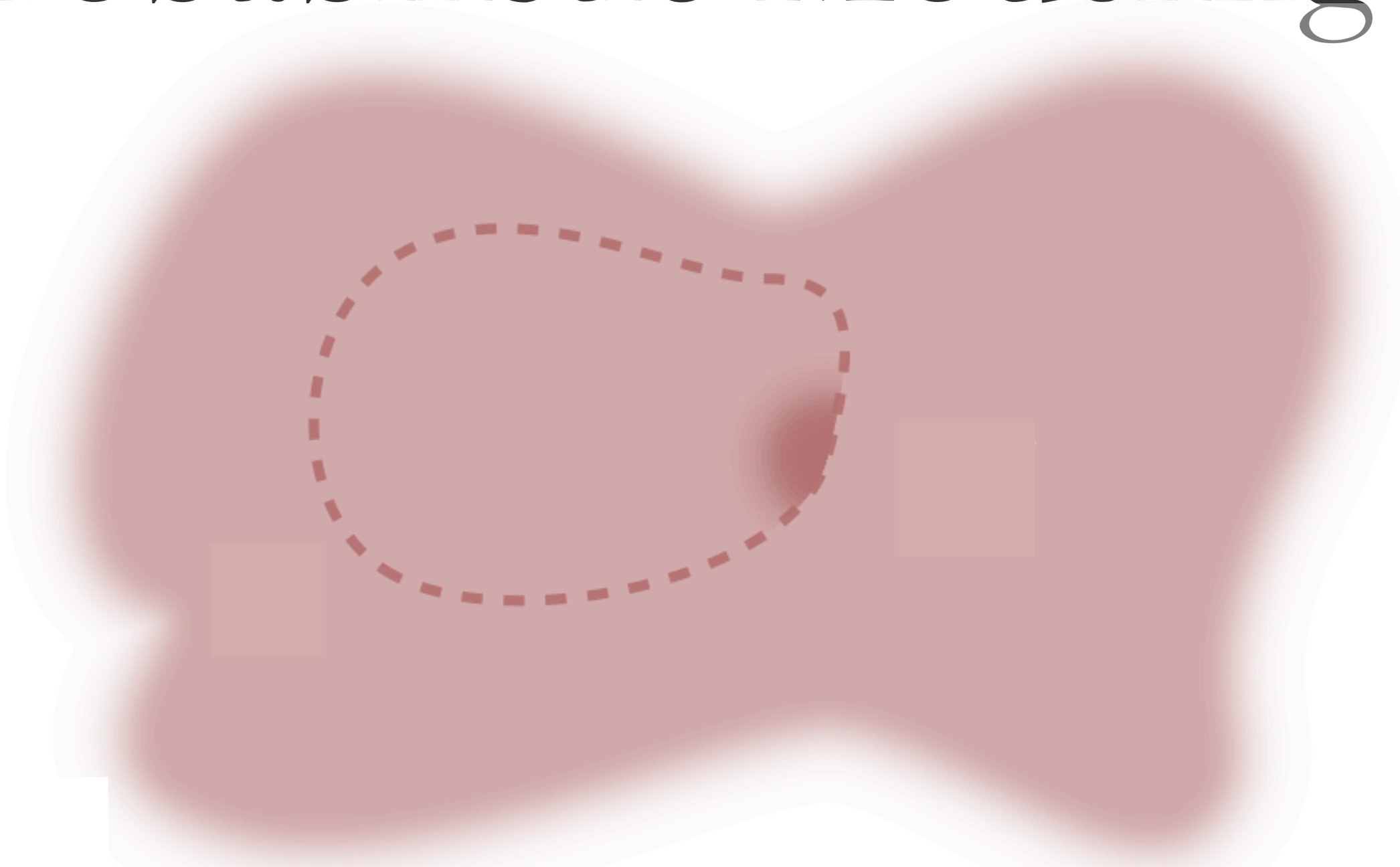
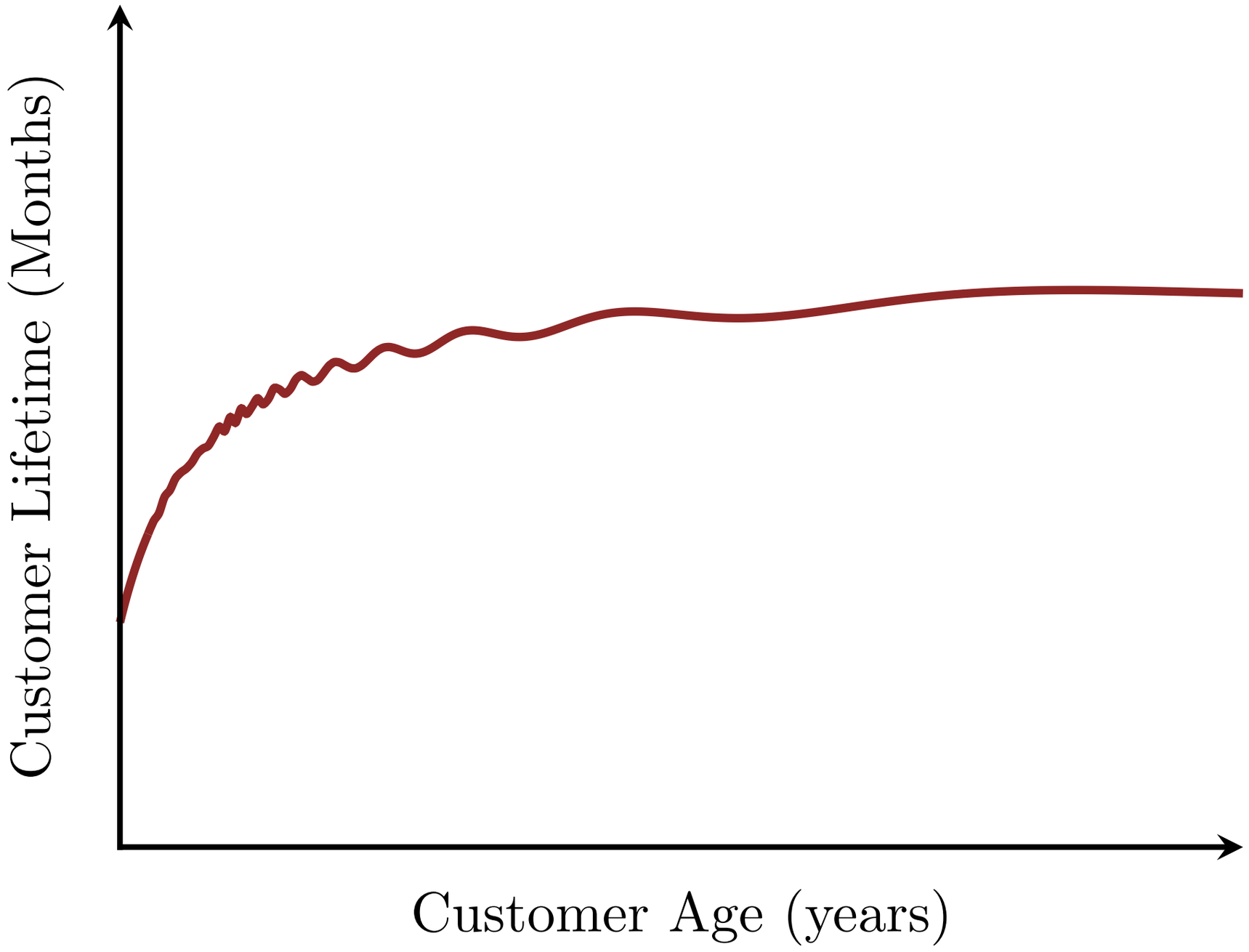
Probabilistic Modeling



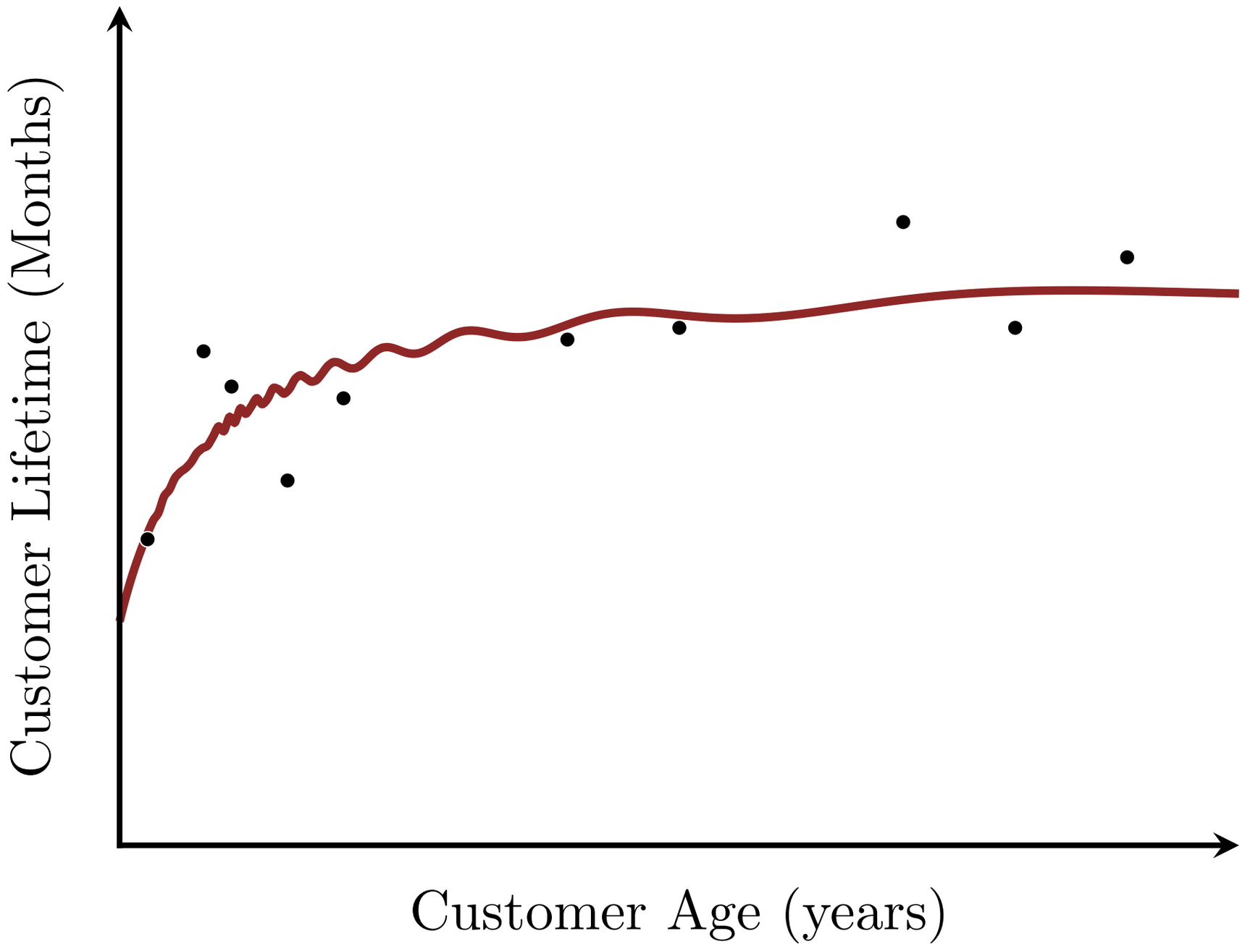
Michael Betancourt
Symplectomorphic, LLC
http://symplectomorphic.com

KBC Group ADAM Bootcamp
Gyöngyös, Hungary
May 20, 2026

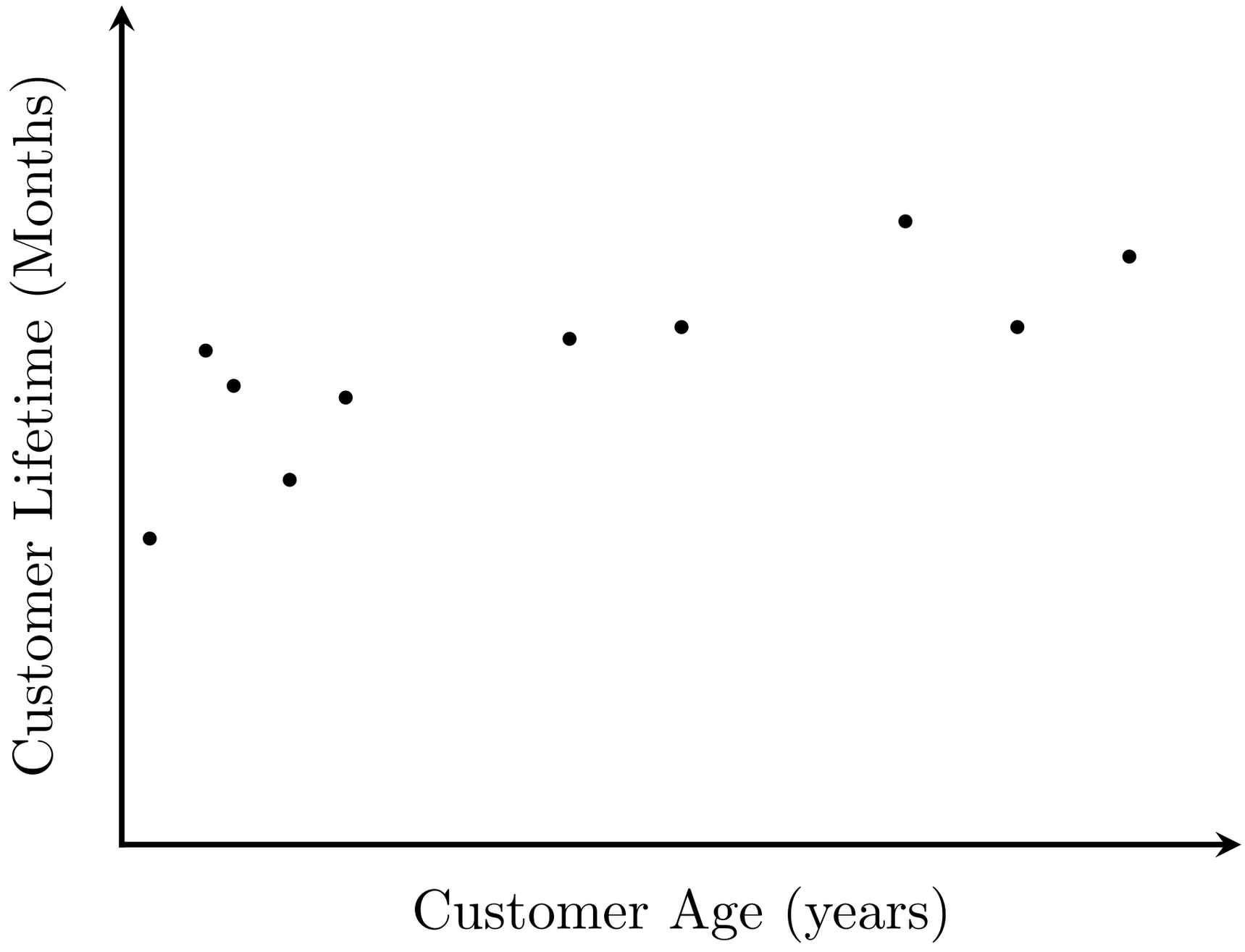
How might the lifetime of a bank customer vary with customer demographics, such as age?



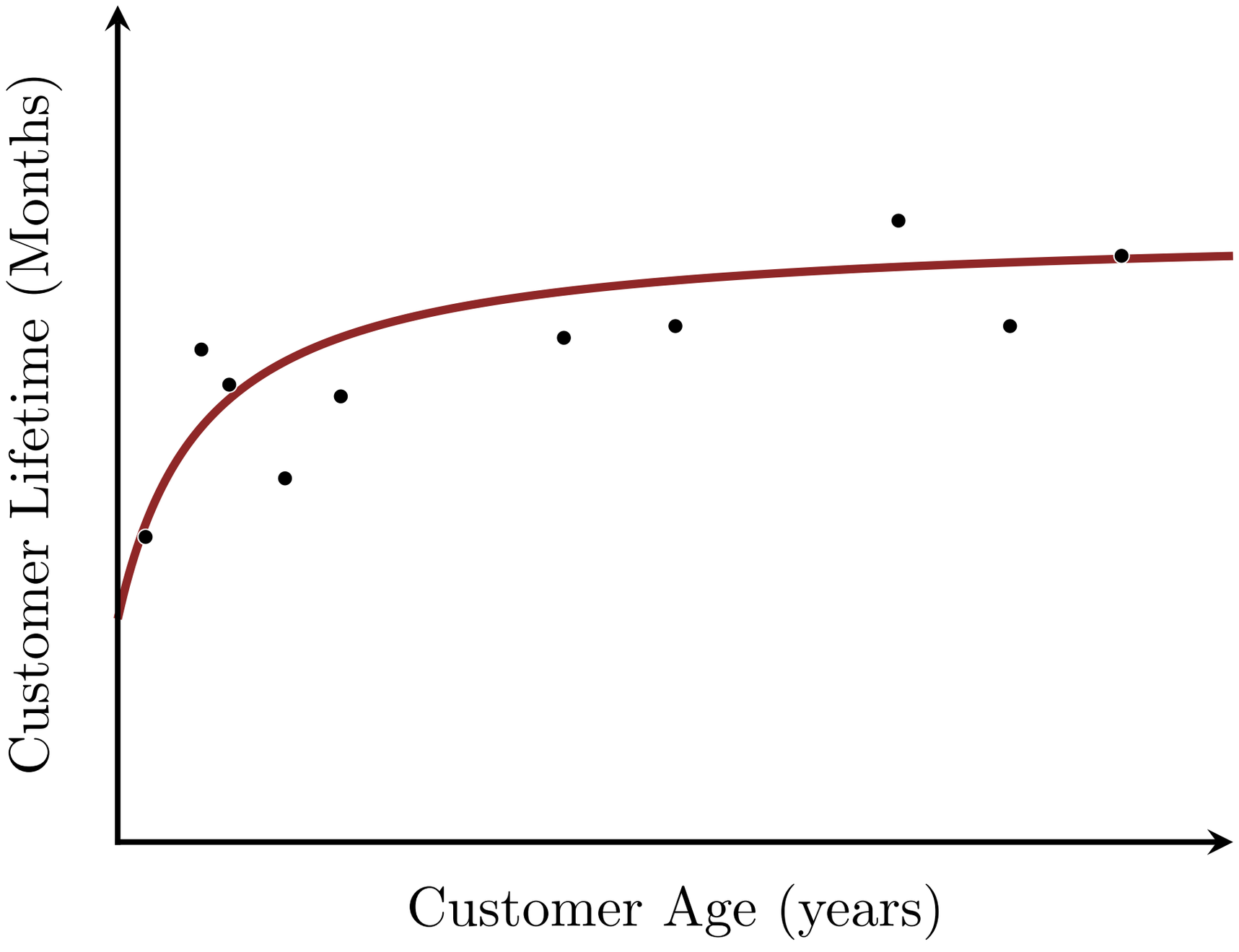
If we know the true relationship between the two, then simulating customer lifetimes is straightforward.



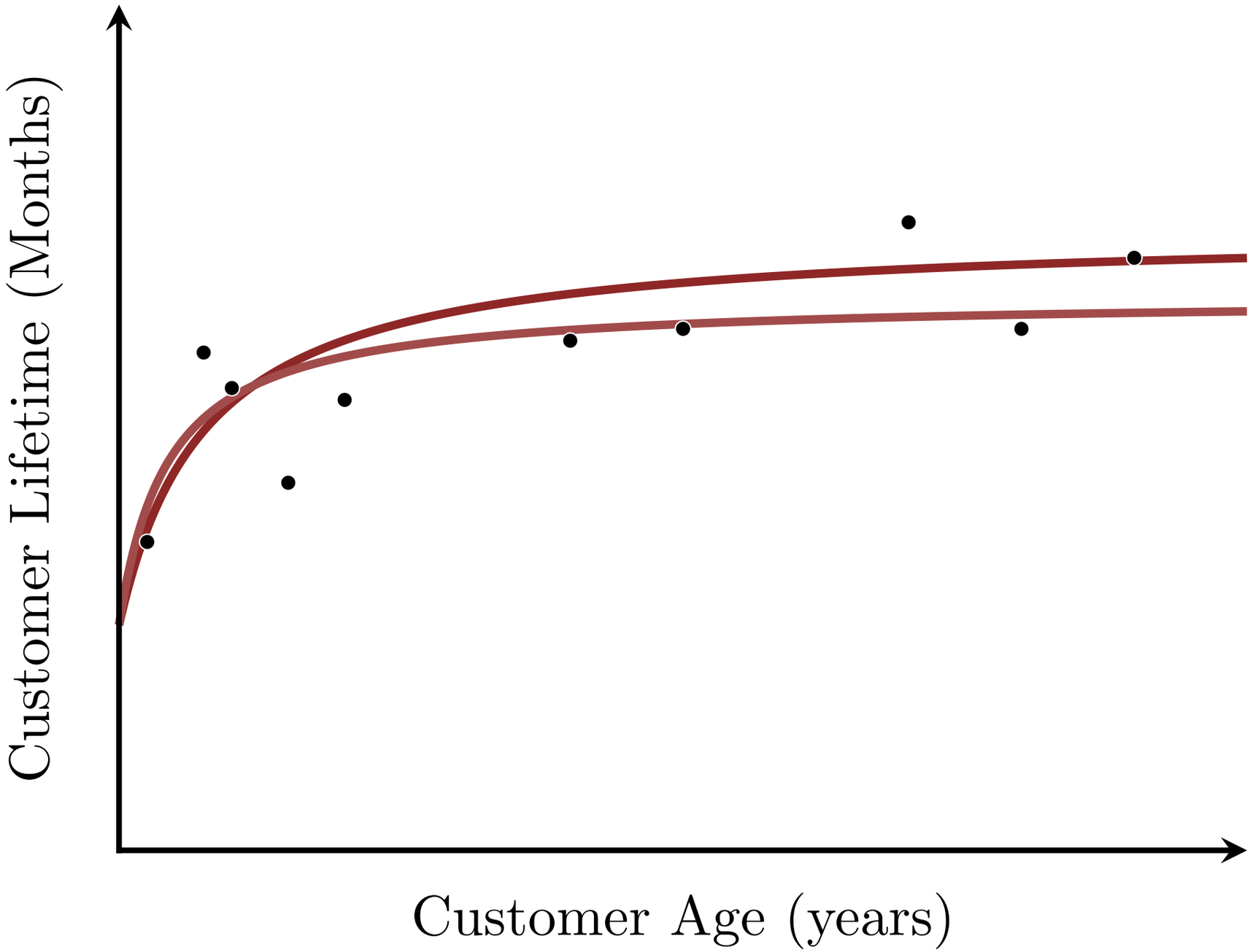
What happens when we don't know the true behavior of the system but have access to only some observations?



