



CONTENTS

1	Inverse Trigonometric Functions	3
1.1	Restricting the Domain	3
1.2	Finding Angles	5
1.2.1	Direction of a Vector	6
2	Laws, Paradoxes, Thought Experiments, and Illusions	9
2.1	The Barber	9
2.2	Mathematical Paradoxes	10
2.2.1	Russell's Paradox	10
2.2.2	Simpson's Paradox	10
2.2.3	Monty Hall Problem	12
2.2.4	Birthday Problem	13
2.2.5	Zeno's Paradoxes	15
2.3	Physics and Time Travel Paradoxes	16
2.3.1	Schrödinger's Cat	16
2.3.2	Time Travel Paradoxes	18
2.4	Architecture and Geometry Paradoxes	18
2.4.1	Penrose Steps	19
2.4.2	Möbius Strip	19
2.4.3	Coastline paradox	20
2.5	Computer Science Paradoxes and Laws	20
2.5.1	Moravec's Paradox	20
2.5.2	Neural Scaling Law	21

3	Monte Carlo Simulation	23
3.1	Random Numbers in Python	23
3.2	Estimating π with Darts	24
3.2.1	How Accurate Is a Simulation?	25
3.3	Monty Hall, the Whole Game	25
3.3.1	Monty Fall	26
3.4	The Birthday Problem	27
3.4.1	A Question Without a Formula	29
4	Numerical Methods of Describing Data	31
4.1	Why Numerical Summaries Matter	31
4.1.1	Summation Notation	32
4.2	Measures of Center	33
4.2.1	The Mean	33
4.2.2	The Median	34
4.2.3	Choosing Between Them	36
4.3	Measures of Spread	37
4.3.1	Range and IQR	37
4.3.2	Variance and Standard Deviation	38
4.3.3	Interpreting Spread	41
5	Shape, Position, and Outliers	43
5.1	Connecting Shape to Numerical Measures	43
5.1.1	Symmetric Data: Mean Approximately Equals Median	43
5.1.2	Skewed Data: Mean Pulled Toward the Tail	44
5.1.3	Effects of Outliers	44
5.1.4	Changing Units on Summary Measures	45
5.2	Z-scores	47
5.3	Percentiles and Quartiles	49
5.3.1	Computing with Python	52
6	Introduction to Graphs	55
6.1	Introduction	55
6.2	Finding Good Paths	57
6.3	Graphs in Python	59
A	Answers to Exercises	65
	Index	73

Inverse Trigonometric Functions

A wheelchair ramp must be gentle: building codes allow it to rise at most 1 meter for every 12 meters it runs. What angle is that? You know $\tan \theta = \frac{1}{12}$. To get from the ratio back to the angle, you need to run tangent backward. That is what inverse trigonometric functions do.

In the inverses chapter, an inverse was a machine that turns y back into x . Let us start with $\sin$ and its inverse, $\sin^{-1}$, also called $\arcsin$. (Careful: $\sin^{-1} x$ means the inverse function, not $\frac{1}{\sin x}$.)

There is a problem. In figure 1.1, the dashed line $y = \frac{2}{3}$ crosses the sine wave again and again. So $\sin x$ is not one-to-one: many angles have a sine of $\frac{2}{3}$. Reflect the whole sine graph across $y = x$, and the result fails the vertical line test (figure 1.2).

1.1 Restricting the Domain

The fix, just as with x^2 in the inverses chapter, is to use only part of the sine graph. We keep the piece from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$. On that piece, sine climbs steadily from -1 to 1 , hitting

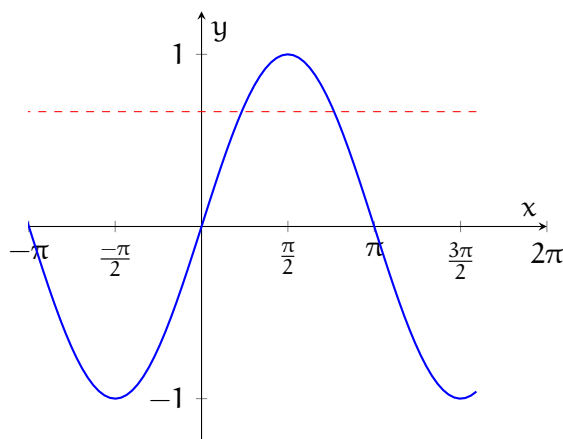


Figure 1.1: The horizontal line $y = \frac{2}{3}$ crosses $y = \sin x$ more than once

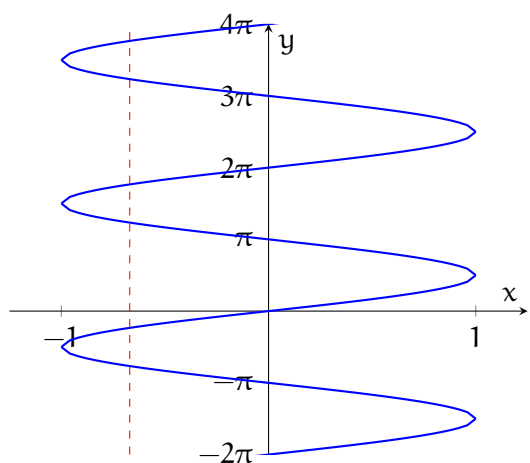
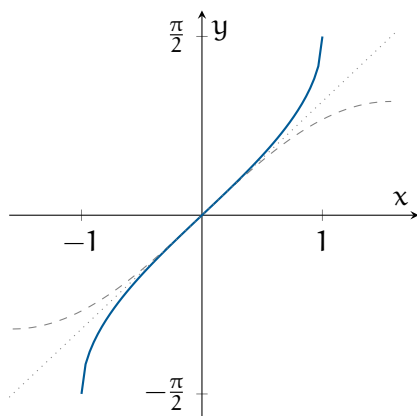


Figure 1.2: The inverse of an unrestricted sin function fails the vertical line test

every value exactly once. So:

Arcsine

$\arcsin x$ is the angle between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$ whose sine is x . Its domain is $-1 \leq x \leq 1$.



The restricted sine (dashed) and its inverse, $\arcsin x$ (solid). They are reflections across $y = x$ (dotted).

Cosine needs a different piece, because between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$ cosine goes up and then down. Cosine falls steadily from 1 to -1 between 0 and π , so we use that piece. Tangent is steady between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$, and it takes on every real number there.

Function	Input x	Output angle
$\arcsin x$	$-1 \leq x \leq 1$	$-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}$
$\arccos x$	$-1 \leq x \leq 1$	$0 \leq \theta \leq \pi$
$\arctan x$	any real number	$-\frac{\pi}{2} < \theta < \frac{\pi}{2}$

Worked example. Find $\arcsin \frac{1}{2}$, $\arccos(-\frac{1}{2})$, and $\arctan(-1)$.

Solution.

- Which angle between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$ has sine $\frac{1}{2}$? From the unit circle, $\frac{\pi}{6}$. So $\arcsin \frac{1}{2} = \frac{\pi}{6}$.
- Which angle between 0 and π has cosine $-\frac{1}{2}$? It is $\frac{2\pi}{3}$.
- Which angle between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$ has tangent -1 ? It is $-\frac{\pi}{4}$.

Watch out for one trap. Since $\sin \frac{5\pi}{6} = \frac{1}{2}$, you might expect $\arcsin(\sin \frac{5\pi}{6}) = \frac{5\pi}{6}$. But $\arcsin$ only answers with angles between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$, so the answer is $\frac{\pi}{6}$.

Exercise 1 Exact values

Find each value exactly, in radians.

Working Space

- $\arcsin \frac{\sqrt{2}}{2}$
- $\arccos 0$
- $\arctan \sqrt{3}$
- $\arcsin(-1)$
- $\arccos(\cos(-\frac{\pi}{3}))$

Answer on Page 65

1.2 Finding Angles

Inverse trig functions shine whenever you know two sides of a right triangle and want an angle.

Worked example. Back to the ramp. It rises 1 m for every 12 m of run. What angle does it make with the ground?

Solution. The rise is opposite the angle and the run is adjacent, so $\tan \theta = \frac{1}{12}$ and

$$\theta = \arctan \frac{1}{12} \approx 0.0831 \text{ radians} \approx 4.76^\circ$$

That is a gentle slope: less than 5 degrees.

On a calculator, the inverse functions are usually above the trig keys, labeled $\sin^{-1}$, $\cos^{-1}$, and $\tan^{-1}$. Check whether you are in degrees or radians mode before you trust the answer.

1.2.1 Direction of a Vector

What direction does the vector $[-3, 4]$ point? You might try $\arctan \frac{4}{-3} \approx -53.1^\circ$. But that points down and to the right, and $[-3, 4]$ points up and to the left! Arctan only answers between -90° and 90° , so it cannot tell $[-3, 4]$ from $[3, -4]$: both have the same ratio.

Programmers solve this with a two-argument version, `atan2`, which looks at the signs of both components:

```
1 import math
2
3 print(math.degrees(math.atan(4 / -3))) # -53.13... (wrong quadrant)
4 print(math.degrees(math.atan2(4, -3))) # 126.869... (correct)
```

Notice that `atan2` takes `y` first, then `x`. Use it whenever you need the direction of a vector.

Exercise 2 Ladder angle

A 6 m ladder leans against a wall with its top 5.5 m above the ground. What angle does the ladder make with the ground? Safety guides recommend about 75° . Is this ladder too steep or too shallow?

Working Space

Answer on Page 65

Exercise 3 A steep road

A road climbs 1.5 m over a horizontal

Working Space

distance of 40 m. What is its angle of incline, in degrees?

Answer on Page 65

Laws, Paradoxes, Thought Experiments, and Illusions

How many people must be in a room before two of them probably share a birthday? Most people guess around 180. The real answer is 23. Your intuition is a wonderful tool, but it can be fooled, and the places where it gets fooled are some of the most interesting places in math and science.

A *paradox* is a statement or situation that seems to contradict itself or defies intuition. Some paradoxes are real contradictions, which tell us that some assumption must be wrong. Others only *feel* impossible, and studying them sharpens your intuition.

Some of the puzzles in this chapter are *thought experiments*: imaginary scenarios that test an idea when a real experiment would be hard or impossible. You have already used them. Newton's cannon on a mountaintop is a thought experiment, and so is every frictionless surface in your physics problems.

Paradoxes are also closely tied to *proof by contradiction*, from the proofs chapter: assume a statement, show that it leads to a contradiction, and conclude that the assumption was false. Let us start with a contradiction you can picture.

2.1 The Barber

The Barber's Paradox is a self-referential paradox that arises in set theory and logic.

The paradox is often stated as follows

Theorem 1. *In a town, there is a barber who shaves all and only those men who do not shave themselves. The question is: Does the barber shave himself?*

Well there are two cases to consider:

- If the barber shaves himself, then according to the definition, he is now someone who *shaves himself*. The rule says the barber only shaves people who do not shave themselves. This is a contradiction because the barber cannot both shave himself and not shave himself at the same time.
- If the barber does not shave himself, then according to the definition, he is someone

who does not shave himself. The rule says the barber shaves all those who do not shave themselves. This means the barber must shave himself, which again leads to a contradiction because he cannot both shave himself and not shave himself at the same time.

No matter which case you try, you reach a contradiction. That is a proof by contradiction: assuming such a barber exists leads to nonsense, so no such barber can exist. Paradoxes like this one often come from statements that refer to themselves.

2.2 Mathematical Paradoxes

2.2.1 Russell's Paradox

Russell's Paradox is a paradox within set theory. It arises from the statement that

Theorem 2. *Consider the set of all sets that do not contain themselves. Does this set contain itself?*

Most sets do not contain themselves. The set of all apples is not an apple. But some sets seem to: the set of all ideas is itself an idea. For Russell's statement, let us create a master set R that contains only the sets that do not contain themselves. There are two cases:

- If R is not a member of itself, then its definition entails that it is a member of itself. $\rightarrow\leftarrow$
- If R is a member of itself, then it is not a member of itself, since it is the set of all sets that are not members of themselves. $\rightarrow\leftarrow$

Using symbols from our sets chapter,

$$\text{Let } R = \{x \mid x \notin x\}. \text{ Then } R \in R \iff R \notin R.$$

This is the barber's paradox in set form. Bertrand Russell found it in 1901, and it shook the foundations of mathematics. Mathematicians fixed it by making the rules for building sets stricter, so that "the set of all sets that do not contain themselves" is simply not allowed.

2.2.2 Simpson's Paradox

Simpson's Paradox is a phenomenon in statistics where a trend or correlation that appears in several different groups of data can disappear or reverse when the groups are combined. Let us look at an example.

Imagine you are choosing between two hospitals for an operation, and you compare their survival rates. We have the following data:

Hospital	Total Patients	Survived	Survival Rate
Hospital A	1000	800	80%
Hospital B	1000	890	89%

From this, we see that the same number of patients were treated at both hospitals, but Hospital B has a higher survival rate. Hospital B looks better. However, let us look at the data when separated by those with *high-risk* surgeries versus those with *low-risk* surgeries.

