

MASARYKOVA UNIVERZITA
PŘÍRODOVĚDECKÁ FAKULTA
ÚSTAV MATEMATIKY A STATISTIKY

Diplomová práce

BRNO 2024

VÁCLAV PECA

Pouliční matematika

Diplomová práce

Václav Peca

Bibliografický záznam

Autor:	BcA. Bc. Václav Peca Přírodovědecká fakulta, Masarykova univerzita Ústav matematiky a statistiky
Název práce:	Pouliční matematika
Studijní program:	Učitelství matematiky pro střední školy
Studijní obor:	Učitelství matematiky pro střední školy Učitelství anglického jazyka a literatury pro střední školy
Vedoucí práce:	Mgr. Michal Bulant, Ph.D.
Akademický rok:	2023/2024
Počet stran:	ix + 75
Klíčová slova:	samoopravné kódy; teorie kódování; inferenční statistika; testování statistických hypotéz; Bayesovská indukce; didaktika matematiky; badatelsky orientovaná výuka; problémové vyučování; odvozování z prvních principů; smysluplnost učiva

Bibliographic Entry

Author:	BcA. Bc. Václav Peca Faculty of Science, Masaryk University Department of Mathematics and Statistics
Title of Thesis:	Street mathematics
Degree Programme:	Teaching of Mathematics for Secondary Schools
Field of Study:	Teaching of Mathematics for Secondary Schools Upper Secondary School Teacher Education in English Language and Literature
Supervisor:	Mgr. Michal Bulant, Ph.D.
Academic Year:	2023/2024
Number of Pages:	ix + 75
Keywords:	error-correcting codes; coding theory; inferential statistics; statistical hypothesis testing; Bayesian induction; didactics of mathematics; inquiry-based learning; problem-based learning; reasoning from first principles; meaningfulness of learning

Abstrakt

V diplomové práci představujeme učební text o aplikacích matematiky určený pro studenty středních škol. Text se skládá ze dvou kapitol věnovaných dvěma různým oblastem, a to teorii samoopravných kódů a inferenční statistice. Didakticky je učivo podáno tzv. „problem-first“ přístupem – student je veden k samostatnému řešení problému a odvození nezbytné matematické teorie.

Abstract

In this thesis, we present an extracurricular applied mathematics textbook for high-school students. The textbook consists of two chapters dealing with two different areas: error-correcting codes and inferential statistics. Didactically, the material is presented in a ‘problem-first’ approach—the student is guided to tackle the problem independently and derive the necessary mathematical theory.

ZADÁNÍ
DIPLOMOVÉ PRÁCE

Akademický rok: 2023/2024

Ústav:	Přírodovědecká fakulta
Student:	BcA. Bc. Václav Peca
Program:	Učitelství matematiky pro střední školy
Specializace:	Učitelství matematiky pro střední školy Učitelství anglického jazyka a literatury pro střední školy

Ředitel ústavu PřF MU Vám ve smyslu Studijního a zkušebního řádu MU určuje diplomovou práci s názvem:

Název práce:	Pouliční matematika
Název práce anglicky:	Street mathematics
Jazyk závěrečné práce:	angličtina

Oficiální zadání:

Student zpracuje výukový text, využitelný ve výuce na střední škole, který provede studenty možnými způsoby řešení (či odhadu řešení) matematických problémů bez přílišného formalismu. Diplomant na základě doporučené literatury vybere vhodná témata, která doplní vlastními náměty ze své zkušenosti či praxe včetně možného využití dalších zdrojů.

Literatura:

GRAHAM, Ronald L., Oren PATASHNIK a Donald Ervin KNUTH. *Concrete mathematics : a foundation for computer science*. 2nd ed. Reading, Massachusetts: Addison-Wesley Publishing Company, 1994, xiii, 656. ISBN 0201558025.

Street-fighting mathematics the art of educated guessing and opportunistic problem solving. Edited by Sanjoy Mahajan. Cambridge, MA: MIT Press, 2010, xv, 134 p. ISBN 9780262514293.

ZEITZ, Paul. *The art and craft of problem solving*. 2nd ed. Hoboken, N.J.: John Wiley & Sons, 2007, xviii, 366. ISBN 9780471789017.

Vedoucí práce:	Mgr. Michal Bulant, Ph.D.
Datum zadání práce:	11. 9. 2022
V Brně dne:	6. 5. 2024

Zadání bylo schváleno prostřednictvím IS MU.

BcA. Bc. Václav Peca, 19. 10. 2022

Mgr. Michal Bulant, Ph.D., 27. 10. 2022

RNDr. Jan Vondra, Ph.D., 27. 10. 2022

Poděkování

Na tomto místě bych chtěl poděkovat panu Mgr. Michalu Bulantovi, Ph.D., za vedení práce a mnoho cenných rad a připomínek, které mi poskytl během psaní práce.

Prohlášení

Prohlašuji, že jsem svoji diplomovou práci vypracoval samostatně pod vedením vedoucího práce s využitím informačních zdrojů, které jsou v práci citovány.

Brno 6. května 2024

.....
Václav Peca

Contents

Notation	ix
Introduction	1
 I The Textbook	 3
Preface	4
Chapter 1. Error-correcting codes	6
1.1 What is the goal?	6
1.2 Pushing the effectiveness of single-error correction	8
1.3 Proper definitions	12
1.4 Linearity	17
1.5 The power of polynomials	22
1.6 Counting in circles	29
1.7 Reed–Solomon codes	36
Chapter 2. Inferential statistics	40
2.1 Data types	40
2.2 Describing the samples	43
2.3 From frequencies to probabilities	46
2.4 Testing simple hypotheses	52
2.5 Experiment design and non-parametric testing	57
2.6 The Bayesian approach	62
 II Theoretical Commentary	 66
Chapter 3. Didactic commentary	67
3.1 Mathematical rigor and good explanations are not the same thing ..	67
3.2 Effective explanations	68
3.2.1 Terminology	68
3.2.2 Simplifying foundations	68
3.2.3 Analogies	69

3.3 Problem-first approach	70
3.4 Narrative style	71
3.5 A smoother transition to high mathematics	71
Conclusion	73
Bibliography	74

Notation

For easier orientation we provide an overview of the basic notation used in the thesis.

$x \in A$	x is an element of the set A
$x \notin A$	x is not an element of the set A
$A \cup B$	set union; the set of elements that are in at least one of the sets A, B
$A \cap B$	set intersection; the set of elements that are in both sets A, B
$A \setminus B$	set difference; the set of elements in A that are not in B
$ A $	cardinality of A ; the number of elements in the set A
$f: A \rightarrow B$	map f from the set A to the set B
$x \mapsto y$	x maps to y
(x, y)	ordered pair
$\sum_{i=0}^n x_i$	sum of the terms x_i for integers i from 0 through n (inclusive)
$\binom{n}{k}$	binomial coefficient; the number of k -element subsets of an n -element set (read ‘ n choose k ’)
$\lfloor x \rfloor$	floor of x ; the largest integer less than or equal to x
$\lceil x \rceil$	ceiling of x ; the smallest integer greater than or equal to x
$x \bmod m$	the remainder after dividing x by m
$\approx$	is approximately equal to
$\mathbb{N}$	the set of all natural numbers (excl. 0)
$\mathbb{Z}$	the set of all integers
$\mathbb{Q}$	the set of all rational numbers
$\mathbb{R}$	the set of all real numbers
$\mathbb{R}_0^+$	the set of all non-negative real numbers

Introduction

It is customary for high-school students to ask in math classes: *What is this good for?* Mathematics is sometimes viewed as a subject detached from reality, enjoyed by impractical nerds pondering on useless problems. However, a large portion of the discoveries in mathematics were driven by practical use, and the majority of people using mathematics are not theoretical mathematicians, but engineers, data scientists, and professionals in various fields.

The thesis aims to create two chapters of an extracurricular applied mathematics textbook for high-school students, based on a problem-first approach. Unlike traditional mathematical textbooks, at the forefront of each chapter is a problem, not a tool. Via tackling the problem, the student is led to realize and understand the necessity of certain mathematical tools and constructs. Owing to the title of the thesis, the language used is colloquial and imitates the way a high-school teacher might talk to a student. The textbook is suitable for self-study—the learner should not need any teacher help on top of the guidance of the text.

The title of the thesis refers to the textbook *Street-fighting Mathematics* [13] and a course of the same name created and taught by Sanjoy Mahajan at MIT. While Mahajan’s primary focus is methods of approximate solutions, the textbook chapters proposed in this thesis provide a more foundational look at two highly practical fields of mathematics: coding theory and inferential statistics. The content assumes the knowledge of high-school mathematics, and the ability to grasp concepts slightly above that level. The text should however be more accessible than the average source for these mathematical topics, since most of the sources build on a solid understanding of undergraduate-level math, such as linear algebra, ring theory, or calculus.

The thesis is divided into two parts: the textbook and a theoretical commentary. The textbook starts with a preface, which intends to tune the student in to the style of the textbook, explaining the structure and the recommended way of reading. The first chapter of the textbook delves into the problem of error correction. It starts with understanding the general goal and leads the student to construct codes from the simplest to more sophisticated, and seeing the various challenges and trade-offs in the code design along the way. The second chapter concerns with the problem of inferring knowledge from data. It explains how various probability distributions emerge, then proceeds to different methods of hypothesis testing,

and finally compares the frequentist and Bayesian approaches to inference. The theoretical commentary serves to explain the didactic principles and the design decisions made when writing the textbook.

The thesis should serve all students who are interested in mathematics beyond the scope of what is taught in high school. It intends to bridge the gap between high school and university mathematics—the two topics were chosen so that they motivate the introductory mathematics courses in university (algebra and calculus). Secondly, the thesis may serve all teachers of mathematics, giving them ideas how to use the problem-first approach in their teaching, to make mathematics more enjoyable and easily understood.

Part I

The Textbook

Preface

You, the reader of this text, were probably born within a few decades of the year 2000. So, we share one crucial experience: being born into a world brimming with knowledge. The vast amounts of knowledge were thrust upon us before we began forming our own.

Mathematics is a perfect example. We were told about numbers before we would have discovered them ourselves. You might think that you have understood numbers intuitively since birth—but that’s likely not the case. As partial evidence, there is at least one indigenous tribe in the Amazon rainforest that has no words for numbers in its language at all [7].

Mathematics, along with all other fields of human knowledge, took millennia to develop. The way mathematics is organized today, built on the solid foundation of set theory, having notation that allows us to think about abstract notions using a visual system—all is a result of immense mental effort.

So, as we were born into this world and learned maths from our ancestors, we took a massive shortcut. Imagine being one of those unfortunate feral children raised by wolves. You could possess Einstein’s exact genetic makeup and still never arrive at the theory of relativity. Maybe you could go as far as Euclid, pondering problems in geometry. But there’s no way you could develop the theory of relativity, from scratch, in a single lifetime.

That we were born into this world is a tremendous stroke of luck. Yet, this shortcut has a flaw in how knowledge is passed from teacher to student. Suppose the teacher (who once was a student) spent her career exploring a specific question. How she tackled it taught her a lot about it and allowed her to develop a good model of the problem. Eventually, she created formalism that helped describe and solve the problem. In the end, the teacher has a handy toolbox of definitions and algorithms to pass on to her students. So, what’s the catch? The student doesn’t grasp the *why*—he lacks the teacher’s experience and doesn’t understand the necessity of these constructs. In this case, the student can ask the teacher and have any detail explained, since the teacher is the one who created the toolbox. However, as the famous phrase ‘So said al-Khwarizmi’—from which the very word algorithm originated—illustrates, the common mathematical toolboxes are rarely taught by their creators.

This is what happens in math classes in Czech schools. Students are trained to perform complex algorithms that the knowledgeable ancestors selected as vital, such as finding the point-normal form of the line tangent to a hyperbola. But the bigger picture of where the theory comes from and its purpose is often skipped. Talented students can connect the dots, while others struggle to memorize the algorithms, aware that they don't fully understand them. It's no wonder most students don't enjoy mathematics.

Grant Sanderson, the author of the explainer math videos on the YouTube channel 3Blue1Brown, makes the distinction between *textbook progression* and *discovery progression* [17]. While textbooks start with theory and work toward applications, the discovery process does the opposite: it starts with a real-world problem and develops a theory to solve it. I still don't understand why textbooks can't follow the discovery progression. Do their authors strive for minimal content, devoid of anything not strictly necessary? Or does deductive reasoning just feel more like serious mathematics?

The textbook you are about to read invites you to make the discoveries for yourself. When you encounter a question in the text, pause for a moment, think deeply, and try to formulate an answer. Often, you'll need a pen and paper to derive or sketch something out. If you can't figure it out, don't worry; continue reading. The most crucial part is to attempt the problem because trying will help you grasp these abstract concepts more easily.

Chapter 1

Error-correcting codes

Imagine that you would like to send someone an important digital message. The problem is that on the way, the message can get damaged—but it's crucial that the recipient gets it in its original form.

At first, that sounds like an engineering problem, not a mathematical one. However, we aren't going to find ways to avoid the damage, instead, we are going to find ways to add *redundancy* to the message so that even when slightly damaged, we can repair it to its initial state.

These mathematical tools that we are going to explore are called error-correcting codes, and they've been around since people started thinking about telecommunications. Today, these codes are everywhere that data is stored or transferred through wire or air.

1.1 What is the goal?

First, let's define a few basic terms that we will be using in this chapter.

An **alphabet** is a finite set of symbols that we can use to write something down.

A **word** or a **string** is a finite sequence of symbols from a given alphabet. Both terms have the same meaning, they are only used in different contexts.

A **message** is a string that we are sending to someone.

The message is transmitted through a noisy communications channel—that means that sometimes it happens that one symbol is mistaken for another, an event that we call an **error**. Each occasion of that event counts as one error.

Example.

Alphabet: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

Sent message: 06592271

Transmitted message: 08592217

During transmission, three errors occurred on positions 2, 7 and 8 of the message.

To have control over the errors, we use an **error-correcting code**. For a well-defined error-correcting code, we can tell how many errors it can detect and how many errors it can correct.

- You want to send the message 1997. Design a method that will encode this message into a longer string and allow the recipient to find out if one of the symbols was changed during the transfer because of an error.

The simplest way to solve this, without constraints on the length of the message, is to duplicate each of the symbols, so the resulting string would be 11999977. Knowing that an error occurred is as simple as finding that any of the pairs has differing symbols in it.

So, this code can detect 1 error, however, it does not allow the recipient to correct it. If they for example receive the string 11998977, they do not know if the original message was 1987 or 1997.

An alternative way of doing this may be using a checksum. Count the digital sum ($1 + 9 + 9 + 7 = 26$) and take its remainder after division by 10 ($26 = 2 \cdot 10 + 6$). Append this check symbol to the message: 19976. If you change any one of the symbols, the check symbol will reveal that there was an error—but not only we cannot correct it, we don't even know on which position it happened.

- You want to send the message 1997. This time, design a method that will not only allow the recipient to detect the error, but also to correct it.

The solution is very similar to the previous one. Instead of doubling each symbol, triple it. In our example, that would be 111999999777. If one error happens, the receiver can spot one digit in the triplets being different and correct it to be the same as the other two.

This class of error-correcting codes is called **repetition codes**.

- Consider the triple-repetition code with a fixed message length of 4 symbols. How many strings of length 12 qualify as errorless encoded messages? How many qualify as an encoded message with 1 error? What fraction of the whole space of strings of length 12 is that?

The set of all triple-repetition encoded messages of length 4 looks like this:

```
000000000000,
000000000111,
000000000222,
000000000333,
000000111000,
    ⋮
999999999888,
999999999999.
```

It contains $10^4 = 10\,000$ items in total.

For each errorless encoded message, there are $12 \cdot 9 = 108$ messages with exactly one error, because we are changing a symbol on 1 of 12 positions to 1 of 9 alternative symbols.

That together makes $10\,000 + 1\,080\,000 = 1\,090\,000$ readable messages.

But how many 12-symbol strings are there? $10^{12} = 1\,000\,000\,000\,000$. So, the readable messages account for only 0,0001 % of the space of all possible messages.

- *Looking at the numbers we just calculated, why is the repetition code ineffective?*

Obviously, the problem is that 0,0001 % is such a tiny fraction. We wasted a lot of space for cases that we don't expect—since our code is designed to correct 1 error, we don't expect to receive a string like 426285251596.

So, tripling length while being able to correct only one error reliably is just not a good deal on the market with error-correcting codes. Surely we can find better ones!

1.2 Pushing the effectiveness of single-error correction

In the previous section, we came to the conclusion that the repetition code is far from efficient. We'll try to construct a code that can still correct one error, but using less check symbols.

It will be useful to move our attention to binary codes. Not only is encoding in binary signal usually necessary due to the nature of the technology used. But there is one big advantage from the code-designer perspective: an error is always a bit-flip ($0 \mapsto 1, 1 \mapsto 0$). So, locating the position of the error is sufficient for correcting it.

- Design a code for a 16-bit message that can correct 1 error by locating it via multiple checksums. (Hint: arrange the message in a 4×4 matrix and don't forget that the checksums must also be in binary.)

The idea that should come to mind is that we do a checksum for each row and each column. The checksums are counted mod 2, so they indicate the parity of the sum—hence we tend to call them parity checks.

When an error occurs in the message, one row parity check and one column parity check will fail, pointing us at the precise location of the error.

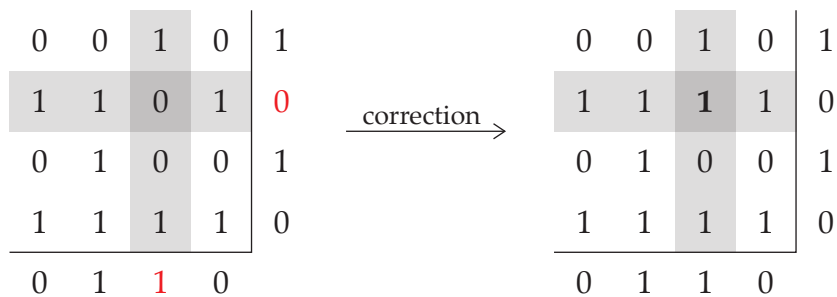


Figure 1.1: Correction of a message symbol

But what if the error occurs in a parity check symbol? Well, it will not affect any other symbol, unlike an error in a message symbol. So, when only one parity check fails, we know to correct the check symbol itself.

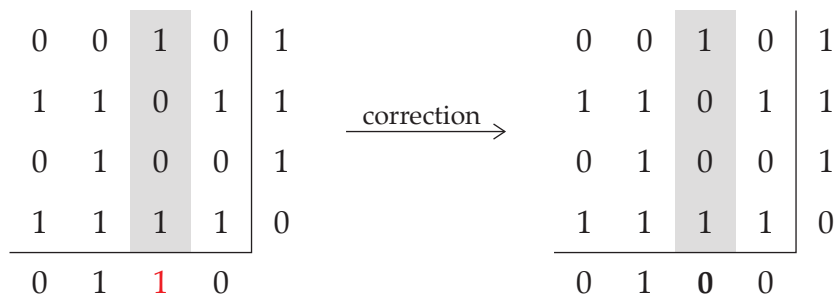


Figure 1.2: Correction of a check symbol

The beauty of this coding is that we can just check that every row and every column has an even number of 1's (including the checksum row and column). Odd number means that an error happened in that line.

- Which 24-bit words are not decodable?

There are three cases that *are* decodable:

- all parity checks succeed (no error)
- 1 parity check fails (1 error in a check symbol)
- 1 row parity check fails and 1 column parity check fails (1 error in a message symbol)

The other cases we cannot decode. They are exactly those in which at least 2 row parity checks or at least 2 column parity checks fail.

- *As we did with the triple repetition code, determine what fraction of all 24-bit words are decodable.*

In the case of the 16-bit message, $\frac{2^{16} \cdot (1+24)}{2^{24}} = 9.8\%$ of the space of all possible messages is readable. That is much better than the triple repetition code, but still, 90.2 % of the space is wasted.

The next step is to move from 9.8 % all the way to 100 %. So, for each message, all states of the check symbols should be decodable—each state pointing at a different position of a single error.

- *Design a code with 4 message symbols and 3 parity check symbols, where the parity checks can produce all variants of success (S) / failure (F) when 0 or 1 errors occur (SSS, SSF, SFS, SFF, FSS, FSF, FFS, FFF).*

We'll use an analogy to make thinking about this simpler. Imagine that the message symbols are prisoners and the check symbols are guards. Each guard has his own button that can be pressed to trigger a prison alarm. Each guard gets the instruction to watch *some* subset of the prisoners. If any of the prisoners is misbehaving (i.e. an error occurs in that position), precisely the guards that are watching this prisoner press their alarm button (i.e. the parity check fails). As a bonus, each guard is also watching himself and presses the alarm button even if he himself is misbehaving.

When any person in the prison misbehaves, we want to be able to locate the person based on the information of which buttons were pressed. To design such code means to select the subsets that the guards are watching.

We can visualize this with a table, where in the top row, we have the list of all people in the prison, and in the left column, we have the list of guards. A tick represents that the guard (row) watches the prisoner / guard (column).

First, we know that the guards are watched only by themselves and not anyone else, so that part of the table is done.

Now, we need to put ticks in the other part of the table so that each column has a unique pattern of ticks. Only then can each person be identified by a pattern of pressed buttons. Also, each prisoner must be watched by at least one guard.

	P ₁	P ₂	P ₃	P ₄	G ₁	G ₂	G ₃
G ₁					✓		
G ₂						✓	
G ₃							✓

Figure 1.3: Step 1

You should end up with a table similar to this, except for the order of rows, which is arbitrary. If you look at the column patterns of ticks, they are numbers 1–7 in binary, which makes perfect sense.

	P ₁	P ₂	P ₃	P ₄	G ₁	G ₂	G ₃
G ₁	✓	✓		✓	✓		
G ₂	✓		✓	✓		✓	
G ₃		✓	✓	✓			✓

Figure 1.4: Step 2

If we order the columns by the numbers in binary (so guards take positions 1, 2 and 4), we have a design that is easy to decode: the parity checks of positions {4, 5, 6, 7}, {2, 3, 6, 7} and {1, 3, 5, 7} interpreted as a binary number give us the position of the error!

	G ₁	G ₂	P ₁	G ₃	P ₂	P ₃	P ₄
G ₁	✓		✓		✓		✓
G ₂		✓	✓			✓	✓
G ₃				✓	✓	✓	✓

Figure 1.5: Step 3

Example.Alphabet: $\{0, 1\}$

Message: 0100

Encoding: $((s_1 + s_2 + s_4), (s_1 + s_3 + s_4), (s_1), (s_2 + s_3 + s_4), (s_2), (s_3), (s_4)) =$
 $= ((0 + 1 + 0), (0 + 0 + 0), (0), (1 + 0 + 0), (1), (0), (0))$

Encoded message: 1001100

An error occurs on position 6 of the encoded message.

Transmitted message: 1001110

Decoding: $((z_4 + z_5 + z_6 + z_7), (z_2 + z_3 + z_6 + z_7), (z_1 + z_3 + z_5 + z_7)) =$
 $= ((1 + 1 + 1 + 0), (0 + 0 + 1 + 0), (1 + 0 + 1 + 0)) = (1, 1, 0)$ Error occurred on position $110_2 = 6_{10} \implies$ flip position 6.

Corrected message: 1001100

Without check symbols: 1001100 = 0100

Finally, we've created a code that uses 100 % of the space. This code belongs to the family called **Hamming codes**, named after their inventor and the coding theory pioneer *Richard W. Hamming*. This specific code is called the Hamming(7,4) code, for it has a length of 7 symbols, 4 of which are message symbols. You can try constructing codes for longer messages (must be powers of 2) analogically.

1.3 Proper definitions

We've successfully constructed some error-correcting codes. However, we only worked with an intuitive concept of what a code is, we haven't defined it in mathematical terms at all. Let's fix it now.