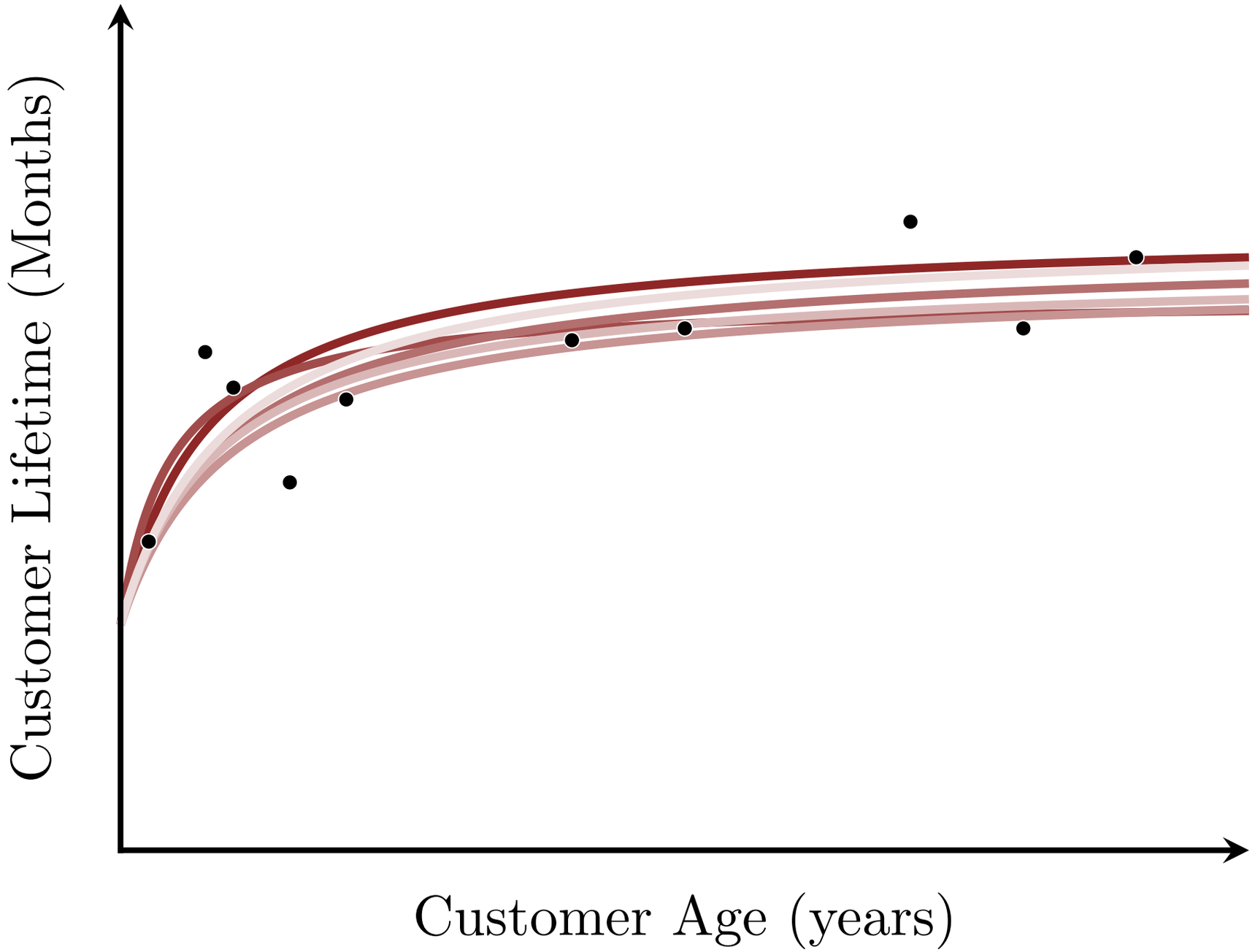
We can certainly find *some* behavior that is consistent with the observations.



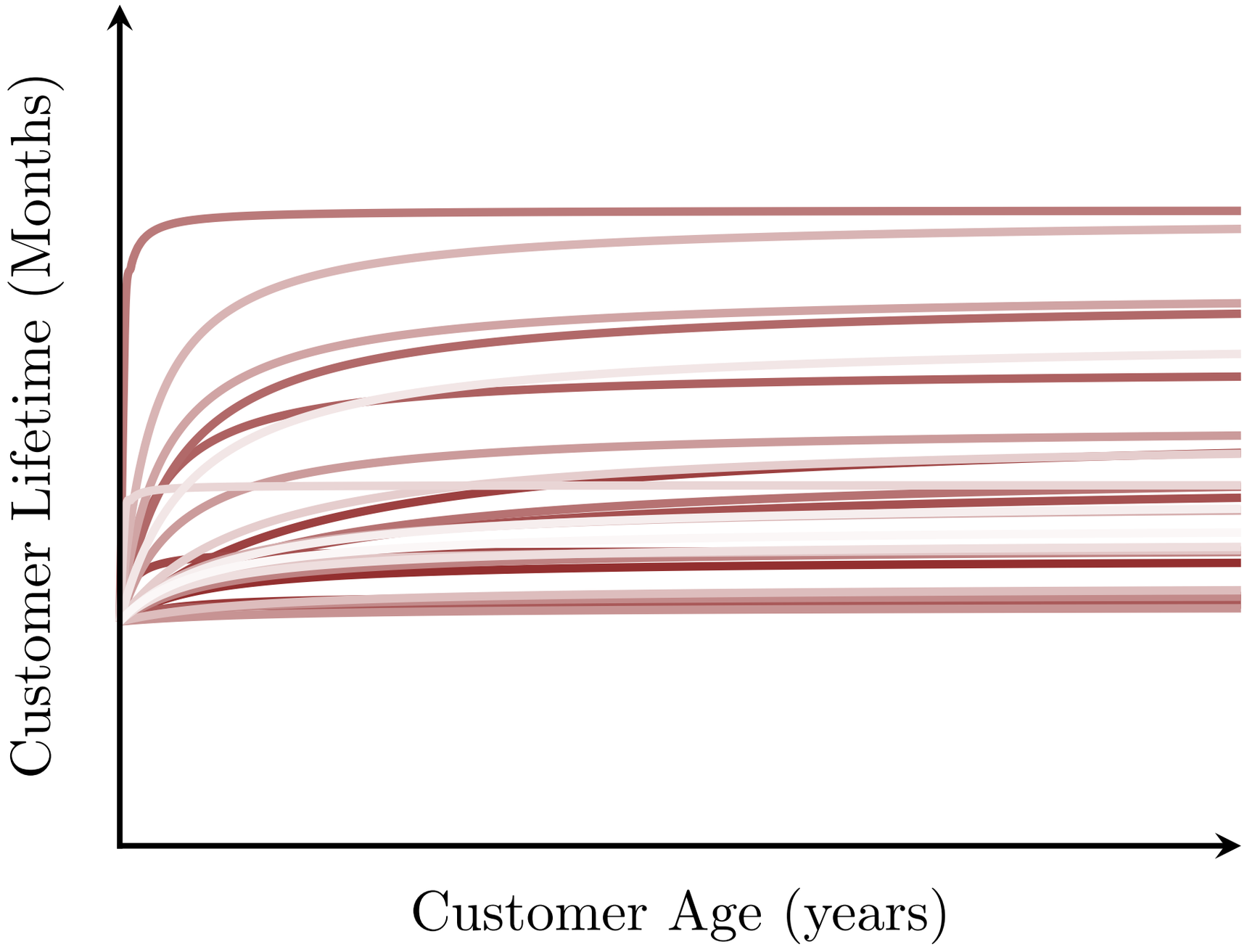
In fact we can typically find *many* behaviors that are consistent with the observations.



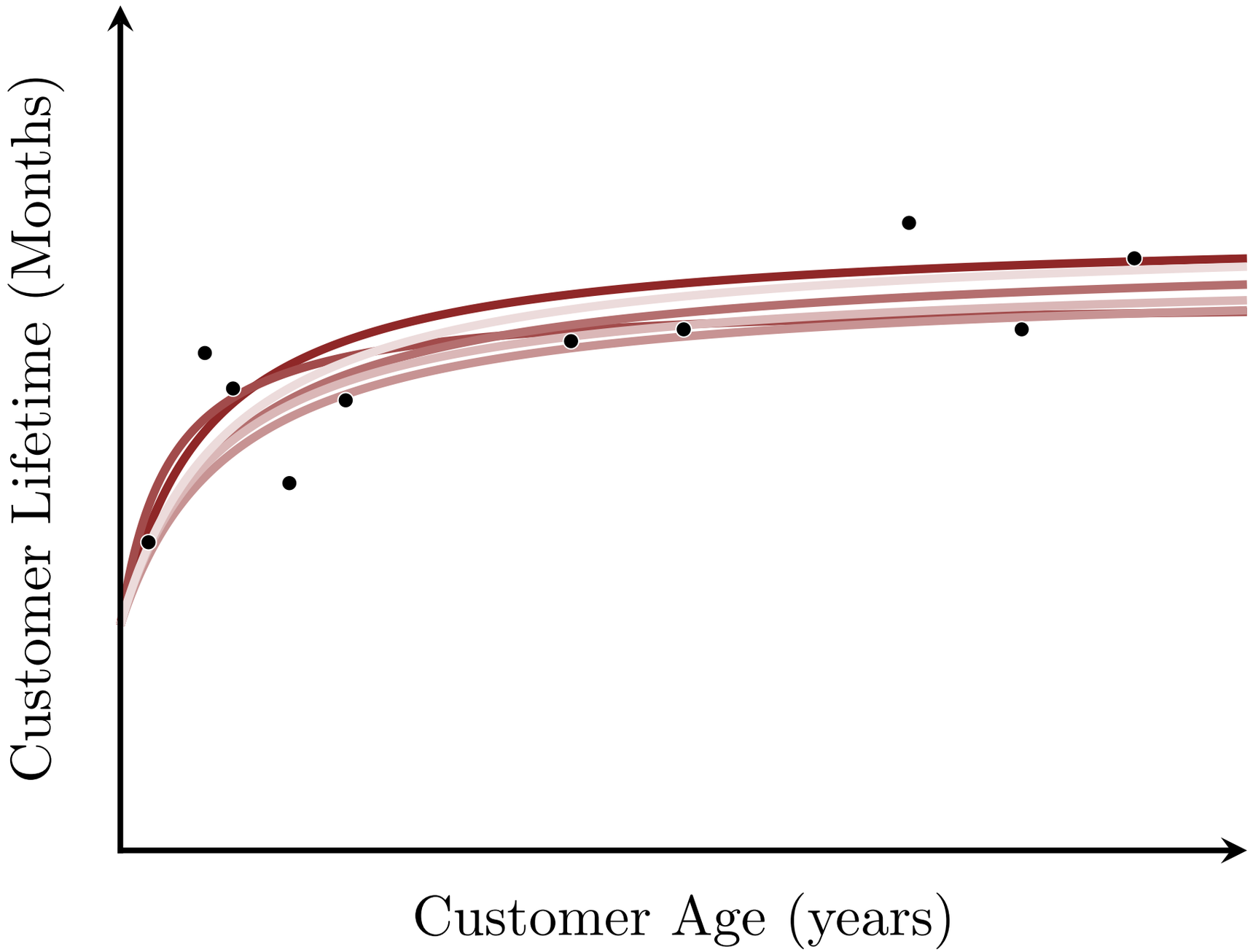
In fact we can typically find *many* behaviors that are consistent with the observations.



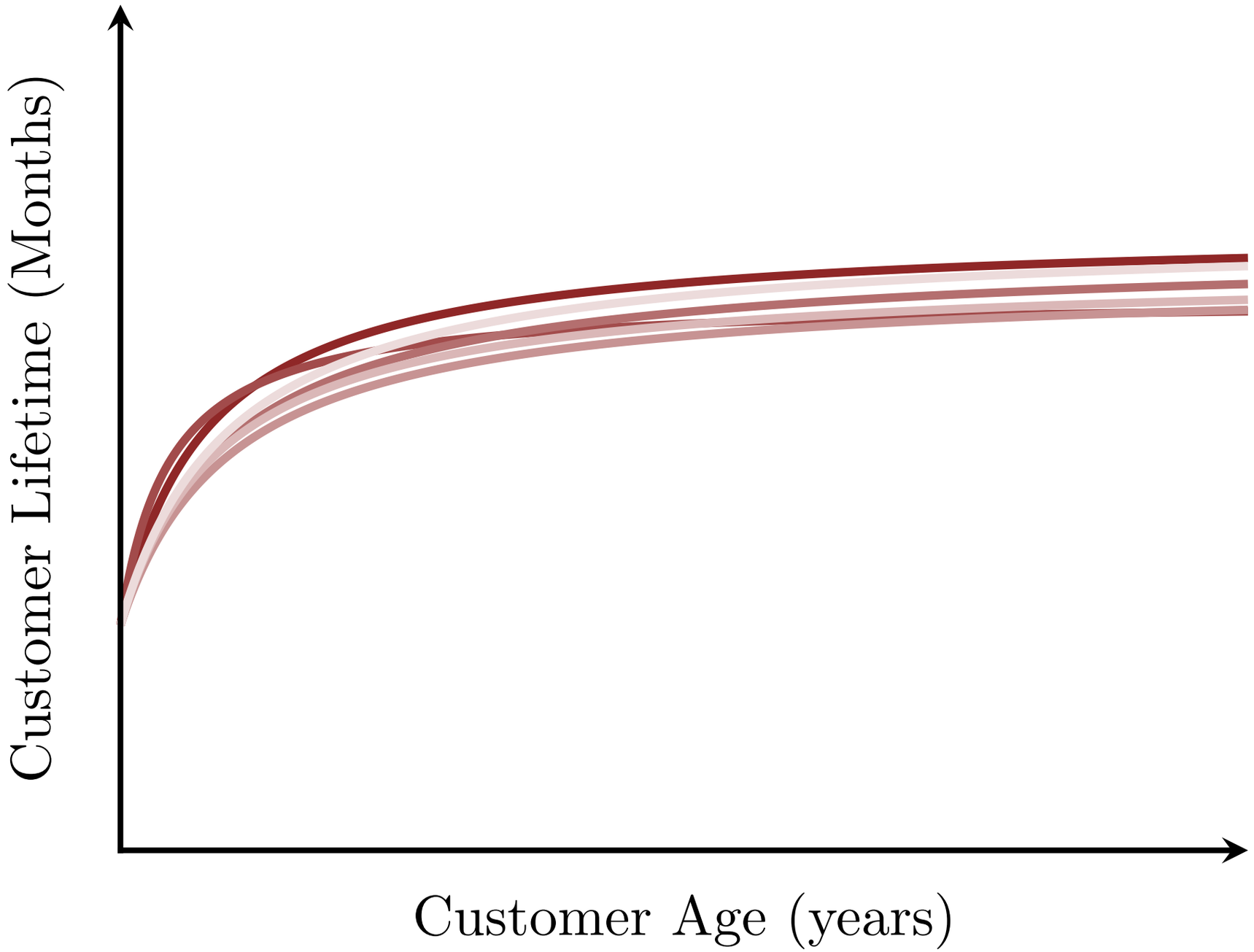
The goal of *probabilistic modeling* is to quantify all of the reasonable behaviors of a system.



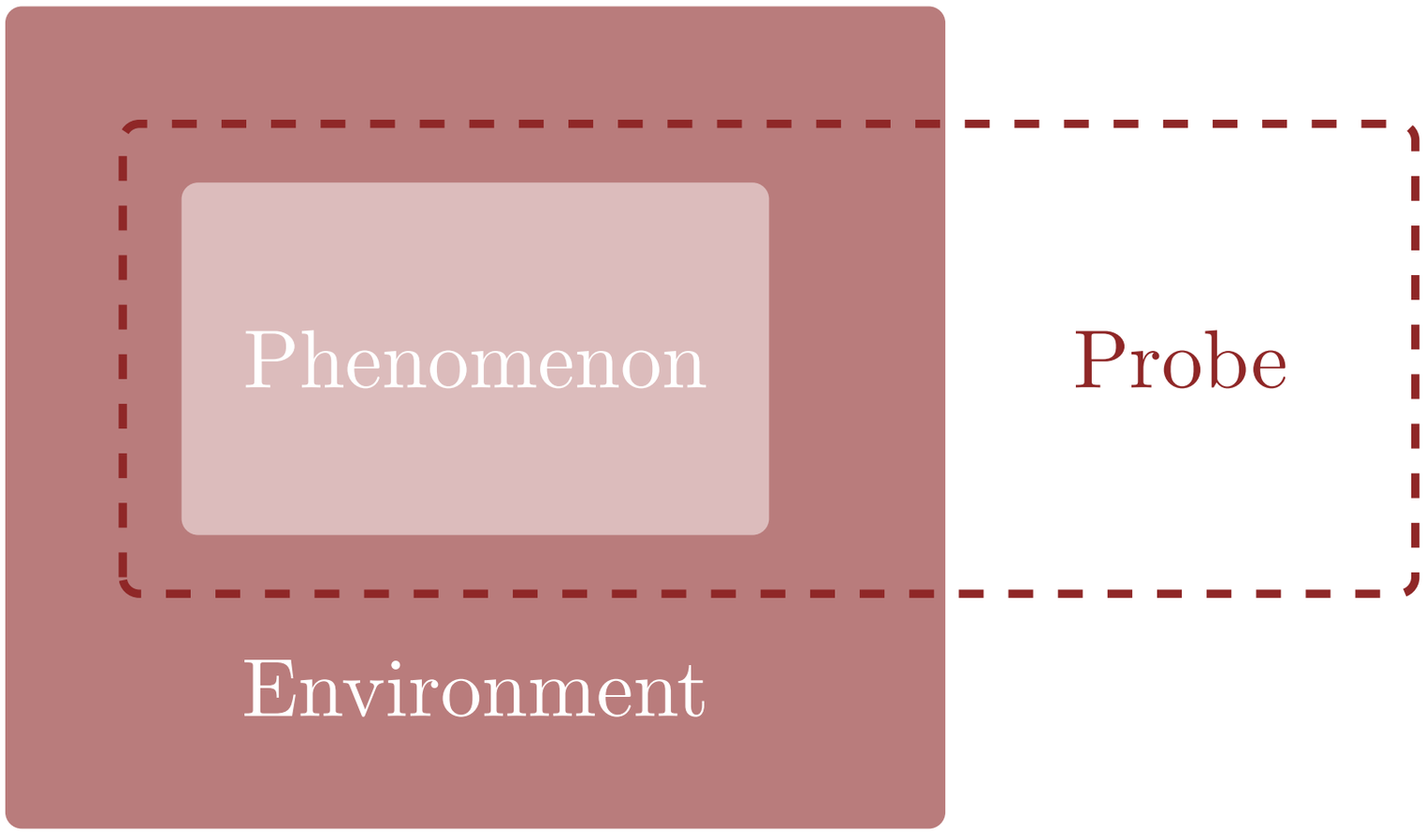
Statistical inference considers how to quantify the consistency of those behaviors with observed data.



Once we have inferred which behaviors are consistent, we can make *decisions* about how to interact with the system.