Patients that had low-risk surgeries, or were in good health, are given by following data

Hospital	Patients	Survived	Survival Rate
Hospital A	100	98	98%
Hospital B	900	850	94.4%

And those with poor health and high-risk surgeries:

Hospital	Patients	Survived	Survival Rate
Hospital A	900	702	78%
Hospital B	100	40	40%

Looking at the data based on the health categorization, Hospital A is actually better in both categories. This “paradox” occurs because of an unaccounted **confounding variable**: the incoming health of the patient. Hospital A took on the harder cases (90% of their patients are high-risk). Hospital B, although it initially was thought to be the better hospital, mainly took on those with good health (90% were low-risk). You will learn how scientists guard against confounding variables in the Experimental Design chapter.

For fun, let us plot this using the pandas, matplotlib, and seaborn libraries in Python. The code is as follows:

```

1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import seaborn as sns
4
5 # aggregate the data
6 data = {
7     'Condition': ['Good', 'Good', 'Poor', 'Poor', 'Total', 'Total'],
8     'Hospital': ['A', 'B', 'A', 'B', 'A', 'B'],
9     'Survival': [0.98, 0.94, 0.78, 0.40, 0.80, 0.89]
10 }
11 df = pd.DataFrame(data)
12
13 # split into subgroups and totals for showing different correlations
14 subgroups = df[df['Condition'] != 'Total']
15 totals = df[df['Condition'] == 'Total']

```

```

16
17
18 fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10, 5), sharey=True)
19 # regression lines and plots
20 ax1.plot([-0.2, 0.8], [0.98, 0.78], 'b--', marker='o', label='Trend A')
21 ax1.plot([0.2, 1.2], [0.94, 0.40], 'y--', marker='o', label='Trend B')
22 ax1.legend()
23 ax2.plot([-0.2, 0.2], [0.80, 0.89], 'k-', marker='o', label='Overall Trend')
24 ax2.legend()
25 sns.barplot(data=subgroups, x='Condition', y='Survival', hue='Hospital', ax=ax1)
26 ax1.set_title('Subgroups (A Wins)')
27
28 sns.barplot(data=totals, x='Condition', y='Survival', hue='Hospital', ax=ax2)
29 ax2.set_title('Total (B Wins)')
30
31 plt.tight_layout()
32 plt.show()

```

This creates two bar plots whose trends point in opposite directions:

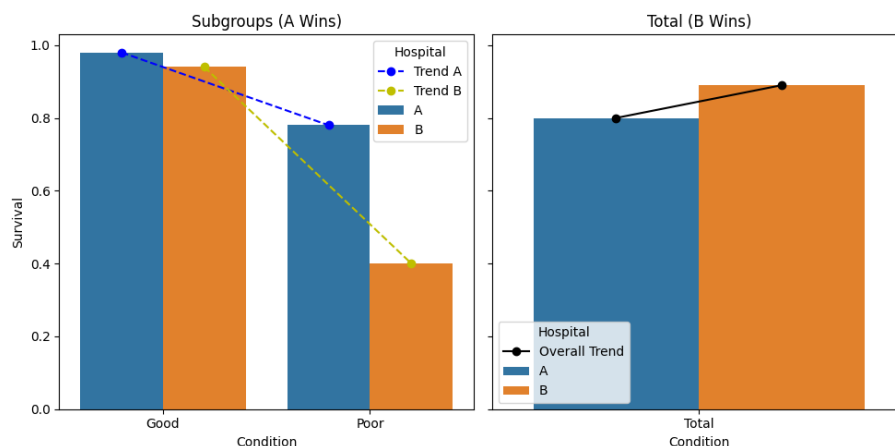


Figure 2.1: Simpson's Paradox: Subgroups vs. Total

2.2.3 Monty Hall Problem

This video introduces the [Monty Hall Problem](#).

The problem is as follows:

Theorem 3. *You are on a game show and are given the choice of three doors. Behind one door is a car; behind the others, goats. You pick a door, say No. 1, and the host, who knows what is behind the doors, opens another door, say No. 3, which has a goat. He then says to you, "Do you want to switch to door No. 2?" Is it to your advantage to switch your choice?*

Two doors are left, so it seems like a 50/50 coin flip. It is not. You should switch, and switching wins $\frac{2}{3}$ of the time.

Here is why. When you first pick, you have a $\frac{1}{3}$ chance of picking the car and a $\frac{2}{3}$ chance of picking a goat. Now think about what the host does. He knows where the car is, and he always opens a goat door that you did not pick.

- If you picked the car ($\frac{1}{3}$ of the time), switching loses.
- If you picked a goat ($\frac{2}{3}$ of the time), the host must open the *other* goat door, so the remaining door has the car. Switching wins.

The host's choice is not random, and that is what gives you information. When this puzzle appeared in a magazine in 1990, thousands of readers, including many with math PhDs, wrote in to insist the answer was wrong. It was right.

Exercise 4 Let the computer play

Write a Python program that plays the Monty Hall game 100,000 times with the always-switch strategy and prints the fraction of wins. Hint: switching wins exactly when your first pick was not the car.

Working Space

Answer on Page 65

The next chapter, Monte Carlo Simulation, plays the full game, with a host who really opens doors, and uses simulation to explore both of these puzzles further. Later, the Bayes' Theorem chapter proves the $\frac{2}{3}$ answer with conditional probability.

2.2.4 Birthday Problem

The birthday problem is a famous problem in probability and combinatorics. It goes as follows:

Theorem 4. *In a group of n people, what is the probability that at least two people share the same birthday?*

With just 23 people, the chance that two share a birthday is better than 50%. (We ignore

February 29 and assume every birthday is equally likely.) Let us prove it.

Let $P(A)$ be the probability that no two of k people share a birthday, and let $P(B) = 1 - P(A)$ be the probability that at least two do. We want to find the smallest k such that $P(B) > 0.5$, as this constitutes a majority.

If there are only two people in a room (meaning $k = 2$), the probability that there does not exist a shared birthday is $P(A) = \frac{364}{365}$, meaning the probability that there is a shared birthday is $P(B) = 1 - P(A) = \frac{1}{365}$. If there are three people in a room, the probability that there does not exist a shared birthday is $P(A) = \frac{364}{365} \cdot \frac{363}{365}$, meaning the probability that there is a shared birthday is $P(B) = 1 - P(A) = 1 - \frac{364}{365} \cdot \frac{363}{365}$.

This pattern continues, and we can generalize the probability that there does not exist a shared birthday for k people as follows:

$$P(A) = \frac{365}{365} \cdot \frac{364}{365} \cdot \frac{363}{365} \cdots \frac{365 - k + 1}{365} = \prod_{i=0}^{k-1} \frac{365 - i}{365}.$$

Therefore, the probability that there is at least one shared birthday is:

$$P(B) = 1 - P(A) = 1 - \prod_{i=0}^{k-1} \frac{365 - i}{365}.$$

We now seek the smallest k such that

$$1 - \prod_{i=0}^{k-1} \frac{365 - i}{365} > 0.5.$$

Equivalently,

$$\prod_{i=0}^{k-1} \frac{365 - i}{365} < 0.5.$$

Computing this product gives

$$P(A) \approx 0.5243 \quad \text{when } k = 22,$$

so

$$P(B) \approx 1 - 0.5243 = 0.4757 < 0.5.$$

But for $k = 23$,

$$P(A) \approx 0.4927,$$

so

$$P(B) \approx 1 - 0.4927 = 0.5073 > 0.5.$$

So 23 people is the smallest group with a better-than-even chance of a shared birthday. (Multiplying the probabilities this way works because each person's birthday is independent of the others. The Discrete Probability and Random Variables chapter explains this multiplication rule carefully.) Why so few? Because with 23 people there are $\frac{23 \times 22}{2} = 253$ different *pairs* of people, and any one of those pairs could match.

Exercise 5 Your class

A class has 30 students. What is the probability that at least two share a birthday? Use a calculator or a short Python loop.

Working Space

Answer on Page 66

2.2.5 Zeno's Paradoxes

Zeno of Elea was a Greek philosopher, around 450 BCE, famous for thought experiments that question our understanding of motion, space, and time. Zeno's paradoxes are often used to illustrate the concept of *infinity* and the nature of space and time.

The Dichotomy Paradox

His most famous is the *Dichotomy Paradox*:

That which is in locomotion must arrive at the half-way stage before it arrives at the goal.

This paradox suggests that before an object can reach its destination, it must first reach the halfway point, and before it can reach the halfway point, it must reach the quarter point, and so on, leading to an infinite number of steps that must be completed before reaching the destination. This seems to imply that motion is impossible, as it would require completing an infinite number of tasks in a finite amount of time.

So then, how is motion possible?

The resolution is that an infinite number of steps can add up to a finite total if the steps shrink fast enough. (In the calculus volumes, this idea becomes *limits* and *infinite series*.)

Each step also takes less time than the one before, so the total time is finite, too. If we consider the remaining distance to be covered as 1 unit, the first step is $\frac{1}{2}$, the second step is $\frac{1}{4}$, the third step is $\frac{1}{8}$, and so on. The total distance covered after n steps can be expressed as a *geometric series*:

$$S = \sum_{n=1}^{\infty} \frac{1}{2^n} = 1$$

For proof's sake, let

$$S = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots$$

Then, multiplying both sides by $\frac{1}{2}$ gives

$$\frac{1}{2}S = \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \cdots$$

$$S - \frac{1}{2}S = \frac{1}{2}$$

$$\frac{1}{2}S = \frac{1}{2} \implies S = 1$$

All the infinitely many steps add up to exactly the whole distance.

Exercise 6 Another infinite sum

Use the same trick to find $S = \frac{1}{3} + \frac{1}{9} + \frac{1}{27} + \cdots$.

Working Space

Answer on Page 66

2.3 Physics and Time Travel Paradoxes

Now for some physics paradoxes and thought experiments that may mess with your brain a bit.

2.3.1 Schrödinger's Cat

Erwin Schrödinger was one of the founders of quantum mechanics. In 1935, in letters with Albert Einstein, he invented a thought experiment to show how strange the theory

becomes when you apply it to everyday objects.

The thought experiment is as follows:

Schrödinger imagined placing a cat in a sealed box with a device triggered by a quantum event (such as the decay of a radioactive atom). The event has a 50% chance of happening and a 50% chance of staying dormant. If the event occurs, the cat dies; if not, it lives. However, you do not know whether the cat is alive or not until the box is opened. According to quantum mechanics, until the box is opened, the cat is in a mix of dead and alive.

Why? In quantum mechanics, a tiny particle like an electron can be in a **superposition** of states: it does not have one definite position or energy until it is measured. Instead, it is described by a *wavefunction* that holds all the possibilities at once. This is not just theory. Fire electrons one at a time at a barrier with two narrow slits, and the screen behind slowly fills with bright and dark bands, an interference pattern that only waves make. Each electron somehow goes through both slits.

The atom in the box is in a superposition of decayed and not decayed. If the cat's fate is tied to the atom, the cat must be in a superposition too. Schrödinger thought that was absurd, and that was his point: something about how quantum rules connect to the everyday world needed explaining. Physicists still debate what really happens.

There is a great TED-Ed video on Schrödinger's cat in your digital resources.

Arrow Paradox

Zeno has another paradox about motion, the *Arrow Paradox*:

If everything when it occupies an equal space is at rest at that instant of time, and if that which is in locomotion is always occupying such a space at any moment, the flying arrow is therefore motionless at that instant of time and at the next instant of time but if both instants of time are taken as the same instant or continuous instant of time then it is in motion.

In short: take a snapshot of a flying arrow. In that single instant, it sits in one spot, not moving. But the flight is made of nothing but instants. So when does it move?

The resolution is that motion is not something that happens *in* an instant; it is the change in position *across* instants. Velocity is the change in position divided by the change in time, and you need two moments to measure it. Each instant is like a single point on a line: it has no length, yet the line made of points does. In calculus, you will learn how to talk about speed *at* an instant, using limits.

2.3.2 Time Travel Paradoxes

As far as anyone knows, traveling back in time is impossible, but it shows up in movies all the time. Time travel raises logical problems because it lets events change their own causes.

The most well-known example is the **grandfather paradox**. Suppose a person travels back in time and prevents their grandfather from ever having children. This would mean the time traveler is never born, as one of the time traveler's parents are never conceived. The time traveler, then, could not have gone back in time to interfere in the first place. The sequence of events becomes logically inconsistent: the same action both must and must not occur. This is a contradiction ($\rightarrow\leftarrow$), as we talked about, and many people use this as an argument to prove that time travel cannot logically exist.

Another time travel paradox is preventing Abraham Lincoln's assassination. Imagine a time traveler goes back to 1865 and prevents Lincoln's assassination. At first glance, this seems like a straightforward intervention. However, if Lincoln survives, the historical conditions that motivated the time traveler to go back, such as learning about the assassination in the first place, no longer exist. This creates a contradiction: if Lincoln was never assassinated, why would the traveler have made the journey to stop it?

More generally, these are consistency paradoxes, where altering the past creates contradictions in the timeline. In physics, these scenarios contradict our existing knowledge, meaning either constraints exist to preserve consistency, or our understanding of time and causality is incomplete.

2.4 Architecture and Geometry Paradoxes

Some paradoxes trick your eyes instead of your logic.

