



CONTENTS

1	Maps and Location Data	5
1.1	Longitude and Latitude	5
1.1.1	Longitude and Latitude	5
1.1.2	Nautical Mile	7
1.1.3	Haversine Formula	7
1.2	Geocoding and Reverse Geocoding	10
1.2.1	Geocoding	10
1.2.2	Reverse Geocoding	11
1.3	Making a Map	12
2	Discrete Probability and Random Variables	13
2.1	Introduction to Discrete Probability	13
2.1.1	The Probability of All Possibilities is 1.0	14
2.1.2	Independence and Dependence	14
2.1.3	Correlation does not Imply Causation	15
2.1.4	Why 7 is the most likely sum of two dice	16
2.1.5	Random Numbers and Python	17
2.2	What Is a Random Variable?	22
2.3	Expected Value	23
2.4	Variance and Standard Deviation	23
2.5	Linearity of Expectation	23
2.6	A Game Example	23
2.7	Looking Ahead	25

3	Counting	27
3.1	Beginning Combinatorics	27
3.1.1	Choose	29
3.1.2	What About Fractional Exponents?	30
3.1.3	Applications in Probability	30
3.2	Permutations and Sorting	31
3.2.1	Notation	31
3.2.2	Sorting in Python	33
3.2.3	Inverses	33
3.2.4	Cycles	34
3.2.5	Further Learning	35
4	Binomial and Geometric Distributions	37
4.1	The Binomial Setting	37
4.1.1	The Binomial Formula	37
4.1.2	“At Least” and “At Most”	38
4.1.3	Mean and Standard Deviation	39
4.1.4	Shape	39
4.1.5	Sampling Without Replacement: The 10% Condition	39
4.2	Binomial Probabilities in Python	39
4.3	The Geometric Setting	42
4.3.1	Mean and Standard Deviation	43
4.3.2	No Memory	43
4.4	Looking Ahead: Rare Events	45
5	Poisson and Exponential Distributions	47
5.1	Counting Events	47
5.1.1	A Worked Example	47
5.2	Mean and Variance	48
5.3	Waiting Times	48
5.3.1	A Worked Example	48
5.4	The Connection	48
5.5	The Memoryless Property	49
5.6	Assumptions	49
6	Conditional Probability	51
6.1	Marginalization	52
6.2	Conditional Probability	52
6.3	Chain Rule for Probability	53

	3
A Answers to Exercises	57
Index	61

Maps and Location Data

1.1 Longitude and Latitude

1.1.1 Longitude and Latitude

When you share your location with a friend, your phone sends just two numbers. Those two numbers pin down one spot out of the 510 million square kilometers on Earth's surface.

The Earth can be represented as a sphere, and the position of a point on its surface can be described using two coordinates: latitude and longitude.



Figure 1.1: A diagram of latitude and longitude.

Latitude is a measure of a point's distance north or south of the equator, expressed in degrees. It ranges from -90° at the South Pole to $+90^\circ$ at the North Pole, with 0° representing the Equator. (See figures [1.2a](#) and [1.2b](#))

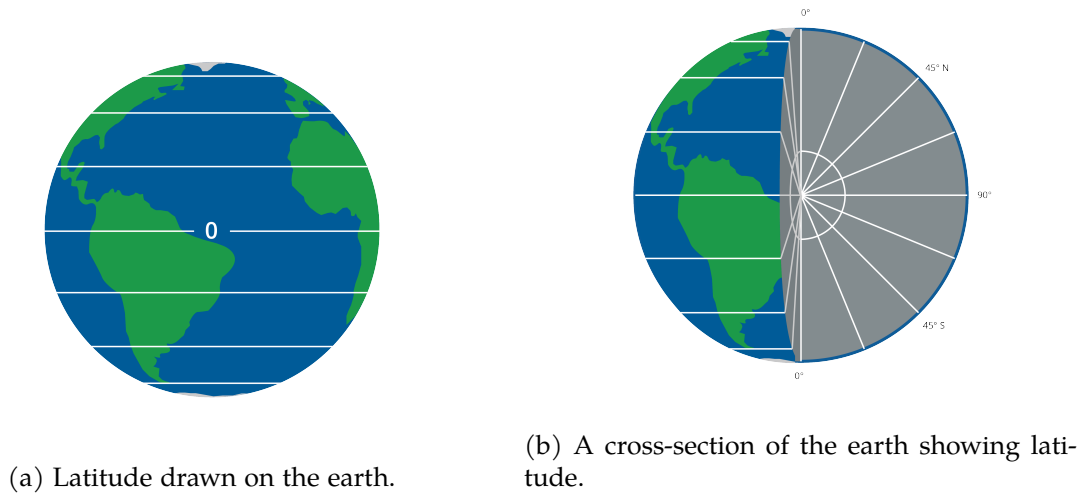


Figure 1.2: Latitude illustrations

Longitude, on the other hand, measures a point's distance east or west of the Prime Meridian (which passes through Greenwich, England). It ranges from -180° to $+180^\circ$, with the Prime Meridian represented as 0° . (See figures 1.3a and 1.3b)

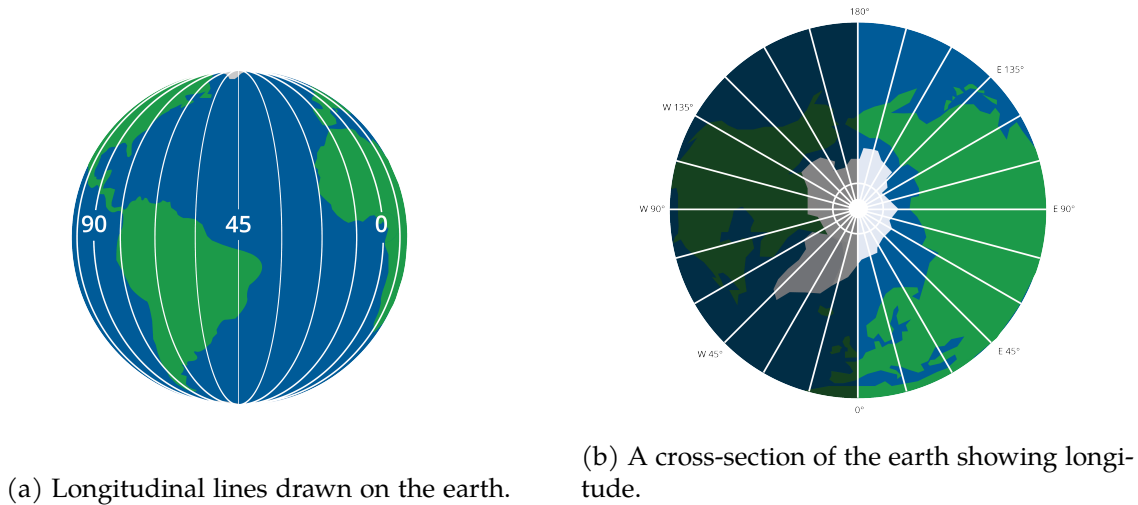


Figure 1.3: Longitude illustrations.

The coordinates follow a system of latitude and then longitude (in the majority of contexts). Here is the Longitude and Latitude of four different large cities:

- **London, England:** 51° N 0° W
- **New York City, New York, USA:** 40° N 74° W

- **Sydney, New South Wales, Australia:** 34° S 151° E
- **Rio de Janeiro, Brazil:** 22° S 43° W

These coordinates may be broken down further into minute and second components as well, for further geometric accuracy. Each degree ($^{\circ}$) is divided into 60 minutes ($'$). Each minute ($'$) is divided into 60 seconds ($''$), just like time. Many apps use decimal degrees instead: $40^{\circ} 42' 46''$ is $40 + \frac{42}{60} + \frac{46}{3600} \approx 40.7128^{\circ}$. South latitudes and west longitudes are written as negative numbers, so New York is $(40.7128, -74.0061)$.

- **London, England:** $51^{\circ} 30' 26''$ N $0^{\circ} 7' 39''$ W
- **New York City, New York, USA:** $40^{\circ} 42' 46''$ N $74^{\circ} 0' 22''$ W
- **Sydney, New South Wales, Australia:** $33^{\circ} 52' 0''$ S $151^{\circ} 12' 0''$ E
- **Rio de Janeiro, Brazil:** $22^{\circ} 54' 0''$ S $43^{\circ} 12' 0''$ W

1.1.2 Nautical Mile

A nautical mile is a unit of measurement used primarily in aviation and maritime contexts. It is based on the circumference of the Earth, and is defined as one minute ($1/60^{\circ}$) of latitude; in other words, **one minute of latitude is equal to 1 nautical mile**. This makes navigation easy: if a ship sails 60 nautical miles due north, it has moved exactly 1° of latitude. (The kilometer was also born from Earth's size: it was first defined so that the distance from the North Pole to the equator would be 10,000 km.) Because the Earth is not a perfect sphere, a minute of latitude varies slightly, so today the nautical mile is defined as exactly 1.852 kilometers (about 1.151 ordinary miles).

1.1.3 Haversine Formula

The *haversine formula* is an important equation in navigation for giving great-circle distances between two points on a sphere from their longitudes and latitudes. It is especially useful when it comes to calculating distances between points on the surface of the Earth, which we represent as a sphere for simplicity. See Figure 1.4

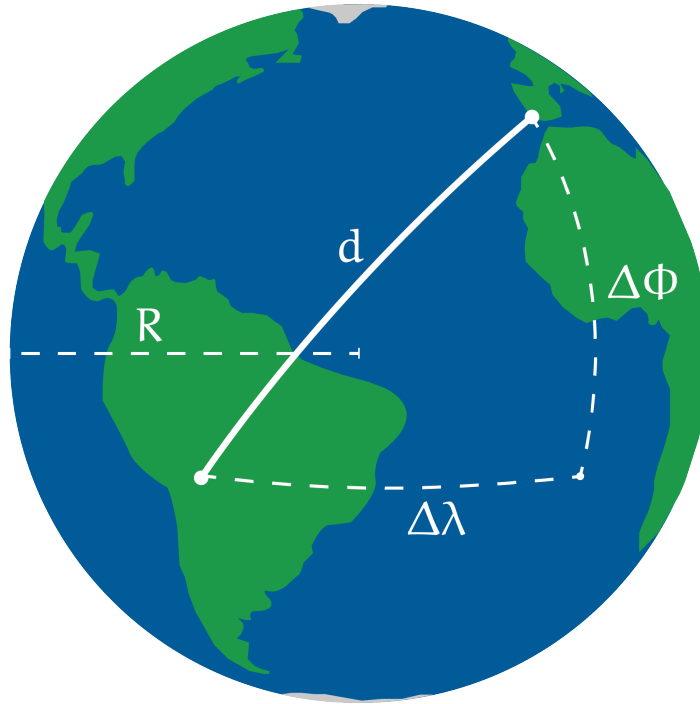


Figure 1.4: A diagram of the haversine formula.

