d_{32}^2 + d_{31}^2 + d_{13}^2 = \\ &= 1^2 + 1^2 + 1^2 + 1^2 + 2^2 + 2^2 = 12 \\ L(C_{red}) &= \sum_{i,j \in \{4,5\}} d_{ij}^2 = d_{45}^2 + d_{54}^2 = \\ &= 1^2 + 1^2 = 2. \end{aligned} \quad (9.2.3)$$

Now the red cluster is pretty good, and the black one is worse. But black cluster deteriorated less than the red cluster gained, and hence the overall loss improved from 30 to 14. It paid off to re-assign x_3 from the red to the black cluster. The “good” partition results in smaller overall loss, and is accordingly a better way to split this data into clusters.

This example, to pick a loss function that computes sum of intra-cluster squared distances, is just one possible way to define clusters (this is the loss function that the popular k -means algorithm is based on). But there are very different ways to define clusters, e.g. based on maximum distance within the cluster. In the 1-D case we analyzed, the distance metric does not play a role, but in more complex cases we always have to decide how to measure distance, and potentially we need experiment with different metrics to find the one that is best suited for the particular task.

9.2.3 k -Means Clustering

k -means is one of the simplest and most popular clustering algorithms. It is intuitive, fast, and always provides a solution, although the solution may sometimes be suboptimal. It is based on intra-cluster distance, similar to the example on Figure 9.2.

Now let’s take N -dimensional data points $\mathbf{x}_i, i \in \{1, \dots, N\}$. Instead of just summing the squared distances as we did in the example above, we now compute average of the squared distance as the loss function for cluster C :

$$L(C) = \frac{1}{\|C\|} \sum_{i', i \in C} d_{ij}^2 = \frac{1}{\|C\|} \sum_{i', i \in C} (\mathbf{x}_i - \mathbf{x}_{i'})' (\mathbf{x}_i - \mathbf{x}_{i'})$$

where $\|C\|$ is number of observations in the cluster (see Section 0.1, Sets). Hence the total loss

$$\min_{C_1, C_2, \dots, C_k} \sum_{i=1}^k \frac{1}{\|C_i\|} \sum_{j', j \in C_i} (\mathbf{x}_j - \mathbf{x}_{j'})' (\mathbf{x}_j - \mathbf{x}_{j'})$$

k -means partitions data into pre-specified k clusters where cluster membership is mutually exclusive—each data point belongs to one and only one cluster. The cluster membership is determined based on the distance between the data point and cluster centers, and the algorithm repeatedly re-assigns the observations to the closest cluster, and thereafter re-computes the cluster centers. These two steps are computed repeatedly until it results in a stable partition—each observation belongs to its closest cluster. It may sound somewhat surprising that such a simple idea works very well, but in most cases it does.

Next, we explain the algorithm in more detail and provide an example.

The k -means algorithm

1. Select the desired number of clusters, k . This must be decided before the algorithm starts.
2. Next, we need to find a *centroid*¹ for each of the k clusters. We can do this in various ways, for instance we can pick a random data point as the centroid for each of the clusters (just pay attention to that each cluster should get a *different* data vector as its centroid).
3. Now assign each actual data point to the cluster with the closest centroid. This immediately causes the cluster partition to clear up as more similar observations tend to fall into the same cluster. As a result we now know for each observation which cluster does it belong to.

Note that “closest” assumes we have decided for a distance metric, in case of k -means we normally use Euclidean distance.

4. Now we compute new cluster centroids by just averaging the data vector components for each cluster.
5. And now we just repeat from 3 until the partition converges, i.e. there are no more changes in the partition $\{C_1, C_2, \dots, C_K\}$.

The algorithm works surprisingly well and always produces a result. Figure 9.3 illustrates how the algorithm works in a simple case. However, sometimes the result may be suboptimal, so it is advisable to run the k -means algorithm several times with different random starting points.

TBD: k-means gets stuck, perhaps as an exercise

¹ Centroid is similar to average or mean value, just in case of multi-dimensional objects (like data vectors) we call the average “centroid”. You can easily visualize it as the “middle point” of a point cloud.

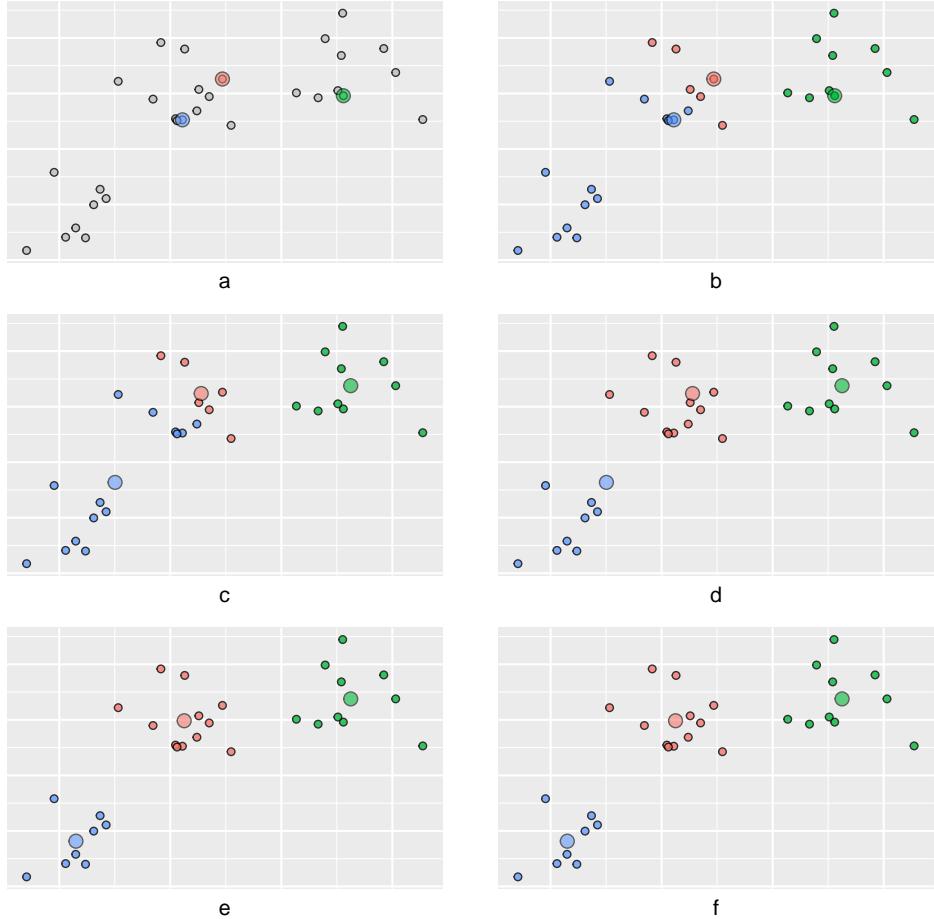


Figure 9.3: k -means algorithm at work. a) Top-left panel shows the plain data with no cluster information, however, the random data points are picked as cluster centers (denoted by color). b) Top-right panel has all data points colored according to the closest random cluster. Already at this stage the k -means algorithm starts to separate data into different clusters. c) Mid-left panel shows the updated cluster centers: based on the previous image, the new cluster centers are centroids of the points that belong to the same cluster (same color). d) We update clusters (colors) again based on the closest cluster center. Now all of the middle blob belongs to a single cluster. e) One more update of cluster centers will position the red and blue cluster center in the middle of the respective clusters. f) Next update of clusters (colors) does not change anything. The process has converged.

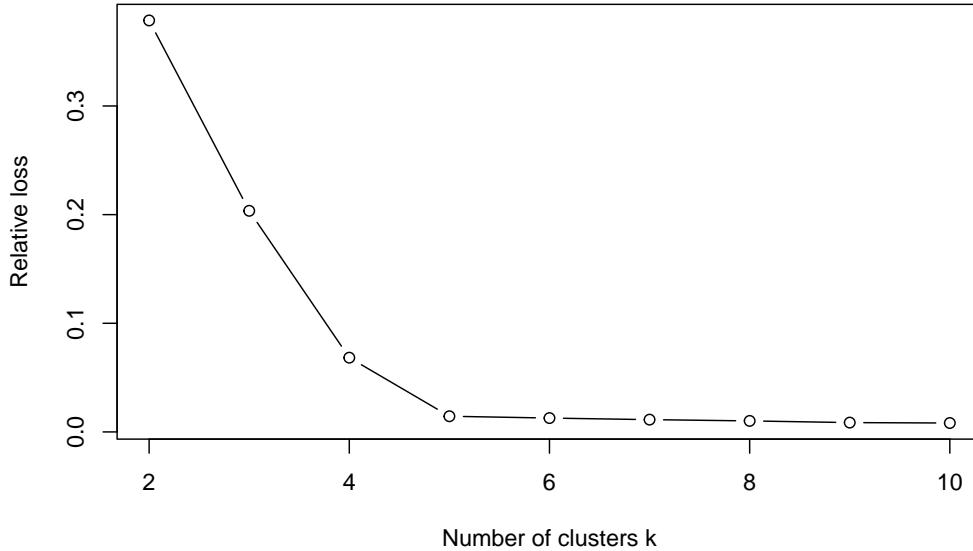


Figure 9.4: Elbow plot for the same data as used in Figure 9.1. The vertical axis denotes the relative loss, within-cluster loss as a percentage of total loss. One can see that at $k = 5$, the relative loss reaches essentially zero. This is the “elbow” of the plot and the best number of the clusters.

Determining the number of clusters

k -means expects the user to provide the required number of clusters. This is easy in case we know enough about the underlying data structure, but sometimes we need more guidance from the algorithm itself. A popular way to find the “best” number of clusters is by using *elbow plot*. The idea of the elbow plot is the following: we allocate the data points to clusters by minimizing the sum of squared errors (or another loss function) within the clusters. In case we choose too few clusters, we have many mis-allocated points and hence the loss is large. But as soon as we pick the correct number of clusters, the loss should fall substantially. Increasing k even further will not substantially change the loss. So one expects to see a kink, the “elbow” on the plot at the correct value of k .

Figure 9.4 displays such a clear kink at $k = 5$. This is the same data as depicted on Figure 9.1. That artificial dataset is extremely well suited for cluster analysis. However, when we move to typical real datasets, the kinks may be much more vague, or completely missing. Figure 9.5 depicts a similar elbow plot for diamonds data. As the data (left panel) do not display any clear cluster structure, the relative loss on the elbow plot (right panel) keeps getting smaller even when we add clusters. The apparent kink at $k = 3$ does not correspond to any clearly distinct clusters (figures on the left panel). The clusters are probably not a useful way to think about this data.

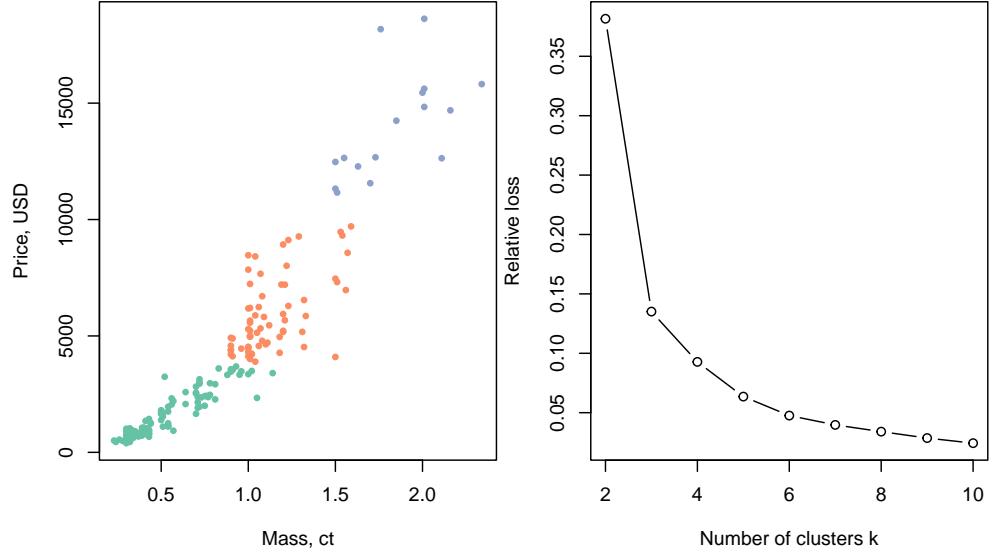


Figure 9.5: Elbow plot for diamonds data. The left panel depicts the diamonds data, the data points are colored according to the detected cluster id. The right panel shows the elbow plot, if anything it suggests $k = 3$ is the optimal number of clusters. However, as the left figure indicates, the detected clusters are rather indistinct. This data is not well suited for clustering.

9.2.4 Hierarchical clustering

k -means was an example of “top-down” clustering, where we started by splitting all data points into a given number of clusters. Hierarchical clustering contains methods that work from “bottom-up”, by connecting individual observations into pairs, and further into larger clusters. Such bottom-up methods are called *agglomerative clustering*, because they proceed by lumping more and more observations together into larger and larger clusters. The concept *hierarchical*, in turn, refers to the fact that the common agglomerative methods produce cluster hierarchy, smaller clusters inside larger clusters.

Next, we demonstrate hierarchical clustering using [iris data](#) (Figure 9.7). The left panel shows a small subset of 10 observations. In order to visualize the results easily, we only use two features: sepal length and sepal width.

TBD: refer to iris data, write about sepals, petals

9.2.5 Discriminant analysis

The clustering algorithms we discussed above are based on proximity—some kind of distance between the data point and the other points in the dataset. Discriminant analysis takes quite a different point of departure: it builds certain mathematical models for different clusters, and afterwards checks which cluster does the particular datapoint resemble. However, in practice the difference may not be large, in any

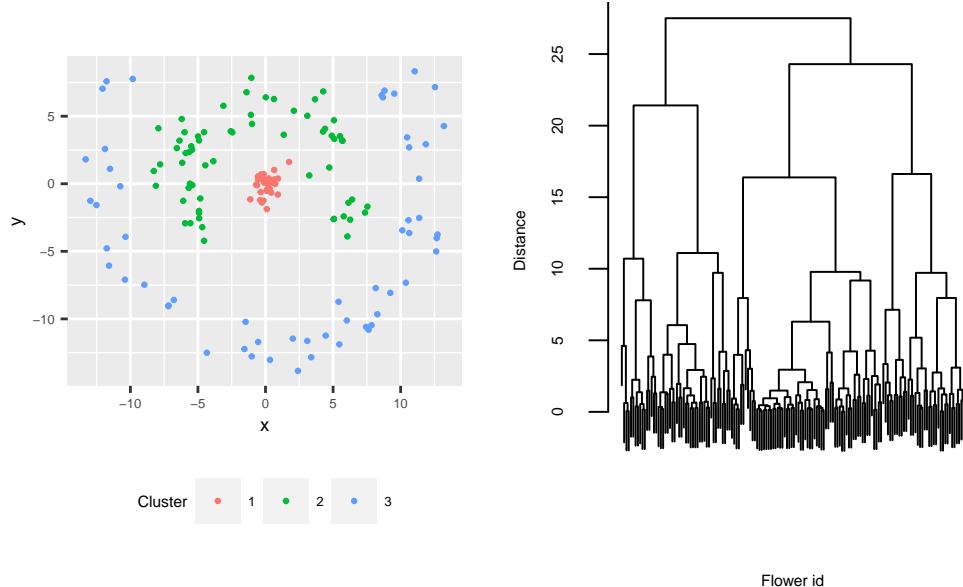


Figure 9.6: Sample data of concentric curved clusters.

case the datapoints that are close in the feature space tend to be placed in the same cluster. Below we discuss a specific method, *gaussian mixture models*, and how to use it for discriminant analysis.

Mixture models

Mixture models assume that the groups in data follow certain distributions, e.g. normal in case of gaussian mixtures.

Figure 9.8, left panel, shows the height histogram for 256 !Kung San adults, 136 females (red) and 120 males (blue). Typical males are noticeably taller than females, with the corresponding average heights being 160.9 and 150 cm. Individually, both genders follow roughly a normal distribution, but as the distributions do not overlap well, we see a resulting distribution (denoted by black bars on the Figure) that looks much more flat-topped than the normal curve.

TBD: Explain data

The right panel attempts to guess how do male and female distributions look when separated. The model is based on *gaussian mixture*, i.e. it is assumed that both sexes follow a normal distribution. The results are the red and blue normal curves, and the black overall density curve, the mixture of both male and female curves. The model estimates that the mean height of males is 158.3 with standard deviation 6.4, and the corresponding values for females are 147 and 3.7. For comparison, the corresponding male and female sample values are 160.9 and 6.1 for males, and 150 and 4.9 for females. As we can see, the two-component mixture identifies values rather well.

This is the idea of the mixture model: the population contains two groups with

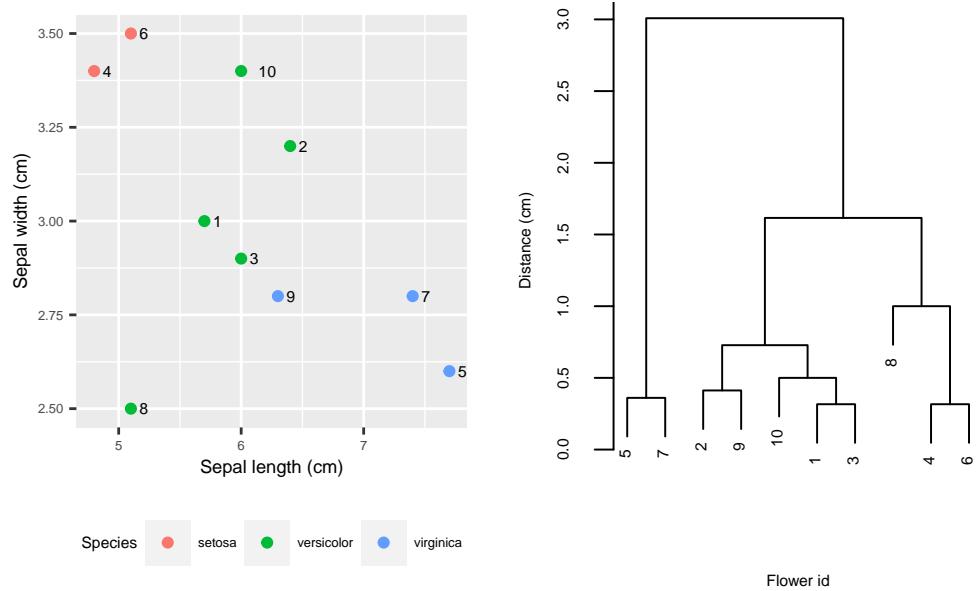


Figure 9.7: Sepal length and width of a sample of 10 iris flowers (left) and the corresponding dendrogram (right).

differently distributed values, but if we do not know which group a particular individual belongs to, we are left of “mixture” of both distributions, the black flat-topped curve. In case of discriminant analysis, the task is to find the correct group based on other data, here to tell gender based on height.

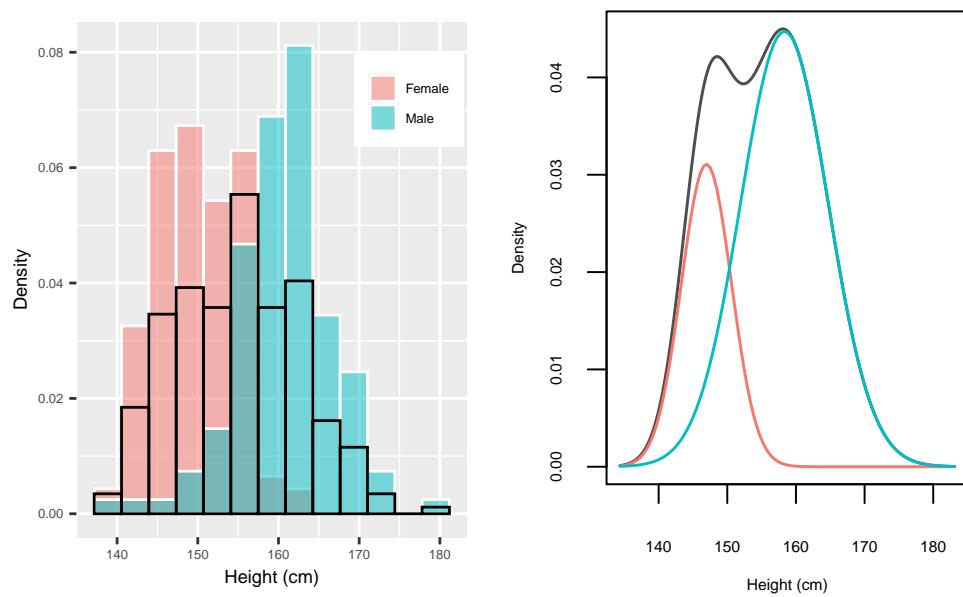


Figure 9.8: Left panel: histogram of height of 136 !Kung San women (red) and 120 men (blue). The black bars denote the overall density values. We can see that the overall height distribution is slightly bimodal.

Right panel: the corresponding estimated mixture density, consisting of two Gaussian components.

9.3 Principal Component Analysis

Principal Component Analysis (PCA) is another popular method to find patterns in data. It is in some way similar to cluster analysis, but unlike the latter, PCA is not concerned about points located close in space, but rather about points placed near hyperplanes in hyperspace. It is used in a wide variety of applications, including exploring and analyzing correlated features, designing new features, and compressing data.

9.3.1 Motivation

There are several motivations that lead to more-or less similar concept of PCA. We list here three problems, the first more a social science problem, and two other more technical.

1. How to measure vague concepts? If we are interested in measures like income or age, the measurement is easy. Pretty much everyone knows their age and people have fairly good understanding what income is (even though they may not know their exact income, or may be unwilling to tell). But how liberal are you? Or how extrovert are you? The respondents may have an idea in both cases and may be willing to answer that they are “rather not liberal” or “fairly extrovert”. But now we are measuring their idea about their liberalism (called *perceived liberalism*) and not how liberal they actually are.

To overcome this problem, the surveys typically ask for a number of questions that we think are closely related to liberalism, instead of asking about liberalism directly. For instance, one may ask (with answers on e.g. 5-point Likert scale):

1. Do you support gun rights?
2. Do you support free abortion?
3. Should the government take more care of the environment?
4. Is cross-border crime the most serious threat nowadays?

As people traditionally understand the concept “liberalism”, we may want to add the second and third answer with a positive weight, and the first and the fourth answer with a negative weight. But is this the correct approach? And what should the weights be? Is “liberalism” even a useful concept in these data, if the answers to these questions look almost random? Maybe we do not really have liberals and conservatives in the first place?

2. How do we aggregate similar measures? The second motivation is quite similar, just originate from the technical world. Imagine you have a number of similar measure—similar, but not precisely the same. For instance, you are working with natural disaster data and you want an estimate of the destructiveness of hurricanes. But FEMA² does not provide a single number of “destructiveness”. Instead, it lists a number of different costs:

- Total Individual Assistance (IA) - Applications Approved
- Total Individual and Households Program - Dollars Approved

²The U.S. Federal Emergency Management Agency

- Total Housing Assistance - Dollars Approved
- Total Other Needs Assistance - Dollars Approved
- Total Public Assistance Grants - Dollars Obligated
- Dollars obligated to emergency work
- Dollars obligated to permanent work

These numbers are clearly related to the destruction—more destruction probably means all these numbers are larger. But if I need a single number then which one should I take? Or should I take the average? But does average over number of applications, dollars approved and dollars obligated even make any sense?

3. How to get rid of a lot of redundant data Finally, there is a data compression problem. Imagine you are collecting cellphone data over time and geographic districts. For each district and each time period you record

- Total number of outgoing phone calls
- Total number of incoming phone calls
- Total number of text messages
- Total number of multimedia messages
- Total number of data connections
- Total seconds of outgoing phone calls
- Total GB of data transfer
- ...

Obviously, all these numbers are highly correlated. A busy afternoon in a large city has all these figure up in millions while there is hardly anything in a tiny rural place in the middle of night. Do we really have to store and analyze all these numbers? Can we only keep one of these and drop the others? But which one? Or should we take average again? But does average over counts, seconds and GB-s even make sense?

All of these tasks are different sides of the same problem: we have a lot of correlated data and we are looking ways to simplify and understand it. While in social sciences the understanding-part has been traditionally in the focus, in technical fields it is more often simplification that we are looking for. But in all cases we are looking for fewer dimensions: in the first example we want to reduce four answers into a single liberalism measure, in the second example we try to reduce seven different cost measures to a “destructiveness” measure, and in the final example we may want to come up with 1-2 numbers that capture the “cellphone activity”.

There are many applications where one may want to collapse a large number of dimensions into a smaller more manageable numbers:

- Genome data: there is a tremendous numbers of parameters to measure genes
- Document and image classification: the algorithm may come up with hundreds of different categories, and we are just interested in a handful
- Product recommendation: as above, we may only be interested in a small number of product categories, not in thousands.

In a similar fashion, there are numerous reasons why we want fewer dimensions:

- To avoid curse of dimensionality: algorithms that work well on a small number of dimensions may get sluggish or fail completely as dimensionality grows. Even

models that can cope with high dimensionality may display unfavorable results, such as large standard errors and low power of statistical tests.

- In predictive modeling, overfitting becomes a more and more important problem as we include more and more features. Hence we want to keep the number of features in check and include only the most relevant ones.
- Visualization: it is hard to visualize anything with more than three dimensions.
- The same is true for interpretation—high-dimensional cases are hard to interpret.
- From computational perspective, we may prefer to keep fewer features to lower the memory and storage needs. It is effectively a way of data compression.

9.3.2 Principal Components: The Idea

The idea behind PCA is describing data as some sort of elongated cloud of points. The task is to find the axes along which data is elongated, and either interpret those, or rotate those in a way that they align with the coordinate axes.