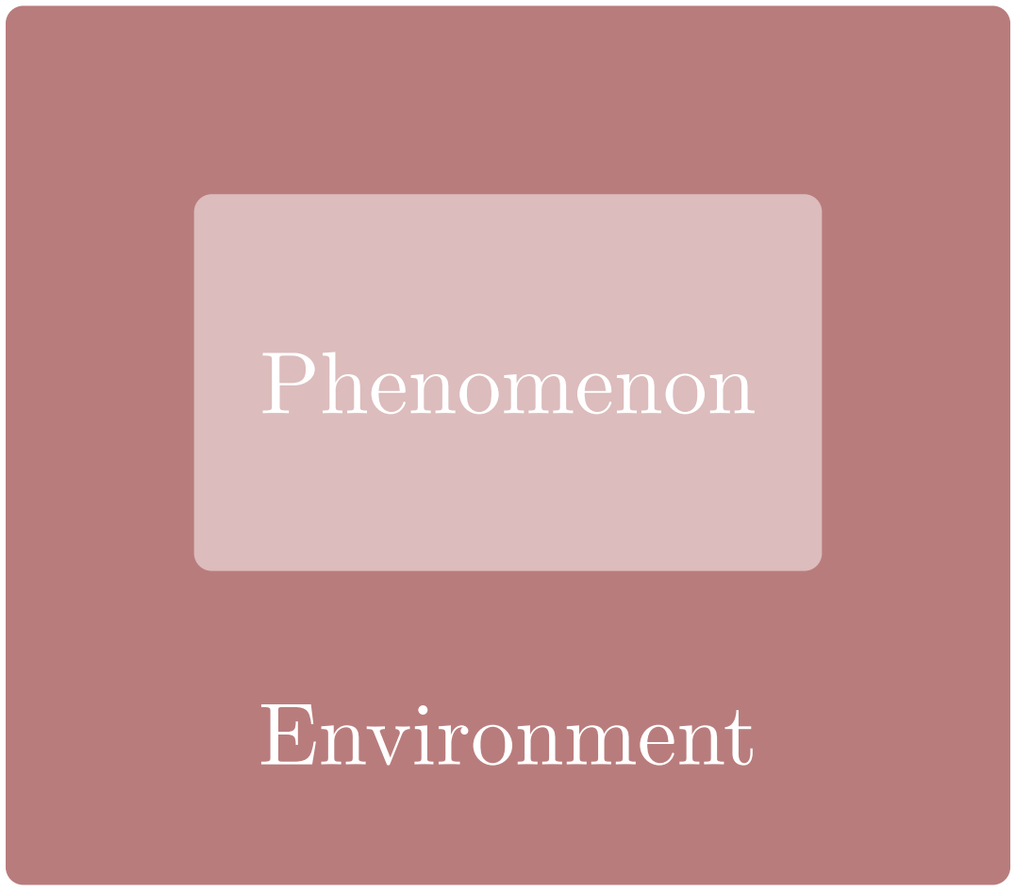


Data Generating Processes

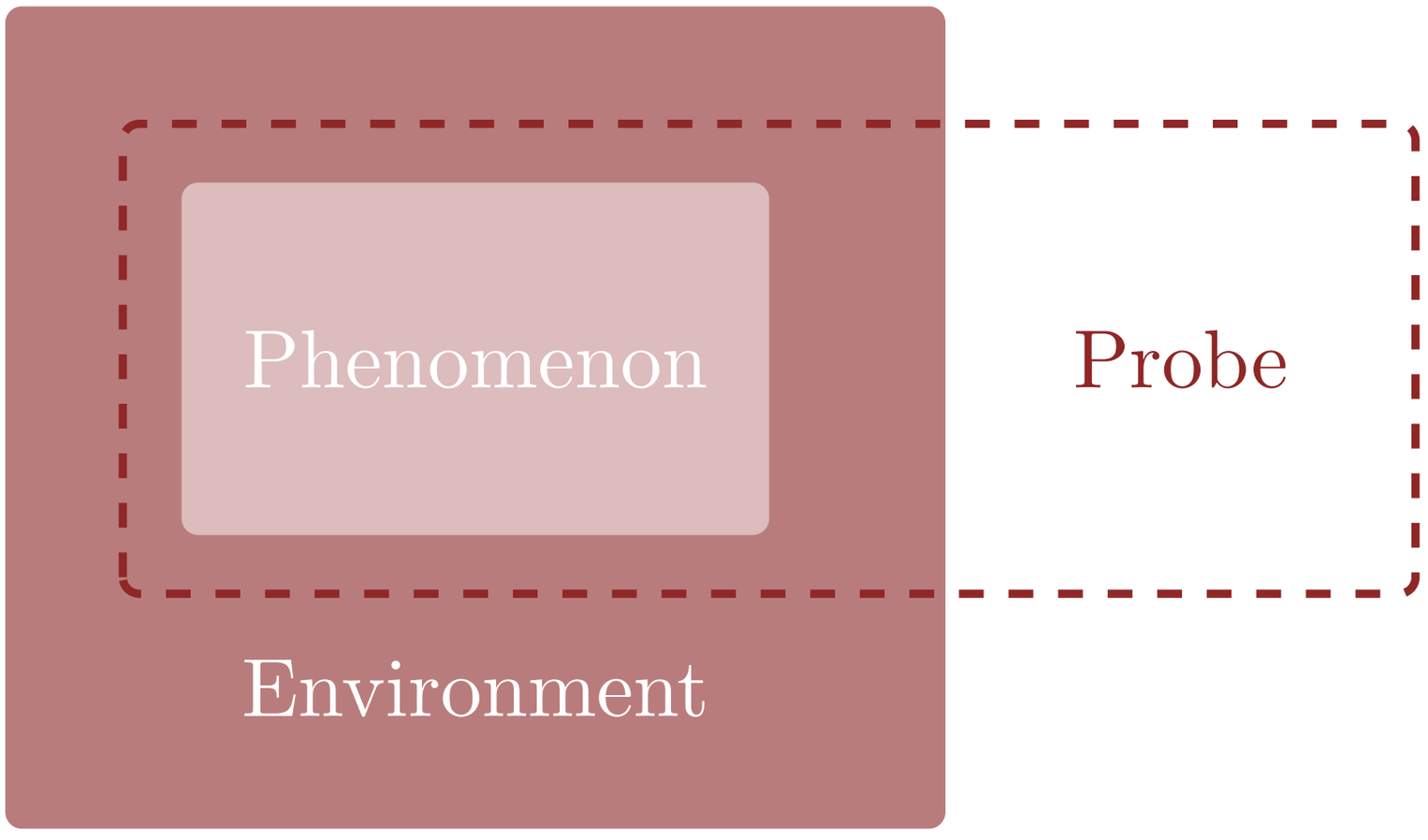


In most applications, we are making decisions about how to interact with some *latent phenomenon*.

We can't interact with a latent phenomenon directly, only through its manifestation in some *environment*.

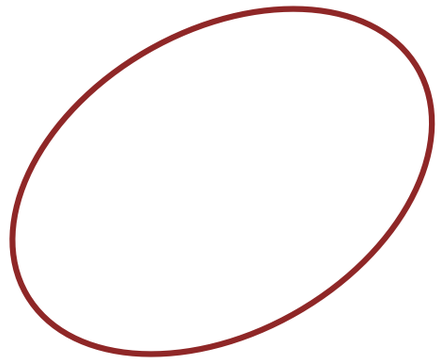


An experimental or observational *probe* explores the environment and, hopefully, the phenomenon of interest.

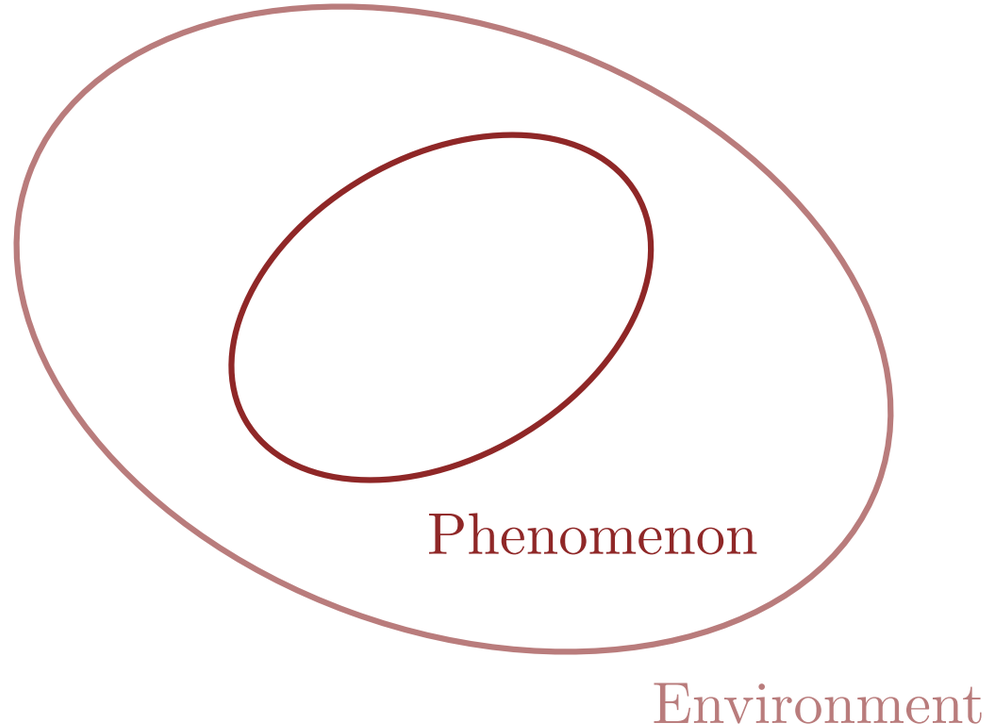


The interaction of the probe with the environment and our phenomenon establishes a *data generating process*.

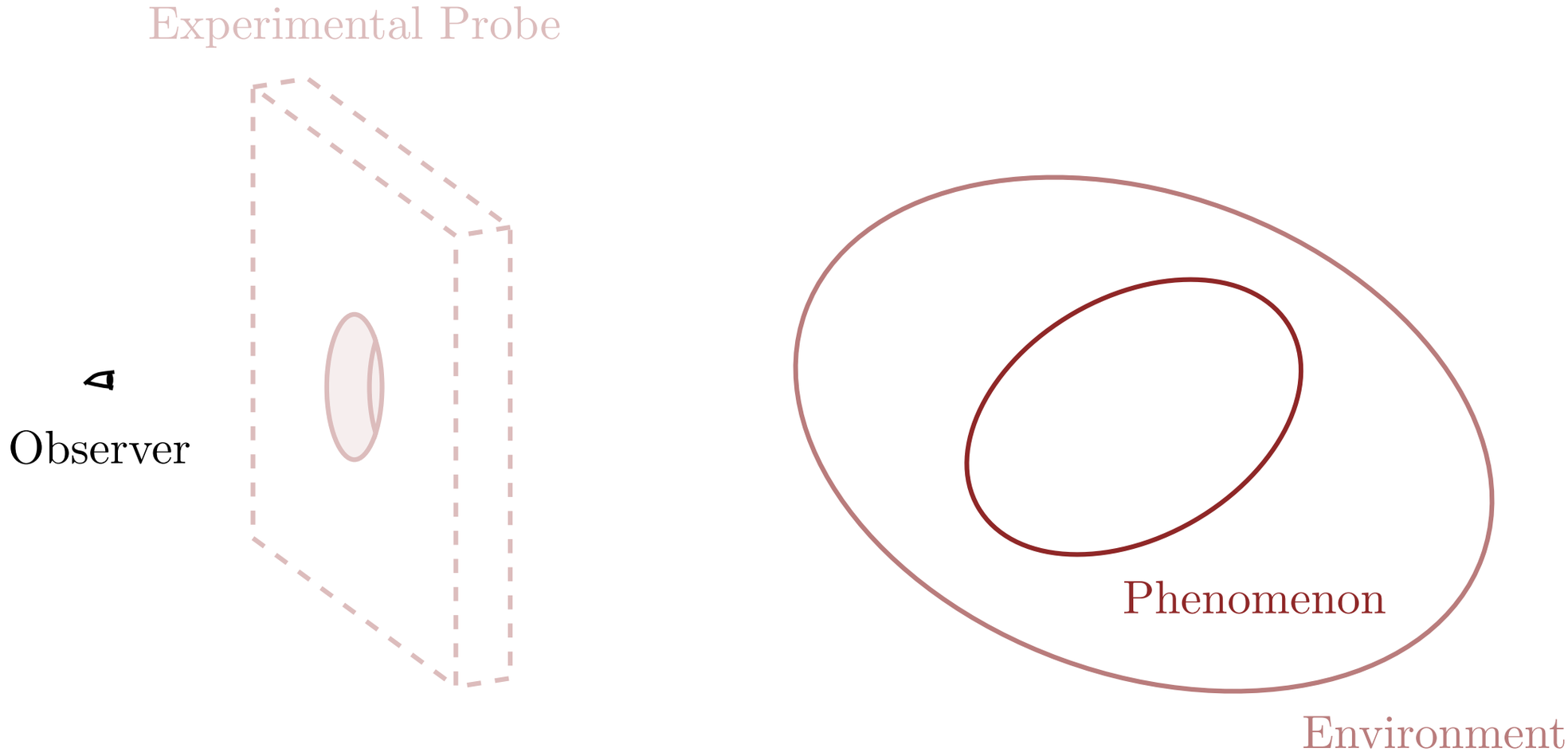
The interaction of the probe with the environment and our phenomenon establishes a *data generating process*.



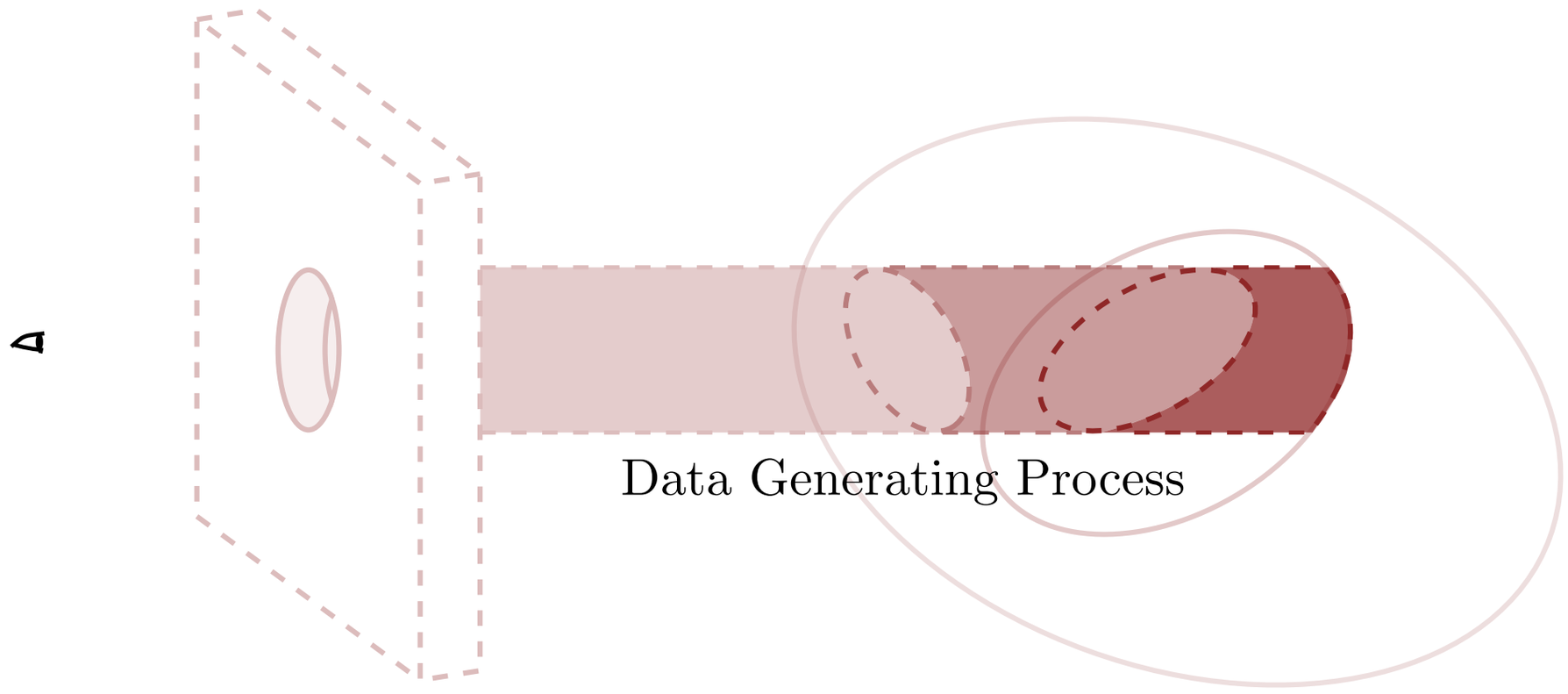
The interaction of the probe with the environment and our phenomenon establishes a *data generating process*.



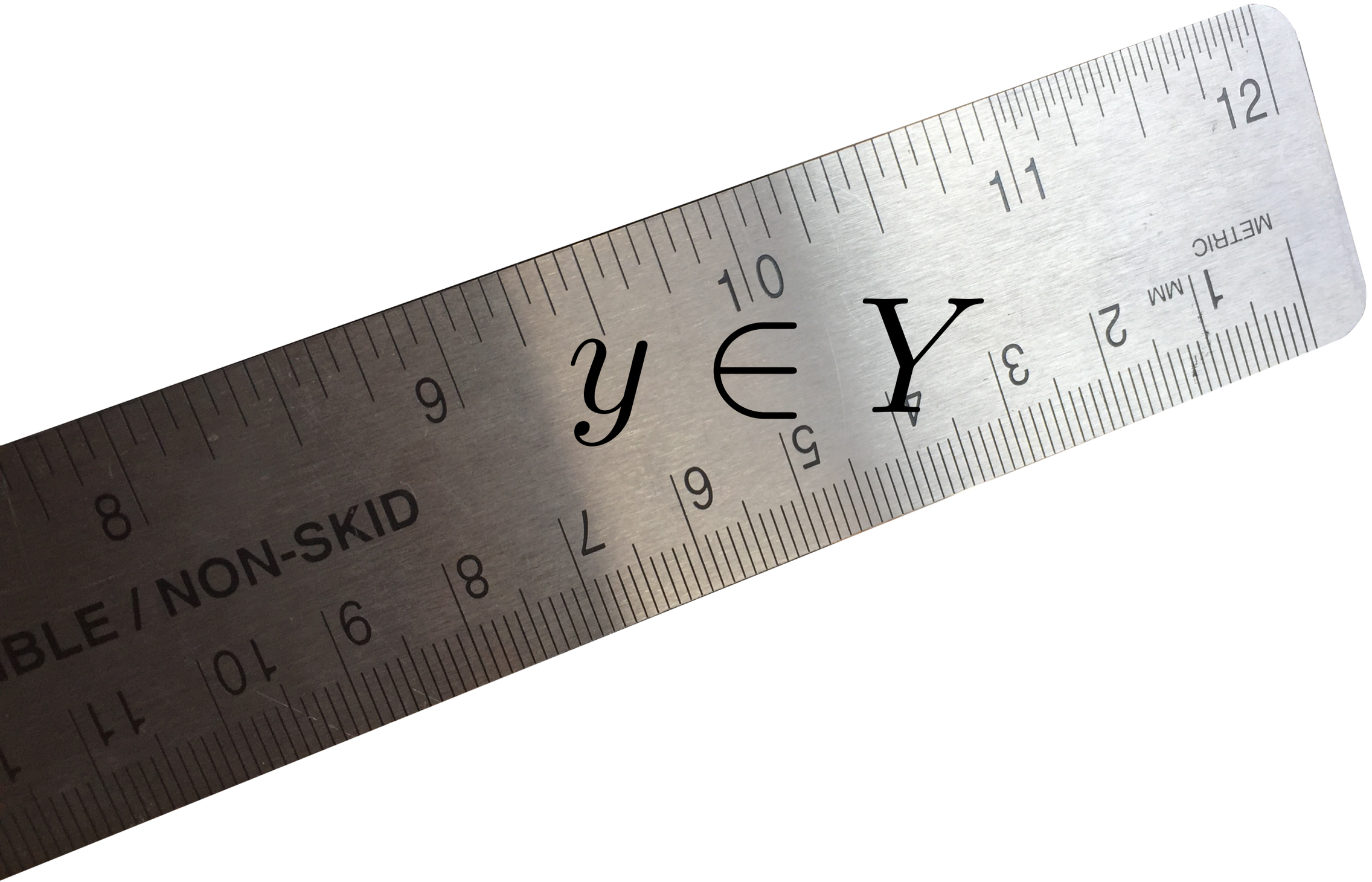
The interaction of the probe with the environment and our phenomenon establishes a *data generating process*.



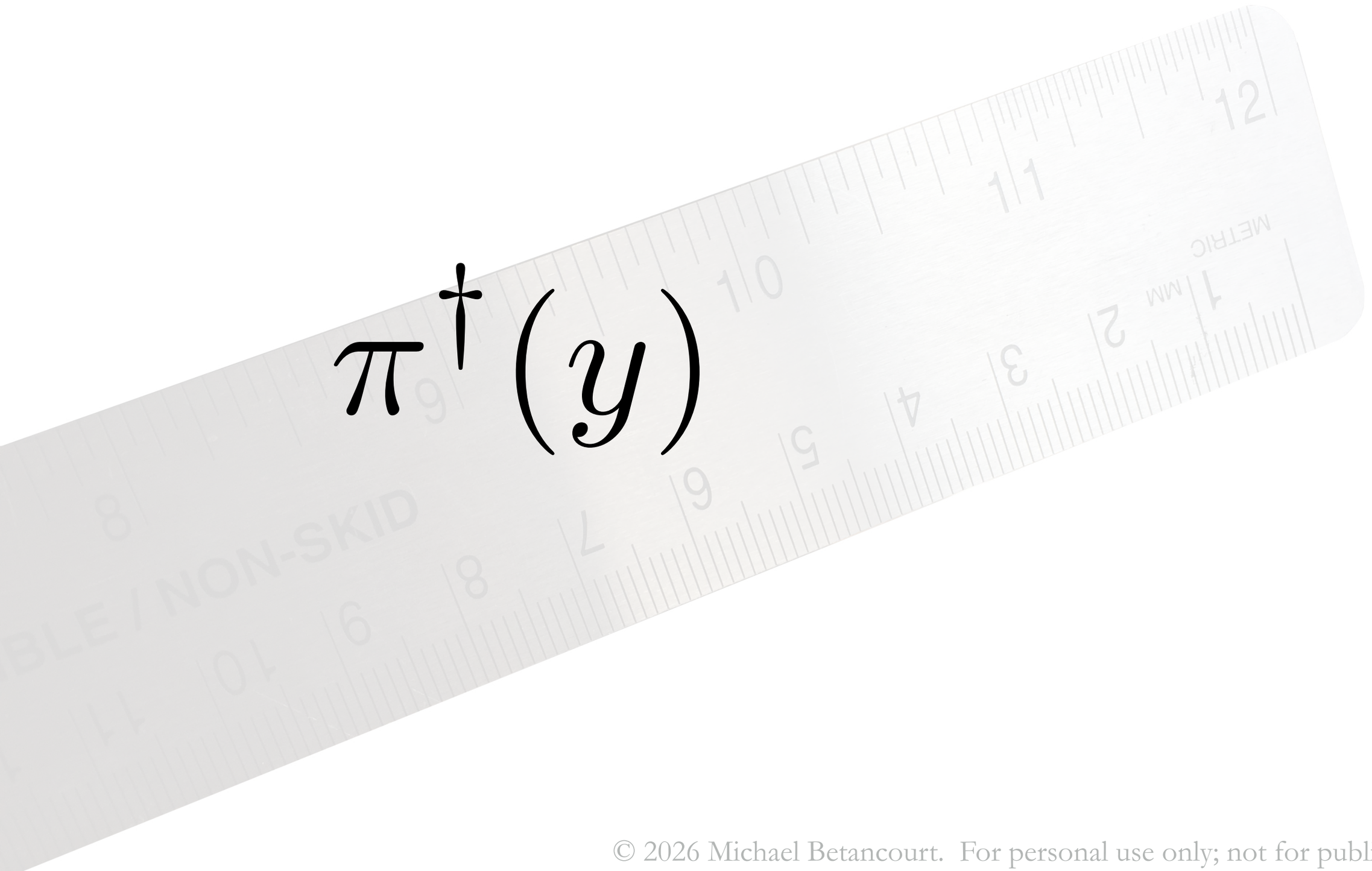
The interaction of the probe with the environment and our phenomenon establishes a *data generating process*.