2.4.1 Penrose Steps

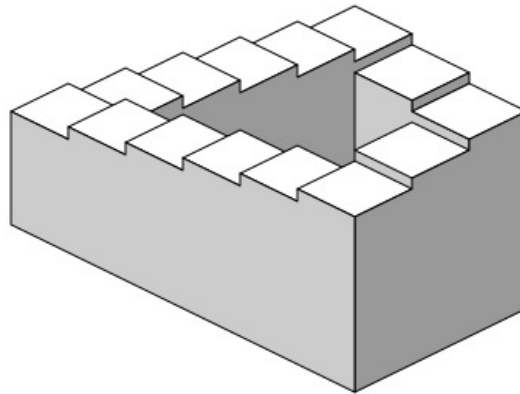


Figure 2.2: Penrose steps diagram from Wikipedia - By Sakurambo - Own work, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=794844>

The Penrose steps are an impossible figure that creates the illusion of an infinite staircase.¹ The steps appear to be continuously ascending or descending, but in reality, they form a closed loop. This paradox is achieved through clever use of perspective and geometry, creating an optical illusion that tricks the viewer's perception of space.

2.4.2 Mobius Strip

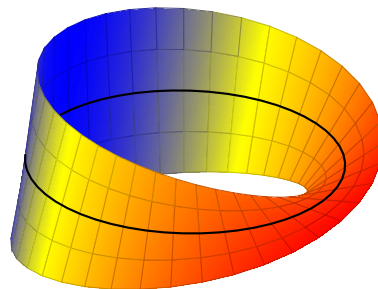


Figure 2.3: A parametric graph of a mobius strip.

A Möbius strip is a surface with only one side and only one edge. You can make one right now: take a strip of paper, give it a half twist, and tape the ends together. Draw a line down the middle without lifting your pen. You will cover *both* sides of the paper, traveling twice the length of the strip, before you return to where you started. There is no inside or outside.

¹Christopher Nolan famously reconstructed this illusion in his film *Inception* (2010). If you are interested, the scene can be found [here](#) and behind the scenes [here](#).

Now try this: cut along the line you drew. You might expect two strips. You get one long strip with two full twists. (Try cutting *that* one down the middle, too.)

If you are curious, the mobius strip is defined by the following parametric equations:

$$\mathbf{r}(u, v) = \left((1 + v \cos \frac{u}{2}) \cos u, (1 + v \cos \frac{u}{2}) \sin u, v \sin \frac{u}{2} \right)$$

where u ranges from 0 to 2π and v ranges from -1 to 1 . Do not worry about this yet; you will learn parametric equations in a later volume.

2.4.3 Coastline paradox

How long is the coast of Britain? It depends on your ruler. That is the *coastline paradox*. Measure with a 100 km ruler, and you skip over every bay. Use a 1 km ruler, and you follow the bays, but skip the coves. Use a 1 m ruler, and you trace every rock. Each smaller ruler finds more wiggles, so the measured length keeps growing.

Lewis Fry Richardson noticed this in the 1950s while studying whether the length of a border between two countries affects their chances of going to war. He found that Spain and Portugal reported very different lengths for their shared border. Neither was wrong: they had used different rulers. For a smooth curve, shrinking the ruler makes the measurement settle down to one answer. For a coastline, it never settles; the length just keeps growing.

In calculus, you will learn to find the exact length of smooth curves. A coastline is not smooth: zoom in, and it looks just as wiggly as before. Shapes like this are called *fractals*, and mathematicians measure them by how quickly their length grows as the ruler shrinks, a number called the *fractal dimension*.

2.5 Computer Science Paradoxes and Laws

2.5.1 Moravec's Paradox

Hans Moravec, a pioneer in robotics and Artificial Intelligence, wrote in 1988,

it is comparatively easy to make computers exhibit adult level performance on intelligence tests or playing checkers, and difficult or [near] impossible to give them the skills of a one-year-old when it comes to perception and mobility

This is *Moravec's paradox*: tasks that are *hard* for humans are often *easy* for machines, and tasks that are *easy* for humans are often *hard* for machines.

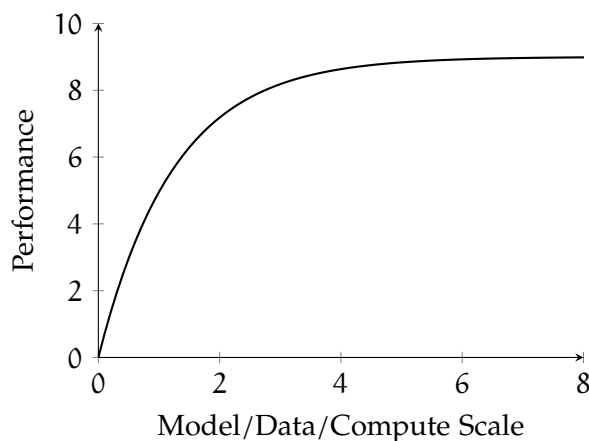
Chess is the classic example. We think of chess as a sign of great intelligence, yet in 1997 IBM's Deep Blue beat the world champion, Garry Kasparov. Multiplying huge numbers, searching millions of possibilities, and solving many kinds of equations turned out to be easy for computers, too. But for decades, no robot could fold laundry, walk across a messy room, or recognize a friend's face in a crowd, things a young child does without thinking.

Why? Moravec's explanation is evolution. Seeing, moving, and grasping have been polished by hundreds of millions of years of evolution, so they feel effortless to us; the hard work happens out of sight. Abstract reasoning like chess and algebra is new, so we do it slowly and consciously, and it feels hard. Computers have no such history, so "hard" and "easy" mean something completely different to them.

2.5.2 Neural Scaling Law

Imagine you are working for a factory that is producing pencils. With a single, static, automatic assembly line, you make pencils at a rate of 100 pencils per hour. By adding a second assembly line, you are now able to make pencils at 200 pencils per hour, double the rate it was previously. This shows that generally, production scales *linearly*.

However, not all systems scale this cleanly. Suppose instead that each additional assembly line becomes harder to integrate: workers need to coordinate more, machines interfere slightly, and space becomes constrained. Now, when you add a second line, production might increase to 180 pencils per hour instead of 200. Adding a third might bring you to 250, not 300. The total output still increases, but each new addition contributes less than the previous one. This is called *diminishing returns*.



AI models like the ones behind chatbots behave more like this second factory. Researchers track four quantities: the number of parameters in the model (N), the size of the training data (D), the amount of computing used to train it (C), and the model's error, called the *loss* (L). As N , D , or C grows, the loss shrinks, but with diminishing returns: each tenfold increase buys a smaller improvement than the last. The surprising part is how *predictable* it is. The loss follows a smooth power law,

the kind of curve you straightened with a log-log plot in the exponentials chapter. So researchers can test small models and predict how well a giant one will do before spending millions of dollars to build it. These predictable patterns are called *neural scaling laws*.

Monte Carlo Simulation

In 1946, the mathematician Stanislaw Ulam was recovering from an illness and playing solitaire to pass the time. He wondered: what is the chance that a game can be won? The combinations were far too many to count. Then he had a better idea: just deal out a hundred games and see what fraction come out. With the new electronic computers, you could deal out a million.

Ulam and his colleague John von Neumann named the idea after the famous casino in Monte Carlo, and used it to design nuclear reactors. Today, *Monte Carlo simulation* forecasts hurricanes, prices stock options, tests new medicines, lights the scenes in animated movies, and helps sports websites predict who will win the playoffs. The idea fits in one sentence:

If you cannot calculate a probability, run the experiment many times on a computer, and count.

In the Paradoxes chapter, you met two famous puzzles, the Monty Hall problem and the birthday problem. In this chapter, you will settle them by experiment, and then ask questions about them that are hard to answer any other way.

3.1 Random Numbers in Python

Python's built-in random library supplies the randomness:

- `random.random()` returns a number between 0 and 1, with every value equally likely.
- `random.randrange(n)` returns a whole number from 0 to $n - 1$.
- `random.randint(a, b)` returns a whole number from a to b , including both.
- `random.choice(items)` picks one item from a list.

Computers cannot really flip coins. These numbers come from a formula that produces a long, unpredictable-looking sequence, so they are called *pseudorandom*. That has a handy side effect: `random.seed(1)` restarts the sequence at the same place, so a program gives the same “random” results every time it runs. That is why you can check your output against this book.

Why does counting work? If an event has probability p , then in n independent trials it happens about pn times, so the fraction of successes is about p . The more trials you run,

the closer that fraction tends to get. This is the *Law of Large Numbers*. You will meet it again in the Limits chapter (Volume 17), where it helps explain what a limit is.

3.2 Estimating π with Darts

Throw darts at a 1-by-1 square, completely at random. The square contains a quarter of a circle of radius 1, with area $\frac{\pi}{4}$. Since the square has area 1, the fraction of darts landing inside the quarter circle should be about $\frac{\pi}{4}$. Multiply that fraction by 4, and you have an estimate of π .

A dart at (x, y) is inside the quarter circle when its distance from the corner is at most 1, that is, when $x^2 + y^2 \leq 1$.

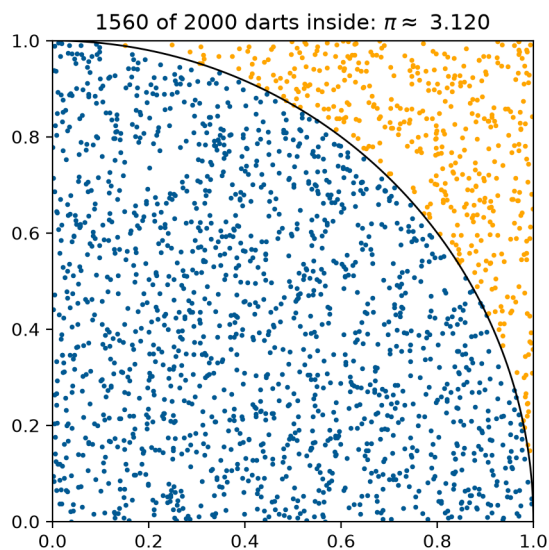


Figure 3.1: Random darts. The blue ones landed inside the quarter circle.

```

1  import random
2  import math
3
4  random.seed(1)                # same "random" darts every run
5
6  def estimate_pi(darts):
7      inside = 0
8      for _ in range(darts):
9          x = random.random()    # a random number from 0 to 1
10         y = random.random()
11         if x**2 + y**2 <= 1:    # inside the quarter circle?
12             inside += 1
13     return 4 * inside / darts
14

```

```

15 for darts in [100, 10_000, 1_000_000]:
16     estimate = estimate_pi(darts)
17     print(f"{darts:>9} darts: {estimate:.5f} (error {abs(estimate -
    ↪ math.pi):.5f})")

```

```

    100 darts: 3.20000 (error 0.05841)
   10000 darts: 3.13960 (error 0.00199)
  1000000 darts: 3.14118 (error 0.00042)

```

A million random darts get π right to about three decimal places. That is not very precise (in Volume 33, infinite series will give you π to as many digits as you like), but this method used no math beyond the Pythagorean theorem.

3.2.1 How Accurate Is a Simulation?

Simulations are never exact, so you should always ask how far off yours might be. For the dart estimate, a typical error is about $\frac{1.6}{\sqrt{n}}$ for n darts: about 0.16 for 100 darts, 0.016 for 10,000, and 0.0016 for a million. (Our runs happened to do a little better than typical.) The important part is the $\sqrt{n}$: to get one more correct digit, you need 100 times as many trials. Monte Carlo is easy to set up but slow to become precise. In Volume 30, you will learn to attach exact margins of error to estimates like these.

Exercise 7 Darts by Hand

You throw 400 darts and 312 land inside the quarter circle. What is your estimate of π ? About how many darts would you need to make the typical error 10 times smaller?

Working Space

Answer on Page 66

3.3 Monty Hall, the Whole Game

The simulation in the Paradoxes chapter used a shortcut: “switching wins exactly when your first pick was wrong.” If you doubt the argument, you should doubt the shortcut too. So here is the whole game, with a host who actually opens doors. The function play

works for any number of doors: the host opens every door except your pick and one other, and he never reveals the car.

```

1 import random
2
3 random.seed(2)
4
5 def play(switch, doors=3):
6     car = random.randrange(doors)
7     pick = random.randrange(doors)
8     # The host opens every other door except one, never revealing the car.
9     if pick == car:
10        # any other door can stay closed
11        closed = random.choice([d for d in range(doors) if d != pick])
12    else:
13        closed = car
14    if switch:
15        pick = closed
16    return pick == car
17
18 games = 100_000
19 for switch in [False, True]:
20     wins = sum(play(switch) for _ in range(games))
21     print("switch" if switch else "stay ", wins / games)
22 print("10 doors, switch:", sum(play(True, 10) for _ in range(games)) / games)

```

```

stay 0.33293
switch 0.66468
10 doors, switch: 0.89813

```

Staying wins about $\frac{1}{3}$ of the time and switching about $\frac{2}{3}$, just as the Paradoxes chapter argued. The 10-door version makes the reason obvious: you pick 1 door, the host opens 8 goat doors, and the one door he deliberately leaves closed hides the car 9 times out of 10.