A little disclaimer before we start: we'll restrict ourselves to **block codes** only. Those are the codes that encode messages in blocks of fixed length like we did previously. There are also **convolutional codes**, which we can imagine as an endless stream of symbols—we won't deal with them in this text.

So, what defines a code? Well, for sure we need a fixed alphabet—because if two codes are based on different alphabets (especially different *size* alphabets), they are not going to be the same. We'll also need a fixed block length. The alphabet together with the block length creates a set of all possible words. The code is then defined by selecting some of those words to be **codewords**.

Let's see that again in set-theory-speak.

Definition. Let A be a non-empty set, and let n be a natural number. An arbitrary subset $C \subseteq A^n$ is called a **code**.

I can hear you asking: *Is that really everything? Where did the process of encoding disappear? And what about the error correction?*

Good questions. In fact, both encoding and error correction are more of a technicality. The important and non-trivial part of coding theory is to find a set of codewords with the right properties. That's why it's useful to have only that in the definition and treat the rest separately.

- *How can we know, given this definition of code, how many errors a code can detect?*

We detect error(s) when the received message is not a codeword.

Let's start with an easy case. How do we know that a code can detect one error? Well, any two codewords must differ on at least two positions, so that when one error occurs in a codeword, we definitely do not end up with another codeword.

Analogously, a code can detect up to two errors if any two codewords differ on at least three positions. So, a code can detect up to t errors if any two codewords differ on at least $t + 1$ positions.

Instead of saying that two words differ on p positions, we can say their **Hamming distance** is p .

Definition. Let A be a non-empty set, and let n be a natural number. Consider the words $\mathbf{a} = (a_1, a_2, \dots, a_n), \mathbf{b} = (b_1, b_2, \dots, b_n) \in A^n$. The **Hamming distance** between $\mathbf{a}$ and $\mathbf{b}$ is defined as

$$d(\mathbf{a}, \mathbf{b}) = \left| \{i \in \{1, 2, \dots, n\} \mid a_i \neq b_i\} \right|.$$

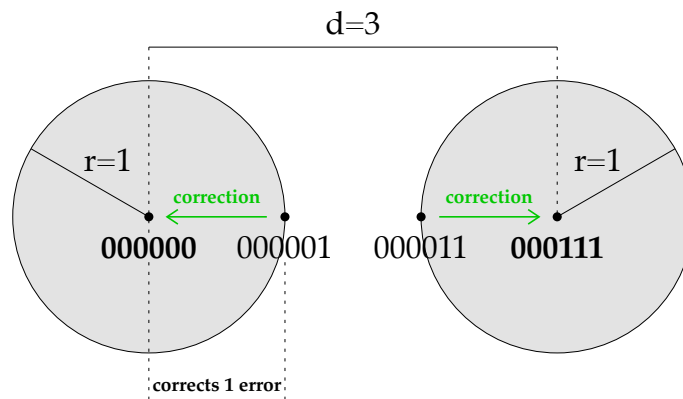
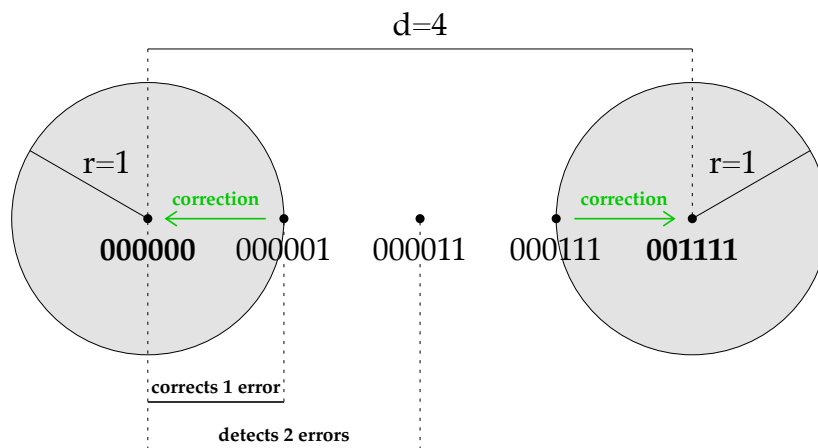
And since a code is just a set of codewords, we can also speak about the **minimal distance of a code**: the minimum Hamming distance across all pairs of codewords.

So, if d is the minimal distance of code C , then C can detect up to $d - 1$ errors.

- *Let d be the minimal distance of code C . How many errors can C correct?*

This will again be a very similar line of thought as in the previous section. To be able to correct the erroneous message, it has to be closer to the original codeword than to any other codeword. The number of errors thus has to be strictly smaller than $\frac{d}{2}$. So, for d odd, C can correct up to $\frac{d-1}{2}$ errors. For d even, C can correct up to $\frac{d-2}{2}$ errors. To avoid distinguishing between odd and even cases, we can use a universal expression $\lfloor \frac{d-1}{2} \rfloor$, which makes use of the floor function.

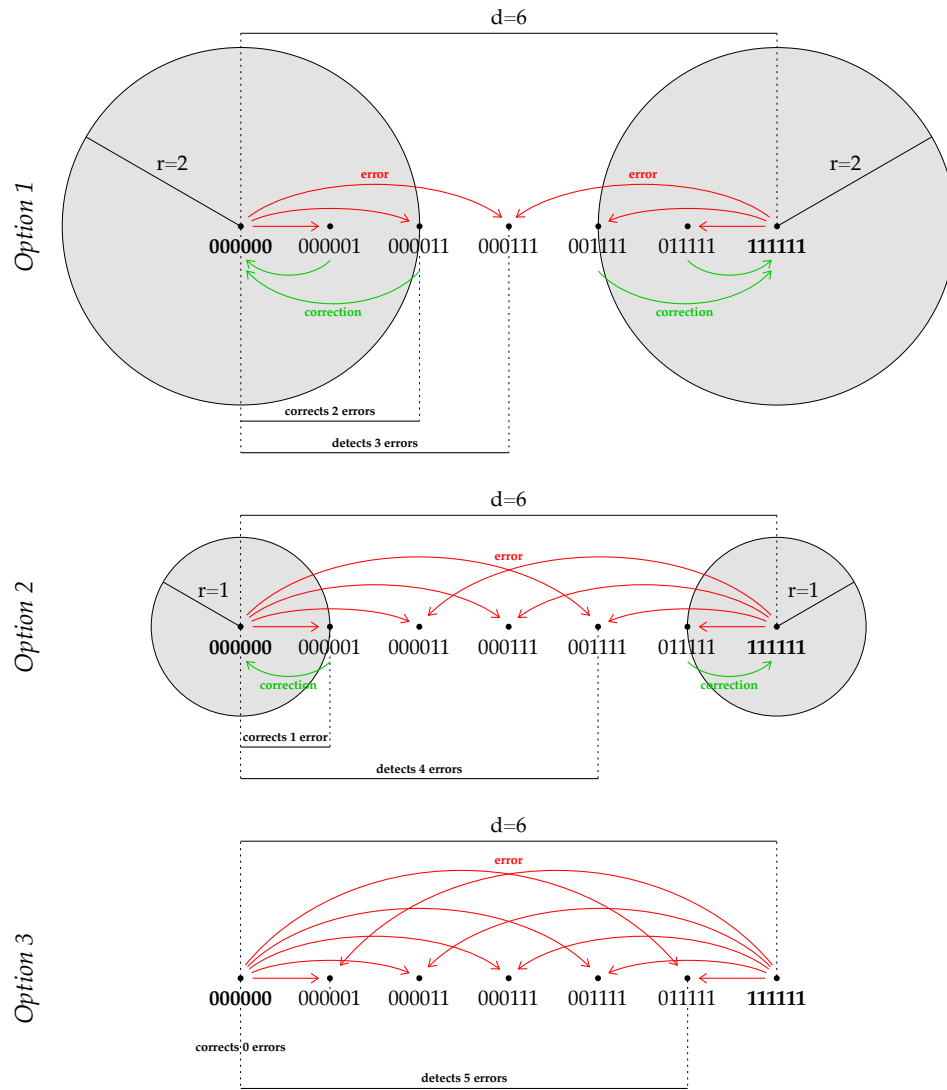
However, we should be careful in interpreting these properties. Take the example of a triple repetition code. It can detect up to 2 errors. It can correct up to 1 error. Can it do both at the same time? The problem is that a codeword received with 1 error is sometimes indistinguishable from a different codeword received with

Figure 1.6: Error correction for $d = 3$ Figure 1.7: Error correction and detection for $d = 4$

2 errors. So, we have to decide if we correct the word (assuming only 1 error occurred) or discard it (assuming 1 or 2 errors occurred), so we cannot use both single-error correction and double-error detection at the same time.

- What does that mean for codes with minimal distance $d > 3$?

How many errors the code detects and corrects is eventually the choice of the engineer—the more errors a code can correct, the less it can detect and vice versa. The possibilities always depend on the minimal distance of the code. In Figure 1.8, we can see that a code with minimal distance d can both detect t errors *and* correct r errors if $t + r < d$ and $0 \leq r \leq t$.

Figure 1.8: Error correction and detection options for $d = 6$

Interpreting the Hamming distance literally, by imagining words as points in space, will often prove useful. Building on the geometric analogy, we can imagine that for a given word w , all words that have a distance of r from w are inside the sphere with a center in w and radius r .

Visualising a code, you should see a space with scattered words, some of them being highlighted (for example being written in **bold**)—those are the codewords (see Figure 1.9). Each codeword has its 'bubble', the bubble has a center in the codeword and radius r . The bubble encapsulates all words created from the codeword by changing up to r positions.

The code can correct r errors if no word lies inside more than one bubble; in other words, no two bubbles overlap. Each word in a bubble is assigned to a unique codeword—the one from it 'originated'.

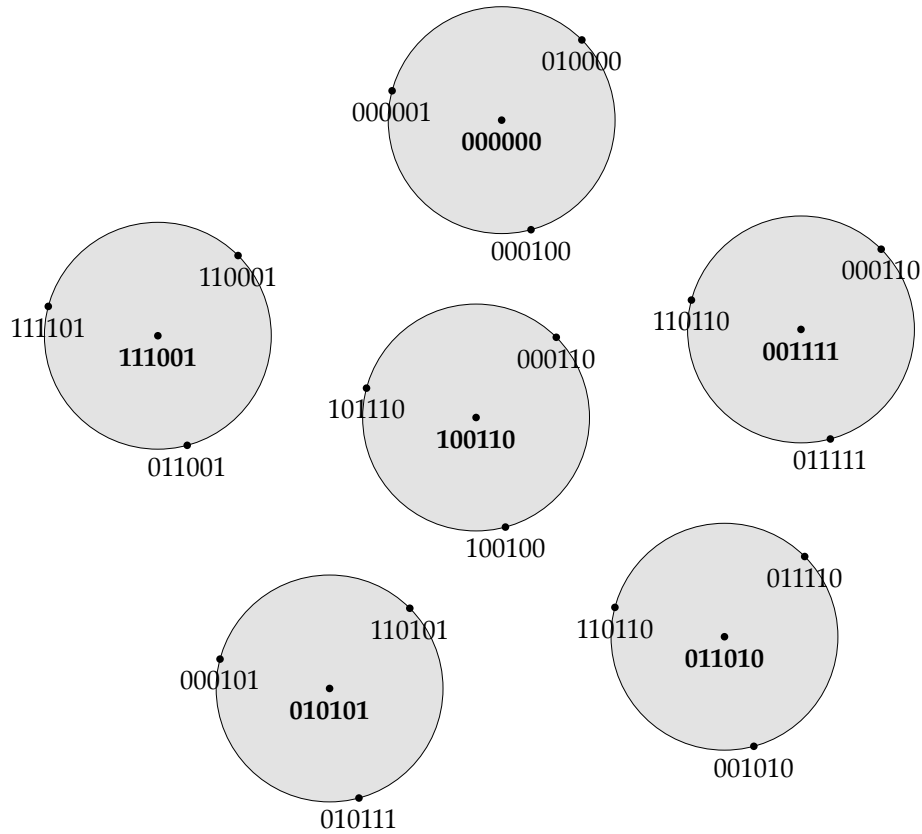


Figure 1.9: Bubbles, also known as Hamming spheres

This idea that each codeword carries some number of words in its bubble, constitutes an upper bound of how many codewords can be fit into a given space, given a minimal distance d (which translates to the bubble size). If this upper bound is attained (the space is completely covered by non-overlapping bubbles), we call such code a **perfect code**. We saw an example of a perfect code at the end of last section.

- *Formulate the upper bound for the number of codewords in a code with alphabet size q , block size n and minimal distance d .*

First, we have to count how many words are within distance $r = \lfloor \frac{d-1}{2} \rfloor$ from an arbitrarily chosen codeword $\mathbf{w}$. We arrive at the sum

$$\sum_{i=0}^r \binom{n}{i} \cdot (q-1)^i.$$

The number of all possible words q^n divided by this sum gives the maximum number of codewords. This is known as the **Hamming bound**.

Theorem. Let $C \subseteq A^n$ be a code with alphabet size q , block length n and minimal distance d . The number of elements in C satisfies

$$|C| \leq \frac{q^n}{\sum_{i=0}^r \binom{n}{i} \cdot (q-1)^i}$$

where $r = \lfloor \frac{d-1}{2} \rfloor$.

If a code attains the Hamming bound, we call it a **perfect code**.

Definition. Let $C \subseteq A^n$ be a code with alphabet size q , block length n and minimal distance d . If the number of elements in C satisfies

$$|C| = \frac{q^n}{\sum_{i=0}^r \binom{n}{i} \cdot (q-1)^i}$$

where $r = \lfloor \frac{d-1}{2} \rfloor$, we call C a **perfect code**.

This really just means that all possible words are within distance r of a unique codeword, so all words are correctable. That's what we managed to construct at the end of last section. Hamming codes are perfect codes!

1.4 Linearity

In the previous section, our concept of code has changed. Instead of viewing a code as a map from a message to an encoded message, we view it as a subset of possible words, called codewords. This view is simpler and more useful to build mathematical theory around it, but it is not useful in practice, yet. To make it usable, we must connect it back to the process of encoding, because ultimately, we want to design a program that will encode a message on one end and decode it on the other end.

One impractical aspect of the latter view is that a lot of memory would be needed to store all the codewords. It would be better to describe the set of codewords in some way that allows them to be generated by a computer.

► *Define*

- a) *triple-repetition code for message of length 4 over $\mathbb{Z}_{10}$,*
- b) *single-checksum code for message of length 4 over $\mathbb{Z}_{10}$,*
- c) *Hamming(7,4) code*

as a set of solutions to a system of equations.

a)

The set of codewords of the form $\mathbf{w} = (w_1, \dots, w_{12})$, $w_1, \dots, w_{12} \in \mathbb{Z}_{10}$ that satisfy

$$\begin{array}{llll} w_1 = w_2, & w_4 = w_5, & w_7 = w_8, & w_{10} = w_{11}, \\ w_1 = w_3, & w_4 = w_6, & w_7 = w_9, & w_{10} = w_{12}. \end{array}$$

b)

The set of codewords of the form $\mathbf{w} = (w_1, \dots, w_5)$, $w_1, \dots, w_5 \in \mathbb{Z}_{10}$ that satisfy

$$w_1 + w_2 + w_3 + w_4 = w_5.$$

c)

The set of codewords of the form $\mathbf{w} = (w_1, \dots, w_7)$, $w_1, \dots, w_7 \in \mathbb{Z}_2$ that satisfy

$$\begin{array}{l} w_1 + w_3 + w_5 + w_7 = 0, \\ w_2 + w_3 + w_6 + w_7 = 0, \\ w_4 + w_5 + w_6 + w_7 = 0. \end{array}$$

If you're not familiar with $\mathbb{Z}_m$ —*rings of integers modulo m*, they are just sets $\mathbb{Z}_m = \{0, \dots, m-1\}$ with addition and multiplication defined in such a way that you always reduce the result modulo m . It works like counting in a circle instead of on an infinite line of numbers.

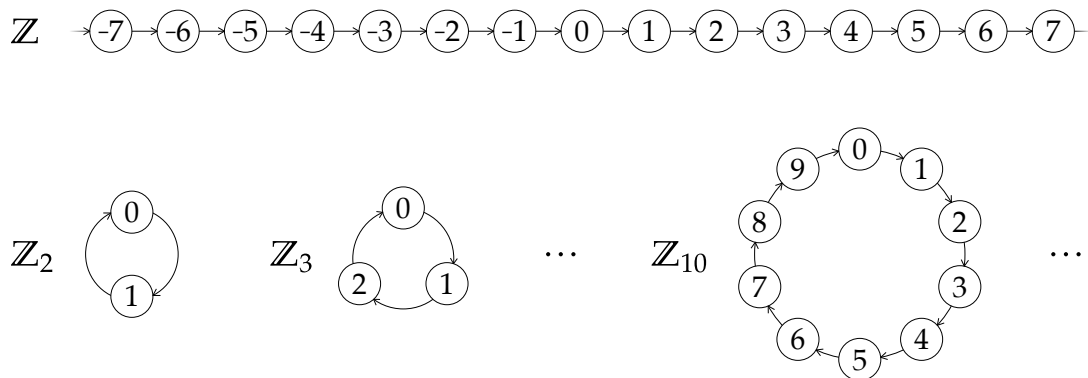


Figure 1.10: Rings of integers modulo m

Now imagine that you wrap the infinite line of integers around the circle like you'd wind a thread onto a spool. Some of the integers now occupy the same place on the spool. This is another view of rings of integers modulo m , where the symbols $0, \dots, m-1$ don't represent the integers themselves but also all the integers 'hiding' behind them. These sets have the property that they give the

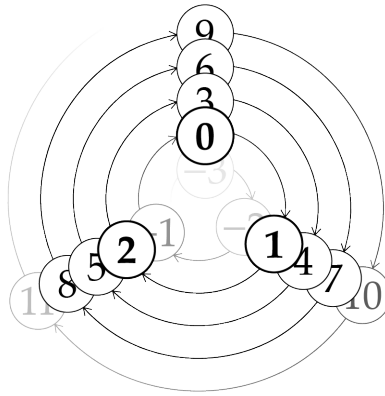


Figure 1.11: The ring of integers modulo 3

same residue after dividing by m . For example, the symbol 2 in $\mathbb{Z}_3$ represents the set $\{\dots, -4, -1, 2, 5, 8, 11, \dots\}$.

Back to the equations. Just like over $\mathbb{R}$, we can write the systems of equations over $\mathbb{Z}_m$ using matrices for compactness. So for example, the system c) can be written like this (note the similarity to the table of guards and prisoners):

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

The matrix on the left is called the **check matrix**. The check matrix is a much denser representation of the code than the collection of codewords (21 bits of the check matrix vs. $16 \cdot 7 = 112$ bits of all 16 codewords of the Hamming(7,4) code).

- *Solve the systems of equations from the previous page using parameters.*

Systems of linear equations over $\mathbb{R}$ are generally solved by eliminating variables using equivalent operations (those are swapping rows, multiplying a row by a constant, and adding one row to another). Can we apply the same methods to any ring of integers modulo m ? Short answer: no. A less short answer: with some we can, and we call those rings *fields*. We'll go into more depth in section 1.6, for now we'll be dealing with systems of equations where we can directly introduce the parameters and express the unknowns in terms of them.

A very helpful realization here is that the parameters are nothing but the message symbols—those are the only degrees of freedom in the codewords, while the check symbols don't contain any new information.

a)

$$\mathbf{w} = (p_1, p_1, p_1, p_2, p_2, p_2, p_3, p_3, p_3, p_4, p_4, p_4), p_1, p_2, p_3, p_4 \in \mathbb{Z}_{10}$$

b)

$$\mathbf{w} = (p_1, p_2, p_3, p_4, p_1 + p_2 + p_3 + p_4), p_1, p_2, p_3, p_4 \in \mathbb{Z}_{10}$$

c)

$$\mathbf{w} = (p_1 + p_2 + p_4, p_1 + p_3 + p_4, p_1, p_2 + p_3 + p_4, p_2, p_3, p_4), p_1, p_2, p_3, p_4 \in \mathbb{Z}_2$$

These solutions can also be written using matrices. All combinations of values in the parameters can be written as the ordered 4-tuple (p_1, p_2, p_3, p_4) , which is multiplied from the left by a matrix that tells us how many times is each parameter added in a given position in the codeword. The first row of the matrix describes the first position, and so on.

a)

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \\ w_8 \\ w_9 \\ w_{10} \\ w_{11} \\ w_{12} \end{pmatrix}$$

b)

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \end{pmatrix}$$

c)

$$\begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \end{pmatrix}$$

This matrix is called the **generator matrix**. Just like the check matrix, it completely characterizes the code.

► How does

- a) the number of equations in a system
 - b) the number of parameters needed to describe the solutions
- relate to the properties of the code?

The equations are basically the instructions to determine the check symbols, and each equation calculates one check symbol. Therefore, the number of equations is equal to the number of check symbols.

On the other hand, the number of parameters needed to describe the solutions (a.k.a. the dimension of the solution) is equal to the number of message symbols. There are as many solutions (codewords) as there are combinations of message symbols.

If you are familiar with linear algebra, you already know where this is going. If not, well, imagine for example a 3-dimensional space with coordinates x, y, z . One linear homogenous equation, say $2x - y + 3z = 0$, has infinitely many solutions

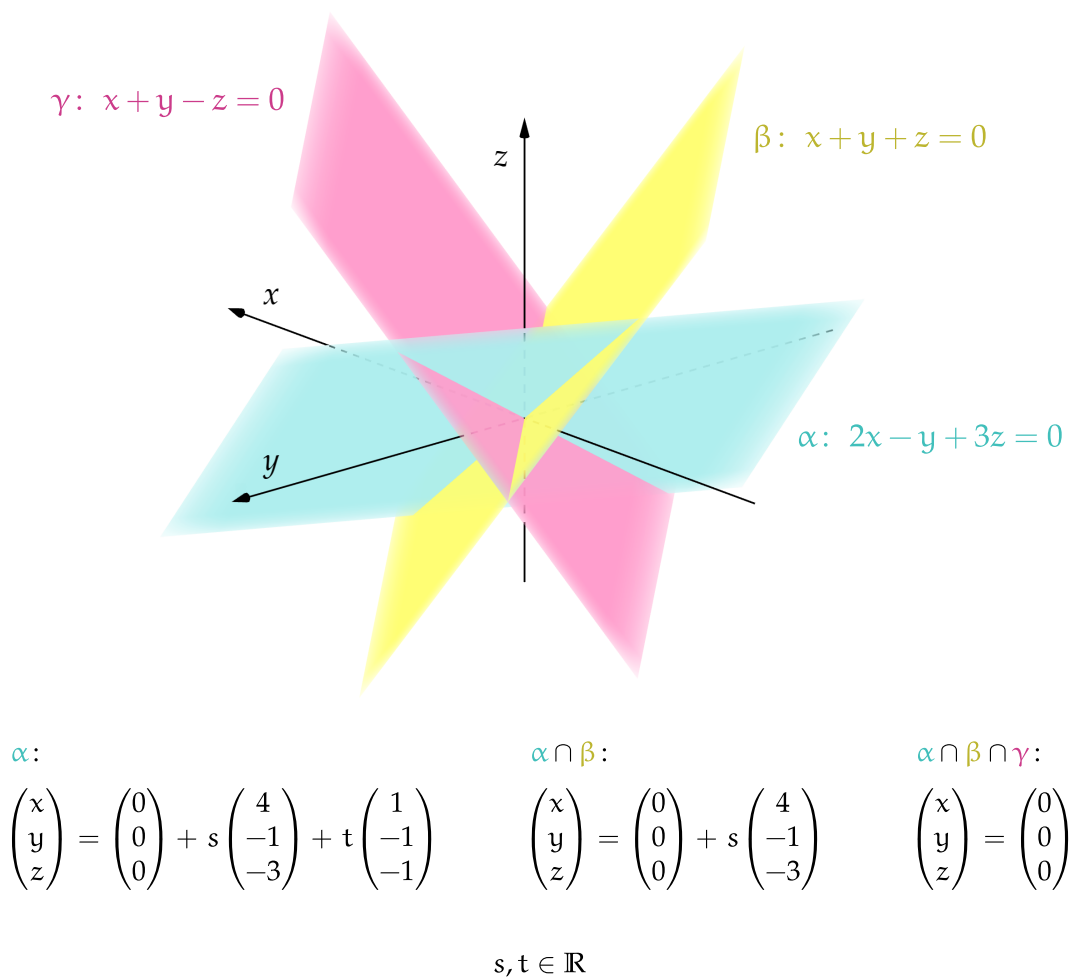


Figure 1.12: Linear equations as planes in 3D

with 2 parameters, which exist in a plane going through the point $(0,0,0)$. If we add another linear homogenous equation to the system (which is not a multiple of the previous one), the solutions of the system reduce to a line going through the point $(0,0,0)$, and can be described with 1 parameter. Each new linearly independent linear homogenous equation added to the system reduces the space of solutions by one dimension, and the number of parameters necessary to describe it by one. So, the number of equations in a system plus the solution dimension is equal to the number of unknowns in the system. Just like the number of check symbols plus the number of message symbols is the total number of symbols of the codeword!