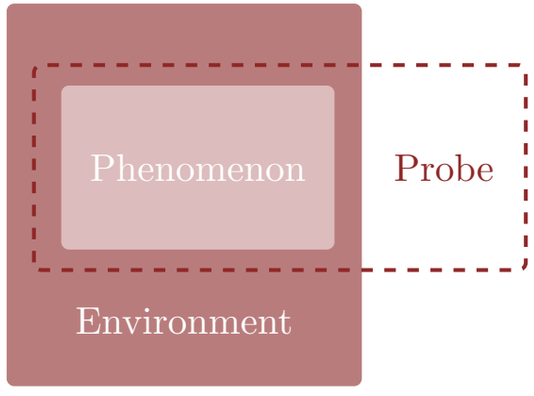
One of the defining features of a data generating process is that the observations it generates are *unpredictable*.



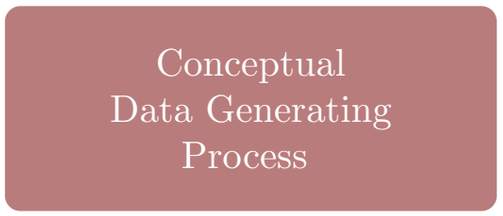
To mathematically model these processes, we assume that the unpredictability can be quantified *probabilistically*.



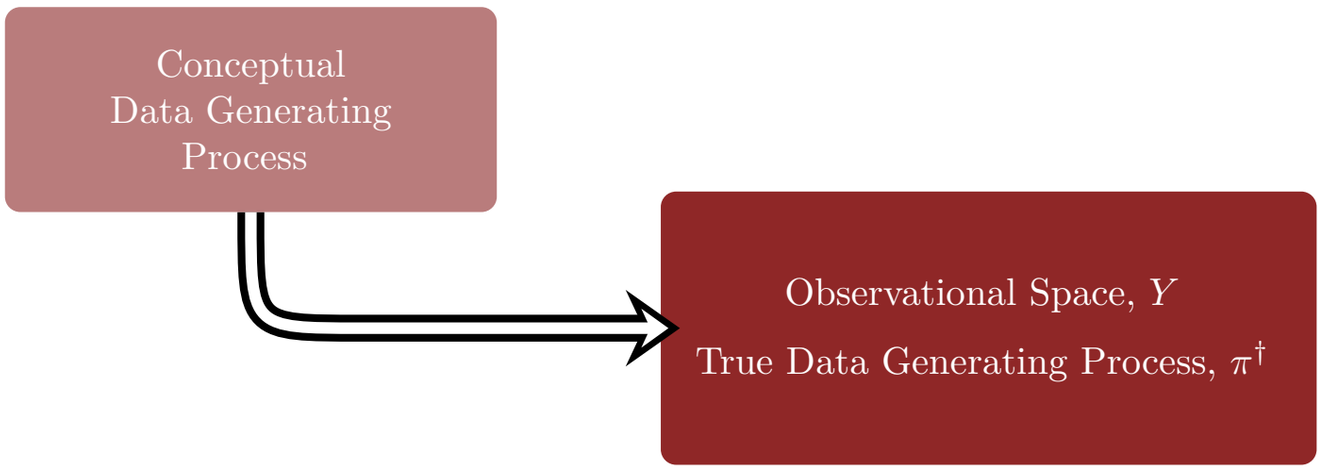
First we reason about the phenomenon, the environment in which it manifests, and an observational probe.



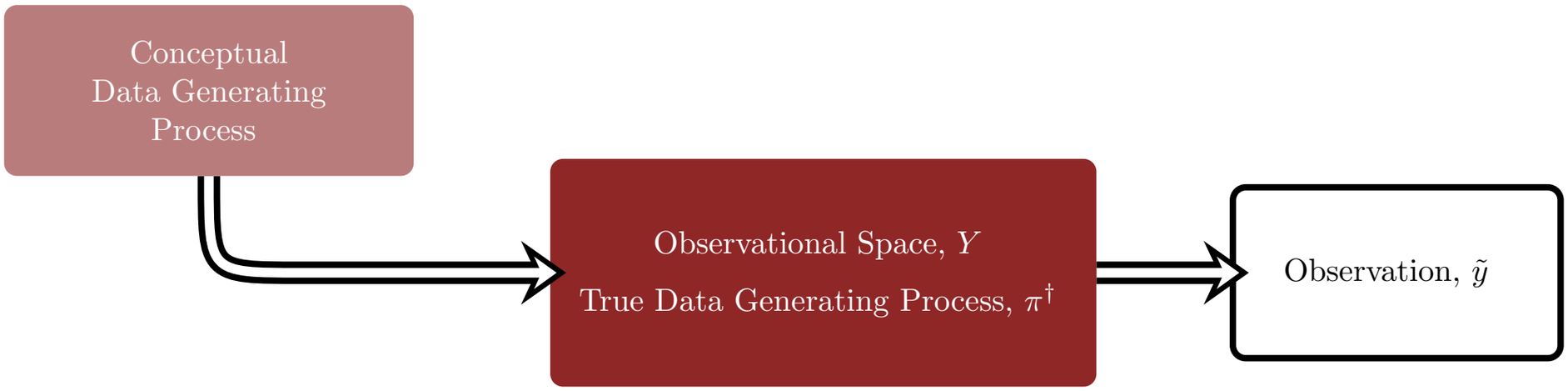
Our understanding of these components defines a *conceptual* data generating process.



We can then mathematically formalize this as an *observational space* and a *true data generating process*.



An explicit realization, or *observation*, is modeled as a sample from the true data generating process.



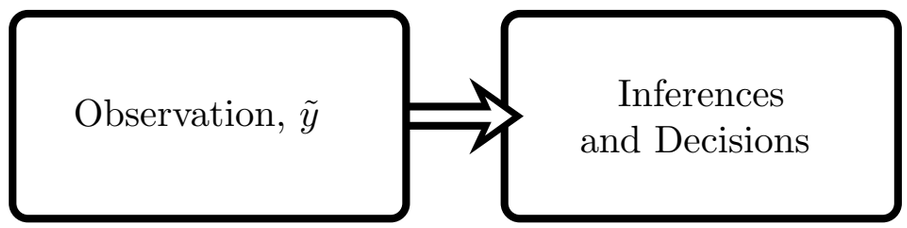
Ultimately we want to use those observations
to make inferences and inform decisions.

Conceptual
Data Generating
Process

Observation, $\tilde{y}$

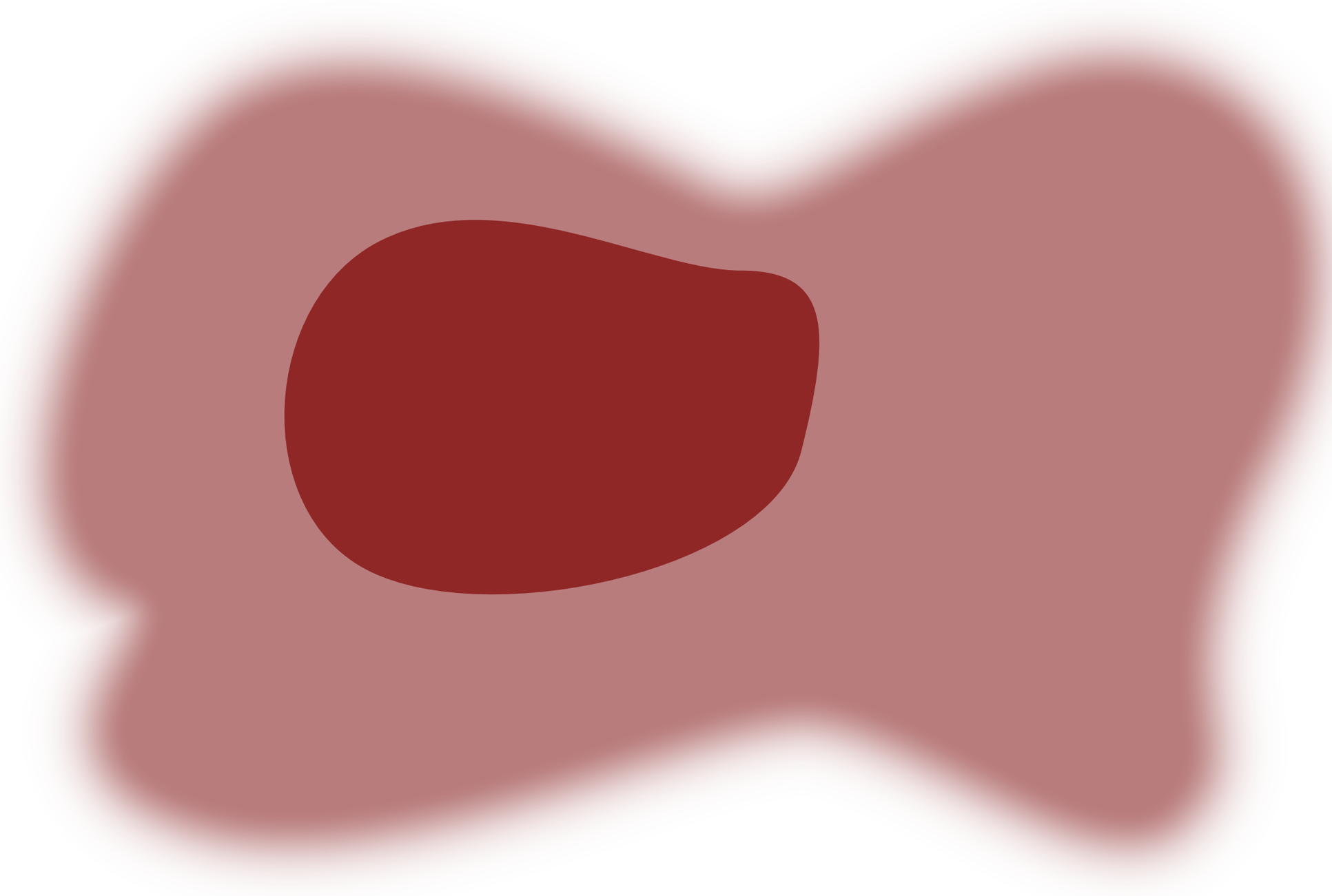
Ultimately we want to use those observations
to make inferences and inform decisions.

Conceptual
Data Generating
Process

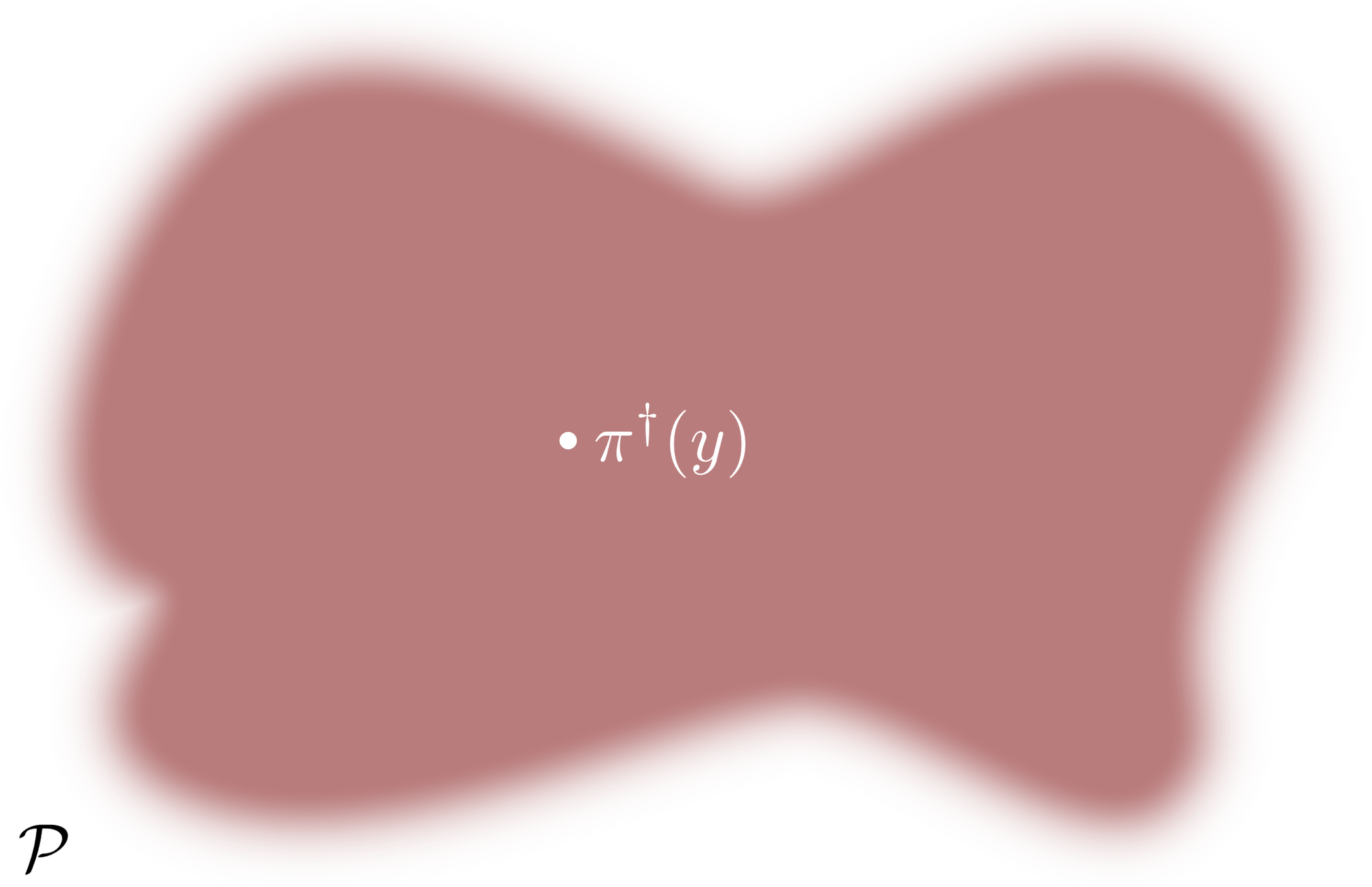


Especially inferences and decisions about how to interact with the latent phenomena of interest.

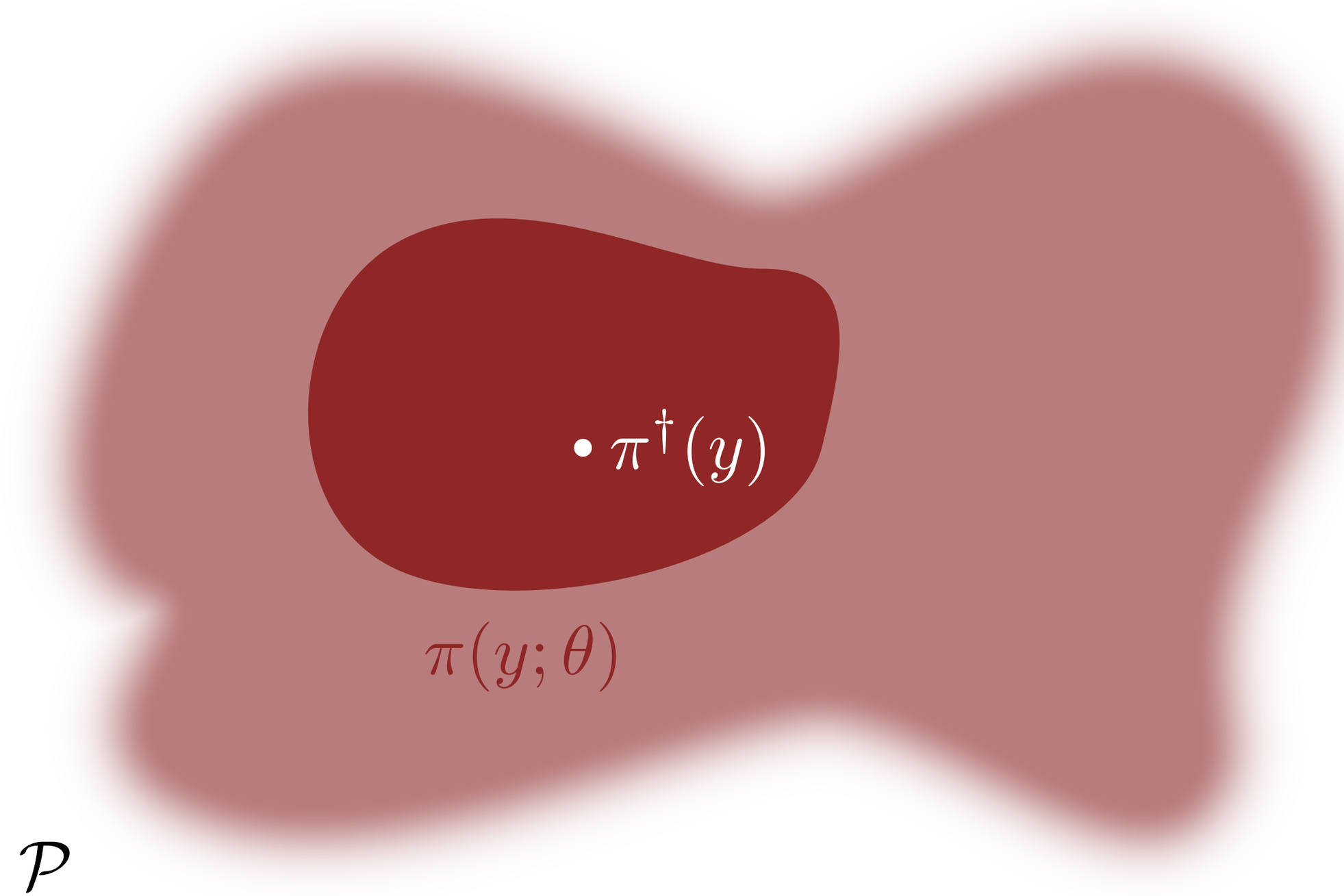
Probabilistic Modeling



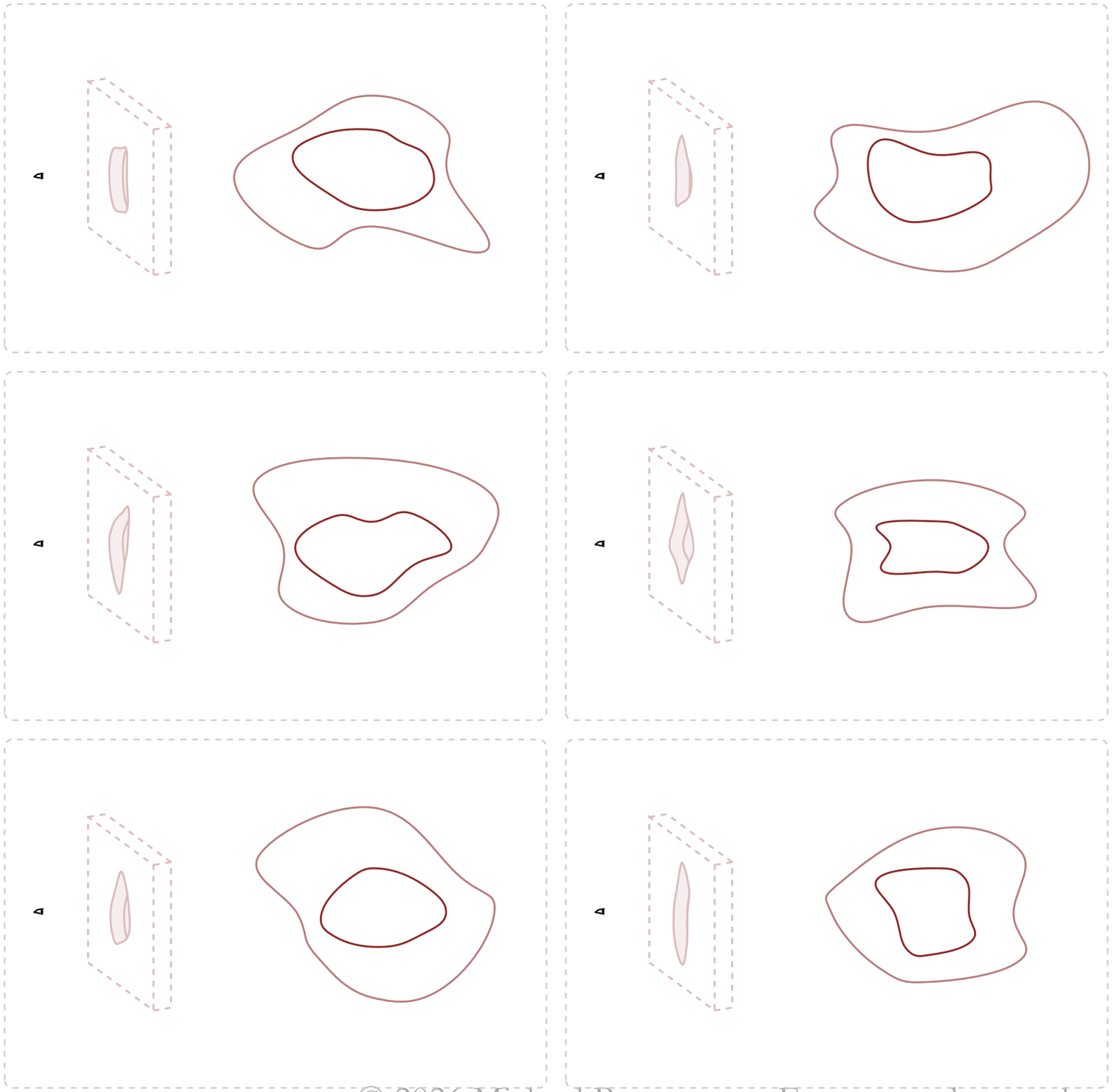
In practice we don't know what the true data generating process is. We also can't search through all possibilities.



