2
Representing Uncertainty

Real-world uncertainty makes decision making hard. Conversely, without uncertainty decisions would be easier. For example, if a cafe knew exactly 10 customers would want a bagel, then they could make exactly 10 bagels in the morning; if a factory's machine could make each part exactly the same, then quality control to ensure the part's fit could be eliminated; if a customer's future ability to pay back a loan was known, then a bank's decision to underwrite the loan would be quite simple.

In this chapter, we learn to represent our real-world uncertainty (e.g. demand for bagels, quality of a manufacturing process , or risk of loan default) in mathematical and computational terms. We start by defining the ideal mathematical way of representing uncertainty, namely by assigning a *probability distribution* to a *random variable*. Subsequently, we learn to describe our uncertainty in a *random variable* using *representative samples* as a pragmatic proxy to this mathematical ideal.

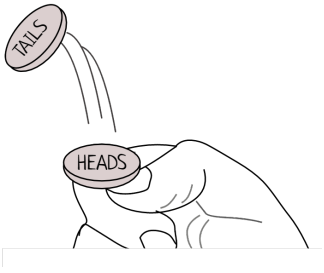


Figure 2.1: The outcome of a coin flip can be represented by a probability distribution.

2.1 Random Variables With Assigned Probability Distributions

Think of a random variable as a mapping of outcomes that interest us, like demand or risk, to numerical values representing the probability we assign to each event. For us business folk, we can often think of this mapping as a table with outcomes on the left and probabilities on the right; take this table of coin flip outcomes as an example:

Table 2.1: Table 2.1:Using a table to represent the probability distribution of a coin flip.

Outcome	Probability
HEADS	50%
TAILS	50%

While Table 2.1 might be adequate to describe the mapping of coin flip outcomes to probability, as we make more complex models of the real-world, we will want to take advantage of the concise (and often terse) notation that mathematicians would use. In addition, we want to gain fluency in *math world* notation so that we can successfully traverse the bridge between *real world* and *math world*. *Random variables* are the fundamental *math world* representation that create the foundation of our studies, so please **resist any temptation to not learn the subsequent mathematical notation**.

Mathematicians love using Greek letters, please do not be intimidated by them - they are just letters. You will learn lots of lowercase letters like α (alpha), β (beta), μ (mu), ω (omega), and σ (sigma). And also some of their uppercase versions like Ω (omega) as the uppercase of ω. See the whole list at wikipedia.org.

2.1.1 Some Math Notation for Random Variables

Above, a random variable was introduced as a mapping of outcomes to probabilities. And, this is how you should think of it most of the time. However, to start gaining fluency in the math-world definition of a random variable, we will also view this mapping process as not just one mapping, but rather a sequence of two mappings: 1) the first mapping is actually the "true" definition of a **random variable** in probability theory - it maps all possible outcomes to real numbers, and 2) the second mapping, known as a **probability distribution** in probability theory, maps the numbers from the first mapping to real numbers representing how plausibility is allocated across all possible outcomes - we often think of this allocation as assigning probability to each outcome.

FOR EXAMPLE, to define a coin flip as a random variable , start by listing the set of possible outcomes (by convention, the greek letter Ω is often used to represent this set and it is called the sample space):

$$\Omega = \{Heads, Tails\}.$$

Then pick an an uppercase letter, like X, to represent the random variable (i.e. an unobserved sample from Ω) and explictly state what it represents using a short description:

$$X \equiv \text{The outcome of a coin flip,}$$

where $\equiv$ is read "defined as".

Please note that the full mathematical formalism of random variables is not discussed here. For applied problems, thinking of a random variable as representing a space of possible outcomes governed by a probability distribution is more than sufficient.

The outcomes in a sample space must be 1) exhaustive i.e. include all possible outcomes and 2) mutually exclusive i.e. non-overlapping.

When real-world outcomes are not interpretable real numbers (e.g. *heads* and *tails*), define an explicit mapping of these outcomes to real numbers:

$$X \equiv \begin{cases} 0, & \text{if outcome is } Tails \\ 1, & \text{if outcome is } Heads \end{cases}$$

For, coin flip examples, it is customary to map *heads* to the number 1 and *tails* to the number 0. Thus, $X = 0$ is a concise way of saying

"the coin lands on *tails*" and likewise $X = 1$ means "the coin lands on *heads*". The terse math-world representation of a mapping process like this is denoted:

Mathematicians use the symbol $\mathbb{R}$ to represent the set of all real numbers.

$$X : \Omega \to \mathbb{R}$$

, where you interpret it as "random variable X maps each possible outcome in sample space omega to a real number."