The easiest way to get an intuitive understanding of this is to use 2-D highly correlated data. Figure 9.9 below shows a such synthetic dataset. In these data, x and y are highly correlated, one can imagine the image as the dots lying on the 30° line, with a small perturbations that push them little bit off that line (left panel). The right panel adds the principal components: these are the yellow and the purple tilted axes.

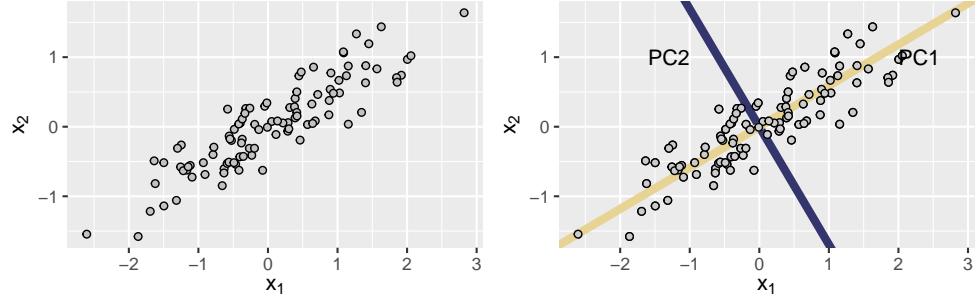


Figure 9.9: Highly correlated data that is perfectly suited for PCA. The left panel depicts only the datapoints, the right panel adds the principal components (PC1 is gold and PC2 is purple).

Already a quick look at the figure suggests that the red axis is much more important than the blue one—after all, the data is elongated along the red axis, the spread along the blue one is much smaller. This is indeed the case, and it is customary to order the axes (principal components) according to their importance. So further below we refer to the yellow axis as the *first principal component* (PC1) and the purple one

as the *second principal component* (PC2).

Table 9.1: Principal components of 2-D data (Figure 9.9).

	PC_1	PC_2
x_1	0.861	0.509
x_2	0.509	-0.861

In numerical form, PC_1 and PC_2 are in Table 9.1. This table should be understood as a summary of two linear equations:

$$PC_1 = -0.840x_1 - 0.542x_2 \quad PC_2 = -0.542x_1 + 0.840x_2, \quad (9.3.1)$$

or, in matrix form,

$$\begin{pmatrix} PC_1 \\ PC_2 \end{pmatrix} = \begin{pmatrix} -0.840 & -0.542 \\ -0.542 & 0.840 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}. \quad (9.3.2)$$

These numbers are called *factor loadings*. Hence PC_1 is somewhat more strongly related to x_1 (factor loading -0.840) than to x_2 (factor -0.542). The negative signs of factor loadings for PC_1 means that PC_1 will be smaller if x_1 and x_2 are larger. PC_1 “points” down-left on the figure. However, PC_2 grows if x_1 gets smaller and x_2 gets larger, hence it points up-left.

9.3.3 Explained Variance

The “importance” of components is normally defined by how much of variation in data do they explain. Figure 9.10 demonstrates how to understand variation and explained variation in data. The left panel shows three datapoints d_1 , d_2 and d_3 (light gray), and their centroid (black). The total variation is just sum of squared distances between the datapoints and the centroid. Here $V = e_1^2 + e_2^2 + e_3^2 = 4.301^2 + 2^2 + 2.915^2 = 31$ in these data. The right panel decomposes the distance into components that are parallel to PC_1 (yellow); and those that are parallel to PC_2 (purple). The variation, explained by PC_1 , is just sum of the squared components that are parallel to it (yellow). In these data it is $V_1 = 0.7071^2 + 1.414^2 + 2.828^2 = 28$. Hence PC_1 explains $28/31 \approx 90\%$ of the total variation. In an analogous fashion, PC_2 explains $3/31 \approx 10\%$ of the total variation.

As the principal components are orthogonal, so are the purple and yellow distance projection. Hence, by Pythagorean theorem, the square of the yellow (PC_1 -aligned) component, plus the square of the purple (PC_2 -aligned) component equals to the distance e squared. This means the sum of explained variations by the principal components equals to the total variation in data. In practice, it is often convenient to work with *explained variance ratio*, the variation, explained by individual PC-s, as a percentage of the total variation.

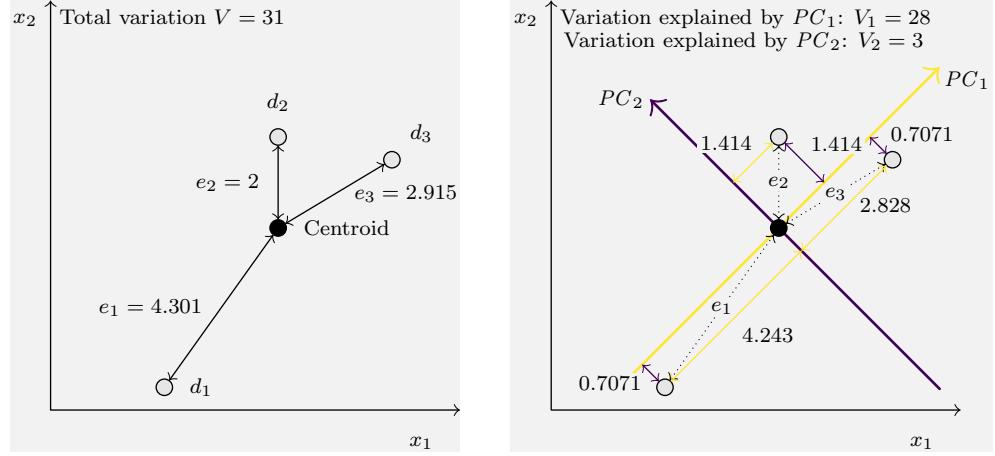


Figure 9.10: Explained variance. Left panel: gray circles are three datapoints d_1 , d_2 and d_3 ; and the dark circle is their centroid. The arrows e_1 , e_2 and e_3 are the distance between the datapoints and the centroid. The right panel shows the same distances, but not decomposed into the components that are parallel to PC_1 (purple), and those that are parallel to PC_2 (yellow).

Example 9.1: How big are emergencies

“Emergency” is a legal concept that opens doors for various government assistance. This may include additional firefighters or monetary assistance for the affected household. FEMA^a publishes data about different emergencies. However, emergencies are of very different size, stretching from broken water mains to major hurricanes. And when we want to know the “scale” of emergency, then no clear number is published. Instead, the measures are (i) total individual assistance (IA) - applications approved;(ii) total individual and households program - dollars approved;(iii) total housing assistance - dollars approved;(iv) total other needs assistance - dollars approved;(v) total public assistance grants - dollars obligated;(vi) dollars obligated to emergency work;(vii) dollars obligated to permanent work. All these numbers describe the scale of the emergency in some sense, we expect all these numbers to be large for major disaster. Obviously, these numbers are highly correlated, here are two examples:

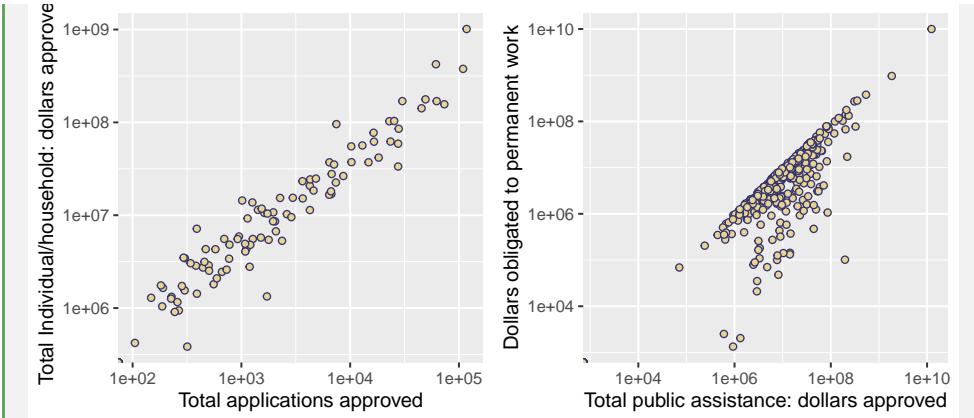


Figure 9.11: Published emergency measures are highly correlated. Dollars approved versus applications approved (left), and total public assistance versus dollars for permanent work (right). The sharp upper bound on the rhs figure shows that no more dollars are obligated than approved.

One way to find a single measure for “scale” of emergency is to do principal component analysis—it is quite likely that the first component will describe something akin a scale of the emergency. When we do this (on normalized data) then we get the following components (for simplicity we display the first four components only):

	PC1	PC2	PC3	PC4
nApplications	0.329	-0.671	0.327	-0.079
indTotal	0.396	-0.179	-0.181	0.360
housing	0.395	-0.177	-0.203	0.541
other	0.393	-0.186	-0.048	-0.728
pubTotal	0.376	0.407	0.292	0.014
emergency	0.381	0.321	-0.657	-0.192
permanent	0.372	0.427	0.548	0.064

The first component, PC_1 , appears to contain all seven variables by a roughly equal amount, all the loadings are between 0.3 and 0.4. Hence the first PC is just a (weighted) sum of all these numbers. The next component, PC_2 , includes the four first components with negative sign and the last three with positive sign. Note that the last three variables are about “dollars obligated” while the previous numbers are about “dollars approved” (except the first one, the number of applications). Hence it captures the difference between approved and obligated dollars. In an intuitive way, we can write the PC-s as

$$PC_1 = \text{Approved} + \text{Obligated} \quad PC_2 = \text{Obligated} - \text{Approved}$$

We do not discuss the further components as those explain virtually no variation in data (see below).

Variance, proportion of variance, and the cumulative variance of the four first components are:

Table 9.2: Standard deviation, relative variance, and cumulative relative variance, explained by the PC-s.

	Std.dev	Rel.var	Cum.var
PC1	2.490	0.886	0.886
PC2	0.840	0.101	0.987
PC3	0.224	0.007	0.994
PC4	0.156	0.003	0.997

The first component has standard deviation 2.490, and it explains 88.6% of total variation of data. The second component is much less important with standard deviation of 0.84, and it explains 10.1% of total variation. These first components together explain 98.7% of total variation in data. We can also display the relative variance as a barplot for all seven PC-s:

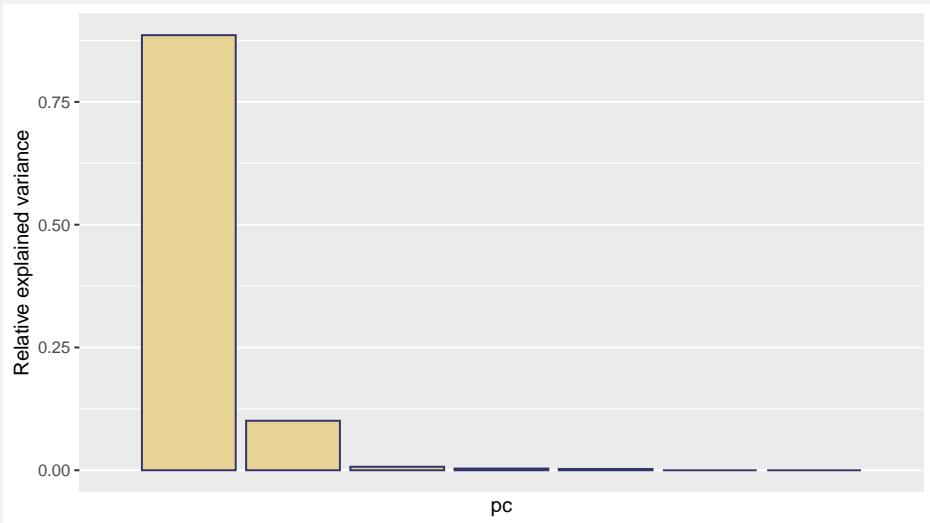


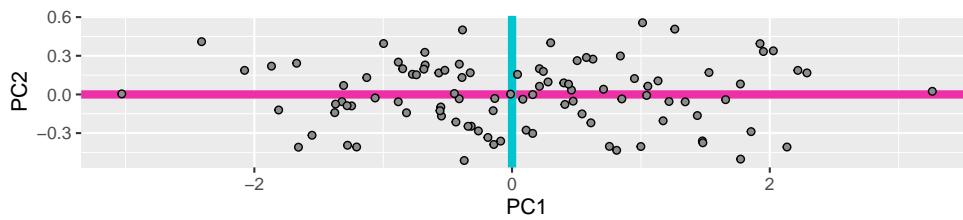
Figure 9.12: Proportion of variance, explained by PC-s. The figure suggests that in most applications, only the first two, or maybe even only the first PC is important.

The picture confirms the impression from Table 9.2—the first two PC-s are much more important than the other PC-s. In most applications we can probably safely ignore the PC-s 3-7, or even all the PC-s besides the first one.

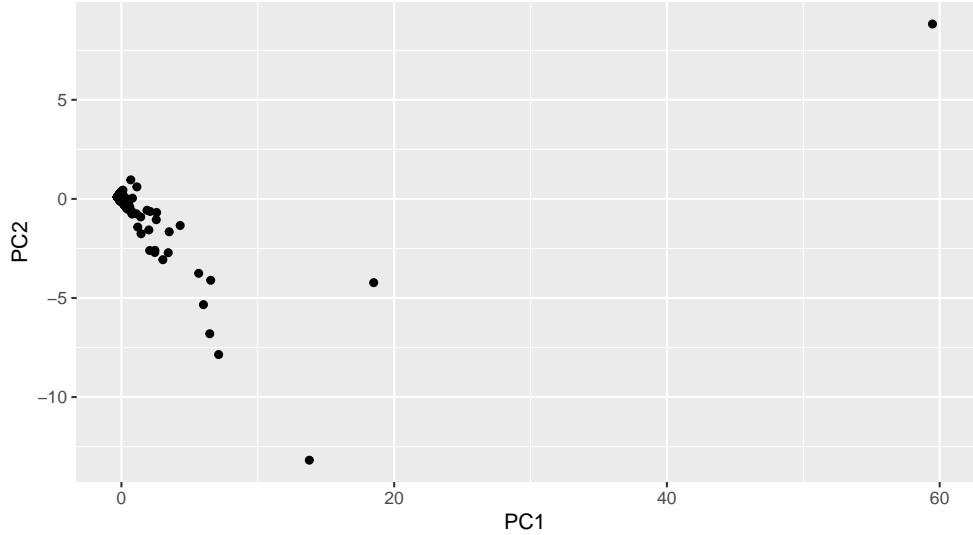
^aThe U.S. Federal Emergency Management Agency

9.3.4 Data Rotation

```
##          PC1         PC2
## x1  0.8609556  0.5086802
## x2  0.5086802 -0.8609556
```



variable	PC_1	PC_2
individual applications approved	-0.203	-0.200
approved individual/household total	-0.402	-0.414
approved housing assistance	-0.397	-0.409
approved other assistance	-0.353	-0.365
obligated to public assistance total	-0.426	0.427
obligated to emergency work	-0.400	0.385
obligated to permanent work	-0.413	0.393



- Note: no distinction b/w x and y
- Don't attempt OLS...

Data:

$$\mathbf{M} = \begin{pmatrix} x & y \\ 1 & 2 \\ 2 & 1 \\ 3 & 4 \\ 4 & 3 \end{pmatrix}$$

Now

$$\mathbf{M}'\mathbf{M} = \begin{pmatrix} 30 & 28 \\ 28 & 30 \end{pmatrix}$$

(Note: $\mathbf{M}\mathbf{e} = \lambda\mathbf{e}$) Solve for eigenvalues:

$$|\mathbf{M}'\mathbf{M}| = (30 - \lambda)(30 - \lambda) - 28^2 = 0$$

The solution:

$$\lambda_1 = 58 \quad \lambda_2 = 2 \quad (9.3.3)$$

The corresponding eigenvectors:

$$\begin{pmatrix} 30 & 28 \\ 28 & 30 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \lambda \begin{pmatrix} x \\ y \end{pmatrix} \quad (9.3.4)$$

and we have

$$\mathbf{e}_1 = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \quad \mathbf{e}_2 = \begin{pmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}$$

The eigenvector matrix

$$\mathbf{E} = \begin{pmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix}$$

Is a rotation matrix for angle $\cos \phi = 1/\sqrt{2}$ or 45° .

Rotated data:

$$\mathbf{ME} = \begin{pmatrix} x & y \\ 1 & 2 \\ 2 & 1 \\ 3 & 4 \\ 4 & 3 \end{pmatrix} \begin{pmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix} = \begin{pmatrix} 3/\sqrt{2} & 1/\sqrt{2} \\ 3/\sqrt{2} & -1/\sqrt{2} \\ 7/\sqrt{2} & 1/\sqrt{2} \\ 7/\sqrt{2} & -1/\sqrt{2} \end{pmatrix}$$

- Eigenvalue decomposition rotates data matrix
 - Orthonormal eigenvectors are the new base
- The largest eigenvalue corresponds to the most important dimension

9.3.5 Principal Component Regression

One widely used application of PCA is in the regression analysis. One can use the principal components instead of the original variables. This may give two advantages:

- Sometimes the principal components have clear interpretation, and hence the resulting coefficients have more meaningful interpretation than when using the original variables.
- Often the less important principal components add little value to the regression, so we can ignore those and get a simpler model.

Note that PC regression may not give any gains if the components that describe little variance in data still describe a lot of variance in the target variable.

Example 9.2: Principal component regression with 2-D data

Here we demonstrate principal component regression using 2-D data. Consider the data below:

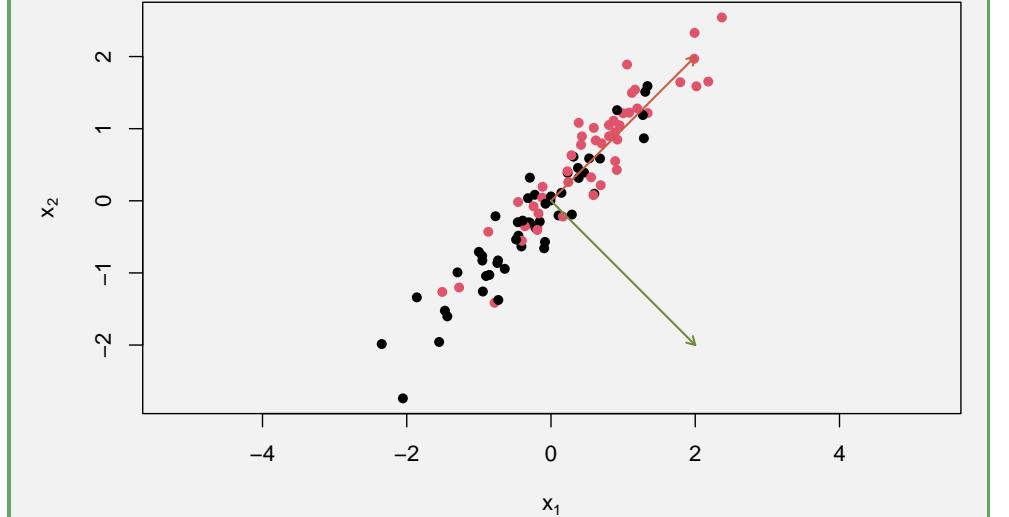


Figure 9.13: Narrow band of data points where color is associated with the bottom-left-top-right direction. This does not correspond exactly to any of the features x_1 and x_2 , but instead to their linear combination (rotated data). The principal components are depicted as arrows, with PC1 (orange) describing the direction maximum variation, and PC2 (dark green) the perpendicular direction.

The data contains two numeric features, x_1 and x_2 , and a color label, “black” or “red”. Our task is to predict the color based on x_1 and x_2 .

It is easy to see that red dots dominate in the top-right corner of the figure. We can use a simple logistic regression model

$$\Pr(\text{color}_i = \text{red}) = \Lambda(\beta_0 + \beta_1 x_1 + \beta_2 x_2). \quad (9.3.5)$$

The results are

	Estimate	Std. Error	z value	$\Pr(> z)$
(Intercept)	-0.1932	0.2289	-0.84	0.3987
x_1	0.0747	0.7820	0.10	0.9239
x_2	1.0835	0.7532	1.44	0.1503

In a counterintuitive fashion, neither x_1 nor x_2 show much significance here while we can clearly see that the red dots are clustered in the top-right corner. Even more, the point estimate for x_2 is negative although these are larger values that are associated with red color. The problem here is the fact that both features contain essentially the same information, and hence the design matrix is ill conditioned. The accuracy on training data, 0.71, is acceptable though for such a noisy image. We can cure the problem with principal component analysis. Doing PCA on the data matrix gives us

Table 9.3: Principal components of data in Figure 9.13

	PC1	PC2
x1	-0.71	-0.71
x2	-0.71	0.71

Table 9.4: Proportion of variance explained by components in Table 9.3

	1	2
variance	1.95	0.05
proportion	0.97	0.03
cumulative	0.97	1.00

As is evident from the tables, $PC1$ loads both x_1 and x_2 of equal amount and hence points to North-East, while $PC2$ contains a positive quantity of x_1 and a negative quantity of x_2 , and hence points South-East (see Figure 9.13). As $PC2$ very little information (its contains only 2% of the total variance), we can drop $PC2$ and use a simpler model

$$\Pr(\text{color}_i = \text{red}) = \Lambda(\beta_0 + \beta_1 \cdot PC1). \quad (9.3.6)$$

The results are as follows:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-0.0519	0.2251	-0.23	0.8177
PC1	-0.8187	0.1998	-4.10	0.0000

The results show that $PC1$ is now highly significant and of reasonable size. Accuracy on training data is the same, 0.71. But we got a simpler, model with more easily interpretable model.

Example 9.3: How are conservative family values and identity related to willingness to do good for society?

The worldview of people can be described with different dimensions, including family values (conservative versus liberal), and identity (global versus local), and many other. But how are these two value sets associated with the willingness to contribute to the society? Let's analyze this based on the World Value Survey, a large world-wide opinion survey.

We estimate a linear regression model in the form

$$\text{contribute}_i = \alpha + \boldsymbol{\beta}^\top \cdot \text{globalist values}_i + \boldsymbol{\gamma}^\top \cdot \text{gender values}_i + \epsilon_i \quad (9.3.7)$$

where *family values* is a vector of family-values related opinion, and *democratic values* is a vector of authoritarianism-democracy related viewpoints. The examples of family values include^a

doingGoodImportant It is important to do something for good of society.

trustUN how much confidence do you have in United Nations?

worldCitizen I see myself as a world citizen.

partOfNation I see myself as part of the nation.

maleLeaders men make better political leaders than women

maleExecutives men make better business executives than women

collegeBoy university education is more important for a boy than for a girl
When we run such a regression, we get the results

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-1.7280	0.0239	-72.24	0.0000
trustUN	0.0021	0.0048	0.44	0.6569
worldCitizen	0.1433	0.0053	26.81	0.0000
partOfNation	0.2503	0.0073	34.37	0.0000
maleLeaders	0.0772	0.0059	13.06	0.0000
maleExecutives	0.0342	0.0063	5.45	0.0000
collegeBoy	-0.0767	0.0057	-13.52	0.0000

The results suggest that those with stronger both global and national identity feel it more important to be good for society. The same is true for those who think men make better leaders, but not for those who believe higher education matters more for boys. The predictive power of the model is low with $R^2 = 0.041$.

Instead of estimating the effect of responses to individual questions, we can combine the answers into principal components, and include those in the regression instead. The principal components are

	PC1	PC2	PC3	PC4	PC5	PC6
trustUN	0.06	-0.43	-0.82	0.37	0.04	-0.01
worldCitizen	0.02	-0.67	0.05	-0.70	0.25	0.02
partOfNation	-0.06	-0.60	0.53	0.50	-0.32	-0.01
maleLeaders	-0.59	-0.01	0.06	0.16	0.50	-0.61
maleExecutives	-0.61	-0.00	-0.01	0.08	0.19	0.76
collegeBoy	-0.52	0.00	-0.21	-0.31	-0.74	-0.21

The first two components are easy to interpret: $PC1$ loads strongly with all three variables that describe the conservative gender roles, hence a large $PC1$ value describes liberal gender values (as the loadings are negative). $PC2$ loads on the globalist/national feelings and describes someone who does not trust UN and does not feel any attachment neither to the world nor her nation.

The importance of the components is

	1	2	3	4	5	6
variance	2.01	1.29	0.96	0.78	0.58	0.37
proportion	0.34	0.22	0.16	0.13	0.10	0.06
cumulative	0.34	0.55	0.71	0.84	0.94	1.00

Let's re-run the regression using the two first principal components only. Together these describe over 50% of the variance, and third component is also much harder to interpret. So we estimate a regression model

$$\text{contribute}_i = \beta_0 + \beta_1 \cdot PC1_i + \beta_2 \cdot PC2_i + \epsilon_i. \quad (9.3.8)$$

The results are

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-2.4423	0.0046	-531.80	0.0000
PC1	-0.0349	0.0032	-10.79	0.0000
PC2	-0.1854	0.0040	-45.87	0.0000

The results indicate that PC_1 —individuals who have more liberal viewpoints about the gender roles—are less likely to think it is important to do something good for the society. But the effect of PC_2 is much stronger—those who do not trust UN and do not feel belonging to a group do not think it is important to do good. The effect of the latter component is much stronger than that of the former, suggesting that feeling of attachment and identity is much more important factor in determining someone’s willingness to contribute to the public good. The model’s explanatory power is weak though, with $R^2 = 0.031$.

^aWVS opinion questions usually state the claim in a very conservative fashion and allow the respondents either to agree or disagree with it. Here the answers are re-coded in a way that larger positive numbers always denote more support for the claim.

9.4 Comparison of Clustering and PCA

As methods of unsupervised learning, both cluster analysis and PCA share a number of traits. Both can be used to discover certain patterns in data, and given such patterns, to simplify, compress, and interpret the data.

