Unit 1: Discrete probability modeling and simulation

Ethan Levien

September 30, 2025

Contents

Introduction

The first step in our journey into regression modeling is to establish a language and system of notation to communicate uncertainty. Before doing so, however, let us say a few words about modeling in general. Broadly speaking, models are simplified representations of the world. For example, astrology is a model of human behavior, and Newton's laws are models of how objects move in physical space. Neither model is perfectly correct, but Newton's laws provide a remarkably good approximation. In science (and in life), we often use mathematical models.

The subject of this course is regression models. We will define these precisely later, but roughly speaking a regression model describes how the distribution of a variable y (the response variable) is related to another variable x (the predictor). Examples include predicting height based on age, predicting the probability of developing a disease given a genetic mutation, or even a large language model predicting the next word in a sentence.

We will mostly focus on linear regression models, which in mathematical notation take the form

$$y = \sum_i \beta_i x_i + \text{“noise”}. \tag{1}$$

This equation says that y—the quantity we are interested in predicting—is expressed as a sum of observed variables plus some randomness. For instance, suppose we want to predict how long someone will live. Even if we know their entire medical history, their parents' medical histories, and detailed demographic information, there is always some uncertainty.

1.1 Basic definitions

References: [1, Ch. 1 Sec. 2 and Ch. 2.1]

1.1.1 Sample spaces, random variables, outcomes and events

Returning to our immediate goal: we need to develop a language to talk about this uncertainty. For our purposes, we can pretty much think of a random variable as any variable which we cannot predict prior to an observation of it, regardless of how much information we have, such as the outcome of a coin flip. We will use capital letters, often X, Y, Z, to denote random variables. A random variable or set of random variables has a sample space, denoted S, which is the set of all possible values it can take. We sometimes write S_X to indicate that S is the sample space of X, but omit the subscript when it is clear from the context. For the coin flip, $S = \{\text{heads}, \text{tails}\}$. Usually we will simply specify the sample space with numerical quantities, e.g. letting heads and tails be represented by 1 and 0. For a dice roll dice $S = \{1, 2, 3, 4, 5, 6\}$. For the height of a tree, $S = \mathbb{R}_{\geq 0}$ (the positive real numbers), although it must become very unlikely to have very high tree so we could replace this with the interval $[0, 1000\,\text{ft}]$ (more on that in the next unit). For now, we will be considering only denumerable (i.e. discrete) sets, so not $\mathbb{R}$.

We draw a distinction between outcomes (elements of S) and events – the latter are subsets of outcomes. For example, we might refer to the event that the roll of a die is greater than 2. We can connect the two by defining a random variable 1_E which is 1 if an event happens and 0 otherwise. Hence, the probability of an events can also be expressed as a the probability of a random variable (try writing it out to convince yourself).

We can characterize a random variable, say X, using a probability model or probability distributionwhich maps events to real numbers between 0 and 1 [1, Definition 1.2.1]. I will use the notation

$$P_X(x) = \text{chance that } x \text{ happens for } x \in S$$

or just $P(x)$ if it is clear from the context we are talking about probabilities of X. For an even $U \subset S$, we will write

$$P(U) = \sum_{x \in U} P(x). \tag{2}$$

This leverages the additivity of probabilities for mutually exclusive events (see below).

Example 1. The Bernoulli distribution [1, Example 2.3.2] is probably the simplest random variable It models a variable with binary outcome, for example the result of a YES/NO survey or a diagnostic test. If Y follows a Bernoulli distribution, then

$$P_Y(y) = \begin{cases} 1 - q & y = 0 \\ q & y = 1 \end{cases} \tag{3}$$

Remember that $P_Y(y) = P(Y = y) = P(\{Y = y\})$. It's important to be flexible with notation. These formulas make sense for any $0 \leq q \leq 1$. We say that q is a parameter of the distribution. Instead of writing out (3) every time we want to indicate that a variable Y follows a Bernoulli distribution, we will write

$$Y \sim \text{Bernoulli}(q).$$

In general, for a random variable with a particular name and set of parameters we will write

$$\text{Variable} \sim \text{Distribution(parameters)}.$$

Not all random variables have specific names, but when they do this notation will allow us to avoid writing down a probability distribution explicitly. Moreover, for more complicated random variables which, we can often express them in terms of random variables with known names.