The second mapping process then assigns a probability distribution to the random variable. By convention, lowercase letters, e.g. x, represent actual observed outcomes. We call x the *realization* of random variable X and define the mapping of outcomes to probability for every $x \in X$ (read as x "in" X and interpret it to mean "for each possible realization of random variable X"). As you would already guess, we have 100% confidence that one of the outcomes will be realized (e.g. *heads* or *tails*), so as such and by convention, we allocate 100% plausibility (or probability) among the possible outcomes. In this book, we will use f, to denote a function that maps each possible realization of a random variable to its corresponding plausibilty measure and use a subscript to disambiguate which random variable is being referred to when necessary. For the coin flip example, we can use our newly learned mapping notation to demonstrate this:

There are several conventions for representing this mapping function which takes a potential realization as input and provides a plausibility measure as output. This textbook uses $f_X(x)$ or more simply just $f(x)$, but other texts will use $\pi(x)$, $\Pr(X = x)$, or $p(x)$. Knowing that there is not just one standard convention will prove useful as you do your own research to find the right probability distribution to assign to a random variable. After a while, this change of notation becomess less frustrating.

$$f_X : X \to [0, 1],$$

where $[0, 1]$ is notation for a number on the interval from 0 to 1; the square brackets mean the interval is *closed* and hence, the mapping of an outcome to exactly 0 or 1 is possible.

Despite all this fancy notation, for small problems it is sometimes the best course of action to think of a random variable as a lookup table as shown here:

Table 2.2: Table 2.2: Probability distribution for random variable X represented as a table showing how real-world outcomes are mapped to real numbers and how 100% plausibility is allocated between all of the outcomes.

Outcome	Realization (x)	$f(x)$
HEADS	1	0.5
TAILS	0	0.5

and where $f(x)$ can be interpreted as the plausability assigned to random variable X taking on the value x. For example, $f(1) = 0.5$ means that $Pr(X = 1) = 50\%$ or equivalently that the probability of heads is 50%.

To reiterate how a random variable is a sequence of two mapping processes, notice that Table 2.2 has these features:

1. It defines a mapping from each real-world outcome to a real number.
2. It allocates plausability (or probability) to each possible realization such that we are 100% certain one of the listed outcomes will occur.

2.1.2 The Power of Abstraction

While the coin flip example may seem trivial, we are going to take that micro-example and abstract a little bit. As Josh Waitzkin advocates (Waitzkin, 2007), we should "learn the macro from the micro." Following this guiding principle, we will now look at modelling uncertain outcomes where there are two possibilities - like a coin flip, but now we assume the assigned probabilities do not have to be 50%/50%. Somewhat surprisingly, this small abstraction now places an enormous amount of real-world outcomes within our math-world modelling capabilities:

"[We will] dive deeply into small pools of information in order to explore and experience the operating principles of whatever we are learning. Once we grasp the essence of our subject through focused study of core principles, we can build on nuanced insights and, eventually, see a much bigger picture. The essence of this approach is to study the micro in order to learn what makes the macro tick." - Josh Waitzkin

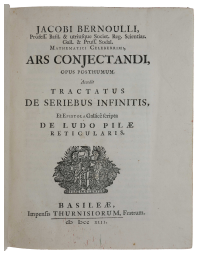

JACOBI BERNOULLI,
ARS CONJECTANDI,
OPUS POSTHUMUM.
TRACTATUS
DE SERIEBUS INFINITIS,
DE LUDO PILÆ RETICULARIS.
BASILEÆ,

Figure 2.2: Ars Conjectandl - Jacob Bernoulli's post-humously published book (1713) included the work after which the notable probability distribution - the Bernoulli distribution - was named.

- Will the user click my ad?
- Will the drug lower a patient's cholesterol?
- Will the new store layout increase sales?
- Will the well yield oil?
- Will the customer pay back their loan?
- Will the passenger show up for their flight?
- Is this credit card transaction fraudulent?

The Bernoulli distribution, introduced in 1713 (see Figure 2.2), is a probability distribution used for random variables of the following form:

Table 2.3: Table 2.3: If X follows a Bernoulli distribution, then the following lookup table describes the mapping process.

Outcome	Realization (x)	$f(x)$
Failure	0	$1-p$
Success	1	p

In Table 2.3, p is called a **parameter** of the Bernoulli distribution. Given the parameter(s) of any probability distribution, you can say everything there is to know about a random variable following that distribution; this includes the ability to know all possible outcomes as well as their likelihood. For example, if X follows a Bernoulli distribution with $p = 0.25$, then you know that X can take the value of 0 or 1, that $\Pr(X = 0) = 0.75$, and lastly that $\Pr(X = 1) = 0.25$

where p represents the probability of success - notice that the following must hold to avoid non-sensical probability allocations $0 \leq p \leq 1$.