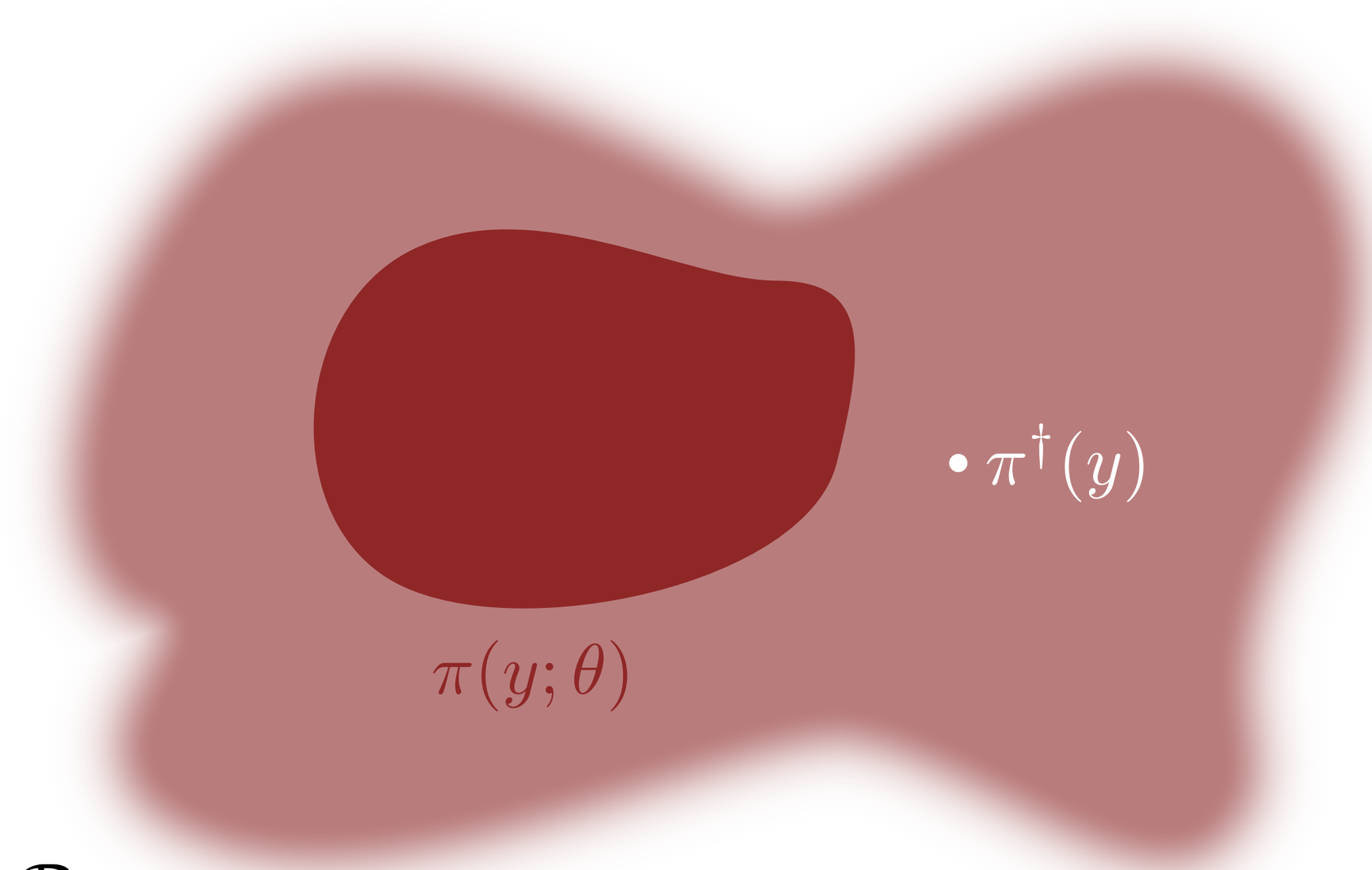
Instead we have to limit our consideration to an *observational model* of possible data generating processes.



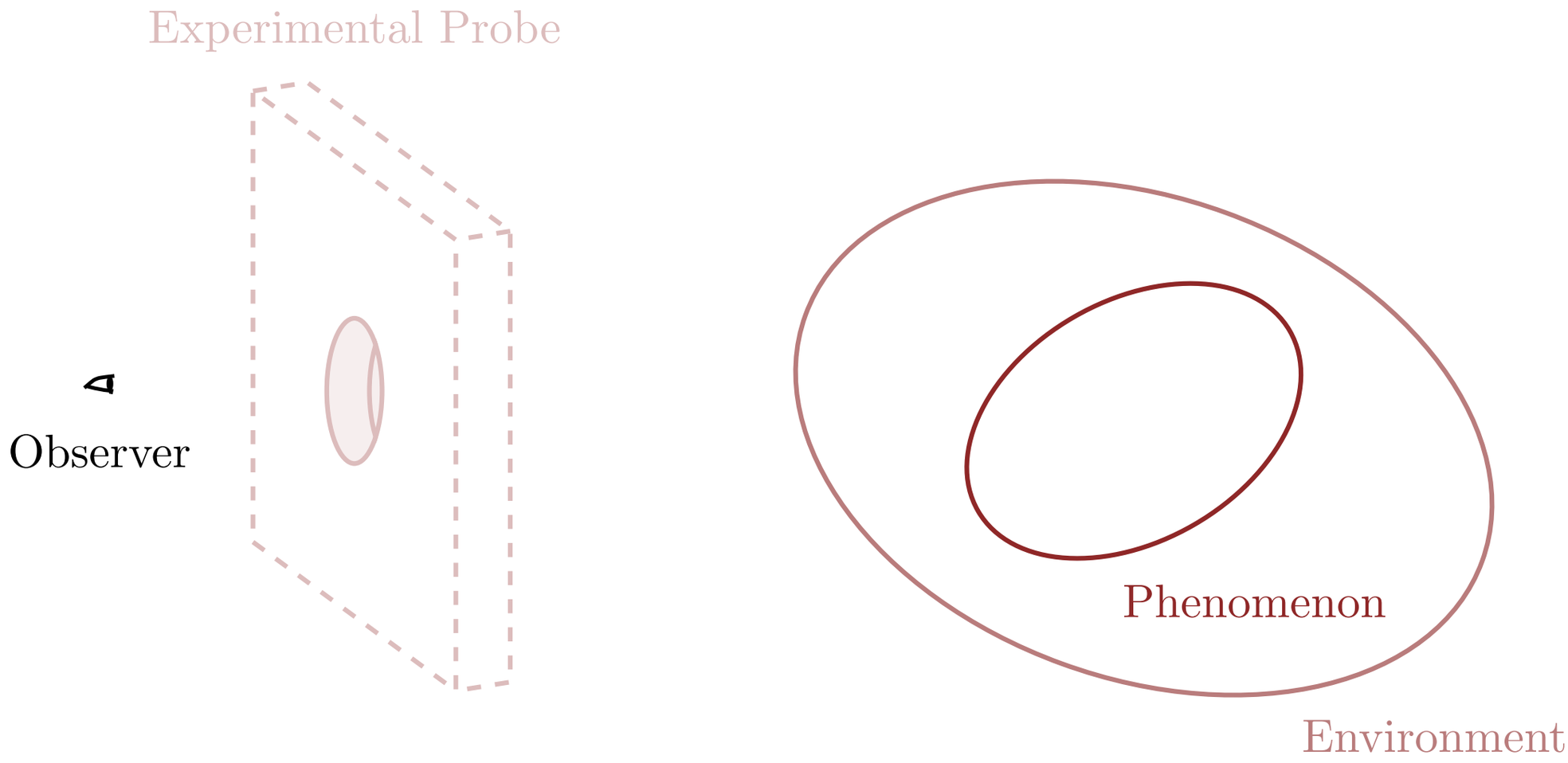
Conceptually, an observational model is a collection of mathematical stories for how data could be generated.



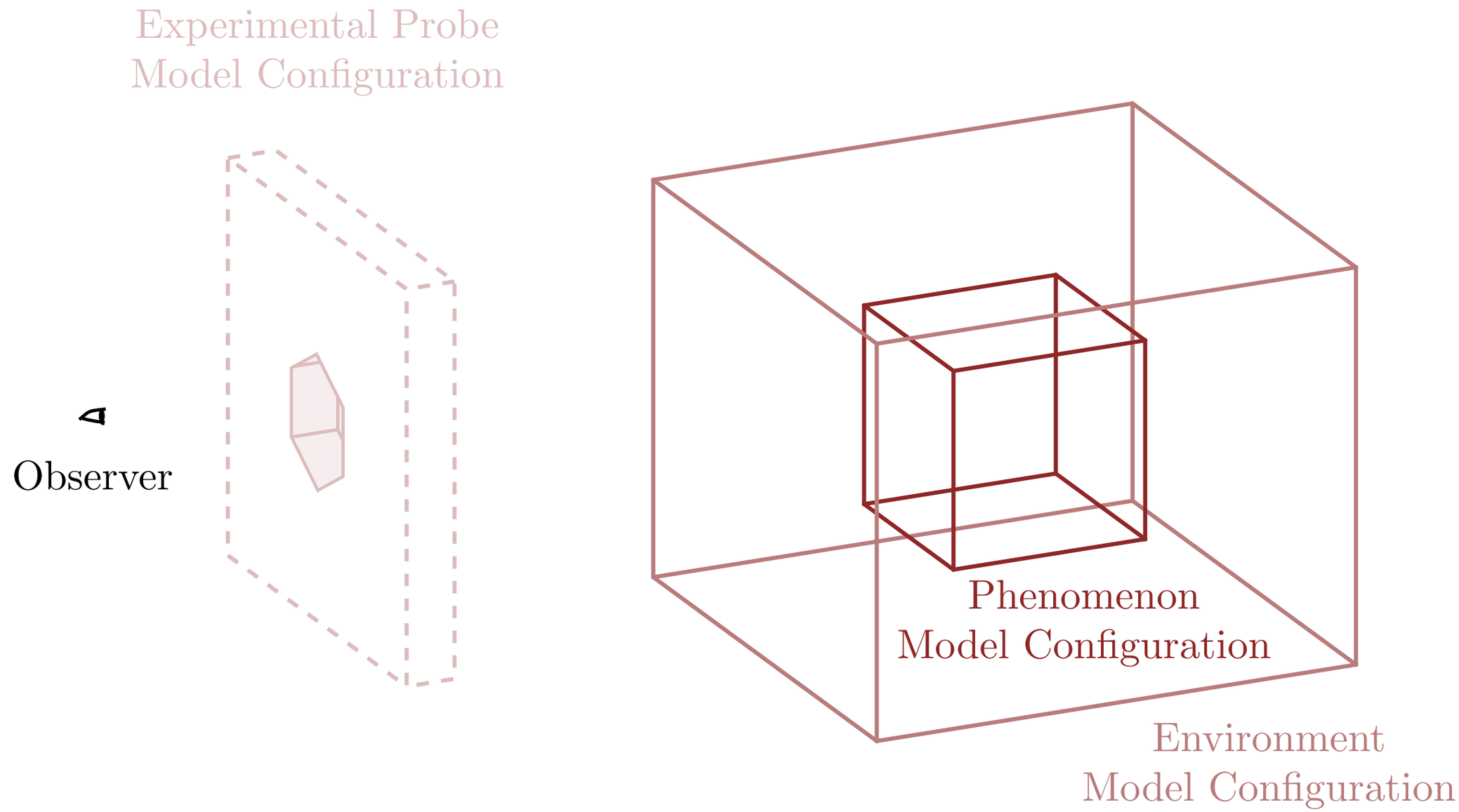
Practical observational models, however, are unlikely to capture every detail of the true data generating process.



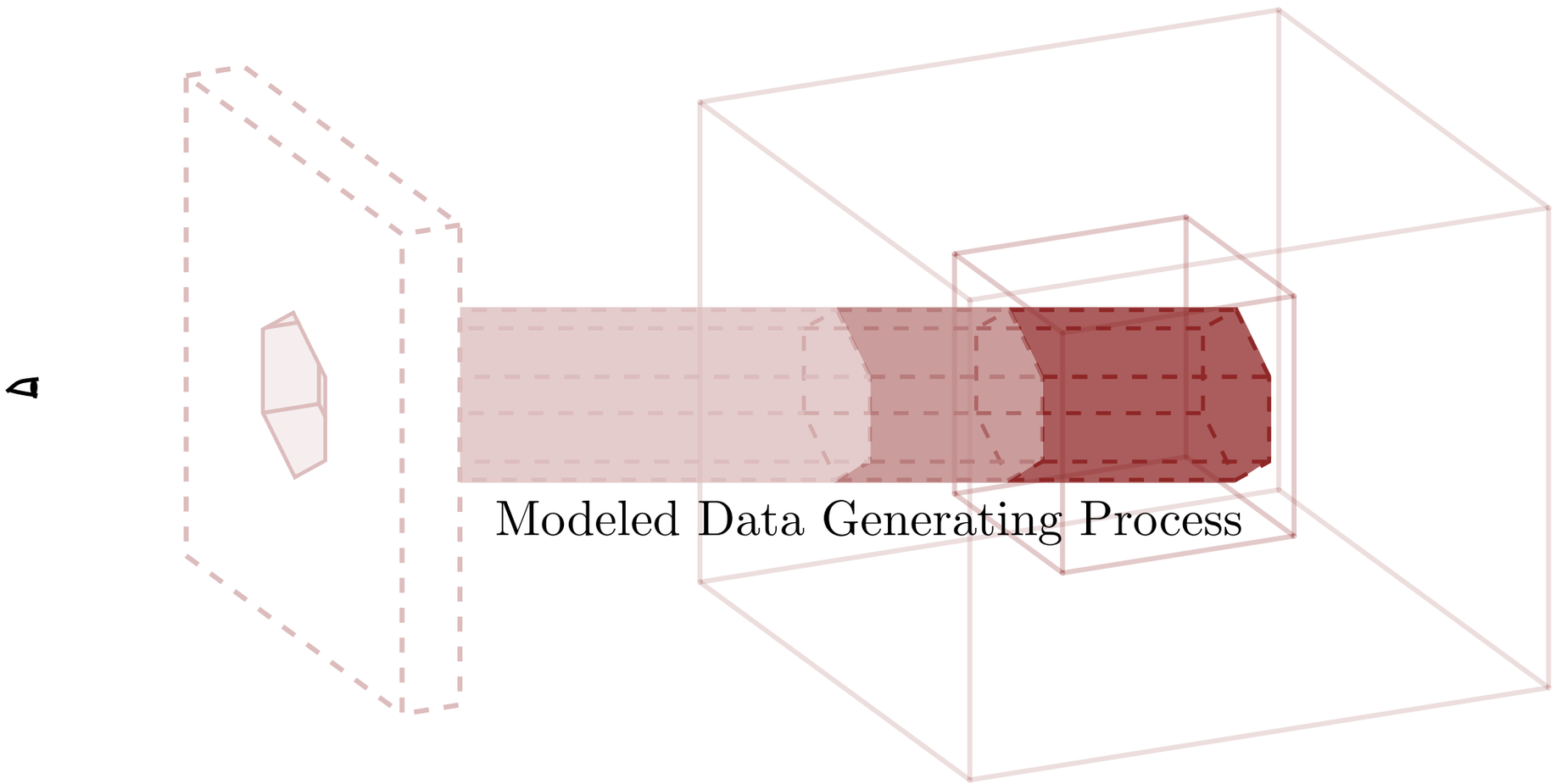
More realistically, any element of an observational model will at best *approximate* the structure of the true process.



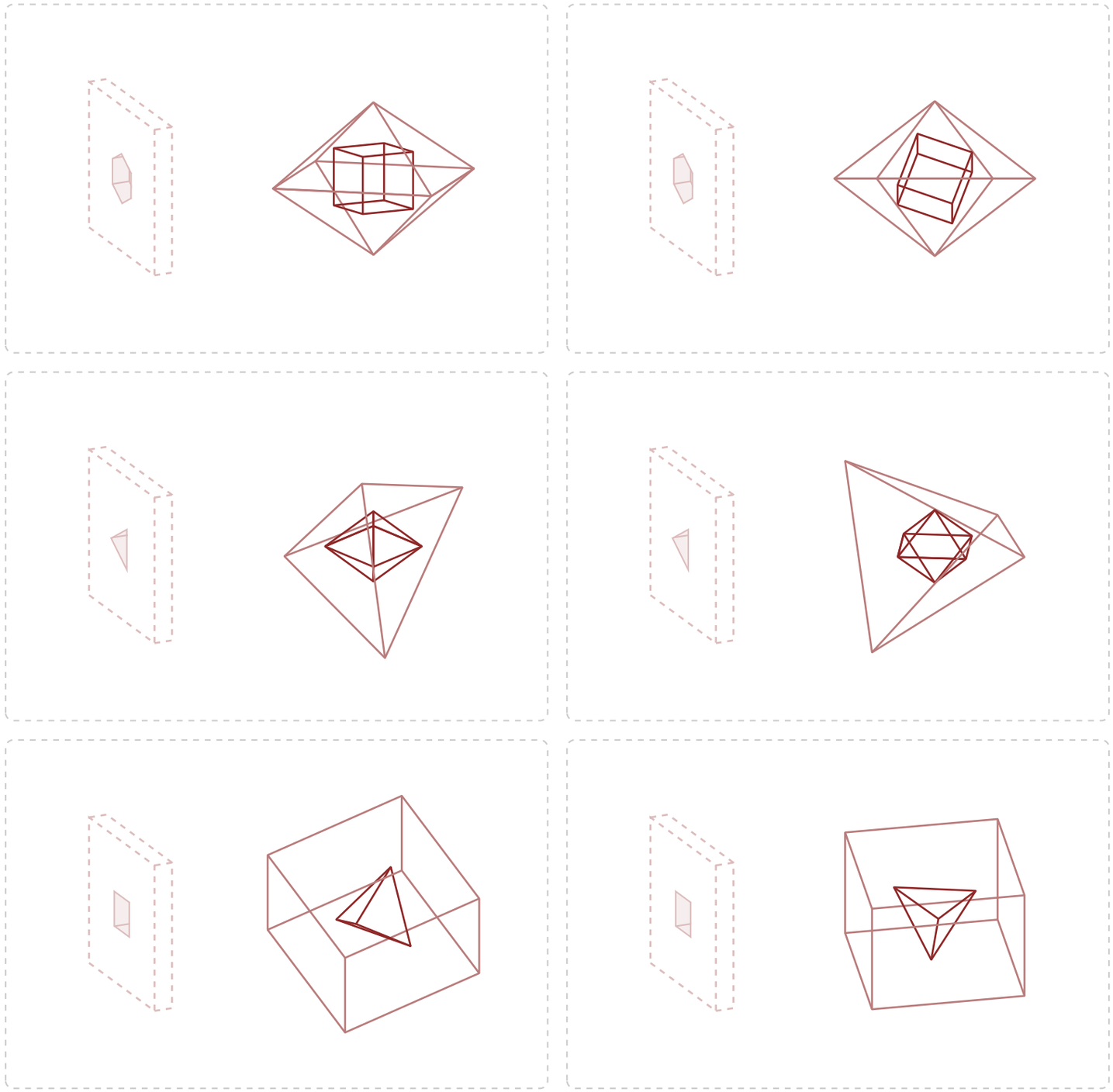
More realistically, any element of an observational model will at best *approximate* the structure of the true process.



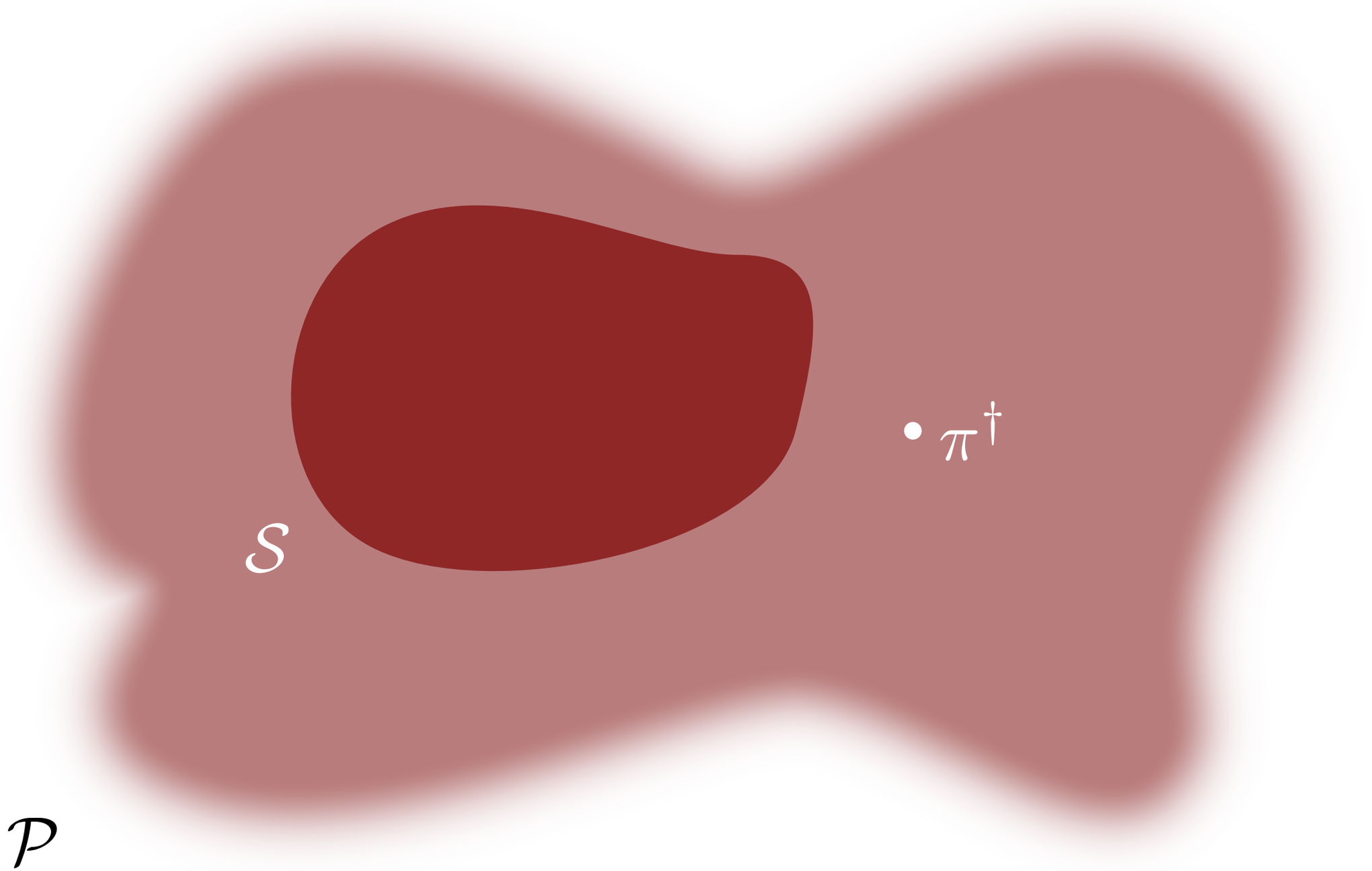
More realistically, any element of an observational model will at best *approximate* the structure of the true process.