But they also differ in a number of ways. In case of clustering, we are primarily interested in homogeneous subgroups. An example case is in Figure 9.14. The left panel of the figure contains 5 reasonably distinct group that we can capture using clustering methods, such as k -means. We can use this group information in several ways:

- If the clusters correspond to the structural properties of the data, this helps us to interpret and understand the it.
- We can also design different measures to address different groups. For instance, different patients may require different treatment even with similar diagnosis.
- We can compress the data by replacing individual observations with the corresponding cluster center. Note that this does not constitute a dimensionality reduction: cluster center vectors are still of the same dimension as individual data vectors.
- Sometimes the group membership itself is of interest: which cases go together?

In case of principal components, our main task is to find a low-dimensional representation of data that contains most of the original information. This representation has a number of applications:

- Sometimes the reduction process itself reveals interesting properties of the data that can be interpreted.
- We can genuinely reduce the dimensionality of data by removing the (rotated) dimensions that carry little information. Unlike clustering, this process genuinely shrinks the data dimensionality. But the individual observations still

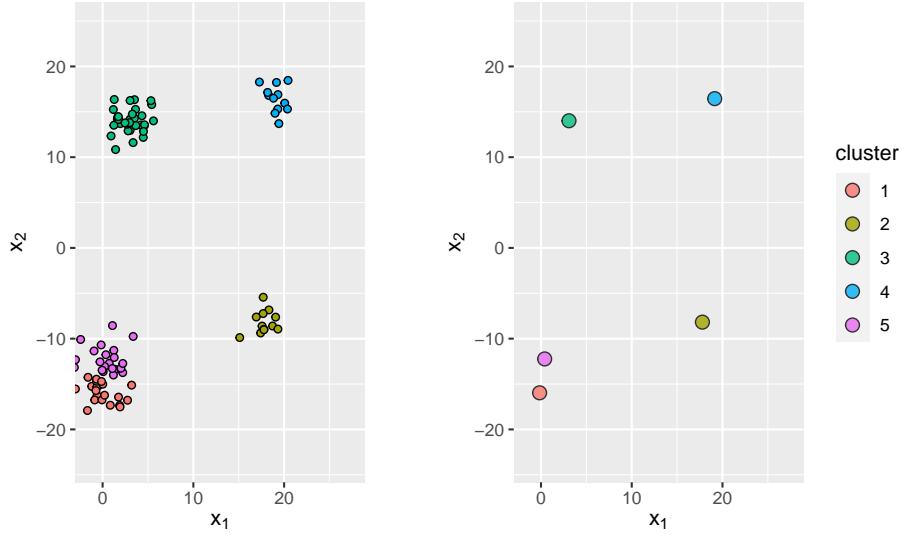


Figure 9.14: How cluster analysis treats data: homogenous subgroups (left panel) can be replaced by their corresponding cluster centers (right panel). In this way we can reduce the original 100-observation dataset to 5 different "types". These types can be either interpreted, one can design separate measures for each type, for instance marketing strategies in case of customer types, and one can also replace each observation with the cluster center in order to compress the data.

remain distinct and are not replaced by certain average observations.

Lower-dimensional data is both easier to analyze and compress.

These methods also work well on different types of data. Cluster analysis is designed for data that contains well-separated relatively homogeneous "blobs" while principal component analysis can handle datasets that form an elongated cloud.

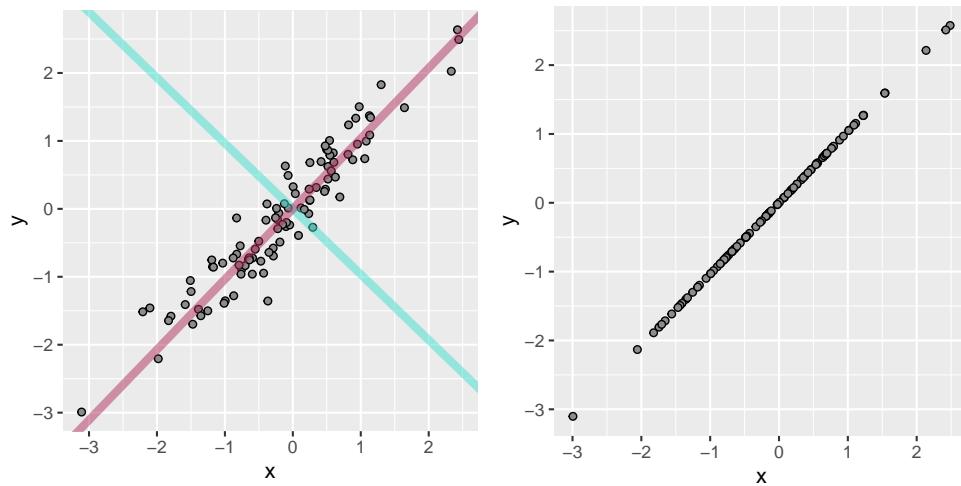


Figure 9.15: How PCA treats data: the dimension of maximum variance is treated as PC1 (red line, left panel) that carries most of the information. The other dimension (blue line) carries little information and is collapsed, resulting in rotated 1-D data (right panel). We may be interested in factor loadings, the relationship between the original features x_1 and x_2 , and the resulting principal components for interpretation and understanding. We may also prefer to analyze and store the reduced-dimensional data (right panel) instead of the original one.

Chapter 10

Applications

Contents

10.1	Recommender Systems	387
10.1.1	Collaborative Filtering	388
10.1.2	Problems with recommender systems	390
10.2	Generating Content: Generative Adversarial Networks	392
10.2.1	Technical details	392

10.1 Recommender Systems

Recommender systems are in many ways similar to ordinary supervised ML methods. Their aim is to predict users’ “product rating” over different products, and recommend those with highest rating. The rating can be numeric, like in cases where users literally rate movies on 1-5 scale, or it may be a binary “rating”, e.g. the indicator if someone bought or did not buy a product.

However, recommenders also differ from the standard supervised models in several important aspects:

- In ordinary supervised models we base the estimates on some sort of universal user characteristics, such as age or education. Recommenders instead rely heavily on other ratings by the same users, so in a way the other ratings the user has done is the main information we have, often it is even the only information.
- Recommendation data is typically sparse. While we can collect common background information for most users and possibly drop those cases where an important variable is missing, the users typically only rate a small minority of products. Hence we cannot rely on traditional methods, at least not without imputing the missing data.

10.1.1 Collaborative Filtering

The idea of collaborative filtering is the following: when predicting ratings by user i , we find a set of users who are “most similar” to the user i , and base our decision on their ratings. The “most similar” here means users who are similar in terms of how do they rate products, not in terms of the background characteristics like education and age. This approach is in many ways similar to k -NN or local regression, just that is adapted to sparse data where many datapoints must be imputed.

We start with a trivial example. Consider three fictional persons, *Ji*, *Chen* and *Su* rating two equally fictional movies, *Under the Bed* and *The Monk*. Movies are rated on numeric scale with “1” denoting the lowest and “5” the highest grade. Ratings given by these users are in Table 10.1.

Table 10.1: Two fictional movies rated by three fictional persons. Average is the users’ average rating over all movies they have rated.

Name	Movie	Rating	Average	Centered rating
Ji	Under the Bed	1	2.5	-1.5
Ji	The Monk	4	2.5	1.5
Chen	Under the Bed	3	2.0	1.0
Chen	The Monk	1	2.0	-1.0
Su	Under the Bed	4	2.5	1.5
Su	The Monk	1	2.5	-1.5

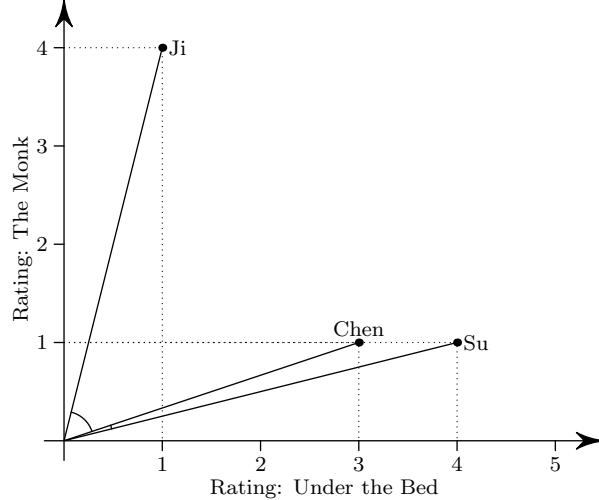


Figure 10.1: Two movies rated by three users, the same data as in Table 10.1 but now displayed graphically.

As this data is complete (all users rated all movies), we can easily depict the users in the 2-D rating space (Figure 10.1). The raters are depicted as vectors pointing from

the origin $(0,0)$ to the corresponding ratings (*Under the Bed* rating on the horizontal axis and *The Monk* rating on the vertical axis). A quick visual inspection also tells that Chen and Su are more “similar” than, for instance, Chen and Ji. Based on this quick picture we may already say that if Su liked a third move very much, Chen may also like it. But we are less certain what will Ji think about it as his tastes seem to be different.

In practice it is better to use centered cosine similarity, (Figure 10.2), not Euclidean distance (our eyes implicitly measure Euclidean distance). The centered distance is computed by subtracting the user’s average rating from all of their ratings (column *Centered rating* in Table 10.1). The figure shows the angles between Chen and Ji, and Chen and Su. The angle between Ji and Chen, and between Ji and Su is 180° , while the angle between Chen and Su is 0 . Hence according to this figure, Chen and Su are very similar, while Ji is their exact opposite. This leads to exactly the same conclusion in this case as the Euclidean distance—if Su likes a third movie, Chen may also like that one.

Cosine similarity:
 $c(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^\top \mathbf{y}}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|}$. See
[Section 6.1.2 Cosine similarity and angular distance, page 260](#).

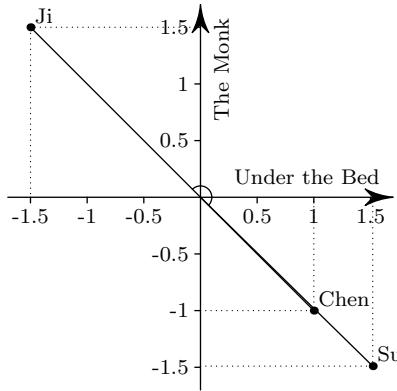


Figure 10.2: Two movies rated by three users, the same data as in Table 10.1, column Centered rating. The point for Su is shifted a little bit on the figure for clarity.

The centered data does not look particularly informative in the 2D case as all the vectors are aligned on the exact same NW-SE line. However, in a 3D or in a higher dimensional space this is not so any more. But even in the 2-D case we see Chen and Su ratings pointing in one direction and Ji’s rating pointing to the other direction. This tells us that Chen and Su are rather similar: both rate “Under the Bed” better than “The Monk”. But Ji thinks the other way around.

This example assumed all users have rated all movies. But what to do in a more realistic example where users only have rated a small fraction of products? An obvious choice is to replace the missing values with the users’ average values. This is where centered distance plays a very useful role—centered versions of such imputed values is zero, and 0-length components do not affect cosine similarity. So computed user similarity is less influenced by our imputations.

After computing the users’ similarity, the rest of the algorithm may proceed as follows:

1. Find the most similar users for any given user, for instance 10 most similar users.
2. Use the ratings of the most similar users to compute predicted rating for the given user. In case of multiple conflicting ratings for a single item, one may, for instance, predict average of these. Care must be taken for not to include the imputed values into the prediction.
3. Finally, the algorithm suggests the highest predicted values from the list computed above. This results in recommendations like “users who like this movie also liked that movie”.

10.1.2 Problems with recommender systems

While good recommenders are valuable both for customers and businesses, recommenders are not without their issues.

“Blind” recommendation algorithms may also suggest items that we consider harmful. There is a well-known but unconfirmed story about teen pregnancy that Target learned, based on her buying decisions, before her parents. As another, more recent example, in 2022 lawmakers accused Amazon for recommending food preservative that has been used for suicides ([Jackson, 2022](#)). From the algorithm’s viewpoint, it is perfectly valid thing to do—if someone is buying items that are helping to commit suicide, the algorithm happily recommends the related items too. But unlike the algorithms, we do not think in this way. It is also not immediately obvious how to address such issues—algorithms are good in picking up all kinds of patterns, including many patterns we are not aware of, and basing their decisions on all of these. First later will humans discover that some of the recommendations are dubious at best from the ethical standpoint.

Social media recommenders may build a network of other individuals, where connections are made of various common links, such as common friends, schools both persons have attended together, events they have both liked, and so on. As an upside it helps to find new friends, or re-connect with old acquaintances. But they may also attempt to reconnect people with their ex-s or abusers, and even worse, remind and reveal an abuser about your presence.

Some recommenders are easy to game. If recommendations are partly based on clicks or likes, then items that many users click on are recommended increasingly more, resulting in even more clicks on these links. This results in “clickbaits”, titles that look interesting although may not contain anything relevant. More advanced users can set up hundreds of bots that like each other’s stories and in this way fool the algorithm to think that this is something the other users want too. This is one way to spread misinformation over social media.

Recommenders have problems related to shifting human taste, e.g. after listening 10 songs of a certain genre, the recommender learns to suggest even more similar songs. However, the user gets bored of such music instead and looks for something different. Such shifts are very hard to model.

Lack of variety in recommendations also make echo chambers possible. If a user is for some reason interested in a certain type of information, the recommenders will start suggesting even more sources that offer just this kind of information. It may

effectively remove all alternative viewpoints from that user's information sphere, and re-enforce the feeling that they represent the majority. This is a mechanism behind political polarization. In extreme cases, it may even breed weird cranky movements, such as Flat Earth movement.

10.2 Generating Content: Generative Adversarial Networks

Prerequisites: [Section 6.6.2 Convolutional Neural Networks](#), page 307, [Section 8.2 Gradient Ascent](#), page 341

Generative Adversarial Networks (*GAN*-s, [Goodfellow et al. \(2014\)](#)) are models that create content according to certain patterns. The idea of *GAN*-s is the following: two networks (adversaries) are working together. One of them (discriminator) is fed actual examples, and its task is to distinguish between the actual examples and generated examples. The other network (generator) is creating new examples out of some random data, and its task is to “fool” the discriminator to think these are real examples. In the process, the networks are teaching each other, and they are getting better in both distinguishing the real and generated examples, and also in generating such examples.

The generator network is designed to produce examples based on some kind of random inputs. For instance, it may take a vector of random numbers as input, and transform these into an image through a series of dense and convolutional layers. Through the training process the network learns to adjust the weights and convolutional filters in such a way that the result resembles example images. Discriminator, in turn is just an image categorization tool, however, it needs to be good in categorizing the actual and generated details.

10.2.1 Technical details

In order to fix the ideas, let’s assume we are creating images. Each image can be represented by a vector $\mathbf{x} \in X$, where X is a set of all images the network can handle. For instance, X may be all possible 100×100 pixel images with three color layers, coded as color values in interval $[0,1]$. So X is a set of $100 \times 100 \times 3$ tensors where each element $x_{ijk} \in [0,1]$. For simplicity, we still refer to \mathbf{x} as vectors, a vector of tensors if you wish.

Denote the discriminator by D . In this context it is a function $D : X \rightarrow [0,1]$, a function that takes in an image \mathbf{x} and based on the image, it computes a number—the probability that the input is a real image, not a generated one.

Generator G is also a function $G : \mathbb{R}^k \rightarrow X$, it takes in a g -dimensional vector of random numbers \mathbf{z} and converts it to an image of type X (e.g. 100×100 pixels and three color channels). So $G(\mathbf{z})$ is an image. We need to give generator some inputs even if these are random numbers, otherwise it will be able to only generate a single image. The inputs do not have to be just random numbers—a good choice is to include something like a “prompt”, a description of what image the user may want to get, or maybe some other kind of context, e.g. the text on the nearby pages if the task is to illustrate a book.

These two networks are “adversaries” with the opposite tasks: the discriminator attempts to compute probabilities in a way that probability of the real image being a real is one, and the probability that the generated image is real is zero:

$$D(\mathbf{x}) = 1 \quad \text{and} \quad D(G(\mathbf{z})) = 0. \tag{10.2.1}$$

The generator, however, attempts to create such images that fool discriminator to think that they are real, it wants to achieve

$$D(G(\mathbf{z})) = 1. \quad (10.2.2)$$

Training adversarial networks proceeds in a broadly following fashion ([Goodfellow et al., 2014](#)). Below, we assume the training is performed using stochastic gradient ascent using batch size m .

1. First, train the discriminator:

- (a) Sample m random inputs \mathbf{z} and use generator G to create m artificial images based on these input vectors.
- (b) Sample another minibatch of m real images \mathbf{x} .
- (c) Train discriminator by a single step of GA maximizing the objective function

$$V_D = \sum_{i=1}^m [\log D(\mathbf{x}^i) + \log(1 - D(G(\mathbf{z}^i)))] \quad (10.2.3)$$

V_D obtains its maximum if all $D(\mathbf{x}^i) = 1$ and all $D(G(\mathbf{z}^i)) = 0$, exactly what we want the discriminator to do.

2. Next, train the generator:

- (a) Sample m new noise vectors \mathbf{z}
- (b) perform a single step of gradient descent by minimizing the objective function

$$V_G = \sum_{i=1}^m \log(1 - D(G(\mathbf{z}^i))) \quad (10.2.4)$$

V_G obtains its minimum if $D(G(\mathbf{z}^i)) = 1$, exactly as needed by the discriminator.

3. Repeat the two steps above until convergence.

Chapter 11

Responsible Data Science

As artificial intelligence is used more and more widely in our everyday lives, we encounter more and more situations that leave the technical/statistical side of AI. In this chapter we discuss these question with focus on social and ethical issues.

Contents

11.1 Explainable AI	395
11.2 Social inequality	396
11.2.1 Who Are Represented in Big Data?	396
11.2.2 Big Data, Big Inequality?	397
11.3 Fairness and discrimination	397
11.3.1 Fairness versus efficiency	397
11.3.2 Individual fairness and group fairness	398
11.3.3 Fairness: Different Measures are Incompatible	399
11.4 Human Versus Algorithmic Decision-Making	402

11.1 Explainable AI

A few simple statistical models can be explained fairly easily. For instance regression models are relatively easy to understand, but example-based k -NN and simpler decision trees are also not that complicated. However, many more advanced models, in particular neural networks, are essentially black boxes.

But humans often want explanations. Imagine a situation where the bank turns down your loan application. Why? Will you be happy with the clerk explaining that “I just pushed the button and this is what the computer told me...” If at the same time another, at least superficially similar customer who also happens to be of a different race got her load approved, then the situation may look quite troublesome for the bank. An explanation is urgently needed but what even constitutes a suitable explanation?

Liao *et al.* (2020) discuss the types of explanations that users of AI applications expect and need. They distinguish four broad types:

1. Global: explain the model. Explain the model using a global structure, e.g. weighting of features, as an approximate decision tree, or other decision rules.
2. Local: explain predictions for a particular instance. Which features of the particular case made the model predict what it predicts?
3. Counterfactual: how will prediction change when we change features? Which features should we change to get a certain prediction?
4. Example-based: provide examples of similar cases where the model provided similar predictions, and slightly different cases where the predictions were different.

Many users ask related questions in order to understand better the limitations of the model (What is permissible? How can I improve the training?) Another important application is to manipulate inputs in order to avoid undesirable outcomes. If the model predicts that the product will not be successful, then we want to know what should be changed to make it a success.

11.2 Social inequality

As any other tool, statistics and machine learning can be used for good and for evil. Ethical dilemmas are nothing new to us, but as technology opens new avenues, we sometimes have to ask the age-old questions in a totally new context. Even if we can do this, should we do it? And how should we proceed in delicate cases?

11.2.1 Who Are Represented in Big Data?

Statistical models are trained on data and hence reflect the properties of the underlying data. When collecting dedicated survey data, such as World Value Survey, the researchers usually spend quite a bit of effort to ensure that the data is representative, and carefully document the sampling methods and resulting sampling weights in case certain populations or geographic regions are over/underrepresented. See Section 1.2.3 Sampling Process.

Unfortunately, such steps are often left undocumented in case of Big Data. Even worse, it is often unclear what would the theoretical population and sample frame even be in many cases. For instance, large NLP models are typically trained with text downloaded from internet. However, what would be a good representative sample of text? We know that different people leave behind a different amount of text depending on their habits and internet access. But should we strive to an equal representation of people? Text—language, is after all most cases produced as a part of communication between several persons. So perhaps it is appropriate that the loud voices are overrepresented in a text corpus? As internet is also extremely complicated, it would be difficult to compute the sampling weights.

So it is not surprising at all that complex models that are trained on complex real data will re-produce various unfavorable traits that we encounter in the real world. And if we do not want the model to reflect such views, then what kind of views do

we want it to reflect? For instance, should we refer to a certain group of immigrants as *undocumented* or *illegal*? As people have different opinion about the appropriate language here, addressing this question is necessarily political (Bender *et al.*, 2021).

11.2.2 Big Data, Big Inequality?

Boyd and Crawford (2012) discuss access to Big Data. Big Data is mainly collected by players in the internet industry, such as social media or online retail companies. These firms will have both the data and the resources for analysis, and they will decide who else have access to data. It probably leads to inequality in terms of research access where those with resources (prestigious universities and rich private research labs) will have access, and the other cannot easily participate in the relevant debate. Neither can they evaluate the quality of published big data-based research. The fact that the private gatekeepers do not follow similar transparency and public access requirements as the public sector data collectors will hamper analysis of topics that the data collectors find inconvenient.

11.3 Fairness and discrimination

Fairness is an intuitive but imprecise concept. As automated decision-making has become more widespread, this has also created more interest for fairness. At the same time, large-scale data collection has made it more easy to assess such decisions. Not surprisingly, we can see that a vague concept like fairness is not easy to operationalize. But nevertheless, it matters in everyday decisionmaking.

11.3.1 Fairness versus efficiency

Let's start with a simple motivating example. Consider a world, populated with two types of people, reds and greens. The color of a person is immediately obvious to everyone, in a similar manner like gender or immigration background, and it is hard to hide it.¹ The people also come in two skill sets: high skilled and low skilled. For historical reasons, 2/3 of reds are high skilled while only 1/3 of the greens are high-skilled. Unlike the color, skills are hard to observe and require costly tests and interviews to assess.

You are a hiring manager of a big firm, and a job posting brought in 10 candidates, 5 of which are red and 5 green. But you only have funds to interview 4 candidates. Which of these candidates will you test? Obviously, it would be most efficient to only interview red candidates—we expect 2 or 3 of them to be high-skilled, while the chances that none of the green candidates are high-skilled is fairly large, approximately 20%. But for the work, the only thing that matters are skills, the color is irrelevant. If you take this approach, you are discriminating the candidates based of an irrelevant trait, color. But if you decide to give everyone a chance and also interview greens, you are

¹One may argue that neither of these characteristics are, in fact, immediately obvious. This is correct. But simple and easily available information about people, such as their name, skin color, or accent is enough to estimate these characteristics fairly well. Even more, what matters below is not how the individuals themselves identify themselves, but what the evaluators think about them.

less likely to find a suitable candidate. What should you do? Is it OK to only focus on economic efficiency² or should you ensure that you treat candidates of both color in a similar manner?

This is an example of a trade-off between economic efficiency (at least in short term) and fairness. Fairness may come at cost. Your decision will probably be affected by your beliefs, the corporate policy, and also by legislation. In everyday lives we need to do similar decisions quite frequently, decisions that may hurt certain other people.

Below, we discuss a few selected aspects of fairness, and show that there is not just a trade-off between efficiency and fairness, but also between different concepts of fairness.

11.3.2 Individual fairness and group fairness

Unfortunately, we cannot just be “fair”. In everyday language, the word *fairness* is typically used in a rather vague way. Depending on the context, it can be understood as equal treatment, appropriate treatment, morally justified treatment, and in a myriad of other ways. But even these, more specific moral principles, are hard to define in a precise manner. Below, we focus on fairness in the equal treatment sense.

One of the central concepts in fairness discussion is *individual fairness*. It captures the idea that individuals who are similar from a particular task’s perspective should be treated similarly. For instance, two candidates on a job interview who are equally qualified for the respective job, should be treated in the same way. In particular, they should not receive different treatment because the job-irrelevant attributes, such as ethnic background.

Another related equal treatment-related concept is *group fairness*. It is conceptually somewhat similar, and requires that the relevant groups should be treated in a similar manner, at least in the statistical sense. In case of the job interview example above, we expect to see that a similar percentage of candidates from both groups will be hired, given they are equally qualified.

Although both individual and group fairness seem largely similar, they are not compatible—one cannot achieve both, unless in very specific circumstances (Kleinberg *et al.*, 2016), see also [Section 11.3.3 Fairness: Different Measures are Incompatible](#), page 399. We need to choose between these two (and potentially more) incompatible fairness definitions.

Measuring fairness has a number of problems. For instance, the centrality of individual fairness—*similar* people are treated in a similar manner—requires us to decide which people are similar. In particular, why are certain traits, such as gender or race, treated as irrelevant while others, such as immigrant status are not? Such decisions rely on our common understanding on relevant traits and permissible discrimination, and hence it cannot be an absolute measure. Another problem is related to assessing the general equilibrium effect in the presence of multiple equilibria. (Fleisher, 2021) provides an example how affirmative action can, over time, result in less qualified minority group to become similar to the majority group. While in sort-term, it includes affirmative action and hence the groups are not treated in a similar manner,

²Situation where you choose to focus solely on economic efficiency and only interview reds is called *statistical discrimination*.

they are treated similarly in long run. What is considered fair treatment depends on the time horizon and the equilibrium type we focus on.

11.3.3 Fairness: Different Measures are Incompatible

Prerequisites: Conditional probability: [Section 1.3.3 Bayes theorem](#), page [29](#)

One of problems with “unfair” treatment and “algorithmic bias” that attracted wide attention in recent years is related to algorithms, used in the U.S. criminal justice system. [Angwin et al. \(2016\)](#) analyzes Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) algorithm, a profiling algorithm that is widely used to predict whether the defendant is likely to commit another crime when released. Their analysis turned up “significant racial disparities”. In particular, whites who were labeled as “high risk” by the algorithm did not re-offend in 23.5% of cases, while African-Americans who were labeled “high-risk” did-not re-offend in 44.9% of cases. To put it differently, substantially more low-risk African-Americans were mis-categorized into high-risk category than the corresponding whites. Unfortunately, this label is not only of interest for academic researchers—the perceived riskiness of recidivism influences sentences, right to bail, probation and other measures that have important real world effects on individual lives. The proponents of the COMPAS score have countered the criticism by demonstrating that at given score,³ both whites and African-Americans have similar probability to re-offend. So COMPAS score is a fair measure.