With all of this background, we are now equipped to model uncertainty in any observable data that has two outcomes. The way we will represent this uncertainty is using two forms: 1) a graphical model

and 2) a statistical model. The graphical model is simply the random variable oval:

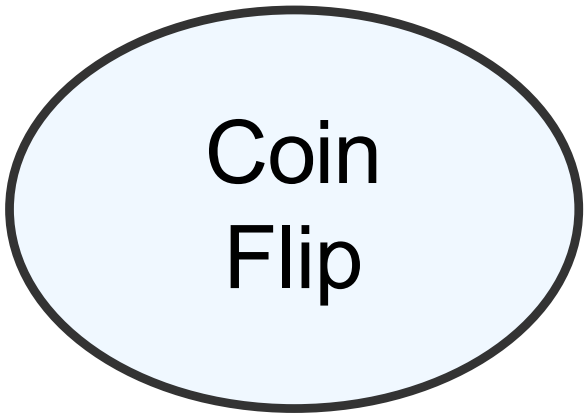


Figure 2.3: This oval represents a random variable indicating the result of a coin flip.

And, the statistical model is represented like this:

$$X \equiv \text{Coin flip outcome with heads} = 1 \text{ and tails} = 0.$$
$$X \sim \text{Bernoulli}(p)$$

where $\sim$ is read "is distributed"; so you might say "X is distributed Bernoulli with parameter p."

We will see in future chapters that the graphical model and statistical models are more intimately linked than is shown here, but for now, suffice to say that the graphical model is good for communicating uncertainty to stakeholders more grounded in the real-world and the statistical model is better for communicating with stakeholders in the math-world.

2.2 Representative Samples

Despite our ability to represent probability distributions using precise mathematics, uncertainty modelling in the practical world is always an approximation. Does a coin truly land on heads 50% of the time (see `https://en.wikipedia.org/wiki/Checking_whether_a_coin_is_fair`)? Its hard to tell. One might ask, how many times must we flip a coin to be sure? The answer might surprise you; it could take over 1 million tosses to reach an estimate that a coin lies within 0.1% of the observed proportion of heads. That is a lot of tosses. So in the real world, seeking that level of accuracy becomes impractical. Rather, we are seeking a model that is good enough; a model where we believe in its insights and are willing to follow through with its recommendations.

Instead of working with probability distributions directly as *mathematical* objects, we will most often seek a representative sample and

A **representative sample** is an incomplete collection or subset of data that exhibits a specific type of similarity to a complete collection of data from an entire (possibly infinite) population. For our purposes, the similarity criteria requires that an outcome drawn from either the sample or the population is drawn with similar probability.

treat them as *computational* objects (i.e. **data**). For modelling a coin flip, the representative sample might simply be a list of *heads* and *tails* generated by someone flipping a coin or by a computer simulating someone flipping a coin.

Turning a mathematical object into a representative sample using R is quite easy as R (and available packages) can be used to generate random outcomes from just about all well-known probability distributions. To generate samples from a random Bernoulli variable, we use the `rbern` function from the `causact` package:

```
# The rbern function is in the causact package
# so make the causact package available in this session
library(causact) # package install from http://github.com/flyaflya/causact
# rbern is a function that takes two arguments:
# 1) n is the number of trials (aka coin flips)
# 2) prob is the probability of success (aka the coin lands on heads)
set.seed(123) #ensure that all computers generate the same random output
rbern(n=7,prob=0.5)
```

```
## [1] 0 1 0 1 1 0 1
```

where the 3 `0`'s and 4 `1`'s are the result of the `n=7` coin flips.

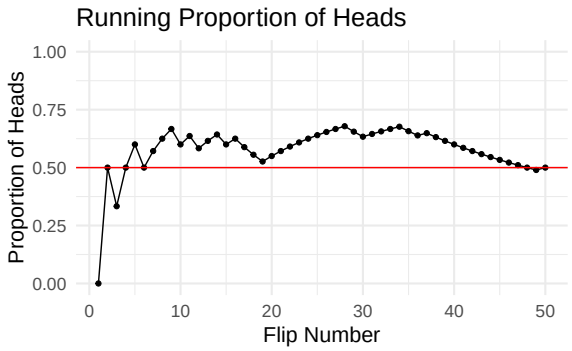
Notice that one might be reluctant to label this a *representative* sample as the proportion of `1`'s is 0.5714286 and not the `0.5` that a representative sample would be designed to mimic. In fact, we can write code to visualize the proportion of heads as a function of the number of coin flips:

```
library(dplyr)
library(ggplot2)
set.seed(123) #ensure that all computers generate the same random output

# Create dataframe of coinflip observations
numFlips = 50 ## flip the coin 100 times
df = data.frame(flipNum = 1:numFlips,
                coinFlip = rbern(n=numFlips,prob=0.5)) %>%
  mutate(headsProportion = cummean(coinFlip))

# Plot results
ggplot(df, aes(x = flipNum, y = headsProportion)) +
  geom_point() +
  geom_line() +
  geom_hline(yintercept = 0.5, color = "red") +
```

```
  ggtitle("Running Proportion of Heads") +
  xlab("Flip Number") +
  ylab("Proportion of Heads") +
  ylim(c(0,1))
```