In its basic form, the haversine formula is as follows:

The angle found using haversine $\text{haversine}(\theta) = \sin^2\left(\frac{\theta}{2}\right)$:

$$a = \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1) \cos(\phi_2) \sin^2\left(\frac{\Delta\lambda}{2}\right)$$

Compute c which determines the angular distance using `atan2`, a computer science tangential function with 2 arguments equivalent to $2 \arcsin(\sqrt{a})$ ¹:

$$c = 2 \cdot \text{atan2}\left(\sqrt{a}, \sqrt{1-a}\right)$$

Find d , the true distance along the sphere:

¹Note that this function is not available on calculators, but commonly used in computer science programs. You may use the singular argument equivalent in calculations.

$$d = R \cdot c$$

In one line, this is:

$$d = 2R \cdot \arcsin \left(\sqrt{\sin^2 \left(\frac{\Delta\phi}{2} \right) + \cos(\phi_1) \cos(\phi_2) \sin^2 \left(\frac{\Delta\lambda}{2} \right)} \right)$$

Here, ϕ_1 and ϕ_2 represent the latitudes of the two points (in radians), $\Delta\phi$ and $\Delta\lambda$ represent the differences in latitude and longitude (also in radians), and R is the radius of the Earth (using whatever unit of distance you desire). The result, d , is the distance between the two points along the surface of the sphere. Notice that $d = Rc$ is the formula for the *arc length* of a circle: the shortest path between two points on a sphere runs along a *great circle*, a circle whose center is the center of the sphere (like the equator), and c is the angle between the two points as seen from that center.

Planes follow these *great-circle routes* to save distance and fuel. On a flat map like the Mercator projection, a great-circle route looks like a curve bending toward the pole: a flight from New York to Tokyo passes over Alaska. The curve is an illusion caused by flattening a sphere; on a globe, it is the straightest possible path.

Worked example. How far is it from New York (40.7128° , -74.0061°) to London (51.5072° , -0.1275°)? Use $R = 6,371$ km.

Solution. In Python:

```

1 import math
2
3 def haversine(lat1, lon1, lat2, lon2, R=6371):
4     phi1, phi2 = math.radians(lat1), math.radians(lat2)
5     dphi = phi2 - phi1
6     dlam = math.radians(lon2 - lon1)
7     a = (math.sin(dphi / 2) ** 2
8         + math.cos(phi1) * math.cos(phi2) * math.sin(dlam / 2) ** 2)
9     return 2 * R * math.asin(math.sqrt(a))
10
11 print(haversine(40.7128, -74.0061, 51.5072, -0.1275))    # about 5570

```

The great-circle distance is about 5,570 km, or about 3,008 nautical miles.

Exercise 1 Sydney to Rio

Use the coordinates in this chapter (Sydney $33^\circ 52'$ S, $151^\circ 12'$ E; Rio de Janeiro

Working Space

22°54' S, 43°12' W) to find the great-circle distance between the two cities. Remember that south and west are negative.

Answer on Page 57

1.2 Geocoding and Reverse Geocoding

Geocoding and reverse geocoding are essential processes in geographic information systems (GIS) that are used to convert between addresses and spatial data.

1.2.1 Geocoding

Geocoding is the process of converting addresses (like "1600 Amphitheatre Parkway, Mountain View, CA") into geographic coordinates (like latitude 37.423021 and longitude -122.083739), which you can use to place markers on a map, or position the map. The resulting latitude and longitude are often used as a key index in merging datasets based on location.

Here is an example of using Google's Geocoding service to get the longitude and latitude of the Dallas County Administration Building:

```
1 import requests
2 import json
3
4 # Encode the parameters
5 parameters = {"address": "411 Elm St, Dallas, TX 75202", "key": "YOUR_API_KEY"}
6 base_url = "https://maps.googleapis.com/maps/api/geocode/json?"
7
8 # Send the GET request
9 response = requests.get(base_url, params=parameters)
10
11 # Convert the response to json
12 data = response.json()
13
14 # Extract the latitude and longitude
15 if len(data["results"]) > 0:
16     latitude = data["results"][0]["geometry"]["location"]["lat"]
17     longitude = data["results"][0]["geometry"]["location"]["lng"]
18
```

```
19     print(latitude, longitude)
20 else:
21     print("Could not find the latitude and longitude.")
22
```

1.2.2 Reverse Geocoding

Reverse geocoding, as the name implies, is the opposite process of geocoding. It involves converting geographic coordinates into a human-readable address. This can be useful in applications where you need to display an actual address to a user instead of latitude and longitude coordinates.

Here is an example of using Google's reverse geocoding API² to find the address at latitude = 33.9474096, longitude = -118.1179069

```
1 import requests
2 import json
3
4 api_key = "YOUR_API_KEY"
5 latitude = 33.9474096
6 longitude = -118.1179069
7
8     # Encode the parameters
9 parameters = {"latlng": f"{latitude},{longitude}", "key": api_key}
10 base_url = "https://maps.googleapis.com/maps/api/geocode/json?"
11
12 # Send the GET request
13 response = requests.get(base_url, params=parameters)
14
15 # Convert the response to json
16 data = response.json()
17
18 # Extract the address
19 if len(data["results"]) > 0:
20     address = data["results"][0]["formatted_address"]
21     print(address)
22 else:
23     print("Could not find the address")
24
```

²Note that the API key is something you obtain on your own. People do not share API keys for privacy and security reasons.

1.3 Making a Map

Plotly is an open-source data visualization library for Python, R, and JavaScript. It allows for interactive plots, including geographical maps. In this brief example, we will learn how to create a simple annotated map using Plotly in Python.

To begin, you need to install Plotly. In Python, you can do this via pip:

```
1 pip install plotly
```

Once installed, you can create a map with annotations as follows:

```
1 import plotly.graph_objects as go
2
3 fig = go.Figure(data=go.Scattergeo(
4     lon = [-75.789],
5     lat = [45.4215],
6     text = ['Ottawa'],
7     mode = 'text',
8 ))
9
10 fig.update_layout(
11     title_text = 'Annotated Map with Plotly',
12     showlegend = False,
13     geo = dict(
14         scope = 'world',
15         projection_type = 'azimuthal equal area',
16         showland = True,
17         landcolor = 'rgb(243, 243, 243)',
18         countrycolor = 'rgb(204, 204, 204)',
19     ),
20 )
21
22 fig.show()
23
24
```

This code creates a world map and places a text annotation at the geographic coordinates for Ottawa. Then, the plotly is opened in your browser using a localhost IP address. The `go.Scattergeo` function is used to define the geographical scatter plot (i.e., the annotation), while the `update_layout` function is used to define the appearance and the properties of the map itself.

In this example, you can replace the latitude, longitude, and text with the values corresponding to your desired location.

Discrete Probability and Random Variables

2.1 Introduction to Discrete Probability

In a game of Monopoly, why do players land on the orange properties so often? How many different ways can your class line up for a photo? What are the odds that your phone's 4-digit passcode is guessed on the first try? All of these are counting problems in disguise, and this chapter gives you the tools to solve them.

Before we begin talking about discrete probability, let us address the word *discrete*. It means "individually separate and distinct" as in "You can think of light as a continuous wave or as a blast of discrete particles."

When we are talking about probabilities, some problems deal with discrete quantities like "What is the probability that I will throw these three dice and the numbers that roll face up sum to 9?" There are also problems that deal with continuous properties like "What is the probability that the next bird to fly over my house will weigh between 97.2 and 98.1 grams?" In this module, we are going to focus on the probability problems that deal with discrete quantities.

Watch Khan Academy's Introduction to Probability at <https://youtu.be/uzkc-qNV0Ok>.

Let us say that you have a cloth sack filled with 100 marbles; 99 are red and 1 is white. If you reach in without looking and pull out one marble, you will probably pull out a red one. We say that "There is a 1 in 100 chance that you would pull out a white marble." Or we can use percentages and say "There is a 1% chance that you will pull out a white marble." Or we can use decimals and say "There is a 0.01 probability that you will pull out a white marble." In probability, we often talk about the probability of certain events. "Pulling out a white marble" is an event, and we can give it a symbol like W . In equations, we use p to mean "the probability of". Thus, we can say "There is a 0.01 probability that you will pull out a white marble," which becomes the equation

$$p(W) = 0.01$$

2.1.1 The Probability of All Possibilities is 1.0

We know that you are either going to pull out a red marble or a white marble, so the probability of a white marble being pulled and the probability of a red marble being pulled must add up to 100%. Therefore, the probability of pulling out a red marble must be 99%, or 0.99. If we let the event “Pull out a red marble” be given by the symbol R , we can say:

$$p(R) = 1.0 - p(W) = 1.0 - 0.01 = 0.99$$

Now, let us say that you take a marble from the bag, then toss a coin. What is the probability that you will pull a white marble, then get heads on the coin? It is the product of the two probabilities: $0.01 \times 0.5 = 0.005$, so one-half of a one percent chance. Do the probabilities still sum to 1?

- White and Heads = $0.01 \times 0.5 = 0.005$
- White and Tails = $0.01 \times 0.5 = 0.005$
- Red and Heads = $0.99 \times 0.5 = 0.495$
- Red and Tails = $0.99 \times 0.5 = 0.495$

$0.005 + 0.005 + 0.495 + 0.495 = 1.000$ Yes, the probabilities of all the possibilities still add to 1. We were allowed to multiply because *the events do not affect each other*: the events are *independent*.

2.1.2 Independence and Dependence

In the last section, you learned that the probability of two events (“Pulling a red marble from the bag” and “Getting tails in a coin toss”) is the product of the probability of each event: $0.99 \times 0.5 = 0.495$.

This is true if the two events are *independent*; that is, the outcome of one does not change the probability of the other. The example we gave is independent: It does not matter what ball you pull from the bag, the outcome of the coin toss will always be 50-50.

What is an example of two events that are not independent? The probability that a person is a professional basketball player and the probability that someone wears a shoe that is size 13 or larger is *not* independent; rather it is dependent. After all, height is an advantage in basketball, and most tall people also have large feet. So, if you know someone is a basketball player, they likely wear large shoes.