The observation model becomes a collection of candidate approximations *which may or may no be good approximations!*

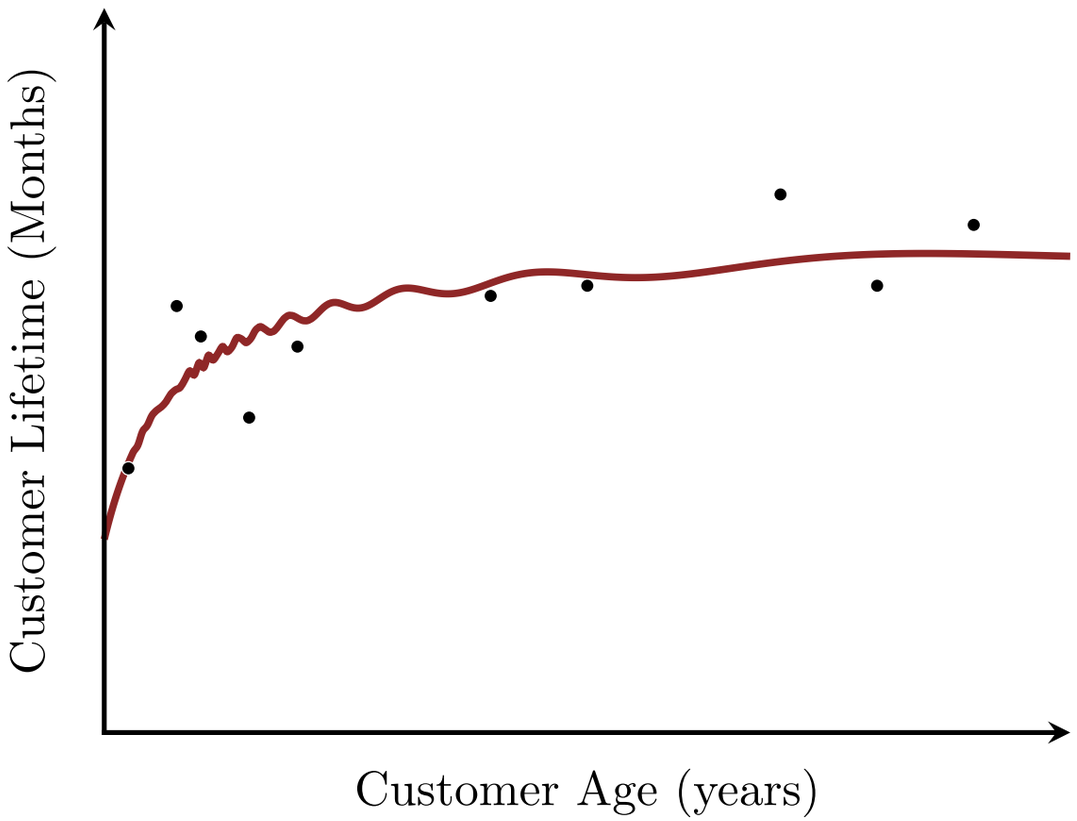


An observational model that can only approximate the true data generating process can still provide insights.



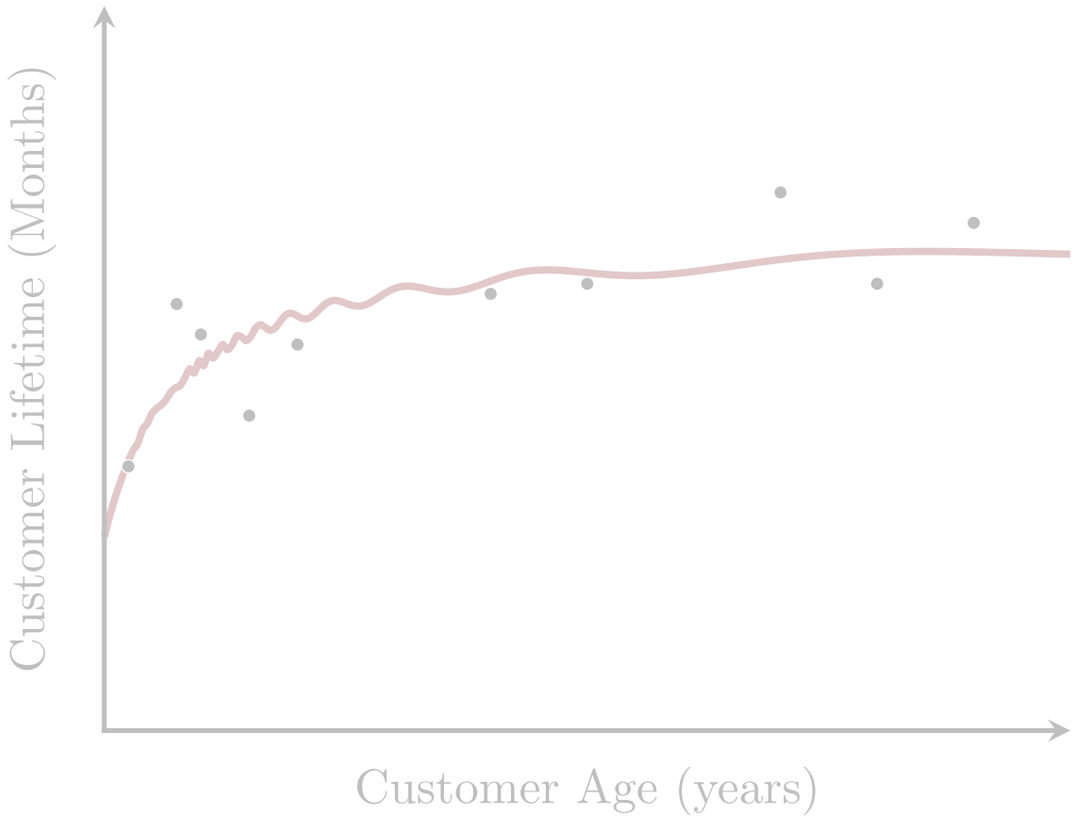
"All models are wrong but some are useful".

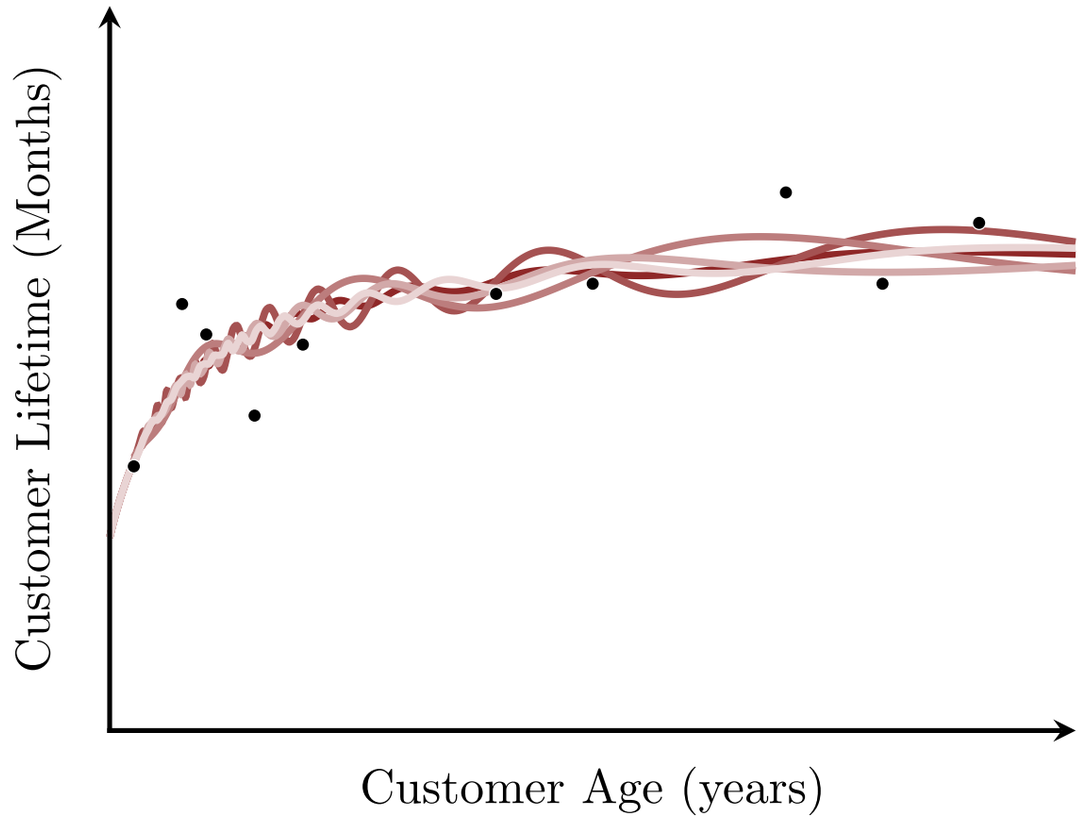
-George Box



"All models are wrong but some are useful".

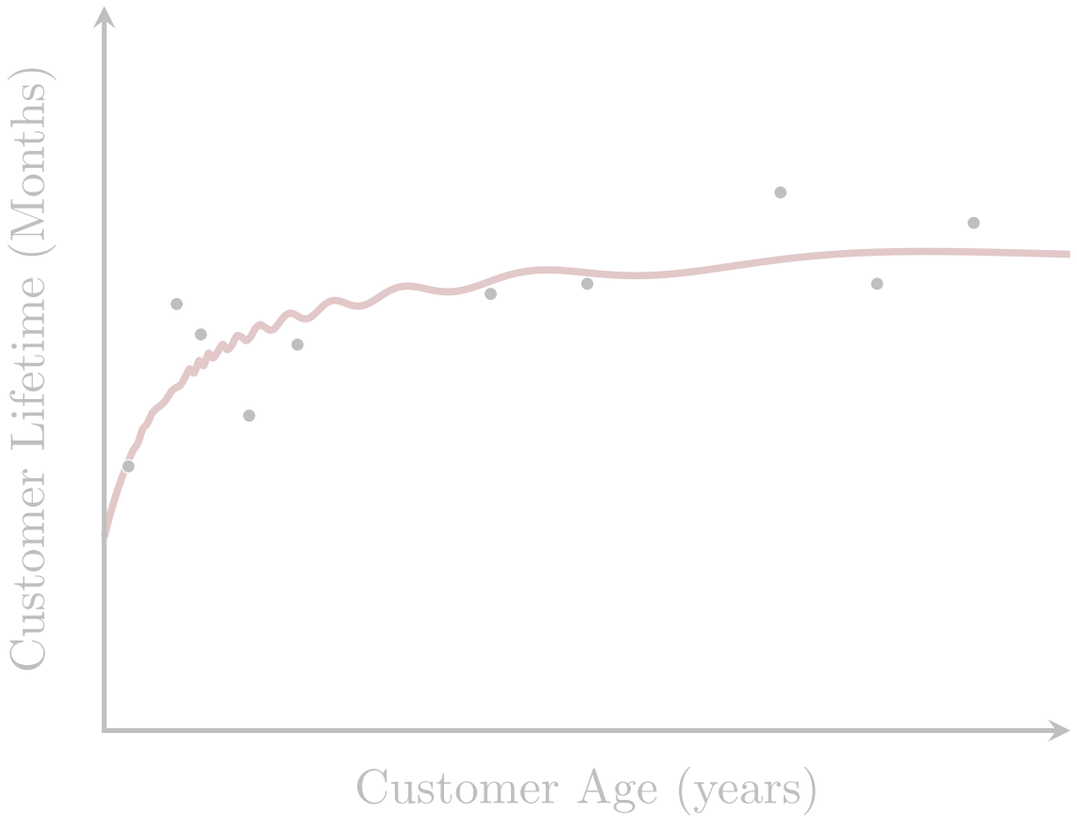
-George Box

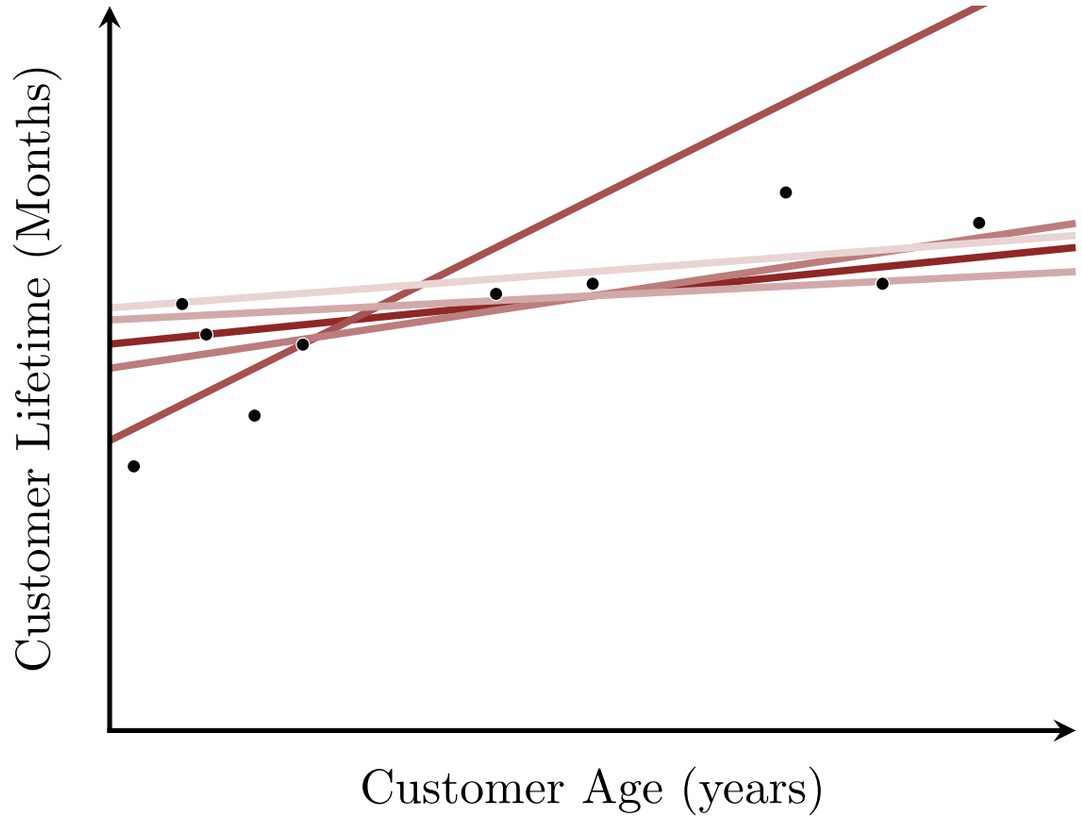




"All models are wrong but some are useful".

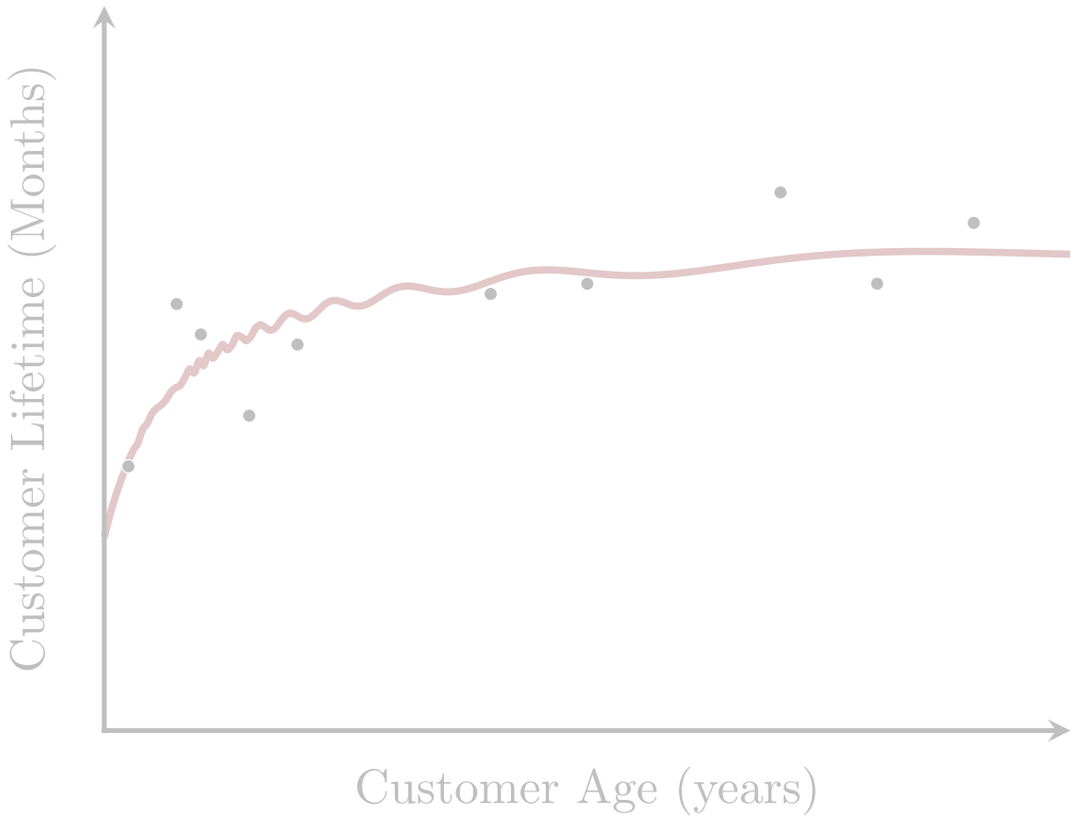
-George Box

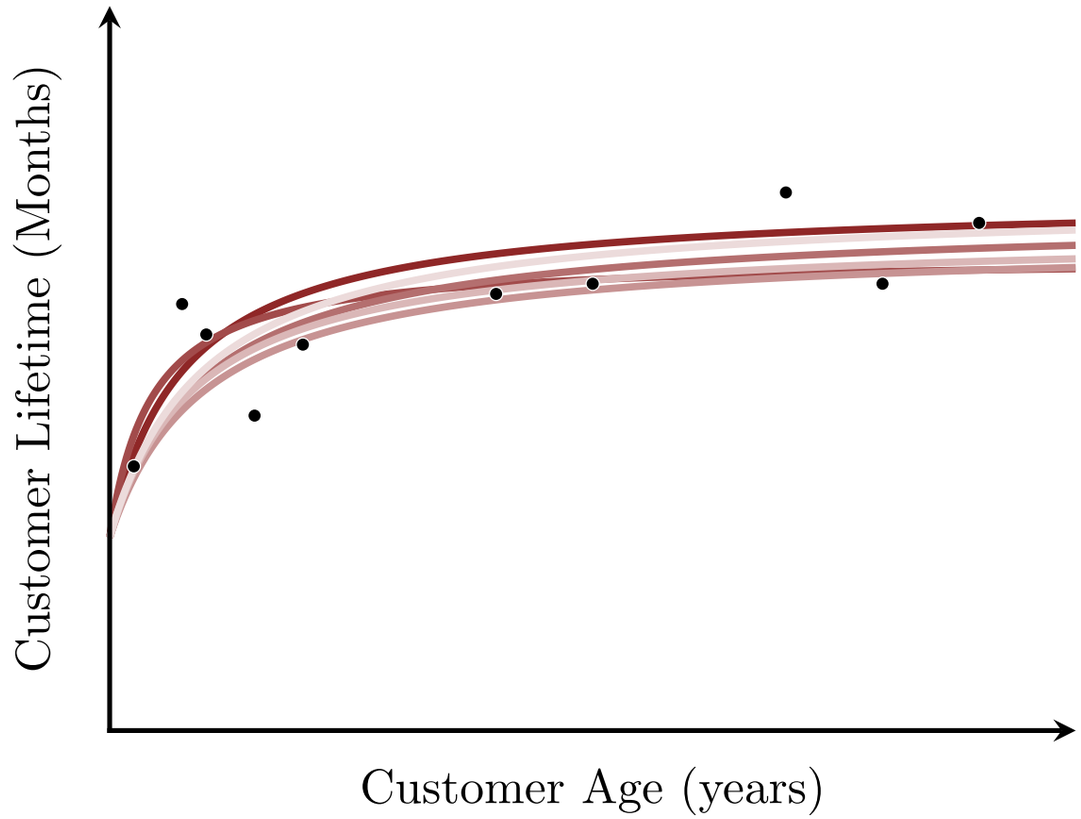




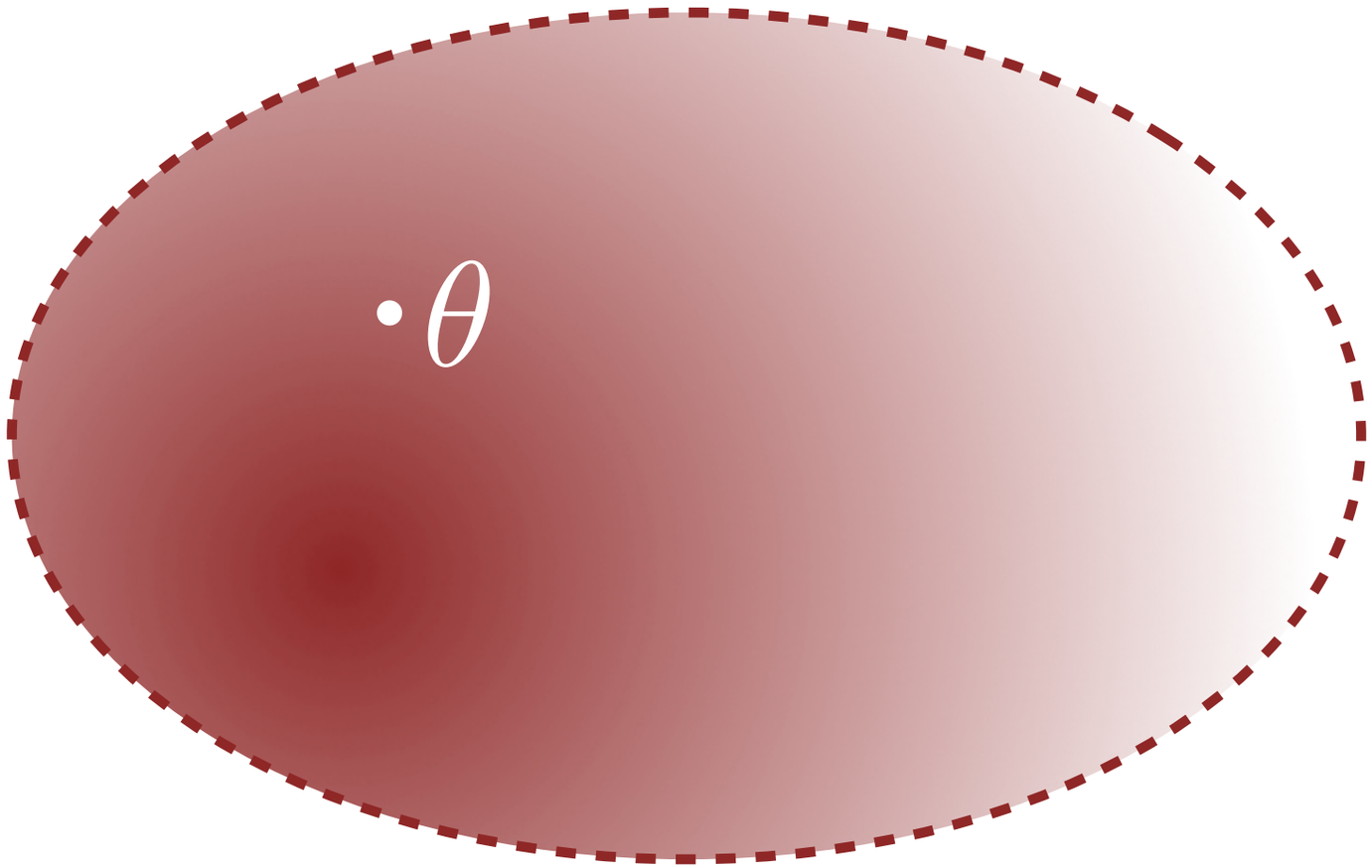
"All models are wrong but some are useful".