The problem boils down to different concepts of fairness. [Angwin et al. \(2016\)](#) criticism centers on *group fairness*, i.e. requirement that similar *groups of people*, here defined by race, should be treated similarly ([Jacobs and Wallach, 2021](#)). So whites who do not re-offend should have the same mis-classification rate as blacks who do not re-offend. This is clearly violated with COMPAS score. However, its advocates rely on *individual fairness*, requirement that similar *individuals* to be treated equally: given they receive equal COMPAS score, the decision should be the same, independent of race and other personal characteristics. Unfortunately, these two concepts of fairness are not compatible in general ([Kleinberg et al., 2016](#)). Except in very specific cases, such as when we can perfectly predict re-offenses, it is only possible to be fair either in one way or the other way, but not in both ways at the same time. Next, we explain it both theoretically and provide a numerical example.

Consider a problem, similar to that of COMPAS. There are two groups of people, Greens and Reds. For every person, we are interested in whether they re-offending behavior R : they may either re-offend ($R = 1$) or not re-offend ($R = 0$). We also know their individual characteristics X . It has only two possible values: either $X = 0$ or $X = 1$. You can imagine that X measures whether they have committed any crimes earlier, with $X = 0$ means no previous offenses and $X = 1$ means a previous criminal record. However, for whatever reason, there are more people with a criminal record among Greens than among Reds so that $\Pr(X = 1|Green) > 0.5$ and $\Pr(X = 1|Red) < 0.5$.

³COMPAS assigns each individual a risk score between 1 and 10, with 1 meaning “very unlikely to re-offend” and 10 meaning “very likely to re-offend”.

Fortunately, we can construct a test, a model similar to COMPAS, that predicts someone's re-offending probability R based on the individual characteristics X . With only two possible categories and two possible X values, we can just compute the re-offending probability, depending on X and color $\Pr(R = 1|X, \text{color})$. Assume that the probabilities we find do not depend on color:

$$\Pr(R = 1|X) = \Pr(R = 1|X, \text{Green}) = \Pr(R = 1|X, \text{Red}). \quad (11.3.1)$$

So in this sense the model is color-blind. The model only looks at X , not at the color, and makes the predictions based on that. Assume that $\Pr(R = 1|X = 0) < 0.5$ and $\Pr(R = 1|X = 1) > 0.5$, hence the test predicts that a person with no previous criminal record will not re-offend, but those who have previous criminal record will re-offend.

Let us illustrate this with a numerical example (Figure 11.1, left panel). There are 24 reds and 24 greens in total. However, out of those 24, 16 reds and 8 greens have never committed a crime ($X = 0$), and 8 reds and 16 greens have committed a crime earlier ($X = 1$). We also know that those with $X = 0$ have probability of re-offending $\Pr(R = 1|X = 0) = 1/4$, so 4 Reds out of 16 and 2 Greens out of 8 will re-offend. For those with previous criminal record, the probability of re-offending $\Pr(R = 1|X = 1) = 3/4$ so out of 8 Reds 6 will re-offend while the same is true for 12 out of 16 greens. See Figure 11.1, left panel. Based on these probabilities, we will predict that everyone with $X = 0$ will not re-offend and everyone with $X = 1$ will, and this does not depend on color. So our test is color-blind, and in this sense fair.

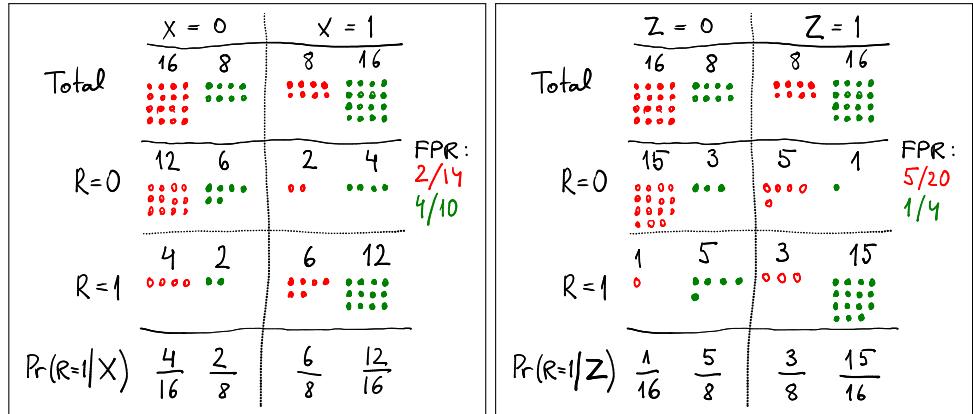


Figure 11.1: Left panel: the test is color blind: $\Pr(R = 1|X)$ does not depend on color. However, now greens' FPR is almost three times that of reds.

Right panel: a test that ensures the FPR-s are equal. However, now color is an important predictor of $\Pr(R = 1|X)$.

However, if we compute the false positive rate, we come to a different conclusion. Here, FPR is the probability that a person who will not re-commit a crime, $R = 0$, will be mis-classified as re-offender. As we classify the re-offenses solely based on X , we mis-classify all those who will not re-offend ($R = 0$) but have previous criminal

$FPR = \frac{FP}{N}$, $FNR = \frac{FN}{P}$. See more in [Section 5.2.1 Confusion matrix](#), page 232.

record ($X = 1$). So $FPR = \Pr(X = 1|R = 0)$. We can easily compute this probability from Bayes' theorem

$$\begin{aligned}\Pr(X = 1|R = 0) &= \frac{\Pr(R = 0|X = 1) \cdot \Pr(X = 1)}{\Pr(R = 0)} = \\ &= \frac{\Pr(R = 0|X = 1) \cdot \Pr(X = 1)}{\Pr(R = 0|X = 1) \cdot \Pr(X = 1) + \Pr(R = 0|X = 0) \cdot \Pr(X = 0)} \quad (11.3.2)\end{aligned}$$

It is obvious that even if $\Pr(R = 0|X = 1)$ and $\Pr(R = 0|X = 0)$ are equal for both groups, these probabilities are not as long as $\Pr(X = 1)$ and $\Pr(X = 0)$ differ. Hence a color-blind model that estimate the re-offending probability *cannot* provide similar FPR for both groups as long as the groups are not equal! The margin of the left panel shows that for reds, $FPR = 1/7$ while for greens, the probability is $4/10$. Hence low-risk greens have almost three times larger chances to be mis-classified as high-risk than the corresponding reds.

This example corresponds broadly to COMPAS model. The model uses a set of individual background variables to compute the re-offending probability, and finds that given the background, the probability does not depend on race. However, the FPR differs by race.

Now assume that instead of characteristic X , we observe feature Z , say the city the people are living. Z is also related to re-offending with $Z = 1$ being associated with higher likelihood to re-offend. However, now it turns out that the FPR is equal to Reds and Greens. Figure 11.1, right panel, shows a numeric example, where based on Z we find that $FPR = 1/4$ for both groups. Whatever your color, the low-risk non-offenders have 25% probability to be mis-categorized as re-offender.

However, the test based on Z is not color blind:

$$\begin{aligned}\Pr(R = 1|Z = 0, \text{red}) &= 1/16 \Pr(R = 1|Z = 0, \text{green}) = 5/8 \\ \Pr(R = 1|Z = 1, \text{red}) &= 3/8 \Pr(R = 1|Z = 1, \text{green}) = 15/16.\end{aligned} \quad (11.3.3)$$

This test is unlikely to satisfy the fairness requirements either. While now the low-risk individuals have similar probability to be mis-classified as high risk, we find that color is a very important predictor of re-offending. In particular, whatever Z , we categorize all reds as non-offenders and all greens as offender. This feels very unfair.

Obviously, in a real application we may find that our model is fair neither in one nor the other sens but gives results somewhere in-between. It all depends on what kind of information we have access to, and how it is correlated with re-offending.

There are three separate issues that give us this unfortunate result. The first problem is pure technical—the test is imperfect, in particular $\Pr(R = 1|X = 1) > 0$ —we are unable to perfectly tell who is low-risk. Unfortunately, there is no reason to believe that we are able to design perfect tests in the future either.

The second problem is that the percentage of high-risk individuals depends on color. Why is it like this? Is it because of some sort of historical discrimination? Because of unequal access to education or other resources? Something else? It is

unlikely that we are able to completely eliminate such inequality in the future, but measures to improve the matters are definitely possible.

The final problem here is the fact that these two fairness concepts—individual fairness and group fairness—are incompatible. We use the same word, “fairness”, to denote somewhat different concepts, and intuitively we feel that both are important. But that does not make these two concepts compatible.

Part of the problem is that the group fairness concept is based on group labels that are irrelevant as predictors, even more, that are supposed to be irrelevant as predictors. If we believe that group labels should not be used for prediction, and they do not carry any information (as in the first example), then why do we want the fairness to be based on the “irrelevant” group labels? There are no good answers. It just feels “fair”.

But whatever is the fundamental problem, the policymakers are facing an inconvenient choice. They have to decide between

- Ignoring the equal treatment principle
- Ignoring the score-balancing requirement
- Not using the test at all. However, in the example above this easily leads to perfect color discrimination where only Reds are hired.

Obviously, one can also use a combination of these options.

11.4 Human Versus Algorithmic Decision-Making

Algorithms are often criticized as “obscure”, in particular when the inner workings of those are not published. Sometimes it is claimed that we should not use algorithms at all as algorithms are no less biased than humans. However, such claims miss a few important points. While complex algorithms are always obscure, human mind is no more transparent. While we can publish the inner details of algorithms (although not necessarily understand these), this is not possible in case of human brains. We can also analyze the data, and access possible problems there, but again, this is not possible to do with humans.

Instead of discussing the “obscurity” and “biasedness”, we should ask if algorithms can do better decisions than the relevant humans. For instance, can judges make better decisions if they have access to an algorithmic result? ([Kleinberg et al., 2018](#)) show that this is indeed the case in case of NYC judges. The judges have to decide whether to jail or release arrested criminals, and the authors show that judges’ decision is much affected by seemingly random factors. They tend to keep too many low-risk defendants in jail while releasing too many high-risk defendants. Algorithm would achieve a similar crime reduction with 20-40% smaller jail rate, or alternatively, at a similar jail rate it had achieved 25-15% smaller crime rate.

Appendix A

Mathematics

These notes assume you are reasonably familiar with basic calculus and a few other mathematical concepts, such as logarithm. Below is a list of the most important rules with little explanations.

A.1 High-School Mathematics

A.1.1 Logarithm

Definition: a is *logarithm* of x if $e^a = x$ where $e = 2.71828 \dots$, and we write $a = \log x$. For instance, $\log 7.389 \approx 2$ as $e^2 \approx 7.389$.

Note: e-based logarithms as defined above are also called *natural logarithms*, and sometimes denoted by *ln* instead of *log*. Often the notation *log* is reserved for *decimal logarithms*, defined as $\log_{10} x = a$ if $10^a = x$. Sometimes (in information theory for instance) we also use *binary logarithms* where $\log_2 x = a$ if $2^a = x$. Notation differs in different fields and between different authors. In these notes, *logarithm* always means natural logarithm and is denoted by *log*. If needed, other logarithms are denoted by \log_{10} or \log_2 by explicitly writing their base.

Properties

$$\log x^\alpha = \alpha \log x \quad \text{and} \quad \log(xy) = \log x + \log y \quad (\text{A.1.1})$$

Logarithms of different base can easily be converted as

$$\log_b x = \frac{\log x}{\log b} \quad (\text{A.1.2})$$

Limits involving logarithm

TBD: $\lim_{x \rightarrow 1} \log x = x - 1$ etc

TBD: $\lim_{\epsilon \rightarrow 0} (1 + \epsilon)^\alpha = 1 + \alpha\epsilon$

$$\lim_{x \rightarrow 0} x \cdot \log x = 0 \quad (\text{A.1.3})$$

Proof A.1.1

Write $x \log x = (\log x)/(1/x)$ and use L'Hospital's rule.

A.1.2 Differentiation**Definition**

Derivative in case of univariate function is defined as

$$f'(x) \equiv \frac{df(x)}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}. \quad (\text{A.1.4})$$

The definition in case of multivariate calculus is defined in an analogous way. If we only change one variable at a time, we treat the others as constants, and essentially we are back at the univariate case. Just the result is called *partial derivative* now, and denoted with ∂ symbol:

$$\frac{\partial f(x, y, z, \dots)}{\partial x} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x, y, z, \dots) - f(x, y, z, \dots)}{\Delta x}. \quad (\text{A.1.5})$$

List of common differentiation rules

Here is a (non-exhaustive) list of the common rules for calculating the derivative of simple functions, and combinations of functions.

$$\frac{d}{dx} x^\alpha = \alpha x^{\alpha-1} \quad (\text{A.1.6})$$

$$\frac{d}{dx} \log x = \frac{1}{x} \quad (\text{A.1.7})$$

$$\frac{d}{dx} \log_b x = \frac{1}{\log b} \frac{1}{x} \quad (\text{A.1.8})$$

A few general rules for differentiation:

$$\frac{d}{dx} [\lambda f(x) + \mu g(x)] = \lambda f'(x) + \mu g'(x) \quad \text{differentiation is a linear operator} \quad (\text{A.1.9})$$

$$\frac{d}{dx} f(g(x)) = f'(g(x)) \cdot g'(x) \quad \text{chain rule} \quad (\text{A.1.10})$$

$$\frac{d}{dx} f(\lambda x) = \Lambda f'(x) \quad \text{application of chain rule} \quad (\text{A.1.11})$$

A.2 Matrix calculus

As linear algebra is the language of statistics, we may often want to do optimization in matrix form too. This requires *matrix calculus*. It is essentially ordinary calculus,

supplemented with a set of rules how to collect back into matrices all the resulting objects that arise when differentiating the individual matrix components.

Start simple. If we have a matrix, we can define its derivative with respect to a scalar just as a similar matrix where we have differentiated each component with respect to that scalar:

$$\frac{\partial}{\partial \lambda} \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1K} \\ x_{21} & x_{22} & \dots & x_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{NK} \end{bmatrix} = \begin{bmatrix} \frac{\partial \lambda}{\partial x_{11}} & \frac{\partial \lambda}{\partial x_{12}} & \dots & \frac{\partial \lambda}{\partial x_{1K}} \\ \frac{\partial \lambda}{\partial x_{21}} & \frac{\partial \lambda}{\partial x_{22}} & \dots & \frac{\partial \lambda}{\partial x_{2K}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \lambda}{\partial x_{N1}} & \frac{\partial \lambda}{\partial x_{N2}} & \dots & \frac{\partial \lambda}{\partial x_{NK}} \end{bmatrix}. \quad (\text{A.2.1})$$

Essentially we take derivatives of each element of a collection (a collection we call matrix) and put these back into a similar collection. So little changes if we differentiate matrices with respect to a scalar.

Differentiation with respect to a vector requires additional rules, however. Let's take the simplest case: differentiate a scalar function $f : \mathbb{R}^K \rightarrow \mathbb{R}$ with respect to a column vector: $\frac{\partial}{\partial \mathbf{x}} f(\mathbf{x})$. Note that as $f(\mathbf{x})$ is a function of the *vector* \mathbf{x} , it can instead be written as $f(x_1, x_2, \dots, x_K)$ if \mathbf{x} has K components. It is also a *scalar function*, i.e. it associates a single number with the input vector \mathbf{x} . We define the derivative with respect to the column vector \mathbf{x} as:

$$\frac{\partial}{\partial \mathbf{x}} f(\mathbf{x}) = \begin{bmatrix} \frac{\partial}{\partial x_1} f(\mathbf{x}) \\ \frac{\partial}{\partial x_2} f(\mathbf{x}) \\ \vdots \\ \frac{\partial}{\partial x_K} f(\mathbf{x}) \end{bmatrix} \quad (\text{A.2.2})$$

and w.r.t. the row vector as

$$\frac{\partial}{\partial \mathbf{x}'} f(\mathbf{x}) = \left[\frac{\partial}{\partial x_1} f(\mathbf{x}) \quad \frac{\partial}{\partial x_2} f(\mathbf{x}) \quad \dots \quad \frac{\partial}{\partial x_K} f(\mathbf{x}) \right]. \quad (\text{A.2.3})$$

So we simply take the partial derivatives of the function with respect to all individual components of the vector, and stack the results in a vector of the same shape. So derivative of a scalar function with respect to a vector is a vector of similar shape. It's not too bad so far.

This rule has a nice application. In case of both \mathbf{x} and $\boldsymbol{\beta}$ are $K \times 1$ column vectors, $\mathbf{x}^\top \boldsymbol{\beta} = \boldsymbol{\beta}^\top \mathbf{x} = \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_K x_K$ is a scalar. Hence

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{x}^\top \boldsymbol{\beta} = \frac{\partial}{\partial \mathbf{x}} \boldsymbol{\beta}^\top \mathbf{x} = \begin{bmatrix} \frac{\partial}{\partial x_1} (\beta_1 x_1 + \beta_2 x_2 + \dots + \beta_K x_K) \\ \frac{\partial}{\partial x_2} (\beta_1 x_1 + \beta_2 x_2 + \dots + \beta_K x_K) \\ \vdots \\ \frac{\partial}{\partial x_K} (\beta_1 x_1 + \beta_2 x_2 + \dots + \beta_K x_K) \end{bmatrix} = \boldsymbol{\beta}. \quad (\text{A.2.4})$$

This is very similar to the ordinary calculus where $\frac{\partial}{\partial x} x \cdot \boldsymbol{\beta} = \boldsymbol{\beta}$.

Things get more complex if we want to differentiate a *vector function* w.r.t a vector. Let's stay with the cases that are easier to represent: derivative of a column vector function w.r.t a row vector, and the way around. A vector function is a function that associates a vector with each argument value. Let's look at a function $\mathbf{f} : \mathbb{R}^K \rightarrow \mathbb{R}^N$, i.e. it associates a N -dimensional vector with each K -dimensional argument. The concept of vector function is simply a shorthand of writing

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_N(\mathbf{x}) \end{pmatrix} \quad (\text{A.2.5})$$

i.e. it is a suitably stacked collection of N scalar functions of a vector arguments, which in turn, can be written with no vector notation at all as

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} f_1(x_1, x_2, \dots, x_K) \\ f_2(x_1, x_2, \dots, x_K) \\ \vdots \\ f_N(x_1, x_2, \dots, x_K) \end{pmatrix}. \quad (\text{A.2.6})$$

Now we have to take a derivative of this stack of functions w.r.t the row vector \mathbf{x}^\top . We just take each individual (vertical) component, differentiate it as in (A.2.3), and stack the resulting row vectors vertically. This gives us a matrix:

$$\frac{\partial}{\partial \mathbf{x}^\top} \mathbf{f}(\mathbf{x}) \begin{bmatrix} \frac{\partial}{\partial x_1} f_1(\mathbf{x}) & \frac{\partial}{\partial x_2} f_1(\mathbf{x}) & \dots & \frac{\partial}{\partial x_K} f_1(\mathbf{x}) \\ \frac{\partial}{\partial x_1} f_2(\mathbf{x}) & \frac{\partial}{\partial x_2} f_2(\mathbf{x}) & \dots & \frac{\partial}{\partial x_K} f_2(\mathbf{x}) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial}{\partial x_1} f_N(\mathbf{x}) & \frac{\partial}{\partial x_2} f_N(\mathbf{x}) & \dots & \frac{\partial}{\partial x_K} f_N(\mathbf{x}) \end{bmatrix}. \quad (\text{A.2.7})$$

So the derivative of N -dimensional column vector w.r.t K dimensional row vector is a $N \times K$ matrix. This is a nice result that can be used in several applications. If \mathbf{A} is a $N \times K$ matrix

$$\frac{\partial \mathbf{A}\mathbf{x}}{\partial \mathbf{x}'} = \mathbf{A} \quad \text{and} \quad \frac{\partial \mathbf{x}'\mathbf{A}}{\partial \mathbf{x}} = \mathbf{A}. \quad (\text{A.2.8})$$

(You simply have to write down the definition of $\mathbf{A}\mathbf{x}$, and use (A.2.3) to get the result).

Additional useful results without proofs:

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{x}^\top = \begin{bmatrix} \frac{\partial x_1}{\partial x_1} & \frac{\partial x_1}{\partial x_2} & \cdots & \frac{\partial x_1}{\partial x_K} \\ \frac{\partial x_2}{\partial x_1} & \frac{\partial x_2}{\partial x_2} & \cdots & \frac{\partial x_2}{\partial x_K} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial x_K}{\partial x_1} & \frac{\partial x_K}{\partial x_2} & \cdots & \frac{\partial x_K}{\partial x_K} \end{bmatrix} = \mathbf{I} \quad \frac{\partial}{\partial \mathbf{x}^\top} \mathbf{x} = \mathbf{I} \quad (\text{A.2.9})$$

$$\frac{\partial \mathbf{A}\mathbf{x}}{\partial \mathbf{x}^\top} = \mathbf{A} \quad \frac{\partial \mathbf{x}^\top \mathbf{A}}{\partial \mathbf{x}} = \mathbf{A} \quad (\text{A.2.10})$$

$$\frac{\partial \mathbf{x}^\top \mathbf{x}}{\partial \mathbf{x}} = \mathbf{x} + \mathbf{x}^\top = 2\mathbf{x} \quad \frac{\partial \mathbf{x}^\top \mathbf{A}\mathbf{x}}{\partial \mathbf{x}} = (\mathbf{A} + \mathbf{A}^\top)\mathbf{x}. \quad (\text{A.2.11})$$

All these results can be proven by using the definition of matrix multiplication, and the differentiation wrt vector (A.2.2) and (A.2.3).

TBD: chain rule

TBD: $f(\mathbf{x})^T \cdot f(\mathbf{x})$

A.2.1 Gradient

Prerequisites: Calculus, and a basic understanding of multivariate calculus

What is Gradient

Gradient is generalization of derivative for functions on \mathbb{R}^n . While the derivative describes the slope of the function in 1-dimensional case, gradient indicates both the slopes (along different axes) and the direction of the steepest ascent for functions of n variables (functions on \mathbb{R}^n). Below, we only look at the scalar functions $\mathbb{R}^n \rightarrow \mathbb{R}$, i.e. the function take an n -dimensional input but return a scalar value.

Perhaps the easiest way to understand this is to think about a hilly landscape. Elevation is a function $\mathbb{R}^2 \rightarrow \mathbb{R}$: from two inputs (longitude and latitude) to a single number (elevation). Gradient tells at which rate the ground rises, and where is the direction of the steepest climb.

Let us take a simple example

$$f(\mathbf{x}) \equiv f(x_1, x_2) = x_1 \cdot \log x_2 \quad (\text{A.2.12})$$

where \mathbf{x} is the 2-dimensional input vector $(x_1, x_2)'$. Two-parameter functions can easily be visualized as surfaces, $f(\mathbf{x})$ is depicted in Figure A.1.

For this function, the partial derivatives are $\frac{\partial}{\partial x_1} f(\mathbf{x}) = \log x_2$ and $\frac{\partial}{\partial x_2} f(\mathbf{x}) = x_1/x_2$. In a way, gradient, commonly denoted by $\nabla f(\mathbf{x})$ or sometimes $\partial f(\mathbf{x})/\partial \mathbf{x}$, is just a compact way to write this in vector form:

$$\nabla f(\mathbf{x}) = \begin{pmatrix} \log x_2 \\ x_1/x_2 \end{pmatrix}. \quad (\text{A.2.13})$$

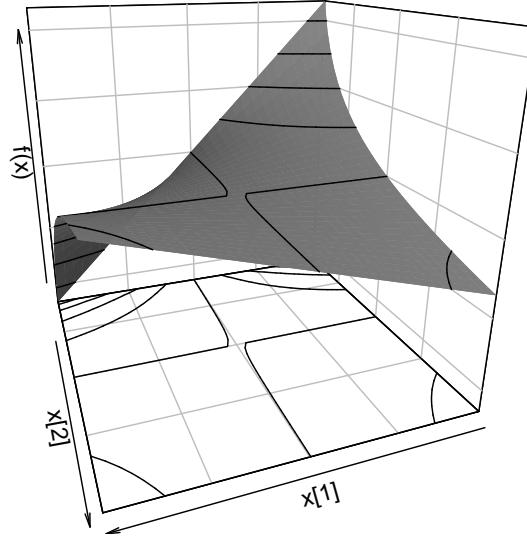


Figure A.1: $f(\mathbf{x}) = x_1 \cdot \log x_2$ as function of x_1 and x_2 . The level sets, contours of equal values, are plotted both on the surface and on bottom of the figure box.

This is a 2×1 vector. So gradient is just a habit to stack the partial derivatives into a vector. Compared to the function itself, gradient is harder to visualize as it has two values (it's range is in \mathbb{R}^2). Figure A.2 shows two options for visualizing $\nabla f(\mathbf{x})$.

In case of n -dimensional argument functions $\mathbb{R}^n \rightarrow \mathbb{R}$, we have n partial derivatives $\frac{\partial}{\partial x_1} f(\mathbf{x}), \frac{\partial}{\partial x_2} f(\mathbf{x}), \dots, \frac{\partial}{\partial x_n} f(\mathbf{x})$ and we stack these into the gradient as

$$\nabla f(\mathbf{x}) = \begin{pmatrix} \frac{\partial}{\partial x_1} f(\mathbf{x}) \\ \frac{\partial}{\partial x_2} f(\mathbf{x}) \\ \vdots \\ \frac{\partial}{\partial x_n} f(\mathbf{x}) \end{pmatrix} \quad (\text{A.2.14})$$

so the gradient is $n \times 1$ vector. Note that gradient is a function too, although not a scalar valued one but a vector valued function $\mathbb{R}^n \rightarrow \mathbb{R}^n$: it associates each n -dimensional argument value \mathbf{x} to a n -dimensional vector $\nabla f(\mathbf{x})$. This is analogous with the derivative in one-dimensional case $\mathbb{R} \rightarrow \mathbb{R}$, that one is also an one-dimensional function $\mathbb{R} \rightarrow \mathbb{R}$.