Notice that even after 30 coin flips, the sample is far from representative as the proportion of heads is 0.6333333. Even after 1,000 coin flips (i.e. `numFlips = 1000`), the proportion of heads `0.493` is still just an approximation as it is not exactly 0.5.

Well if 1,000 coin flips gets us an approximation close to 0.5, then 10,000 coin flips should get us even closer. To explore this idea, we generate ten simulations of 10,000 coin flips and print out the average proportion of heads for each:

```
set.seed(123)
for (i in 1:10){
  proportionOfHeads = mean(rbern(n=10000,prob=0.5))
  print(proportionOfHeads)
}
```

```
## [1] 0.4943
## [1] 0.4902
## [1] 0.4975
## [1] 0.4888
## [1] 0.4997
## [1] 0.5006
## [1] 0.4954
## [1] 0.4982
## [1] 0.5083
## [1] 0.5011
```

Notice that the average distance away from the exact proportion 0.5 is 0.459%. So on average, it appears we are around 0.5% away from the true value. This is the reality of representative samples; they

will prove enormously useful, but are still just approximations of the underlying mathmatical object - in this case, $X \sim$ Bernoulli(0.5). If this approximation bothers you, remember the mathematical object is just an approximation of the real-world object. Might it be possible that certain coins are weighted in one way or another to deviate - even ever so slightly - from the ideal? Of course, but it does not mean the approximations are useless … on the contrary, we will see how powerful the math-world and computation-world can be in bringing real-world insight.

2.2.1 Generating representative samples

So far, we have represented uncertainty in a simple coin flip - using both the mathematics of random variables and just using a sample of data. As we try to model more complex aspects of the business world, we will use our simple building blocks to build models of ever-increasing complexity.

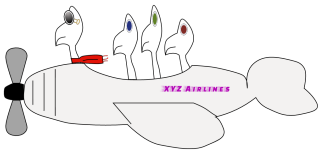


Figure 2.4: How many passengers will show up if XYZ Airlines accepts three reservations.

Example 2.1. The XYZ Airlines company owns the one plane shown in Figure 2.4. XYZ operates a 3-seater airplane to show tourists the Great Barrier Reef in Cairns, Australia. The company uses a reservation system, wherein tourists call in advance and make a reservation for aerial viewing the following day. Unfortunately, often passengers holding a reservation might not show up for their flight. Assume that the probability of each passenger not showing up for a flight is 15% and that each passenger's arrival probability is independent of the other passengers. Assuming XYZ takes three reservations, use your ability to simulate the Bernoulli distribution to estimate a random variable representing the number of passengers that show up for the flight.

To solve Example 2.1, we model the data generating process using both a graphical model and a statistical model. The graphical model would look like this:

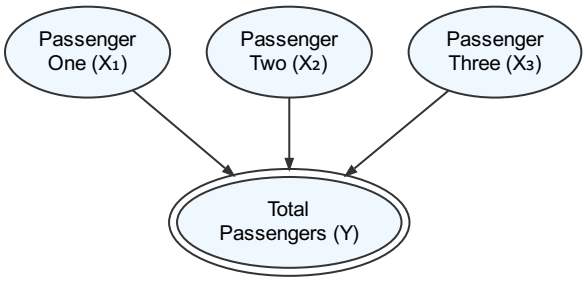


Figure 2.5: The graphical model of number of passengers who show up for the XYZ airlines tour.

And, the statistical model is represented like this:

$X_i \equiv$ If passenger i shows up, then $X = 1$. Otherwise, $X = 0$. Note: $i \in \{1, 2, 3\}$.

$X_i \sim \text{Bernoulli}(p = 0.85)$

$Y \equiv$ Total number of passengers that show up.