Example 2. Suppose a political scientist is studying whether individuals support a new policy reform. Imagine that each person in the study is asked the same yes/no question twice, perhaps one week apart, to measure the consistency of their views. Each response can be modeled as a Bernoulli random variable with $(0, 1) = (\text{NO}, \text{YES})$. When we put the two answers together, we obtain a pair (X_1, X_2). The sample space is

$$S = \{(0, 0), (0, 1), (1, 0), (1, 1)\} \tag{4}$$

and there is some distribution $P_{X_1,X_2}(x_1, x_2)$. We call this the joint distribution of X_1 and X_2 (see below). We can think of this pair as a new random variable (random variables can be vectors or lists of numbers).

1.1.2 Properties of probability distributions

References: [1, Ch. 1, Sec. 1.3]

A probability measure P on a sample space S satisfies the following axioms:

- **Nonnegativity:** For every event $U \subseteq S$,
$$P(U) \geq 0.$$
- **Normalization:**
$$P(S) = 1.$$
- **Countable additivity:** For any countable collection of pairwise disjoint sets $\{U_i\}_{i=1}^{\infty} \subseteq S$,
$$P\left(\bigcup_{i=1}^{\infty} U_i\right) = \sum_{i=1}^{\infty} P(U_i).$$

Example. If $S = \{1, 2, 3, 4, 5, 6\}$ is the outcome space of a die roll, then $\{1, 2, 3\} \subseteq S$. If the coin-flip space is $S = \{\text{heads}, \text{tails}\}$, then

$$P(\text{heads} \cup \text{tails}) = P(S) = 1.$$

Example 3. Suppose we flip two fair coins and let X_1 and X_2 denote the outcomes. The sample space is the same as Example 2; that is,

$$S = \{(0, 0), (0, 1), (1, 0), (1, 1)\}$$

where $0 = \text{T}$ and $1 = \text{H}$. Since the coins are fair, each of the outcomes above should have the same probability. Thus each should have probability q and $q = 1/4$ since they all need to add to 1:

$$P_{A,B}(0, 0) + P_{A,B}(0, 1) + P_{A,B}(1, 0) + P_{A,B}(1, 1) = 4q = 1 \implies q = \frac{1}{4}.$$

Notice that

$$P_{A,B}(1, 1) = P_A(1)P_B(1) = \frac{1}{2} \times \frac{1}{2}$$

This makes sense intuitively: We will get a heads half the time on the first flip, and then half of those times we will get one on the second flip. We will soon see that this is related to the fact that the two variables are independent – in other words, knowing one does not influence the other.

For more discussion and examples, see [1, Ch. 1, Sec. 1.2.1].

1.2 Sampling and simulation

1.2.1 IID samples and simulations

A measurement of a random variable is a sample and statistical inference is the process of estimating the parameters θ from a sample of a random variable. In statistics, we seek to answer key questions about what we can learn from a sample:

- Consider the example of a survey: let's suppose we don't have information about every student in the college. Rather, a survey of five students from this class is conducted, finding 4 yeses and 1 no. What is our best prediction of the total fraction of students in the college who answered YES? What assumption do we make when we answer this question?
- How many experiments do we need to do know if a drug is effective?

We will talk about statistical inference in more detail when we get to Unit 3. We usually assume (although it is not strictly true), that we are given independent samples of the same random variable. We call this iid samples (for independent and identically distributed). We have not defined independence mathematically, but we can understanding it intuitively as meaning: The value of any particular sample has not influence on the others. This is the situation we are in when flipping a coin for example.

1.2.2 From probabilities to samples

For now, we note the basic relationship between a probability and a sample: If we have n samples of a random variable and the outcome $x \in S_X$ occurred $N(x)$ times, then

$$P(X = x) \approx \frac{N(x)}{N}. \tag{5}$$

In the colab notebook for this unit, you will see how to implement a statement like this from a numpy array or dataframe.

Going forward, I will use the notation N instead of a P to denote the number of times an event or outcome occurs. As with probability, there are a few different notation we will use depending on the context: Just as with probabilities, we have the following equivalent ways of denoting the number of samples where a random variable X is equal to $x \in S$.

$$N(\{X = x\}) = N(X = x) = N_X(x) = N(x) \tag{6}$$

I will use $N(x)$ when there is no ambiguity.