While gradient itself is a function, when we calculate its value for any particular argument of \mathbf{x} , the result will be a vector (not function). This is exactly analogous to the ordinary derivative, which is a function but when calculated at a particular x value it is a number.

As an example, let's take the function $f(\mathbf{x})$ we defined above and let's calculate

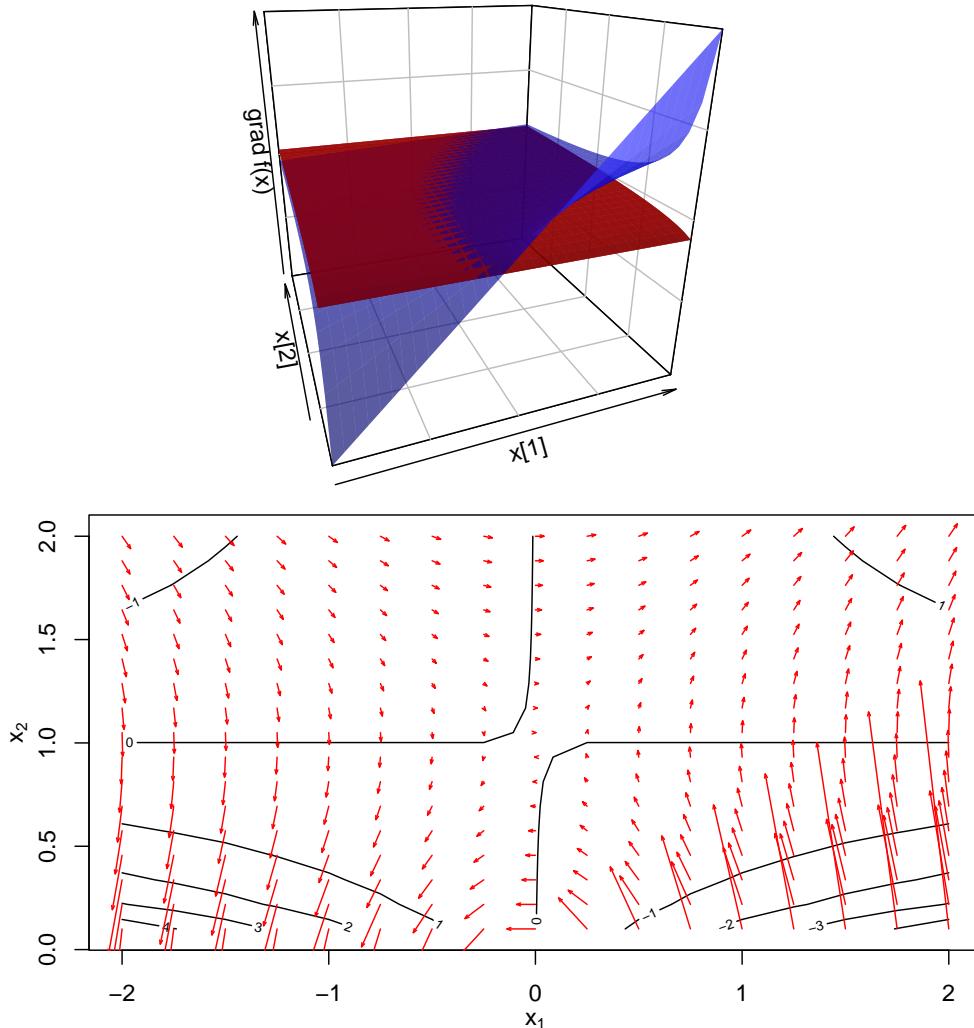


Figure A.2: Gradient of $f(\mathbf{x})$, depicted as two surfaces (upper panel). The blue surface corresponds to $\frac{\partial}{\partial x_2} f(\mathbf{x}) = x_1/x_2$, the red one to $\frac{\partial}{\partial x_1} f(\mathbf{x}) = \log x_2$. The lower panel depicts the gradient as arrows plotted on the levels (contours) of the function. The length of the arrows is proportional to the gradient length, their direction is equal to the gradient direction. One can easily see that the norm of gradient is proportional to the steepness of the function surface, and gradient points to the direction of the steepest climb.

it's value, and it's gradient's value at $\mathbf{x} = (1, 2)'$:

$$f(\mathbf{x})|_{\mathbf{x}=(1,2)'} = 1 \cdot \log 2 \approx 0.693. \quad (\text{A.2.15})$$

This is seldom written in such a long way, almost all texts use a shorter but somewhat misleading version of $f((1,2)')$, or just $f(1,2)$ instead. A similar notation applies to gradient. It's value at $(1,2)'$ can be written as

$$\nabla g(\mathbf{x})|_{\mathbf{x}=(1,2)'} = \begin{pmatrix} \log x_2 \\ x_1/x_2 \end{pmatrix} \Big|_{\mathbf{x}=(1,2)'} = \begin{pmatrix} \log 2 \\ 1/2 \end{pmatrix}. \quad (\text{A.2.16})$$

As before, this value is often written in shorter but imprecise way as

$$\nabla f \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} \log 2 \\ 1/2 \end{pmatrix}. \quad (\text{A.2.17})$$

It is imprecise because $f \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ is a constant, $\log 2$, and gradient of a constant is always 0. However, it is a widely used shortcut in the literature. It is important to distinguish between gradient of a function, calculated at a fixed argument value; and between a function, computed at the same fixed argument values. When using the shortcut notation we have to understand that $\begin{pmatrix} 1 \\ 2 \end{pmatrix}$ is the argument, gradient is computed with respect of the argument, and afterwards evaluated at this value of the argument.

However, it appears gradient is much more than a handy way to write a number of derivatives in a compact form. It naturally generalizes a number of properties of 1-dimensional derivatives.

Gradient and Direction of Steepest Ascent

As components of gradient are just ordinary partial derivatives, each component indicates the slope of the function along that axis: how much will the function value grow if we move along the axis, while keeping the location on the other axes constant. If one component is large and another is small, we have a steep hill in the first direction while it is pretty flat along the other axis.

Let's look at linear functions, simple even surfaces with no curvature whatsoever. A linear function may look something like the plane depicted on Figure A.3. If the surface is rising rapidly along x_2 while staying constant along x_1 , the direction of fastest climb is just along x_2 . Analogously, if the function grows along x_1 while staying flat along x_2 , we have to move toward x_1 . Such a situation is depicted on Figure A.4 although here the first gradient component $\partial f(\mathbf{x})/\partial x_1 < 0$ and hence we have to move toward smaller values of x_1 instead if we want to climb uphill. Obviously, if none of the gradient components are zero, we have to move somewhere in-between of these two directions. This is shown on Figure A.5. It is also intuitive, that the "somewhere in-between" should be closer to the steeper gradient than to the smaller gradient component.

It is easy to show that the exact direction of the steepest climb is the same as the direction of the gradient vector. Let's choose a point (x_1, x_2) where the function's

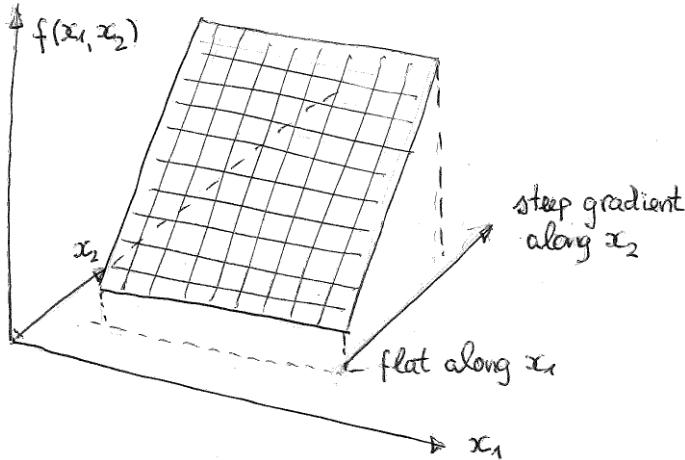


Figure A.3: Function increasing along x_2 while constant along x_1 .

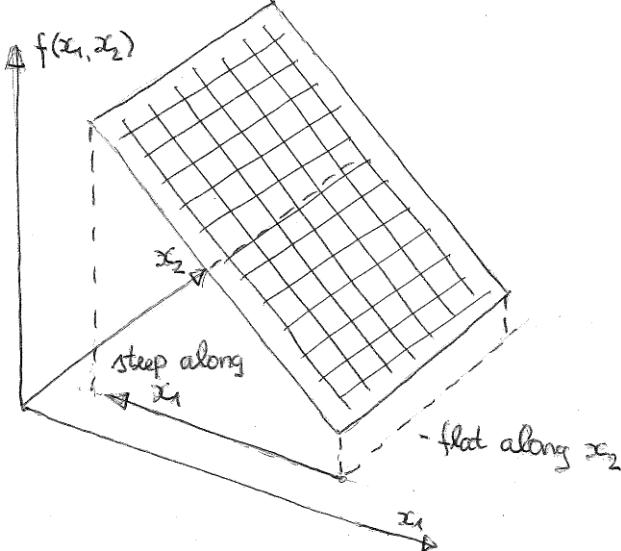


Figure A.4: Function increasing along $-x_1$ while constant along x_2 .

value is $f(x_1, x_2)$. Now move away from this point by $(\Delta x_1, \Delta x_2)$. This causes the function to grow by

$$\Delta f \equiv f(x_1 + \Delta x_1, x_2 + \Delta x_2) - f(x_1, x_2) \approx g_1 \cdot \Delta x_1 + g_2 \cdot \Delta x_2 \quad (\text{A.2.18})$$

where g_1 and g_2 are the corresponding gradient components, calculated at (x_1, x_2) . However, for not to go too much wild, we'll change the coordinates in this way that the total move will be of length one. Hence $\Delta x_1^2 + \Delta x_2^2 = 1$, or alternatively $\Delta x_2 =$

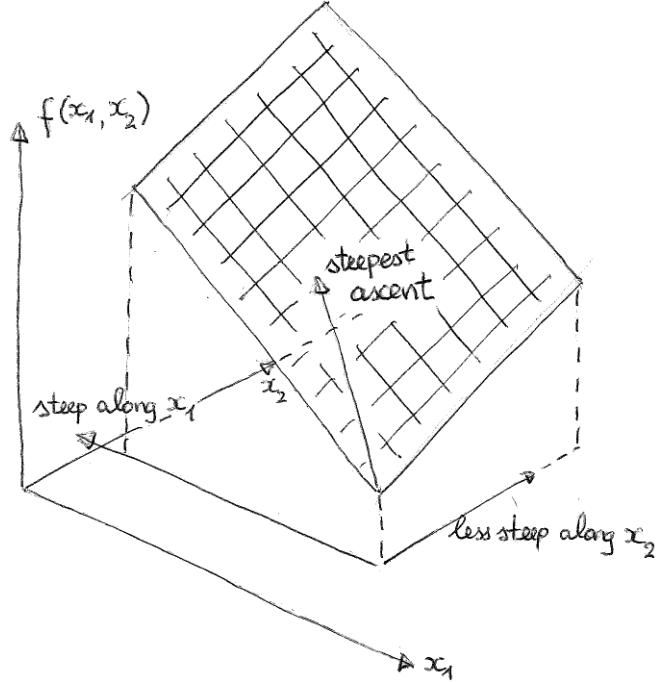


Figure A.5: Function increasing both $-x_1$ and x_2 . The direction of steepest climb is somewhere between these two gradient components.

$\sqrt{1 - \Delta x_1^2}$, and the change of the function can be written as

$$\Delta f \approx g_1 \cdot \Delta x_1 + g_2 \cdot \Delta x_2 \quad \text{or} \quad \Delta f \approx g_1 \cdot \Delta x_1 + g_2 \cdot \sqrt{1 - \Delta x_1^2}. \quad (\text{A.2.19})$$

Which Δx_1 results in the largest value of Δf ? The optimality condition gives us

$$\frac{\partial \Delta f}{\partial \Delta x_1} = g_1 + g_2 \frac{-2\Delta x_1}{2\sqrt{1 - \Delta x_1^2}} = g_1 - g_2 \frac{\Delta x_1}{\Delta x_2} = 0 \quad (\text{A.2.20})$$

where we used the fact that $\sqrt{1 - \Delta x_1^2} = \Delta x_2$. The solution is

$$\frac{\Delta x_1}{\Delta x_2} = \frac{g_1}{g_2}. \quad (\text{A.2.21})$$

In other words, this means that the direction vector $(\Delta x_1, \Delta x_2)$ must be parallel to the gradient vector (g_1, g_2) .

Appendix B

Datasets

Datasets used in this book originate from various sources. Some are copied from R packages, others are scraped by me. R packages are typically used as-is (e.g. *SmokeBan* from *AER* package), but others are provided as CSV files in the [book's repo](#). This appendix gives a brief overview of these.

Boston housing This is a popular dataset for machine learning, available from various sources. Version here, [boston.csv.bz2](#) is copied from R's *MASS* package, but it is identical to other versions. It has 506 rows, 14 numeric variables and no missings. Each row contains data for one neighborhood (town/tract). The central variable is to be analyzed is typically *medv*, median value of single-family homes in that neighborhood. Variables:

crim per capita crime rate by town.
zn proportion of residential land zoned for lots over 25,000 sq.ft.
indus proportion of non-retail business acres per town.
chas Charles River dummy variable (= 1 if tract bounds river; 0 otherwise).
nox nitrogen oxides concentration (parts per 10 million).
rm average number of rooms per dwelling.
age proportion of owner-occupied units built prior to 1940.
dis weighted mean of distances to five Boston employment centres.
rad index of accessibility to radial highways.
tax full-value property-tax rate per \$10,000.
ptratio pupil-teacher ratio by town.
black $1000(B_k - 0.63)^2$ where B_k is the proportion of blacks by town.
lstat lower status population (percent)
medv median value of owner-occupied homes in \$1000s.

Males [males.csv.bz2](#) originates from R package *Ecdat*. It is a subset of NSLY panel that contains 4360 observations for 545 young men in the U.S. from 1980 to 1987. The variables are:

nr identifier
year year

school years of schooling
exper years of experience (=age-6-school)
union wage set by collective bargaining ?
ethn a factor with levels (*black, hisp, other*)
married married ?
health health problem ?
wage log of hourly wage
industry a factor with 12 levels
occupation a factor with 9 levels
residence a factor with levels (*rural area, north east, northern central, south*)

Smoke ban Included in *AER* R package. A dataset of 10000 and 7 variables. It is a subset of 1991 National Health Survey.

smoker factor. Is the individual a current smoker?

ban factor. Is there a work area smoking ban?

age age in years.

education factor indicating highest education level attained: high school (hs) drop out, high school graduate, some college, college graduate, master's degree (or higher).

afam factor. Is the individual African-American?

hispanic factor. Is the individual Hispanic?

gender factor indicating gender.

Treatment [treatment.csv](#) included from R package *Ecdat*. A U.S. dataset from 1974, used for evaluating treatment effect of training on earnings.

treat treated (TRUE/FALSE)

age age

educ education in years

ethn three categories: "other", "black", "hispanic"

married married (TRUE/FALSE)

re74 real annual earnings in 1974 (pre-treatment)

re75 real annual earnings in 1975 (pre-treatment)

re78 real annual earnings in 1978 (post-treatment)

u74 unemployed in 1974 (TRUE/FALSE)

u75 unemployed in 1975 (TRUE/FALSE)

Appendix C

Exercise Solutions

C.1 Introduction to Statistics

Solution (1.2). We have data $x = (1, 2, 3, 3, 3, 5, 5, 10)$.

1. Mean is

$$\bar{x} = (1 + 2 + 3 + 3 + 3 + 5 + 5 + 10)/8 = 32/8 = 4$$

2. Median is 3 as the “middle” of the data is between two “3”-s (there are three numbers smaller than 3 and three numbers larger than 3).

3. mode is 3 as this is the most frequent number.

If one data point is missing, we cannot compute mean. For median, we can tell that it must be between 3 and 5: if the missing data point is smaller than 3, the median is still 3. If it is larger than 5, there are 3 numbers no larger than 3 and 3 numbers no smaller than 5 in the data, and hence median is between 3 and 5 (potentially equal to either 3 or 5). So we have bounds on the median. The mode will be either 3 (if the missing value is not 5), or it is a bimodal dataset with modes both 3 and 5 (if the missing value is 5).

Solution (1.1). • Talent show result: this is clearly ordinal measure: we can say that first place is better than second, or 7th place is better than 8th; but the difference between 1st and 2nd, and 7th and 8th is undetermined. We can order the results, but their difference does not mean anything.

- Height in cm: this is ratio. Height differences are well defined and height has a well-defined zero.
- Height in feet, inches. This is ratio as well. It is measured in a different way than in case of cm, but it is height nevertheless with the same properties.
- Colors by name: this is a nominal measure. Colors do not have any inherent order, humans have invented many different orderings and all of those are equally valid.
- Temperature in C: this is an interval measure: temperature difference makes much sense (“*today is 10 degrees warmer...*”) but the zero is fairly arbitrary (“*today is twice as warm*” does not tell much).

- IMDB movie ratings: movie ratings are ordered measures. The order is well defined, but the difference does not carry much real meaning: the movies rated 7 and 7.5, and movies rates 9 and 9.5 may not differ by equal amount (whatever it means).

Solution (1.3). It is convenient to order the sample first. We get (1,1,1,2,2,3).

1. 0.5-th quantile: the lower 50% of the sample is (1,1,1) and the upper 50% is (2,2,3). Hence any number in the interval [1,2] will be no smaller than the lower half, and no larger than the upper half. We can pick 1.5, a value in the middle of the interval.
2. 0.25-th quantile: when we order six values, then 0.25th quantile is at “position 1.5”, i.e. overlapping with the second-lowest value. In this example it is 1, the lowest 25% of values are no larger than 1, and the largest 75% of values are no smaller than 1.
3. 0.8-th quantile: in a similar fashion, the 0.8-th quantile is “inside” of the next-to-last value, 2. So 0.8-th quantile is 2.

Solution (1.4). The sample space of the problem in simple events is

		Die 2						
		1	2	3	4	5	6	
Die 1		1	(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)
		2	(2,1)	(2,2)	(2,3)	(2,4)	(2,5)	(2,6)
3	(3,1)	(3,2)	(3,3)	(3,4)	(3,5)	(3,6)		
4	(4,1)	(4,2)	(4,3)	(4,4)	(4,5)	(4,6)		
5	(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	(5,6)		
6	(6,1)	(6,2)	(6,3)	(6,4)	(6,5)	(6,6)		

where we have marked the simple events of interest that correspond to the compound event in blue. All these events are equally likely ($1/36$) as the dies are independent and fair.

So the compound event of interest is made of 11 simple events, all of equal probability of $1/36$, and hence its probability is $11/36$.

Solution (1.5). 1. The conditioning event is traveling in the 1st class. From the table, we see that there were $200 + 123 = 323$ 1st class passengers, out of whom 200 survived. Hence $\Pr(\text{survived} | \text{traveled in 1st class}) = 200/323 = 0.619$.

2. Now the conditioning event is survival. We have $200 + 119 + 181 = 500$ survivors, out of whom 200 traveled in the 1st class. Hence $\Pr(\text{traveled in 1st class} | \text{survived}) = 200/500 = 0.4$.

Solution (1.6). We can denote the simple events regarding the gender as (g_1, g_2) where g_1 means the gender of the first child and g_2 the gender of the second child. The sample space contains 4 simple events: $(G, G), (G, B), (B, G), (B, B)$ where G

and B denote that the corresponding child is a girl/boy. All these simple events are equally likely, and have probability $1/4$.

The event of interest, *the other one is also a girl*, corresponds to the event (G, G) . The conditioning event *one of them is a girl* removes the last option, (B, B) from considerations. Hence we are left with one event of interest out of 3 possible events, all of which have probability $1/4$. The conditional probability (from (1.3.1)) is

$$p = \frac{1/4}{3/4} = \frac{1}{3}.$$

Alternatively, we can think about 100 families with two children. 25 of them are (G, G) , 25 are (G, B) , 25 are (B, G) and 25 are of “type” (B, B) . Here 75 families fit the description of having a daughter, and $25/75 = 1/3$ of them have two daughters.

This is a problem where clear understanding of the concepts of events and sample space is extremely helpful.

Solution (1.7). First, note that conditional probability is not related to causality. The fact that someone survived did not make her more or less likely to have been in first class. We can imagine this is an answer to a question: “Take all Titanic survivors. Pick a random survivor. What is the probability she was in first class”?

This is a simple task employing Bayes theorem, where we have denoted $\Pr(S = 1|C = 1) = 0.619$, $\Pr(C = 1) = 0.247$, and $\Pr(S = 1) = 0.382$. Hence the probability of interest

$$\Pr(C = 1|S = 1) = \frac{\Pr(S = 1|C = 1) \cdot \Pr(C = 1)}{\Pr(S = 1)} = \frac{0.619 \cdot 0.247}{0.382} = 0.400.$$

So 40% of survivors were first class passengers.

If one has access to the actual numbers, it is easy to check: there were 200 first class passengers among 500 survivors.

Solution (1.8). We have $\Pr(A) = 0.5$ because two types of bags are equally likely. We also know that $\Pr(\text{Red}|A) = 2/3$ and $\Pr(\text{Red}|B) = 1/3$.

1. From Bayes theorem

$$\Pr(A|\text{Red}) = \frac{\Pr(\text{Red}|A) \cdot \Pr(A)}{\Pr(\text{Red})}.$$

We can compute

$$\Pr(\text{Red}) = \Pr(\text{Red}|A) \cdot \Pr(A) + \Pr(\text{Red}|B) \cdot \Pr(B) = 2/3 \cdot 1/2 + 1/3 \cdot 1/2 = 1/2.$$

Plugging this into the expression above, we have

$$\Pr(A|\text{Red}) = \frac{2/3 \cdot 1/2}{1/2} = 2/3.$$

So pulling a red candy out of the bag makes it more likely it is an A -bag, but it is by no means certain.

2. Now she pulls out two red candies. The solution is similar as above, just the event of interest is not *Red*, but *Red, Red*; and we have to adjust the probabilities accordingly. From Bayes' theorem, we have

$$\Pr(B|Red, Red) = \frac{\Pr(Red, Red|B) \cdot \Pr(B)}{\Pr(Red, Red)}.$$

Now we need to compute

$$\begin{aligned} \Pr(Red, Red) &= \Pr(Red|A) \cdot \Pr(A) + \Pr(Red, Red|B) \cdot \Pr(B) = \\ &= (2/3)^2 \cdot 1/2 + (1/3)^2 \cdot 1/2 = 5/18. \end{aligned}$$

The probabilities of the compound event *Red, Red*, $(2/3)^2$ for the bag *A* and $(1/3)^2$ for the bag *B*, assume that the events are independent. Here it means that removing one candy does not alter the probabilities of the remaining candies in the bag. This is (approximately) true if the bags are large.

Plugging this into the expression above, we have

$$\Pr(A|Red, Red) = \frac{(1/3)^2 \cdot 1/2}{5/18} = 1/5$$

So when she pulls out two red M&M-s, it is not that likely that it is an *A*-bag. But 20% is still probability we should not ignore.

Solution (1.9). We can use Bayes theorem to find

$$\Pr(\text{lion|steps}) = \frac{\Pr(\text{steps|lion}) \cdot \Pr(\text{lion})}{\Pr(\text{steps})}. \quad (\text{C.1.1})$$

Before we can do this, we need to compute $\Pr(\text{steps})$. As $\Pr(\text{neighbor}) = 0.9$, we have that $\Pr(\text{lion}) = 0.1$, and now

$$\begin{aligned} \Pr(\text{steps}) &= \Pr(\text{steps|lion}) \cdot \Pr(\text{lion}) + \Pr(\text{steps|neighbor}) \cdot \Pr(\text{neighbor}) = \\ &= 0.6 \cdot 0.1 + 0.2 \cdot 0.9 = 0.24. \quad (\text{C.1.2}) \end{aligned}$$

Plugging this into (C.1.1) above we get

$$\Pr(\text{lion|steps}) = \frac{0.6 \cdot 0.1}{0.24} = 0.25.$$

So the probability that the noise is made by lion is 25%. But should you smile or should you fight? I would probably grab a burning stick and be ready to fight—if it turns out my neighbor, it is embarrassing. But if I sit and smile, and a lion suddenly jumps out of the shadows, then it was my last smile.

Solution (1.10). Let 0 correspond to the case where there were no 6-s on the dice, and 1 if there were at least one 6. We can write the RV as

$$X = \begin{cases} 1 & \text{if } X \in \{(1,6), (2,6), (3,6), (4,6), (5,6), (6,6), \\ & (6,5), (6,4), (6,3), (6,2), (6,1)\} \\ 0 & \text{otherwise} \end{cases} \quad (\text{C.1.3})$$

Solution (1.11). Here is the 6×6 table of all possible outcomes with sum 6 highlighted in blue:

		Sum of two dies					
		Die 2					
		1	2	3	4	5	6
Die 1	1	2	3	4	5	6	7
	2	3	4	5	6	7	8
	3	4	5	6	7	8	9
	4	5	6	7	8	9	10
	5	6	7	8	9	10	11
	6	7	8	9	10	11	12

There are clearly 5 ways to get sum 6, hence the corresponding probability $\Pr(Z = 6) = 5/36$.