$Y = X_1 + X_2 + X_3$

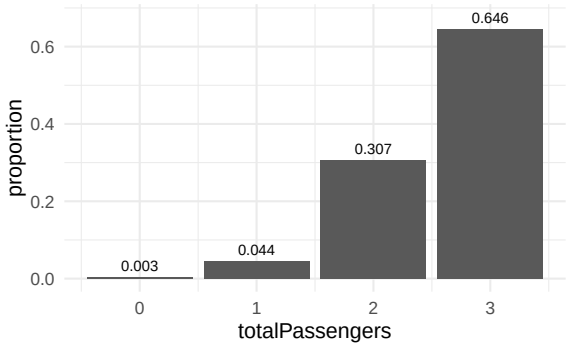
The last line gives us a path to generate a representative sample of the number of passengers who show up for the flight; we simulate three Bernoulli trials and add up the result. Computationally, we can create a data frame to simulate as many flights as we want. Let's simulate 1,000 flights and see the probabilities associated with Y:

```
library(causact)
numFlights = 1000 ## number of simulated flights
probShow = 0.85 ## probability of passenger showing up
set.seed(111) ## choose random seed so others can replicate results
pass1 = rbern(n = numFlights, prob = probShow)
pass2 = rbern(n = numFlights, prob = probShow)
pass3 = rbern(n = numFlights, prob = probShow)

# create data frame (use tibble to from tidyverse)
flightDF = tibble(
  simNum = 1:numFlights,
  totalPassengers = pass1 + pass2 + pass3
)

# transform data to give proportion
propDF = flightDF %>% group_by(totalPassengers) %>% summarize(numObserved = n()) %>%
  mutate(proportion = numObserved / sum(numObserved))

# plot data with estimates
ggplot(propDF, aes(x = totalPassengers, y = proportion)) +
  geom_col() +
  geom_text(aes(label = proportion), nudge_y = 0.03)
```



Wow, that was pretty cool. We created a representative sample for a random variable whose distribution was not Bernoulli, but could be constructed as the sum of three Bernoulli random variables. We can now answer questions like "what is the probability there is at least one empty seat?" This is the same as saying what is $\Pr(Y <= 2)$ or equivalently $1 - \Pr(Y = 3)$. And the answer, albeit an approximate answer, is 0.354.

You may be wondering how much the approximated probabilities for the number of passengers might vary with a different simulation. The best way to find out is to try it again. Remember to eliminate or change the `set.seed` function prior to trying the simulation again.

2.3 Mathematics As A Simulation Shortcut

Simulation will always be your friend in the sense that if given enough time, a simulation will always give you results that approximate mathematical exactness. The only problem with this friend is it is sometimes slow to yield representative results. In these cases, sometimes mathematics provides a shortcut. For example, mathematicians realized that just one Bernoulli trial is sort of uninteresting (would you predict the next president by polling just one person?). Enter the **binomial distribution**.

2.3.1 The Binomial Distribution

The binomial distribution is a two-parameter distribution and models scenarios where we are interested in something like the number of heads in multiple coin flips or the number of passengers that arrive given three reservations. More formally, a binomially distributed random variable (let's call it X) represents the number of successes in n Bernoulli trials where each trial has success probability p.

The two parameters of a binomial distribution map to the real-world in a fairly intuitive manner. The first parameter, n, is simply the number of Bernoulli trials your random variable will model. The second parameter, p, is the probability of observing success on each trial. So if $X \equiv$ number of heads in 10 coin tosses and $X \sim \text{Binomial}(n = 10, p = 0.5)$, then an outcome of $x = 4$ means that four heads were observed in 10 coin flips.

Going back to our airplane example (Example 2.1), we can take advantage of the mathematical shortcut provided by the binomial distribution and use the following graphical/statistical model combination to yield exact results.

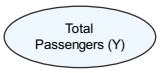


Figure 2.6: The graphical model of number of passengers who show up for the XYZ airlines tour.

And, the statistical model is represented like this:

$$Y \equiv \text{Total number of passengers that show up.}$$
$$Y \sim \text{Binomial}(n = 3, p = 0.85)$$

This more compact representation combined with the power of R can now yield the exact probability distribution of Y; we just need to know the right function to use. More generally, functions for probability distributions in R will follow the following syntax:

`foo` is called a placeholder name in computer programming. The word `foo` itself is meaningless, but you will substitute more meaningful words in its place. In the examples here, `foo` will be replaced by an abrreviated probability distribution name like `binom` or `norm`.

- `dfoo` - is the probability mass function (discrete) or the probability density function (continuous). For **discrete** random variables, this is $\Pr(X = x)$. For continuous random variables, this number is less helpful (see this Khan Academy video for more background

information). Corresponding math notation for this function is $f(x)$.