We cannot say that the codewords of the codes from this section constitute a subspace of a vector space (because vector space is defined over a *field*, and we're still using the more general *rings*), but we can say that they constitute a submodule of a module, which is a generalization of a vector space.

Since the whole set of codewords can be generated by linear combinations of some well-picked codewords, we call this class of codes **linear codes**. The simple rule to find out if a code is linear is to check if adding two codewords always creates a codeword. The great thing about linear codes is that they can be densely represented.

1.5 The power of polynomials

In designing codes for various communication and storage media, there is no single best code because, as we saw, codes have different parameters (alphabet size, block size, number of errors they can correct or detect, etc.). What are the most important aspects of the use cases for which we are designing the codes?

One of the crucial aspects is the **error rate**—that is the percentage of symbols received with an error using the given technology. The error rate can greatly vary based on the technology in question, from *very close to zero* all the way to tens of percent.

Another aspect of the use cases is the tolerance to an incorrectly received message. This is also represented as a fraction (the maximum percentage of blocks which is tolerable to be incorrect), and there can be a big range, too, but the value will be smaller than the error rate (otherwise there would be no point in error correction).

- *Let's say that the error rate of our communication channel is 10 %, and we use the Hamming(7,4) code for single-error correction in the channel. What is the probability that the receiver receives an incorrect message?*

The correct message is received if and only if 0 or 1 errors occur in a block. Therefore, an incorrect message is received when 2 or more errors occur in one block—the decoding process produces an incorrect codeword that is read as a message different from the original one.

What is the probability that we get 2 to 7 errors in a block of 7 symbols? It's easier to calculate the inverse, and then subtract that from 100 %.

probability of 0 errors	0.9^7	$\approx 47.8 \%$
probability of 1 error	$7 \times 0.9^6 \times 0.1$	$\approx 37.2 \%$
probability of 0 or 1 errors	$0.9^7 + 7 \times 0.9^6 \times 0.1$	$\approx 85.0 \%$
probability of 2 to 7 errors	$1 - (0.9^7 + 7 \times 0.9^6 \times 0.1)$	$\approx 15.0 \%$

So, the probability that we receive a wrong message is 15 %.

- *Why is the probability of receiving a wrong message (with error correction being applied) higher than the probability of receiving a wrong symbol?*

At first glance, this is a counter-intuitive result. Didn't we say just a moment ago that error correction is used to lower the chances that we receive an incorrect message?

Before making a hasty conclusion that error-correcting codes don't work, we should try doing the calculation for different error rates. If you try that, you'll see that it's not always the case that the probability goes up. For example, with the same code and an error rate of 1 %, the probability of receiving a wrong message is 0.2 %.

So, this was a perfect example of a bad choice for an error-correcting code. The Hamming(7,4) code is too weak for an error rate of 10 %, and so it is even counter-productive. In fact, single-error correction is almost always too weak. The probability of getting n errors in one block shrinks exponentially with increasing n (since the probabilities multiply)¹, and we should take advantage of that.

We still haven't seen a good example of a multiple-error-correcting code. The next step is obvious: find an efficient code that can correct more than one error. Or even better—find a system to generate codes that have the right ratio of number of errors correction and information efficiency.

As spoiled by the name of this section, the magic word that opens a cave full of these invaluable codes is *polynomials*.

¹Supposed that the errors are not locally correlated. If they are (which often happens), there are some relatively simple tricks to get rid of the correlation.

This is how polynomials are defined over $\mathbb{R}$:

Definition. Let $n \in \mathbb{N}$, $a_0 \in \mathbb{R} \setminus \{0\}$ and $a_1, \dots, a_n \in \mathbb{R}$. The expression

$$a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n$$

is called a **polynomial of degree n** .

One of the possible interpretations of polynomials is that they are functions: you plug a value into x , and you get a function value $f(x)$ out.

Lower degree polynomial functions have their own names: a **constant function** ($n = 0$), a **linear function** ($n = 1$), a **quadratic function** ($n = 2$), and a **cubic function** ($n = 3$).

There is an important relationship between the degree of the polynomial and how many distinct points $(x, f(x))$ uniquely determine the polynomial. This is intuitive for the lower degree polynomial functions. It takes just one point to

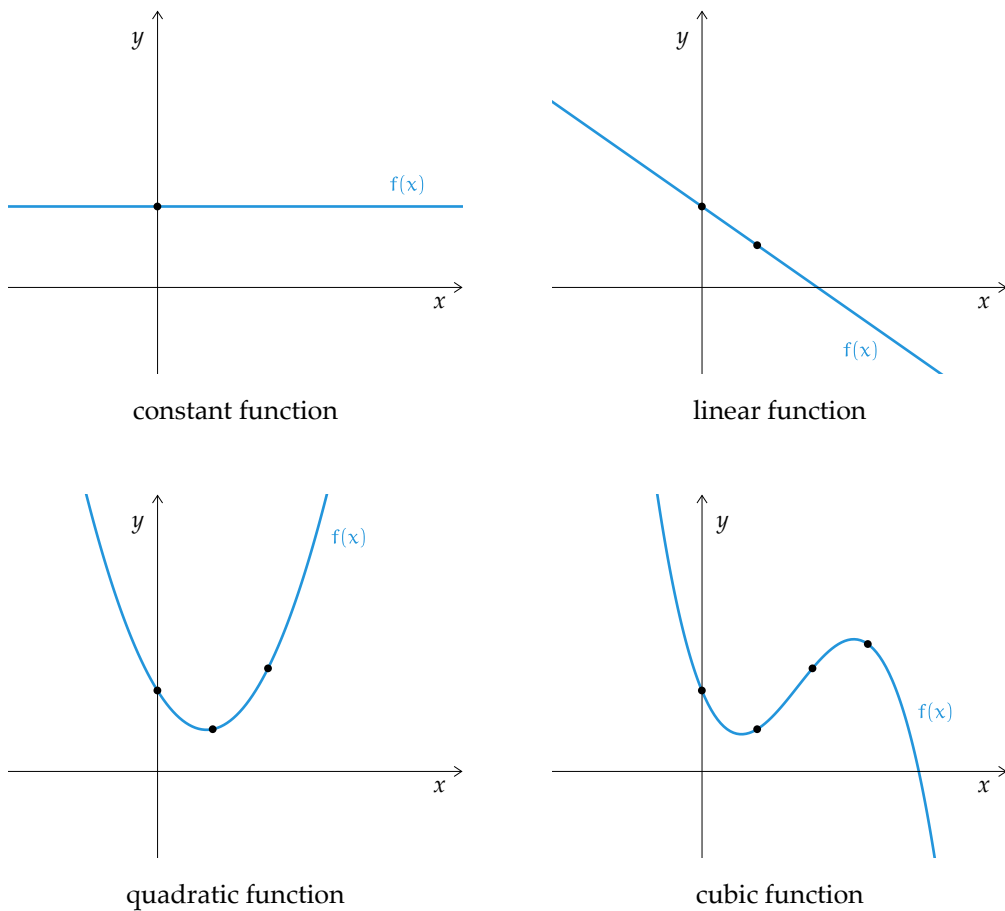


Figure 1.13: Polynomial functions

determine a constant function, because the graph of a constant function is a line parallel with the x -axis, and there is just one such line going through a given point. To determine a linear function, we need 2 points, because the graph of a linear function is a line. The pattern seems to be that a polynomial function of degree n is uniquely determined by $n + 1$ points.

Theorem. Let $n \in \mathbb{N}$.

For any $n + 1$ distinct points $(x_0, y_0), \dots, (x_n, y_n) \in \mathbb{R}^2$ where $x_i \neq x_j$ for all $i, j \in \{0, \dots, n\}, i \neq j$, there exists exactly one polynomial function $f(x)$ of degree n or lower, such that $f(x_i) = y_i$ for all $i \in \{0, \dots, n\}$.

Note that there is ‘of degree n or lower’. For example, if we choose the points $\{(0, 2), (1, 3), (2, 4)\}$, there is no parabola (quadratic function graph) going through those, only the linear function $f(x) = x + 2$.

Now, we’ll show how this theorem can be proven for $n = 3$.

Proof. We want to show that for 4 pairs of arguments and their function values $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$, there exists exactly one polynomial function of degree at most 3. In other words, the system of equations

$$\begin{aligned} a_0x_1^3 + a_1x_1^2 + a_2x_1 + a_3 &= y_1, \\ a_0x_2^3 + a_1x_2^2 + a_2x_2 + a_3 &= y_2, \\ a_0x_3^3 + a_1x_3^2 + a_2x_3 + a_3 &= y_3, \\ a_0x_4^3 + a_1x_4^2 + a_2x_4 + a_3 &= y_4 \end{aligned}$$

has exactly one solution $(\hat{a}_0, \hat{a}_1, \hat{a}_2, \hat{a}_3)$. If this were a concrete example, the x ’s and y ’s would be numbers and the a ’s would be the unknown coefficients of the polynomial that we are looking for.

The system of equations has exactly one solution if the left sides of the equations are linearly independent. In other words, the matrix

$$\begin{pmatrix} x_1^3 & x_1^2 & x_1^1 & 1 \\ x_2^3 & x_2^2 & x_2^1 & 1 \\ x_3^3 & x_3^2 & x_3^1 & 1 \\ x_4^3 & x_4^2 & x_4^1 & 1 \end{pmatrix}$$

must be invertible. How do we know if this matrix is invertible?

It suffices to show that the columns of this matrix are linearly independent. That is, there is no linear combination of the column vectors that makes a zero vector but the trivial one.

So, let's assume for contradiction that the system of equations

$$\begin{aligned}c_1x_1^3 + c_2x_1^2 + c_3x_1 + c_4 &= 0, \\c_1x_2^3 + c_2x_2^2 + c_3x_2 + c_4 &= 0, \\c_1x_3^3 + c_2x_3^2 + c_3x_3 + c_4 &= 0, \\c_1x_4^3 + c_2x_4^2 + c_3x_4 + c_4 &= 0\end{aligned}$$

has such a solution $(\hat{c}_1, \hat{c}_2, \hat{c}_3, \hat{c}_4) \neq (0, 0, 0, 0)$. That means that x_1, x_2, x_3, x_4 are all roots of the polynomial $\hat{c}_1x^3 + \hat{c}_2x^2 + \hat{c}_3x + \hat{c}_4$. We recall the fundamental theorem of algebra and realize that this is a contradiction, because a 3rd degree polynomial may have at most 3 roots. ■

We proved that a polynomial of degree 3 is uniquely determined by 4 points. It can be shown similarly that a polynomial of degree n is uniquely determined by $n + 1$ points.

- *How can we utilize the property that a polynomial of degree n is uniquely determined by $n + 1$ points to create an error-correcting code?*

If we connect a message with a concrete polynomial, then we can generate as many points as we need as the redundant data.

There are two ways to do that: our message will either be the coefficients of a polynomial, or the function values of some polynomial for some pre-selected arguments. In either scenario, we can generate more points that will serve as a redundancy that can be used to find and correct the erroneous symbols.

Let's try the first method. Let our message be 1997. The polynomial that it represents is $f(x) = 1x^3 + 9x^2 + 9x + 7$. Then, we calculate the function values for, let's say, 5 arguments $x \in \{0, 1, 2, 3, 4\}$, and the sequence $(f(0), f(1), f(2), f(3), f(4))$ will be our encoded message. In practice, we would have to solve the problem

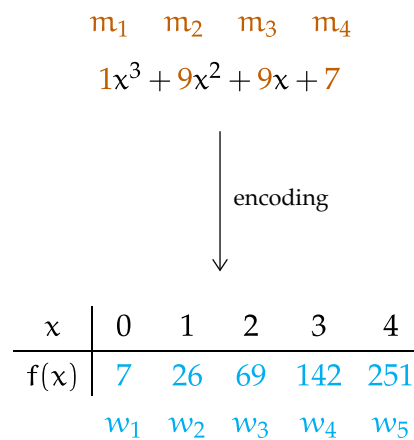


Figure 1.14: Encoding method no. 1

of how to represent these numbers with a finite alphabet in a fixed-size block of symbols, but we're going to ignore the problem and pretend that we can have an infinite alphabet where any number is a unique symbol.

In our example, we have sent the message $(7, 26, 69, 142, 251)$. How do we decode it? If no error occurred, the points $((0, 7), (1, 26), (2, 69), (3, 142), (4, 251))$ will all fit into a 3rd degree polynomial. We can figure out the polynomial with a system of linear equations and read the message from the coefficients of the polynomial.

Note that if one error occurred, we would definitely notice, because the other 4 points would already uniquely determine the polynomial, and the polynomial cannot produce different $f(x)$ for a single x . So, if there is one error, we detect it, but we cannot correct it—any 4-element subset of the 5 points determines a different polynomial, so we would have 5 candidates for the original message.

Therefore, we need more redundancy to be able to correct an error. As we know from the previous sections, the number of errors that a code can detect is a function of its minimal distance. What are the codewords now? They are the sequences $(f(0), \dots, f(n-1))$ generated by all the different 3rd degree polynomials. To figure out their minimal distance, let's write two different codewords under each other.

$$\begin{array}{cccccc} f(0) & f(1) & f(2) & f(3) & f(4) & f(5) \\ g(0) & g(1) & g(2) & g(3) & g(4) & g(5) \end{array}$$

Suppose that the codewords have corresponding numbers on 4 positions. These 4 points could be used to determine both polynomials $f(x)$ and $g(x)$, but since they are the same points, $f(x) = g(x)$. And since both sequences were produced by the same polynomial, they must be the same codewords.

$$\begin{array}{cccccc} f(0) & f(1) & f(2) & f(3) & & f(4) & f(5) \\ = & = & = & = & \implies & = & = \\ g(0) & g(1) & g(2) & g(3) & & g(4) & g(5) \end{array}$$

Therefore, with a 3rd degree polynomial, any two codewords may agree on at most 3 positions. Codewords of length 6 have a minimal distance of $6 - 3 = 3$, and $d = 3$ means the code can correct 1 error.

points	minimal distance	corrects	detects
5	2		1
6	3	1	
7	4	1	+1
8	5	2	
9	6	2	+1
10	7	3	
p	$p - 3$	$\lfloor \frac{p-4}{2} \rfloor$	$(p - 4) \bmod 2$

Table 1.1: 3rd degree polynomial

points	minimal distance	corrects	detects
p	p - n	$\lfloor \frac{p-n-1}{2} \rfloor$	$(p - n - 1) \bmod 2$

Table 1.2: n-th degree polynomial

One small disadvantage of this method is that the encoded message is completely different from the original message. That is fixed by the other method: we can first find a polynomial that produces the message as its function values, and we will calculate some extra function values as the redundancy.

Note that this method has the same properties regarding the minimal distance: for every two check symbols added, the code can correct one more error.

Both of these methods have the already mentioned downside: they work in the realm of integers, not in a finite alphabet. The integers would have to be represented in some base with multiple symbols, that would increase the length of the codewords and render the code less efficient.

Luckily, we can take this brilliant idea of using polynomials to the realm of finite fields.

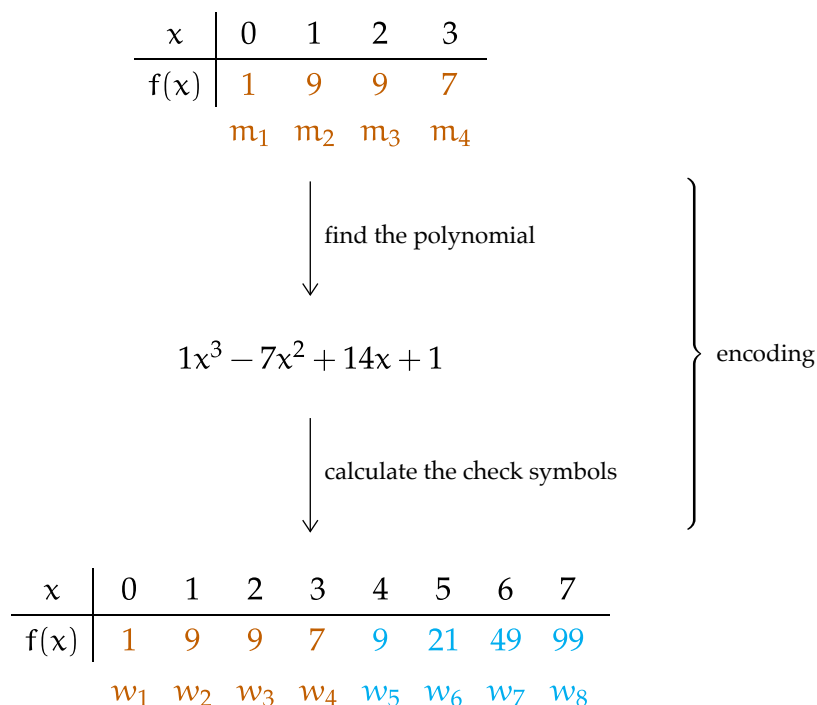


Figure 1.15: Encoding method no. 2

1.6 Counting in circles

Earlier we saw that counting in rings instead of on an infinite line is useful to create single-error-correcting codes over a finite alphabet. Maybe we could define polynomials over rings of integers modulo m to have both finite alphabets and multiple-error correction?

Defining polynomials over a ring of integers modulo m is intuitive—the coefficients are now the numbers 0 to $m - 1$, and if we want to evaluate the polynomial in $x \in \mathbb{Z}_m$, we plug it in, and perform the operations in $\mathbb{Z}_m$ in the canonical order (exponentiation, multiplication, addition).

Or we can do it another way—we can pretend that the polynomial is over $\mathbb{Z}$, perform the operations in $\mathbb{Z}$ and reduce it modulo m at the end. We'll arrive at the same value! This is thanks to a fact that hasn't been mentioned yet—the mapping

$$f: \mathbb{Z} \rightarrow \mathbb{Z}_m: \quad f(x) = x \bmod m$$

(the one we illustrated by winding a thread onto a spool) 'preserves' addition and multiplication. We call this property a **homomorphism**.

Definition. Let A, B be algebraic structures with one or more operations. The mapping $f: A \rightarrow B$, which has the property

$$\forall x, y \in A: f(x *_A y) = f(x) *_B f(y)$$

for all corresponding operations $*_A$ and $*_B$, is called a **homomorphism**.

The fact that granted us the ability to correct multiple errors was that a polynomial of degree n is uniquely determined by $n + 1$ points. Is this still true for polynomials over rings of integers modulo m ? Even in very simple cases, we can notice that there are counterexamples.

- Find an example of
 - a) two points with different x -coordinates that do not satisfy any polynomial over $\mathbb{Z}_{10}$ of degree at most 1,
 - b) two points with different x -coordinates that satisfy multiple polynomials over $\mathbb{Z}_{10}$ of degree at most 1.

We have to find two points $(x_1, y_1), (x_2, y_2), x_1 \neq x_2$ such that the system of equations

$$ax_1 + b = y_1$$

$$ax_2 + b = y_2$$

has a) no solution, and b) two or more solutions.

If we subtract the equations and factor out the linear coefficient a , we get

$$a(x_1 - x_2) = y_1 - y_2,$$

which is just another linear equation.

The second step is to realize that some linear equations over $\mathbb{Z}_{10}$ don't have a solution. For example, $2a = 3$ has no solution over $\mathbb{Z}_{10}$; multiplying a number by 2 and then reducing it by 10 always yields an even number.

Now we just have to pick x_1, x_2, y_1, y_2 such that $x_1 - x_2 = 2$ and $y_1 - y_2 = 3$. That can be for example $x_1 = 2, x_2 = 0, y_1 = 3, y_2 = 0$. So, the points $(0, 0)$ and $(2, 3)$ do not satisfy any polynomial over $\mathbb{Z}_{10}$ of degree at most 1.

Similarly, let's find a linear equation over $\mathbb{Z}_{10}$ that has multiple solutions. For example the equation $2a = 4$ has the two solutions $a \in \{2, 7\}$. The points $(0, 0)$ and $(2, 4)$ are points of both polynomials $2x + 0$ and $7x + 0$.

There are of course many more examples of both a) and b), but all lead to a linear equation $ax = b$ where a is a multiple of 2 or 5.

It is clear that we won't be successful with $\mathbb{Z}_{10}$, but what about different rings? If the problem in $\mathbb{Z}_{10}$ is somehow with the divisors of 10, would $\mathbb{Z}_p$ where p is a prime number (so has no non-trivial divisors) serve us better? Let's explore the differences.

- *Make addition and multiplication tables for the rings $\mathbb{Z}_2$ up to $\mathbb{Z}_{10}$. What are the similarities and differences between the different rings, especially between prime modules and composite modules?*

All the tables can be found the next page, but it is highly recommended to try to calculate and write down at least some of them by yourself.

First, we can notice that all tables are symmetric by the diagonal, because both addition and multiplication are commutative.

The addition tables are all very similar—there is something like a diagonal gradient of numbers. Each row and each column of these tables has each element present exactly once.

The multiplication tables look more messy at first glance. A closer look reveals that in $\mathbb{Z}_2, \mathbb{Z}_3, \mathbb{Z}_5$ and $\mathbb{Z}_7$, all rows and columns except for the first ones have each element present exactly once. In $\mathbb{Z}_4, \mathbb{Z}_6, \mathbb{Z}_8, \mathbb{Z}_9$ and $\mathbb{Z}_{10}$, this is true for the rows and columns of elements coprime with the module; the rows and columns that divide the module feature one or more zeroes.

It really looks like there are two types of rings of integers modulo m based on the multiplication tables. But as we know, this is no proof that these observations generalize for all m .