-George Box

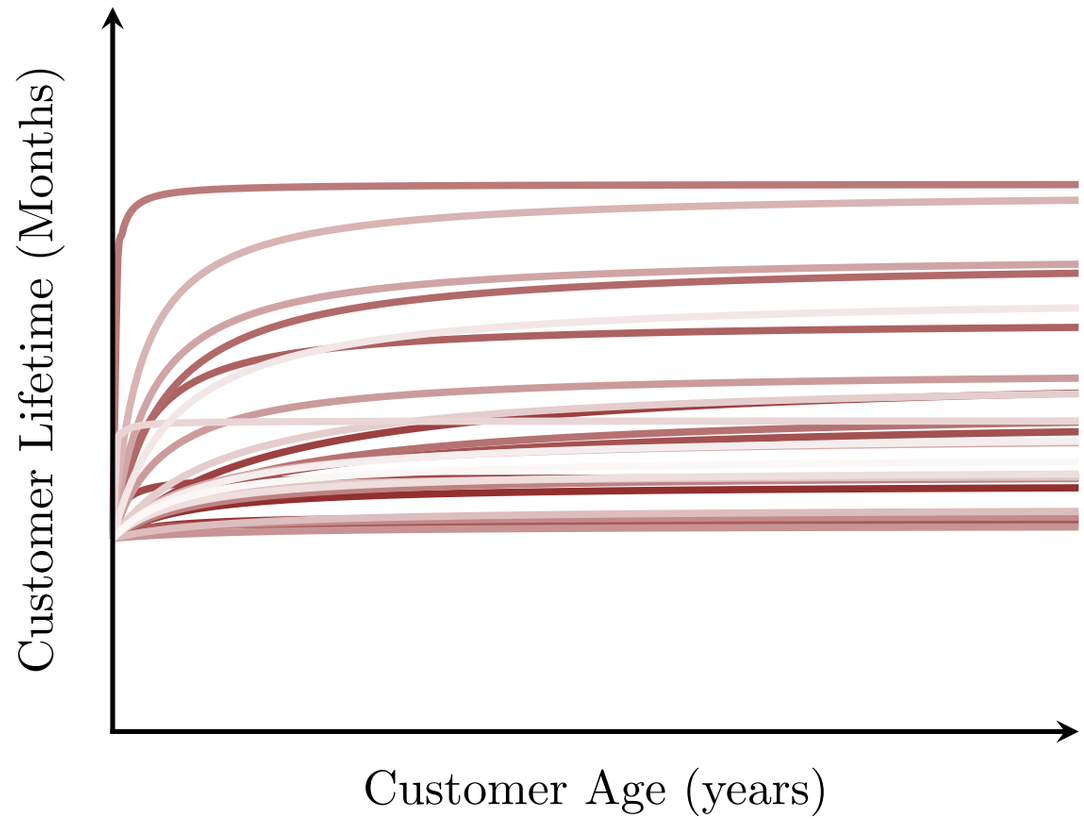




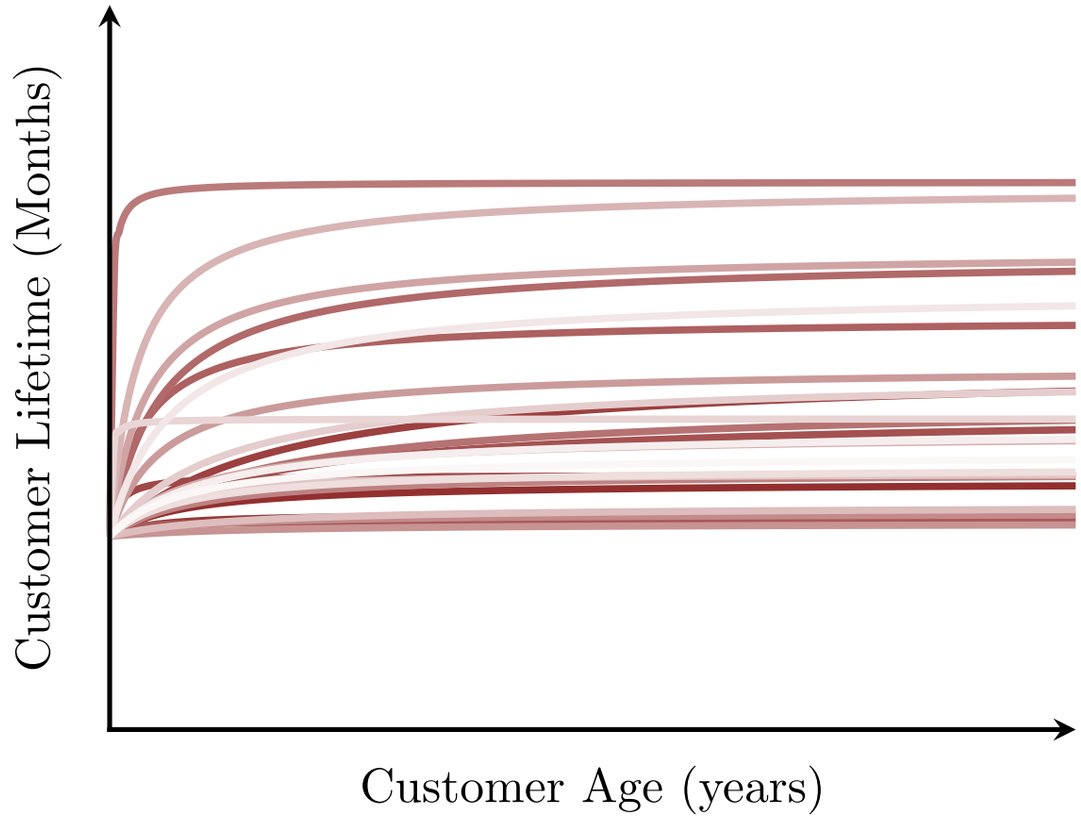
Bayesian Inference



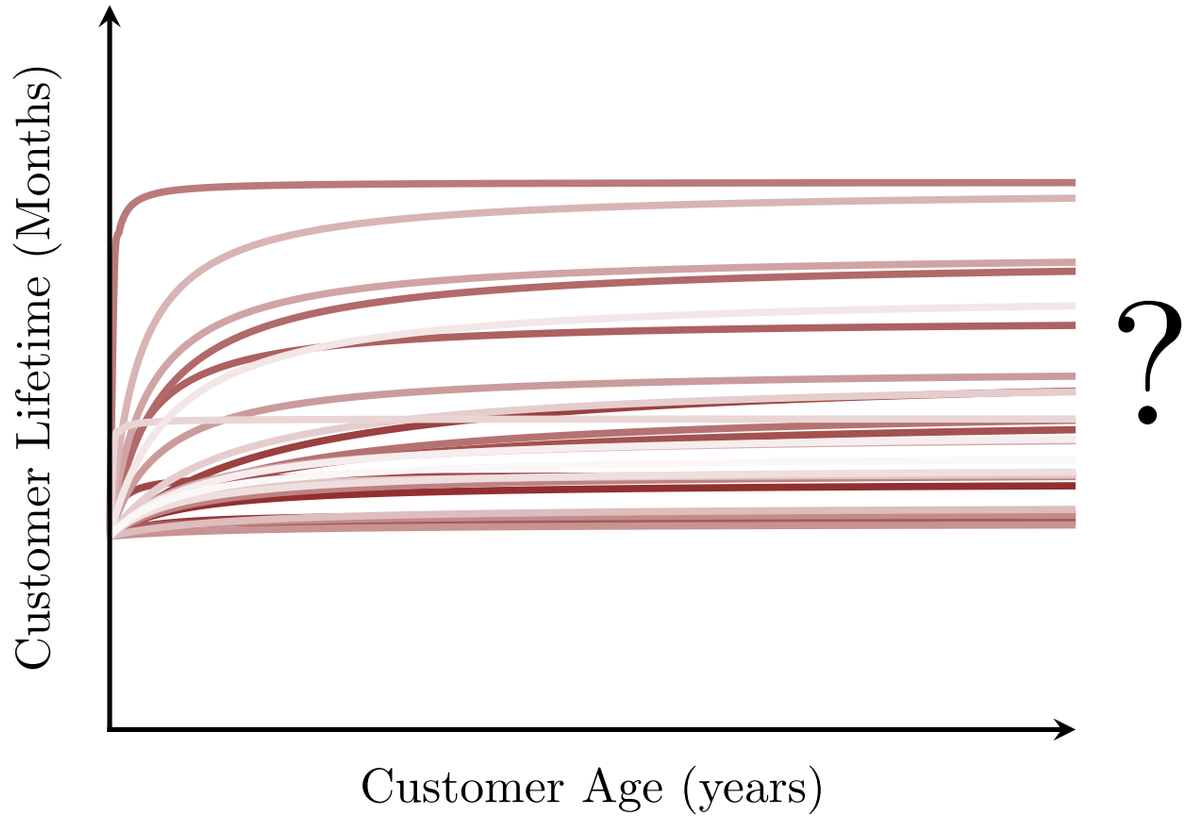
Model-based inference is any quantification of how *consistent* model configurations are with any observed data.



Model-based inference is any quantification of how *consistent* model configurations are with any observed data.

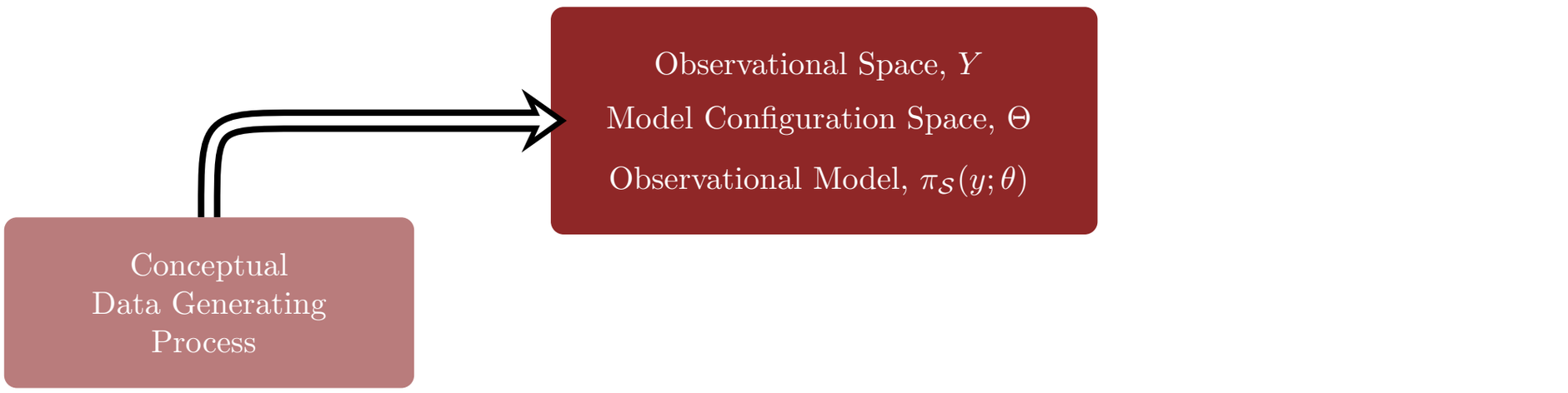


Model-based inference is any quantification of how *consistent* model configurations are with any observed data.



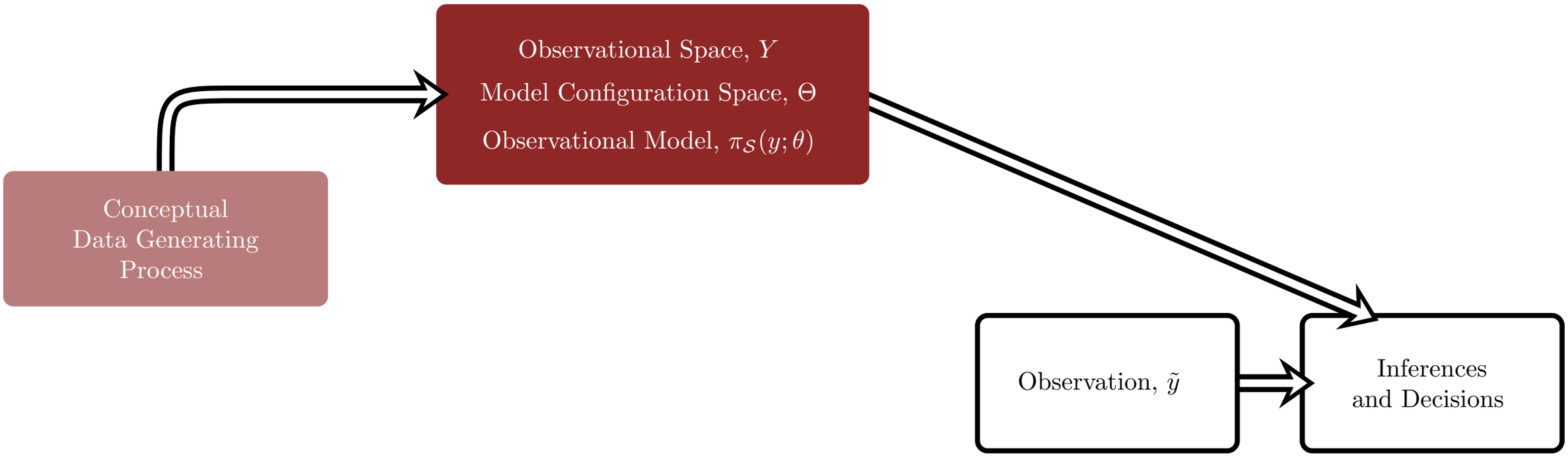
Conceptually, an observational model bridges the gap between observations and a true data generating process.

Conceptually, an observational model bridges the gap between data and the true data generating process.

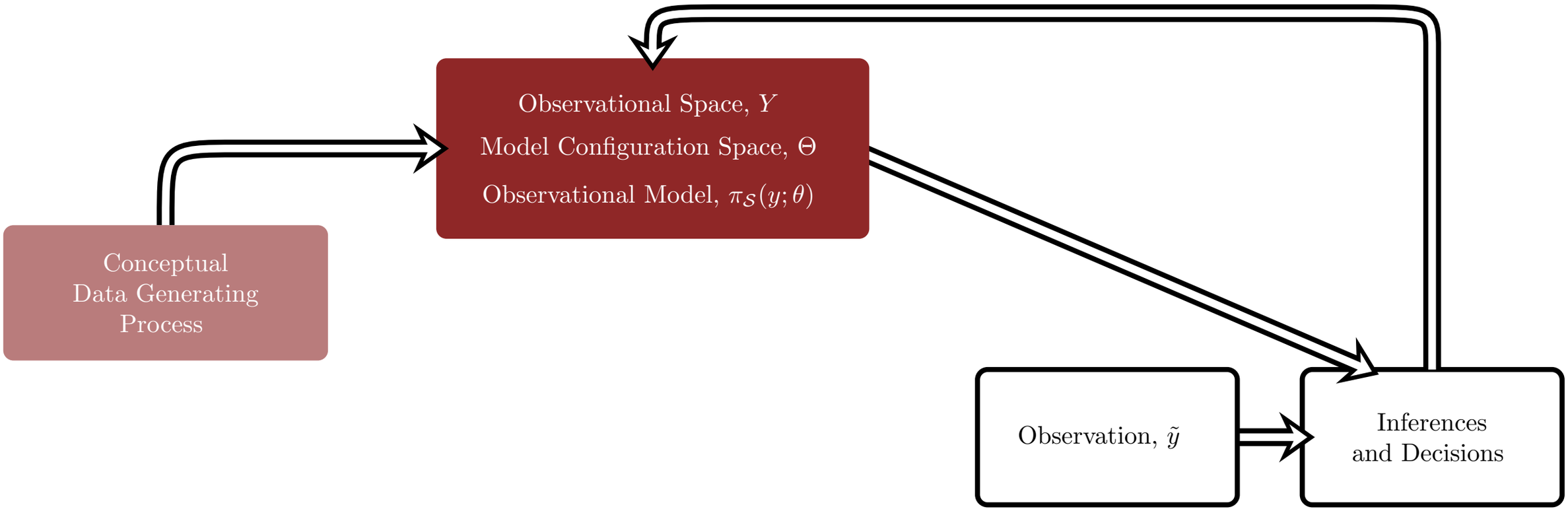


Observation, $\tilde{y}$

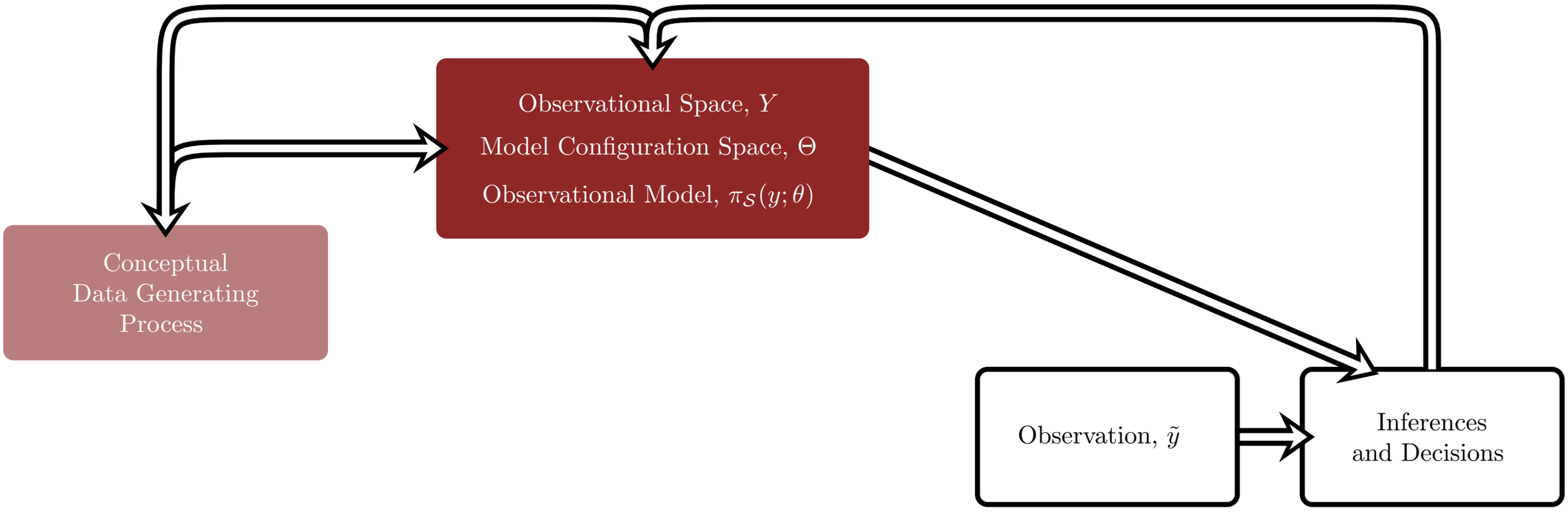
Conceptually, an observational model bridges the gap between data and the true data generating process.



Model-based inferences use observations to *learn* about the model configurations in an observational model.



What we learn then informs how we should interact with the true data generating process.