2.1.3 Correlation does not Imply Causation

It is often said that “Correlation does not imply causation”. What does this mean? Breaking it down:

- Correlation is a state in which two things increase or decrease together; two variables are *linearly* related
- Imply means suggests, as in *there is evidence for*
- Causation means there is a cause and effect relationship in place between the two variables.

So saying correlation does not imply causation simply means: just because two variables are linearly connected, it cannot be said one variable causes the other.

Here are some examples:

- Ice Cream Sales, Mosquito bites, and Sunburn cases all increase during the summertime. However, none of them causes another: hot, sunny weather drives all three.
- Dinosaurs were not able to read books. However, this does not mean that the inability to read caused dinosaur extinction.
- There was a large increase in toilet paper sales and people working from home in the years 2020-2021. Does this mean that one causes the other? Not necessarily. In this case a third factor C , the pandemic, caused a large global quarantine, leading to increased toilet paper usage and the inability to work in an office.

Exercise 2 Rolling Dice

If you have three six-sided dice to roll, what is the probability that you will roll a five on all three dice?

Working Space

Answer on Page 57

Exercise 3 Flipping Coins

If you have five coins to flip, what is the probability that at least one coin will

Working Space

come up heads?

Answer on Page 57

2.1.4 Why 7 is the most likely sum of two dice

If you roll two dice, the sum will be 2, 12, or any number in between. It is very tempting to assume that the likelihood of any of those numbers is the same. In fact, the probability of a 2 is $\frac{1}{36} \approx 3\%$ and the probability of a 7 is $\frac{1}{6} \approx 17\%$. A 7 is six times more likely than a 12! Why?

When you roll the first die, there are six possibilities with equal probability. When you roll the second die, there are six possibilities with equal probability. So, there are a total of 36 possible events with equal probabilities: 1 then 1, 1 then 2, 2 then 1, 1 then 3, 3 then 1, and so on. Only one of these (1 then 1) adds to 2. However, six of these sum to 7: 1 then 6, 6 then 1, 2 then 5, 5 then 2, 3 then 4, and 4 then 3. So, a 7 is six times more likely than a 2.

Here is the complete table:

Sum		Count	Probability
2	1,1	1	1/36
3	1,2 2,1	2	1/18
4	1,3 2,2 3,1	3	1/12
5	1,4 2,3 3,2 4,1	4	1/9
6	1,5 2,4 3,3 4,2 5,1	5	5/36
7	1,6 2,5 3,4 4,3 5,2 6,1	6	1/6
8	2,6 3,5 4,4 5,3 6,2	5	5/36
9	3,6 4,5 5,4 6,3	4	1/9
10	4,6 5,5 6,4	3	1/12
11	5,6 6,5	2	1/18
12	6,6	1	1/36

If you do not believe these numbers, you could roll a pair of dice hundreds of times and make a histogram. However, it would be a tedious and time-consuming task — just the sort of thing that we make computers do for us.

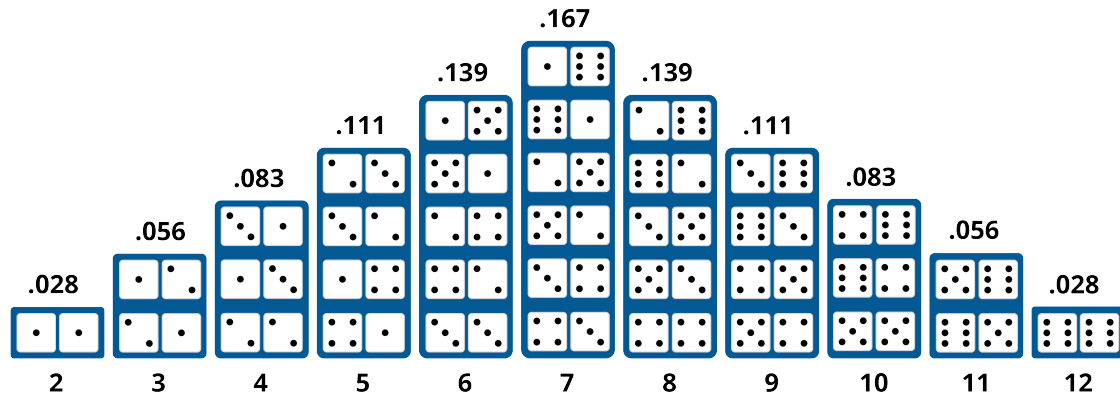


Figure 2.1: A histogram of probabilities of dice sums. With two dice, the shape is a triangle; with more dice, it gets closer and closer to a bell curve.

2.1.5 Random Numbers and Python

You are going to write a simulation of rolling dice in Python. To do this, you will need to generate a random sequence of numbers. The numbers will need to be in the range 1 to 6, and they will need to appear in the sequence with the same frequency. We say the sequence will follow *the uniform distribution*. In other words, the probability is uniformly distributed among the 6 possibilities.

Start python and try a few of the different ways to generate random numbers:

```

1 > python3
2 >>> import random
3 >>> random.random() # Generates a random floating point number between 0 and 1
4 0.6840892758539989
5 >>> random.randrange(5) # Generates an integer in the range 0 - 4
6 2
7 >>> x = ['Rock', 'Paper', 'Scissors']
8 >>> random.choice(x) # Pick a random entry from the sequence
9 'Paper'
10 >>> x
11 ['Rock', 'Paper', 'Scissors']
12 >>> random.shuffle(x) # Shuffle the order of the sequence
13 >>> x
14 ['Scissors', 'Paper', 'Rock']
15 >>> a = list(range(30))

```

```

16 >>> a
17 [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
18  16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29]
19 >>> random.sample(a, 10) # Return 10 randomly chosen items from the sequence
20 [8, 7, 20, 9, 25, 13, 23, 11, 14, 16]

```

Clearly, Python has many ways to do things that look random. However, here is the hidden truth: they are not really random. The computer that you are using cannot generate random data. Instead, it uses tricks to create data that look random; we call this *pseudo-random* data. Good pseudorandom algorithms are very important for cryptography and data security.

What if you want real random data? Some companies are using the decay of radioactive materials to generate real random data. You can pay to download it. For our purposes, Python's pseudorandom numbers are quite sufficient.

If we generate two random numbers in the range 1 through 6 and add them together, we will have simulated rolling a pair of dice. Like this:

```

1 >>> a = random.randrange(6) + 1
2 >>> b = random.randrange(6) + 1
3 >>> a + b
4 8

```

First, let us write a program that just rolls the dice 100 times and shows the result. Make a file `dice.py`:

```

1 import random
2
3 roll_count = 100
4
5 for i in range(roll_count):
6     a = random.randrange(6) + 1
7     b = random.randrange(6) + 1
8     roll = a + b
9     print(f"Toss {i}: {a} + {b} = {roll}")

```

When you run it, you should see something like:

```

1 > python3 dice.py
2 Toss 0: 6 + 6 = 12
3 Toss 1: 4 + 4 = 8
4 Toss 2: 4 + 2 = 6
5 Toss 3: 4 + 6 = 10
6 Toss 4: 4 + 4 = 8

```

```

7 ...
8 Toss 98: 5 + 2 = 7
9 Toss 99: 5 + 2 = 7

```

Now we want to count occurrences of each possible outcome. Let us use an array of integers. We will start with an array of zeros. When we roll a 3, we will add 1 to item 3 in the array. (We can never roll a zero or a one, so those two entries will always be zero.)

```

1 import random
2
3 roll_count = 100
4
5 # Make an array containing 13 zeros
6 counts = [0] * 13
7
8 for i in range(roll_count):
9     a = random.randrange(6) + 1
10    b = random.randrange(6) + 1
11    roll = a + b
12    print(f"Toss {i}: {a} + {b} = {roll}")
13
14    # Increment the count for roll
15    counts[roll] += 1
16
17 print(f"Counts: {counts}")

```

When you run this, at the end you will see a count for each possible outcome:

```

1 ...
2 Toss 98: 3 + 2 = 5
3 Toss 99: 6 + 1 = 7
4 Counts: [0, 0, 2, 6, 16, 11, 13, 14, 11, 11, 6, 9, 1]

```

What was the count that we expected? For example, we expected to see a 2 about once every 36 rolls, right? It might be nice to compare our count to what we expected. Add a few more lines, and we are going to increase the number of rolls. You will probably want to delete the line that prints each roll separately:

```

1 import random
2
3 # Can't ever be 0 or 1
4 p = [0.0, 0.0, 1/36, 1/18, 1/12, 1/9, 5/36, 1/6, 5/36, 1/9, 1/12, 1/18, 1/36]
5 roll_count = 1000
6
7 # Make an array containing 13 zeros
8 counts = [0] * 13

```

```
9
10 for i in range(roll_count):
11     a = random.randrange(6) + 1
12     b = random.randrange(6) + 1
13     roll = a + b
14
15     # Increment the count for roll
16     counts[roll] += 1
17
18 for i in range(2,13):
19     print(f"{i} appeared {counts[i]} times, expected {p[i] * roll_count:.1f}")
```

Now you should see something like:

```
1 2 appeared 39 times, expected 27.8
2 3 appeared 55 times, expected 55.6
3 4 appeared 84 times, expected 83.3
4 5 appeared 110 times, expected 111.1
5 6 appeared 160 times, expected 138.9
6 7 appeared 176 times, expected 166.7
7 8 appeared 124 times, expected 138.9
8 9 appeared 93 times, expected 111.1
9 10 appeared 87 times, expected 83.3
10 11 appeared 49 times, expected 55.6
11 12 appeared 23 times, expected 27.8
```

Whenever you are dealing with random numbers, the outcome will seldom be *exactly* what you expected. In this case, however, you should see that your predictions are pretty close.

Making a bar graph

A bar graph is a nice way to look at quantities like this. Let us make a bar graph that shows the actual count and the expected count:

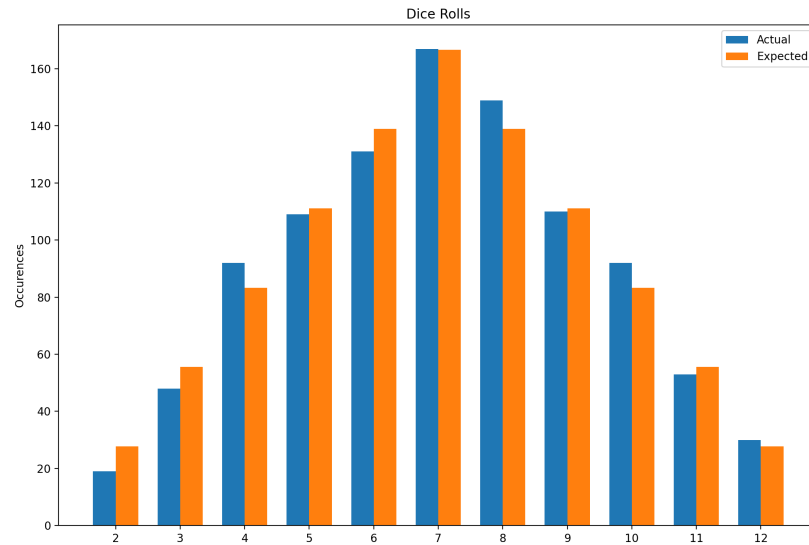


Figure 2.2: Dice roll histogram.