$\mathbb{Z}_2:$	<table><tr><td>+</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	+	0	1	0	0	1	1	1	0	<table><tr><td>·</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr></table>	·	0	1	0	0	0	1	0	1																																																																																																																																																																																																																																
+	0	1																																																																																																																																																																																																																																																		
0	0	1																																																																																																																																																																																																																																																		
1	1	0																																																																																																																																																																																																																																																		
·	0	1																																																																																																																																																																																																																																																		
0	0	0																																																																																																																																																																																																																																																		
1	0	1																																																																																																																																																																																																																																																		
$\mathbb{Z}_3:$	<table><tr><td>+</td><td>0</td><td>1</td><td>2</td></tr><tr><td>0</td><td>0</td><td>1</td><td>2</td></tr><tr><td>1</td><td>1</td><td>2</td><td>0</td></tr><tr><td>2</td><td>2</td><td>0</td><td>1</td></tr></table>	+	0	1	2	0	0	1	2	1	1	2	0	2	2	0	1	<table><tr><td>·</td><td>0</td><td>1</td><td>2</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>2</td></tr><tr><td>2</td><td>0</td><td>2</td><td>1</td></tr></table>	·	0	1	2	0	0	0	0	1	0	1	2	2	0	2	1																																																																																																																																																																																																																		
+	0	1	2																																																																																																																																																																																																																																																	
0	0	1	2																																																																																																																																																																																																																																																	
1	1	2	0																																																																																																																																																																																																																																																	
2	2	0	1																																																																																																																																																																																																																																																	
·	0	1	2																																																																																																																																																																																																																																																	
0	0	0	0																																																																																																																																																																																																																																																	
1	0	1	2																																																																																																																																																																																																																																																	
2	0	2	1																																																																																																																																																																																																																																																	
$\mathbb{Z}_4:$	<table><tr><td>+</td><td>0</td><td>1</td><td>2</td><td>3</td></tr><tr><td>0</td><td>0</td><td>1</td><td>2</td><td>3</td></tr><tr><td>1</td><td>1</td><td>2</td><td>3</td><td>0</td></tr><tr><td>2</td><td>2</td><td>3</td><td>0</td><td>1</td></tr><tr><td>3</td><td>3</td><td>0</td><td>1</td><td>2</td></tr></table>	+	0	1	2	3	0	0	1	2	3	1	1	2	3	0	2	2	3	0	1	3	3	0	1	2	<table><tr><td>·</td><td>0</td><td>1</td><td>2</td><td>3</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>2</td><td>3</td></tr><tr><td>2</td><td>0</td><td>2</td><td>0</td><td>2</td></tr><tr><td>3</td><td>0</td><td>3</td><td>2</td><td>1</td></tr></table>	·	0	1	2	3	0	0	0	0	0	1	0	1	2	3	2	0	2	0	2	3	0	3	2	1																																																																																																																																																																																																
+	0	1	2	3																																																																																																																																																																																																																																																
0	0	1	2	3																																																																																																																																																																																																																																																
1	1	2	3	0																																																																																																																																																																																																																																																
2	2	3	0	1																																																																																																																																																																																																																																																
3	3	0	1	2																																																																																																																																																																																																																																																
·	0	1	2	3																																																																																																																																																																																																																																																
0	0	0	0	0																																																																																																																																																																																																																																																
1	0	1	2	3																																																																																																																																																																																																																																																
2	0	2	0	2																																																																																																																																																																																																																																																
3	0	3	2	1																																																																																																																																																																																																																																																
$\mathbb{Z}_5:$	<table><tr><td>+</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>1</td><td>1</td><td>2</td><td>3</td><td>4</td><td>0</td></tr><tr><td>2</td><td>2</td><td>3</td><td>4</td><td>0</td><td>1</td></tr><tr><td>3</td><td>3</td><td>4</td><td>0</td><td>1</td><td>2</td></tr><tr><td>4</td><td>4</td><td>0</td><td>1</td><td>2</td><td>3</td></tr></table>	+	0	1	2	3	4	0	0	1	2	3	4	1	1	2	3	4	0	2	2	3	4	0	1	3	3	4	0	1	2	4	4	0	1	2	3	<table><tr><td>·</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>2</td><td>0</td><td>2</td><td>4</td><td>1</td><td>3</td></tr><tr><td>3</td><td>0</td><td>3</td><td>1</td><td>4</td><td>2</td></tr><tr><td>4</td><td>0</td><td>4</td><td>3</td><td>2</td><td>1</td></tr></table>	·	0	1	2	3	4	0	0	0	0	0	0	1	0	1	2	3	4	2	0	2	4	1	3	3	0	3	1	4	2	4	0	4	3	2	1																																																																																																																																																																										
+	0	1	2	3	4																																																																																																																																																																																																																																															
0	0	1	2	3	4																																																																																																																																																																																																																																															
1	1	2	3	4	0																																																																																																																																																																																																																																															
2	2	3	4	0	1																																																																																																																																																																																																																																															
3	3	4	0	1	2																																																																																																																																																																																																																																															
4	4	0	1	2	3																																																																																																																																																																																																																																															
·	0	1	2	3	4																																																																																																																																																																																																																																															
0	0	0	0	0	0																																																																																																																																																																																																																																															
1	0	1	2	3	4																																																																																																																																																																																																																																															
2	0	2	4	1	3																																																																																																																																																																																																																																															
3	0	3	1	4	2																																																																																																																																																																																																																																															
4	0	4	3	2	1																																																																																																																																																																																																																																															
$\mathbb{Z}_6:$	<table><tr><td>+</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr><tr><td>0</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr><tr><td>1</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>0</td></tr><tr><td>2</td><td>2</td><td>3</td><td>4</td><td>5</td><td>0</td><td>1</td></tr><tr><td>3</td><td>3</td><td>4</td><td>5</td><td>0</td><td>1</td><td>2</td></tr><tr><td>4</td><td>4</td><td>5</td><td>0</td><td>1</td><td>2</td><td>3</td></tr><tr><td>5</td><td>5</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr></table>	+	0	1	2	3	4	5	0	0	1	2	3	4	5	1	1	2	3	4	5	0	2	2	3	4	5	0	1	3	3	4	5	0	1	2	4	4	5	0	1	2	3	5	5	0	1	2	3	4	<table><tr><td>·</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr><tr><td>2</td><td>0</td><td>2</td><td>4</td><td>0</td><td>2</td><td>4</td></tr><tr><td>3</td><td>0</td><td>3</td><td>0</td><td>3</td><td>0</td><td>3</td></tr><tr><td>4</td><td>0</td><td>4</td><td>2</td><td>0</td><td>4</td><td>2</td></tr><tr><td>5</td><td>0</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr></table>	·	0	1	2	3	4	5	0	0	0	0	0	0	0	1	0	1	2	3	4	5	2	0	2	4	0	2	4	3	0	3	0	3	0	3	4	0	4	2	0	4	2	5	0	5	4	3	2	1																																																																																																																																																
+	0	1	2	3	4	5																																																																																																																																																																																																																																														
0	0	1	2	3	4	5																																																																																																																																																																																																																																														
1	1	2	3	4	5	0																																																																																																																																																																																																																																														
2	2	3	4	5	0	1																																																																																																																																																																																																																																														
3	3	4	5	0	1	2																																																																																																																																																																																																																																														
4	4	5	0	1	2	3																																																																																																																																																																																																																																														
5	5	0	1	2	3	4																																																																																																																																																																																																																																														
·	0	1	2	3	4	5																																																																																																																																																																																																																																														
0	0	0	0	0	0	0																																																																																																																																																																																																																																														
1	0	1	2	3	4	5																																																																																																																																																																																																																																														
2	0	2	4	0	2	4																																																																																																																																																																																																																																														
3	0	3	0	3	0	3																																																																																																																																																																																																																																														
4	0	4	2	0	4	2																																																																																																																																																																																																																																														
5	0	5	4	3	2	1																																																																																																																																																																																																																																														
$\mathbb{Z}_7:$	<table><tr><td>+</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr><tr><td>0</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr><tr><td>1</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>0</td></tr><tr><td>2</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>0</td><td>1</td></tr><tr><td>3</td><td>3</td><td>4</td><td>5</td><td>6</td><td>0</td><td>1</td><td>2</td></tr><tr><td>4</td><td>4</td><td>5</td><td>6</td><td>0</td><td>1</td><td>2</td><td>3</td></tr><tr><td>5</td><td>5</td><td>6</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>6</td><td>6</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>	+	0	1	2	3	4	5	6	0	0	1	2	3	4	5	6	1	1	2	3	4	5	6	0	2	2	3	4	5	6	0	1	3	3	4	5	6	0	1	2	4	4	5	6	0	1	2	3	5	5	6	0	1	2	3	4	6	6	0	1	2	3	4	5	<table><tr><td>·</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr><tr><td>2</td><td>0</td><td>2</td><td>4</td><td>6</td><td>1</td><td>3</td><td>5</td></tr><tr><td>3</td><td>0</td><td>3</td><td>6</td><td>2</td><td>5</td><td>1</td><td>4</td></tr><tr><td>4</td><td>0</td><td>4</td><td>1</td><td>5</td><td>2</td><td>6</td><td>3</td></tr><tr><td>5</td><td>0</td><td>5</td><td>3</td><td>1</td><td>6</td><td>4</td><td>2</td></tr><tr><td>6</td><td>0</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr></table>	·	0	1	2	3	4	5	6	0	0	0	0	0	0	0	0	1	0	1	2	3	4	5	6	2	0	2	4	6	1	3	5	3	0	3	6	2	5	1	4	4	0	4	1	5	2	6	3	5	0	5	3	1	6	4	2	6	0	6	5	4	3	2	1																																																																																																																		
+	0	1	2	3	4	5	6																																																																																																																																																																																																																																													
0	0	1	2	3	4	5	6																																																																																																																																																																																																																																													
1	1	2	3	4	5	6	0																																																																																																																																																																																																																																													
2	2	3	4	5	6	0	1																																																																																																																																																																																																																																													
3	3	4	5	6	0	1	2																																																																																																																																																																																																																																													
4	4	5	6	0	1	2	3																																																																																																																																																																																																																																													
5	5	6	0	1	2	3	4																																																																																																																																																																																																																																													
6	6	0	1	2	3	4	5																																																																																																																																																																																																																																													
·	0	1	2	3	4	5	6																																																																																																																																																																																																																																													
0	0	0	0	0	0	0	0																																																																																																																																																																																																																																													
1	0	1	2	3	4	5	6																																																																																																																																																																																																																																													
2	0	2	4	6	1	3	5																																																																																																																																																																																																																																													
3	0	3	6	2	5	1	4																																																																																																																																																																																																																																													
4	0	4	1	5	2	6	3																																																																																																																																																																																																																																													
5	0	5	3	1	6	4	2																																																																																																																																																																																																																																													
6	0	6	5	4	3	2	1																																																																																																																																																																																																																																													
$\mathbb{Z}_8:$	<table><tr><td>+</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr><tr><td>0</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr><tr><td>1</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td></tr><tr><td>2</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td></tr><tr><td>3</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td></tr><tr><td>4</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td></tr><tr><td>5</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>6</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr><tr><td>7</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr></table>	+	0	1	2	3	4	5	6	7	0	0	1	2	3	4	5	6	7	1	1	2	3	4	5	6	7	0	2	2	3	4	5	6	7	0	1	3	3	4	5	6	7	0	1	2	4	4	5	6	7	0	1	2	3	5	5	6	7	0	1	2	3	4	6	6	7	0	1	2	3	4	5	7	7	0	1	2	3	4	5	6	<table><tr><td>·</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr><tr><td>2</td><td>0</td><td>2</td><td>4</td><td>6</td><td>0</td><td>2</td><td>4</td><td>6</td></tr><tr><td>3</td><td>0</td><td>3</td><td>6</td><td>1</td><td>4</td><td>7</td><td>2</td><td>5</td></tr><tr><td>4</td><td>0</td><td>4</td><td>0</td><td>4</td><td>0</td><td>4</td><td>0</td><td>4</td></tr><tr><td>5</td><td>0</td><td>5</td><td>2</td><td>7</td><td>4</td><td>1</td><td>6</td><td>3</td></tr><tr><td>6</td><td>0</td><td>6</td><td>4</td><td>2</td><td>0</td><td>6</td><td>4</td><td>2</td></tr><tr><td>7</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr></table>	·	0	1	2	3	4	5	6	7	0	0	0	0	0	0	0	0	0	1	0	1	2	3	4	5	6	7	2	0	2	4	6	0	2	4	6	3	0	3	6	1	4	7	2	5	4	0	4	0	4	0	4	0	4	5	0	5	2	7	4	1	6	3	6	0	6	4	2	0	6	4	2	7	0	7	6	5	4	3	2	1																																																																																
+	0	1	2	3	4	5	6	7																																																																																																																																																																																																																																												
0	0	1	2	3	4	5	6	7																																																																																																																																																																																																																																												
1	1	2	3	4	5	6	7	0																																																																																																																																																																																																																																												
2	2	3	4	5	6	7	0	1																																																																																																																																																																																																																																												
3	3	4	5	6	7	0	1	2																																																																																																																																																																																																																																												
4	4	5	6	7	0	1	2	3																																																																																																																																																																																																																																												
5	5	6	7	0	1	2	3	4																																																																																																																																																																																																																																												
6	6	7	0	1	2	3	4	5																																																																																																																																																																																																																																												
7	7	0	1	2	3	4	5	6																																																																																																																																																																																																																																												
·	0	1	2	3	4	5	6	7																																																																																																																																																																																																																																												
0	0	0	0	0	0	0	0	0																																																																																																																																																																																																																																												
1	0	1	2	3	4	5	6	7																																																																																																																																																																																																																																												
2	0	2	4	6	0	2	4	6																																																																																																																																																																																																																																												
3	0	3	6	1	4	7	2	5																																																																																																																																																																																																																																												
4	0	4	0	4	0	4	0	4																																																																																																																																																																																																																																												
5	0	5	2	7	4	1	6	3																																																																																																																																																																																																																																												
6	0	6	4	2	0	6	4	2																																																																																																																																																																																																																																												
7	0	7	6	5	4	3	2	1																																																																																																																																																																																																																																												
$\mathbb{Z}_9:$	<table><tr><td>+</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr><tr><td>0</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr><tr><td>1</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>0</td></tr><tr><td>2</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>0</td><td>1</td></tr><tr><td>3</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>0</td><td>1</td><td>2</td></tr><tr><td>4</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>0</td><td>1</td><td>2</td><td>3</td></tr><tr><td>5</td><td>5</td><td>6</td><td>7</td><td>8</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>6</td><td>6</td><td>7</td><td>8</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr><tr><td>7</td><td>7</td><td>8</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr><tr><td>8</td><td>8</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr></table>	+	0	1	2	3	4	5	6	7	8	0	0	1	2	3	4	5	6	7	8	1	1	2	3	4	5	6	7	8	0	2	2	3	4	5	6	7	8	0	1	3	3	4	5	6	7	8	0	1	2	4	4	5	6	7	8	0	1	2	3	5	5	6	7	8	0	1	2	3	4	6	6	7	8	0	1	2	3	4	5	7	7	8	0	1	2	3	4	5	6	8	8	0	1	2	3	4	5	6	7	<table><tr><td>·</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr><tr><td>2</td><td>0</td><td>2</td><td>4</td><td>6</td><td>8</td><td>1</td><td>3</td><td>5</td><td>7</td></tr><tr><td>3</td><td>0</td><td>3</td><td>6</td><td>0</td><td>3</td><td>6</td><td>0</td><td>3</td><td>6</td></tr><tr><td>4</td><td>0</td><td>4</td><td>8</td><td>3</td><td>7</td><td>2</td><td>6</td><td>1</td><td>5</td></tr><tr><td>5</td><td>0</td><td>5</td><td>1</td><td>6</td><td>2</td><td>7</td><td>3</td><td>8</td><td>4</td></tr><tr><td>6</td><td>0</td><td>6</td><td>3</td><td>0</td><td>6</td><td>3</td><td>0</td><td>6</td><td>3</td></tr><tr><td>7</td><td>0</td><td>7</td><td>5</td><td>3</td><td>1</td><td>8</td><td>6</td><td>4</td><td>2</td></tr><tr><td>8</td><td>0</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr></table>	·	0	1	2	3	4	5	6	7	8	0	0	0	0	0	0	0	0	0	0	1	0	1	2	3	4	5	6	7	8	2	0	2	4	6	8	1	3	5	7	3	0	3	6	0	3	6	0	3	6	4	0	4	8	3	7	2	6	1	5	5	0	5	1	6	2	7	3	8	4	6	0	6	3	0	6	3	0	6	3	7	0	7	5	3	1	8	6	4	2	8	0	8	7	6	5	4	3	2	1																																										
+	0	1	2	3	4	5	6	7	8																																																																																																																																																																																																																																											
0	0	1	2	3	4	5	6	7	8																																																																																																																																																																																																																																											
1	1	2	3	4	5	6	7	8	0																																																																																																																																																																																																																																											
2	2	3	4	5	6	7	8	0	1																																																																																																																																																																																																																																											
3	3	4	5	6	7	8	0	1	2																																																																																																																																																																																																																																											
4	4	5	6	7	8	0	1	2	3																																																																																																																																																																																																																																											
5	5	6	7	8	0	1	2	3	4																																																																																																																																																																																																																																											
6	6	7	8	0	1	2	3	4	5																																																																																																																																																																																																																																											
7	7	8	0	1	2	3	4	5	6																																																																																																																																																																																																																																											
8	8	0	1	2	3	4	5	6	7																																																																																																																																																																																																																																											
·	0	1	2	3	4	5	6	7	8																																																																																																																																																																																																																																											
0	0	0	0	0	0	0	0	0	0																																																																																																																																																																																																																																											
1	0	1	2	3	4	5	6	7	8																																																																																																																																																																																																																																											
2	0	2	4	6	8	1	3	5	7																																																																																																																																																																																																																																											
3	0	3	6	0	3	6	0	3	6																																																																																																																																																																																																																																											
4	0	4	8	3	7	2	6	1	5																																																																																																																																																																																																																																											
5	0	5	1	6	2	7	3	8	4																																																																																																																																																																																																																																											
6	0	6	3	0	6	3	0	6	3																																																																																																																																																																																																																																											
7	0	7	5	3	1	8	6	4	2																																																																																																																																																																																																																																											
8	0	8	7	6	5	4	3	2	1																																																																																																																																																																																																																																											
$\mathbb{Z}_{10}:$	<table><tr><td>+</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr><tr><td>0</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr><tr><td>1</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td></tr><tr><td>2</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr><tr><td>3</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td></tr><tr><td>4</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td></tr><tr><td>5</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>6</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr><tr><td>7</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr><tr><td>8</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr><tr><td>9</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr></table>	+	0	1	2	3	4	5	6	7	8	9	0	0	1	2	3	4	5	6	7	8	9	1	1	2	3	4	5	6	7	8	9	0	2	2	3	4	5	6	7	8	9	0	1	3	3	4	5	6	7	8	9	0	1	2	4	4	5	6	7	8	9	0	1	2	3	5	5	6	7	8	9	0	1	2	3	4	6	6	7	8	9	0	1	2	3	4	5	7	7	8	9	0	1	2	3	4	5	6	8	8	9	0	1	2	3	4	5	6	7	9	9	0	1	2	3	4	5	6	7	8	<table><tr><td>·</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr><tr><td>2</td><td>0</td><td>2</td><td>4</td><td>6</td><td>8</td><td>0</td><td>2</td><td>4</td><td>6</td><td>8</td></tr><tr><td>3</td><td>0</td><td>3</td><td>6</td><td>9</td><td>2</td><td>5</td><td>8</td><td>1</td><td>4</td><td>7</td></tr><tr><td>4</td><td>0</td><td>4</td><td>8</td><td>2</td><td>6</td><td>0</td><td>4</td><td>8</td><td>2</td><td>6</td></tr><tr><td>5</td><td>0</td><td>5</td><td>0</td><td>5</td><td>0</td><td>5</td><td>0</td><td>5</td><td>0</td><td>5</td></tr><tr><td>6</td><td>0</td><td>6</td><td>2</td><td>8</td><td>4</td><td>0</td><td>6</td><td>2</td><td>8</td><td>4</td></tr><tr><td>7</td><td>0</td><td>7</td><td>4</td><td>1</td><td>8</td><td>5</td><td>2</td><td>9</td><td>6</td><td>3</td></tr><tr><td>8</td><td>0</td><td>8</td><td>6</td><td>4</td><td>2</td><td>0</td><td>8</td><td>6</td><td>4</td><td>2</td></tr><tr><td>9</td><td>0</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr></table>	·	0	1	2	3	4	5	6	7	8	9	0	0	0	0	0	0	0	0	0	0	0	1	0	1	2	3	4	5	6	7	8	9	2	0	2	4	6	8	0	2	4	6	8	3	0	3	6	9	2	5	8	1	4	7	4	0	4	8	2	6	0	4	8	2	6	5	0	5	0	5	0	5	0	5	0	5	6	0	6	2	8	4	0	6	2	8	4	7	0	7	4	1	8	5	2	9	6	3	8	0	8	6	4	2	0	8	6	4	2	9	0	9	8	7	6	5	4	3	2	1
+	0	1	2	3	4	5	6	7	8	9																																																																																																																																																																																																																																										
0	0	1	2	3	4	5	6	7	8	9																																																																																																																																																																																																																																										
1	1	2	3	4	5	6	7	8	9	0																																																																																																																																																																																																																																										
2	2	3	4	5	6	7	8	9	0	1																																																																																																																																																																																																																																										
3	3	4	5	6	7	8	9	0	1	2																																																																																																																																																																																																																																										
4	4	5	6	7	8	9	0	1	2	3																																																																																																																																																																																																																																										
5	5	6	7	8	9	0	1	2	3	4																																																																																																																																																																																																																																										
6	6	7	8	9	0	1	2	3	4	5																																																																																																																																																																																																																																										
7	7	8	9	0	1	2	3	4	5	6																																																																																																																																																																																																																																										
8	8	9	0	1	2	3	4	5	6	7																																																																																																																																																																																																																																										
9	9	0	1	2	3	4	5	6	7	8																																																																																																																																																																																																																																										
·	0	1	2	3	4	5	6	7	8	9																																																																																																																																																																																																																																										
0	0	0	0	0	0	0	0	0	0	0																																																																																																																																																																																																																																										
1	0	1	2	3	4	5	6	7	8	9																																																																																																																																																																																																																																										
2	0	2	4	6	8	0	2	4	6	8																																																																																																																																																																																																																																										
3	0	3	6	9	2	5	8	1	4	7																																																																																																																																																																																																																																										
4	0	4	8	2	6	0	4	8	2	6																																																																																																																																																																																																																																										
5	0	5	0	5	0	5	0	5	0	5																																																																																																																																																																																																																																										
6	0	6	2	8	4	0	6	2	8	4																																																																																																																																																																																																																																										
7	0	7	4	1	8	5	2	9	6	3																																																																																																																																																																																																																																										
8	0	8	6	4	2	0	8	6	4	2																																																																																																																																																																																																																																										
9	0	9	8	7	6	5	4	3	2	1																																																																																																																																																																																																																																										

Figure 1.16: Addition and multiplication tables for rings of integers modulo m

For the sake of simpler formulation of the next question, let's coin a new term: *rolumn*. An n -th **rolumn** is the union of n -th row and n -th column, less all rows and columns to the left and above.

·	0	1	2	3	
0	0	0	0	0	1st rolumn
1	0	1	2	3	2nd rolumn
2	0	2	0	2	3rd rolumn
3	0	3	2	1	4th rolumn

Figure 1.17: n -th rolumn

- Can you prove that in the multiplication table of $\mathbb{Z}_p$ where p is prime, there are no 0's except for the first rolumn?

Proof. First, let's rule out the possibility that a 0 could be in the second rolumn. If $a, b \in \{1, \dots, p-1\} \wedge (a = 1 \vee b = 1)$, then $0 < ab < p$.

Now we can just consider the rest of the table. We have to disprove the following statement: for given prime number p and $a, b \in \{2, \dots, p-1\}$, there exists $k \in \mathbb{N} \setminus \{0\}$ such that $ab - kp = 0$.

That is $ab = kp$. The left side is divisible by a and greater than 0, therefore the right side must also be divisible by a and greater than 0. If kp is divisible by a , since p is prime, k must be divisible by a . If k is divisible by a , there exists $l \in \mathbb{N} \setminus \{0\}$ such that $k = al$. Therefore $ab = alp \implies b = lp$ but that cannot happen because $b \in \{2, \dots, p-1\}$. ■

Now, we can also prove that in the multiplication table of $\mathbb{Z}_p$, there are no two same elements in a row other than the first row. If there are not, then there must be all elements 0 to $p-1$ (because there are exactly p columns).

- Can you prove that in the multiplication table of $\mathbb{Z}_p$ where p is prime, there are no two same elements in a row other than the first row?

Proof. Again, we can dismiss the second row, since it is trivial to show that there are no two same elements.