Example 4. Suppose we flip a (possibly biased) coin 1000 times and record the outcomes. Each flip is a Bernoulli random variable $X \sim \text{Bernoulli}(p)$, where p is the probability of heads. We do not know p in advance, but we can estimate it from the observed data. Another example: if a survey asks each respondent whether they support a new policy (YES/NO), then the proportion of YES answers in the sample is our best estimate of the true support rate in the whole population.
Below is Python code that simulates coin flips and uses sample frequencies to estimate p.

```
import numpy as np

# True probability of YES (e.g. support for a policy)
p_true = 0.6

# Collect 1000 samples (YES=1, NO=0)
n = 1000
samples = np.random.binomial(n=1, p=p_true, size=n)
```

```
# Estimate probability from sample frequency
p_hat = np.mean(samples)

print("Estimated probability:", p_hat)
```

On running this code, the output will be close to 0.6, but not exactly, since the data are random. The difference between the estimate $\hat{p}$ and the true parameter p is the central problem of statistical inference.

Note on probabilities as fraction vs. belief

There are two different ways we can interpret a statement like: The probability someone in this room is over 6 feet is 95%. Either it can be interpreted as a measure how likely it is to find someone in the room over 6 feet, or if we were to hypothetically generate random samples over and over what fraction of them would contain someone over 6 feet.

1.3 Independence, conditioning and marginal distributions

References: [1, Sec. 2.8.1]

1.3.1 Independence and marginal distributions

Two random variables X, Y are independent if

$$P(X = x, Y = y) = P(X = x)P(Y = y) \tag{7}$$

for all $(x, y) \in S$. This might make more sense after we introduce the idea of conditional probability.

Example 5 (Gene model)**.** Let's consider once again the case of two random variables taking values in $\{0, 1\}$, so the sample space is again the same as 3. To be concrete, we introduce another application: Y_A and Y_B represent the probabilities that an individual has a mutation on genes A and B. Let us be given the probability distribution

$$P(Y_A = y_A, Y_B = y_B) = P(\{Y_A = y_A\} \cap \{Y_B = y_B\}) = \begin{cases} 1/2 & \text{if } y_A = 0 \text{ and } y_B = 0 \\ 1/8 & \text{if } y_A = 0 \text{ and } y_B = 1 \\ 1/8 & \text{if } y_A = 1 \text{ and } y_B = 0 \\ 1/4 & \text{if } y_A = 1 \text{ and } y_B = 1 \end{cases}$$

The sample space is the same as Example 3, but we can check that Y_A and Y_B are no longer independent:

$$P(Y_A = 0, Y_B = 0) = \frac{1}{2}$$

on the other hand

$$P(Y_A = 0) = P((0, 1) \text{ or } (0, 0)) = P(0, 1) + P(0, 0) = \frac{1}{8} + \frac{1}{2} = \frac{5}{8}$$
$$P(Y_B = 0) = P((0, 0) \text{ or } (1, 0)) = P(0, 0) + P(1, 0) = \frac{1}{2} + \frac{1}{8} = \frac{5}{8}$$

and $25/64 \approx 0.39 \neq 1/2$.

The joint distribution does not directly tell us the probabilities of observing a value of only one of the random variables, e.g. the probability that $Y_A = 1$. The procedure used above to obtain these probabilities, by summing over the other variable, is called marginalization. In general, for a joint distribution $P(x, y)$ the marginal distribution of x is

$$P(x) = P_X(x) = \sum_{y \in S_Y} P(x, y). \tag{8}$$

Notice that in the case of independent variables,

$$\sum_{y \in S_Y} P(x, y) = P(x) \sum_{y \in S_Y} P(y) = P(x) \tag{9}$$

Example 6 (Calculating probabilities from a specified joint distribution)**.** Suppose random variables $A, B, C \in \{0, 1\}$ have the joint probability distribution $P(a, b, c) = P(A = a, B = b, C = c)$ given on $S = \{0, 1\}^3$ by

$$p(a, b, c) = \begin{cases} 0.1, & (a, b, c) = (0, 0, 0), \\ 0.2, & (a, b, c) = (0, 0, 1), \\ 0.1, & (a, b, c) = (0, 1, 0), \\ 0.1, & (a, b, c) = (0, 1, 1), \\ 0.1, & (a, b, c) = (1, 0, 0), \\ 0.1, & (a, b, c) = (1, 0, 1), \\ 0.1, & (a, b, c) = (1, 1, 0), \\ 0.2, & (a, b, c) = (1, 1, 1) \end{cases} \tag{10}$$

Question 1: Compute the marginal probability $P(C = 1)$.