3.3.1 Monty Fall

Here is a question that is easy to simulate and hard to argue about. Suppose the host does *not* know where the car is. He trips, and accidentally opens one of the two doors you did not pick, at random. It happens to show a goat. Should you switch now?

```

1 import random
2
3 random.seed(3)
4
5 kept = 0          # games where the host happened to show a goat
6 switch_wins = 0

```

```

7  for _ in range(100_000):
8      car = random.randrange(3)
9      pick = random.randrange(3)
10     # The host slips and opens one of the other two doors at random.
11     opened = random.choice([d for d in range(3) if d != pick])
12     if opened == car:
13         continue                # car revealed: this game does not count
14     kept += 1
15     other = 3 - pick - opened    # the one remaining closed door
16     if other == car:
17         switch_wins += 1
18 print(kept, switch_wins / kept)

```

66603 0.49958710568593007

In about a third of the games, the clumsy host reveals the car, and we throw those games out. In the rest, switching wins only half the time. The doors are the same, the goat is the same, but the answer changed from $\frac{2}{3}$ to $\frac{1}{2}$. The advantage of switching comes entirely from the host's *knowledge*: a host who knows where the car is and avoids it is giving you information. A host who reveals a goat by luck is not.

Exercise 8 Four Doors

In a 4-door version, you pick a door, the host (who knows where the car is) opens one goat door, and you switch to one of the two remaining closed doors at random. Write a simulation to estimate your chance of winning. Then explain the answer.

Working Space

Answer on Page 66

3.4 The Birthday Problem

In the Paradoxes chapter, a product formula showed that 23 people have a better than even chance of a shared birthday. A simulation checks it without any formula: fill a room with random birthdays and see whether any repeat. A Python set keeps only one copy of each value, so the room has a repeat exactly when the set is smaller than the list.

```

1 import random
2
3 random.seed(4)
4
5 def shared_birthday(people, trials=20_000):
6     hits = 0
7     for _ in range(trials):
8         birthdays = [random.randrange(365) for _ in range(people)]
9         if len(set(birthdays)) < people: # a set drops repeats
10             hits += 1
11     return hits / trials
12
13 def exact(people):
14     no_match = 1
15     for i in range(people):
16         no_match *= (365 - i) / 365
17     return 1 - no_match
18
19 for n in [10, 23, 30, 50]:
20     print(f"{n} people: simulated {shared_birthday(n):.3f}, exact {exact(n):.3f}")

```

```

10 people: simulated 0.121, exact 0.117
23 people: simulated 0.505, exact 0.507
30 people: simulated 0.712, exact 0.706
50 people: simulated 0.970, exact 0.970

```

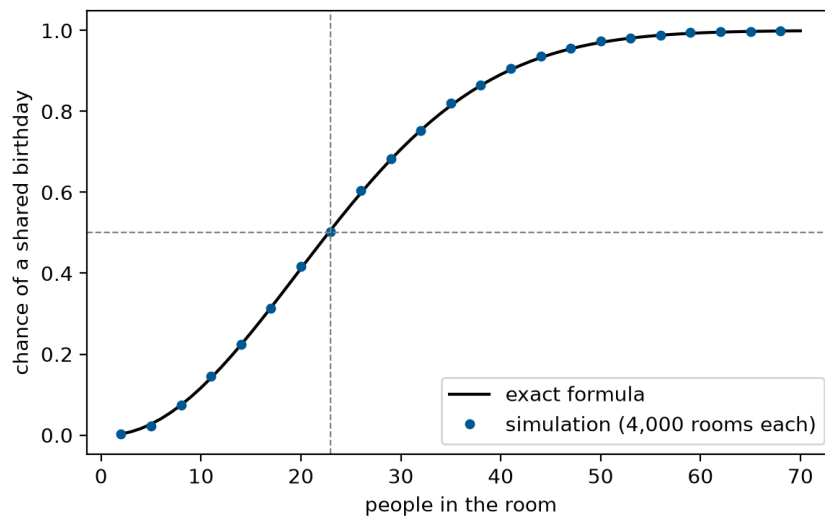


Figure 3.2: Simulated rooms (dots) match the exact formula (curve). The dashed lines cross at 23 people and a 50% chance.

3.4.1 A Question Without a Formula

Here is where simulation shines. How many people do you need for a better than even chance that *three* of them share a birthday? The neat product formula no longer works, because there are many ways for birthdays to overlap. But the simulation needs only a small change: count how many people have each birthday, and look for a day with three or more. Python's Counter does the counting.

```

1 import random
2 from collections import Counter
3
4 random.seed(5)
5
6 def triple_birthday(people, trials=10_000):
7     hits = 0
8     for _ in range(trials):
9         counts = Counter(random.randrange(365) for _ in range(people))
10        if max(counts.values()) >= 3:      # some day has 3 or more people
11            hits += 1
12    return hits / trials
13
14 for n in [23, 50, 80, 88, 100]:
15     print(n, triple_birthday(n))

```

```

23 0.0129
50 0.1274
80 0.4118
88 0.5157
100 0.6497

```

The chance first passes 50% at 88 people. Mathematicians have confirmed that number with much harder exact calculations, but the simulation found it in half a second.

Exercise 9 Lucky Seven

Simulate rolling two dice 100,000 times, and estimate the probability that the sum is 7. Then find the exact answer by listing all 36 equally likely outcomes.

Working Space

Answer on Page 67

Exercise 10 Two Heads in a Row

You flip a fair coin until you get two heads in a row. On average, how many flips does it take? Write a simulation to estimate the answer.

Working Space

Answer on Page 67

Exercise 11 Area by Darts

Change the dart program to estimate the area under the curve $y = x^2$ between $x = 0$ and $x = 1$. (Count a dart as a hit when $y \leq x^2$.) What does the estimate come out close to?

Working Space

Answer on Page 67

Numerical Methods of Describing Data

In the Introduction to Data Visualization chapter, you learned to read a distribution from a graph. A histogram or boxplot can show you immediately whether data is skewed, where the center sits, and how spread out the values are. But a picture has limits. You cannot subtract two histograms. You cannot use a boxplot to predict a salary or compare a test score to a national average. For that, you need numbers.

This chapter develops the numerical tools that complement the graphical ones you already know. By the end, you will be able to describe the center and spread of any distribution precisely, and avoid the most common errors people make when summarizing them.

4.1 Why Numerical Summaries Matter

In the Basic Statistics and Spreadsheets chapter, you met the mean, the median, and the standard deviation. Standard deviations show up in almost every statistical method you will meet later, from polls to clinical trials. This chapter explains exactly what they measure and how to compute them, along with the other key numbers that summarize data.

There are three families of numerical measures, each answering a different question about a data set:

- *Measures of central tendency* ask: what is a typical value?
- *Measures of variation* ask: how spread out are the values?
- *Measures of position* ask: where does a particular value sit relative to the rest?

Each family has more than one tool, and choosing the right one depends on the shape of your data. This chapter covers center and spread. Measures of position, and the connection between shape and the choice of numerical summary, are the subject of the next chapter.

4.1.1 Summation Notation

Nearly every numerical measure involves adding things up. Before computing anything, we need a compact way to write sums.

Suppose you record the daily sales at a shop over n days, calling them $X_1, X_2, \dots, X_n$. Writing out every term quickly becomes unwieldy. The Greek capital letter Σ (read as “sigma”) gives us a cleaner way:

$$\sum_{i=1}^n X_i = X_1 + X_2 + \cdots + X_n$$

The expression below Σ tells you where to start; the expression above tells you where to stop. When the range is clear from context, we often write $\sum X_i$ or simply $\sum X$.

Consider Ben, who scores the following on ten quizzes:

8, 9, 0, 10, 10, 8, 7, 9, 10, 5

Ben’s total score is

$$\sum_{i=1}^{10} X_i = 8 + 9 + 0 + 10 + 10 + 8 + 7 + 9 + 10 + 5 = 76$$

This notation will appear in every formula in this chapter. When you see $\sum X_i$, read it as “add up all the values of X .”

Exercise 12 Summation Practice

Sasha records the daily high temperatures (in degrees Celsius) over one week: 22, 19, 25, 21, 18, 23, 20.

Working Space

1. Write the sum of all seven values using summation notation and compute it.

2. Compute $\sum_{i=1}^7 (X_i - 20)$. What do you

notice about how this compares to your answer in (a)?

[Answer on Page 68](#)

4.2 Measures of Center

You already know how to read the center of a distribution from a graph: it is roughly where the data balances. Now we want to pin that idea down to a single number. There are two main tools for this, and they do not always agree.

4.2.1 The Mean

The *mean* is the arithmetic average of a data set. The mean is the balancing point of the distribution: the point at which the data would balance if every value were a weight on a number line. This is the same seesaw idea from the graphical chapter, now expressed as a formula.

When the data represent an entire *population* of N values, the mean is denoted by the Greek letter μ (read “mu”):

$$\mu = \frac{\sum_{i=1}^N X_i}{N}$$

When the data are a *sample* of n values drawn from a larger population, the mean is denoted $\bar{X}$ (read “X-bar”):

$$\bar{X} = \frac{\sum_{i=1}^n X_i}{n}$$

The formula is identical in both cases. Only the notation changes to signal whether you are describing a full population or a sample.

Return to Ben’s ten quiz scores from the previous section:

8, 9, 0, 10, 10, 8, 7, 9, 10, 5

$$\bar{X} = \frac{8 + 9 + 0 + 10 + 10 + 8 + 7 + 9 + 10 + 5}{10} = \frac{76}{10} = 7.6$$

Notice what that zero does. One failed quiz pulled the mean down from what would otherwise be a strong set of scores. This is the key property of the mean: it is not *resistant*. Every value in the data set contributes to it, so extreme values, whether unusually high or low, can pull it noticeably away from the center of the bulk of the data.

4.2.2 The Median

The *median* takes a different approach. Rather than averaging all the values, it finds the middle one. Half the data falls at or below the median; half falls at or above it.

To find the median M of a data set with n measurements:

1. Arrange all measurements in increasing order.
2. Compute $l = \frac{n+1}{2}$.
3. The median M is the value of the l th measurement, counted from the smallest.

If n is odd, l is a whole number and the median is a value that actually appears in the data. If n is even, l falls halfway between two observations and the median is their average; it may not correspond to any value in the data.

Consider Daniel's car dealership, which tracks its daily sales. Over 14 days, the number of cars sold was:

14, 9, 23, 7, 11, 23, 17, 11, 3, 24, 21, 2, 20, 20

Arranged in order:

2, 3, 7, 9, 11, 11, 14, 17, 20, 20, 21, 23, 23, 24

With $n = 14$:

$$l = \frac{14 + 1}{2} = 7.5$$

The median is the average of the 7th and 8th observations. Counting from the smallest, the 7th is 14 and the 8th is 17:

$$M = \frac{14 + 17}{2} = 15.5 \text{ cars}$$

Now notice what happens if the dealership's best day (24 cars) were replaced by an exceptional day of 80 cars. The mean would shift upward. The median would not move at all, as the 7th and 8th values in the sorted list are unchanged. This is what it means for the median to be resistant! That is, it does not react to extreme values the way the mean does.

Exercise 13 Computing Mean and Median

A small bookshop records the number of books sold each day over eight days:

12, 15, 9, 11, 14, 8, 13, 52

1. Compute the mean.
2. Compute the median.
3. The value 52 represents an unusually high day. It could be a sale event. Which measure is more affected by it? Explain why.
4. Remove the value 52 and recompute both the mean and the median using the remaining seven values. How much did each change?

Working Space

Answer on Page 68

4.2.3 Choosing Between Them

The mean and median each have their place, and the choice between them follows a clear rule. The right choice depends on the shape of the distribution, which you can now read directly from a histogram or boxplot.

For a *symmetric distribution*, the mean and median are close to each other. Either works, but the mean is generally preferred because it uses every observation and forms the foundation of most inferential statistics you will encounter later.

For a *skewed distribution* or any data set with outliers, the mean is pulled toward the tail while the median stays near the bulk of the data. In these cases, the median gives a more honest picture of a typical value.

You saw this in *Introduction to Data Visualization*. Most people earn within a moderate range, but a small number earn vastly more. The mean income for a country is often much higher than what a typical person earns, precisely because it is dragged upward by those extreme high values. The median income is usually the figure reported when the goal is to describe a typical household.

The relationship between mean, median, and skew follows a consistent pattern:

- Symmetric: mean $\approx$ median
- Right-skewed: mean $>$ median
- Left-skewed: mean $<$ median

Exercise 14 Choosing a Measure of Center

For each situation, state whether you would report the mean or the median as the measure of center, and explain your reasoning.

Working Space

1. The distribution of exam scores in a class of 30 students is roughly symmetric with no outliers.
2. The distribution of monthly salaries at a company is strongly right-skewed. The CEO earns considerably more than all other employees.
3. A histogram of the ages at which peo-

ple in a city receive their first driving licence shows a large cluster between 16 and 19, and a long tail extending into the 30s and 40s.

4. After removing the CEO's salary from the company data in (b), the remaining distribution is roughly symmetric. Now which measure would you report, and does your answer change?

Answer on Page 68

4.3 Measures of Spread

Two data sets can share exactly the same mean and still tell completely different stories. What distinguishes them is spread: how far the values are scattered around the center. This section develops the numerical tools for measuring that scatter precisely.

4.3.1 Range and IQR

You already encountered these in the graphical chapter when building boxplots, so this is a brief review before introducing the more powerful measures.