Solution (1.12). If the die is fair, all sides have probability $1/6$. Hence the expected value

$$\mathbb{E} D = \frac{1}{6} \cdot 1 + \frac{1}{6} \cdot 2 + \dots + \frac{1}{6} \cdot 6 = \frac{1}{6}(1 + 2 + \dots + 6) = 3.5$$

Solution (1.13). 1. The easiest way to prove the probabilities is to present the sample space as a 6×6 table where we list the number of sixes:

		Die 2					
		1	2	3	4	5	6
Die 1	1	0	0	0	0	0	1
	2	0	0	0	0	0	1
	3	0	0	0	0	0	1
	4	0	0	0	0	0	1
	5	0	0	0	0	0	1
	6	1	1	1	1	1	2

One can easily see that out of the 36 cases, in 25 we have no sixes, in 10 cases we have a single six, and in one case we have two sixes.

2. Using the definition (1.3.13) we have

$$\mathbb{E} X = 25/36 \cdot 0 + 10/36 \cdot 1 + 1/36 \cdot 2 = 12/36 = 1/3.$$

Solution (1.14). a) First we have to find the expected value: $\mathbb{E} X = 0.25 \cdot (-1) + 0.5 \cdot 0 + 0.25 \cdot 1 = 0$. It is also immediately obvious as the values are symmetric around 0. Next, let us do a table, similar to Table 1.4:

x	$\Pr(X = x)$	$X - \mathbb{E} X$	$(X - \mathbb{E} X)^2$
-1	0.25	-1	1
0	0.50	0	0
1	0.25	1	1

Note that as $EX = 0$, the columns 1 and 3 are the same. Variance, the expected value of the last column is $\text{Var } X = \mathbb{E}(X - \mathbb{E} X)^2 = 0.25 \cdot 1 + 0.25 \cdot 1 = 0.5$.

b) We already know $\mathbb{E} X$, so we have to find $\mathbb{E} X^2$: $\mathbb{E} X^2 = 0.25 \cdot (-1)^2 + 0.5 \cdot 0^2 + 0.25 \cdot 1^2 = 0.5$. The variance is $\text{Var } X = \mathbb{E} X^2 - (\mathbb{E} X)^2 = 0.5 - 0 = 0.5$.

As the example demonstrates, in case of $\mathbb{E} X = 0$, the variance is equal to just $\mathbb{E} X^2$.

Solution (1.15). First let's use the **variance definition** (1.3.17). Write a similar extended table as Table 1.2 on page 11:

1	2	3	4	
x	$\Pr(X = x)$	$x - \mathbb{E} X$	$(x - \mathbb{E} X)^2$	x^2
0	$1 - p$	$-p$	p^2	
1	p	$1 - p$	$(1 - p)^2$	

From the first two columns we can immediately see that the expected value is

$$\mathbb{E} X = (1 - p) \cdot 0 + p \cdot 1 = p.$$

Now we can compute the deviations $-p$ and $1 - p$ in the 3rd column; and deviations-squared p^2 and $(1 - p)^2$ in the 4th column. Variance is the expected value of the last column:

$$\text{Var } X = (1 - p) \cdot p^2 + p \cdot (1 - p)^2 = p(1 - p).$$

When using the **shortcut formula** (1.3.18), we first need to compute $\mathbb{E} X^2$. As the possible values are just 0 and 1, the squares of the values are the same, and hence $\mathbb{E} X^2 = p$, the same number as $\mathbb{E} X$. Now we have

$$\text{Var } X = \mathbb{E} X^2 - (\mathbb{E} X)^2 = p - p^2 = p(1 - p).$$

TBD: continuous case

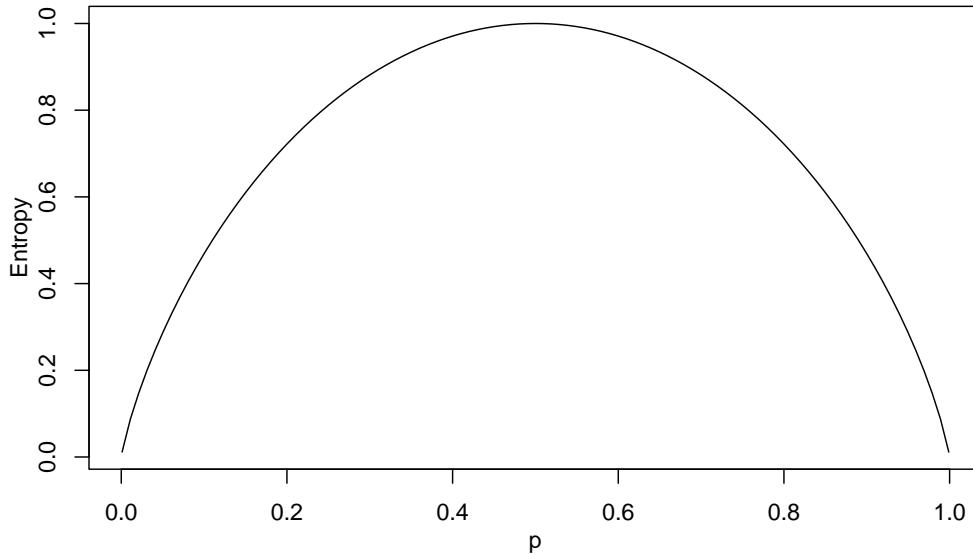
Solution (1.16). X must be ordinal for the comparison in (1.4.3) to have a meaning.

Solution (1.17). Standard uniform as specified in (1.4.16) has density 1 in the interval $[0,1]$. So all values between 0 and 1 are equally likely, and values outside of this interval are impossible. Hence the lower 2.5% quantile is the lowest 2.5% end of this interval, $q_{0.025} = 0.025$ and upper $q_{0.975} = 0.975$.

Solution (1.18). Bernoulli distribution has two states: the event happens with probability p and does not happen with probability $1 - p$. Inserting this into the definition of entropy (1.4.23), we get

$$\mathbb{H}(X) = -p \log_2 p - (1 - p) \log_2 (1 - p). \quad (\text{C.1.4})$$

The entropy as a function of p will look like



The entropy is 0 at both ends of the curve: we are (almost) certain the event either does not happen ($p = 0$) or happens ($p = 1$), and hence there is no uncertainty. The largest uncertainty is in the middle where both outcomes are equally likely, and we can gain 1 bit of information.

Solution (1.19). The naive answer would be to place armor in the most damaged parts of the airplanes. However, note that we face a missing data problem here: we can only observe those places that actually return to the base. We don't know where were those planes hit that did not return.

However, we can still guess: as anti-aircraft fire is very imprecise, there is no reason to believe that engines and cockpits were not hit. The fact that we do not see much damage in those parts hints that those are the weakest points. If engines or cockpit are hit, the plane won't return. Hence you should recommend to armor those areas that are *not damaged*!

Solution (1.20). According to media, the net worth of Bill Gates is \$105 billion, almost three times the total wealth of Iceland. Hence an option would be to grant Bill Gates Iceland citizenship and in this way to make him an “Icelander”. This will make the average wealth of Icelanders to grow from \$95,000 to \$370,000 per person.

The problem is the meaning of the expression “all Icelanders”. People understand it intuitively that it applies to everyone (everyone individually), but here it is (most likely deliberately) used in a different sense, something like “everyone combined together”.

C.2 Linear Algebra

Solution (2.1). Dimension is just the number of components in the vector. Hence \mathbf{v}_i is of dimension 8. The table of data itself does not reveal how long are \mathbf{x}_i -s, but as

the data is about “50 U.S. States”, its dimension must be 50.

Solution (2.2). We can compute individual components as $e(\text{Berlin})_1 - e(\text{Germany})_1 + e(\text{France})_1 = -0.562 - 0.194 + 0.605 = -0.151$, $e(\text{Berlin})_2 - e(\text{Germany})_2 + e(\text{France})_2 = 0.630 - 0.507 - 0.678 = -0.555$, and for the following 3 components we have -1.176 , -0.450 , and -0.016 . So the vector

$$e(\text{Berlin}) - e(\text{Germany}) + e(\text{France}) = (-0.151, -0.555, -1.176, -0.450, -0.016)$$

while

$$e(\text{Paris}) = (-0.074, -0.855, -0.689, -0.057, -0.139)$$

As one can see, the result is not exact, but broadly agrees in terms of size and sign of the components, unlike any other word listed here.

Solution (2.3). As in case of Example 2.3, we can express

$$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = 2 \cdot \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix} - \begin{pmatrix} 7 \\ 8 \\ 9 \end{pmatrix}, \quad (\text{C.2.1})$$

or

$$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} - 2 \cdot \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix} + \begin{pmatrix} 7 \\ 8 \\ 9 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \quad (\text{C.2.2})$$

Hence these vectors are not linearly independent.

Solution (2.4). The first product:

$$\begin{aligned} \left[\begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ -1 & 2 & 3 \end{pmatrix} \cdot \begin{pmatrix} -1 & 1 \\ 1 & -1 \\ -1 & 1 \end{pmatrix} \right] \cdot \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} &= \begin{pmatrix} -2 & 2 \\ -2 & 2 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} = \\ &= \begin{pmatrix} 2 & -2 & 2 \\ 2 & -2 & 2 \end{pmatrix}. \quad (\text{C.2.3}) \end{aligned}$$

The second product:

$$\begin{aligned} \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ -1 & 2 & 3 \end{pmatrix} \cdot \left[\begin{pmatrix} -1 & 1 \\ 1 & -1 \\ -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \right] &= \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} \cdot \begin{pmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{pmatrix} = \\ &= \begin{pmatrix} 2 & -2 & 2 \\ 2 & -2 & 2 \\ 2 & -2 & 2 \end{pmatrix}. \quad (\text{C.2.4}) \end{aligned}$$

These are indeed equal.

Solution (2.6). As we are in 2-D space, we can write (2.2.5) using components as

$$\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \alpha \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} + \beta \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \quad (\text{C.2.5})$$

Note that this expression is equivalent to

$$\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \end{pmatrix} \cdot \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \quad (\text{C.2.6})$$

Hence $\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ can be isolated using the inverse

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \end{pmatrix}^{-1} \cdot \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \quad (\text{C.2.7})$$

Solution (2.7). By the properties of trigonometric functions we have

$$R(-\alpha) = \begin{pmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{pmatrix}$$

and hence

$$\begin{aligned} R(\alpha) \cdot R(-\alpha) &= \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \cdot \begin{pmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{pmatrix} = \\ &= \begin{pmatrix} \cos^2 \alpha + \sin^2 \alpha & \cos \alpha \sin \alpha - \sin \alpha \cos \alpha \\ \sin \alpha \cos \alpha - \cos \alpha \sin \alpha & \sin^2 \alpha + \cos^2 \alpha \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{aligned}$$

C.3 Regression models

C.3.1 Linear regression

Solution (3.1). From (3.1.3) we find easily that

$$\epsilon_i = y_i - \frac{5}{6} - \frac{1}{2}x_i.$$

Hence

$$\begin{aligned} \epsilon_1 &= 1 - \frac{5}{6} - \frac{1}{2}0 = \frac{1}{6} \\ \epsilon_2 &= 1 - \frac{5}{6} - \frac{1}{2}1 = -\frac{1}{3} \\ \epsilon_3 &= 2 - \frac{5}{6} - \frac{1}{2}2 = \frac{1}{6}. \end{aligned}$$

Solution (3.2). Using (3.1.9) we get

$$\begin{aligned} \hat{y}_1 &= \frac{5}{6} + \frac{1}{2} \cdot 0 = \frac{5}{6} && (3.1.9): \hat{y}(x) = \beta_0 + \beta_1 \cdot x. \\ \hat{y}_2 &= \frac{5}{6} + \frac{1}{2} \cdot 1 = 1\frac{1}{3} \\ \hat{y}_3 &= \frac{5}{6} + \frac{1}{2} \cdot 2 = 1\frac{5}{6}. \end{aligned}$$

Solution (3.3). Intercept β_0 : if education is 0 years, income is \$1000 (in average). This number is not interesting in developed economies, as almost no-one has no education at all. We are extrapolating to where there are no data.

Slope β_1 : those with one year more of schooling earn \$5000 more. This is a very meaningful number.

Note that if data shows 0 years of education, it may also be related to data problems, e.g. missings may be coded as 0-s. So we *may* see a lots of zeros, even if everyone has at least a few years of schooling. If this is the case, then the β_0 is no more interesting—it is just an average income for those with no data, assuming the linear relationship holds.

Solution (3.4). The interpretation of intercept β_0 : son's of 0-height fathers are 86.1 cm tall (in average). Interpretation of β_1 : sons of fathers who are 1 cm taller are 0.51 cm taller themselves.

The β_0 interpretation clearly does not make sense as there are no sons who are 0 cm tall. The second effect is a manifestation of regression to mean—sons of taller fathers are taller, but not as much as fathers themselves.

Solution (3.5). 1. $t = \text{coefficient}/\text{std.error} = 4/1.6 = 2.5$.

2. In this case $df = 105 - 5 = 100$, so we pick the row in Table 1.8 that corresponds to $df = 100$. In that line, the value $t = 2.5$ will be between 1.98 and 2.63. The former corresponds to significance 0.05 and the latter to 0.01. So we can conclude that the p -value is between 0.05 and 0.01.
3. Yes, it is, as $2.5 > 1.98$, the 5% critical t -value.
4. No, it is not, as $2.5 < 2.63$, the 1% critical t -value.
5. It means that $H_0 : \beta = 0$ is unlikely (less likely than 5% or another chosen significance level), so we reject it. This means the true coefficient is likely not zero, so these two variables are related.

Note: all software presents the t and p values assuming your H_0 is $\beta = 0$. This is usually what you want, but sometimes you may need other tests, e.g. $H_0 : \beta = 1$.

Solution (3.6). We have $\mathbf{x} = (0,0,2,2)$ and $\mathbf{y} = (1, -1, 3, 1)$, and the regression coefficients $\beta_0 = 0$ and $\beta_1 = 1$. Hence we can compute the predicted values $\hat{\mathbf{y}} = 0 + 1 \cdot \mathbf{x} = \mathbf{x}$, so $\hat{\mathbf{y}} = (0,0,2,2)$.

Now the residuals are $\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}} = (1, -1, 1, -1)$. Hence $SSE = \sum_i e_i^2 = 4$. The average y value is $\bar{y} = 1$, and hence the deviations from mean are $(0, 2, 2, 0)$ and hence $TSS = 8$. Accordingly, $R^2 = 1 - SSE/TSS = 0.5$.

Such calculations are often useful to do as a table:

i	x_i	y_i	\hat{y}_i	e_i	e_i^2	$(y_i - \bar{y})$	$(y_i - \bar{y})^2$
1	0	1	0	1	1	0	0
2	0	-1	0	-1	1	2	4
3	2	3	2	1	1	2	4
4	2	1	2	-1	1	0	0
					$SSE = 4$		$TSS = 8$

where $e_i = y_i - \hat{y}_i$ and $\bar{y} = 1$.

Solution (3.7). Before answering these questions we should refresh the interpretation of the coefficients (see Section 3.1.3).

1. Intercept corresponds to the predicted value where $u = 0$. Hence the average log wage for non-union members is 1.605.
2. For union members ($u = 1$) we have to add Intercept 1.605 and u estimate 0.179, the result is 0.178.
3. Finally, the difference is the estimate for u , 0.179. This can also be understood from the interpretation of β -s, it is the expected difference between the cases where $u = 1$ and $u = 0$.

Solution (3.8). 1. Reference category is the category that does not have a dummy variable displayed in the results table. In this case it is *rural area*.

2. Predicted value for North Central is the sum of intercept and *north central* estimate: $1.584 + 0.047 = 1.631$.
3. As rural areas is the reference category, the predicted log salary there is equal to the intercept 1.584.
4. This difference is captured by the *south* dummy. It is 0.032.
5. There is no variable that captures the difference between two categories where none of these is a reference category. But we can just compute the predictions in North East and South. Now when doing this you notice that both of those contain intercept that just cancels out. Hence what is left is the difference between the two dummies: log wages in North East are larger than those in south by $0.164 - 0.032 = 0.132$.

Solution (3.9). This is because in the original data *ethn* only allows a single category (*black*, *hispanic* and *other*). It is definitely possible to describe multi-racial identity using dummies but in that case the multi-race cases must be included in the original categorical data somehow. For instance, one can introduce two categories: *ethn₁* and *ethn₂*. Now we can have both $e_b = 1$ and $e_h = 1$ for someone who responds *ethn₁* = black and *ethn₂* = hispanic.

Solution (3.10). Let's multiply X and β in (3.1.49) using the ordinary matrix multiplication rules:

$$\begin{pmatrix} 1 & x_1^1 & x_2^1 & \dots & x_K^1 \\ 1 & x_1^2 & x_2^2 & \dots & x_K^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^N & x_2^N & \dots & x_K^N \end{pmatrix} \cdot \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_K \end{pmatrix} = \begin{pmatrix} \beta_0 & \beta_1 x_1^1 & \beta_2 x_2^1 & \dots & \beta_K x_K^1 \\ \beta_0 & \beta_1 x_1^2 & \beta_2 x_2^2 & \dots & \beta_K x_K^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \beta_0 & \beta_1 x_1^N & \beta_2 x_2^N & \dots & \beta_K x_K^N \end{pmatrix} \quad (\text{C.3.1})$$

and hence we can write (3.1.49) as

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} \beta_0 & \beta_1 x_1^1 & \beta_2 x_2^1 & \dots & \beta_K x_K^1 \\ \beta_0 & \beta_1 x_1^2 & \beta_2 x_2^2 & \dots & \beta_K x_K^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \beta_0 & \beta_1 x_1^N & \beta_2 x_2^N & \dots & \beta_K x_K^N \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_N \end{pmatrix}. \quad (\text{C.3.2})$$

Each line of this equation is equivalent to (3.1.48) after performing the matrix multiplication there, and there are N lines. Hence (3.1.49) is equivalent to (3.1.48) for each $i = 1 \dots N$.

C.3.2 Logistic regression

Solution (3.11). 1. Shipwreck: logistic, as this has binary outcome (survived/did not survive)

2. Cancer patients: this is also survival, but not binary any more. Now we ask *how long time*, i.e. a number. So linear regression is more appropriate.
3. GPA: linear, as the outcome can have any value (between 0 and 5 or so).
4. Admission to school: logistic, as binary outcome (admitted/not admitted)
5. Retweeting: logistic as binary outcome (retweeted/not retweeted)
6. How many people read tweets: linear, as it can be any number (well, a count—non-negative number). Note: dedicated count data models may offer a better solution here. See more in [Section 1.1.2 Counts](#), page 6.

Solution (3.12). The table reveals that the 5% critical z value is 1.96. Hence *Education* is statistically significant, but the other two variables are not. More precisely, they are *statistically significantly different from zero at 5% significance level*.

Remember: z -values, like t -values measure distance between the H_0 value and what we find in data. A large z value means that data and H_0 are rather different.

See [Section 1.5.1](#) and [Example 1.13](#).

C.4 Causality

Solution (4.1). For a downward bias, we need a situation for those who have flu are more likely to get a flu shot. One may think along these lines: those who do not feel well get anxious about falling ill, and quickly get the shot. However, as they get it too late (they have already contracted flu), it does not help them. We see that flu shot is more often associated with flu, but this is not because flu shot causes flu, but because falling ill “causes” flu shot.

Solution (4.2). The above example explained how concern about health can make people to both get flu shot and be less likely to contract flu. Here we need some kind of opposite process. One possibility is that people know something about how likely

they are to get flu, and act upon it. For instance, those with fragile health or weak immune system may get the shot, while healthy people do not bother. So even if the shot is effective, we may not even see it in data, if the first group is still more likely to contract flu. But in any case, the observed effect size will be smaller.

This process seems more plausible, but both of these can be assessed through behavioral studies. But in any case, there are probably many other mechanisms that determine health behavior and morbidity.

C.5 Predictive Modeling

Solution (5.1). The data looks like:

case:	1	2	3	4	5	6	7	8	9	10
Actual	1	0	0	1	1	0	0	0	1	0
Expert	0	0	0	1	0	0	1	0	1	1

Consider “0” to be negative and “1” to be positive. We have $TN = 4$ (cases 2, 3, 6, 8) as both the actual value is “0” and the expert predicted “0”. $TP = 2$ (cases 4, 9) as the actual value is “1” and the expert predicted “1”. $FP = 2$ (cases 7, 10) as the expert predicted “1” but the actual value was “0”. Finally, $FN = 2$ (cases 1, 5) where the expert predicted “0” while the actual value was “0”. Hence the confusion matrix is

Actual	Predicted		Total
	0	1	
0	$TN = 4$	$FP = 2$	6
1	$FN = 2$	$TP = 2$	4
Total	6	4	10

Note that the predicted values do not have to be related to any particular model. In terms of constructing the confusion matrix, expert opinion or even a random guess is perfectly good.

Solution (5.2). Let’s take participants as positives. The actual data contains 2490 non-participants (the majority) and 185 participants. Hence the model predicts that everyone is a non-participant.

The confusion matrix will be:

Actual	Predicted			Total
	Non-Participants	Participants	Total	
Non-Participants	2490	0	2490	
Participants	185	0	185	
Total	2675	0	2675	

As everyone is predicted to be a non-participant, the predicted participants’ column only contains zeros. We have $TN = N = 2490$, $FP = 0$, $FN = P = 185$ and $TP = 0$.

Solution (5.3). We have

$$TN = 10, TP = 60, FP = 20, FN = 10 \quad (\text{C.5.1})$$

and $T = 100$. Hence

$$\begin{aligned} A &= \frac{TN + TP}{T} = \frac{70}{100} = 0.7 \\ P &= \frac{TP}{TP + FP} = \frac{60}{80} = 0.75 \\ R &= \frac{TP}{TP + FN} = \frac{60}{70} \approx 0.86 \\ F &= \frac{2}{\frac{1}{P} + \frac{1}{R}} = 0.8 \end{aligned} \quad (\text{C.5.2})$$

Solution (5.4). a) If participants are positives, the confusion matrix is the same as in Example 5.1:

Actual	Predicted		
	Non-Participants	Participants	Total
Non-Participants	2452	38	2490
Participants	89	96	185
Total	2541	134	2675

The model goodness measures are: $Accuracy = (2452+96)/2675 = 95.3\%$, $Precision = 96/134 = 71.6\%$ and $Recall = 96/185 = 51.9\%$.

b) If participants are negatives, the confusion matrix's rows and columns are swapped around:

Actual	Predicted		
	Participants	Non-Participants	Total
Participants	96	89	185
Non-Participants	38	2452	2490
Total	134	2541	2675

Now $Accuracy = (2452 + 96)/2675 = 95.3\%$ is the same, $Precision = 2452/2541 = 96.5\%$ and $Recall = 2452/2490 = 98.5\%$.

c) Accuracy does not depend on the choice of positives/negatives, it is only concerned about correct predictions, so here the choice does not matter. But in one case we get moderate precision and recall scores, in the other case those figures are very high. The moderate scores are appropriate if we are mainly interested in spotting the participants. The model is not very good at that. The high scores are good if we are mainly concerned in non-participants—that group is easy to find.

Solution (5.5). Specificity is the same as recall for negative outcomes, and sensitivity is the same as recall. Specificity 100% means we do not have any false positives, and

sensitivity 63.5% means we capture 63.5% of the positive cases. We can, for instance, take 1000 actual negative cases and 1000 actual positive cases. Now all 1000 actual negatives will be predicted as negative (specificity = 100%), but only 635 of 1000 positives will be categorized as positive (sensitivity = 63.5%), see the left panel of the table below.

		All cases		Asymptotic cases	
		Predicted Negative	Predicted Positive	Predicted Negative	Predicted Positive
Actual	Negative	1000	0	1000	0
	Positive	365	635	650	350

For asymptotic cases we still have 1000 true negatives and 0 false positives (specificity is still 100%) but now only 350 out of 1000 actual positives are categorized correctly (sensitivity is 35%).

The test seems to be of dubious quality if roughly 1/3 of all cases, and 2/3 of asymptotic cases slip through.

Solution (6.2). Section 2.2.2 lists three properties of distance metric. The first one is

$$d(\mathbf{x}, \mathbf{y}) = 0 \Leftrightarrow \mathbf{x} = \mathbf{y}.$$

This property is not satisfied for cosine-related distances: $d_{cos}(\mathbf{x}, \mathbf{y}) = 0$ means cosine is 1 (or angle is 0), but this is true for all vectors that point to the same direction, not just for equal vectors.

Another issue arises from the fact that these distances are not defined for null vector, hence distance between null-vector and any other vector is undefined.

C.6 Machine Learning Models

C.6.1 Metric distance: A revisit

Solution (6.1). We have $\mathbf{x}_1 = (1.000, 2.000, 3.000)$, $\mathbf{x}_2 = (3.000, 2.000, 1.000)$ and $\mathbf{x}_3 = (1.000, 1.000, 1.000)$. The norms are

$$\begin{aligned} \|\mathbf{x}_1\| &= \sqrt{\mathbf{x}_1^\top \mathbf{x}_1} = \sqrt{1^2 + 2^2 + 3^2} = 3.742 \\ \|\mathbf{x}_2\| &= \sqrt{\mathbf{x}_2^\top \mathbf{x}_2} = \sqrt{3^2 + 2^2 + 1^2} = 3.742 \\ \|\mathbf{x}_3\| &= \sqrt{\mathbf{x}_3^\top \mathbf{x}_3} = \sqrt{1^2 + 1^2 + 1^2} = 1.732. \end{aligned} \tag{C.6.1}$$

Hence the normed versions are

$$\begin{aligned}\mathbf{x}_1^n &= \frac{\mathbf{x}_1}{\|\mathbf{x}_1\|} = \frac{(1.000, 2.000, 3.000)}{4} = (0.267, 0.535, 0.802) \\ \mathbf{x}_2^n &= \frac{\mathbf{x}_2}{\|\mathbf{x}_2\|} = \frac{(3.000, 2.000, 1.000)}{4} = (0.802, 0.535, 0.267) \\ \mathbf{x}_3^n &= \frac{\mathbf{x}_3}{\|\mathbf{x}_3\|} = \frac{(1.000, 1.000, 1.000)}{2} = (0.577, 0.577, 0.577).\end{aligned}\quad (\text{C.6.2})$$

Finally, the cosine similarity is just the inner product of the normed vectors:

$$\begin{aligned}c(\mathbf{x}_1, \mathbf{x}_2) &= \mathbf{x}_1^{n^T} \cdot \mathbf{x}_2^n = (0.267, 0.535, 0.802)^T \cdot (0.802, 0.535, 0.267) = 0.7142857 \\ c(\mathbf{x}_1, \mathbf{x}_3) &= \mathbf{x}_1^{n^T} \cdot \mathbf{x}_3^n = (0.267, 0.535, 0.802)^T \cdot (0.577, 0.577, 0.577) = 0.9258201\end{aligned}\quad (\text{C.6.3})$$

So \mathbf{x}_1 is more similar to \mathbf{x}_3 than to \mathbf{x}_2 .