Solution: By marginalization,

$$\begin{aligned} P(C = 1) &= \sum_{a \in \{0,1\}} \sum_{b \in \{0,1\}} p(a, b, 1) \\ &= p(0, 0, 1) + p(0, 1, 1) + p(1, 0, 1) + p(1, 1, 1) \\ &= 0.2 + 0.1 + 0.1 + 0.2 \ = \ 0.6. \end{aligned} \tag{11}$$

Example 7. ([1, Example 2.3.4]) Suppose we flip a fair coin until we see a heads. Let Y be the number of flips until we see a heads. This is example of a geometric distribution, which is the number of trials of independent, identically distributed (iid) Bernoulli random variables until we see k successes. In more mathematical notation, if

$$X_i \sim \text{Bernoulli}(q), \quad i = 1, 2, 3, \ldots$$

then

$$Y = \min_{i \geq 1} \{i : X_i = 1\}$$

and we would say

$$Y \sim \text{Geometric}(q).$$

The sample space of Y is $\{1, 2, \ldots, \infty\}$. What is the probability distribution?

$$\begin{aligned} P(Y = k) &= P(X_1 = 0, X_2 = 0, \ldots, X_{k-1} = 0, X_k = 1) \\ &= P(X_1 = 0) \cdots P(X_{k-1} = 0) P(X_k = 1) \\ &= (1 - q)^{k-1} q \end{aligned}$$

This has the expected properties of Y. In particular, it decays as k increases and the decay is faster the larger q is.

1.3.2 Conditioning

What if we are interested in the chance that someone has a mutation in gene A and we know they do not have a mutation in gene B? In this case, we introduce the conditional probability $P(Y_A = 1|Y_B = 0)$. This is defined as the chance that gene A has a mutation in a person if we know there is no mutation at gene B. If we want to think about this in terms of population averages, it is the fraction of mutations in gene A among only those people without mutations in gene gene B. More generally, $P(X|Y = y)$ is the distribution of X if we know the value of $Y = y$.

It will be extremely useful to write this in terms of the joint and marginal probabilities. To do so, we can interpret the probabilities as fractions and use the definition above:

$$P(Y_A = 1|Y_B = 0) = \frac{N(Y_A = 1, Y_B = 0)}{N(Y_B = 0)} = \frac{N(Y_A = 1, Y_B = 0)/n}{N(Y_B = 0)/n}$$
$$= \frac{P(Y_A = 1, Y_B = 0)}{P(Y_B = 0)}$$

Example 8 (Example 5 cont.)**.** Consider Example 5. In this case the conditional probability of $Y_A = 1$ given $Y_B = 0$ is

$$P(Y_A = 1|Y_B = 0) = \frac{P(1, 0)}{P(Y_B = 0)} = \frac{1/8}{5/8} = \frac{1}{5}$$

Example 9 (Python: Sampling and conditional expectation from the gene model)**.** Here is Python code to generate samples from the gene model in Example 5 and compute the conditional probability $P(Y_A = 1 \mid Y_B = 0)$.

```
import numpy as np

# Define probabilities for each (Y_A, Y_B) pair
probs = [1/2, 1/8, 1/8, 1/4]  # (0,0), (0,1), (1,0), (1,1)
pairs = [(0,0), (0,1), (1,0), (1,1)]
# we could have defined this as [1,2,3,4]
# then we would just need to remember which outcomes they map to

# Generate samples
N = 10000
samples = np.random.choice(len(pairs), size=N, p=probs)
Y_A = np.array([pairs[i][0] for i in samples])
Y_B = np.array([pairs[i][1] for i in samples])

# Compute conditional probability P(Y_A = 1 | Y_B = 0)
mask = (Y_B == 0)
cond_prob = np.mean(Y_A[mask] == 1)
print("P(Y_A = 1 | Y_B = 0) =", cond_prob)
```

This code simulates the joint distribution, selects samples where $Y_B = 0$, and computes the fraction of those samples where $Y_A = 1$, which estimates $P(Y_A = 1 \mid Y_B = 0)$.

We use the notation $Y|(X = x)$ for the random variable Y conditioned on another random variable, Y, taking the value x. This variable has a probability distribution which is a function of x. As shorthand, we might write $Y|X$ and then write the probability distribution as a function of X (even though we like to reserve capital letters for random quantities and X is treated as a independent variable in this case).