We need to describe the set of rectangles. To do this we will loop through each possible roll (2 - 12) and put data in four lists for each:

```

1  import random
2  import matplotlib.pyplot as plt
3
4  # Can't ever be 0 or 1
5  p = [0.0, 0.0, 1/36, 1/18, 1/12, 1/9, 5/36, 1/6, 5/36, 1/9, 1/12, 1/18, 1/36]
6  roll_count = 1000
7
8  # Make an array containing 13 zeros
9  counts = [0] * 13
10
11  for i in range(roll_count):
12      a = random.randrange(6) + 1
13      b = random.randrange(6) + 1
14      roll = a + b
15
16      # Increment the count for roll
17      counts[roll] += 1
18
19  # Gather data for bar chart
20  bar_width = 0.35
21  expected = []
22  actual_starts = []
23  expected_starts = []
24  labels = []

```

```

25 actual = []
26 for i in range(2,13):
27     expected.append(p[i] * roll_count)
28     actual.append(counts[i])
29     actual_starts.append(i - bar_width/2)
30     expected_starts.append(i + bar_width/2)
31     labels.append(i)
32
33 fig, ax = plt.subplots()
34
35 # Create the bars
36 ax.bar(actual_starts, actual, bar_width, label='Actual')
37 ax.bar(expected_starts, expected, bar_width, label='Expected')
38 ax.set_xticks(labels)
39
40 # Provide labels
41 ax.set_ylabel('Occurrences')
42 ax.set_title('Dice Rolls')
43 ax.legend()
44 plt.show()

```

2.2 What Is a Random Variable?

Casinos never lose money in the long run, even though individual gamblers sometimes win big. The secret is a single number, the *expected value*, which predicts the average result of a game played many times. The rest of this chapter shows how to compute it.

A *random variable* assigns a number to each outcome of a random experiment. Rolling a die gives an outcome, and the random variable X is the number that shows. Flipping a coin twice has four outcomes, and the random variable X can be the number of heads.

A *discrete* random variable takes a countable set of values. Its *probability mass function* (PMF) lists each value and its probability. For two coin flips:

x (number of heads)	$P(X = x)$
0	$\frac{1}{4}$
1	$\frac{1}{2}$
2	$\frac{1}{4}$

The probabilities must add up to 1.

2.3 Expected Value

The *expected value* of X is the average value we would get over many repetitions. For a discrete random variable,

$$E[X] = \sum_x x \cdot P(X = x)$$

For the coin example, $E[X] = 0 \cdot \frac{1}{4} + 1 \cdot \frac{1}{2} + 2 \cdot \frac{1}{4} = 1$.

The expected value does not have to be a value the variable can take. A fair die has $E[X] = \frac{7}{2}$, even though no roll gives 3.5.

2.4 Variance and Standard Deviation

The *variance* measures how far X typically is from its mean $\mu = E[X]$:

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

The standard deviation is $\sigma = \sqrt{\text{Var}(X)}$.

For a fair die, $E[X^2] = \frac{1+4+9+16+25+36}{6} = \frac{91}{6}$, so

$$\text{Var}(X) = \frac{91}{6} - \left(\frac{7}{2}\right)^2 = \frac{35}{12}$$

2.5 Linearity of Expectation

Expected value is linear. For constants a and b :

$$E[aX + b] = aE[X] + b \quad \text{Var}(aX + b) = a^2 \text{Var}(X)$$

Adding a constant shifts the mean but does not change the spread.

If X and Y are independent, their variances add: $\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y)$. For two dice, the sum S has $E[S] = 7$ and $\text{Var}(S) = 2 \cdot \frac{35}{12} = \frac{35}{6}$.

2.6 A Game Example

A game costs \$10 to enter. You win \$30 with probability $\frac{1}{4}$ and win nothing otherwise. Let W be your net winnings. Then $W = 30 - 10 = 20$ with probability $\frac{1}{4}$ and $W = -10$ with probability $\frac{3}{4}$. So

$$E[W] = 20 \cdot \frac{1}{4} + (-10) \cdot \frac{3}{4} = -2.5$$

On average, you lose \$2.50 per game.

Exercise 4 Expected value of a die

Let X be the number on a fair six-sided die. Find $E[2X + 1]$.

Working Space

Answer on Page 57

Exercise 5 Variance after scaling

Using the die from the previous exercise, find $\text{Var}(2X + 1)$.

Working Space

Answer on Page 57

Exercise 6 Two coin flips

Use the PMF table above to check that $E[X] = 1$ and $\text{Var}(X) = \frac{1}{2}$.

Working Space

Answer on Page 58

Exercise 7 Is the game worth playing?

A game costs \$5 to enter. You win \$20 with probability $\frac{1}{5}$ and nothing otherwise. What is the expected net

Working Space

winnings? Should you play?

Answer on Page 58

2.7 Looking Ahead

Everything in this chapter used *discrete* random variables, where we add up $x \cdot P(X = x)$. Many quantities, like waiting times and heights, are *continuous*. For those, the sum becomes an integral, $\mu = \int xf(x) dx$. You will compute such means after you learn integration by parts (in the Integration by Parts chapter), and expected values of two random variables after double integrals.

Counting

How many different ways can you arrange the songs on a 12-song playlist? How many five-card poker hands are there? How many ways can a team of 4 be chosen from a class of 30? Counting sounds like the easiest thing in math, until the numbers get big. In the Discrete Probability and Random Variables chapter, you found probabilities by counting outcomes one at a time. This chapter gives you the shortcuts: factorials, permutations, combinations, and the binomial theorem.

3.1 Beginning Combinatorics

Discrete probability problems often include some counting. For example, we figured out that there were 36 different ways the two dice could land, but all of them summed to some number 2 through 12. How many different ways could three eight-sided dice come up? We would need to count them, right? As the numbers get bigger, we will need some tricks so that we do not need to write them all down and count them one-by-one.

The branch of mathematics that focuses on tricks for counting is called *combinatorics*.

How can we be sure that there were 36 different configurations for the two 6-sided dice? The first die could have come up as any one of six numbers. For each of those, the second could have come up with any one of six numbers. Thus, the number of possibilities is $6 \times 6 = 36$.

How many different configurations for three 8-sided dice? $8 \times 8 \times 8 = 8^3 = 512$.

What about seven dice, each with 20 sides? There would be $20^7 = 1,280,000,000$ configurations. See, are you glad that we do not need to write them all down?

Now, let us say that six people (Anne, Brock, Carl, Dev, Edgar, and Fred) are going to run a race. You have to make a plaque that says who won first place, who won second place, and who won third. If you want to get all the possible plaques created beforehand, and just pull the right one out as soon as the race ends, how many plaques would you need to get engraved?

Since you are making these plaques before the race even happens, you need one plaque for every possible outcome. So think of it as filling in three blanks: first place, second place, and third place. Any of the 6 people could fill the first blank. Whoever fills that blank cannot also fill the second blank — a person cannot come in both first *and* second — so only

5 people are left to fill the second blank. Likewise, only 4 people remain to fill the third blank, since two people have already been assigned to the first two spots. Multiplying these together, $6 \times 5 \times 4 = 120$, tells you exactly how many different first-second-third outcomes are possible — and thus how many plaques you would need engraved.

What if the plaque includes all 6 places? Using the same reasoning, you would fill in six blanks instead of three: $6 \times 5 \times 4 \times 3 \times 2 \times 1 = 720$ plaques engraved. We use this process often enough that we gave it a name. We say “I need 6 factorial plaques engraved.” When we write a *factorial*, we use an exclamation point:

$$6! = 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 720$$

We use the word “permutation” to mean a particular ordering. This rule says n items can be ordered in $n!$ ways. Thus, mathematicians actually say “If you have a list of n items then we can generate $n!$ different permutations of those items.” We will return to permutations at the end of this chapter.

In Python, there is a factorial function in the math library:

```
1 >>> import math
2 >>> math.factorial(6)
3 720
```

Handy, right? Now you do not need to write a loop to calculate factorials.

Remember when we only wanted the first three names on the plaque? We can solve that problem using factorials:

$$6 \times 5 \times 4 = \frac{6 \times 5 \times 4 \times 3 \times 2 \times 1}{3 \times 2 \times 1} = \frac{6!}{3!}$$

This formulation makes it easy to figure out on any calculator with a “!” button.

The rule on this is to fill m positions from n items. It can be done this many ways:

Permutations

For permutations where order is a factor, the formula for ${}^n P_m$ (n objects pick m) is

$$\frac{n!}{(n - m)!}$$

3.1.1 Choose

Let us say that there are 12 kids in a classroom and you need a team of four to clean the top of the desks. How many different possible teams are there? You know that if you were giving out four different positions (Like the race gave out 1st, 2nd, and 3rd), the answer would be $12 \times 11 \times 10 \times 9$ or $12!/(12-4)!$.

However, once we pick the 4 people, we do not care what order they are in, right? In this problem, the team “Anne, Brad, Carl, and Don” is the same as the team “Carl, Don, Brad, and Anne”.

Thus, the quantity $12!/(12-4)!$ is many times too large, because it counts each permutation separately. To get the right number, we just divide this by the number of possible permutations for a group of four people: $4!$

That gets us our answer: How many different teams of four can be chosen from 12 people?

$$\frac{12!}{(12-4)!4!} = 495$$

This number is called a *binomial coefficient*. In combinatorics, we use this quantity a lot, so we have given it a name: *choose*

Combinations

For combinations, where order does not matter, the formula for ${}^nC_k$ (n objects choose k) is

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

We have also given it a notation. “12 choose 4” is written like this:

$$\binom{12}{4}$$

Binomial coefficients are the key to the binomial distribution, which counts successes in repeated trials. The Binomial and Geometric Distributions chapter puts them to work.