C.6.2 Trees and tree-based methods

Solution (6.3). The right split in Figure 6.12 contains two branches: the larger Branch 1 one contains one circle and 5 crosses, the smaller Branch 2 one contains a single cross and 3 circles. The best approach is to make a table for calculations:

Size	Branch 1		Branch 2	
	6		4	
	✗	○	✗	○
Count	5	1	1	3
Pr	0.833	0.167	0.25	0.75
$\log_2 \text{Pr}$	-0.263	-2.585	-2	-0.415
$\text{Pr} \cdot \log_2 \text{Pr}$	-0.219	-0.431	-0.5	-0.311
Branch entropy	0.65		0.811	
Total entropy		0.715		

As the original entropy was 0.971, the entropy gain is 0.256. This is much more than what we found for the left split of Figure 6.12.

C.6.3 Naive Bayes

Solution (6.4). Re-write the Bayesian expression for the case of no-viagra-no-spam:

$$\Pr(S = 0|V = 0) = \frac{\Pr(V = 0|S = 0) \cdot \Pr(S = 0)}{\Pr(V = 0)} \quad (\text{C.6.4})$$

Based on the table in Example 6.5, we can compute the necessary probabilities:

- $\Pr(V = 0|S = 0)$, probability of no “viagra” in no-spam emails. From the table we can see that it is $500/600 = 5/6 \approx 0.833$.

- The prior, $\Pr(S = 0)$, the proportion of legitimate emails. It is $600/1000 = 3/5 = 0.6$.
- The normalizer, $\Pr(V = 0)$, the probability not to see “viagra” in emails, $650/1000 = 13/20 = 0.65$.

Inserting the values in (C.6.4), we get

$$\begin{aligned}\Pr(S = 0|V = 0) &= \frac{\Pr(V = 0|S = 0) \cdot \Pr(S = 0)}{\Pr(V = 0)} = \\ &= \frac{\frac{5}{6} \cdot \frac{3}{5}}{\frac{13}{20}} = \frac{60}{78} = \frac{10}{13} \approx 0.769.\end{aligned}\quad (\text{C.6.5})$$

Based on the information that the email contains no word “viagra”, we update the prior 0.6 to 0.769, $\approx 28\%$.

Solution (6.5). We use the same training data as in [Example 6.7 Naive Bayes Classifier](#), page 292 and hence all the probabilities are the same. The bow for the new email, “life is life”, is

	good	in	is	life	viagra
\mathbf{x}_4	0	0	1	1	0

The log-likelihood for spam is:

$$\begin{aligned}\ell(S = 1|\mathbf{x}_4) &= \\ &= \log \Pr(S = 1) + \log \Pr(\text{is} = 1|S = 1) + \log \Pr(\text{life} = 1|S = 1) = \\ &= -1.099 + 0 + 0 = -1.099\end{aligned}\quad (\text{C.6.6})$$

and log-likelihood for non-spam is

$$\begin{aligned}\ell(S = 0|\mathbf{x}_4) &= \\ &= \log \Pr(S = 0) + \log \Pr(\text{is} = 1|S = 0) + \log \Pr(\text{life} = 1|S = 0) = \\ &= -0.405 - 0.6930 = -1.099.\end{aligned}\quad (\text{C.6.7})$$

As both log-likelihoods are the same, we have a tie as in [Example 6.7](#). This is because the data here is essentially the same as in the example, we have swapped “viagra” for “is”, but “is” has exactly the same probabilities as “viagra” for both spam and non-spam.

Solution (6.6). An easy solution is $w_1 = w_2 = 1$, $\bar{z} = 0.5$.

Solution (6.7). From the Table 6.8 we have weights

$$\mathbf{w}_{h1} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \mathbf{w}_{h2} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \mathbf{w}_y = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \quad (\text{C.6.8})$$

and biases

$$b_{h1} = 1.5 \quad b_{h2} = 0.5 \quad b_y = 0.5. \quad (\text{C.6.9})$$

We can compute the h_1 node values:

$$\chi_1 = \mathbf{x}^\top \cdot \mathbf{w}_{h1} = (0 \quad 1) \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 1 \quad \text{and} \quad h_1 = \mathbb{1}(\chi_1 > b_{h1}) = \mathbb{1}(1 > 1.5) = 0. \quad (\text{C.6.10})$$

Analogously, for the second hidden node h_2 we have

$$\chi_2 = \mathbf{x}^\top \cdot \mathbf{w}_{h2} = (0 \quad 1) \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 1 \quad \text{and} \quad h_2 = \mathbb{1}(\chi_2 > b_{h2}) = \mathbb{1}(1 > 0.5) = 1. \quad (\text{C.6.11})$$

So we have $\mathbf{h} = (h_1, h_2)^\top = (0, 1)^\top$. Now we can perform a similar operation with the output layer:

$$z = \mathbf{h}^\top \cdot \mathbf{w}_{h2} = (0 \quad 1) \cdot \begin{pmatrix} -1 \\ 1 \end{pmatrix} = 1 \quad \text{and} \quad y = \mathbb{1}(z > b_y) = \mathbb{1}(1 > 0.5) = 1. \quad (\text{C.6.12})$$

So we have $0 \text{ XOR } 1 = 1$.

Solution (6.8). a) Remember the softmax definition (6.6.8):

$$\Lambda(\mathbf{x})_i = \frac{e^{x_i}}{\sum_j e^{x_j}}.$$

Let's do the solution in a table:

	1	2	3	1	2	3
inputs x_i	1.00	2.00	3.00	4.00	5.00	6.00
exponents e^{x_i}	2.72	7.39	20.09	54.60	148.41	403.43
Sums $\sum_i e^{x_i}$	30.19			606.44		
probabilities $\frac{e^{x_i}}{\sum_i e^{x_i}}$	0.09	0.24	0.67	0.09	0.24	0.67

As visible here, both probabilities are exactly the same.

b) From the definition, it is clear that

$$\Lambda(\lambda + \mathbf{x})_i = \frac{e^{\lambda+x_i}}{\sum_j e^{\lambda+x_j}} = \frac{e^\lambda e^{x_i}}{\sum_j e^\lambda e^{x_j}} = \frac{e^{x_i}}{\sum_j e^{x_j}} \quad \square \quad (\text{C.6.13})$$

as e^λ cancels out.

Solution (6.9). At the first filter position, top-left of \mathbf{M} , we have $0 \cdot (-1) + 0 \cdot (-1) + 0 \cdot (-1) + 1 \cdot 3 = 3$. For the second position, we have $0 \cdot (-1) + 1 \cdot (-1) + 1 \cdot (-1) + 1 \cdot 3 = 1$,

$$\begin{array}{ccc} 3 & 1 & 0 \\ \hline -1 & 1 & -3 \\ 0 & -1 & -1 \end{array}$$

 and so on. The final result is

We see the largest value (3) at the top-left corner, and the smallest value (-3) at the middle-right position. The top-left position of the image reflect the filter: positive value at bottom-right and zeros around it. This is the shape that the filter is most sensitive for. Right-middle is a negative of the same pattern: zero at bottom-right and ones around it. This makes the filter to respond with the same value, just with a flipped sign.

List of Cheatsheets

1.1 Descriptive Statistics	16
1.2 Events, Probability and Conditional Probability	35
1.3 Random variable and realization	37
1.4 Expected value, mean, variance	46
1.5 Summary of the concepts	70
3.1 Simple Regression: Definition	132
3.2 Simple Regression: Interpretation	139
3.3 SSE and related terms	148
3.4 Categorical variables in linear regression	158
3.5 Log transformations in linear regression	164
3.6 Linear regression vs logistic regression	188
4.1 OLS Estimators for causal inference	225
5.1 Confusion matrix and related measures	243

List of Examples

1.1 Predicting election results	19
1.2 Monty Hall Problem	22
1.3 Red and green, nice and bad	27
1.4 Gender and Titanic Survival	30
1.5 Probability of diagnosis	32
1.6 Do you have cancer?	33
1.7 Expectation of a 3-valued RV	41
1.8 Rolling a die	51
1.9 Expected value of uniform RV	54
1.10 Entropy of uniform distribution	64
1.11 Rejecting and not rejecting a statistical hypothesis	67
1.12 Unemployment example with bad data	68
1.13 Significance and p -value	68
1.14 Confidence intervals for human height	74
1.15 Sample mean of sons' height	75
1.16 Two daughter problem: girl has a name	84
1.17 Are hospitals unsafe during the weekends?	89
2.1 Graphical way to add vectors	94
2.2 Application of 2-D vector space \mathbb{Z}^2	96
2.3 Are these vectors linearly independent?	98
2.4 L_3 norm of vector (1,1)	100
2.5 Manhattan norm of vector (1,1)	101
2.6 Product of non-square matrices	109
2.7 Matrix product is not commutative	110
2.8 Transpose of matrix product	111
2.9 Inner and outer product	112
2.10 Matrix trace	113
2.11 Condition numbers	116
3.1 How fast does the universe expand?	128
3.2 Predicted Velocity of Galaxies	131
3.3 Unemployment versus GDP growth	134
3.4 Interpreting Regression Table	138
3.5 SSE for the iris sepals regression	142
3.6 R^2 for <i>setosa</i> sepals regression	144
3.7 R^2 of Hubble diagram: 100 years later	147
3.8 How is income related to education and literacy?	149

3.9 Income, education and literacy: interpretation	153
3.10 How does income depend on age?	161
3.11 Linear, log-linear, and log-log transformations	163
3.12 Importance of social skills	167
3.13 Convert data to design matrix	174
4.1 Smoking and lung cancer	198
4.2 Do parachutes help to survive a “gravitational challenge”?	199
4.3 RCT—how to determine the effect of pneumonia vaccine	201
4.4 Do more extensive public health measures during pandemic help economy? Correia <i>et al.</i> (2020)	203
4.5 Flu Vaccine Efficacy: a Case-Control Study	204
4.6 Former outcome as counterfactual	208
4.7 Expected value of unobserved characteristics	212
4.8 Cross-sectional estimator of college effect is biased	213
4.9 COVID-19 stay-at-home orders in Nordic countries	214
4.10 COVID-19 stay-at-home orders in Nordic countries: regression approach	215
4.11 President’s approval: before and after September 11th	217
4.12 Presidents approval: before and after September 11th, the regression approach	218
4.13 COVID-19 Epidemic and Presidents Approval	222
4.14 President’s approval rating: the regression approach	224
5.1 Confusion Matrix	234
5.2 Accuracy, Precision, Recall, <i>F</i> -score	236
5.3 Computing ROC curve	239
5.4 Overfitting in case of categorization	248
6.1 Data normalization	255
6.2 Mahalanobis transformation of iris data	259
6.3 Cosine similarity in \mathbb{R}^2	261
6.4 Splitting data for decision trees: income and education	273
6.5 Bayesian spam filter based on a single word	282
6.6 Bayesian spam filter with two phrases	286
6.7 Email classification	292
6.8 Email classification with smoothing	294
6.9 Softmax outputs	305
6.10 1-D convolution	307
6.11 Edges on image	309
7.1 DTM of Laozi quotes	325
7.2 TF-IDF of Laozi quotes	328
7.3 TCM of Laozi quotes	329
7.4 Long embedding vector of Laozi quotes	332
9.1 How big are emergencies	373
9.2 Principal component regression with 2-D data	379
9.3 How are conservative family values and identity related to willingness to do good for society?	381

List of Figures

1.1	Age of Titanic passengers	13
1.2	Computing Pareto ratio	16
1.3	Monty Hall Problem	23
1.4	Venn diagram	26
1.5	Venn diagram of three event	27
1.6	At least one six, given one die has an odd number	29
1.7	Gender distribution of Titanic passengers	30
1.8	Bernoulli p.m.f	50
1.9	Density plot: from discrete to continuous	53
1.10	Distribution of house prices	56
1.11	Log-normal distribution	57
1.12	Pareto distribution	59
1.13	CLT: How distributions converge to normal	61
1.14	Entropy in case of different probability over states. Different shades of gray denote different probability, uniformly spread over 8 states A – H . The rightmost column is the corresponding entropy.	65
1.15	Confidence intervals for temperature	71
1.16	Distribution of son's height in fathers-sons data. It is well approximated by normal distribution (red line) with mean $\mu = 174.5$ and standard deviation $\sigma = 7.2$, this is common for measures of adult animals. Dotted vertical lines are the boundaries of the 95% confidence intervals.	74
1.17	Distribution of mean height of four sons. The distribution of means are well approximated by normal (red line) with mean $\mu = 174.6$ and standard deviation $\sigma = 3.6$. Dotted vertical lines are the boundaries of the 95% confidence intervals.	76
1.18	Testing $H_0: Z = X_1 - X_2 = 0$ where $X_1 \sim N(2, 1)$ and $X_2 \sim N(0, 1)$. The histogram displays the distribution of Z for 1000 replications.	77
1.19	Simulated difference in smoking habit across two types of workplaces	81
1.20	Simulated and actual data: the smoking ban example. All simulations give results near 0, but the difference in actual data is much larger.	82
1.21	Hypothetical damage in allied bombers. Red dots denote damage in any of the thousands of bombers, marked here on a single figure. Original image: Emoscopes CC BY-SA.	87

2.1	Vector space: all vectors on a plane can be computed as a linear combination of two base vectors, here \mathbf{a} (red) and \mathbf{b} (blue). The dotted red and blue arrows show which linear combinations are needed to create vectors \mathbf{c} and \mathbf{d} (black).	97
2.2	Vector \mathbf{v} has both components, v_x and v_y equal to one. From Pythagorean theorem, its length is $\sqrt{2}$. Generalized “length” of a vector is called <i>norm</i> and denoted by $\ \mathbf{v}\ $, in this case $\ \mathbf{v}\ = \sqrt{2}$.	99
2.3	King in chess can move one field in every direction. The numbers on the chessboard denote the number of moves king at d_4 needs to reach that position. For instance, it needs three moves to get to to $f7$, and hence the vector from d_4 to $f7$ has chessboard norm 3.	102
2.4	Unit circles – points sets of distance 1 from the origin (0,0) (the central dot) in different 2-D L_p spaces. If $p < 2$, the circle looks more like a star, with the Manhattan distance, $p = 1$, being diamond-shaped. If $p > 2$, the circles are more and more box-shaped.	103
2.5	Wireframe image of the β -rune defined by matrix \mathbf{B} in (2.4.1). All vertices are plotted and thereafter sequentially connected.	118
2.6	The same object as in Figure 2.5 but rotated 30 degrees by multiplication with the corresponding rotation matrix. The rotated vertices are given as \mathbf{B}^{30} in (2.4.5).	119
3.1	Iris <i>setosa</i>	124
3.2	Many possible relationships	125
3.3	Original Hubble diagram	129
3.4	Original Hubble diagram	131
3.5	Interpretation of regression coefficients	134
3.6	Relationship between unemployment and GDP growth across countries in 2016, and the corresponding regression line. World Bank data.	135
3.7	Range-based construction of R^2	145
3.8	Original versus modern Hubble diagram	147
3.9	Regression plane with two explanatory variables (<i>HS Grad</i> and <i>Illiteracy</i>). The gray plane represents the 2-D regression plane, the large dots are the actual income values, the small dots are the predicted values on the regression plane, and the vertical lines that connect those values are the corresponding residual errors. Colors correspond to the actual income values.	151
3.10	Direct and indirect effects	152
3.11	Distribution of UK household income in early 1980-s. Income distribution (left panel) does not look normal, it has a long thin tail of high-income households reaching up to weekly income 1000£. Log income (right panel) is fairly close to normal as logarithm spreads low-income observations out and squeezes the high-income ones closer together. Ecdat package data.	161

3.12 Diamond mass ($\text{carat}=0.2$ gram) and price data, including the corresponding regression lines. Left panel shows the linear model in <i>price</i> and <i>carat</i> . One can see that the line does not capture the convex pattern in data. Middle panel shows a model that is linear in $\log \text{price}$ and <i>carat</i> . Now the data pattern is concave and again the line fails to capture it well. On the right panel we log-transform both variables, and the result looks very good visually.	163
3.13 Interpretation of interaction effects. The blue line depicts the relationship between income and social skills for low-social-skill individuals, and red line that for the high-social-skills individuals.	167
3.14 How participation depends on age	180
3.15 Logistic function	182
3.16 Participation as a function of age: logistic curves	184
3.17 Interpretation of logistic regression results	186
4.1 Does flu shot help to avoid flu?	196
4.2 Time trends of smoking and lung cancer in Sweden (red) and the U.S. (blue), 1975=100. Solid lines depict the cigarette sales, and dashed lines are age-normalized death rates in lung cancer. We can see cigarette sales in both countries peaking around 1980. This is followed by a fall in death rate somewhat later.	198
4.3 Causal versus correlational regression	207
4.4 Mean independence	210
4.5 U.S. President G. W. Bush approval rating through summer and fall 2001. The dashed vertical line corresponds to September 11 terrorist attacks.	217
4.6 Hypothetical education data. Both the levels and trends differ for the treatment and control provinces. Dashed line denotes the counterfactual assumption, the difference between the actual and counterfactual value is the DiD estimate, here 1 year of extra schooling.	221
4.7 Presidents' average approval rate before and after March 15th of their 4th year in office. We can see that Obama's approval increased by more than two percentage points over this period while that of Trump grew by slightly less than one point. The dashed blue line depicts the counterfactual—the path of Trump approval rate, if it had been similar to that of the Obama's. The difference between the counterfactual and the actual approval (green dashed line), -1.25 points, is the effect.	223
4.8 Effect on Brexit referendum on the business investments in UK: an example of graphical DiD approach.	226
5.1 Example ROC curve for linear probability model (black) and logistic regression (pink). The figure suggest that in most cases logit outperforms LPM as it is able to achieve higher TPR over TPR range 0 to 0.3.	239
5.2 Two possible patterns to explain the same data	245

5.3	Artificial age-income data. Left panel shows just the data points, the right panel displays the same data and a number of polynomial regression models with various polynomial degrees.	247
5.4	Categorization task with blue and red dots. Logistic regression (left panel) does not capture the red and blue “peninsulas” extending into the other color area. This is underfitting. Nearest neighbors (with $k = 1$) carves out a separate island for every single dot. The decision boundary seems too complex. This is overfitting.	248
5.5	The same artificial age-income data as on Figure 5.3. The training data points are denoted with green, validation points with red. The polynomial regression curves up to degree 7 are fitted through the data training data. One can see the 7-th degree polynomial that fits all training data perfectly, predicts values that are far off from the actual validation values.	250
5.6	Training and validation RMSE	252
6.1	Non-normalized features (left) and normalized features (right). Dark blue, green and yellow mark the same three datapoints on both images. The dotted line depicts a circle in the original feature space, the solid line is a circle in the normalized feature space. Note how the relative distance between dark blue and green, and dark blue and yellow dots differ in the original and in the normalized features.	257
6.2	Boston housing data: neighborhood crime rate (<i>crim</i>) versus average number of rooms (<i>rm</i>). Non-normalized (left) versus normalized features (right). While the images look exactly the same, the Euclidean distance rankings are different: the nearest (colored) neighbor the green dot is the orange on the left panel, and the blue dot on the right panel.	258
6.3	Original features (left) and Mahalanobis-transformed features (right). The same three cases are marked with different colors on both images. The dotted line depicts a circle in the original feature space, the solid line is circle in Mahalanobis feature space.	259
6.4	Iris data: petal width versus petal length in the original coordinates (left panel) and in the corresponding Mahalanobis-transformed coordinates (right panel).	260
6.5	Example data: some of the datapoints are categorized into yellow and violet, but some are not (left panel). Intuitively, the empty circles should be classified according to a colored one nearby. This is the intuition of the <i>nearest neighbor</i> method. On the right panel, all the points that are closer to a violet one are painted violet and those that are closer to a yellow one are colored yellow. All the empty circles now lie in one of these areas of solid color and can be categorized either as yellow or violet.	264
6.6	The same data points as in Figure 6.5, but now categorized based on 5 (left) and 25 (right) nearest neighbors. We can see that in the latter case, there are several groups of points that are embedded in the area of different color.	265

6.7 Animal game as a decision tree	268
6.8 Titanic survival as decision tree	269
6.9 Decision tree solving a 2-D task	270
6.10 Regression tree for 1-D task	271
6.11 Recursive binary split	273
6.12 Comparing binary splits	276
6.13 Overfitting in 1-D regression tree	278
6.14 SVM Decision Boundary	297
6.15 Schematic look of neuron	299
6.16 And Perceptron. The inputs x_1 and x_2 form the <i>input layer</i> , the single computing node forms the <i>output layer</i> . While the input layer only provides output to the node, the output node itself performs two operations: linear transformation $z = w_1x_1 + w_2x_2$, and <i>activation</i> , $y = \mathbb{1}(z > \bar{z})$	300
6.17 XOR Perceptron. The inputs x_1 and x_2 form the <i>input layer</i> , but now both input layer nodes are connected to both <i>hidden layer</i> nodes h_1 and h_2 , and not to the output layer. Both hidden layer nodes perform linear transformation and activation, using different weights \mathbf{w}_h and biases b_h . The single <i>output layer</i> node behaves exactly like in case of <i>AND</i> -perceptron, just it gets its inputs from the hidden layer, not from the input layer.	302
6.18 Multi-layer perceptron: this is a dense network with four inputs, two outputs, and with two hidden layers, the first one with five and the second one with four nodes. It is a dense network in a sense that all nodes in the previous layer are connected to all nodes in the following layer.	304
6.19 A few popular activation functions for neural networks. Leaky ReLU with $\alpha = 0.2$ is shifted slightly up for clarity.	306
6.20 Vertical edge detection using 2D convolution filter. Moving the convolution filter across the image, the top-left corner (green frame) results in zero, as all the pixel values are 0. In the next position (red frame) the negative filter weights overlap with 0-values pixels while positive weights overlap with 1-value pixels. As a result, the filter returns 2. Finally, in the rightmost position (blue frame), both positive and negative weights overlap with equal pixel values (1) and hence the output is again 0. The filter only “fires” if there is a vertical edge on the image.	309
6.21 Edge detection with convolutions	310
6.22 Color image and multiple filters	311
6.23 Complex and simple pattern	314
7.1 Black-and-white image: Lyman trestle	316
7.2 Pixels: detail of Lyman trestle	318
7.3 Storing image data: flag of Scotland	319
7.4 The same image rotated 10 degrees.	321

7.5	Image of a text page (left panel). It is rotated 24 degrees counterclockwise. Right panel depicts the gray value density along the vertical axis. The galaxy image in the form of a triangular dip, centered at row 200, is clearly visible. However, the text lines cannot be distinguished in the plot.	321
7.6	The same image as in Figure 7.5 but now rotated into correct position (left panel). The image is now visible as the rectangular dip with vertical sides. Now also the text lines are represented by a regular wavy pattern of lighter and darker stripes. Smooth slopes on both sides of the true image are related to the tilted white background embedded in the image.	322
7.7	Word and country similarity	334
8.1	<i>SSE</i> as a function of β_0 and β_1	351
8.2	Likelihood value for the coin toss, depending on the head probability p	352
8.3	One step of Gradient Ascent. We start from an initial guess \mathbf{x}^0 and take a step along the gradient. This moves us uphill to \mathbf{x}^1 . The function $f(\mathbf{x})$ is depicted by the surface overlaid by (rather circular) <i>level sets</i>	352
8.4	One Gradient Ascent step for function $f(\mathbf{x}) = x_1 \cdot \log x_2$. At the initial point $\mathbf{x}^0 = (1,1)'$, the gradient $(0, 1)'$ points straight up. We move in that direction by the amount (learning rate) $R = 0.5$. This leads us to $(1,1.5)'$, our next approximation for the maximum.	353
8.5	Linear regression (left panel) versus ridge regression (right panel). Solid dots represent data and empty triangles are predictions. Black is training and red testing data. Linear regression make much more noisy predictions than regularized ridge regression. There are 6 other highly correlated features not visible in this figure.	355
9.1	5 distinct clusters and the cluster centers	359
9.2	Loss and partitioning	361
9.3	k -means algorithm at work	364
9.4	Elbow plot of 5 clusters	365
9.5	Elbow plot with now structural clusters in data	366
9.6	Agglomerative clustering on sample data	367
9.7	Agglomerative clustering of iris flowers	368
9.8	Male-female height data	369
9.9	PCA on 2-D data	372
9.10	PCA: Explained variance	374
9.11	Highly correlated variables	375
9.12	Proportion of variance, explained by PC-s	376
9.13	Principal component regression on 2-D data	380