Example 10 (Conditional Bernoulli Model)**.** Suppose $Y \sim$ Bernoulli$(1/2)$, and $X|Y \sim$ Bernoulli$(1/4Y + 1/4)$. This means:

- First, sample Y from Bernoulli$(1/2)$ (so $Y = 1$ with probability $1/2$, $Y = 0$ with probability $1/2$).
- Then, given Y, sample X from Bernoulli$(1/4Y + 1/4)$.

To connect this to a probability distribution function, we can write:

$$P(Y = y) = \frac{1}{2} \text{ for } y = 0, 1$$

$$P(X = x|Y = y) = \begin{cases} 1/4 & \text{if } x = 1, y = 0 \\ 3/4 & \text{if } x = 0, y = 0 \\ 1/2 & \text{if } x = 1, y = 1 \\ 1/2 & \text{if } x = 0, y = 1 \end{cases}$$

The joint probability is then $P(X = x, Y = y) = P(X = x|Y = y)P(Y = y)$.

Example 11. In this example use the same distribution as Example 6.
Question: Make a Python data frame whose columns are `A`, `B`, `C` and whose rows are i.i.d. samples from (10). From this data frame, estimate the conditional probability $P(A = 0 \mid B = 1, C = 1)$.

Solution: By definition,

$$P(A = 0 \mid B = 1, C = 1) = \frac{P(A = 0, B = 1, C = 1)}{P(B = 1, C = 1)}. \tag{12}$$

From (10),

$$P(A = 0, B = 1, C = 1) = 0.1, \qquad P(B = 1, C = 1) = p(0, 1, 1) + p(1, 1, 1) = 0.1 + 0.2 = 0.3,$$

so

$$P(A = 0 \mid B = 1, C = 1) = \frac{0.1}{0.3} = \frac{1}{3} \approx 0.3333. \tag{13}$$

To estimate this empirically in Python, sample from the eight outcomes with their probabilities, place the samples into a `pandas` DataFrame, and compute the empirical conditional probability:

```
import numpy as np
import pandas as pd

# Support and probabilities (match Eq. (1))
outcomes = np.array([
    (0,0,0), (0,0,1), (0,1,0), (0,1,1),
    (1,0,0), (1,0,1), (1,1,0), (1,1,1)
], dtype=int)
probs = np.array([0.1, 0.2, 0.1, 0.1, 0.1, 0.1, 0.1, 0.2], dtype=float)

# Draw N i.i.d. samples
```

```python
N = 200_000
idx = np.random.choice(len(outcomes), size=N, p=probs)
samples = outcomes[idx]

# Build DataFrame with columns A, B, C
df = pd.DataFrame(samples, columns=["A", "B", "C"])

# Estimate P(A=0 | B=1, C=1)
mask = (df["B"]==1) & (df["C"]==1)
est = (df.loc[mask, "A"]==0).mean()
print("Estimated P(A=0 | B=1, C=1):", est)
```

The printed estimate should be close to the exact value in (13).

In general, we have

$$P(x|y) = \frac{P(y,x)}{P(y)}. \tag{14}$$

Notice that we can replace $P(y,x) = P(y|x)P(x)$, to obtain Bayes' formula

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}. \tag{15}$$

An equivalent definition of independence is $P(y|x) = P(y)$ and $P(x|y) = P(y)$. In summary:

$$X \text{ and } Y \text{ are independent} \iff P(x,y) = P(x)P(y) \iff P(y|x) = P(y) \iff P(x|y) = P(x)$$

Equation (16) is also true for events, for example, we can write things like:

$$P(\{Y > y\}|X) = \frac{P(\{Y > y\}, X)}{P(X)}. \tag{16}$$

This is because, as mentioned earlier, we can always associated the even $\{Y > y\}$ with a Bernoulli random variable which is 0 when $Y \leq y$ and 1 otherwise.

1.4 Binomial Distribution

Suppose

$$Y_i \sim \text{Bernoulli}(q), \quad i = 1, \ldots, N$$

are independent. We will use the convention that $Y_i = 1$ with probability q. Let

$$Y = \sum_{i=1}^{N} Y_i$$

Then we say Y follows binomial distribution and write

$$Y \sim \text{Binomial}(N, q)$$

Example 12 (Calculating probabilities)**.** Let $N = 3$ and $k = 2$.