Python has the `math.comb` function:

```
1 >>> import math
2 >>> math.comb(12, 4)
3 495
```

We briefly saw this in the common polynomial products chapter. The binomial theorem says that the coefficients of the terms in the expansion of $(a+b)^n$ are given by the binomial coefficients $\binom{n}{k}$.

$$(a+b)^4 = 1a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + 1b^4 = \binom{4}{0}a^4 + \binom{4}{1}a^3b + \binom{4}{2}a^2b^2 + \binom{4}{3}ab^3 + \binom{4}{4}b^4$$

3.1.2 What About Fractional Exponents?

What is $\sqrt{1+5x} = (1+5x)^{1/2}$? Pascal's triangle only works when the exponent is a non-negative integer. The formula $\binom{n}{k} = \frac{n(n-1)(n-2)\dots(n-k+1)}{k!}$ only multiplies and divides, so it makes sense even when n is $\frac{1}{2}$ or -1 . But then none of the factors $n, n-1, n-2, \dots$ is ever zero, so the expansion never stops: it becomes an infinite sum, called the binomial series. Infinite sums need some care, and the Taylor Series chapter shows how to use this one safely.

3.1.3 Applications in Probability

Suppose you flip a fair coin 5 times. What is the probability that you get exactly 3 heads?

Note that there are only two options: Heads (H) or Tails (T). Thus, there are $2^5 = 32$ possible sequences of flips. Each sequence is equally likely, so the probability of any particular sequence is $\frac{1}{32}$. Only a certain number of sequences will have exactly 3 heads.

One way to get 3 heads is HHHTT. There is also HTHHT, THHHT, and many other orderings also give you 3 heads. Before we can find the probability, we need to know *how many* different sequences of 5 coin flips contain exactly 3 heads. As we have discussed, we can use the n -choose- k formula. There are

$$\binom{5}{3} = \frac{5!}{3!2!} = 10$$

such sequences. This is said "5 choose 3", or, more explicitly "get 3 successes in 5 trials".

Each individual sequence, like HHHTT, has probability $\left(\frac{1}{2}\right)^3 \left(\frac{1}{2}\right)^2$ of occurring, since each flip is independent. Since there are 10 equally likely sequences that give exactly 3 heads, we add up 10 copies of that probability:

$$P(\text{exactly 3 heads}) = \binom{5}{3} \left(\frac{1}{2}\right)^3 \left(\frac{1}{2}\right)^2 = 10 \cdot \frac{1}{32} = \frac{10}{32} = \frac{5}{16}$$

This pattern of counting the arrangements with $\binom{n}{k}$, then multiplying by the probability of one arrangement works whenever you have repeated *independent* trials with only two possible outcomes each time (success or failure).

Exercise 8 Free Throws

A basketball player has a past record of making 60% of their free throws. Over the next 6 free throws, what is the probability that she makes exactly 4 of them?

Working Space

Answer on Page 58

This probability situation has been rightfully called the *binomial distribution*. This is a brief introduction to the binomial distribution, and we will discuss it in more detail in a later chapter.

3.2 Permutations and Sorting

We counted permutations earlier in this chapter. If you have a list of four letters, like [a, b, c, d], you can rearrange them in 4! ways:

a,b,c,d	a,b,d,c	a, d, b, c	a, d, c, b	a, c, b, d	a, c, d, b
b,a,c,d	b,a,d,c	b, d, a, c	b, d, c, a	b, c, a, d	b, c, d, a
c,b,a,d	c,b,d,a	c, d, b, a	c, d, a, b	c, a, b, d	c, a, d, b
d,b,c,a	d,b,a,c	d, a, b, c	d, a, c, b	d, c, b, a	d, c, a, b

You can make Python generate all the permutations for you:

```
1 from itertools import permutations
2 all_permutations = permutations(('a', 'b', 'c', 'd'))
3 for p in all_permutations:
4     print(p)
```

3.2.1 Notation

How do we define or write down a single permutation? You could say something like “Swap the first and second items and swap the third and fourth items.” However, that

gets pretty difficult to read, so we usually write a permutation as two lines: the first line is before the permutation and the second line is after. Like this:

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{pmatrix}$$

We can also assign permutations to variables. For example, if we wanted the variable A to represent “swapping the first and second item”, we would write this:

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 3 & 4 \end{pmatrix}$$

And if we wanted B to represent “swapping the third and fourth item”, we would write:

$$B = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 4 & 3 \end{pmatrix}$$

Now, we can *compose* permutations together. For example, we might say:

$$B \circ A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{pmatrix}$$

In other words, if we have the list $[a, b, c, d]$ and we apply permutation A , followed by permutation B , we get $[b, a, d, c]$.

Important: Note that permutations are applied from right to left. $B \circ A$ means “Applying A and then B .” Why does this matter? Permutations are not necessarily commutative. That is, if you have two permutations S and T , $S \circ T$ is not always the same as $T \circ S$.

Also, note that “do not change anything” is a permutation. We call it *the identity permutation*. If you have four items, the identity permutation would be written:

$$I = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix}$$

(We use a capital “ I ” for the identity.)

Challenge

Find an example of two permutations S and T , such that $S \circ T$ does not equal $T \circ S$.

Hint: on a list of three items, try swapping the first two items and swapping the last two items.

(One answer: let S swap positions 1 and 2, and T swap positions 2 and 3. Starting from $[a, b, c]$, $S \circ T$ gives $[c, a, b]$, but $T \circ S$ gives $[b, c, a]$.)

3.2.2 Sorting in Python

One of the common forms of permutation in software is sorting. Sorting is putting data in a particular order. For example, in Python, if you had a list of numbers, you can sort it in ascending order like this:

```

1 my_grades = [92, 87, 76, 99, 91, 93]
2 grades_worst_to_best = sorted(my_grades)

```

Do you want to sort backwards?

```

1 my_grades = [92, 87, 76, 99, 91, 93]
2 grades_best_to_worst = sorted(my_grades, reverse=True)

```

Note that `sorted` makes a new list with the correct order. If you want to sort the array in place, you can use the `sort` method:

```

1 my_grades = [92, 87, 76, 99, 91, 93]
2 my_grades.sort(reverse=True)

```

3.2.3 Inverses

Think for a second about this permutation:

$$S = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 2 & 1 \end{pmatrix}$$

You could say this permutation shuffles a list. What is its inverse? That is, what is the permutation that unshuffles the items back to where they were originally?

$$S^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 1 & 2 \end{pmatrix}$$

In other words, the original moved an item in the first spot to the third spot; the inverse must move whatever was in the third spot back to the first spot.

(Notation note: In multiplication, $b \times b^{-1} = 1$, so we use “to the negative one” to indicate inverses in lots of places.)

Mechanically, how do you find the inverse? Flip the rows, then sort the columns using the top number:

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 2 & 1 \end{pmatrix} \text{ flip } \rightarrow \begin{pmatrix} 3 & 4 & 2 & 1 \\ 1 & 2 & 3 & 4 \end{pmatrix} \text{ sort } \rightarrow \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 1 & 2 \end{pmatrix}$$

Let us say you have two permutations A and B. Permuting by B then A would look like this:

$$C = A \circ B$$

If you know A^{-1} and B^{-1} , what is C^{-1} ? You would undo-A, then undo-B, so

$$C^{-1} = B^{-1} \circ A^{-1}$$

3.2.4 Cycles

Here is a permutation:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 4 & 5 & 1 & 3 \end{pmatrix}$$

When this is applied, whatever is at 1 gets moved to 2, 2 gets moved to 4, and 4 gets moved to 1. That is a *cycle*: $1 \rightarrow 2 \rightarrow 4$, then it goes back to 1. It involves three locations, so we say it is a *3-cycle*.

There is another cycle in this permutation: $3 \rightarrow 5$, then it goes back to 3.

Because these cycles share no members, we say the cycles are *disjoint*.

Every permutation can be broken down into a collection of disjoint cycles.

$$T = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 4 & 5 & 1 & 3 \end{pmatrix} = (1 \rightarrow 2 \rightarrow 4)(3 \rightarrow 5)$$

The first handy thing about this notation is that it makes it easy for us to describe the inverse. We just run the cycles backward:

$$T^{-1} = (4 \rightarrow 2 \rightarrow 1)(5 \rightarrow 3)$$

Starting with the list $[a, b, c, d, e]$, let us repeatedly apply the permutation T

Initial	a, b, c, d, e
T applied	d, a, e, b, c
$T \circ T$ applied	b, d, c, a, e
$T \circ T \circ T$ applied	a, b, e, d, c
$T \circ T \circ T \circ T$ applied	d, a, c, b, e
$T \circ T \circ T \circ T \circ T$ applied	b, d, e, a, c
$T \circ T \circ T \circ T \circ T \circ T$ applied	a, b, c, d, e

After six applications, the list is back where it started. The number of applications needed, called the *order* of the permutation, is the least common multiple of the cycle lengths. In this case, there is a 3-cycle and a 2-cycle. The least common multiple of 2 and 3 is 6.

3.2.5 Further Learning

Check out this leetcode problem which combines permutations and programming! <https://leetcode.com/problems/permutations/description/>

Binomial and Geometric Distributions

Your school's best free-throw shooter makes 70% of their shots. In tonight's game, they will probably shoot about 10. What are the chances they make exactly 7? At least 8? And a different question: a pack of trading cards has a 1-in-8 chance of holding a rare card. How many packs will you probably have to open before you get one?

The first question counts *successes in a fixed number of tries*. The second counts *tries until the first success*. These two patterns show up everywhere: free throws and penalty kicks, defective parts on an assembly line, patients who respond to a medicine, days until a rainy day. Each has its own probability distribution, and both are on the AP Statistics exam.

4.1 The Binomial Setting

A random variable X has a *binomial distribution* when it counts the number of successes in a situation that meets four conditions. A handy way to remember them is **BINS**:

- **Binary**: each trial has two outcomes, success or failure.
- **Independent**: the result of one trial does not change the chances for another.
- **Number**: the number of trials, n , is fixed in advance.
- **Same probability**: every trial has the same chance of success, p .

Ten free throws by a 70% shooter fit: each shot goes in or not, one shot does not affect the next (we assume), there are 10 shots, and each has $p = 0.7$. So the number made, X , is binomial with $n = 10$ and $p = 0.7$.