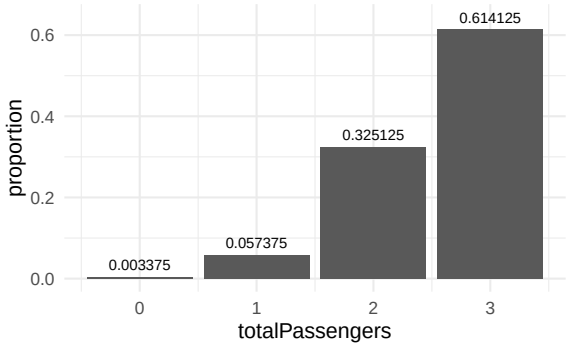
- `pfoo` - is the cumulative distribution function. User inputs q and parameters of the distribution, this returns a probability p such that $\Pr(X \leq q) = p$. Corresponding math notation for this function is $F(q)$.
- `qfoo` - is the quantile function. User inputs p and parameters of the distribution, this returns the realization value q such that $Pr(X \leq q) = p$. Corresponding math notation for this function is $F^{-1}(p)$.
- `rfoo` - is the random generation function. User inputs n and the distribution parameters, this returns n random observations of the random variable.

Since we are interested in the binomial distribution, we can replace `foo` by `binom` to take advantage of the functions for probability distributions listed above. For example, to answer "what is the probability there is at least one empty seat?" We find $1 - \Pr(Y = 3)$ which is the same as `1 - dbinom(x=3, size = 3, prob = 0.85)`. And the exact answer is 0.385875 and close, but not identical, to our previously approximated answer of 0.354. Note, we could have chosen to use the CDF instead of the PDF to answer this question by finding $\Pr(Y \leq 2)$ using `pbinom(q=2, size = 3, prob = 0.85)`. To reproduce our approximated results using the exact distribution, we can use the following code:

Take notice of the transformatiion from the math world to the computation world. In the math world, we might say $Y \sim \text{Binomial}(n = 3, p = 0.85)$. But in the computation world of R, n is replaced by the `size` argument and p is replaced by the `prob` argument. Also notice that `n` is an argument of the `rfoo` function, but it is not the same as the math-world n. In the computer-world `n` is the number of random observations of a specified distribution that you want generated. So if you wanted 10 samples of Y, you would use the function `rbinom(n=10,size=3,prob=0.85)`. Be careful when doing these translations.

```
# transform data to give proportion
propExactDF = tibble(totalPassengers = 0:3) %>%
  mutate(proportion = dbinom(x = totalPassengers,
                             size = 3,
                             prob = 0.85))

# plot data with estimates
ggplot(propExactDF, aes(x = totalPassengers, y = proportion)) +
  geom_col() +
  geom_text(aes(label = proportion), nudge_y = 0.03)
```



The above code is both simpler and faster than the approximation code run earlier. In addition, it gives exact results. Hence, when we can take mathematical shortcuts, we will to save time and reduce the uncertainty in our results introduced by approximation error.

2.4 Big Picture Takeaways

This chapter is our first foray into representing uncertainty. Our representation of uncertainty takes place in three worlds: 1) the real-world - we use graphical models (i.e. ovals) to convey the story of uncertainty, 2) the math-world - we use statistical models to rigorously define how random outcomes are generated, and 3) the computation-world - we use R functions to answer questions about exact distributions and representative samples to answer questions when the exact distribution is unobtainable. As we navigate this course, we will traverse across these worlds and learn to translate from one world's representation of uncertainty to another's. In the next chapter, we will formalize how data can play an active role in reducing our uncertainty.

2.5 Getting Help

As mentioned previously, Google and YouTube are great resources to supplement, reenforce, or further explore topics covered in this book. For the mathematical notation and conventions regarding random variables, I highly recommend listening to Sal Khan, founder of Khan Academy, for a more thorough introduction/review of these concepts. Sal's videos can be found at `https://www.khanacademy.org/math/statistics-probability/random-variables-stats-library`.

3
Joint Distributions Tell You Everything

3.1 Joint Distributions

The most complete method of reasoning about sets of random variables is by having a *joint probability distribution*. A joint probability distribution, $\mathcal{P}(X_1, \ldots, X_n)$, assigns a probability value to all possible assignments or realizations of sets of random variables. The goal of this chapter is to 1) introduce you to the notation of joint probability distributions and 2) convince you that if you are given a joint probability distribution that you would be able to answer some very useful questions related to probability.

Consider the graphical model from Shenoy and Shenoy (2000) and depicted in Figure 3.1.

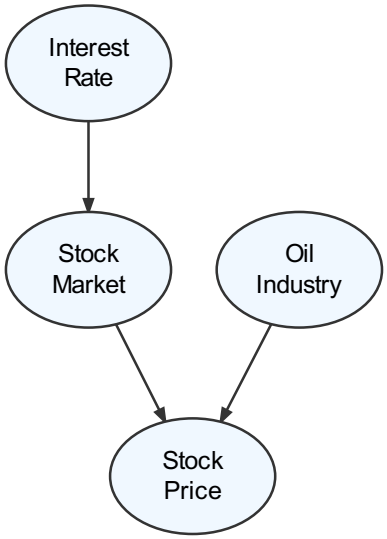


Figure 3.1: Model of how stock prices for oil companies are influenced by other factors.