Question: Calculate $P(Y = 2)$?

Solution: There are 3 possible sequences that give $Y = 2$.

$$(1, 0, 1), (1, 1, 0), (0, 1, 1)$$

The probability that we see any particular one of these is $(1 - q)q^2$. For example,

$$\begin{aligned} P(y_1 = 1, y_2 = 0, y_3 = 1) &= P(y_1 = 1)P(y_2 = 0)P(y_3 = 1) \\ &= q(1 - q)q = q^2(1 - q). \end{aligned}$$

Therefore the chance to observe $Y = 2$ with $N = 3$ is

$$P(Y = 2) = P((1, 0, 1)) + P((1, 1, 0)) + P((0, 1, 1)) = 3q^2(1 - q).$$

Note that the binomial distribution has two parameters, N and q, representing the number of flips and probability of success respectively. Now let's think about what the probability distribution will look like. The chance to find any **particular** configuration of k ones is

$$q^k(1 - q)^{N-k}$$

because they are independent.

However, we need to account for the fact that there are many configurations with k ones. In general, there are

$$\binom{N}{k} = \frac{N!}{k!(N - k)!} = \frac{N \times (N - 1) \times (N - 2) \times \cdots \times (N - k + 1)}{k \times (k - 1) \times (k - 2) \times \cdots \times 1}$$

way to have k ones among N samples. [1]

This implies

$$P(Y = k) = \binom{N}{k} q^k(1 - q)^{N-k}.$$

A graph of this function looks like a bell curve when N is large:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import binom

# parameters
N = 100          # number of trials
q = 0.3          # success probability
M = 100000     # number of Monte Carlo replicates

rng = np.random.default_rng(123)
# Simulate Y ~ Binomial(N, q) via direct binomial draws
samples = rng.binomial(n=N, p=q, size=M)
```

[1] To better understand the formula above, let $C_{N,k}$ denote the number of sequences with k ones. We can break $C_{N,k}$ up into the number of terms for which 1 appears as the first element of the sequence and those for which zero is the first. If one comes first, we have $N - 1$ remaining slots to place $k - 1$ ones, thus there are $C_{N-1,k-1}$ of these sequences. Similarly, if zero comes first, we have $N - 1$ slots but now all k ones to place, thus there are $C_{N-1,k}$ of these. It follows that

$$C_{N,k} = C_{N-1,k-1} + C_{N-1,k}.$$

Notice that the quantity $C_{N,k}$ will be smallest when $k = 1$ or $k = N$, since in these cases there is only one way to configure the sequence: $C_{N,1} = C_{N,N} = 1$. You can solve the recursion to obtain the formula.

```
# Histogram settings for a discrete variable: bin edges at half-integers
bins = np.arange(-0.5, N + 1.5, 1)

# Compute analytical pmf on the support 0..N
k = np.arange(0, N + 1)
pmf = binom.pmf(k, N, q)

fig, ax = plt.subplots(figsize=(7, 4))

# Monte Carlo histogram (normalized to probabilities)
ax.hist(samples, bins=bins, density=True, alpha=0.5, label="Monte Carlo (
    hist)")

# Analytical pmf as stems
(markerline, stemlines, baseline) = ax.stem(k, pmf, label="Analytical
    distribution")
plt.setp(baseline, visible=False)