4.1.1 The Binomial Formula

What is the chance of making exactly 7 of the 10? One way is

make, make, make, make, make, make, make, miss, miss, miss

By independence, its probability is $0.7^7 \cdot 0.3^3$. But the 3 misses could come at any 3 of the 10 shots, and every arrangement has the same probability. The number of arrangements is the number of ways to choose which 7 shots go in, $\binom{10}{7} = 120$, from the Counting chapter. So

$$P(X = 7) = \binom{10}{7} (0.7)^7 (0.3)^3 = 120(0.0823543)(0.027) \approx 0.2668$$

The Binomial Distribution

If X counts the successes in n independent trials, each with success probability p , then for $k = 0, 1, \dots, n$,

$$P(X = k) = \binom{n}{k} p^k (1 - p)^{n-k}$$

Each piece has a job: p^k for the k successes, $(1 - p)^{n-k}$ for the $n - k$ failures, and $\binom{n}{k}$ for all the orders they could come in.

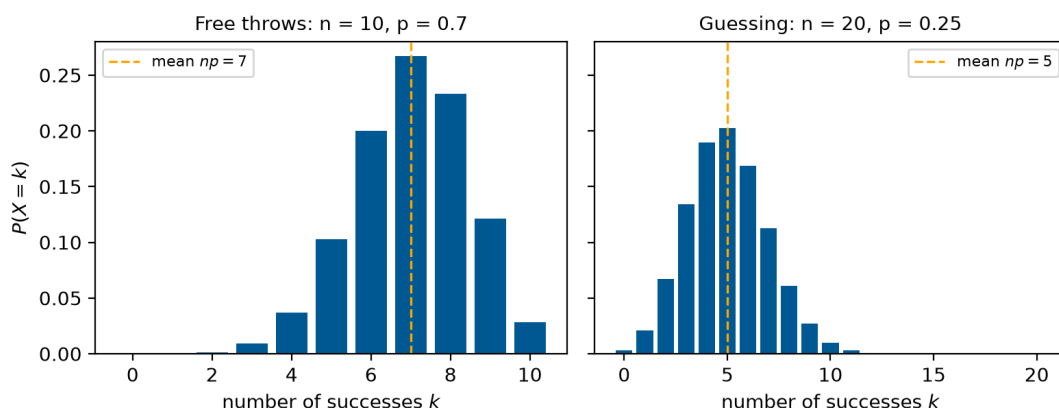


Figure 4.1: Two binomial distributions. Left: a 70% shooter taking 10 free throws. Right: guessing on 20 multiple-choice questions with 4 choices each.

4.1.2 “At Least” and “At Most”

Most real questions ask about a range. The chance of making at least 8 is the sum of three cases:

$$P(X \geq 8) = P(X = 8) + P(X = 9) + P(X = 10) \approx 0.2335 + 0.1211 + 0.0282 = 0.3828$$

When a range has many terms, use the **complement**. For example, the chance of making at least 1 shot is $1 - P(X = 0) = 1 - 0.3^{10}$, which is extremely close to 1. On AP-style problems, read the wording carefully: “at least 8” includes 8, while “more than 8” does not.

4.1.3 Mean and Standard Deviation

Here is a slick way to find the mean. Think of X as a sum of n simple random variables, one per trial: $X_i = 1$ if trial i succeeds and 0 if it fails. Each X_i has

$$E[X_i] = 1 \cdot p + 0 \cdot (1 - p) = p, \quad \text{Var}(X_i) = E[X_i^2] - p^2 = p - p^2 = p(1 - p)$$

By linearity of expectation (from the Discrete Probability and Random Variables chapter), and because variances of independent variables add,

Binomial Mean and Standard Deviation

$$\mu = np, \quad \sigma = \sqrt{np(1 - p)}$$

The 70% shooter taking 10 shots averages $\mu = 10(0.7) = 7$ makes, with standard deviation $\sqrt{10(0.7)(0.3)} = \sqrt{2.1} \approx 1.45$. A night with 5 makes is unremarkable; a night with 2 would be very unusual.

4.1.4 Shape

Look again at figure 4.1. When p is close to $\frac{1}{2}$, the distribution is nearly symmetric. When p is near 0 or 1, it piles up at one end and has a tail on the other side (the free-throw distribution is skewed left). As n grows, the shape becomes more and more bell-shaped. A common rule: if both $np \geq 10$ and $n(1 - p) \geq 10$, the binomial distribution is close to a normal distribution. You will use that fact constantly in the inference chapters of Volume 30.

4.1.5 Sampling Without Replacement: The 10% Condition

When you survey people, you never ask the same person twice, so the trials are not quite independent. If 15% of a school's 2,000 students are left-handed and you pick 25 at random, finding a lefty slightly lowers the chance that the next person is one. But the effect is tiny when the sample is small compared with the population. The *10% condition* says: if the sample is no more than 10% of the population, you can treat the count as binomial anyway. Here, 25 is far less than 200, so a binomial model is fine.

4.2 Binomial Probabilities in Python

Calculators have `binompdf` and `binomcdf` functions. In Python, the SciPy library's `stats` module has `binom` (and `geom`, for the next section). The `pmf` function gives $P(X = k)$, `cdf`

gives $P(X \leq k)$, and `sf` (for “survival function”) gives $P(X > k)$.

```

1 from scipy.stats import binom, geom
2
3 # Free throws: n = 10 shots, p = 0.7
4 print(round(binom.pmf(7, 10, 0.7), 4))      # P(X = 7)
5 print(round(binom.sf(7, 10, 0.7), 4))      # P(X > 7), that is, P(X >= 8)
6 print(binom.mean(10, 0.7), round(binom.std(10, 0.7), 3))
7
8 # Rare cards: p = 1/8 per pack
9 print(round(geom.pmf(3, 0.125), 4))        # first rare card in pack 3
10 print(round(geom.sf(10, 0.125), 4))       # P(X > 10): none in 10 packs
11 print(geom.mean(0.125), round(geom.std(0.125), 2))

```

```

0.2668
0.3828
7.0 1.449
0.0957
0.2631
8.0 7.48

```

You can also check a formula by simulation, as in the Monte Carlo Simulation chapter. Here are 100,000 nights of 10 free throws each:

```

1 import random
2
3 random.seed(10)
4
5 trials = 100_000
6 exactly_7 = 0
7 for _ in range(trials):
8     made = sum(random.random() < 0.7 for _ in range(10)) # 10 shots
9     if made == 7:
10         exactly_7 += 1
11 print(exactly_7 / trials)

```

```
0.26648
```

The simulation agrees with the formula’s 0.2668 to about three decimal places.

Exercise 9 Is It Binomial?

For each random variable, decide whether it has a binomial distribution. If not, say which BINS condition fails.

Working Space

1. The number of sixes in 10 rolls of a fair die.
2. The number of cards you draw from a shuffled deck until you get an ace.
3. The number of hearts in 5 cards dealt from a shuffled deck.
4. The number of days next week on which it rains, if rainy days tend to come in streaks.

Answer on Page 58

Exercise 10 **Lefties**

About 15% of people are left-handed. In a random group of 25 students:

1. What is the probability that exactly 3 are left-handed?
2. What is the probability that at least one is left-handed?
3. Find the mean and standard deviation of the number of lefties.

Working Space

Answer on Page 58

Exercise 11 **Just Guessing**

A quiz has 20 multiple-choice questions with 4 choices each. A student guesses on every question.

1. What are the mean and standard deviation of the number correct?

Working Space

2. Use Python to find the probability of getting 5 or fewer right, and the probability of getting at least 12 right (a passing grade).

Answer on Page 59

4.3 The Geometric Setting

Now change the question. Instead of fixing the number of trials and counting successes, keep going *until the first success*, and count the trials. If the trials are binary and independent with the same success probability p , the number of the trial on which the first success happens, X , has a *geometric distribution*.

For the first success to come on trial k , the first $k - 1$ trials must all fail, and then trial k must succeed. There is only one way for that to happen, so no binomial coefficient is needed:

The Geometric Distribution

If X is the trial on which the first success occurs, then for $k = 1, 2, 3, \dots$,

$$P(X = k) = (1 - p)^{k-1}p$$

and the chance of needing more than k trials is

$$P(X > k) = (1 - p)^k$$

The second formula holds because needing more than k trials means the first k all failed.

Example: Each pack of cards has a rare card with probability $p = \frac{1}{8}$.

- The chance that the first rare card is in the 3rd pack is $\left(\frac{7}{8}\right)^2 \frac{1}{8} \approx 0.0957$.
- The chance that you open 10 packs and still have no rare card is $\left(\frac{7}{8}\right)^{10} \approx 0.263$. More than a quarter of collectors will be that unlucky.

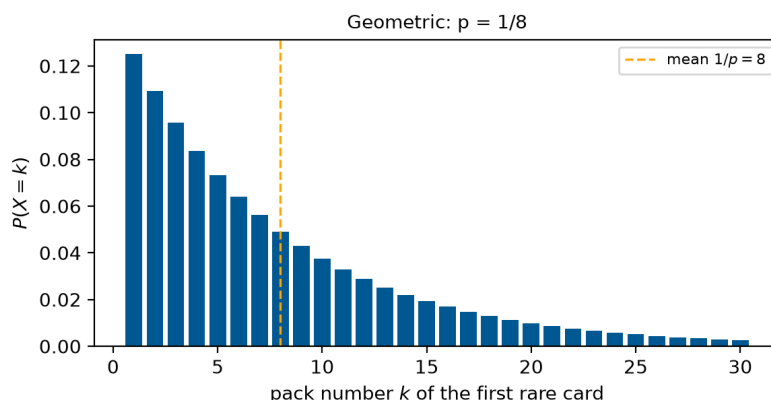


Figure 4.2: The geometric distribution with $p = \frac{1}{8}$. Each bar is $\frac{7}{8}$ as tall as the one before it, so the most likely single outcome is always $k = 1$.

4.3.1 Mean and Standard Deviation

How many packs should you expect to open? There is a neat argument. Let μ be the expected number of trials. You always open the first pack. With probability $1 - p$, it has no rare card, and then you are back where you started, needing μ more packs on average. So

$$\mu = 1 + (1 - p)\mu \implies p\mu = 1 \implies \mu = \frac{1}{p}$$

Geometric Mean and Standard Deviation

$$\mu = \frac{1}{p}, \quad \sigma = \frac{\sqrt{1-p}}{p}$$

With $p = \frac{1}{8}$, you expect to open 8 packs, which matches common sense: a 1-in-8 event takes 8 tries on average. The standard deviation is $\frac{\sqrt{7/8}}{1/8} \approx 7.48$, almost as large as the mean. Geometric waits are very unpredictable: you might get lucky on the first pack or wait 25.