In the diagram, there are four random variables: 1) *Interest Rate* (IR), 2) *Stock Market* (SM), 3) *Oil Industry* (OI), and 4) *Stock Price* (SP) (assume for an oil company). For simplicity and to gain intuition about joint distributions, assume that each of these four random variables is binary-valued, meaning they can each take two possible assignments:

Random Variable (X_i)	Set of Possible Values (i.e. $Val(X_i)$)
IR	$high, low$
SM	$good, bad$
OI	$good, bad$
SP	$high, low$

Thus, our probability space has $2 \times 2 \times 2 \times 2 = 16$ values corresponding to the possible assignments to these four variables. So, a joint distribution must be able to assign probability to these 16 combinations. Here is one possible joint distribution:

IR	SM	OI	SP	$P(IR, SM, OI, SP)$
high	*good*	*good*	*high*	0.016
low	*good*	*good*	*high*	0.168
high	*bad*	*good*	*high*	0.04
low	*bad*	*good*	*high*	0.045
high	*good*	*bad*	*high*	0.018
low	*good*	*bad*	*high*	0.189
high	*bad*	*bad*	*high*	0.012
low	*bad*	*bad*	*high*	0.0135
high	*good*	*good*	*low*	0.004
low	*good*	*good*	*low*	0.042
high	*bad*	*good*	*low*	0.04
low	*bad*	*good*	*low*	0.045
high	*good*	*bad*	*low*	0.012
low	*good*	*bad*	*low*	0.126
high	*bad*	*bad*	*low*	0.108
low	*bad*	*bad*	*low*	0.1215

Collectively, the above 16 probabilities represent the *joint distribution* $P(IR, SM, OI, SP)$ - meaning, you plug in values for all four random variables and it gives you a probability. For example, $P(IR = low, SM = bad, OI = bad, SP = low)$ yields a probability assignment of 12.15%.

3.2 Marginal Distributions

One might also be curious about probability assignments for just a subset of the random variables. This smaller subset of variables can be called *marginal variables* and their probability distribution is called a *marginal distribution*. For example, the marginal distribution for *oil industry* (OI) is notated as $P(OI)$ and represents a probability distribution over just one of the four variables - ignoring the others. The marginal distribution can be derived from the joint distribution using the formula:

$$P(OI = x) = \sum_{i \in IR, j \in SM, k \in SP} (P(OI = x, IR = i, SM = j, SP = k))$$

Applying the above formula to determine the marginal distribution of OI yields the following tabular representation of the marginal distribution:

More generally speaking, a marginal distribution is a compression of information where only information regarding the marginal variables is maintained. Take a set of random variables, X (e.g. $\{IR, SM, OI, SP\}$), and a subset of those variables Y (e.g. $\{OI\}$). And using standard mathematical convention, let $Z = X \setminus Y$ be the set of random variables in X that are not in Y (i.e. $Z = \{IR, SM, SP\}$. Assuming discrete random variables, then the marginal distribution $P(Y)$ is calculated from the joint distribution $P(Y) = \sum_Z P(Y = y, Z = z)$. Effectively, when the joint probability distribution is in tabular form, one just sums up the probabilities in each row where $Y = y$.

Think of a marginal distribution as a function of the marginal variables. Given realizations of the marginal variables, the function returns a probability.

Realization (x)	$P(OI = x)$
Good	$0.016 + 0.168 + 0.04 + 0.045 + 0.004 + 0.042 + 0.04 + 0.045 = \mathbf{0.4}$
Bad	$0.018 + 0.189 + 0.012 + 0.0135 + 0.012 + 0.126 + 0.108 + 0.1215 = \mathbf{0.6}$

EXERCISE1: Suppose we are only interested in the Oil Company Stock Price (SP). Given the probabilities in the above joint distribution, what is the *marginal distribution* for SP (i.e. $P(SP = High)$ and $P(SP = Low)$)?

EXERCISE2: Suppose we are interested in both the Stock Market (SM) and the Oil Industry (OI). We can find the marginal distribution for these two variables, $P(SM, OI)$. This is sometimes called a *joint marginal distribution*; joint referring to the presence of multiple variables and marginal referring to the notion that this is a subset of the original joint distribution. So, given the probabilities in the above joint distribution, what is the *marginal distribution* for $\{SM, OI\}$ - i.e. a probability function for the four possible realizations:

1. $P(SM = good, OI = good)$,
2. $P(SM = bad, OI = good)$,
3. $P(SM = good, OI = bad)$, and
4. $P(SM = bad, OI = bad)$)?