Suppose for contradiction that there are two same elements in a row other than the first two rows:

·	0	1	...	x_1	x_2
0	0	0	...		
1	0	1			
$\vdots$	$\vdots$				
a				b	b

That is, there exist $k, l \in \mathbb{Z}$ such that $ax_1 - kp = b$ and $ax_2 - lp = b$. Subtracting the equations, we get

$$(ax_1 - kp) - (ax_2 - lp) = 0,$$

and after reordering and factoring out a and p , we have

$$a(x_1 - x_2) = (k - l)p.$$

The argument follows similarly to the previous proof: the left side is divisible by a and non-zero, so the right side must be divisible by a and non-zero, too. Since p isn't divisible by a , $(k - l)$ must be, therefore there exists $m \in \mathbb{Z}, m \neq 0$ such that $(k - l) = am$. Then

$$a(x_1 - x_2) = amp$$

is simplified to

$$x_1 - x_2 = mp.$$

From the initial assumption that $x_1, x_2 \in \mathbb{Z}_p, x_1 \neq x_2$ follows $-p < x_1 - x_2 < p$, which leads to a contradiction. ■

We have proven that in the multiplication tables of $\mathbb{Z}_p$ where p is prime, each but the first row contains all the elements of $\mathbb{Z}_p$ exactly once. Since multiplication is commutative, the same is true for columns. That means that all equations $ax = b$ where $a, b \in \mathbb{Z}_p, a \neq 0$ have a unique solution. This means that in $\mathbb{Z}_p$, we can divide by non-zero elements.

In fact, we've just discovered **finite fields**. Finite fields are algebraic structures that mimic the behavior of $\mathbb{R}$, except they are finite. This gives us hope for the polynomials to work the same, too!

Definition. A set F with two binary operations $+, \cdot$ which satisfies the following properties:

1. $\forall a, b \in F: \quad a + b = b + a$
2. $\forall a, b \in F: \quad a \cdot b = b \cdot a$
3. $\forall a, b, c \in F: \quad (a + b) + c = a + (b + c)$
4. $\forall a, b, c \in F: \quad (a \cdot b) \cdot c = a \cdot (b \cdot c)$
5. $\forall a, b, c \in F: \quad a \cdot (b + c) = a \cdot b + a \cdot c$
6. $\exists 0 \in F: \quad \forall a \in F: \quad a + 0 = a$
7. $\exists 1 \in F: \quad \forall a \in F: \quad a \cdot 1 = a$
8. $\forall a \in F: \quad \exists -a \in F: \quad a + (-a) = 0$
9. $\forall a \neq 0 \in F: \quad \exists a^{-1} \in F: \quad a \cdot a^{-1} = 1$

is called a **field**. If F is finite, then it is a **finite field**.

The properties are also called axioms, and they can be colloquially summarized as

- both operations are commutative,
- both operations are associative,
- they are connected by the distributive law,
- both operations have neutral elements,
- there are additive inverses for all elements,
- there are multiplicative inverses for all non-zero elements.

While the first 8 axioms are true of all $\mathbb{Z}_m$, the ninth axiom is what makes the difference between $\mathbb{Z}_m$ and $\mathbb{Z}_p$. With that, we can apply the results of linear algebra, because vector spaces are 'built' over fields.

Note that so far we only showed that $\mathbb{Z}_p$ where p is prime are finite fields, but there may be (and there *are*) other finite fields.

In fields, addition and multiplication is defined, and subtraction and division is then defined through addition and multiplication: $a - b$ is defined as $a + (-b)$ and $\frac{a}{b}$ as $a \cdot b^{-1}$.

What about exponentiation? We define a^n as the number a multiplied by itself n times. This can be done only with $n \in \mathbb{N}$ (n is not an element of the field). Acknowledging that $a^n \cdot a^m$ can be simplified to a^{n+m} , it makes sense to extend the definition to non-positive integers, too, with $a^0 = 1$ and $a^{-n} = (a^{-1})^n$.

- Create exponentiation tables for $\mathbb{Z}_2, \mathbb{Z}_3, \mathbb{Z}_5$ and $\mathbb{Z}_7$ with rows corresponding to the base (elements of $\mathbb{Z}_p$) and columns corresponding to the exponents (elements of $\mathbb{N}$). What is the pattern?

Except for the 0, the pattern in each line is a cycle. If you analyze the periods of the cycles, you'll find that it is always some divisor of $p - 1$. For example, in the table for $\mathbb{Z}_7$, one element (6) cycles with a period of 2, two elements (2, 4) cycle with a period of 3 and two elements (3, 5) cycle with a period of 6.

$\mathbb{Z}_2 :$	$\wedge$	0	1	2	3	4	5	6	7	8	...
	0	1	0	0	0	0	0	0	0	0	...
	1	1	1	1	1	1	1	1	1	1	...
$\mathbb{Z}_3 :$	$\wedge$	0	1	2	3	4	5	6	7	8	...
	0	1	0	0	0	0	0	0	0	0	...
	1	1	1	1	1	1	1	1	1	1	...
	2	1	2	1	2	1	2	1	2	1	...

Figure 1.18: Exponentiation tables for $\mathbb{Z}_2$ and $\mathbb{Z}_3$

	$\wedge$	0	1	2	3	4	5	6	7	8	...
	0	1	0	0	0	0	0	0	0	0	...
	1	1	1	1	1	1	1	1	1	1	...
$\mathbb{Z}_5 :$	2	1	2	4	3	1	2	4	3	1	...
	3	1	3	4	2	1	3	4	2	1	...
	4	1	4	1	4	1	4	1	4	1	...

	$\wedge$	0	1	2	3	4	5	6	7	8	...
	0	1	0	0	0	0	0	0	0	0	...
	1	1	1	1	1	1	1	1	1	1	...
$\mathbb{Z}_7 :$	2	1	2	4	1	2	4	1	2	4	...
	3	1	3	2	6	4	5	1	3	2	...
	4	1	4	2	1	4	2	1	4	2	...
	5	1	5	4	6	2	3	1	5	4	...
	6	1	6	1	6	1	6	1	6	1	...

Figure 1.19: Exponentiation tables for $\mathbb{Z}_5$ and $\mathbb{Z}_7$

- Can you see why we didn't define exponentiation so that the exponent is also from $\mathbb{Z}_p$?

It would break the homomorphism between $\mathbb{Z}$ and $\mathbb{Z}_m$. Take for example the expression 2^7 :

$$\begin{array}{ll}
 2^7 = 128 & \text{(operation in } \mathbb{Z}) \\
 128 \mapsto 3 & (\mathbb{Z} \rightarrow \mathbb{Z}_5) \\
 \\
 2^7 \mapsto 2^2 & (\mathbb{Z} \rightarrow \mathbb{Z}_5) \\
 2^2 = 4 & \text{(operation in } \mathbb{Z}_5)
 \end{array}$$

Since we want fields to mirror the behavior of real numbers, we define exponentiation with exponents from $\mathbb{Z}$ so that they are not reduced modulo p .

Thanks to the fact that finite fields behave like real numbers, the fact that an n -th degree polynomial has at most n roots holds, and the proof of the theorem that an n -th degree polynomial is uniquely determined by $n + 1$ points, too, except for two caveats.

First, the exponentiation cycling makes different polynomials behave like the same functions. For example, the polynomials $f(x) = x^2$ and $g(x) = 1$ over $\mathbb{Z}_3$ would behave as the same function. Therefore, when working in $\mathbb{Z}_p$, we should only consider polynomials of degree $p - 2$ or lower.

And second, regardless of the polynomial degree, we may only generate p distinct points because that's the size of the domain.

1.7 Reed–Solomon codes

We're finally ready to use the polynomial approach in finite fields. Let's try the two methods from section 1.5, but over finite fields.

- *Encode the message 1997 using a polynomial code that can correct 2 errors. Choose the smallest finite field that allows this.*

A message of length 4 leads to a 3rd degree polynomial, which we will use to generate 8 function values.

To generate 8 function values, we need the domain of the function to be at least 8, so the smallest prime number $p \geq 8$ is 11. Therefore, we'll use $\mathbb{Z}_{11}$.

First method

Using the first method, we set

$$f(x) = 1x^3 + 9x^2 + 9x + 7$$

and calculate the codeword

$$\mathbf{w} = (f(0), f(1), f(2), f(3), f(4), f(5), f(6), f(7)) = (7, 4, 3, 10, 9, 6, 7, 7).$$

To decode it, we go through the 6-element subsets of the set of points given by the codeword, and we solve the overdetermined systems of equations such as

$$\begin{aligned} 0^3 \cdot c_1 + 0^2 \cdot c_2 + 0 \cdot c_3 + c_4 &= 7 \\ 1^3 \cdot c_1 + 1^2 \cdot c_2 + 1 \cdot c_3 + c_4 &= 4 \\ 2^3 \cdot c_1 + 2^2 \cdot c_2 + 2 \cdot c_3 + c_4 &= 3 \\ 3^3 \cdot c_1 + 3^2 \cdot c_2 + 3 \cdot c_3 + c_4 &= 10 \\ 4^3 \cdot c_1 + 4^2 \cdot c_2 + 4 \cdot c_3 + c_4 &= 9 \\ 5^3 \cdot c_1 + 5^2 \cdot c_2 + 5 \cdot c_3 + c_4 &= 6. \end{aligned}$$

If we find a solution (c_1, c_2, c_3, c_4) , that is the message.

Second method

Using the second method, we first find the 3rd degree polynomial $g(x)$ for which

$$\begin{aligned} g(0) &= 1 \\ g(1) &= 9 \\ g(2) &= 9 \\ g(3) &= 7. \end{aligned}$$

We do that by solving

$$\begin{aligned} 0^3 \cdot c_1 + 0^2 \cdot c_2 + 0 \cdot c_3 + c_4 &= 1 \\ 1^3 \cdot c_1 + 1^2 \cdot c_2 + 1 \cdot c_3 + c_4 &= 9 \\ 2^3 \cdot c_1 + 2^2 \cdot c_2 + 2 \cdot c_3 + c_4 &= 9 \\ 3^3 \cdot c_1 + 3^2 \cdot c_2 + 3 \cdot c_3 + c_4 &= 7. \end{aligned}$$

The solution $(1, 3, 4, 1)$ leads to the polynomial $g(x) = x^3 + 3x^2 + 4x + 1$. We use it to generate 4 more points

$$\begin{aligned} g(4) &= 1 \cdot 4^3 + 3 \cdot 4^2 + 4 \cdot 4 + 1 = 8 \\ g(5) &= 1 \cdot 5^3 + 3 \cdot 5^2 + 4 \cdot 5 + 1 = 1 \\ g(6) &= 1 \cdot 6^3 + 3 \cdot 6^2 + 4 \cdot 6 + 1 = 8 \\ g(7) &= 1 \cdot 7^3 + 3 \cdot 7^2 + 4 \cdot 7 + 1 = 2. \end{aligned}$$

The codeword is

$$\mathbf{w} = (g(0), g(1), g(2), g(3), g(4), g(5), g(6), g(7)) = (1, 9, 9, 7, 8, 1, 8, 2).$$

Decoding is analogical to the previous method: we go through the 6-element subsets and we solve the overdetermined systems of equations. Once we find a solution, we take that polynomial and use it to reconstruct the message (if necessary).

Both methods we just tried are the two original coding schemes proposed by *Irving S. Reed* and *Gustave Solomon* in 1960 [16]. The advantage of the first one is slightly faster for encoding, while the second one produces a **systematic code**, which means that the encoded message contains the message literally.

If you tried solving the systems on your own, you felt how time-consuming the decoding is. If two errors happen, you have to solve up to $\binom{8}{6} = 28$ systems. And that's just 4 message symbols and 2 errors, way less than what we need in practice. To do this with a block length 50 and 5 errors, you have up to 2,118,760 systems of equations to solve per block.

- Does any of these methods produce a linear code?

In the first method, any codeword is of the form

$$\mathbf{w} = \begin{pmatrix} m_4 \\ m_1 + m_2 + m_3 + m_4 \\ 8m_1 + 4m_2 + 2m_3 + m_4 \\ 27m_1 + 9m_2 + 3m_3 + m_4 \end{pmatrix},$$

$$m_1, m_2, m_3, m_4 \in \mathbb{Z}_p.$$

The code is linear if the sum of two codewords is generally a codeword, so

$$\begin{aligned}
 \mathbf{v} + \mathbf{w} &= \begin{pmatrix} a_4 \\ 1^3 a_1 + 1^2 a_2 + a_3 + a_4 \\ 2^3 a_1 + 2^2 a_2 + 2a_3 + a_4 \\ \vdots \\ 7^3 a_1 + 7^2 a_2 + 7a_3 + a_4 \end{pmatrix} + \begin{pmatrix} b_4 \\ 1^3 b_1 + 1^2 b_2 + b_3 + b_4 \\ 2^3 b_1 + 2^2 b_2 + 2b_3 + b_4 \\ \vdots \\ 7^3 b_1 + 7^2 b_2 + 7b_3 + b_4 \end{pmatrix} = \\
 &= \begin{pmatrix} a_4 + b_4 \\ a_1 + a_2 + a_3 + a_4 + b_1 + b_2 + b_3 + b_4 \\ 2^3 a_1 + 2^2 a_2 + 2a_3 + a_4 + 2^3 b_1 + 2^2 b_2 + 2b_3 + b_4 \\ \vdots \\ 7^3 a_1 + 7^2 a_2 + 7a_3 + a_4 + 7^3 b_1 + 7^2 b_2 + 7b_3 + b_4 \end{pmatrix} = \\
 &= \begin{pmatrix} a_4 + b_4 \\ (a_1 + b_1) + (a_2 + b_2) + (a_3 + b_3) + (a_4 + b_4) \\ 2^3(a_1 + b_1) + 2^2(a_2 + b_2) + 2(a_3 + b_3) + (a_4 + b_4) \\ \vdots \\ 7^3(a_1 + b_1) + 7^2(a_2 + b_2) + 7(a_3 + b_3) + (a_4 + b_4) \end{pmatrix}
 \end{aligned}$$

where $(a_1 + b_1), (a_2 + b_2), (a_3 + b_3), (a_4 + b_4)$ are from $\mathbb{Z}_p$, so $\mathbf{v} + \mathbf{w}$ is a valid codeword.

In the second method, the same argument shows that for any sum of two codewords $\mathbf{v} + \mathbf{w}$, there is a 3rd degree polynomial of the form

$$f(x) = (a_1 + b_1)x^3 + (a_2 + b_2)x^2 + (a_3 + b_3)x + (a_4 + b_4),$$

which guarantees the existence of a respective original message

$$\begin{pmatrix} (a_1 + b_1)0^3 + (a_2 + b_2)0^2 + (a_3 + b_3)0 + (a_4 + b_4) \\ (a_1 + b_1)1^3 + (a_2 + b_2)1^2 + (a_3 + b_3)1 + (a_4 + b_4) \\ (a_1 + b_1)2^3 + (a_2 + b_2)2^2 + (a_3 + b_3)2 + (a_4 + b_4) \\ (a_1 + b_1)3^3 + (a_2 + b_2)3^2 + (a_3 + b_3)3 + (a_4 + b_4) \end{pmatrix}.$$

So, both codes are linear.

- *Imagine that you know that a polynomial $f(x)$ produces the codeword $(1, 0, 0, 0)$ and the polynomial $g(x)$ produces the codeword $(0, 1, 0, 0)$. How can you find the polynomial that produces the codeword $(1, 1, 0, 0)$? How to turn this into a trick for easy finding of an n -th degree polynomial that passes through $n + 1$ points? (Hint: Find some universally useful polynomials to be used repeatedly.)*

The first question is rather easy: it will be the polynomial $f(x) + g(x)$. Adding the coefficients of polynomials affects the function values in the very same way. This leads to an easy but very helpful trick. We can pre-calculate the polynomials that produce the words $(1, 0, \dots, 0), (0, 1, \dots, 0), \dots, (0, 0, \dots, 1)$, and use these to easily calculate a polynomial corresponding to any codeword.

- Calculate the basis polynomials in $\mathbb{Z}_{11}$. Use them to find the polynomial of the word 1997.

We are looking for the 3rd degree polynomials that produce the words 1000, 0100, 0010, 0001.

word	points	polynomial
1000	$\{(0, 1), (1, 0), (2, 0), (3, 0)\}$	$9x^3 + x^2 + 1$
0100	$\{(0, 0), (1, 1), (2, 0), (3, 0)\}$	$6x^3 + 3x^2 + 3x$
0010	$\{(0, 0), (1, 0), (2, 1), (3, 0)\}$	$5x^3 + 2x^2 + 4x$
0001	$\{(0, 0), (1, 0), (2, 0), (3, 1)\}$	$2x^3 + 5x^2 + 4x$

Table 1.3: Lagrange interpolation

These are now our basis polynomials. To find the polynomial that produces the codeword 1997, we take the respective linear combination of the basis polynomials:

$$\begin{aligned}
 &1(9x^3 + x^2 + 1) + 9(6x^3 + 3x^2 + 3x) + 9(5x^3 + 2x^2 + 4x) + 7(2x^3 + 5x^2 + 4x) = \\
 &= 1x^3 + 4x^2 + 3x + 1
 \end{aligned}$$

This speeds up the process of checking each subset, but still, the problem is the number of the subsets—it grows super-linearly as we protect against more errors. So, we’re still not in the error-correcting Garden of Eden. We can construct perfect error-correcting codes with any minimal distance, but we don’t have a truly efficient decoding mechanism.

For fast decoding, we would need some low-complexity algorithm that crunches the erroneous message and outputs the instructions to correct it—just like the decoding of the Hamming codes. This was later achieved with **BCH codes**. These require deeper understanding of algebra, and are based on Galois fields, which are the finite fields that haven’t been covered in this chapter.

For all interested in the matter of coding theory, the book *A First Course in Coding Theory* by Raymond Hill [12] is highly recommended.

Chapter 2

Inferential statistics

Humans are naturally curious. We observe events, and whether or not we systematically record what we see, hear or feel, our brains are trying to make sense of the data.

Undoubtedly, there are causes and effects in our world. Some things are predictable with high level of certainty: for instance, if you drop an object that is heavier than air, it will fall to the ground. Most of the things we deal with in our everyday lives are less certain. Not that they would like to break the rules of physics—but they are parts of complex systems, where causes and effects are deeply intertwined, making it challenging to unravel them.

What usually helps in the process of making sense of the world is being systematic. Breaking down the complexity into smaller, more manageable parts. Designing experiments. Measuring and writing down the data. And finally, transforming this data to knowledge—ideally, to propositions with some specified level of certainty.

We'll swiftly guide you from high-school-level combinatorics and statistics to conducting experiments and testing statistical hypotheses. We'll reason through the basic methods of inferential statistics, the part of statistics which is concerned with the final part of the process, drawing conclusions and asserting statements about the world.

2.1 Data types

At the heart of statistical analysis is data. You must have at least once collected some! If you collect data systematically, you usually organize them in a table. For each observation, you have a row, and for each characteristic, you have a column. If, for example, we conduct a survey by interviewing random people on the street, then each row represents an interview, and each column contains the answers to one question.

In statistics, the rows are called **objects**, and the columns are called **attributes**.

Mathematically, objects are n -tuples consisting of attribute values in a pre-specified order, and the set of objects is called a **sample**.¹

Let's say we have collected data by interviewing random people on the street. We asked them:

- when they woke up this morning,
- what political party they voted for in the last election,
- to what extent they agree with the proposition *Ownership of more than one property should be progressively taxed*. (answers: *strongly disagree, disagree, neutral, agree, strongly agree*),
- how much money they have in their primary bank account in CZK.

For the sake of simplicity, we excluded all respondents who were not willing to answer one or more questions. Note that this is a made-up survey; we do not recommend asking strangers such questions in reality! The results might look like this:

ID	Political party	Property tax	Wake-up time	Bank balance
1	SPOLU	strongly agree	6:24	1,536,098.28
2	did not vote	agree	7:13	465,004.89
3	ANO 2011	neutral	8:55	115,754.50
4	ČSSD	strongly agree	8:03	84,834.06
5	SPOLU	neutral	7:42	6,937,123.02
6	Pir-STAN	disagree	8:15	32,897.53

Table 2.1: Example of a sample

- Can you characterize the differences between the attribute types? (Hint: Consider the nature of the sets of possible answers.)

We first notice that some questions ask for a number, some for a word-answer.

For the political party vote question, the set of possible answers includes the political parties that could be voted for, and *did not vote*. For the property tax opinion, there are obviously the five fixed options. For the wake-up time, the answer is a time (theoretically any time from 0:00 to 23:59—we do not say what *this morning* means, so for some people it might be very early or very late). Finally, the answer for the bank balance can be any number with a given decimal precision, ranging from 0 to infinity.

These domains are not just sets—they have some structure if we consider what the individual elements mean, how we can compare them, interpret their differences, and so on.

¹The term *set* is used to emphasize that the order of objects has no significance. However, it entails the inaccuracy that a set cannot contain two identical elements, while a sample can contain two identical objects. To fix this, we can add an identifier attribute to make every object distinct.

In the *Political party* attribute, the values are categories, and we can only determine whether they are the same or different. Such attributes are called **categorical**.

In the *Property tax* attribute, the values are categories with a meaningful order. We can say that the first respondent agrees more with the proposition than the second, but we cannot say how much. We also cannot say that the first respondent agrees more than the second one *to the extent* that the second one agrees more than the third one. Even if we asked the respondents to answer on a scale from 1 to 5, we still couldn't meaningfully interpret the differences—even though technically the numbers allow that. This type of attribute is called **ordinal**.

In the *Wake-up* time attribute, the values are ordered, and we can also compare their differences. For example, the difference between 6:24 and 7:13 (49 minutes) is less than half the difference between 7:13 and 8:55 (102 minutes). However, it is meaningless to say that the second respondent woke up 13 % later than the first respondent—we cannot say this because there is *no natural zero*. We call this attribute type **interval-scaled**.

In the *Bank balance* attribute, we can not only order the values and compare their differences, but also interpret their ratios—for example, the balance of the first respondent is approximately 3.3 times larger than the balance of the second respondent. This property leads to the attribute type being called **ratio-scaled**.

Sometimes it is useful to lump together the last two types. For that, we use the term **numerical attribute**.

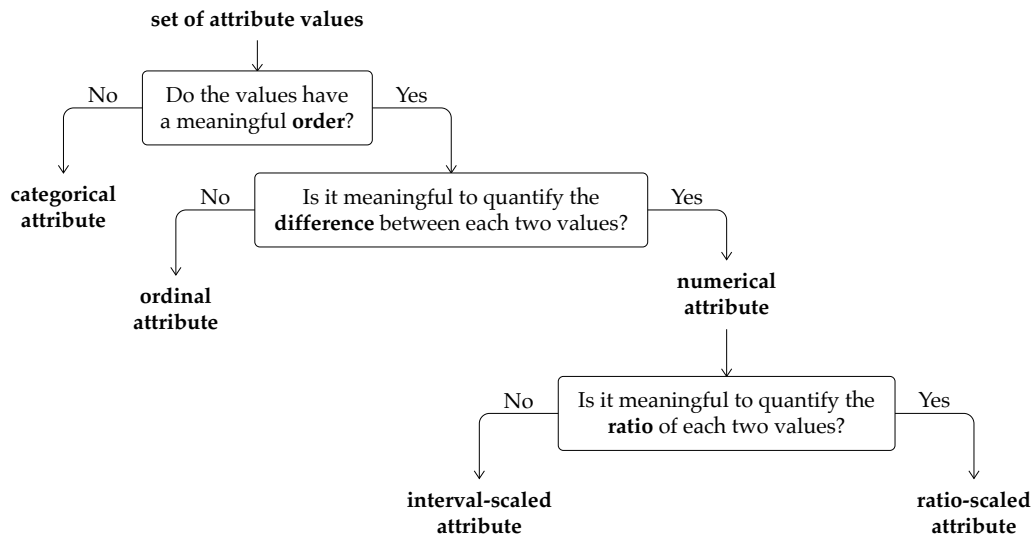


Figure 2.1: Attribute types

Knowing these types is vital not only for interpreting the data, but also for performing any statistical analysis. What you could notice is that the order in which we introduced the types wasn't random but deeply logical—each type has the properties of all the previous types. Therefore, the types are 'backwards compatible'—we can always treat an attribute as a lower type if needed.