9.14 How cluster analysis treats data: homogenous subgroups (left panel) can be replaced by their corresponding cluster centers (right panel). In this way we can reduce the original 100-observation dataset to 5 different "types". These types can be either interpreted, one can design separate measures for each type, for instance marketing strategies in case of customer types, and one can also replace each observation with the cluster center in order to compress the data.	384
9.15 How PCA treats data	385
10.1 Two movies rated by three users, the same data as in Table 10.1 but now displayed graphically.	388
10.2 Two movies rated by three users, the same data as in Table 10.1, column <i>Centered rating</i> . The point for Su is shifted a little bit on the figure for clarity.	389
11.1 Fair test that gives unfair results	400
A.1 $f(\mathbf{x}) = x_1 \cdot \log x_2$ as function of x_1 and x_2 . The <i>level sets</i> , contours of equal values, are plotted both on the surface and on bottom of the figure box.	408
A.2 Gradient of $f(\mathbf{x})$, depicted as two surfaces (upper panel). The blue surface corresponds to $\frac{\partial}{\partial x_2} f(\mathbf{x}) = x_1/x_2$, the red one to $\frac{\partial}{\partial x_1} f(\mathbf{x}) = \log x_2$. The lower panel depicts the gradient as arrows plotted on the levels (contours) of the function. The length of the arrows is proportional to the gradient length, their direction is equal to the gradient direction. One can easily see that the norm of gradient is proportional to the steepness of the function surface, and gradient points to the direction of the steepest climb.	409
A.3 Function increasing along x_2 while constant along x_1	411
A.4 Function increasing along $-x_1$ while constant along x_2	411
A.5 Function increasing both $-x_1$ and x_2 . The direction of steepest climb is somewhere between these two gradient components.	412

List of Tables

1	Greek alphabet	vi
1.1	Quantitative measures and associated statistical operations	5
1.2	Computing variance. The last row displays the averages, the of those is just the sample average $\bar{x} = 2$, and the last one is variance s^2 . Note that the average of the middle column is 0. This is always true through the definition of mean.	11
1.3	Partitioning the sample space into two subsets. The left side contains all simple events in A , the right side the simple events not in A	29
1.4	Computing variance of a discrete random variable	43
1.5	Multiplication of RV by a scalar	45
1.6	Possible outcomes number of heads when tossing two coins, and corresponding probabilities.	48
1.7	Log-normal 20/80 ratios depending on σ . For instance, if $\sigma = 3.29$ then the upper 5% of population possesses 95% of total resources. See Figure 1.11 for the shape of the corresponding p.d.f-s.	58
1.8	Critical t -value table	74
1.9	Simulated smoking habits (percent of smokers) at smoking-ban/no-smoking-ban workplaces for 5 random simulations. The difference in data is 7.76 percent. We can see that simulated values in these 5 examples are much smaller than what is visible in the data.	80
3.1	Example cases from Iris dataset	128
3.2	Software output table from sepal length–sepal width regression. Different software package may provide slightly different output, but the main information is very much the same.	137
3.3	Computing SSE for <i>setosa</i> data. <i>Sepal length</i> and <i>Sepal width</i> are the actual datapoints. \hat{y} is the predicted width, given $\beta_0 = 0$ and $\beta_1 = 1$. e is the corresponding deviance and e^2 is squared deviance, “squared error”. The last line gives the sum of all rows.	142
3.4	Computing R^2 for <i>setosa</i> data. The table is analogous to the table in Example 3.5, just this time using the actual regression coefficient values instead of 0 and 1.	144
3.5	Sample of Males data (left), binary (dummy) variable m denoting status “married” (center). Dummies for three possible ethnic categories are in the rightmost three columns.	155

3.6	Results of three different regression models: linear-linear, log-linear, and log-log.	163
3.7	Example skill-income data.	165
3.8	An example of “Treatment” data	179
4.1	Example flu shot data. <i>Id</i> is the patient id, <i>Flu shot</i> is a dummy variable denoting whether the person got ($S = 1$) or did not get ($S = 0$) a flu shot, and <i>Flu</i> denotes whether they got flu ($F = 1$) or not ($F = 0$). The table shows four observations only, but there can be many more.	195
4.2	G.W.Bush approval ratings	219
4.3	DiD regression estimate for the effect of 9/11 terror attacks on presidents approval rating. Standard errors in italics.	219
4.4	Four datapoints	220
4.5	An excerpt of approval ratings data for presidents Obama and Trump during their fourth year in office. Polling data from RealClearPolitics. The displayed period, from mid-January to mid-April centers on mid-March, the weeks in 2020 where the world, including the US, rapidly realized the magnitude of the unfolding health crisis.	222
4.6	The effect of COVID-19 pandemic on president’s approval rate	223
4.7	DiD regression estimate for the effect of COVID-19 epidemic on the US president’s approval rating. Standard errors in italics.	225
4.8	Four potential outcomes in binary treatment/binary outcome data. “0” and “1” denote presence and absence of treatment and outcome, the letters in cells are the corresponding case counts.	228
5.1	Example confusion matrix	233
5.2	Confusion matrix for two categories	233
5.3	Prediction errors from polynomial regression on validation data as shown in Figure 5.5. The linear model (1st-degree polynomial) achieves the smallest <i>RMSE</i> on validation data.	251
6.1	Recent house sales	254
6.2	Recent house sales	265
6.3	DTM of the three example emails (rows \mathbf{x}_1 , \mathbf{x}_2 and \mathbf{x}_3), the corresponding word counts (N_W), and conditional probabilities of word in spam ($\Pr(W = 1 S = 1)$) and no-spam ($\Pr(W = 1 S = 0)$) emails.	292
6.4	The new email as BOW. Note that the word “no” is missing in the training vocabulary. We ignore it here as we have no way of telling what would the corresponding probabilities be.	292
6.5	DTM of the three example emails from example 6.7. The first row (the first email) is spam, the following two are not spam.	295
6.6	Smoothed probabilities for the DTM above for $\alpha = 0.2$	295
6.7	AND, OR and XOR operations	300
6.8	<i>XOR</i> -perceptron parameters	303

7.1	Two BOW-s \mathbf{x}_1 and \mathbf{x}_2 , corresponding to the two Laozi quotes in the text. Both BOW-s, stacked horizontally underneath each other as in this table, form a numeric DTM that can be used in various machine learning models.	325
7.2	Example vocabulary, bag-of-word vectors, and TF-IDF transformation for quotes: “Knowing others is wisdom, knowing yourself is Enlightenment” and “Mastering others is strength. Mastering yourself is true power”.	328
7.3	GloVe most similar words	335
9.1	Principal components of 2-D data	373
9.2	Explained variance across PC-s	376
9.3	Principal components of data in Figure 9.13	381
9.4	Proportion of variance explained by components in Table 9.3	381
10.1	Two fictional movies rated by three fictional persons. <i>Average</i> is the users’ average rating over all movies they have rated.	388

List of Exercises

1.1 Of what measure type are these values?	5
1.2 Mean, median, mode	9
1.3 Compute Sample Quantiles	14
1.4 Rolling two dice	24
1.5 First class survivors	28
1.6 A family has two children...	31
1.7 First class given survived	32
1.8 Two bags of M&M	34
1.9 Smile or fights?	35
1.10 Rolling two dice	39
1.11 Find $\Pr(Z = 6)$	39
1.12 Expected value of die	41
1.13 How many sixes do we get?	41
1.14 Compute variance of a RV	44
1.15 Variance of Bernoulli RV	44
1.16 Measure for c.d.f	49
1.17 Quantiles of standard uniform distribution	54
1.18 Entropy of Bernoulli random variable	65
1.19 Damage in Allied bombers	86
1.20 How to multiply wealth of all Icelanders	89
2.1 Vector dimension	94
2.2 What is the capital of France?	95
2.3 Are these vectors linearly independent?	98
2.4 Test the associative property	110
2.5 Explain why Example 2.7 works	111
2.6 Find base vector multiplier for a given vector	115
2.7 Inverse of rotation matrix	119
3.1 Compute ϵ	127
3.2 Predict using linear regression	132
3.3 Income and education	135
3.4 How is sons' height related to fathers' height?	136
3.5 Interpreting regression table	139
3.6 Compute TSS, SSE, R^2	146
3.7 Do union members earn more?	156
3.8 Interpret multi-category dummies	158
3.9 Why a single race only?	158

3.10 Matrix form	173
3.11 Linear or logistic regression?	181
3.12 Which values are statistically significant?	184
3.13 Prove (3.2.9)	187
4.1 Self-selection and downward bias	197
4.2 Confounding factors and downward bias	197
4.3 Does smoking cause lung cancer?	198
5.1 Compute the confusion matrix	235
5.2 Confusion matrix for the naive model	235
5.3 Accuracy, Precision, Recall	236
5.4 Flipping positives and negatives	237
5.5 COVID test sensitivity	238
6.1 Cosine similarity	262
6.2 Cosine, angular distance are not proper metric distances	263
6.3 Compute entropy gain	276
6.4 Probability of spam given no “viagra”	283
6.5 Categorize using Naive Bayes	293
6.6 <i>OR</i> -perceptron	301
6.7 Use the perceptron for <i>XOR</i>	303
6.8 Softmax property	306
6.9 Corner detection with convolutions	310

Index

- $\mathbb{1}()$, *see* indicator function
80-20 rule, *see* pareto ratio, 60
- accuracy, 235
actual outcome, 208
AdaBoost, 280
agglomerative clustering, 366
alternative hypothesis, 67
and, 300
angular distance, 262
associated with, 137, 195
atomic event, 39
average, *see* mean
axon, 298
- bag of words, 325, 329
bagging, 279
before-after estimator, 216
Bernoulli process, 80
bimodal, 9
binary outcome, 181
bit, 64
bootstrap, 279
- case-control study, 204
categorization
 naive model, 235
causal diagram, 195
causal inference, 192
cause, 193
cause-density bias, 228
central limit theorem, 60, 62, 81
central tendency, 40
centroid, 363
Chebyshev norm, 101
chessboard norm, 101
CLT, *see* central limit theorem
cluster analysis, 359
- compound event, 22, 23
conditional expectation, 209
conditional probability, 26
confidence interval, 19, 66, 70, 72
confidence level, 67, 70, 71
confounding factor, 228
confounding factors, 197
confusion matrix, 232
context, 329, 332
continuous random variable, 40
contributing cause, 193
control for, 152
convolution, 307
 filter, 307
 kernel, 307
cosine similarity, 102, 260, 389
counterfactual, 208
counterfactual assumption, *see* identifying assumption
counterfactual outcome, *see* counterfactual
coverage error, 19
cross-sectional estimator, 213
cumulative distribution function, 49
curse of dimensionality, 215
- datasets
 Hubble, 138
 males, 155, 274, 413
 smokeban, 79, 414
 treatment, 234, 414
- decile, 14
decision boundary, 248
decision boundary plot, 270
decision tree, 267
 leaf, 268
 node, 267

- recursive binary splitting, **272**
- degrees of freedom, **12**, **73**, **77**, **138**
- dendrite, **298**
- dependent variable, *see* outcome variable
- design matrix, **93**, **173**, **174**
- discrete random variable, **39**
- distance metric, **264**
- distributions
 - bernoulli, **49**
 - binomial, **50**
 - discrete uniform, **50**
 - log-normal, **56**
 - normal, **55**
 - Pareto, **58**, **60**
 - uniform, **54**
- document-term-matrix, **292**, **325**
- dose, **192**, **194**, **211**
- double-blind experiment, **200**
- DTM, **323**, *see* document-term-matrix
- dummies, **155**, **159**
- ecological fallacy, **87**
- effect, **193**
- elasticity, **162**
- elbow plot, **365**
- embeddings, *see* word embeddings
- endogenous variable, *see* outcome variable
- ensemble method, **267**, **279**
- entropy, **63**, **275**
- Euclidean norm, **100**
- event, **21**
- expectation, **8**, **33**, **37**, **40**, **54**, **209**
- expected value, *see* expectation
- explainability, **268**
- explanatory variables, **126**
- external validity, **19**
- F-score, **236**
- factor loading, **373**
- fairness, **398**
- false negative, **69**, **234**
- false positive, **69**, **234**
- false positive rate, **237**, **238**
- fat tails, **256**
- feature normalization, **255**
- feature normalization, **328**
- feature selection, **354**
- feed-forward neural network, **298**
- fixed effects, **213**
- forward selection, **169**
- frequentist probability, **24**
- GAN, *see* generative adversarial network
- gaussian mixture, **367**
- gaussian mixture model, **367**
- generative adversarial network, **392**
- GloVe, **333**
- gradient, **342**, **407**
- gradient ascent, **341**, **393**
- gradient descent, **341**, **342**
- group fairness, **398**, **399**
- hidden layer, **301**
- hierarchical clustering, **366**
- hierarchical clustering, **360**
- histogram, **12**
- hyperparameters, **251**
- hyperplane, **149**
- identifying assumption, **208**, **209**, **217**, **220**
- independent events, **35**, **289**
- indicator function, **viii**, **327**
- individual fairness, **398**, **399**
- instance-based-learning, **265**
- intersectionality, **170**
- interval measure, **2**
- iris data, **124**, **366**
- k*-means, **362**, **362**
- k*-nearest neighbors, **243**, **263**, **263**
- law of large numbers, **47**, **60**
- lemma, **324**
- lemmatization, **324**
- likelihood, **290**
- linear regression
 - polynomial regression, **246**
- linear independence, **98**
- linear probability model, **180**, **239**
- linear regression
 - mean squared error, **143**
 - prediction, **130**
 - residual, **141**
 - SSE, *see* sum of squared errors

- sum of squared errors, 141, 143
- total sum of squares, 144
- Lipschitz continuity, 305
- log-likelihood, 290
- logistic function, 181, 305
- logistic regression, 181
 - link, 182
 - log-odds, 187
 - odds ratio, 187
- logistic transformation, 181
- logit, *see* logistic regression
- loss function, 337
- L_p -norm, 100
- LPM, *see* linear probability model
- Mahalanobis distance, 258
- majority voting, 263, 279, 280
- manhattan norm, 101
- marginal effect
 - logistic regression, 185
- matrix, 103
 - column, 104
 - component, *see* element
 - condition number, 116, 257
 - diagonal, 105
 - diagonal matrix, 105
 - dimension, 104
 - eigenvalue decomposition, 258
 - element, 104
 - identity matrix, *see* unit matrix
 - index, 104
 - left-multiplication, 110
 - lower triangle, 105
 - multiplication, *see* product
 - post-multiplication, *see* left-multiplication
 - see also* right-multiplication
 - product, 107
 - right-multiplication, 110
 - rotation matrix, 118
 - row, 104
 - square, 105
 - square matrix, 105
 - symmetric matrix, 105
 - trace, 113
 - transposition, 106
 - unit matrix, vii, 106
- upper triangle, 105
- maximum likelihood, 183
- mean, 8
- mean independence, 209
- measure level, 2
- median, 3, 8, 14
- metric, 254
- metric distance, 102
- min-max scaling, 257
- Minkowski norm, *see* L_p -norm
- mode, 3, 9
- MSE, *see* mean squared error
- multi-layer perceptron, 303
- multimodal, 9
- mutually exclusive events, 24
- naive bayes, 285
- nat, 64
- natural experiment, 202
- nearest neighbors, 254, 263
- neural networks
 - activation, 299
 - bias, 299
 - input layer, 299
 - perceptron, 298
 - weights, 299
- neuron, 298
- nominal measure, 2
- non-linear optimization, 337
- norm, 99
- normalizer, 32
- null hypothesis, 67, 70
- objective function, 338
- observed value, 37
- one-hot encoding, 157
- one-tailed confidence interval, 73
- optimization, 338
- or, 300
- ordinal measure, 2
- outcome, 194
- outcome variable, 126
- outcome-density bias, 228
- overfitting, 246
- p*-value, 68, 70, 138
- p.m.f, *see* probability mass function

- padding, 313
 paired data, 79
 pareto ratio, 15
 PCA, *see* principal component analysis
 pdf, *see* probability density function
 penalty, 356
 percent, 4
 percentage point, 4, 187
 percentile, 13
 pixel, 316
 pooling, 312
 population, 18
 population variance, 12
 posterior, 32
 precision, 236, 283
 principal component analysis, 370
 prior, 32
 probability, 24
 probability density function, 53, 56
 probability mass function, 48
 pruning, 277

 QSR, *see* quintile share ratio
 quantile, 3, 13, 72
 quartile, 14
 quasi-experiment, 202
 quintile, 14
 quintile share ratio, 15

 R-squared, 144
 R^2 , 144
 random variable, 37, 71
 continuous, 52
 function,
 textbf{44}
 randomized controlled trial
 see RCT, 200
 ratio measure, 2
 RCT, 200
 realization, 37, 71
 recall, 236, 283
 reference category, 159
 regression
 coefficients, 127
 constant, *see* intercept
 cross-effect, *see* interaction effect
 deviation, 132

 error term, 127, 149
 explanatory variable, 149
 interaction effect, 165, 224
 interaction term, 166
 intercept, 128, 134
 log transform, 160
 log-log transform, 162
 multiple regression, 149
 outcome variable, 149
 parameters, 127
 reference category, 157
 regression plane, 150
 regression to mean, 424
 residual, 132
 slope, 128, 134
 standardized features, 160, 168
 regularization, 354
 ReLU, 305
 RMSE, *see* root mean squared error
 robust statistic, 8
 ROC curve, 238
 root mean squared error, 143, 247, 249,
 275
 RV, *see* random variable

 sample, 18
 sample space, 21
 sample variance, 10
 sampling error, 19
 scalar, 94
 self-selection, 196, 197
 sensitivity, 237
 sigmoid function, 181, 305
 significance level, 68, 70, 73, 74
 simple event, 22, 23, 85
 softmax, 305
 specificity, 237
 SSE, 174
 standard deviation, 10
 standard error, 10
 statistical discrimination, 398
 statistical hypothesis, 66
 statistical model, 126
 stemming, 323
 stochastic, 21
 stopwords, 324, 326

stride, **313**
supervised learning, **126**
support vector machine, **296**
SVM, *see* support vector machine

t-statistic, **69**
t-value, **138**
t-value table, **73**
TCM, *see* term co-occurrence matrix
term co-occurrence matrix, **329**
tertile, **14**
tesseract, **120**
test statistic, **68**, **70**
TF-IDF, **323**, **327**
token, **323**
tokenization, **323**
training data, **245**, **249**
treatment, **194**
tree, *see* decision tree
true negative, **234**
true positive, **233**
true positive rate, **237**, **238**
two daughter problem, **31**
two daughter problem, **84**
type-I error, **69**, **70**, **234**
type-II error, **69**, **70**, **234**

underfitting, **249**
unimodal, **9**
unsupervised learning, **126**

validation data, **249**
variance, **12**, **43**
vector, **92**
 column vector, **106**, **112**
 component, *see* element
 dimension, **93**
 element, **93**
 inner product, **112**
 linear combination, **97**
 norm, **99**, **100**, **261**
 outer product, **112**
 row vector, **106**
 row vector, **112**
vector space, **96**
 dimension, **98**
Venn diagram, **25**

Bibliography

- Amoros, E., Chiron, M., Martin, J.-L., Thélot, B. and Laumon, B. (2012) Bicycle helmet wearing and the risk of head, face, and neck injury: a french case-control study based on a road trauma registry, *Injury Prevention*, **18**, 27–32.
- Angwin, J., Larson, J., Mattu, S. and Kirchner, L. (2016) Machine bias: There's software used across the country to predict future criminals. and it's biased against blacks., ProPublica website, proPublica.
- Bender, E. M., Gebru, T., McMillian-Major, A. and Shmitchell, S. (2021) On the dangers of stochastic parrots: Can language models be too big?, in *Conference on Fairness, Accountability, and Transparency (FAccT '21)*, pp. 610–623.
- Bonten, M. J., Huijts, S. M., Bolkenbaas, M., Webber, C., Patterson, S., Gault, S., van Werkhoven, C. H., van Deursen, A. M., Sanders, E. A., Verheij, T. J., Patton, M., McDonough, A., Moradoghli-Haftvani, A., Smith, H., Mellelieu, T., Pride, M. W., Crowther, G., Schmoele-Thoma, B., Scott, D. A., Jansen, K. U., Lobatto, R., Oosterman, B., Visser, N., Caspers, E., Smorenburg, A., Emini, E. A., Gruber, W. C. and Grobbee, D. E. (2015) Polysaccharide conjugate vaccine against pneumococcal pneumonia in adults, *New England Journal of Medicine*, **372**, 1114–1125, PMID: 25785969.
- Bottou, L., Curtis, F. and Nocedal, J. (2018) Optimization methods for large-scale machine learning, *SIAM Review*, **60**, 223–311.
- Boyd, D. and Crawford, K. (2012) Critical questions for big data, *Information, Communication & Society*, **15**, 662–679.
- Correia, S., Luck, S. and Verner, E. (2020) Pandemics depress the economy, public health interventions do not: Evidence from the 1918 flu, Tech. rep., SSRN.
- Craven, D. (2015) The statistical sins of jeremy hunt, *BMJ*, **351**.
- Cripton, P. A., Dressler, D. M., Stuart, C. A., Dennison, C. R. and Richards, D. (2014) Bicycle helmets are highly effective at preventing head injury during head impact: Head-form accelerations and injury criteria for helmeted and unhelmeted impacts, *Accident Analysis & Prevention*, **70**, 1 – 7.
- Deming, D. J. (2017) The growing importance of social skills in the labor market, *Quarterly Journal of Economics*.

- Elvidge, C. D., Baugh, K., Zhizhin, M., Hsu, F. C. and Ghosh, T. (2017) Viirs night-time lights, *International Journal of Remote Sensing*, **38**, 5860–5879.
- Ferdinands, J. M., Olsho, L. E. W., Agan, A. A., Bhat, N., Sullivan, R. M., Hall, M., Mourani, P. M., Thompson, M. and Randolph, A. G. (2014) Effectiveness of Influenza Vaccine Against Life-threatening RT-PCR-confirmed Influenza Illness in US Children, 2010–2012, *The Journal of Infectious Diseases*, **210**, 674–683.
- Ferté, T., Ramel, V., Cazanave, C., Lafon, M.-E., Bébéar, C., Malvy, D., Georges-Walryck, A. and Dehail, P. (2021) Accuracy of covid-19 rapid antigenic tests compared to rt-pcr in a student population: The studycov study., *Journal of clinical virology : the official publication of the Pan American Society for Clinical Virology*, **141**, 104878.
- Fleisher, W. (2021) What's fair about individual fairness?, in *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, Association for Computing Machinery, New York, NY, USA, AIES '21, p. 480–490.
- Freemantle, N., Ray, D., McNulty, D., Rosser, D., Bennet, S., Keogh, B. E. and Pagano, D. (2016) Increased mortality associated with weekend hospital admission: a case for expanded seven day services?, *BMJ*, **352**.
- Fyhri, A., Sundfør, H., Weber, C. and Phillips, R. (2018) Risk compensation theory and bicycle helmets – results from an experiment of cycling speed and short-term effects of habituation, *Transportation Research Part F: Traffic Psychology and Behaviour*, **58**, 329 – 338.
- Galton, F. (1886) Regression towards mediocrity in hereditary stature., *The Journal of the Anthropological Institute of Great Britain and Ireland*, **15**, 246–263.
- Godlee, F. (2016) How jeremy hunt derailed clinician led progress towards a seven day nhs, *BMJ*, **352**.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014) Generative adversarial nets, in *Advances in Neural Information Processing Systems* (Eds.) Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence and K. Q. Weinberger, Curran Associates, Inc., vol. 27.
- Greene, W. H. (2003) *Econometric Analysis*, Prentice Hall.
- Hubble, E. (1929) A relation between distance and radial velocity among extra-galactic nebulae, *Proceedings of the National Academy of Sciences*, **15**, 168–173.
- Iacobucci, G. (2016) Demonstrating junior doctors ask jeremy hunt to stop misusing statistics, *BMJ*, **352**.
- Jackson, S. (2022) A group of bipartisan lawmakers is grilling Amazon for its continued sale of a chemical compound used in suicides, *Business Insider*, February 22.

- Jacobs, A. Z. and Wallach, H. (2021) Measurement and fairness, *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*.
- Kleinberg, J., Lakkaraju, H., Leskovec, J., Ludwig, J. and Mullainathan, S. (2018) Human decisions and machine predictions, *The Quarterly Journal of Economics*, **133**, 237–293.
- Kleinberg, J., Mullainathan, S. and Raghavan, M. (2016) Inherent trade-offs in the fair determination of risk scores, Tech. rep., arXiv.
- Liao, Q. V., Gruen, D. and Miller, S. (2020) Questioning the ai: Informing design practices for explainable ai user experiences, in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, Association for Computing Machinery, New York, NY, USA, CHI '20, p. 1–15.
- Markel, H., Lipman, H. B., Navarro, J. A., Sloan, A., Michalsen, J. R., Stern, A. M. and Cetron, M. S. (2007) Nonpharmaceutical interventions implemented by US cities during the 1918-1919 influenza pandemic, *JAMA*, **298**, 644–654.
- Matute, H., Blanco, F., Yarritu, I., Díaz-Lago, M., Vadillo, M. A. and Barberia, I. (2015) Illusions of causality: how they bias our everyday thinking and how they could be reduced, *Frontiers in psychology*, **6**, 1–14.
- Mills, N. and Gilchrist, A. (2008) Oblique impact testing of bicycle helmets, *International Journal of Impact Engineering*, **35**, 1075 – 1086.
- Murphy, K. P. (2012) *Machine Learning: A Probabilistic Perspective*, MIT Press, Cambridge, MA.
- of England, B. (2019) Inflation report, february 2019, Tech. rep., Bank of England, London, UK.
- Pennington, J., Socher, R. and Manning, C. D. (2014) Glove: Global vectors for word representation, in *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543.
- Smith, G. C. and Pell, J. P. (2003) Parachute use to prevent death and major trauma related to gravitational challenge: systematic review of randomised controlled trials., *BMJ*, **327**, 1459–1461.
- Walker, I. (2007) Drivers overtaking bicyclists: Objective data on the effects of riding position, helmet use, vehicle type and apparent gender, *Accident Analysis & Prevention*, **39**, 417 – 425.