3.3 Conditional Distributions

Conditional distributions can be used to model scenarios where a subset of the random variables are known (e.g. data) and the remaining subset is of interest (e.g. model parameters). For example, we might be interested in getting the conditional distribution of *Stock Price* (SP) given that *Interest Rate* is *high*. The notation for this is $P(SP|IR = high)$ and can be retrieved in a two-step process:

1. First, to simplify the problem, we get the marginal distribution for $\{SP, IR\}$ and rid ourselves of the variables that we are not interested in:

IR	SP	$P(IR, SP)$
high	*high*	0.086
low	*high*	0.4155
high	*low*	0.164
low	*low*	0.3345

2. Then, we calculate the conditional distribution using Bayes rule:

$$\begin{aligned} P(SP = high|IR = high) &= \frac{P(SP = high, IR = high)}{P(IR = high)} \\ &= \frac{P(SP = high, IR = high)}{P(SP = high, IR = high) + P(SP = low, IR = high)} \\ &= \frac{0.086}{0.086 + 0.164} \\ &= 0.344 \end{aligned}$$

and,

$$\begin{aligned} P(SP = low|IR = high) &= \frac{P(SP = low, IR = high)}{P(IR = high)} \\ &= \frac{P(SP = low, IR = high)}{P(SP = high, IR = high) + P(SP = low, IR = high)} \\ &= \frac{0.164}{0.086 + 0.164} \\ &= 0.656 \end{aligned}$$

which yields the following tabular representation of the conditional distribution for $P(SP = x|IR = high)$:

x	$P(SP = x\|IR = high)$
$high$	0.344
low	0.656

EXERCISE3: Now, suppose we learn that $InterestRate$ is low. What is the conditional distribution for SP (i.e. $P(SP = High|IR = low)$ and $P(SP = Low|IR = low)$)?

3.4 MAP Estimates

Sometimes, we are not interested in a complete probability distribution, but rather seek a high-probability assignment to some subset of variables. For this, we can use a MAP *query* (maximum a posteriori query). A MAP finds the most likely assignment of all non-evidentiary variables (i.e. unknown values). Basically, you search the joint distribution for the largest probability value. For example, the maximum a posterior estimate of stock price given $IR = high$ would be given by the following formula:

$\arg\max_{x \in SP} P(SP = x|IR = high)$

which in natural language asks for the *argument* (i.e. the realization of stock price) that maximizes the conditional probability $P(SP =$

$x|IR = high$). From above, we realize that $P(SP = high|IR = high) = 0.344$ and $P(SP = low|IR = high) = 0.656$ and hence, the MAP estimate is that $SP = low$ because $0.656 > 0.344$.

EXERCISE4: Let $\mathbf{X} = \{IR, SM, OI, SP\}$, what is $\text{MAP}(\mathbf{X}) = \arg\max_{\mathbf{x}} P(\mathbf{x})$, the most likely joint assignment?

EXERCISE5: Now, suppose we learn that *StockPrice* is low. Let $\mathbf{Y} = \{IR, SM, OI\}$, what is $\text{MAP}(\mathbf{Y}|SP = low) = \arg\max_{\mathbf{y}} P(\mathbf{y}|SP = low)$, the most likely joint assignment??

3.5 Limitations of Joint Distributions

WHY DON'T WE JUST USE JOINT PROBABILITY DISTRIBUTIONS ALL THE TIME? Despite the expressive power of having a joint probability distribution, they are not that easy to directly construct due to the *curse of dimensionality*. As the number of random variables being considered in a dataset grows, the number of potential probability assignments grows too. Even in the era of big data, this curse of dimensionality still exists. Generally speaking, an exponential increase is required in the size of the dataset as each new descriptive feature is added. Let's assume we have n random variables with each having k values. Thus, the joint distribution requires k^n probabilites. Even if $k = 2$ and $n = 34$, this leads to 17,179,869,184 possibilities (over 17 billion). To make this concrete, a typical car purchase decision might easily look at 34 different variables (e.g. make, model, color, style, financing, etc.). So, to model this decision would require a very large joint distribution which actually dwarfs the amount of data that is available. As a point of comparison, just under 100 million motor vehicles were sold worldwide in 2017 - i.e. less than one data point per possible combination of features. Despite this "curse", we will learn to get around it with more compact representations of joint distributions. These representations will require less data, but will still yield the power to answer queries of interest; just as if we had access to the full joint distribution.

4
Bibliography

Shenoy, C. and Shenoy, P. P. (2000). Bayesian network models of portfolio risk and return. The MIT Press.

Waitzkin, J. (2007). *The art of learning: A journey in the pursuit of excellence*. Simon and Schuster.