2.2 Describing the samples

The basic thing to do in statistics is to describe the sample data by an individual number (or an attribute value). Such value that is determined by some formula is called a **statistic**. For example, one well-known statistic is the arithmetic mean.

- *What aspects of the data in one attribute of a sample might we want to describe by a single number?*

The two basic aspects are location and variability. To describe these aspects, we use certain statistics called **measures of location** and **measures of variability**. Measures of location indicate *where the data are*, and measures of variability indicate *how spread out they are*.

- *For each attribute type, which statistics serve as measures of location and measures of variability? What should and what shouldn't affect their values?*

Let's answer the second question first. It would be nice if

- they weren't affected by the sample size (the number of objects in the sample),
- the measure of location wasn't affected by the variability of the values and vice versa,

and for the numerical attributes if

- they were in the same units as the attribute values,
- shifting all values by the same offset shifted the measure of location by the same offset and didn't change the measure of variability.

To find out what works and what doesn't work, you might find helpful to fix a sample and try out different statistics. Once you have a candidate statistic, model small samples so that the given statistic results in the same values, and then ask: *Are these samples equivalent in terms of the measured aspect?*

Measures of location

Measures of location are well known and frequently used in everyday life. In numerical attributes, we can calculate the **arithmetic mean**—sum the values and divide by their count. In ordinal attributes, we order the values and identify the one in the middle.² This is also a well-known statistic—the **median**. Finally, in categorical attributes, the best we can do is find the most common value, known as the **mode**. As written earlier, we can find and use median and mode for the attributes with a richer structure, too (but with caution—if used inappropriately, it might do more harm than good).

²When there is an even number of values, we take the mean of the two values around the middle point. If the attribute type does not allow that, the median is not defined.

Example 1.

Attribute values: 5, 8, 5, 7, 6, 12, 4

Arithmetic mean: $\frac{5+8+5+7+6+12+4}{7} \approx 6.71$

Median: 4, 5, 5, 6, 7, 8, 12 $\Rightarrow 6$

Mode: 5

Example 2.

Attribute values: 10, 9, 3, 6, 8, 4, 4, 8

Arithmetic mean: $\frac{10+9+3+6+8+4+4+8}{8} = 6.5$

Median: 3, 4, 4, 6, 8, 8, 9, 10 $\Rightarrow \frac{6+8}{2} = 7$

Mode: not defined (no frequency is strictly highest)

Measures of variability

Measures of variability are not frequently used in everyday life, but are crucial in statistics as they relate to random error and uncertainty. We will only define them for numerical data. These data are numerical values on a measurable axis, so we want to measure how far these values are from the center. Taken literally, we arrive at the **mean absolute deviation**—the arithmetic mean of the absolute differences between each value and the arithmetic mean.

Although intuitive and easy to calculate by hand, this statistic is not often used in statistical practice. This is because of multiple reasons, one of which is that the arithmetic mean is actually not *the center* with respect to the semi-metric of absolute differences. By the center we mean the point on the number line that minimizes the sum of distances from the values, and by the semi-metric we mean the function assigning distances to pairs of numbers.

Definition. A function $d: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}_0^+$ that satisfies the properties

$$1. \forall a, b \in \mathbb{R}: \quad d(a, b) = 0 \iff a = b$$

$$2. \forall a, b \in \mathbb{R}: \quad d(a, b) = d(b, a)$$

is called a **semi-metric** on $\mathbb{R}$.

Definition. Let d be a semi-metric on $\mathbb{R}$ and let $x_1, \dots, x_n$ be real numbers. The number $c \in \mathbb{R}$ satisfying

$$\forall y \in \mathbb{R}: \sum_{i=1}^n d(c, x_i) \leq \sum_{i=1}^n d(y, x_i)$$

is called the **center** of the numbers $x_1, \dots, x_n$ with respect to the semi-metric d .

If we look at Example 1, the sum of distances from the mean are approximately $1.71 + 1.39 + 1.71 + 0.39 + 0.71 + 5.39 + 2.71 = 14.01$ while the sum of distances from the median are $1 + 2 + 1 + 1 + 0 + 6 + 2 = 13$. In fact, the median is the center with respect to the semi-metric of absolute differences.³

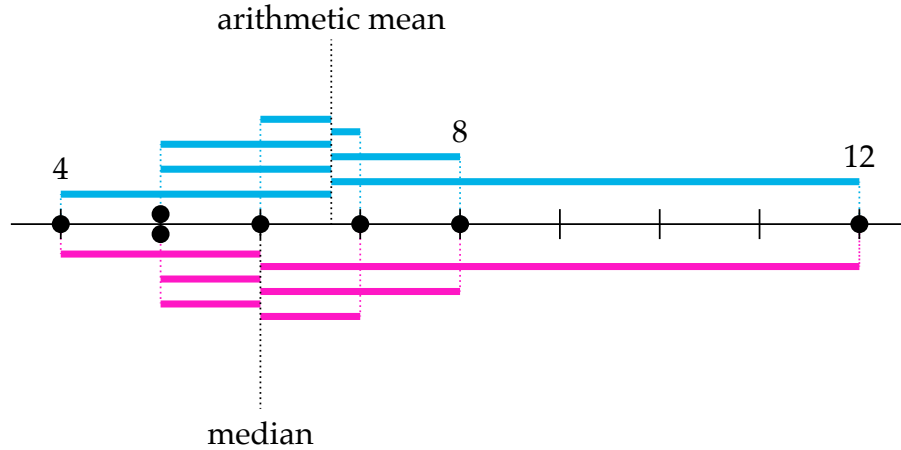


Figure 2.2: Absolute deviations

The more common measures of variability are **variance** and **standard deviation**. Variance is the arithmetic mean of the squared deviations (deviation is just another word for the difference of the mean and the particular value). In the semi-metric of squared deviations, the arithmetic mean is the center, so it fixes the previous issue. But, since the deviations are squared, the unit of variance is the square of the unit of the values. As mentioned earlier, it is desirable that the unit of the statistic is the same as the unit of the values—so to fix this, we take a square root of variance, and that is the standard deviation.

³For even number of objects n , the center is also any number between the $\frac{n}{2}$ -th and the $(\frac{n}{2} + 1)$ -th value.

Definition. Let $x_1, x_2, \dots, x_n \in \mathbb{R}$ be values in a sample, and let $\bar{x}$ represent their arithmetic mean. Then

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

is the sample **variance** and

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

is the sample **standard deviation**.

2.3 From frequencies to probabilities

So far, we have only done descriptive statistics. Descriptive statistics focuses only on describing the data we collected. It produces statements that are 100 % certain, like *A third of people in our sample voted for SPOLU in the last election*. However, our mission is to learn to generalize from this, so that we can estimate when a random person from our city wakes up or the probability that they agree with the proposition *Ownership of more than one property should be progressively taxed*.

In the previous sections, we operated with the term *sample*. A sample is taken randomly from a **population**. For example, the sample of six respondents of our survey was taken from the population of people living in the city where the survey was conducted. Using samples instead of populations is practical

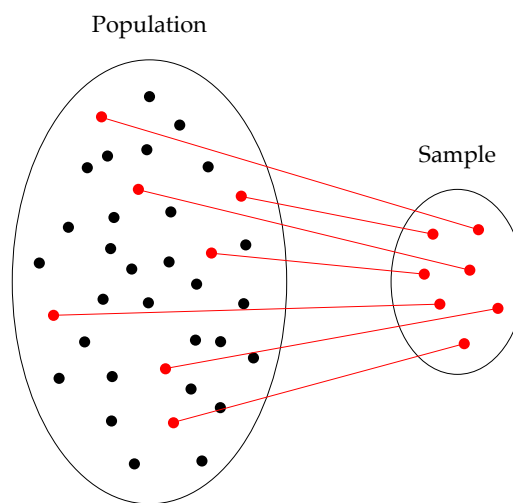


Figure 2.3: Population and sample

for several reasons. First, it is easier to interview a thousand randomly selected people than ten million inhabitants of the Czech Republic. But sometimes it is even impossible to analyze the whole population because it could be infinite—for example, if we were interested in the rate of faulty products produced by some machine, we could theoretically leave the machine running forever. We have to decide how many products we will let the machine produce before ending the experiment, and that will be our sample.

In the inductive process, we assume that there is a fixed probability of each outcome, for example that a probability of a faulty product is 6 %. Then, as we make more observations, the relative frequency will approach 6 out of 100. Note that the last sentence is a mathematical statement, and we can prove that it is true. However, the converse is not: if it looks like the relative frequency approaches some value as we make more observations, we cannot be sure that there actually exists that fixed probability governing the stochastic process.

- Imagine that you conduct the survey from table 2.1 with a large sample of people. How can you visualize the data in each attribute?

For categorical and ordinal attributes, the important information is *how many* objects in the sample have a given attribute value. These quantities can be effectively visualized by columns, leading to a **column chart**. For ordinal data, we order the columns by the implicit order of the values. For categorical attributes, there is no implicit order, so we can order them as we like. Popular choices include alphabetical order or descending order based on frequency.

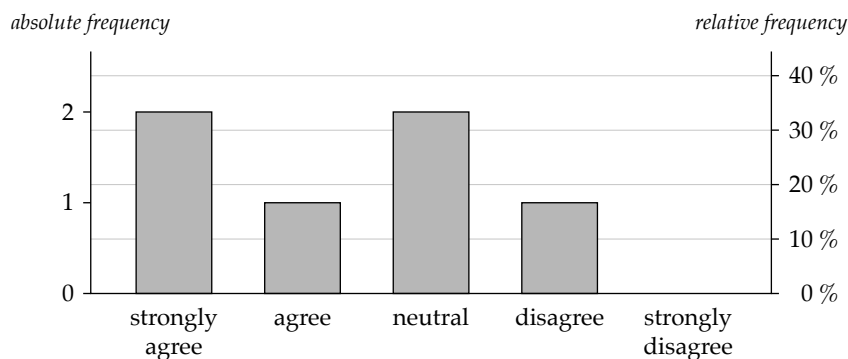


Figure 2.4: Column chart

Note that the heights of the columns not only show the absolute frequencies of the values but also their relative frequencies in the sample, if we relabel the y -axis.

For numerical attributes, there are usually too many values, so a column chart would be too wide and hard to read. To reduce the number of columns, we divide the domain into intervals and sort the values into bins based on the interval they fall into. This process of transforming (more-or-less) continuous data into discrete is known as **discretization** or **data binning**.

For example, the intervals for the wake-up time might be *before 6:00*, *6:00–6:59*, *7:00–7:59*, *8:00–8:59*, and *after 8:59*. A column chart of this type is called a **histogram**.

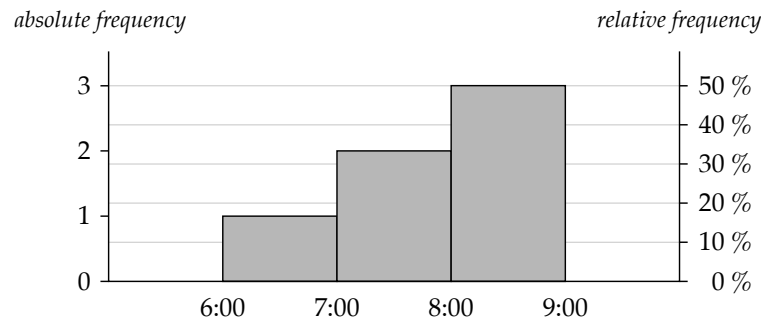


Figure 2.5: Histogram

► *What information is lost during discretization?*

When we discretize the data, we basically round the numbers. In doing this, we lose detail, as 7:01 falls into the same bin as 7:59. The rounding errors do partially cancel out, but if we want to reduce the overall rounding error, we can discretize more finely (by dividing the axis into more narrower intervals), if the sample size allows it.

Now, back to the assumption that the relative frequencies of attribute values approach their true probability as we increase the sample size. The column chart then becomes a **probability distribution**, a function whose domain is the set of all possible attribute values, and whose codomain is the set of real numbers from 0 to 1, describing the probability of observing a given attribute value in a randomly selected object. Since all the values are disjoint states, the probabilities must sum to 1.

For categorical and ordinal data, the probability distribution resembles the column chart; the only difference is that the y -axis represents probability, not frequency.

What about numerical data? If we don't round the measurements (i.e. the attribute values are from a continuous interval), the situation is different in that there are infinitely many possibilities for an outcome. The probability of obtaining any specific real number is 0, yet the sum of these probabilities must equal 1. These paradoxes involving infinities and infinitesimals are typically resolved by introducing limits, so let's take that perspective.

Imagine that we measured the height of infinitely many people. We make a column chart binning the values into 2 intervals in the first step, then into 4 intervals in the second step, then into 8 intervals, and so on. In each step, we adjust the height of the columns so that the sum of their areas is constant. When we reach infinity, there is a column for every real number, and the columns become indistinguishable

from vertical line segments. The top borders of the columns become points, and their union is a smooth curve.

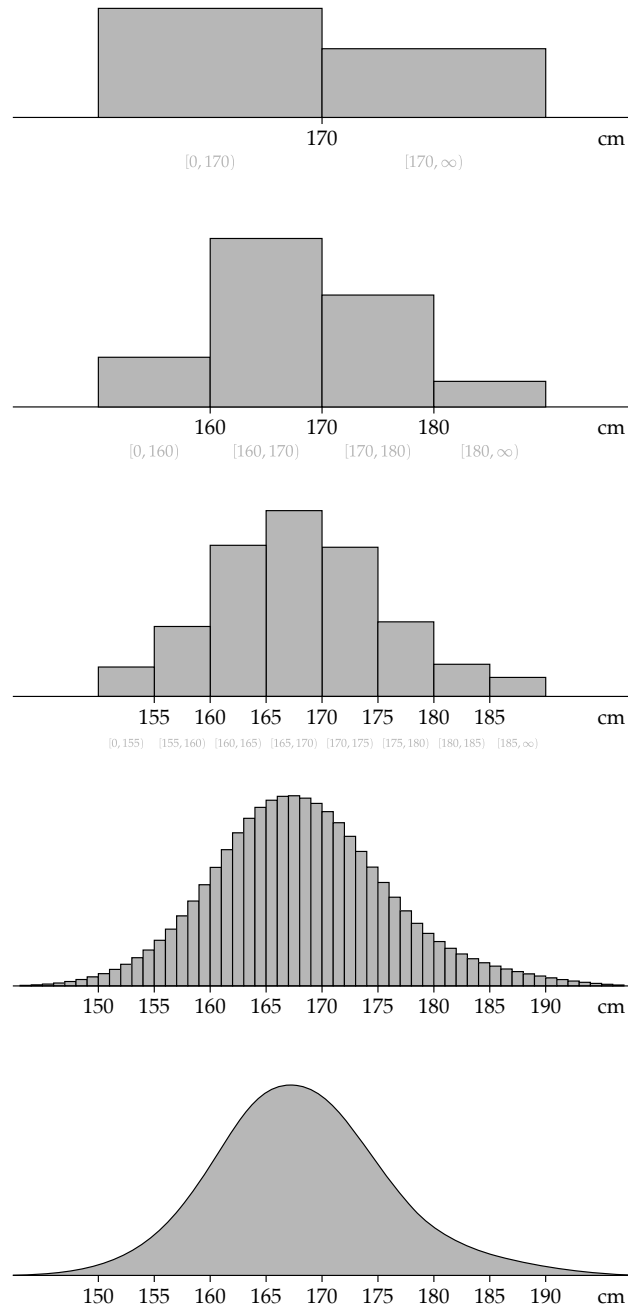


Figure 2.6: Continuous probability density function

► *So how do we read the probability from the function now?*

Instead of focusing on the function values, we should pay attention to the area under the curve for specific intervals. Let's select the interval 160–165 cm.

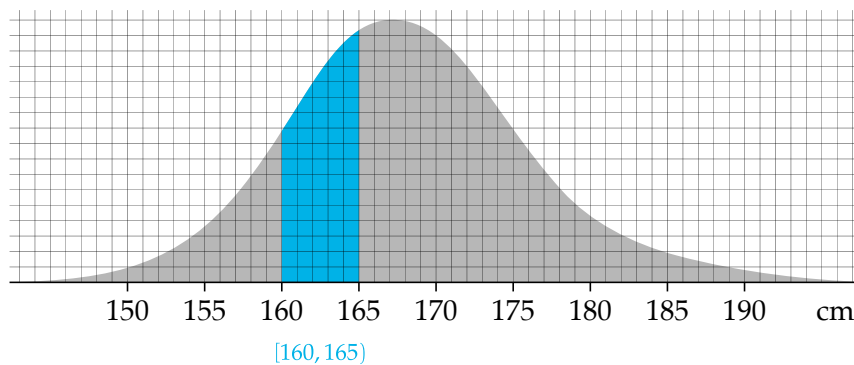


Figure 2.7: Reading from a probability density function

The area under the curve over this interval looks something like 20 % of the whole area. So the probability of a random person being 160–165 cm tall is ~20 %.

Note that since the function is continuous, we can choose the interval freely—it can be 160.0–160.1 cm if we wanted to.

► *How do we interpret the y-axis of the graph then?*

Let's reason from what we know: we know that the area under the graph is the probability. The area can be approximated as a rectangle whose one side is an interval in centimeters. We can formulate the equation

$$\text{probability (no unit)} = \text{height (cm)} \cdot ? (?).$$

For the equation to make sense, the question mark must be in the units of cm^{-1} , and the quantity is probability per centimeter, or probability density. This is why the function is called the **probability density function**. Roughly speaking, it is the probability of a randomly chosen person's height rounded to the nearest centimeter.

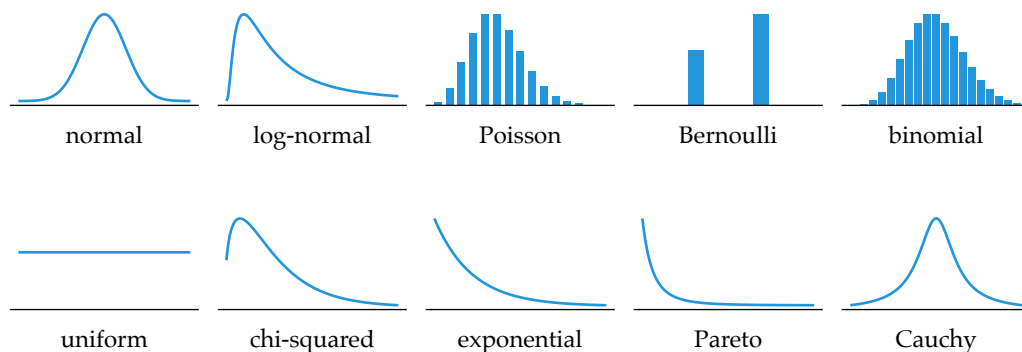


Figure 2.8: Different types of probability distributions

Let's now focus on what shapes these distributions take. There are various types of probability distributions. They primarily depend on the domain over which they are defined, but also on the nature of the stochastic process generating the observed data. The simplest example is coin tossing.

- Assuming that the coin is fair, what is the probability that a given side will land exactly k times in n coin tosses? Draw the probability distribution over the possible values of k after 10 tosses.

If we toss a coin three times, we can get one of four outcomes: 0 heads, 1 head, 2 heads, or 3 heads. It is crucial to realize that there are multiple ways to get 2 heads in 3 tosses: HHT, HTH, and THH; but only one way to get 3 heads in 3 tosses: HHH. The *number of ways* is the binomial coefficient $\binom{n}{k}$, which we can also find in Pascal's triangle.

To get a probability, we must divide this number by the number of all binary n -tuples (HHH, HHT, HTH, HTT, THH, THT, TTH, TTT). There are 2^n of those (which corresponds to the fact that each line of Pascal's triangle sums to a power of two).

The probabilities for the outcomes of 10 tosses are shown in the table below.

number of heads landed	probability
0	$\frac{1}{2^{10}} = \frac{1}{1024} \approx 0.10 \%$
1	$\frac{\binom{10}{1}}{2^{10}} = \frac{10}{1024} \approx 0.98 \%$
2	$\frac{\binom{10}{2}}{2^{10}} = \frac{45}{1024} \approx 4.39 \%$
3	$\frac{\binom{10}{3}}{2^{10}} = \frac{120}{1024} \approx 11.72 \%$
4	$\frac{\binom{10}{4}}{2^{10}} = \frac{210}{1024} \approx 20.51 \%$
5	$\frac{\binom{10}{5}}{2^{10}} = \frac{252}{1024} \approx 24.61 \%$
6	$\frac{\binom{10}{6}}{2^{10}} = \frac{210}{1024} \approx 20.51 \%$
7	$\frac{\binom{10}{7}}{2^{10}} = \frac{120}{1024} \approx 11.72 \%$
8	$\frac{\binom{10}{8}}{2^{10}} = \frac{45}{1024} \approx 4.39 \%$
9	$\frac{\binom{10}{9}}{2^{10}} = \frac{10}{1024} \approx 0.98 \%$
10	$\frac{1}{2^{10}} = \frac{1}{1024} \approx 0.10 \%$
k	$\frac{\binom{n}{k}}{2^n}$

Figure 2.9: Probabilities of 10 fair coin tosses

Since the denominator is the same in each line, plotting the probability distribution after 10 tosses is as simple as plotting the numbers on the 11th line of Pascal's triangle.

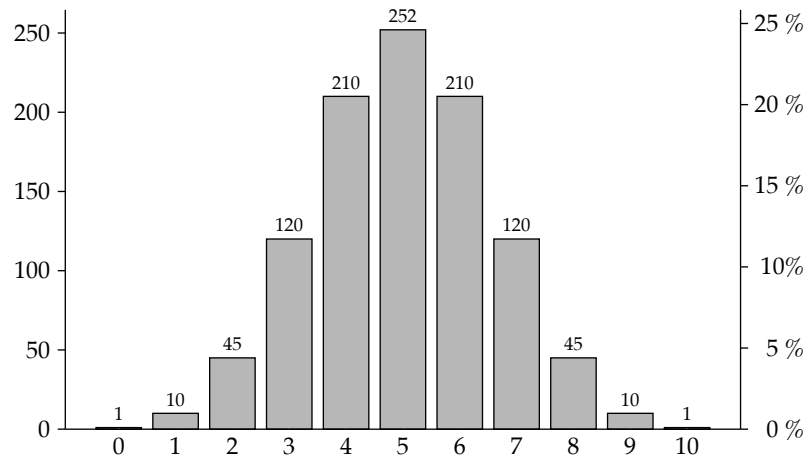


Figure 2.10: Binomial distribution of 10 fair coin tosses

As you can see, the shape of this distribution resembles the Gaussian curve. The larger the value of n , the smoother the curve. You can observe this by plotting subsequent lines of Pascal's triangle.

If we conducted an experiment with 1024 rounds of 10 coin tosses and created a column chart from the results, it is very unlikely we would obtain this distribution exactly. However, the overall shape of the column chart would resemble this ideal one. You might wonder: how can I tell if the deviation of my experimental results from the ideal distribution is due to random chance? And that is why we need hypothesis testing.

2.4 Testing simple hypotheses

Let's simplify the experiment to a single round of 10 tosses. We want to find out if the outcome aligns with what we would expect from a fair coin. A coin is fair if the probabilities of landing on each side are equal. So, our hypothesis is that the probability of landing heads is 50 %.

- *What outcomes would lead us to reject the hypothesis?*

Landing 0 heads or 10 heads out of 10 tosses would be very unlikely. But is 1 or 9 heads out of 10 still too unlikely? As you can see, there is no hard-and-fast rule; we must say how much bad luck is too unlikely so that we reject the hypothesis. That's why before running the experiment, we should state the **significance level** (α). To explain what a significance level is, let's color the binomial distribution.