4.3.2 No Memory

Suppose you have already opened 10 packs with no rare card. Are you “due”? No. The packs are independent, so the chance of needing more than 5 *additional* packs is $(\frac{7}{8})^5$, exactly the same as for someone just starting. The geometric distribution is *memoryless*: the past does not change the future. Believing otherwise is called the *gambler’s fallacy*. The

next chapter's exponential distribution, which models waiting times in continuous time, has the same property.

Exercise 12 The First Miss

The 70% free-throw shooter keeps shooting until they miss.

1. What is the probability that their first miss is on their 4th shot?
2. How many shots should they expect to take, including the miss?

Working Space

Answer on Page 59

Exercise 13 How Many Packs to Be Sure?

How many packs must you buy to have at least a 90% chance of getting a rare card ($p = \frac{1}{8}$)? (Hint: you want $P(X > k) \leq 0.1$. Use logarithms.)

Working Space

Answer on Page 59

Exercise 14 Binomial or Geometric?

For each question, say which distribution applies, and answer it.

1. You roll a die until you get a 6. What is the probability it takes exactly 6 rolls?
2. You roll a die 6 times. What is the

Working Space

probability you get exactly one 6?

Answer on Page 59

4.4 Looking Ahead: Rare Events

What happens to a binomial distribution when n is huge and p is tiny? Suppose a factory makes 1,000 phones a day, and each has a 0.3% chance of a defect. The number of defects is binomial with $n = 1000$ and $p = 0.003$, and the chance of exactly 2 defects is $\binom{1000}{2}(0.003)^2(0.997)^{998} \approx 0.2242$. Computing $\binom{1000}{2}$ and 0.997^{998} is clumsy. The next chapter's Poisson distribution gives almost the same answer, 0.2240, using only the mean $np = 3$. It is the binomial distribution's limit for rare events.

Poisson and Exponential Distributions

5.1 Counting Events

On a typical Friday night, a pizza shop gets about 12 orders an hour. Some hours bring 8, some bring 17. How many delivery drivers should it schedule? And after one order comes in, how long until the next? Two distributions answer these questions: one counts events, the other times the gaps between them.

The *Poisson distribution* models the number of events that occur in a fixed interval of time or space, when events happen independently at a constant average rate λ . Customer arrivals in an hour, calls to a help desk in a day, and defects on a production line are typical examples. If X counts the events, then

$$P(X = k) = \frac{e^{-\lambda} \lambda^k}{k!}, \quad k = 0, 1, 2, \dots$$

The probabilities sum to 1, because $\sum_k \frac{\lambda^k}{k!} = e^\lambda$.

5.1.1 A Worked Example

A store averages $\lambda = 3$ customers per hour. The probability of no customers in an hour is

$$P(X = 0) = e^{-3} \approx 0.0498$$

The probability of exactly two customers is

$$P(X = 2) = \frac{e^{-3} \cdot 3^2}{2!} \approx 0.224$$

The probability of at most two customers is $P(X \leq 2) \approx 0.423$, and the probability of at least one customer is $1 - 0.0498 \approx 0.950$.

5.2 Mean and Variance

A Poisson random variable has the same mean and variance:

$$E[X] = \lambda, \quad \text{Var}(X) = \lambda$$

So the store with $\lambda = 3$ expects 3 customers per hour, with variance 3 and standard deviation about 1.73.

5.3 Waiting Times

The *exponential distribution* models the time T until the next event, when events occur at a constant rate λ . Unlike a count, a waiting time can be any positive number, like 1.37 minutes, so T is a *continuous* random variable. The chance of waiting *exactly* 1.37000... minutes is zero; instead we ask about ranges, like $P(T > 3)$. The *density* $f(t)$ describes how the probability is spread out (you will study densities in the Continuous Distributions chapter). For the exponential distribution, the density and the tail probability are

$$f(t) = \lambda e^{-\lambda t}, \quad P(T > t) = e^{-\lambda t}, \quad t \geq 0$$

Its mean is $E[T] = \frac{1}{\lambda}$, and its variance is $\text{Var}(T) = \frac{1}{\lambda^2}$.

5.3.1 A Worked Example

Calls arrive at a rate of $\lambda = 0.5$ per minute, so the average time between calls is 2 minutes. The probability that the next call takes more than 3 minutes is

$$P(T > 3) = e^{-0.5 \cdot 3} = e^{-1.5} \approx 0.223$$

The probability that a call arrives within one minute is $P(T \leq 1) = 1 - e^{-0.5} \approx 0.393$.

5.4 The Connection

If events form a Poisson process with rate λ , then the number of events in an interval of length t is Poisson with mean λt , and the waiting time to the first event is exponential with rate λ . These are two views of the same process. For example, the probability of no calls in 2 minutes is $P(X = 0) = e^{-0.5 \cdot 2} = e^{-1}$, which equals $P(T > 2) = e^{-1}$.

5.5 The Memoryless Property

The exponential distribution is the only continuous distribution with the *memoryless* property:

$$P(T > s + t \mid T > s) = P(T > t)$$

Knowing that 3 minutes have passed without a call says nothing about the remaining wait. For the example above, $P(T > 5 \mid T > 3) = \frac{e^{-2.5}}{e^{-1.5}} = e^{-1} \approx 0.368$, which equals $P(T > 2)$.

5.6 Assumptions

These models are reasonable only when the events are roughly independent, the rate is roughly constant, and two events rarely happen at exactly the same moment. A rush hour that doubles the arrival rate, or a machine that wears out, violates the constant-rate assumption, and the models must then use a rate that changes over time.

Exercise 15 Poisson probability

A call center averages 3 calls per hour. What is the probability of exactly 5 calls in an hour?

Working Space

Answer on Page 59

Exercise 16 Exponential tail

Using the rate of 0.5 calls per minute from the example, find the probability that the next call takes more than 4 minutes.

Working Space

Answer on Page 60

Exercise 17 **Mean and variance**

The time between arrivals at a counter is exponential with mean 4 minutes. Find the rate λ , the variance, and the probability that the next arrival takes more than 4 minutes.

Working Space

Answer on Page 60

Exercise 18 **Pizza Drivers**

A pizza shop averages 12 orders per hour on Friday nights. Use a Poisson model.

1. What is the expected number of orders in a 15-minute period?
2. What is the probability of no orders in a 15-minute period?
3. What is the average wait between orders, in minutes?

Working Space

Answer on Page 60

CHAPTER 6

Conditional Probability

Your weather app says there is a 30% chance of rain. But if you already know that dark clouds are rolling in, the chance is higher. A *conditional probability* is a probability that has been updated by something you already know.

Let us say there is a virus going around, and there is a vaccine for it that requires two shots. You are working at a school, and you are wondering how effective the vaccines are. Some students are unvaccinated, some have had one shot, and some have had two shots. One day, you test all 644 students to see who has the virus. You end up with the

following table:

	V_0	V_1	V_2
T_+	88 students	36 students	96 students
T_-	92 students	76 students	256 students

Here is what each symbol means:

- V_0 : student has had zero vaccination shots
- V_1 : student has had one vaccination shot
- V_2 : student has had both vaccination shots
- T_+ : student tested positive for the virus
- T_- : student tested negative for the virus

So, for example, your data indicates that 76 students who had only one of the two shots tested negative for the virus.

Your principal has a few questions. The first is, “If I put five randomly chosen students in a study group together, what is the probability that at least one of them has the virus?”

The first thing you might do is make a new table that shows the probability of a randomly chosen student being in any particular group. You just divide each entry by 644 (the total number of students).

	V_0	V_1	V_2
T_+	$p(V_0 \text{ AND } T_+) = 13.7\%$	$p(V_1 \text{ AND } T_+) = 5.6\%$	$p(V_2 \text{ AND } T_+) = 14.9\%$
T_-	$p(V_0 \text{ AND } T_-) = 14.3\%$	$p(V_1 \text{ AND } T_-) = 11.8\%$	$p(V_2 \text{ AND } T_-) = 39.8\%$

(In this table, we expressed the number as a percentage with a decimal point — you had to round off the numbers. If you wanted exact answers, you would have to keep each as

a fraction: 36 students represents $\frac{9}{161}$ of the student body.)

6.1 Marginalization

Now we can sum across the columns and rows.

	V_0	V_1	V_2	sum
T_+	0.137	0.056	0.149	$p(T_+) = 0.342$
T_-	0.143	0.118	0.398	$p(T_-) = 0.658$
sum	$p(V_0) = 0.280$	$p(V_1) = 0.174$	$p(V_2) = 0.547$	

If a student is chosen randomly from the entire student body, there is a 34.2% chance that the student has tested positive for the virus, and there is 17.4% chance that the student has one shot of the vaccine.

This summing of the probabilities across one dimension is known as *marginalizing*. Marginalization is just summing across all the variables that you do not care about. If you do not care who has the virus, just the probability that a student has not received even one shot of the vaccine, you can simply marginalize all the vaccine statuses. To answer the principal's question, the easy thing to do is find the answer of the opposite: "if I put five randomly chosen students in a study group together, what is the probability that *none* of them has tested positive for the virus?"

The chance that a randomly chosen student does not have the virus ($p(T_-)$ is 65.8%. This means the chance that 5 randomly chosen students do not have the virus is $0.658^5 \approx 0.124$. Thus, the probability of the opposite is $1.0 - 0.124 = 0.876$

The answer, then, is "If you put 5 kids in a study group together, there is an 87.6 % probability that at least one of them has the virus."

6.2 Conditional Probability

Now the principal asks you, "What if I make a group of five kids who have had both shots of the vaccine? What are the odds that one of them has tested positive for the virus?"

This involves the idea of *Conditional probability*. You want to know the odds that a student does not have the virus, given that the student has had both shots of the vaccine.

There is a mathematical notation for this:

$$p(T_-|V_2)$$

That is the probability that a student who has had both vaccination shots will test negative for the virus.

How would you calculate this? You would count all the students who had a negative test *and* both vaccination shots, which you would divide by the total number of students who had both vaccination shots.

$$p(T_-|V_2) = \frac{256}{96 + 256} = \frac{8}{11} \approx 72.7\%$$

If we are working from the probabilities, you can get the same result this way. Divide the probability that a randomly chosen student had a negative test *and* both vaccination shots by the probability that a student had both vaccination shots:

$$p(T_-|V_2) = \frac{p(T_- \text{ AND } V_2)}{p(V_2)} = \frac{0.398}{0.547} \approx 72.7\%$$

Notice that this is different from $p(V_2|T_-)$, which is the probability that a student has had both vaccinations, given they tested negative for the virus.

Back to the principal's question: "If you have five students who have had both vaccinations, what is the probability that all of them tested negative for the virus?" The probability that one student is virus-free is $\frac{8}{11}$, so the probability that five students are virus-free is $(\frac{8}{11})^5 \approx 0.203$. So, there is a 79.7% chance that at least one of the five has the virus.