The *range* is the simplest measure of spread:

$$R = \text{largest value} - \text{smallest value}$$

For the car dealership data, $R = 24 - 2 = 22$ cars. Range is easy to compute but unreliable, because it depends entirely on the two most extreme values and ignores everything in between. One unusual observation can make the range misleading.

The *interquartile range* (IQR) improves on this by focusing on the middle 50% of the data:

$$\text{IQR} = Q_3 - Q_1$$

Here Q_1 is the median of the lower half of the data, and Q_3 is the median of the upper half. For the 14 dealership values, the lower half is the 7 smallest values, $\{2, 3, 7, 9, 11, 11, 14\}$, so $Q_1 = 9$. The upper half is $\{17, 20, 20, 21, 23, 23, 24\}$, so $Q_3 = 21$. Therefore $IQR = 12$ cars. Some software uses a different rule for quartiles, so its values may differ slightly from these. Because IQR ignores the tails entirely, it is a resistant measure, so outliers do not affect it. When the median is the appropriate measure of center, IQR is the matching measure of spread.

4.3.2 Variance and Standard Deviation

Range and IQR both have a blind spot: they ignore most of the data. The *standard deviation* fixes this by taking every value into account. It measures how far, on average, each observation sits from the mean.

Before we can compute the standard deviation, we need the *variance*. The variance is the average of the squared distances from the mean. Squaring serves two purposes: it removes the sign so that values above and below the mean do not cancel each other out, and it penalizes large deviations more heavily than small ones.

The *population variance* is denoted σ^2 :

$$\sigma^2 = \frac{\sum_{i=1}^N (x_i - \mu)^2}{N}$$

The *sample variance* is denoted s^2 :

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

We divide by $n - 1$ rather than n when working with a sample. Dividing by n would systematically underestimate the true population spread because a sample tends to cluster closer to its own mean than the full population does. Subtracting one corrects for that.

The standard deviation is simply the square root of the variance, which brings the units back to the same scale as the original data:

$$\sigma = \sqrt{\sigma^2} \qquad s = \sqrt{s^2}$$

A population standard deviation is denoted σ ("sigma"); a sample standard deviation is denoted s .

Computing Standard Deviation by Hand

Let us work through Ben's quiz score data step by step. The ten scores are 8, 9, 0, 10, 10, 8, 7, 9, 10, 5 with $\bar{X} = 7.6$.

Step 1. Compute each deviation from the mean, $x_i - \bar{x}$:

x_i	$x_i - \bar{x}$	$(x_i - \bar{x})^2$
8	0.4	0.16
9	1.4	1.96
0	-7.6	57.76
10	2.4	5.76
10	2.4	5.76
8	0.4	0.16
7	-0.6	0.36
9	1.4	1.96
10	2.4	5.76
5	-2.6	6.76
		86.40

Step 2. Divide the sum of squared deviations by $n - 1 = 9$ to get the sample variance:

$$s^2 = \frac{86.40}{9} = 9.60$$

Step 3. Take the square root:

$$s = \sqrt{9.60} \approx 3.10$$

Notice that the zero score contributes a squared deviation of 57.76, which is larger than the contributions of all nine other scores combined. One extreme value had an outsized effect on the standard deviation.

Checking Your Work with Python

Hand computation is worth doing once, so the mechanics stay visible. After that, software does the arithmetic. Python's built-in statistics module computes the same sample variance and standard deviation directly:

```

1 import statistics
2
3 scores = [8, 9, 0, 10, 10, 8, 7, 9, 10, 5]
4
5 print("mean:    ", statistics.mean(scores))
6 print("variance:", statistics.variance(scores)) # divides by n - 1
7 print("stdev:   ", statistics.stdev(scores))   # divides by n - 1

```

This prints mean: 7.6, variance: 9.6, and stdev: 3.0983866769659336, matching the by-hand result from Steps 1 through 3. `statistics.variance` and `statistics.stdev` use the sample formulas, dividing by $n - 1$, which is the same convention this chapter uses. The population versions, `statistics.pvariance` and `statistics.pstdev`, divide by n instead.

Exercise 15 Computing Variance and Standard Deviation

Daniel's car dealership records the number of cars sold each day over 14 days:

Working Space

14, 9, 23, 7, 11, 23, 17, 11, 3, 24, 21, 2, 20, 20

The mean is $\bar{X} \approx 14.64$ cars.

1. The sum of squared deviations $\sum (x_i - \bar{x})^2$ for this data set is approximately 743.21. Use this to compute the sample variance s^2 .
2. Compute the sample standard deviation s .
3. Standard deviation is measured in the same units as the original data. What are the units of s here? What are the units of s^2 ?

4. A second dealership has a mean of 14.64 cars per day and $s = 2.1$ cars. Without any other information, which dealership has more consistent daily sales? Explain.

Answer on Page 69

4.3.3 Interpreting Spread

Standard deviation is a unit of distance. Once you have computed s for a data set, you can ask of any individual observation: how many standard deviations away from the mean is this value? That question will become the foundation of the next section on z-scores.

For now, focus on two properties that are always true.

First, a larger standard deviation means a wider distribution. Compare two students' quiz scores:

Susan: 8, 9, 4, 8, 6, 8, 7, 9, 10, 5 $s_S \approx 1.90$
 Joaquin: 7, 8, 6, 7, 6, 6, 7, 5, 5, 7 $s_J \approx 0.97$

Susan's scores vary more from quiz to quiz. Joaquin's are tightly packed. The standard deviations reflect exactly what you see in the data: $s_S > s_J$.

Second, standard deviation is not resistant. Because it is computed from the mean, and the mean is pulled by extreme values, any outlier also inflates the standard deviation. Recall Ben's quiz example: the single zero score contributed a squared deviation of 57.76 out of a total of 86.40. One observation accounted for two-thirds of the total spread.

This is why the choice of spread measure mirrors the choice of center measure:

- Symmetric data, no outliers: use mean and standard deviation
- Skewed data or data with outliers: use median and IQR

Exercise 16 Choosing a Measure of Spread

For each situation, state which measure

Working Space

of spread (standard deviation or IQR) is more appropriate, and explain why.

1. The distribution of heights of adult men in a country is roughly symmetric and bell-shaped with no outliers.
2. The distribution of house prices in a city is strongly right-skewed, with a small number of luxury properties selling for many times more than the typical home.
3. Reine removes all outliers from a right-skewed data set. The remaining distribution is approximately symmetric. Which measure of spread should she now report?

Answer on Page 69

The next chapter, Shape, Position, and Outliers, shows how the shape of a distribution affects these measures, and how to describe where a single value sits within the data.

Shape, Position, and Outliers

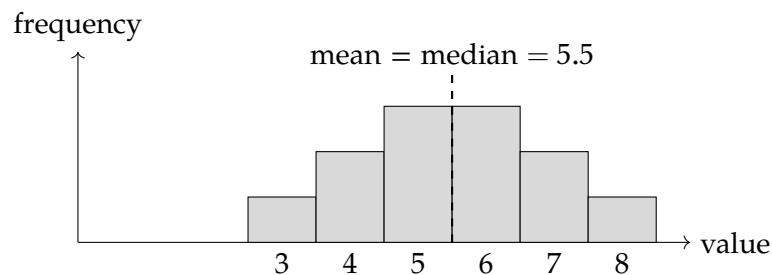
Two classes take the same test. Both have a mean score of 75, yet in one class almost everyone scored between 70 and 80, while the other has a cluster of A's and a long tail of failing grades. You scored 85: is that impressive? It depends on the shape of the scores and on where your score sits among them. In the Numerical Methods of Describing Data chapter, you learned to measure center and spread. This chapter shows how the shape of a distribution changes which measures you should trust, and how to describe the position of a single value with z-scores, percentiles, and quartiles.

5.1 Connecting Shape to Numerical Measures

You already know how to read shape from a histogram or boxplot. This section translates what you see graphically into numerical consequences: specifically, what skew and outliers do to the measures you just computed.

5.1.1 Symmetric Data: Mean Approximately Equals Median

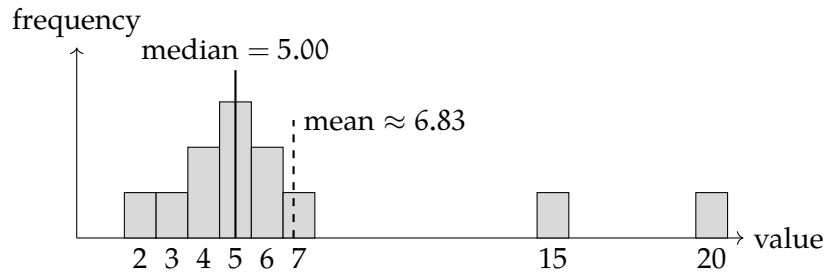
When a distribution is roughly symmetric, the mean and median land close to each other. Neither tail is pulling the mean away from the center of the data. This is the situation where mean and standard deviation are both appropriate, and where most inferential procedures work as intended.



A roughly symmetric data set: 3, 4, 4, 5, 5, 5, 6, 6, 6, 7, 7, 8. The mean and median land at exactly the same value.

5.1.2 Skewed Data: Mean Pulled Toward the Tail

When a distribution is skewed, the mean chases the tail while the median stays near the bulk of the data. The direction of the skew tells you exactly which way the mean is pulled:



The right-skewed data set below: two extreme values at 15 and 20 pull the mean well past the median.

- Right-skewed: $\text{mean} > \text{median}$. The right tail pulls the mean upward.
- Left-skewed: $\text{mean} < \text{median}$. The left tail pulls the mean downward.

This pattern follows directly from what you already know about the mean. Because the mean uses every value in its calculation, a cluster of extreme values in the right tail will drag it rightward. The median, which depends only on rank order, does not move.

Consider a small right-skewed data set:

2, 3, 4, 4, 5, 5, 5, 6, 6, 7, 15, 20

The mean is approximately 6.83 and the median is 5.00. The two extreme values at 15 and 20 pull the mean well above where most of the data sits. On a histogram, most bars would be clustered at the left with a long right tail. This shape corresponds to $\text{mean} > \text{median}$.

5.1.3 Effects of Outliers

An *outlier* is a value that sits far from the rest of the data. You saw the numerical definition in the graphical chapter: any value more than $1.5 \times \text{IQR}$ below Q_1 or above Q_3 is flagged as an outlier.

What outliers do to each measure follows directly from whether that measure is resistant:

- Mean: pulled toward the outlier. One extreme value can move the mean substantially.
- Median: unaffected. Moving the most extreme value further into the tail does not change the middle rank.
- Standard deviation: inflated. Because it is computed from the mean, an outlier that moves the mean also increases the squared deviations.
- IQR: unaffected. It depends only on Q_1 and Q_3 , which are determined by the middle of the data.

This is why mean and standard deviation always travel together, and median and IQR always travel together. They are basically matched pairs.

5.1.4 Changing Units on Summary Measures

One practical question comes up often: if every value in a data set is shifted or scaled, what happens to the summary measures? Suppose a teacher adds 20 points to every student's test score, or a government increases all property taxes by 2%. You do not need to recompute everything from scratch.

The rules depend on whether you are adding a constant or multiplying by one.

Adding a constant a to every value. Shifting every observation by the same amount shifts the measures of center and position by the same amount. Measures of spread are unaffected, because the distances between values do not change.

Multiplying every value by a constant b . Scaling every observation multiplies all measures, such as center, spread, and position, by the same factor. The absolute value $|b|$ is used for spread measures, since spread is always non-negative.

These rules are summarized in the table below.

Measure	$Y_i = X_i + a$	$Y_i = b \cdot X_i$
Mean	$\bar{Y} = \bar{X} + a$	$\bar{Y} = b\bar{X}$
Median	$\text{Median}(Y) = \text{Median}(X) + a$	$\text{Median}(Y) = b \cdot \text{Median}(X)$
Range	Unaffected	$\text{Range}(Y) = b \text{Range}(X)$
Std dev	Unaffected	$s_Y = b \cdot s_X$
Q_1, Q_3	$Q_1 + a, Q_3 + a$	$b \cdot Q_1, b \cdot Q_3$
IQR	Unaffected	$\text{IQR}(Y) = b \cdot \text{IQR}(X)$

The key pattern: adding a constant moves everything but changes no distances; multiplying stretches or compresses everything including distances.

Example. A class takes a five-question test worth 20 points each. The summary statistics

for their scores are:

Mean	62
Median	60
Range	45
Standard deviation	8
Q_1	48
Q_3	71
IQR	23

After grading, the teacher discovers a typographical error in question 2 that made it unanswerable. She adds 20 points to every student's score. The new summary statistics are:

Measure	Original	After adding 20
Mean	62	$62 + 20 = 82$
Median	60	$60 + 20 = 80$
Range	45	45 (unaffected)
Std dev	8	8 (unaffected)
Q_1	48	$48 + 20 = 68$
Q_3	71	$71 + 20 = 91$
IQR	23	23 (unaffected)

Every student's score shifted by the same amount, so the spread of the distribution is unchanged. The center and quartiles each moved up by exactly 20.

