```
begin
    import Pkg
    # activate the shared project environment
    Pkg.activate(Base.current_project())
    using Omega, Distributions, UnicodePlots, OmegaExamples
end
```

```
Activating project at `~/Documents/GitHub/Omega.jl/OmegaExamples`
```

As we noted in an earlier chapter, there is an interesting parallel between `randsample`, which separates model specification from inference methods. For most of this book we are interested in the *computational* level of describing what people know about the world and what inferences that knowledge licenses. That is, we treat the variables as the scientific hypothesis, and the optional (including `alg`) arguments as an engineering detail needed to derive predictions. We can make a great deal of progress with this abstraction.

Is it reasonable to interpret the inference algorithms that we borrow from statistics as psychological hypotheses at the algorithmic level? Which *algorithm* does the brain use for inference? Could it be MCMC? Enumeration?

If we take the algorithms for inference as psychological hypotheses, then the approximation and resource-usage characteristics of the algorithms will be the signature phenomena of interest.

How is uncertainty represented?

A signature of probabilistic ("Bayesian") cognitive models is the central role of uncertainty. Generative models, our main notion of knowledge, capture uncertain causal processes. After making observations or assumptions, conditional random variables capture uncertain answers. At the computational level we work with this uncertainty by manipulating variables in Omega, without needing to explore (much) how they are created or represented. Yet cognitively there is a key algorithmic question: how is uncertainty represented in the human mind?

We have at least three very different possible answers to this question:

- Explicit representation of probabilities.
- Parametric representation of distribution families.
- Sampling-based representations.

Explicit representations

The most straightforward interpretation of Bayesian models at the algorithmic level is that explicit probabilities for different states are computed and represented. Attempts have been made to model how neural systems might capture these representations, via ideas such as probabilistic population codes. (See Bayesian inference with probabilistic population codes, Ma, Beck, Latham, Pouget (2006).)

Yet it is difficult to see how these explicit representations can scale up to real-world cognition.

Approximate distribution representations

Another possible representation of uncertainty is via the parameters of a family of distributions. For instance, the mean and covariance of a Gaussian is a flexible and popular (in statistics) way to approximate a complex distirbution. It is thus possible that all uncertainty is represented in the human mind as parameters of some family. A version of this idea can be seen in the free energy hypothesis. (See The free-energy principle: a unified brain theory?, Friston (2010).)

The sampling hypothesis

Finally, it is possible that there is no explicit representation of uncertainty in the human mind. Instead, uncertainty is implicitly represented in the tendencies of a dynamical system. This is the sampling hypothesis: that the human mind has the ability to generate samples from conditional distributions when needed. Thus the mind implicitly represents an entire distribution, but can only work explicitly with a few samples from it.

We have seen a number of methods for creating dynamical systems that can sample from any desired distribution (rejection sampling, various Markov chain Monte Carlo methods, etc). This type of representation is thus possible in principle. What behavioural or neural signatures would we expect if it were correct? And *which* of the many sampling methods might be neurally plausible?

As a first analysis, we can assume that the human mind is always capable of drawing perfect samples from the conditional distributions of interest (e.g. via rejection sampling) but doing so is costly in terms of time or energy. If we assume that only a few samples are going to be used by any individual in answering any question, a profound behavioural prediction arises: individuals' choice probability will match the posterior probability distribution. Note that this is a somewhat radical departure from a "fully Bayesian" agent. If we assume a choice is to be made, with 100$ reward for the correct answer and no reward for incorrect answers, a rational (that is utility-maximizing) agent that explicitly represents the answer distribution should *always* choose the most likely answer. Across a population of such agents we would see no variation in answers (assuming a priori identical beliefs). In contrast, an agent that has only a single sample from their answer distribution will choose this answer; across a population of such agents we would see answers distributed according to the answer distribution!

Let's see this in practice in a simple example, where each person sees a coin of unknown probability flipped five times, coming up heads four of them. Each person is asked to bet on the next outcome:

```
weight =   StdUniform()
```

```julia
weight = StdUniform{Float64}()
```

```
agent_belief (generic function with 1 method)
```

```julia
function agent_belief(i, ω)
    w = (i ~ weight)(ω)
    a =  ((@uid, i...)~ Bernoulli(w)) |ᶜ (Variable(ω -> (i~ Binomial(5, w))(ω)) .== 4)
    a(ω)
end
```

```
agent_belief_samples (generic function with 2 methods)
```

```julia
agent_belief_samples(i, N_samples=100) = randsample(ω -> agent_belief((@uid, i), ω), N_samples)
```

```
max_agent (generic function with 2 methods)
```

```julia
max_agent(i, N_samples=100) =
    count(agent_belief_samples(i, N_samples)) > N_samples/2
```

```
max_agent_samples =
 [true, true, true, true, true, true, true, true, true, true, true, true, true, true, true, tru
```

```julia
max_agent_samples = map(i -> max_agent(i), 1:100)
```

```
        ┌                                        ┐
   true ┤■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■ 100
        └                                        ┘
```

```julia
viz(max_agent_samples)
```

```
sample_agent =
 [false, true, true, false, true, false, false, true, true, true, false, false, true, false, f
```

```julia
sample_agent = map(i -> randsample(ω -> agent_belief((@uid, i), ω)), 1:100) # assigns a new uid
```

```
         ┌                                        ┐
   false ┤■■■■■■■■■■■■■■■■■■■■ 36
    true ┤■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■ 64
         └                                        ┘
```

```julia
viz(sample_agent)
```

The maximizing agent chooses the most likely outcome by examining the conditional probability they assign to outcomes – the result is all such agents choosing 'true'. In contrast, a population of agents that each represents their belief with a single sample will choose 'false' about 13% of the time. This behavioral signature – *probability matching* – is in fact a very old and well studied psychological phenomenon. (See for instance, Individual Choice Behavior: A Theoretical Analysis, Luce (1959).)

Vul, Goodman, Griffiths, Tenenbaum (2014) further ask how many samples a rational agent *should* use, if they are costly. This analysis explores the trade off between expected reward increase from more precise probability estimates (more samples) with resource savings from less work (fewer samples). The, somewhat surprising, result is that for a wide range of cost and reward assumptions it is optimal to decide based on only one, or a few, samples.