Exercise 19 Conditional probability from the table

Using the table of students, find $p(V_1 | T_+)$, the probability that a student has had one shot given that the student tested positive. Also find $p(T_+ | V_1)$. Are they the same?

Working Space

Answer on Page 60

6.3 Chain Rule for Probability

You just used this equality: For any events A and B

$$p(A|B) = \frac{p(A \text{ AND } B)}{p(B)}$$

This is more commonly written like this:

$$p(A \text{ AND } B) = p(A|B)p(B)$$

This is an abstract way of writing the idea, but the idea itself is pretty intuitive. The probability that you are going to buy a ticket and win the lottery is equal to the probability that you buy a ticket times the probability that you win, given that you have bought a ticket. (Here A is “win the lottery” and B is “buy a ticket”.)

This is known as *The Chain Rule of Probability*. We can chain together as many events as we want. Suppose you apply to a college (Z), get accepted (Y), and receive a scholarship (X). Then

$$p(X \text{ AND } Y \text{ AND } Z) = p(X | Y \text{ AND } Z) p(Y | Z) p(Z)$$

In English: the probability that you apply, get in, and get a scholarship is the probability that you apply, times the probability that you get in given that you applied, times the probability of a scholarship given that you applied and got in. Each step only has to consider the world in which the earlier steps already happened.

Exercise 20 Chaining Probabilities

You apply to a college with probability 0.9. If you apply, you are accepted with probability 0.4. If accepted, you receive a scholarship with probability 0.25. What is the probability that you end up with a scholarship to this college?

Working Space

Answer on Page 60

Exercise 21 Two Shots

Using the student table, a principal picks one student at random. What is the probability that the student had zero

Working Space

shots, given that they tested positive? Compare it with the probability that a student had two shots, given that they tested positive. Can you conclude that the vaccine does not work, since more positive students had two shots than one?

Answer on Page 60

Answers to Exercises

Answer to Exercise 1 (on page 9)

In decimal degrees, Sydney is $(-33.8667, 151.2)$ and Rio is $(-22.9, -43.2)$. `haversine(-33.8667, 151.2, -22.9, -43.2)` gives about 13,500 km. (Surprisingly, the shortest route crosses Antarctica, reaching about 77° S.)

Answer to Exercise 2 (on page 15)

probability of all 5's = $\frac{1}{6} \times \frac{1}{6} \times \frac{1}{6} = \left(\frac{1}{6}\right)^3 = \frac{1}{216} \approx 0.0046$

Answer to Exercise 3 (on page 15)

probability of at least one heads = $1.0 - \text{probability of all tails} = 1.0 - \left(\frac{1}{2}\right)^5 = 1.0 - \frac{1}{32} = \frac{31}{32} \approx 0.97$



Answer to Exercise 4 (on page 24)

$E[2X + 1] = 2 \cdot \frac{7}{2} + 1 = 8.$

Answer to Exercise 5 (on page 24)

$\text{Var}(2X + 1) = 4 \cdot \frac{35}{12} = \frac{35}{3}.$

Answer to Exercise 6 (on page 24)

$$E[X^2] = 0 + \frac{1}{2} + 1 = \frac{3}{2}, \text{ so } \text{Var}(X) = \frac{3}{2} - 1 = \frac{1}{2}.$$

Answer to Exercise 7 (on page 24)

The net winnings are 15 with probability $\frac{1}{5}$ and -5 with probability $\frac{4}{5}$. The expected value is $3 - 4 = -1$. The expected loss is \$1 per game, so the game is not worth playing.

Answer to Exercise 8 (on page 31)

$$P(\text{exactly 4 makes}) = \binom{6}{4} (0.6)^4 (0.4)^2 = 15 \times 0.1296 \times 0.16 = 0.31104$$

About 31%.

Answer to Exercise 9 (on page 40)

1. Binomial, with $n = 10$ and $p = \frac{1}{6}$.
2. Not binomial: the number of trials is not fixed. (If cards were replaced and reshuffled each time, this would be geometric.)
3. Not exactly binomial: the cards are drawn without replacement, so the trials are not independent. But 5 cards is less than 10% of 52 (5.2), so a binomial model with $n = 5$ and $p = \frac{1}{4}$ is a reasonable approximation.
4. Not binomial: streaks mean the days are not independent.

Answer to Exercise 10 (on page 41)

1. $\binom{25}{3} (0.15)^3 (0.85)^{22} \approx 0.217$.
2. $1 - P(X = 0) = 1 - 0.85^{25} \approx 0.983$.
3. $\mu = 25(0.15) = 3.75$ and $\sigma = \sqrt{25(0.15)(0.85)} \approx 1.79$.

Answer to Exercise 11 (on page 41)

1. $\mu = 20(0.25) = 5$ and $\sigma = \sqrt{20(0.25)(0.75)} \approx 1.94$.
2. `binom.cdf(5, 20, 0.25)` gives $P(X \leq 5) \approx 0.617$, and `binom.sf(11, 20, 0.25)` gives $P(X \geq 12) \approx 0.0009$. Guessing passes less than one time in a thousand.

Answer to Exercise 12 (on page 44)

Here a “success” is a miss, with $p = 0.3$.

1. $(0.7)^3(0.3) \approx 0.103$.
2. $\mu = \frac{1}{0.3} \approx 3.33$ shots.

Answer to Exercise 13 (on page 44)

We need $\left(\frac{7}{8}\right)^k \leq 0.1$. Taking logs, $k \ln \frac{7}{8} \leq \ln 0.1$, and since $\ln \frac{7}{8}$ is negative, dividing flips the inequality: $k \geq \frac{\ln 0.1}{\ln(7/8)} \approx 17.2$. So you need 18 packs, more than twice the average of 8.

Answer to Exercise 14 (on page 44)

1. Geometric with $p = \frac{1}{6}$: $\left(\frac{5}{6}\right)^5 \frac{1}{6} \approx 0.067$.
2. Binomial with $n = 6$ and $p = \frac{1}{6}$: $\binom{6}{1} \frac{1}{6} \left(\frac{5}{6}\right)^5 = \left(\frac{5}{6}\right)^5 \approx 0.402$. Notice that it is exactly 6 times the first answer: the single 6 can be in any of 6 positions, instead of only the last.

Answer to Exercise 15 (on page 49)

$$P(X = 5) = \frac{e^{-3.3} 3.3^5}{5!} \approx 0.101.$$

Answer to Exercise 16 (on page 49)

$$P(T > 4) = e^{-0.5 \cdot 4} = e^{-2} \approx 0.135.$$

Answer to Exercise 17 (on page 50)

$\lambda = \frac{1}{4} = 0.25$ per minute. The variance is $\frac{1}{\lambda^2} = 16$. The probability is $P(T > 4) = e^{-0.25 \cdot 4} = e^{-1} \approx 0.368$.

Answer to Exercise 18 (on page 50)

1. 15 minutes is a quarter hour, so $\lambda t = 12 \times 0.25 = 3$ orders.
2. $P(X = 0) = e^{-3} \approx 0.050$, about 5%.
3. The rate is $\frac{12}{60} = 0.2$ orders per minute, so the mean wait is $\frac{1}{0.2} = 5$ minutes.

Answer to Exercise 19 (on page 53)

$p(V_1 | T_+) = \frac{36}{220} = \frac{9}{55} \approx 16.4\%$, since 220 students tested positive. $p(T_+ | V_1) = \frac{36}{112} = \frac{9}{28} \approx 32.1\%$, since 112 students had one shot. These are different, as the notation section warns.

Answer to Exercise 20 (on page 54)

$0.9 \times 0.4 \times 0.25 = 0.09$, a 9% chance.

Answer to Exercise 21 (on page 54)

$p(V_0 | T_+) = \frac{88}{220} = 40\%$ and $p(V_2 | T_+) = \frac{96}{220} \approx 43.6\%$. No. Far more students had two shots (352) than zero shots (180), so of course more positive students are double-vaccinated. The fair comparison is $p(T_+ | V_2) = \frac{96}{352} \approx 27.3\%$ versus $p(T_+ | V_0) = \frac{88}{180} \approx 48.9\%$: a double-vaccinated student was much less likely to test positive.



INDEX

- 10% condition, 39
- annotated map, 12
- API key, 11
- arc length, 9
- bar graph
 - in python, 20
- binomial coefficient, 29
- binomial distribution, 31, 37
 - mean and standard deviation, 39
- binomial series, 30
- binomial setting, 37
- binomial theorem, 30
- chain rule of probability, 54
- choose function, 29
- combinations
 - formula, 29
- combinatorics, 27
- commutativity, 32
- complement rule, 52
- conditional probability, 51, 52
 - chain rule, 53
 - notation, 52
- coordinates, 10
- correlation and causation, 15
- cycle, 34
- dependent event, 14
- dice sum distribution, 16
- discrete, 22
- disjoint cycles, 35
- equator, 5
- expected count, 19
- expected value, 23
- exponential distribution, 48
- factorial, 28
- gambler's fallacy, 43
- geocoding, 10
- geographic coordinates, 12
- geometric distribution, 37, 42
 - mean, 43
- geometric setting, 42
- Google Geocoding API, 10
- great-circle route, 9
- Haversine formula, 7
- haversine formula, 7
- independent event, 14
- latitude, 5, 10
- least common multiple, 35
- linearity of expectation, 23
- longitude, 6, 10
- Longitude and Latitude
 - coordinates of, 6
- marginal probability, 52
- marginalizing, 52

- memoryless, [43](#), [49](#)
- memoryless property, [49](#)
 - geometric, [43](#)
- Mercator projection, [9](#)
- minutes and seconds, [7](#)
- multiplication principle, [27](#)

- Nautical Mile, [7](#)
- nautical mile, [7](#)

- order, [35](#)

- permutations, [31](#)
 - composing, [32](#)
 - cycles, [34](#)
 - formula, [28](#)
 - identity permutation, [32](#)
 - inverses, [34](#)
- plotly, [12](#)
- Poisson distribution, [47](#)
- Prime Meridian, [6](#)
- probability, [13](#)
 - sum to one, [14](#)
- probability mass function, [22](#)
- pseudorandom numbers, [18](#)

- random number generation, [17](#)
- random variable, [22](#)
- reverse geocoding, [11](#)

- Scattergeo, [12](#)
- simulation, [17](#)
- sorted(), [33](#)
- sorting, [33](#)

- variance, [23](#)
 - random variable, [23](#)