Exercise 17 Changing Units

County residents vote to increase property taxes by 2% to fund local schools. The current summary statistics for property tax (in dollars per year) are:

Mean	12,000
Median	8,000
Range	30,000
Standard deviation	5,000
Q_1	5,000
Q_3	14,000
IQR	9,000

Working Space

Each resident's new tax will be $1.02 \times$ their current tax. Compute the new

value of each summary measure.

Answer on Page 69

5.2 Z-scores

The previous section showed that adding a constant to every value shifts the mean but leaves the standard deviation unchanged. That means the distances between values, relative to the center, stay exactly the same. There is a natural way to express those distances as a single number: the *z-score*, also called a *standardized score*.

The z-score of a measurement tells you how far that value sits from the mean, expressed in units of standard deviations:

$$z = \frac{x - \bar{x}}{s}$$

For population data, replace $\bar{x}$ with μ and s with σ . The interpretation is the same either way.

A positive z-score means the value is above the mean. A negative z-score means it is below. A z-score of zero means the value equals the mean exactly. The magnitude tells you how many standard deviations separate the value from the center.

Example. A class takes a test with mean 74 and standard deviation 6. Two students receive the following scores:

$$\begin{array}{ll} \text{Amaka} & 88 \quad z = \frac{88 - 74}{6} = \frac{14}{6} \approx 2.33 \\ \text{George} & 70 \quad z = \frac{70 - 74}{6} = \frac{-4}{6} \approx -0.67 \end{array}$$

Amaka scored about 2.33 standard deviations above the class average. George scored about 0.67 standard deviations below it. George is actually closer to the mean than Amaka: $|-0.67| < |2.33|$.

Comparing Across Different Distributions

Z-scores become especially powerful when comparing measurements from distributions with different means and standard deviations. Without standardization, comparing values measured on different scales is meaningless. With z-scores, it becomes straightforward.

Suppose two cities report their daily high temperatures:

City	Mean daily high	Standard deviation
North Bend	80°F	12°F
South Bend	84°F	4°F

Both cities report a high of 95°F on the same day. Which city had the more unusually warm day?

The raw difference from the mean is larger for North Bend ($95 - 80 = 15$) than South Bend ($95 - 84 = 11$), but that ignores how variable the temperatures are in each city. Compute the z-scores:

$$\text{North Bend: } z = \frac{95 - 80}{12} = 1.25 \quad \text{South Bend: } z = \frac{95 - 84}{4} = 2.75$$

South Bend's 95°F sits 2.75 standard deviations above its mean, which is far more unusual than North Bend's 1.25. The day was more remarkably warm in South Bend, even though it was not further above South Bend's mean in raw degrees.

Exercise 18 Computing and Interpreting Z-scores

Daniel's car dealership from earlier in this chapter had mean daily sales of $\bar{X} \approx 14.64$ cars and standard deviation $s \approx 7.56$ cars.

Working Space

1. Compute the z-score for the best sales day (24 cars). Interpret it in context.
2. Compute the z-score for the worst sales day (2 cars). Interpret it in context.
3. A second dealership reports a best day of 31 cars, with mean 20 and standard deviation 4. Which dealer-

ship had the more unusually good best day? Show your work.

4. Daniel adds a fixed weekend bonus, increasing every daily total by 5 cars. Without recomputing, what is the z-score for the original best day of 24 cars under the new totals? Explain.

Answer on Page 70

Key points:

- A z-score expresses a value as a number of standard deviations above or below the mean
- Positive z-scores are above the mean; negative z-scores are below it
- Z-scores allow meaningful comparison across data sets measured on different scales
- Adding a constant to every value does not change any z-score; z-scores are invariant under shifting

5.3 Percentiles and Quartiles

The mean, median, and standard deviation describe the whole distribution. Sometimes the question is narrower: where does one particular value sit relative to everyone else? Mary, who scores 1400 on the SAT, wants to know not just that the median was 1040, but what percentage of students she scored above. That is what percentiles and quartiles answer.

Percentiles

The k th percentile P_k is the value such that $k\%$ of observations fall at or below it and $(100 - k)\%$ fall at or above it. For example, P_{95} is the 95th percentile: a value at P_{95} is higher than 95% of the data and lower than the remaining 5%.

Three percentiles you already know appear here under new names: $P_{25} = Q_1$, $P_{50} = Q_2 = M$ (the median), and $P_{75} = Q_3$.

To find P_k for a data set of n measurements:

1. Arrange all measurements in increasing order.
2. Compute $l = \frac{(n+1)k}{100}$.
3. P_k is the value of the measurement in position l , counted from the smallest. If l is not a whole number, interpolate between the two surrounding observations.

Return to the dealership data ($n = 14$, sorted):

2, 3, 7, 9, 11, 11, 14, 17, 20, 20, 21, 23, 23, 24

To find P_{90} :

$$l = \frac{(14+1) \times 90}{100} = 13.5$$

The 90th percentile falls halfway between the 13th and 14th observations. The 13th is 23 and the 14th is 24:

$$P_{90} = \frac{23 + 24}{2} = 23.5 \text{ cars}$$

Roughly 90% of days fall at or below 23.5 cars. (With only 14 days, the match is not exact: 12 of the 14 days, about 86%, are at or below 23.5.)

To find P_{10} :

$$l = \frac{(14+1) \times 10}{100} = 1.5$$

Halfway between the 1st observation (2) and the 2nd (3):

$$P_{10} = \frac{2 + 3}{2} = 2.5 \text{ cars}$$

Roughly 10% of days fall at or below 2.5 cars.

Quartiles and the Five-Number Summary

Quartiles divide a sorted data set into four equal parts using the 25th, 50th, and 75th percentiles. You first encountered these when building boxplots in the *Introduction to Data Visualization* chapter. Here is the formal connection:

- $Q_1 = P_{25}$: 25% of values fall at or below Q_1
- $Q_2 = P_{50} = M$: the median
- $Q_3 = P_{75}$: 75% of values fall at or below Q_3

In practice, quartiles are computed by splitting the sorted data into a lower half and an upper half and taking the median of each. For even n , split the data at the midpoint with no overlap. For odd n , exclude the median from both halves.

For the dealership data, split at the midpoint ($n = 14$, so seven values on each side):

Lower half: 2, 3, 7, **9**, 11, 11, 14 $\Rightarrow Q_1 = 9$
 Upper half: 17, 20, 20, **21**, 23, 23, 24 $\Rightarrow Q_3 = 21$

You may notice that the median-of-halves method gives $Q_1 = 9$, while the percentile formula gives $P_{25} = 8.5$. Both are correct, as different software and textbooks use slightly different conventions for quartile computation. Small disagreements between methods are normal and expected. On the AP exam, the median-of-halves method is standard.

Together with the minimum and maximum, the quartiles form the *five-number summary*: the same five values you used to build a boxplot.

Min	Q_1	Median	Q_3	Max
2	9	15.5	21	24

The five-number summary and the boxplot are two representations of exactly the same information: one numerical, one graphical.

Exercise 19 Relative Standing

Use the dealership data:

2, 3, 7, 9, 11, 11, 14, 17, 20, 20, 21, 23, 23, 24
 ($n = 14$).

Working Space

1. Compute P_{50} using the formula $l =$

$\frac{(n+1)k}{100}$. Does it match the median you computed earlier? What position in the sorted list does l point to?

2. On what percentage of days did the dealership sell at most 21 cars?
3. A sales consultant tells Daniel that a day with 17 cars sold falls in the “third quarter” of the distribution. Verify this by identifying which quarter 17 belongs to.
4. Daniel wants to flag the bottom 10% of days as underperforming. Using $P_{10} = 2.5$ cars, how many of the 14 days fall in this category? List them.

Answer on Page 70

5.3.1 Computing with Python

You have now computed measures of center, spread, and position by hand. In practice, you will rarely compute them this way. Python’s built-in `statistics` module handles all of them directly, using the dealership data set from throughout this chapter:

```
1 import statistics
2
3 sales = [14, 9, 23, 7, 11, 23, 17, 11, 3, 24, 21, 2, 20, 20]
4
5 print("mean: ", statistics.mean(sales))
6 print("median: ", statistics.median(sales))
7 print("variance:", round(statistics.variance(sales), 2))
8 print("stdev: ", round(statistics.stdev(sales), 2))
9
10 q1, q2, q3 = statistics.quantiles(sales, n=4)
11 print("Q1, Q2, Q3:", q1, q2, q3)
```

This prints a mean of 14.642857142857142, a median of 15.5, a variance of 57.17, and a standard deviation of 7.56, matching this chapter’s hand-computed values. The quartiles come out as 8.5, 15.5, and 21.5: `statistics.quantiles` uses the percentile formula, not

the median-of-halves method, which is why its Q_1 and Q_3 differ slightly from ours.

Introduction to Graphs

6.1 Introduction

How does a map app find the fastest route across town? How does a social network suggest people you might know? Both problems are about *connections*, and the math of connections is called graph theory.

In this chapter, a *graph* is not a plot on axes. It is a collection of *nodes* (also called vertices) joined by *edges*. For example, on some social networks, each user can follow any number of other users. We can think of each user as a node, and the edge points from the user who follows to the user they follow:

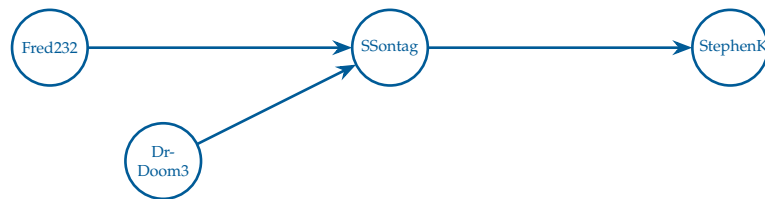


Figure 6.1: A graph showing four users and three “following” relationships.

Following is a *directed* relationship: Fred232 follows SSontag, but SSontag does not follow Fred232. So we would say that this is a *directed graph* with four nodes and three edges.

There are also undirected graphs. Imagine a graph of the big data lines between cities. Data flows both ways on every line, like traffic on a two-way road:

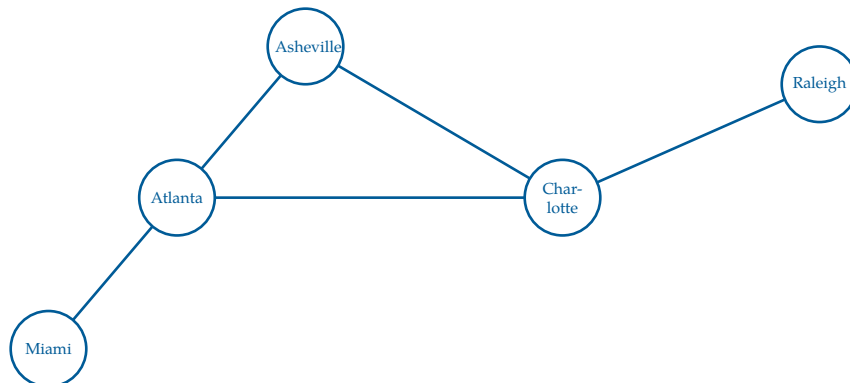


Figure 6.2: Cities in the southeastern United States (nodes) connected by data lines (edges).

The arrows are gone; if data can flow from Miami to Raleigh, then data can flow from Raleigh to Miami. A route that follows edges from one node to another is called a *path*. The number of edges touching a node is its *degree*. Atlanta and Charlotte each touch 3 edges, so each has a degree of 3.

There is a whole branch of mathematics called *Graph Theory* that studies the properties of graphs. Here are two questions that mathematicians might ask about this graph:

- What is the smallest number of edges that we would need to follow to get from Miami to Raleigh?
- Does the graph have any paths where you could end up where you started?

That second question asks if there is a *cycle* in the graph. This graph has one cycle: Atlanta $\rightarrow$ Asheville $\rightarrow$ Charlotte $\rightarrow$ Atlanta.

Some graphs are *connected*: You can get from one node to any other node by following edges. Is this graph connected?

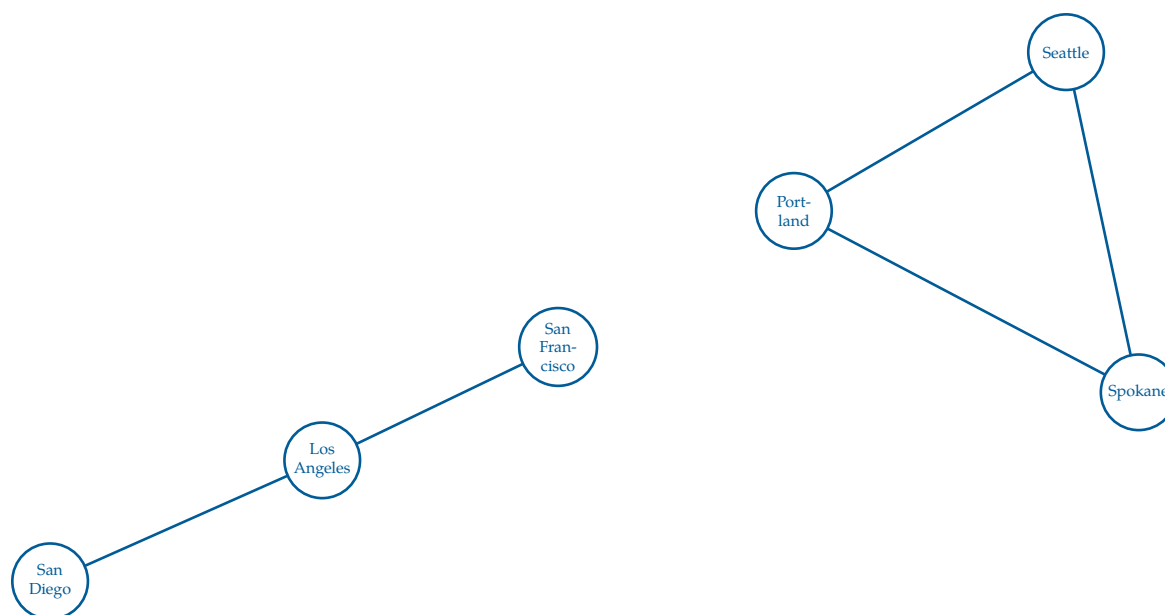


Figure 6.3: A set of cities (nodes) that are not all connected. Each set of *connected* cities forms a *connected component*.