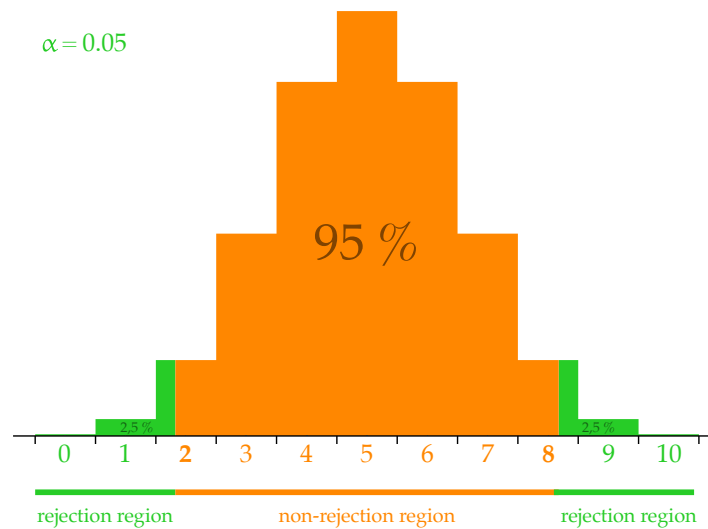


Figure 2.11: Rejection and non-rejection region

This binomial distribution shows the probabilities of the experiment outcomes, assuming that the hypothesis is true. The orange area is where most of the outcomes fall, and the green area is where only a small percentage of them lie. The significance level is the proportion of the green area to the total area. The most common significance level in science is 5 %, though this choice is largely due to convention. If the outcome falls into the green area, we reject the hypothesis. That's why we call this area the **rejection region**.

Note that the tails of the distribution were arbitrarily colored so that the left green tail comprises 2.5 % of the total area, and the right green tail comprises another 2.5 %. There are many ways how to color the distribution so that 5 % of it is green and the rest is orange!

- *Why did we color the graph like this? In other words, why do we reject the hypothesis in these 5 % of cases and not some other 5 % of cases?*

To answer this question, we need to revisit the formulation of the hypothesis. The hypothesis being tested is that the coin is fair. The complement to this hypothesis is that the coin is biased, in which case we would expect that either heads or tails will prevail. So, we in fact have two competing hypotheses: the coin is fair or the coin is biased.

This leads us to name these two competing hypotheses. The hypothesis that we are testing (that the coin is fair) is called the **null hypothesis** (H_0). The other hypothesis (that the coin is biased) that we test *against* is called the **alternative hypothesis** (H_1). The alternative hypothesis is the more exciting one—typically, it asserts that there exists some phenomenon, while the null hypothesis represents the scenario where the phenomenon is absent and the outcome is random.

Now, assume that we have a fair coin (we have tested it thoroughly, so we are sure about it). We wonder if putting a little glue on the heads side will bias it. In trying that, we noticed that when the coin falls on the sticky side, it doesn't bounce, but it bounces if it falls on the non-sticky side. This process should bias the coin to land on the tails side more frequently. Therefore, it's safe to assume that the glued coin is either biased in favor of tails or still fair, ruling out any bias toward heads.

- What is the alternative hypothesis and the null hypothesis in this case? What will the rejection region look like?

The best way to think about this is by imagining different concrete outcomes.

Imagine that we get 0 heads out of 10. That speaks for the hypothesis that gluing the coin works as expected (=the phenomenon is present). Now imagine that we get 5 heads out of 10. Well, that would likely happen if the coin was still fair, so gluing probably does not make a difference (=the phenomenon is absent). And finally, we get 10 heads out of 10. This is strange—we don't expect that this would happen because we *know* that the coin is either fair or biased in favor of tails. So, the most likely explanation is that the coin is fair and we were just very unlucky (=the phenomenon is absent).

Thus the alternative hypothesis is that the probability of heads is less than 50 %. The null hypothesis is the complement: the probability of heads is 50 % or more. The rejection region will therefore be only on the side where heads prevail.

These are the two basic types of statistical tests: **two-sided** and **one-sided**. We should choose a two-sided test if the suspected phenomenon may be located at either end of the distribution. One-sided test should be chosen when we think the phenomenon might be only located at one end.

In the one-sided test, the rejection region is one interval on one end of the domain. In the two-sided test, the rejection region is the union of two intervals on both ends of the domain. The rejection region is demarcated by the **critical value(s)**. Critical value is the most extreme value for which we still don't reject the null hypothesis.

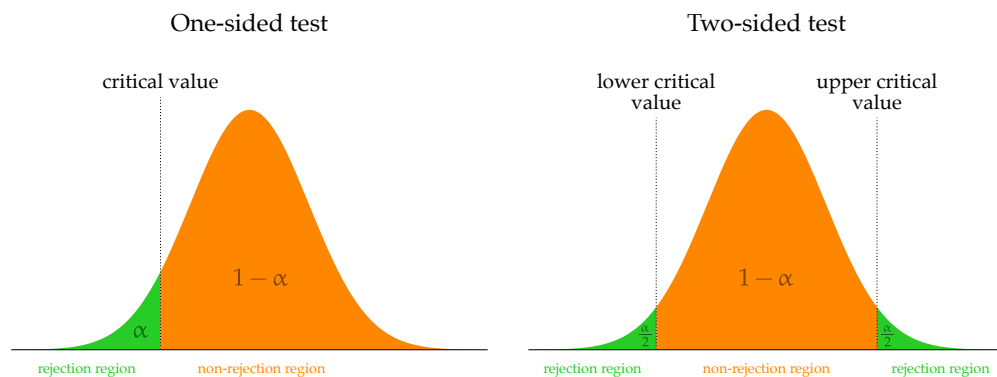


Figure 2.12: Types of statistical tests

Let's move to more practical usage of statistical testing. Last year, I noticed an article in the Czech newspaper Deník.cz with the title *Tragic Quarter: Record Number of Cyclists Died on Czech Roads* [6]. The article says that 15 cyclists died in the first quarter of the year 2023, and also includes the data from the first quarters of the years 2012 to 2022. The title could be even more dramatic if the journalist mentioned that in Q1 of 2022, only 5 cyclists died, so the number tripled from year to year!

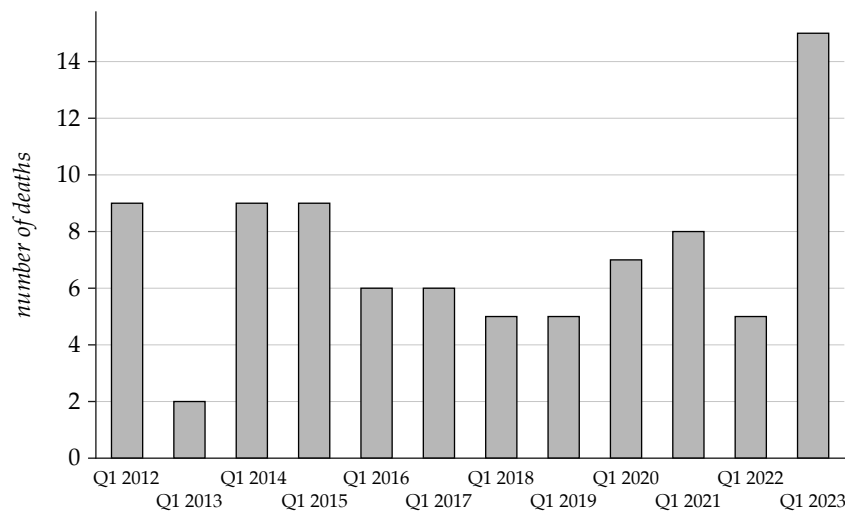


Figure 2.13: Cycling fatalities in the Czech Republic

The question is: is there some bad trend that was kicked off this year, or is the little peak random? We have the data from the past 11 years, where the numbers are relatively stable, so we can estimate the mean quantity of cycling fatalities in the first quarter, and formulate the null hypothesis.

► *Formulate the null hypothesis.*

There were 71 deaths in the Q1s of the years 2012 to 2022 in the Czech Republic, so there were on average 6.45 deaths per quarter, or one every 14.1 days. Thus, the null hypothesis is that the mean of the distribution of Q1 2023 is also 6.45. There is a priori no reason to expect a significantly high or significantly low number of fatalities, so we should stick to a two-sided test.⁴

But what is the distribution? As mentioned earlier, we should consider the nature of the stochastic process. We can assume that the fatalities happen independently. For car accidents, this would not be the case—there are often two or more dead

⁴Whether we use one-sided or two-sided test should always be decided before we collect data, as well as formulating the hypothesis and choosing the significance level. This is harder to do when we work with already existing data—the very first look at the data might bias our view and lead us to formulating a hypothesis that is *suited* for the data. With enough data, you can always find anomalies and tailor hypotheses that will be confirmed by them. That's why in science, it is a good practice to preregister hypotheses and methodologies before conducting experiments.

when a dangerous car accident happens. But it is quite rare that two or more cyclists die in the same accident. Also, we can assume that there is some ‘true probability’ of an accident every small unit of time, and this probability is all that generates the distribution. This distribution is known as Poisson distribution, but we will approximate it by a binomial distribution with an asymmetric probability.

So, a small enough unit of time might be an hour—given these numbers, it is negligibly unlikely that two cycling deaths happen in the same hour. We can then simplify our model—every hour, there is either 0 or 1 cycling fatality. The probability of a fatality in a given hour is $\frac{71}{11 \cdot 90 \cdot 24} \approx 0.3\%$.

- What would be the rejection region for this hypothesis, given a significance level of 0.05?

First, let’s sketch the distribution. How likely is it that no fatality happens in the whole quarter? That is when in every of the $90 \cdot 24 = 2160$ hours no fatality happens, that is $0.997^{2160} \approx 0.15\%$. How likely is it that exactly 1 fatality happens in the whole quarter? That is when in one of the 2160 hours, 1 fatality happens and in the rest, no fatality happens, that is $\binom{2160}{1} \cdot 0.003^1 \cdot 0.997^{2159} \approx 0.99\%$. Generally, the probability of k casualties in a given quarter is $\binom{2160}{k} \cdot 0.003^k \cdot 0.997^{2160-k}$. Plotting this, we get this distribution:

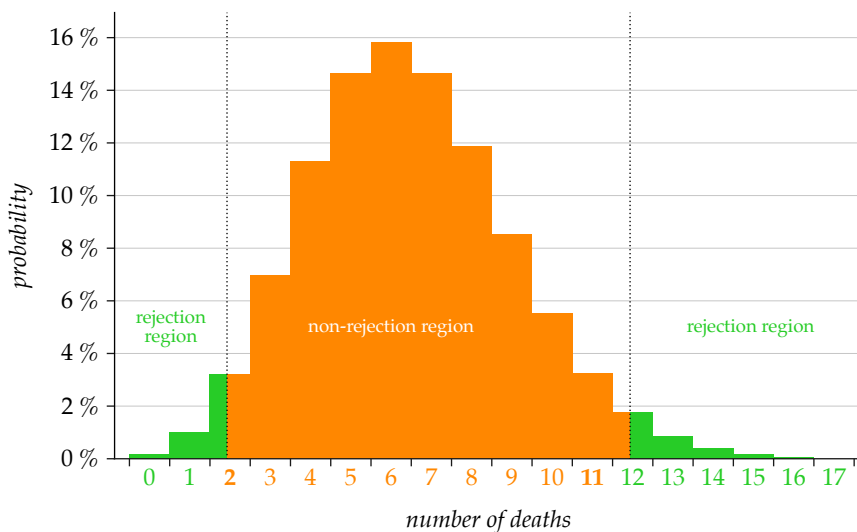


Figure 2.14: Poisson distribution of cycling fatalities

Because we are doing a two-sided test, we want to color approximately 2.5 % of the area in each tail of the distribution. Doing that, the critical value for the left tail will be 2 and the critical value for the right tail will be 11.

Since 15 lies in the rejection region, we can reject the hypothesis that the mean number of cycling fatalities in Q1 2023 is 6.45, under the assumption of a Poisson distribution at a 0.05 significance level.

2.5 Experiment design and non-parametric testing

Hypothesis testing is especially useful when you want to demonstrate if something is or is not effective. For example, UX designers might find it useful to evaluate whether a newly added feature increases a key metric, and pharmacologists often rely on hypothesis testing to understand the effectiveness of new drugs. However, before conducting any testing, it's crucial to design the experiment properly.

Suppose you are an athletics coach. You designed a long jump training program and you wonder whether it is effective. With 20 trainees who train weekly, you plan to assess the program's effectiveness using this plan:

- **Week 1 – Pre-training test:** Each trainee attempts the long jump 3 times. The best attempt is recorded as the trainee's pre-training test result.
- **Week 2 – Training:** Trainees complete the planned training program.
- **Week 3 – Post-training test:** Each trainee attempts the long jump 3 times again. The best attempt is recorded as the trainee's post-training test result.

Afterward, you statistically compare the results from the pre-training and post-training tests to determine whether the training program significantly improved the trainees' long jump performance.

► *What is the flaw of this procedure?*

The main issue is that the improvement might be attributed to factors other than the training program itself. First, the trainees could have improved simply by doing the test for a second time. If they had little prior experience with the long jump, we might expect that they would get better with every attempt. Second, external factors like weather conditions could influence results.

To control for these variables, we should introduce a control group that does not undergo the training program. Therefore, the whole group of 20 trainees should be divided into two subgroups: the intervention group and the control group.

When we have the data in hand, the next step is to analyze the results to determine whether the training program had a significant impact on performance. The control group (C) did not receive any special training, while the intervention group (I) underwent the program. By comparing the pre-training and post-training test results for both groups, we can identify the improvement of the trainees.

In the previous section, we conducted tests that relied on knowing the distribution type, such as binomial or Poisson. Such tests are called parametric because the hypotheses they test are formulated around the parameters of a known distribution. In this case, however, we don't really know what the distribution type is.

What we can do is calculate the improvement of each trainee and then order the improvements from highest to lowest. Then, we can forget about the concrete numbers and only consider the order.

ID	Group	Pre-training test result	Post-training test result
1	C	372 cm	416 cm
2	C	345 cm	356 cm
3	C	378 cm	395 cm
4	C	415 cm	401 cm
5	C	341 cm	345 cm
6	C	346 cm	358 cm
7	C	417 cm	409 cm
8	C	383 cm	406 cm
9	C	331 cm	330 cm
10	C	374 cm	383 cm
11	I	333 cm	349 cm
12	I	376 cm	409 cm
13	I	366 cm	393 cm
14	I	255 cm	276 cm
15	I	279 cm	325 cm
16	I	330 cm	331 cm
17	I	311 cm	343 cm
18	I	369 cm	354 cm
19	I	315 cm	329 cm
20	I	294 cm	325 cm

Table 2.2: Results of the pre-training and post-training tests

- How can we test whether the intervention group did significantly better than the control group, based on the order of improvements?

Let's make the ordered improvements table.

improvement	46	44	33	32	31	27	23	21	17	16	14	12	11	9	4	1	-1	-8	-14	-15
group	I	C	I	I	I	I	C	I	C	I	I	C	C	C	C	I	C	C	C	I

After removing the numeric values, we are left with the binary sequence

ICIIICICIICCCCICCCI.

There are $\binom{20}{10}$ possible sequences of this type, and all have the same probability of occurring. Hereafter, we'll refer to these sequences as (10,10)-sequences, where (10,10) means each sequence contains 10 I's and 10 C's. Our task is to arrange them in a manner that represents a spectrum, ranging from *the improvements of I were much, much bigger* at one extreme to *the improvements of C were much, much bigger* at the other.

So, how to order these sequences? It is always helpful to start with a smaller case. Let's try to order (2,2)-sequences.

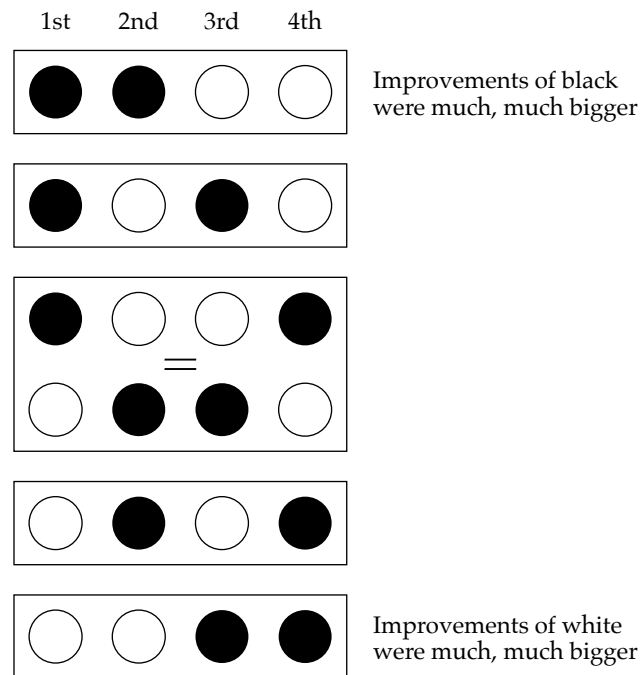


Figure 2.15: Ordering of (2,2)-sequences

That was quite easy—if we take any two of those sequences, we can show why one is strictly ‘higher’ in the order, except for 0110 and 1001, which are symmetrical, so they are the only ones which are equally probable.

If we try it with (3,3)-sequences, we may run into situations that are not so clear. For example, how do you compare 110010 and 101100? That is not exactly clear, so we should invent some function that will assign a number to a sequence, based on which we will order them.

There are two different methods that might come to mind: the *rank sum* and the *inversion count*.

The rank sum method involves multiplying each position index (rank) by the value at that position and then summing the resulting numbers. This method says that the sequences 110010 and 101100 are equal (see Figure 2.16).

The inversion count method involves iterating through all the 1’s, counting how many 0’s precede it, and then summing these counts (see Figure 2.17).

If you apply both methods to all 22 (3,3)-sequences, you’ll see that they yield identical results. You can use whichever method you find easier. Either of these statistics we will use for the statistical test. Such statistic is called a **test statistic**.

1st	2nd	3rd	4th	5th	6th	rank sum
1	2	3				6
1	2		4			7
1	2			5		8
1		3	4			8
1	2				6	9
1		3		5		9
	2	3	4			9
...						

Figure 2.16: Rank sum method

1st	2nd	3rd	4th	5th	6th	inversion count
0	0	0				0
0	0		1			1
0	0			2		2
0		1	1			2
0	0				3	3
0		1		2		3
	1	1	1			3
...						

Figure 2.17: Inversion count method

Now, we just need to calculate how many (10,10)-sequences have each given inversion count. It turns out that this is quite a complex problem involving integer partitions that could be a separate chapter—if you feel like trying to solve it, go for it! But will skip it for now.

Back to the original situation. The number of existing (10,10)-sequences for a given inversion count correspond with the number of possibilities how this test statistic can be achieved. If we order them and make a column chart, we have our test statistic distribution. Then we just color the rejection region (this is a one-sided situation, so the columns on the left that sum to 5 % probability).

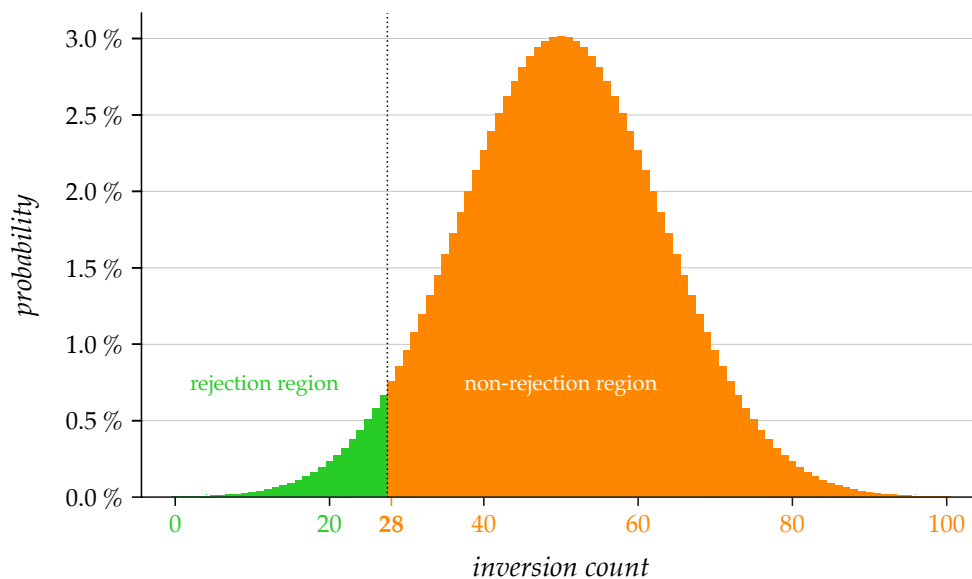


Figure 2.18: Mann–Whitney U test statistic distribution

The test statistic (inversion count) of our sample is 29. Although it is very close to the rejection region, it isn't quite there, so we do not reject the null hypothesis. That does not mean that the training program does not work—there is just not enough evidence. The non-parametric test that we just did is called the **Mann–Whitney U test**.

Since we were so close to rejecting the null hypothesis, we might wonder what would be the lowest significance level on which the null hypothesis would be rejected. That number is the ***p*-value**. To calculate it from the distribution, we calculate the graph area over the interval from zero to our test statistic (inclusive). In our case, the *p*-value is 0.0562.

2.6 The Bayesian approach

In the previous chapter, we introduced the concept of the p -value. It may be tempting to interpret the p -value as the probability that the null hypothesis is true. But that would be a mistake! If a statistical test yields a p -value of 0.33, it is obvious that this result does not demonstrate any significant effect. Thus, it would be strange to conclude that the null hypothesis has a 33% probability.

So, does the conventional significance level of 0.05 imply that 5 % of null hypothesis rejections, that is 5 % of discoveries backed by statistical testing, are incorrect?

Before answering this question, let's consider a different problem. Imagine that the local hospital offers its patients a preventive rapid screening test for celiac disease. The hospital says the test has 75% sensitivity, meaning that three out of four celiacs will be detected. And even better, the test has 90% specificity, that is 90 % of people who are not celiacs will be correctly diagnosed as not having it. That sounds like a really good test—should you go and have yourself tested?

- Suppose that 2 % of the global population are celiacs. How many of 1,000 people would be
1. correctly diagnosed as having the disease,
 2. incorrectly diagnosed as having the disease,
 3. correctly diagnosed as not having the disease,
 4. incorrectly diagnosed as not having the disease?

Let's draw a tree diagram.

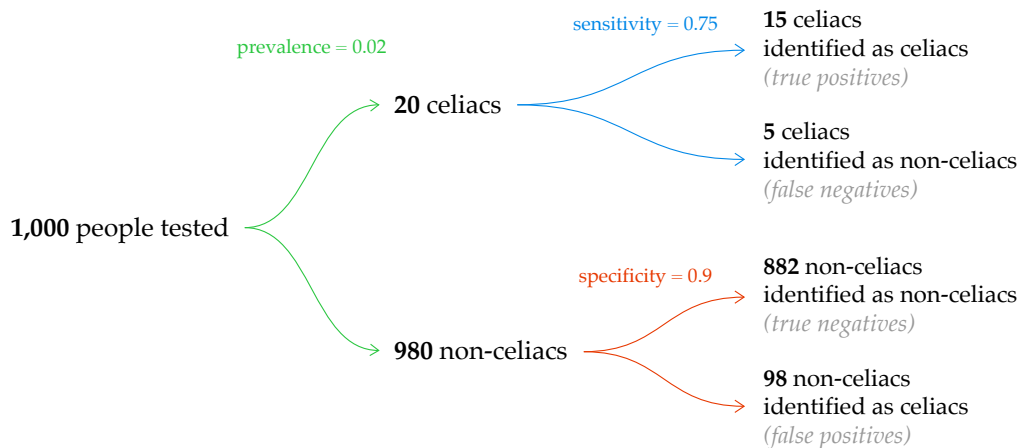


Figure 2.19: Tree diagram of diagnostic test outcomes

We first divide the sample of 1,000 people by their condition: 2 % have the disease, that is 20 individuals, and the remaining 980 do not have the disease. Then, we can subdivide these two groups by sensitivity and specificity. With a sensitivity of 75 %, the test correctly identifies 15 out of the 20 celiacs, missing 5 cases (known as the Type II error, or false negatives). The test's specificity of 90 % results in 882

correctly identified negatives among the 980 people who don't have the disease. However, 98 individuals without the disease are incorrectly identified as having it (known as the Type I error, or false positives).