ax.set_xlabel("k")
ax.set_ylabel("Probability")
ax.set_title(f"Binomial(N={N}, q={q})")
ax.legend()
ax.set_xlim(-0.5, N + 0.5)
plt.tight_layout()
plt.show()
```

Exercises

Exercise 1 (Conditional probability from a joint table ❑)**:** Suppose the joint probability table for A and B is:

	$B=0$	$B=1$
$A=0$	0.1	0.3
$A=1$	0.2	0.4

(a) What is $P(A=1)$?

(b) What is $P(B=1)$?

(c) What is $P(A=1|B=1)$?

(d) What is $P(B=0|A=0)$?

Exercise 2 (Marginalization from a joint distribution ❑)**:** Let X and Y have joint probabilities:

$$P(X=0,Y=0)=0.2$$
$$P(X=0,Y=1)=0.2$$
$$P(X=1,Y=0)=0.1$$
$$P(X=1,Y=1)=0.5$$

(a) Compute the marginal distribution $P_X(x)$ for $x=0,1$.

(b) Compute the marginal distribution $P_Y(y)$ for $y=0,1$.

(c) Compute $P(Y=1|X=0)$.

(d) Compute $P(X=1|Y=0)$.

Exercise 3 (Conditional probability from a piecewise function ❑)**:** Let Z be a random variable with probability function

$$P_Z(z)=\begin{cases}0.2 & z=0\\ 0.5 & z=1\\ 0.3 & z=2\end{cases}$$

Let $W|Z\sim\text{Bernoulli}(Z/4+1/4)$.

(a) What is $P(W=1|Z=2)$?

(b) What is $P(W=1)$?

(c) What is $P(Z=1|W=1)$?

Exercise 4 (Marginalization and conditioning with three variables ❑)**:** Suppose

$$P(A=a,B=b,C=c)=\begin{cases}0.1, & (a,b,c)=(0,0,0),\\ 0.2, & (a,b,c)=(0,0,1),\\ 0.1, & (a,b,c)=(0,1,0),\\ 0.1, & (a,b,c)=(0,1,1),\\ 0.1, & (a,b,c)=(1,0,0),\\ 0.1, & (a,b,c)=(1,0,1),\\ 0.1, & (a,b,c)=(1,1,0),\\ 0.2, & (a,b,c)=(1,1,1)\end{cases}$$

(a) Compute $P(A = 1)$.

(b) Compute $P(B = 1|A = 0)$.

(c) Compute $P(C = 1|A = 1, B = 1)$.

(d) Compute $P(B = 0)$.

Exercise 5 (Conditional probability from a piecewise function ❑)**:** Let X be a random variable with probability function

$$P_X(x) = \begin{cases} 1/4 & x = 0 \\ 1/2 & x = 1 \\ 1/4 & x = 2 \end{cases}$$

Let $Y|X$ be defined as $Y|X \sim \text{Bernoulli}(X/4 + 1/4)$.

(a) What is $P(Y = 1|X = 2)$?

(b) What is $P(Y = 1)$?

(c) What is $P(X = 1|Y = 1)$?

Exercise 6 (Working with probability distributions and modeling ❑)**:** The first two problems are inspired by those in section 2 of [1]. You should look there for more practice.

(a) Suppose that

$$Y \sim \text{Bernoulli}(q)$$

and let $Z = 1/(1 + Y) + Y$. What is the sample space of Z and what is the probability distribution of Z?

(b) Suppose a coin is flipped. If the coin is heads, we write down 0. If the coin is tails, we roll a dice and write down the number. Let Y be the number we write down. What is the sample space and the probability distribution for Y?

(c) For the previous problem, conditioned on the dice rolling a 4, what is the probability we write down 0? Conditioned on the coin being tails, what is the probability the dice rolls a 3?

(d) Consider the geometric distribution. Provide three real-world examples of variables where the geometric distribution is a good model. Explain your reasoning.

Exercise 7 (Working with nested for loops ❑)**:** Consider the following code:

```
> for i in range(5):
>   for j in range(i+1):
>     print(i,end='')
>   print("")
```

prints out

```
> 0
> 11
> 222
> 3333
> 44444
```

Modify this code to print

```
> 0
> 01
> 012
> 0123
> 01234
> 012345
```

Exercise 8 (Working with more complex data)**:** Using an AI tool of your choosing, write python code to plot a map of the world with Hanover indicated by a red star. Then examine this code and answer the following questions:

- How was the data loaded and what variable was it stored in? Where is the information about the geometric shape of each country stored? Can you print out some of this information? Is there other information that is not used in the plot?
- Where is the information about the location of Hanover, NH stored?
- **Without using ChatGPT**, plot another point at Salt Lake City, UT with a green dot? (you can look up the coordinates).

Exercise 9 (Washington post data)**:** Below I load some data on homocide victims in US from the washington post. Don't worry about how I process it, all you need to work with is the DataFrame "data" on the very last line.