This graph is *not* connected! You cannot follow edges from San Diego to Seattle. It falls into two *connected components*: groups of nodes that are connected to each other but not to anything else.

- San Diego, Los Angeles, San Francisco

- Portland, Seattle, Spokane

To say it precisely: a graph is connected if there is a path between every pair of its nodes. A *subgraph* is any part of a graph: some of its nodes and some of the edges between them. Every connected component is a subgraph, but most subgraphs are not connected components.

In graph data, the nodes and edges often have attributes. For example, a node representing a city might have a name and a population. An edge representing a data line might have a bandwidth (bits per second) and a latency (how long a bit takes to travel from one end to the other).

Exercise 20 Word Graphs

Working Space

Consider the concept of a graph as a set of nodes and edges. How would you create a graph where the nodes are words and the edges represent a one letter difference between two words?

For example, the word *cat* would be connected to *cot*, *car*, and *cut*, but not to *dog*.

Define explicitly what the nodes would be in this graph, and what the edges would be. Draw a simple example of this graph with at least 5 nodes and 5 edges.

Answer on Page 70

6.2 Finding Good Paths

For many problems, we want the best path from one node to another. If all the edges are alike, the best path is usually the one with the fewest edges.

Worked example. A graph has edges A–B costing 3, B–C costing 4, and A–C costing 9.

What is the cheapest way from A to C?

Solution. The direct edge costs 9, but the path $A \rightarrow B \rightarrow C$ costs $3 + 4 = 7$. The path with the fewest edges is not always the cheapest.

Sometimes the edges have a cost attribute. For example, you might want to find the cheapest way to ship a container from New York City to Long Beach, California. In this case the nodes are train depots. Each train line between the depots has a cost. What is the cheapest path?

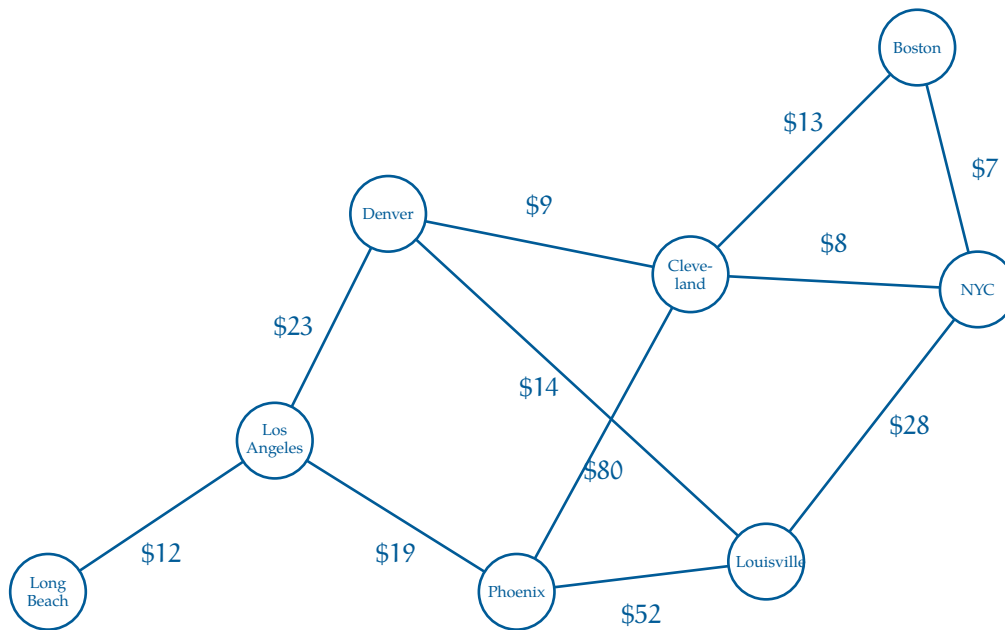


Figure 6.5: A graph of interconnected cities where each edge has a respective cost.

Edges with costs are called *weighted edges*, and they form a *weighted graph*.

(In a *directed* graph, a node's degree splits into two numbers: its *in-degree*, the number of edges pointing into it, and its *out-degree*, the number pointing out. In the "following" graph, a user's in-degree is their number of followers.)

Exercise 21 Ship it

Using figure 6.5, find the cheapest way to ship a container from New York City

Working Space

to Long Beach. What does it cost?

Answer on Page 71

The graphs that you see here are really small, so finding efficient paths is not difficult — you could just try all of them! However, in many computer programs, we are working with *millions* of nodes and edges. Efficient graph algorithms are *really* important.

6.3 Graphs in Python

There are many ways to represent graph data. There are database systems that are specifically designed to hold and analyze graph data. Not surprisingly, these are called *Graph Databases*.

You may also hear about *adjacency lists*, which store, for each node, the list of nodes it is connected to. Each direct connection from one node to another is called a *neighbor*. Simply, adjacency lists store lists of neighbors.

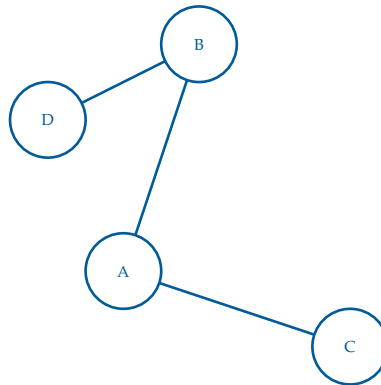


Figure 6.6: A simple (undirected unweighted) graph used to make an adjacency list.

As an adjacency list, this would look like:

- A: [B, C]
- B: [A, D]
- C: [A]
- D: [B]

Exercise 22 **Weighted Adjacency Lists**

Working Space

Consider the concept of adjacency lists discussed above. How would the storage method change if the edges were to have weights?

Answer on Page 71

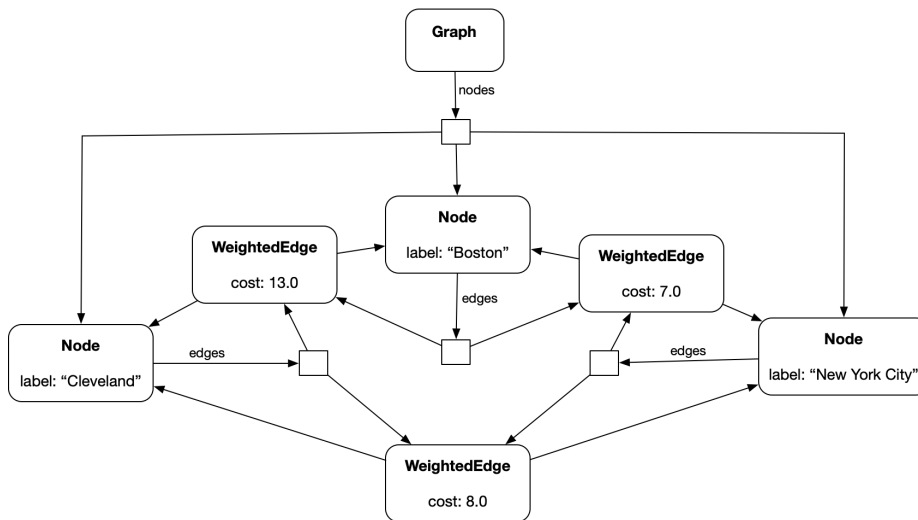
For simplicity, you are going to write Python classes that will let you represent an undirected graph with weighted edges, like the shipping problem above.

(Naturally, things would look a little different if the graph were directed or the edges were unweighted, but this is a good starting place.)

Create a file called `graph.py`. This will hold the code for your `Node` and `WeightedEdge` classes. We will also create a `Graph` class that will just hold onto the list of its nodes.

- A `Node` will have a label string and a list of edges that touch it.
- A `WeightedEdge` will have a cost and two nodes: `node_a` and `node_b`.
- A `Graph` will have a list of nodes.

Here is what the object diagram would look like if you had only three cities:



Put this code into graph.py

```

1 class Node:
2     def __init__(self, label):
3         self.label = label
4         self.edges = []
5
6     def __repr__(self):
7         return f"(node:{self.label}, edges:{len(self.edges)})"
8
9 class WeightedEdge:
10     def __init__(self, cost, node_a, node_b):
11         self.cost = cost
12         self.node_a = node_a
13         node_a.edges.append(self)
14         self.node_b = node_b
15         node_b.edges.append(self)
16
17     def other_end(self, node_from):
18         if self.node_a == node_from:
19             return self.node_b
20         if self.node_b == node_from:
21             return self.node_a
22         raise ValueError("node is not an endpoint of this edge")
23
24 class Graph:
25     def __init__(self):
26         self.nodes = []
27
28     def add_node(self, new_node):
29         self.nodes.append(new_node)
30
31     def __repr__(self):
32         return f"(Graph:{self.nodes})"

```

The `other_end` method answers “if I am at this node, where does this edge take me?” You will need it to walk through the graph. Now, let us create some instances of `Node` and `WeightedEdge` and wire them together. Create another file in the same directory called `cities.py`. Put in this code:

```
1 import graph
2
3 # Create an empty graph
4 network = graph.Graph()
5
6 # Create city nodes and add to graph
7 long_beach = graph.Node("Long Beach")
8 network.add_node(long_beach)
9 los_angeles = graph.Node("Los Angeles")
10 network.add_node(los_angeles)
11 denver = graph.Node("Denver")
12 network.add_node(denver)
13 phoenix = graph.Node("Phoenix")
14 network.add_node(phoenix)
15 louisville = graph.Node("Louisville")
16 network.add_node(louisville)
17 cleveland = graph.Node("Cleveland")
18 network.add_node(cleveland)
19 boston = graph.Node("Boston")
20 network.add_node(boston)
21 nyc = graph.Node("New York City")
22 network.add_node(nyc)
23
24 # Create edges
25 graph.WeightedEdge(12, long_beach, los_angeles)
26 graph.WeightedEdge(23.0, los_angeles, denver)
27 graph.WeightedEdge(19, los_angeles, phoenix)
28 graph.WeightedEdge(52, phoenix, louisville)
29 graph.WeightedEdge(14, denver, louisville)
30 graph.WeightedEdge(80, phoenix, cleveland)
31 graph.WeightedEdge(9, denver, cleveland)
32 graph.WeightedEdge(8, cleveland, nyc)
33 graph.WeightedEdge(28, louisville, nyc)
34 graph.WeightedEdge(7, nyc, boston)
35 graph.WeightedEdge(13, cleveland, boston)
36
37 print(network)
```

Run it:

```
1 python3 cities.py
```

You should see some rather unexciting output. Each node reports its degree:

```
1 (Graph:[(node:Long Beach, edges:1), (node:Los Angeles, edges:3), (node:Denver,  
↩ edges:3),  
2 (node:Phoenix, edges:3), (node:Louisville, edges:3), (node:Cleveland, edges:4),  
3 (node:Boston, edges:2), (node:New York City, edges:3)])
```

Do not worry; in the shortest path chapter, you will teach this code to find the cheapest route on its own.

Exercise 23 Cheapest Route

Edges A–B cost 2, B–D cost 5, A–C cost 4, and C–D cost 3. What is the cheapest route from A to D, and what does it cost?

Working Space

Answer on Page 72

Answers to Exercises

Answer to Exercise 1 (on page 5)

1. $\frac{\pi}{4}$
2. $\frac{\pi}{2}$
3. $\frac{\pi}{3}$
4. $-\frac{\pi}{2}$
5. $\cos\left(-\frac{\pi}{3}\right) = \frac{1}{2}$, and $\arccos$ answers between 0 and π , so the result is $\frac{\pi}{3}$, not $-\frac{\pi}{3}$.

Answer to Exercise 2 (on page 6)

The height is opposite the angle and the ladder is the hypotenuse, so $\theta = \arcsin \frac{5.5}{6} \approx 66.4^\circ$. That is shallower than 75° , so the bottom of the ladder is too far from the wall and it could slide out.

Answer to Exercise 3 (on page 6)

$$\theta = \arctan \frac{1.5}{40} \approx 2.15^\circ.$$

Answer to Exercise 4 (on page 13)

```
1 import random
2
3 wins = 0
4 games = 100_000
5 for _ in range(games):
6     car = random.randint(1, 3)
7     pick = random.randint(1, 3)
8     if pick != car: # switching wins when the first pick was a goat
9         wins += 1
```

```
10 print(wins / games)
```

It prints a number close to 0.667 (for example, 0.6672).