As a result, the test identifies a total of 113 individuals as positive, of which only 15 truly have the disease. The false positives dominate, leading to a false discovery rate of $\frac{98}{113} \approx 87\%$. That means if you go test yourself and the test comes out positive (stating that you have celiac disease), you have a whopping 13 % chance that you actually have the disease!

How is this related to the p -value misinterpretation? Not only diagnostic tests have sensitivity and specificity, statistical tests, too! The first we call the **power of a test**, and the second we already know—that is the significance level.

The power of a test is the probability that the statistical test correctly rejects the null hypothesis. The significance level is the probability that the test incorrectly rejects the null hypothesis, so $1 - \alpha$ corresponds to specificity.

Let's estimate that only 10 % of scientific hypotheses are true. A standard power of a test is perhaps 0.8 and the conventional significance level is 0.05.

Suppose that we conduct 1000 tests. 900 will be cases where the effect does not exist, so the null hypothesis is true. Of these 900, 5 % are false positives, that is 45. So, we have 45 false positives and 855 true negatives.

The remaining 100 cases are the ones where there is a real effect and the null hypothesis is false. There, the effect is correctly detected in 80 % of the cases, 20 % tests will fail to detect the effect. So, we have 80 true positives and 20 false negatives.

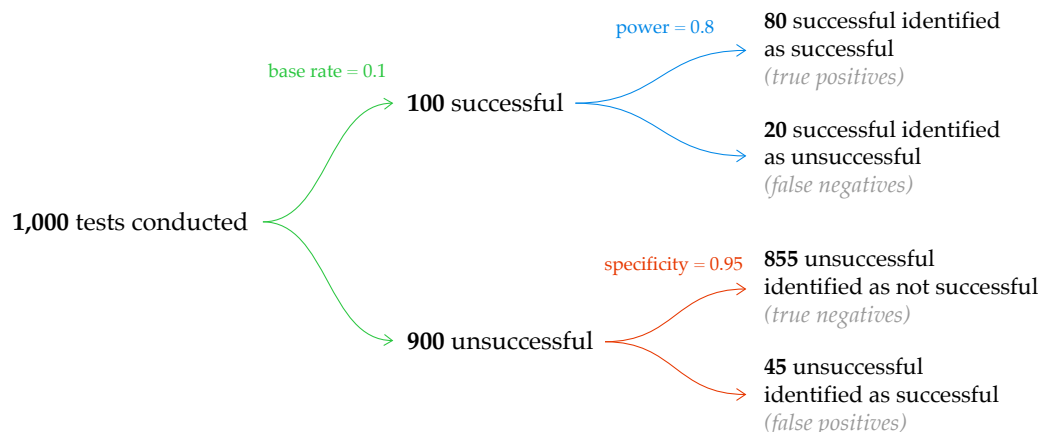


Figure 2.20: Tree diagram of tested scientific hypotheses

The number of positive tests is therefore $80 + 45 = 125$. That means you have a $\frac{45}{125} \approx 36\%$ chance of reporting a discovery when there is no real effect!

- What is the key variable here? What part of this did we previously ignore?

It's intuitive that low sensitivity or specificity affects the reliability of a test. However, the influence of prevalence is often surprising to many. It's a subtle yet crucial variable that we should always take into consideration in similar situations—this is the key to Bayesian statistics.

In each of the two examples, there were two events involved. In the first, it was *being a celiac* and *testing positive*; in the second, it was *hypothesis being true* and *statistic test demonstrating an effect*. Generally, we can call one of the events a *hypothesis* and the other one an *evidence*.

In terms of conditional probabilities, Bayes' theorem is stated as the equation

$$P(H|E) = \frac{P(E|H) \cdot P(H)}{P(E)},$$

where $P(H)$ and $P(E)$ are probabilities of events H (hypothesis is true) and E (we observe evidence, e.g. a statistic test demonstrates an effect); $P(H|E)$ is the probability of that the hypothesis is true given that we observe evidence, and $P(E|H)$ is the probability of observing evidence given that the hypothesis is true.

For the purpose of simplifying the usage of the Bayes' theorem, let's introduce another formulation of probability. Probability of an event can be expressed not only as a number between 0 and 1, but also as the ratio of the probability that the event happens to the probability that it does not happen. This alternative way of expressing probability is called **odds**. So, the probability 0.5 corresponds to odds of 1:1, the probability of 0.01 corresponds to odds of 1:99, and so on.

We can express odds of A as $O(A) = \frac{P(A)}{P(\neg A)}$, where $\neg A$ is the negation of A .

- Reformulate the Bayes' theorem for $O(H|E)$ instead of $P(H|E)$.

We divide each side of a Bayes' equation by another Bayes' equation with $\neg H$ instead of H

$$O(H|E) = \frac{P(H|E)}{P(\neg H|E)} = \frac{\frac{P(E|H) \cdot P(H)}{P(E)}}{\frac{P(E|\neg H) \cdot P(\neg H)}{P(E)}}$$

and simplify to

$$O(H|E) = O(H) \frac{P(E|H)}{P(E|\neg H)}.$$

This is known as the **Bayes rule**. The $O(H)$ is called **prior probability**—the initial probability before we observe evidence. $O(H|E)$ is **posterior probability**—the probability after observing evidence (this is what we want to know). And finally, $\frac{P(E|H)}{P(E|\neg H)}$ is the **Bayes factor**. The Bayes factor is the ratio of the probability of observing evidence given a hypothesis is true to the probability of observing evidence given a hypothesis is not true. It is the power of the evidence, quantified by how many times more likely the evidence is under the hypothesis than under its negation.

The great thing that the Bayes rule does is it formulates a rigorous way how we can update our beliefs upon observing a new evidence. When we see a new piece of evidence, we ask ourselves: *how many times more likely am I seeing the evidence when the hypothesis is true than when it is not?* If the answer is 3, then the odds of 1:1 (50% probability) become 3:1 (75% probability), and the odds of 1:99 (1% probability) become $3:99 = 1:33$ (~3% probability). Note that this is trivial to calculate with odds, but not so much with probability.

We can even use the rule repeatedly—the posterior probability just becomes the new prior probability. So, suppose that you have tested positive in the celiac disease diagnostic test. You have done the calculation and you know that there is only 13 % probability that you have the disease. However, in the last year, you are experiencing something that might be a symptom of the disease—stomach pains. That is another piece of evidence that you can add to your model.

So, you should ask: *how many times more likely am I having these pains given that I am a celiac than given that I am not?* This is a tough question—there are many other diseases that could cause stomach pain, and the connection to celiac disease specifically isn't definitive. However, let's assume that you did a quick online research which led you to the belief that stomach pains are about five times more likely if you have celiac disease than if you don't.

To apply the Bayes rule, we multiply the Bayes factor of 5 by the prior odds of having celiac disease. Odds of 13 % corresponds to ~0.15:1. Therefore, your new odds become 0.75:1, that is you believe there is now 43% probability that you are a celiac.

The disadvantage of the Bayesian approach is that in most cases, it is not clear what the prior should be. Think of any scientific hypothesis that you read about in the newspaper. Can you give it a probability before you know anything about it? If not, should the default be 50 %? Estimating these probabilities is usually subjective, which is the most common critique of Bayesian analysis. On the other hand, the Bayes rule is a perfect tool to improve your probabilistic model of the world, which is why it has become not just a foundational principle, but also an iconic symbol for rationalists.

The 20th century witnessed a vigorous debate between two camps of statisticians: frequentists, who emphasized the methods of statistical testing we got acquainted with in the previous sections, and Bayesians, who proposed a wider usage of Bayesian inference—various applications of the theorem we saw in this chapter. This debate is an interesting intersection of mathematics and the epistemology, and its history is well summarized in [11]. If you enjoyed this chapter, both [11] and [2] are recommended for further study.

Part II

Theoretical Commentary

Chapter 3

Didactic commentary

In this chapter, we aim to explain the motivation behind the didactic approach taken when creating the textbook. This includes the structure, the language, the choices of progression, and other decisions made in the process of writing.

3.1 Mathematical rigor and good explanations are not the same thing

The primary goal of a mathematics textbook is to effectively teach mathematical concepts and methods. It is a tool for fostering a thorough understanding of the topic. This is a different goal than the one prevalent in mathematical practice—that is to provide proofs of theorems, which requires a purely deductive approach typical of mathematical rigor. It should be noted that producing the most rigorous text does not always make it easy to understand, let alone readable.

One example of a piece of information vital for understanding, however not a necessary part of a proof, is a high-level view of the topic. When entering a new mathematical topic as a reader, it is useful to know why the topic is relevant, and have an overview of the broad strategy for exploring it.

The task of the author of a mathematical textbook is to balance the degree of the rigor and the context-adding commentary. Both depend primarily on the learner's starting level. Since this thesis focuses on high-school students, we should not expect the learner to have experience with reading mathematical texts structured in a definition-theorem-proof format (hereafter referred to as DTP). Therefore, the textbook emphasizes the context-adding commentary, analogies, examples and visual aids, and only sporadically features definitions, theorems and proofs, to introduce the learner to DTP without intimidation. It is the entry point for a more thorough study using more rigorous sources, or a complement to them, certainly not a replacement for them.

Advancing to a professional level of mathematics, we can notice that some texts are

written in a more reader-friendly fashion than others. The art of explaining maths is not constrained to textbooks for high-school or university students only. The Fields Medal laureate William Thurston argues that even research papers should be written with the effort of clear explanation, because they will be read by future doctorate students, seeking to understand the matter in order to subsequently build on it. In his essay *On Proof and Progress in Mathematics* [19], he shares his personal experience from the 1970s, when he rapidly proved many theorems in a certain field of geometric topology, and published them without adequately explaining the proofs well in the papers. He writes that he then perceived an ‘exodus’ from the field, not because there was no more work, but because he created a barrier by not providing good enough scaffolding for the next to come.

3.2 Effective explanations

Christopher Olah, the co-founder of *Distill*—a digital scientific journal devoted to clear explanations, believes that there are two parts to a good explanation: first, having a really clear way of thinking about the topic, and second, executing the explanation well. He also mentions that people, who are ‘extremely knowledgeable about a topic’, don’t always produce good explanations, because they falsely believe that more detail and rigor will make the explanation clearer. [15]

Another reason for this paradoxical phenomenon might be that the teacher has the advantage of already understanding the topic. That entails remembering the defined terms, their definitions, and all logically preceding theorems. That allows her to connect all of it much quicker, while the student gets lost when the amount of new knowledge surpasses some critical level. Because of this, it is beneficial to think twice about every introduction of a new term.

3.2.1 Terminology

Introduction of terms serves to simplify the subsequent statements. If we say *a word* instead of *a finite sequence of symbols*, we have shortened the statement by three words each time we use it. The downside is that to understand the statements, each term must be connected to its definition in the memory of the learner. Too much terminology may overload the learner’s working memory. To make terms memorable, they should target the learner’s intuition, and evoke a picture that serves as a good representation of the term.

3.2.2 Simplifying foundations

Every mathematical theory today builds on some foundations relating it to an axiomatic theory. For example, in statistics, the concepts of Borel σ -algebra and the Kolmogorov axioms are the cornerstones to build upon. These foundations are essential for proving theorems from axioms but are difficult to grasp for high-school-level learners due to their abstractness.

There are two types of such mathematical constructs:

1. constructs that can be left out for the purpose of intuition-based explaining (as opposed to theoretical-foundations-based explaining),
2. constructs that are vital for understanding the theory, but their formal definition is too abstract for a high-school student.

The first type constructs are completely omitted in the textbook (for example the Borel σ -algebra). The second type constructs are either explained intuitively, or not explained generally, but only in a narrowed sense. For example, in an algebra course, we would learn about congruence classes and operations with them. Then, a ring would be defined as a set equipped with two operations, satisfying certain properties; and a field as a ring satisfying stronger properties. Then, it would be proved that a set of congruence classes modulo m is a ring for all m , and a field if and only if m is prime. In our textbook, we start with a ring of integers modulo m , which is the simplest type of finite ring—thanks to that, the reader has a concrete example of the structure from the start. The formal definition of a ring can be omitted at this moment, because the more important structure is a field, which is introduced through its most important property, which is the existence of multiplicative inverses. We try to work with the bare minimum, and not clog the reader's working memory.

3.2.3 Analogies

The abstractness of mathematics is a challenge. One way to deal with it is by leveraging analogies with physical objects, natural laws or social relations that we know from the real world. This method is frequently used by teachers—for example, a function is often introduced to elementary-school students as a vending machine. Analogies can serve a dual purpose—they make reasoning about them easier, just because they give us a relatable thing to work with in our head, and if they are precise analogies, they also explain the structure of the mathematical object.

Analogies as enablers of reasoning

How merely swapping content of a statement in a form of implication enables correct reasoning was shown by an experiment conducted by the psychologists Griggs and Cox in 1982 [8]. Griggs and Cox replicated an earlier experiment known as the Wason selection task. In this task, the subjects of the study are asked about the minimal information that must be inquired to verify that a statement 'if A , then B ' is correct. In one variant of the experiment, the statement is *If there is a 'D' on one side of a card, then there is a '3' on the other side*. and the subject is presented with four cards on a table, with 'D', 'F', '3' and '7' on the visible sides. It is also known that a card always has a number on one side and a letter on the other side. In another version, the statement is *If a person is drinking beer, they must be over 20*. and the subject is presented with a situation where there are four people in a pub,

one 18 years old, one 50 years old, one drinking soda and one drinking beer. The correct answer is to check the flip side of 'D' and '7' in the first variant, and to check the drink of the 18 year-old person and the age of the beer drinker in the second. The abstract variant of the task was answered correctly by less than 25 % subjects, while the 'social contract' variant by 75 %.

This suggests that changing the content of some mathematical problems may make them easier to solve. In the textbook, we attempted to utilize a different social contract to explain the Hamming code—the relation between a guard and a prisoner as an analogy for the relation between message symbols and check symbols.

Analogies as representations of structure

The second purpose of analogy is to illustrate the structure of a mathematical object.

One example of such analogy used in the textbook is how the homomorphism between the set of all integers and a given ring of integers modulo m was explained by winding a thread onto a spool. This analogy is great in that it represents well the mapping—we can instantly see that it is surjective, that there are m elements in the ring, each representing infinitely many integers, that each integer is included in exactly one element of the ring, etc. However, it is not that great in representing that the mapping is in fact a homomorphism—that is better verified algebraically.

That an analogy does not show all properties of the underlying structure is not fatal—another analogy might serve better. However, some analogies have limitations in their accuracy and may demonstrate some properties falsely. For example, the spatial analogy for Hamming distance is limited because Hamming distance and Euclidean distance just work differently. For example, we might think that there can be at most 4 words for which each pair has the same distance (because that is true for points in a 3D space with Euclidean distance), while in fact there can be as many as there are symbols in the alphabet.

A special type of analogy for representing structure is a physical analogy. For example, a mean of a distribution is essentially its centre of mass. So, if we cut the graph of the distribution out of paper, and find a point on the x -axis on which the cut-out balances on a skewer, that x value is the mean.

3.3 Problem-first approach

As touched on in the preface, one of the key features of the textbook is the employment of a problem-first approach, as an opposite of a theory-first approach. Starting with a problem not only creates curiosity about its possible solution, but also allows the learner to derive the theory from the first principles. This process is difficult as it basically imitates the path of how these mathematical theories were invented. The mathematics popularizer Grant Sanderson says in one of his videos

that mathematicians do not start with definitions, but with ‘chewing on specific problems, then generalizing those problems, then coming up with constructs that might be helpful in those general cases, and only then they write down a new definition.’ [17] He calls this process the *discovery progression*, as opposed to the deductive *textbook progression*. In pedagogical literature, the form of learning that employs problem-first approach or discovery progression is called inquiry-based learning [5].

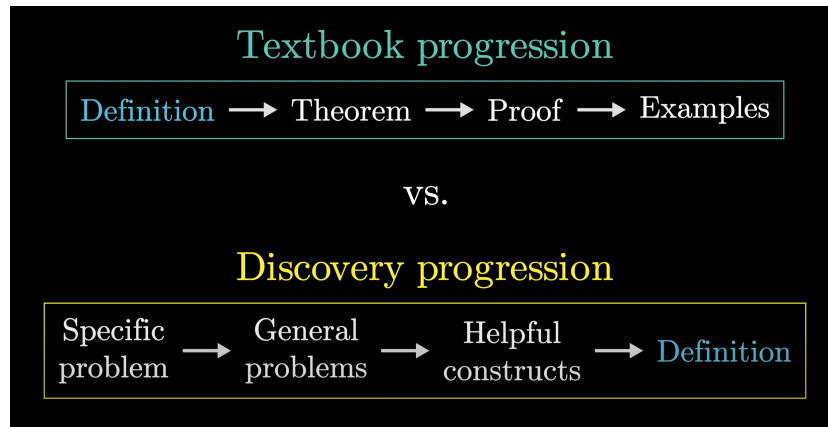


Figure 3.1: Textbook progression and discovery progression (source: [17])

In Chapter 1, the progression toward the Hamming code was inspired by Richard Hamming’s actual discovery process, which is documented in his book *The Art of Doing Science and Engineering* [10].

3.4 Narrative style

Each chapter of the textbook starts with a formulation of the problem. The author then acts as a teacher, talking to the reader and asking him the questions. The questions are graphically separated from the rest of the text, set in italics and marked by a black triangle to the left of them. The reader is instructed to pause after reading the question and try to answer it. Sometimes, the question is just a conceptual one which can be thought about in one’s head, sometimes it is a problem that needs to be sketched out. The guidance of the text always keeps the reader on the right track, so the discovery process is not full of dead-ends as it would be in reality.

3.5 A smoother transition to high mathematics

The textbook attempts to bridge the gap between high-school mathematics and university-level mathematics. This gap is characterized by a sudden change of both the form and the content. High-school mathematics is mostly concerned with algorithms for solving certain problems, such as simplification of expressions,

solving systems of equations or combinatorial problems. The problems are solved in concrete forms. Conversely, university mathematics mostly takes the form of DTP. It operates on a more abstract level, and the mathematical objects are more difficult to visualize.

This sudden switch to 'high mathematics' is hard for some students, so it is best if they can come in contact with it before university. The textbook tries to do that by slowly building up terminology, and including formal definitions in grey boxes. The first chapter also contains some easy proofs.

Conclusion

In this thesis, we have presented two chapters of an extracurricular applied mathematics textbook for high-school students, inspired by the course *Street-fighting Mathematics* by Sanjoy Mahajan. A ‘problem-first’ approach was used to introduce the reader to two areas of mathematics usually not covered at high school: error-correcting codes and inferential statistics. The designed curriculum not only equips students with knowledge in coding theory and statistical inference but also trains them in problem solving, and prepares them for entering more advanced studies in mathematics and related disciplines.

Looking forward, this work lays a foundation for a potential expansion into other areas of mathematics. It is hoped that this thesis will serve as a valuable resource for both students and teachers.

Bibliography

- [1] ADÁMEK, Jiří. *Kódování. Matematika pro vysoké školy technické*. Praha: Státní nakladatelství technické literatury, 1989.
- [2] BRUCE, Peter C. *Introductory Statistics and Analysis: A Resampling Perspective*. Hoboken: Wiley, 2015. ISBN 978-1-118-88135-4.
- [3] COLQUHOUN, David. An investigation of the false discovery rate and the misinterpretation of p-values. *Royal Society open science* [online]. London: The Royal Society Publishing, 2014, vol. 1, no. 3. Available at: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4448847/>
- [4] COSMIDES, Leda and John TOOBY. Cognitive Adaptations for Social Exchange. In: BARKOW, Jerome H., Leda COSMIDES and John TOOBY. *The Adapted Mind: Evolutionary Psychology and the Generation of Culture*. New York: Oxford University Press, 1992, pp. 164–228. ISBN 978-01-951-0107-2.
- [5] DOSTÁL, Jiří. *Inquiry-based instruction. Concept, essence, importance and contribution*. Univerzita Palackého v Olomouci, 2015. ISBN 978-80-244-4507-6.
- [6] ELSNIC, Miroslav. Tragické čtvrtletí. Na českých silnicích zemřel rekordní počet cyklistů. In: *Deník.cz* [online]. Prague: VLTAVA LABE MEDIA, 2023-04-30 [cited 2024-05-02]. Available at: <https://www.denik.cz/nehody/rekord-mrtvi-cykliste-policie-2023.html>
- [7] FRANK, Michael C., Daniel L. EVERETT, Evelina FEDORENKO and Edward GIBSON. Number as a cognitive technology: evidence from Pirahã language and cognition. *Cognition*. Elsevier, 2008, vol. 108, no. 3, pp. 819–824. Available at: <https://www.sciencedirect.com/science/article/pii/S0010027708001042>
- [8] GRIGGS, Richard and James COX. The elusive thematic-materials effect in Wason's selection task. *The British Journal of Psychology*. Hoboken: Wiley, 1982, vol. 73, no. 3, pp. 407–420. ISSN 0007-1269.
- [9] HÁJEK, Jaroslav. *A course in nonparametric statistics*. San Francisco: Holden-Day, 1969.
- [10] HAMMING, Richard W. *The Art of Doing Science and Engineering*. Amsterdam: Gordon and Breach Science Publishers, 1997. ISBN 90-5699-500-6.

- [11] HEBÁK, Petr. *Statistické myšlení a nástroje analýzy dat*. Praha: Informatorium, 2013. ISBN 978-80-7333-105-4.
- [12] HILL, Raymond. *A First Course in Coding Theory*. New York: Oxford University Press, 1986. ISBN 0-19-853803-0.
- [13] MAHAJAN, Sanjoy. *Street-Fighting Mathematics: The Art of Educated Guessing and Opportunistic Problem Solving*. Cambridge: MIT Press, 2010. ISBN 978-02-6251-429-3.
- [14] MATOUŠEK, Jiří. *Thirty-three Miniatures: Mathematical and Algorithmic Applications of Linear Algebra*. Student Mathematical Library. Providence, R.I.: American Mathematical Society, 2010. ISBN 978-0-8218-4977-4.
- [15] OLAH, Christopher and Robert WIBLIN. *108 – Chris Olah on working at top AI labs without an undergrad degree*. 80,000 Hours Podcast. Available at: <https://80000hours.org/podcast/episodes/chris-olah-unconventional-career-path/>
- [16] REED, Irving S. and Gustave SOLOMON. Polynomial Codes Over Certain Finite Fields. *Journal of the Society for Industrial and Applied Mathematics*. 1960, vol. 8, no. 2, pp. 300–304. ISSN 0368-4245.
- [17] SANDERSON, Grant. *How (and why) to raise e to the power of a matrix — DE6* [video]. YouTube: 3Blue1Brown [online]. Published 2021-04-01 [cited 2024-04-18]. Available at: <https://youtu.be/O85OWBJ2ayo>
- [18] SAVOV, Ivan. *No Bullshit Guide to Linear Algebra*. Second edition. Montréal: Minireference Co., 2018. ISBN 978-0-9920010-2-5.
- [19] THURSTON, William P. On proof and progress in mathematics. *Bulletin of the American Mathematical Society*. 1994, vol. 30, no. 2, pp. 161–177. ISSN 0273-0979. Available at: <https://arxiv.org/pdf/math/9404236>
- [20] WASON, Peter C. Reasoning about a rule. *The Quarterly Journal of Experimental Psychology*. 1968, vol. 20, no. 3, pp. 273–281. ISSN 1747-0218. Available at: <https://doi.org/10.1080/14640746808400161>
- [21] Zen MathMind. *How to get better at learning mathematics - Intuition versus Rote Learning* [video]. YouTube: Zen MathMind [online]. Published 2021-01-02 [cited 2024-04-18]. Available at: <https://youtu.be/GDPQ3YDHLCQ>