```
> data = pd.read_csv("https://raw.githubusercontent.com/washingtonpost
> /data-homicides/master/homicide-data.csv",encoding = "ISO-8859-1")
> data["victim_age"] = pd.to_numeric(data["victim_age"],errors="coerce")
```

(a) For each age $a = 1, \cdots, 100$ determine the number of victims $n(a)$ with an age $< a$ and put these values in a list. You can ignore the effects of those entries with missing ages.

(b) Now think for a moment about what you expect a plot of $n(a) vs. a$ to look like, then make a plot of $n(a)$ vs. a. Does it look like as expected?

(c) Divide the data into groups of white and non-white victims and repeat part (a) for each group. Then, for each group, make the plot from part (a). Comment on what you find.

Exercise 10 (Getting a sequence of wins)**:** Let J denote a random variable representing the number of times a fair coin is flipped before two heads appear in a row. As we saw in class, the following code generates simulations of J:

```
> def flip_until_two():
>   num_heads = 0
>   total_flips = 0
>   while num_heads <2:
>     y = np.random.choice([0,1])
>     if y == 0:
>       num_heads = 0
>     else:
>       num_heads = num_heads + 1
>     total_flips = total_flips + 1
>   return total_flips
```

(a) By modifying the above code, write a function rolluntil(n) that rolls a dice until we get n ones in a row. You should change the variable names accordingly. We will call this random variable R_n.

(b) Make a DataFrame where each column represents a value of n from 1 to 6 and each row is a simulation from the model R_n. There should be 100 rows.

(c) Create a plot comparing the maximum and minimum values of R_n as a function of n. You might notice one of these increases much faster than the other – why?

Exercise 11 (Joint distribution)**:** Consider the probability model defined by

$$\begin{aligned} Y_B &\sim \text{Bernoulli}(2/3) \\ Y_A|(Y_B = 0) &\sim \text{Bernoulli}(1/3) \\ Y_A|(Y_B = 1) &\sim \text{Bernoulli}(1/2) \end{aligned}$$

(a) Write done the joint probability distribution pf Y_A and Y_B.

(b) What are the marginal distributions of Y_A and Y_B?

(c) Are Y_A and Y_B independent? Confirm your answer with simulations (AI allowed, but first try to describe the approach without using AI)

Exercise 12 (Working with Washington Post Data)**:** This a continuation of Exercise 9 Consider the quantities

$$\begin{aligned} &P(age < z) \\ &P(age < z|\text{white}) \\ &P(age < z|\text{not white}). \end{aligned}$$

(a) Explain how each of these are related to the plot you made in Exercise 9.

(b) Make plots of them and comment of the difference between the plot in Exercise 9. Do you think age and race are independent based on these plots.

(c) Using the data, approximate (for this dataset)

$$P(\text{white}|10 < age < 60)$$

Hint: One way to do this is to use Bayes' rule

Exercise 13 (Covid modeling)**:** Suppose we are interested in modeling how likely we are to contract SARS CoV-3, a new more dangerous version of Covid, after a night out. To do this, we make the following assumptions:

- You interact with at most exactly N people in sequence, meaning no repeated interactions with the same person.
- Each person either has covid or does not (we do not distinguish between their viral load or how long they have had the disease).
- 10% of people in the student population have Covid.
- Given that someone has the Covid AND we interact with them, there is a 50% chance you contract the virus.

(a) Fill in the question marks in the following function so that it simulates whether or not you got covid from the night out; that is, so it returns 1 if you got covid and 0 if you didn't.

```
> def sim_covid(N):
>   got_covid = 0
>   for k in range(N):
>       got_covid_interaction = ????
>       if got_covid_interaction ==1:
>         got_covid =1
>   return got_covid
```

(b) The probability of getting covid form the entire night has the form

$$P(\text{get covid}) = 1 - (1 - x)^N \tag{17}$$

where $x \in [0, 1]$. Provide some justification of this formula and determine the value of x.

(c) Confirm Equation 17 using Monte Carlo simulations simulations. You should make a plot of this probability vs. N, similar to what we did for the Bernoulli distribution in the class notebook.

Addition student contributed exercises

Exercise 14: Given that:

```
> import numpy as np
> x = np.random.choice([0,1,2])
> y = np.random.choice([0,x],p = [1-(x/3), (x/3)])
>
```

(a) What is the sample space?

(b) Find the joint probability of x and y.

Exercise 15: If we are given that:

```
> import numpy as np
> u = np.random.choice([1,3],p= [0.6, 0.4])
> if u == 1:
>         v = np.random.choice([1, u],p= [0.8,0.2])
> else:
>         v = np.random.choice([1, u],p= [0.3,0.7])
```

Calculate the conditional probabilities $P(u, v)$.

References

[1] Michael J Evans and Jeffrey S Rosenthal. *Probability and statistics: The science of uncertainty*. Macmillan, 2004.