Answer to Exercise 5 (on page 15)

$$1 - \frac{365}{365} \cdot \frac{364}{365} \cdots \frac{336}{365} \approx 0.706$$

About 71%. With 50 people, it is about 97%, and with 70 people it is 99.9%.

Answer to Exercise 6 (on page 16)

Multiply by $\frac{1}{3}$: $\frac{1}{3}S = \frac{1}{9} + \frac{1}{27} + \cdots$. Subtract: $S - \frac{1}{3}S = \frac{1}{3}$, so $\frac{2}{3}S = \frac{1}{3}$ and $S = \frac{1}{2}$.

Answer to Exercise 7 (on page 25)

$\pi \approx 4 \cdot \frac{312}{400} = 3.12$. Since the error shrinks like $\frac{1}{\sqrt{n}}$, making it 10 times smaller takes $10^2 = 100$ times as many darts: about 40,000.

Answer to Exercise 8 (on page 27)

```
1 import random
2
3 def four_doors():
4     car = random.randrange(4)
5     pick = random.randrange(4)
6     goats = [d for d in range(4) if d != pick and d != car]
7     opened = random.choice(goats)
8     rest = [d for d in range(4) if d != pick and d != opened]
9     return random.choice(rest) == car
10
11 games = 100_000
12 print(sum(four_doors() for _ in range(games)) / games)
```

It prints about 0.375. Explanation: your first pick is wrong $\frac{3}{4}$ of the time. Then the car is behind one of the two doors you might switch to, and you choose the right one half the time: $\frac{3}{4} \cdot \frac{1}{2} = \frac{3}{8}$. That beats staying, which wins only $\frac{1}{4}$.

Answer to Exercise 9 (on page 29)

```

1 import random
2
3 rolls = 100_000
4 sevens = sum(random.randint(1, 6) + random.randint(1, 6) == 7 for _ in
5               ↪ range(rolls))
6 print(sevens / rolls)

```

It prints about 0.167. Exactly: the outcomes (1,6), (2,5), (3,4), (4,3), (5,2), and (6,1) give 7, so the probability is $\frac{6}{36} = \frac{1}{6} \approx 0.1667$.

Answer to Exercise 10 (on page 30)

```

1 import random
2
3 def flips_until_hh():
4     flips = 0
5     previous_heads = False
6     while True:
7         flips += 1
8         heads = random.random() < 0.5
9         if heads and previous_heads:
10             return flips
11         previous_heads = heads
12
13 trials = 100_000
14 print(sum(flips_until_hh() for _ in range(trials)) / trials)

```

It prints a number close to 6. (The exact answer is 6, which you can prove with the tools of the probability chapters.)

Answer to Exercise 11 (on page 30)

Replace the test with `if y <= x**2:`, and report `inside / darts` instead of multiplying by 4. With 100,000 darts, the estimate is about 0.333. The exact area is $\frac{1}{3}$, as you will be able to show with integrals in Volume 30. Using random points to estimate areas like this is called *Monte Carlo integration*; for complicated shapes in many dimensions, it is often the only practical method.

Answer to Exercise 12 (on page 32)

1. $\sum_{i=1}^7 X_i = 22 + 19 + 25 + 21 + 18 + 23 + 20 = 148$.
2. $(2) + (-1) + (5) + (1) + (-2) + (3) + (0) = 8$. The sum decreased by $7 \times 20 = 140$, which is n times the constant subtracted. Shifting every value by a constant shifts the total by n times that constant.

Answer to Exercise 13 (on page 35)

1. $\bar{X} = \frac{12 + 15 + 9 + 11 + 14 + 8 + 13 + 52}{8} = \frac{134}{8} = 16.75$ books.
2. Sorted: 8, 9, 11, 12, 13, 14, 15, 52. $l = \frac{9}{2} = 4.5$, so $M = \frac{12 + 13}{2} = 12.5$ books.
3. The mean is more affected. It uses every value in the calculation, so the outlier of 52 pulls it upward. The median depends only on the middle two values, which are unaffected by how extreme the highest value is.
4. Without 52: $\bar{X} = \frac{82}{7} \approx 11.71$ books. Sorted: 8, 9, 11, 12, 13, 14, 15; median = 12. The mean dropped by about 5 books; the median changed by only 0.5.

Answer to Exercise 14 (on page 36)

1. Mean. Symmetric data with no outliers means the mean and median are close, and the mean is preferred for further calculations.
2. Median. The CEO's salary is an extreme outlier that would pull the mean well above what a typical employee earns. The median better represents the center of the bulk of the data.
3. Median. The long right tail indicates right skew; the mean would be pulled toward the older ages and away from where most people cluster.
4. Mean. Once the outlier is removed and the distribution is symmetric, the mean is the appropriate choice.

Answer to Exercise 15 (on page 40)

1. $s^2 = \frac{743.21}{14-1} = \frac{743.21}{13} \approx 57.17 \text{ cars}^2$.
2. $s = \sqrt{57.17} \approx 7.56 \text{ cars}$.
3. s is measured in cars. s^2 is measured in cars^2 , which has no natural interpretation. This is one reason we prefer the standard deviation over the variance for reporting.
4. The second dealership. Both have the same mean, but a smaller standard deviation means the daily sales are more tightly clustered around the average.

Answer to Exercise 16 (on page 41)

1. Standard deviation. Symmetric data with no outliers is exactly the situation where the mean and standard deviation are appropriate.
2. IQR. The extreme high values would inflate the standard deviation substantially, making it a misleading measure of typical spread. IQR focuses on the middle 50% and is unaffected by the luxury properties.
3. Standard deviation. Once the distribution is symmetric and outlier-free, the standard deviation is the preferred measure and will now accurately reflect the spread of the remaining data.

Answer to Exercise 17 (on page 46)

Multiplying every value by $b = 1.02$:

Measure	Original	After multiplying by 1.02
Mean	12,000	$1.02(12,000) = 12,240$
Median	8,000	$1.02(8,000) = 8,160$
Range	30,000	$1.02(30,000) = 30,600$
Std dev	5,000	$1.02(5,000) = 5,100$
Q_1	5,000	$1.02(5,000) = 5,100$
Q_3	14,000	$1.02(14,000) = 14,280$
IQR	9,000	$1.02(9,000) = 9,180$

Unlike adding a constant, multiplying affects the spread measures too. Every distance in the distribution is stretched by the same factor.

Answer to Exercise 18 (on page 48)

1. $z = \frac{24 - 14.64}{7.56} \approx 1.24$. The best day was about 1.24 standard deviations above the mean. It was a notably good day but not extreme.
2. $z = \frac{2 - 14.64}{7.56} \approx -1.67$. The worst day was about 1.67 standard deviations below the mean and further from typical than the best day was above it.
3. Second dealership: $z = \frac{31 - 20}{4} = 2.75$. The second dealership's best day was more unusually high at 2.75 standard deviations above its mean, compared to 1.24 for the first dealership.
4. Still $z \approx 1.24$. Adding a constant to every value shifts the mean by the same constant and leaves the standard deviation unchanged. The relative position of every observation in the distribution stays the same, so z-scores are unaffected.

Answer to Exercise 19 (on page 51)

1. $l = \frac{(14 + 1)(50)}{100} = 7.5$. The 50th percentile falls halfway between the 7th observation (14) and the 8th (17): $P_{50} = \frac{14 + 17}{2} = 15.5$. This matches the median exactly.
2. Count them: 11 of the 14 days had 21 or fewer cars, so $\frac{11}{14} \approx 79\%$. (Since $Q_3 = 21$, you could also estimate about 75%; with small data sets the quartiles are only approximate percentages.)
3. $Q_1 = 9$ and $Q_3 = 21$. The value 17 lies between $Q_2 = 15.5$ and $Q_3 = 21$, placing it in the third quarter (between the median and Q_3). The consultant is correct.
4. Days with 2 or fewer cars: only the day with 2 cars sold. That is 1 out of 14 days, or about 7%, consistent with the bottom 10% cutoff of 2.5.

Answer to Exercise 20 (on page 57)

The nodes in this graph are unique words. The edges represent a valid one letter operations that can transform one word into another word. These operations include adding a letter, removing a letter, or changing a letter into another. For example, the word cat would be connected to cot (change), cart (add), and cut (change), but not to dog. Notice that this graph has cycles: cat $\rightarrow$ cot $\rightarrow$ cut $\rightarrow$ cat. Word games like “change one letter at a time to turn cold into warm” are really path-finding puzzles on this graph.

Here is an example of a word graph:

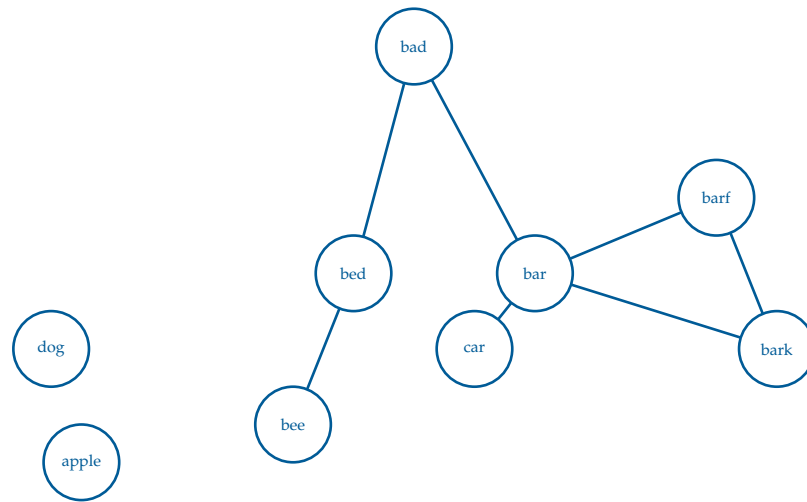


Figure 6.4: A word graph where the nodes are unique words and the edges represent a one letter operations between two words.

Answer to Exercise 21 (on page 58)

NYC $\rightarrow$ Cleveland $\rightarrow$ Denver $\rightarrow$ Los Angeles $\rightarrow$ Long Beach costs $8 + 9 + 23 + 12 = \$52$. The next best route, through Louisville and Denver, costs $28 + 14 + 23 + 12 = \$77$, and any route through Phoenix costs more still.

Answer to Exercise 22 (on page 60)

In a weighted graph, each node stores its neighbors along with the weight of each edge. A tuple works well. For example, if the edge A–B costs 4, A–C costs 3, and B–D costs 8:

- A: [(B, 4), (C, 3)]
- B: [(A, 4), (D, 8)]
- C: [(A, 3)]
- D: [(B, 8)]

Answer to Exercise 23 (on page 63)

Route $A \rightarrow C \rightarrow D$ costs $4 + 3 = 7$. Route $A \rightarrow B \rightarrow D$ costs $2 + 5 = 7$. Both routes tie at 7.



INDEX

- kth percentile, 49
- adjacency list, 59
- adjacency lists, 59
- arccosine, 4
- arcsine, 3
- arctangent, 4
- arrow paradox, 17
- Barber's paradox, 9
- Bessel's correction, 38
- birthday problem, 13
 - simulation, 27
- coastline paradox, 20
- connected, 56
- connected component, 56
- connected components, 56
- cycle, 56
- degree, 56
- dichotomy paradox, 15
- diminishing returns, 21
- directed, 55
- directed graph, 55
- edge, 55
- edges, 55
- five-number summary, 51
- fractals, 20
- geometric series, 16
- grandfather paradox, 18
- graph, 55
 - connected, 56
 - directed, 55
 - undirected, 55
- Graph Databases, 59
- Graph Theory, 56
- Hans Moravec, 20
- in-degree, 58
- infinite series, 15
- interquartile range, 37
- inverse trigonometric functions, 3
- Law of Large Numbers, 24
- limits, 15
- linear transformation, 45
- mean, 33
- measures of central tendency, 31
- Measures of central tendency, 31
- Measures of position, 31
- measures of position, 31
- Measures of variation, 31
- measures of variation, 31
- median, 34
- mobius strip, 19
- Monte Carlo integration, 67
- Monte Carlo simulation, 23
- Monty Hall Problem, 12
 - simulation, 25

- Moravec's paradox, 20
- neighbor, 59
- neural scaling laws, 21
- node, 55
- nodes, 55
- out-degree, 58
- outlier, 44
- paradox, 9
- path, 56
- penrose steps, 19
- percentile, 49
- pi
 - Monte Carlo estimate, 24
- population, 33
- population variance, 38
- proof by contradiction, 9
- pseudorandom, 23
- Quartiles, 51
- quartiles, 51
 - median-of-halves method, 51
- range, 37
- resistant, 34
- resistant measure, 34
- Russell's Paradox, 10
- sample, 33
- sample variance, 38
- Schrödinger's cat, 17
- set theory, 10
- Simpson's paradox, 10
- skewed distribution, 36
- standard deviation, 38
- standardized score, 47
- statistics module, 40
- subgraph, 57
- summation notation, 32
- superposition, 17
- symmetric distribution, 36
- thought experiment, 9
- thought experiments, 9
- time travel paradoxes, 18
- variance, 38
- weighted edges, 58
- weighted graph, 58
- z-score, 47
- Zeno's paradoxes, 15