Bayesian inference models both the data and the model configuration variables *probabilistically*.

$$\pi(y \mid \theta)$$

Bayesian inference models both the data and the model configuration variables *probabilistically*.

$$\pi(y \mid \theta)\, \pi(\theta)$$

Bayesian inference models both the data and the model configuration variables *probabilistically*.

$$\pi(y, \theta) = \pi(y \mid \theta)\, \pi(\theta)$$

A full Bayesian model allows us to probabilistically quantify inferences with an application of *Bayes' Theorem*.

$$\pi_S(\theta \mid \tilde{y}) = \frac{\pi_S(\tilde{y} \mid \theta)}{\pi_S(\tilde{y})} \pi_S(\theta)$$

The *prior distribution* quantifies relevant domain expertise available before an observation is made.

$$\pi_S(\theta \mid \tilde{y}) = \frac{\pi_S(\tilde{y} \mid \theta)}{\pi_S(\tilde{y})} \pi_S(\theta)$$

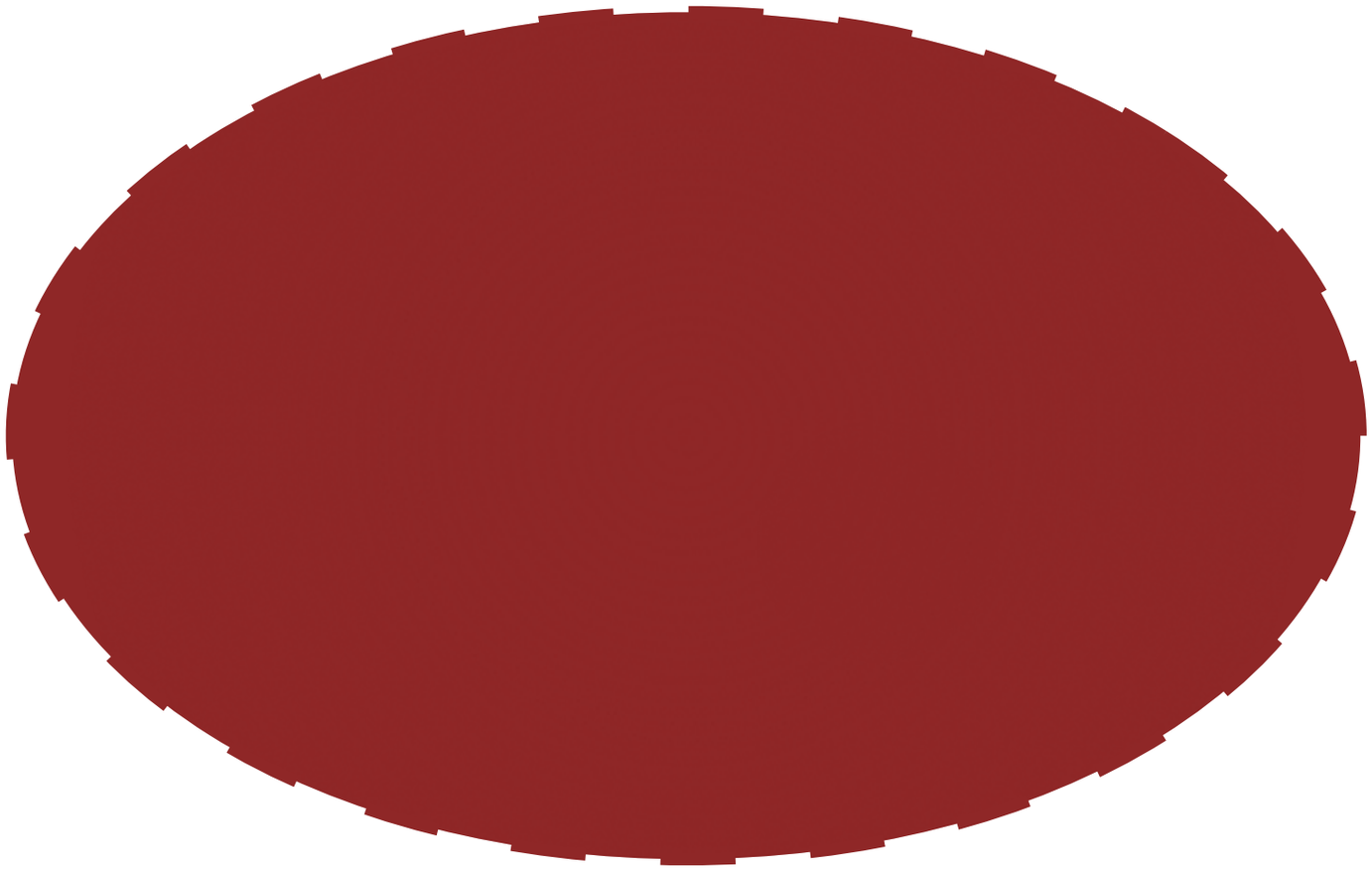
The *likelihood ratio* quantifies what we learn about the small world from the given observation.

$$\pi_S(\theta \mid \tilde{y}) = \frac{\pi_S(\tilde{y} \mid \theta)}{\pi_S(\tilde{y})} \pi_S(\theta)$$

The posterior aggregates what we knew and what we learned into what we know *after* the observation.

$$\pi_S(\theta \mid \tilde{y}) = \frac{\pi_S(\tilde{y} \mid \theta)}{\pi_S(\tilde{y})} \pi_S(\theta)$$

The posterior quantifies the configurations consistent with *both* our domain expertise and the observed data.



$$\pi_{S^{-1}(\theta)}(\theta \mid \tilde{y})$$

Conveniently, Bayes' Theorem is equivalent to partially evaluating the full Bayesian model on the observed data.

$$\pi(\theta \mid \tilde{y}) \propto \pi(\tilde{y}, \theta)$$

Making Predictions

$$\tilde{y} \sim \pi_S(y \mid \theta)$$

We can generate predictions from any *single* data generating process.

$$\tilde{y} \sim \pi_S(y \mid \theta)$$

Generate predictions from an entire observational model requires aggregating these individual predictions together.

$$\tilde{y} \sim \pi_S(y \mid \tilde{\theta})$$

$$\ldots$$

$$\tilde{y}' \sim \pi_S(y \mid \tilde{\theta}')$$

$$\ldots$$

$$\tilde{y}'' \sim \pi_S(y \mid \tilde{\theta}'')$$

A full Bayesian model can generate predictions by probabilistically weighting the individual predictions.

$$\pi_S(y \mid \theta)$$

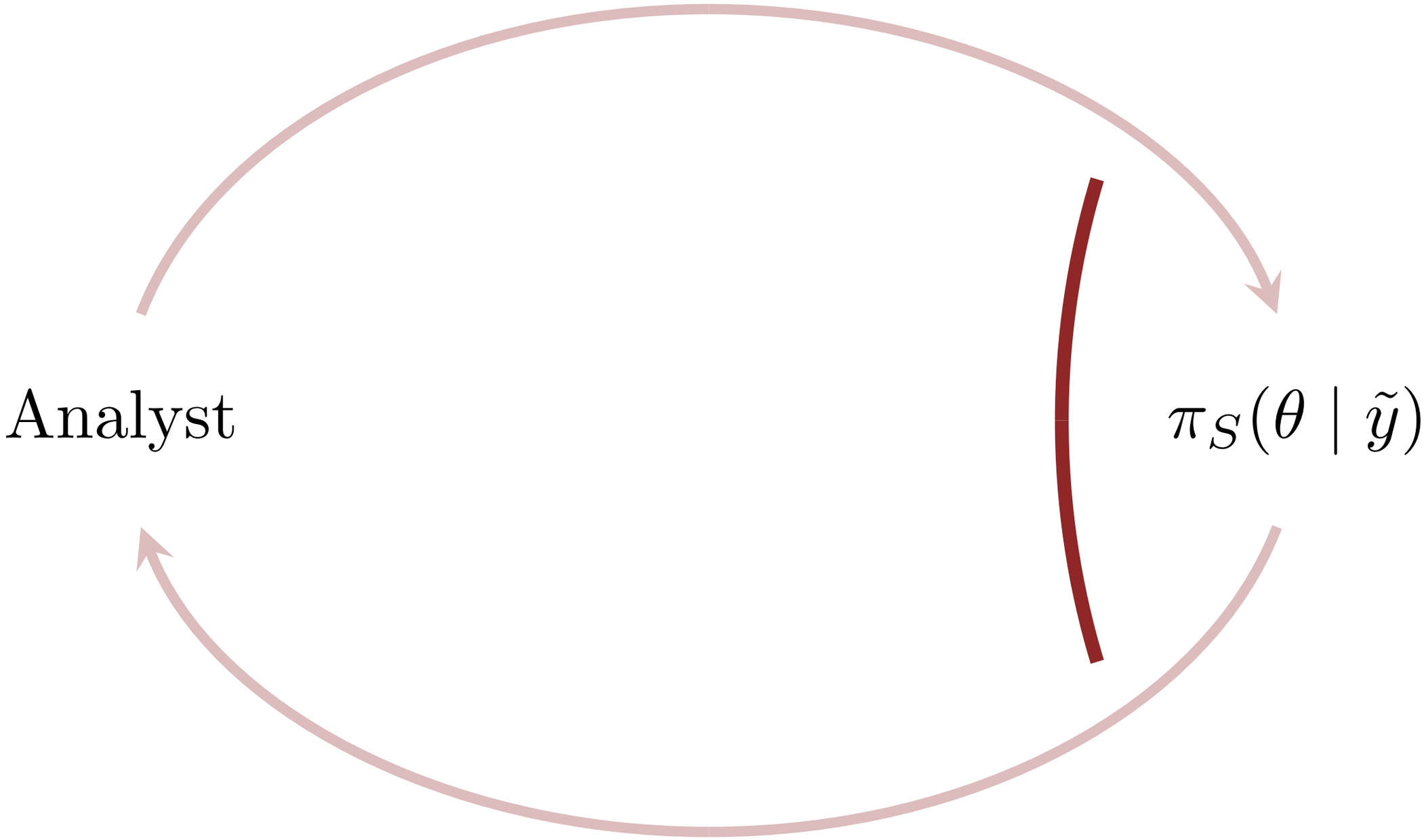
The *prior predictive distribution* weights the possible data generating processes with the prior distribution.

$$\pi_S(y) = \int \mathrm{d}\theta \, \pi_S(\theta) \pi_S(y \mid \theta)$$

The *posterior predictive distribution* weights the possible data generating processes with the posterior distribution.

$$\pi_S(y \mid \tilde{y}) = \int \mathrm{d}\theta \, \pi_S(\theta \mid \tilde{y}) \, \pi_S(y \mid \theta)$$

Utilizing The Posterior Distribution



Deriving a posterior density function from a full Bayesian model is a straightforward, *and inexpensive*, operation.

$$\pi(\theta \mid \tilde{y}) \propto \pi(\tilde{y}, \theta)$$

The expense of Bayesian inference is using the posterior distribution to answer specific inferential questions.

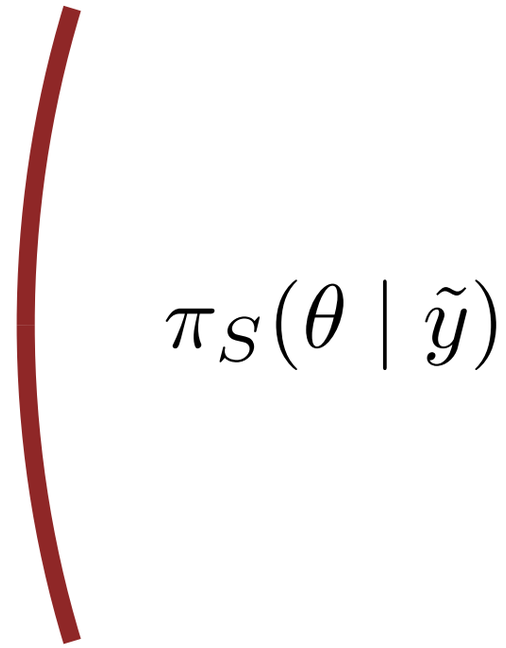
$$\pi(\theta \mid \tilde{y}) \propto \pi(\tilde{y}, \theta)$$

Well-defined inferential questions are coded as functions,
with the answers given by *posterior expectation values*.

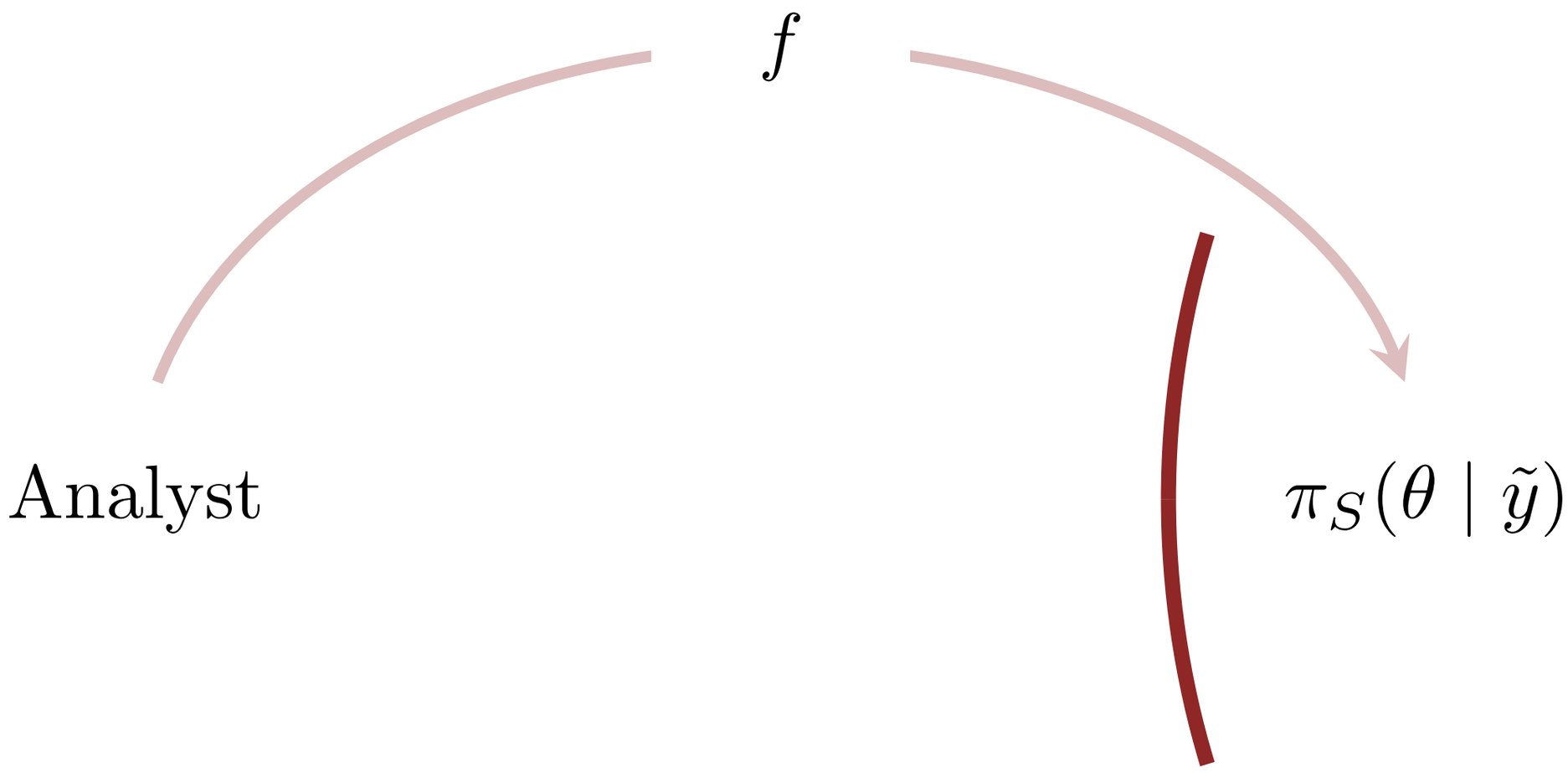
$$\mathbb{E}_{\pi}[f] = \int \mathrm{d}\theta \, \pi_S(\theta \mid \tilde{y}) \, f(\theta)$$

Well-defined inferential questions are coded as functions,
with the answers given by *posterior expectation values*.

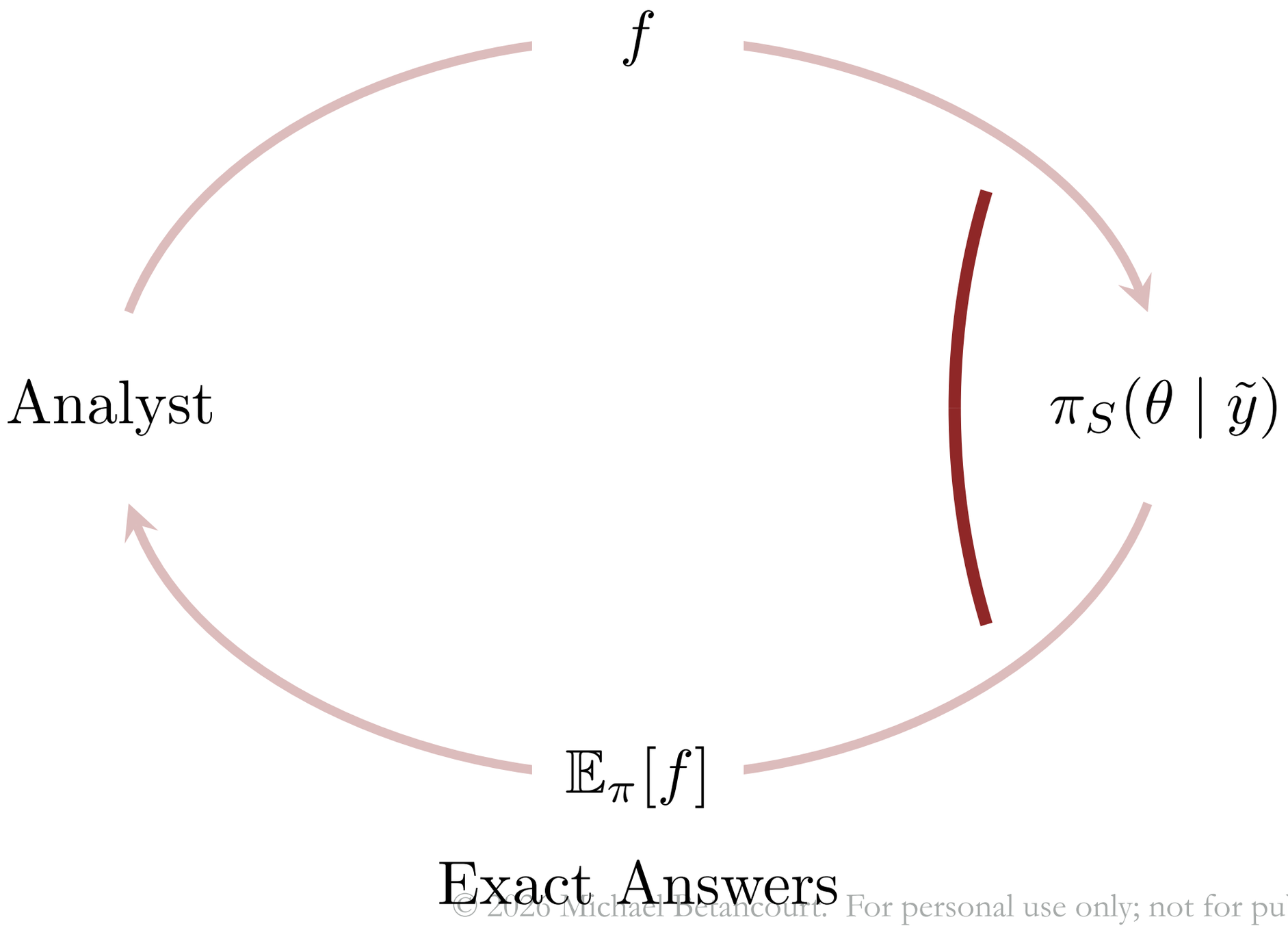
Analyst



Well-defined inferential questions are coded as functions,
with the answers given by *posterior expectation values*.



Well-defined inferential questions are coded as functions,
with the answers given by *posterior expectation values*.

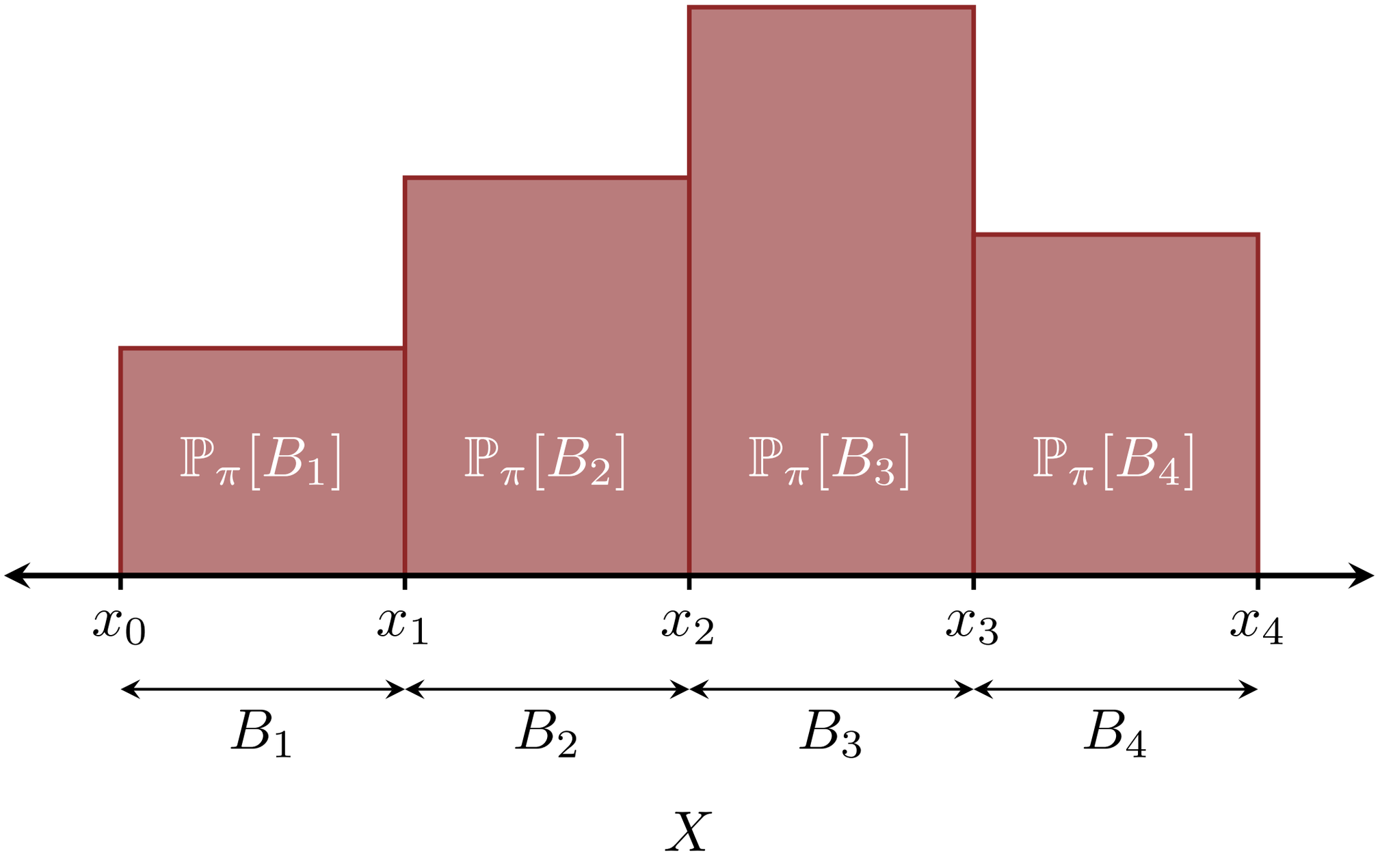


Expectations include means and variances for quantifying the centrality and breadth of the posterior distribution.

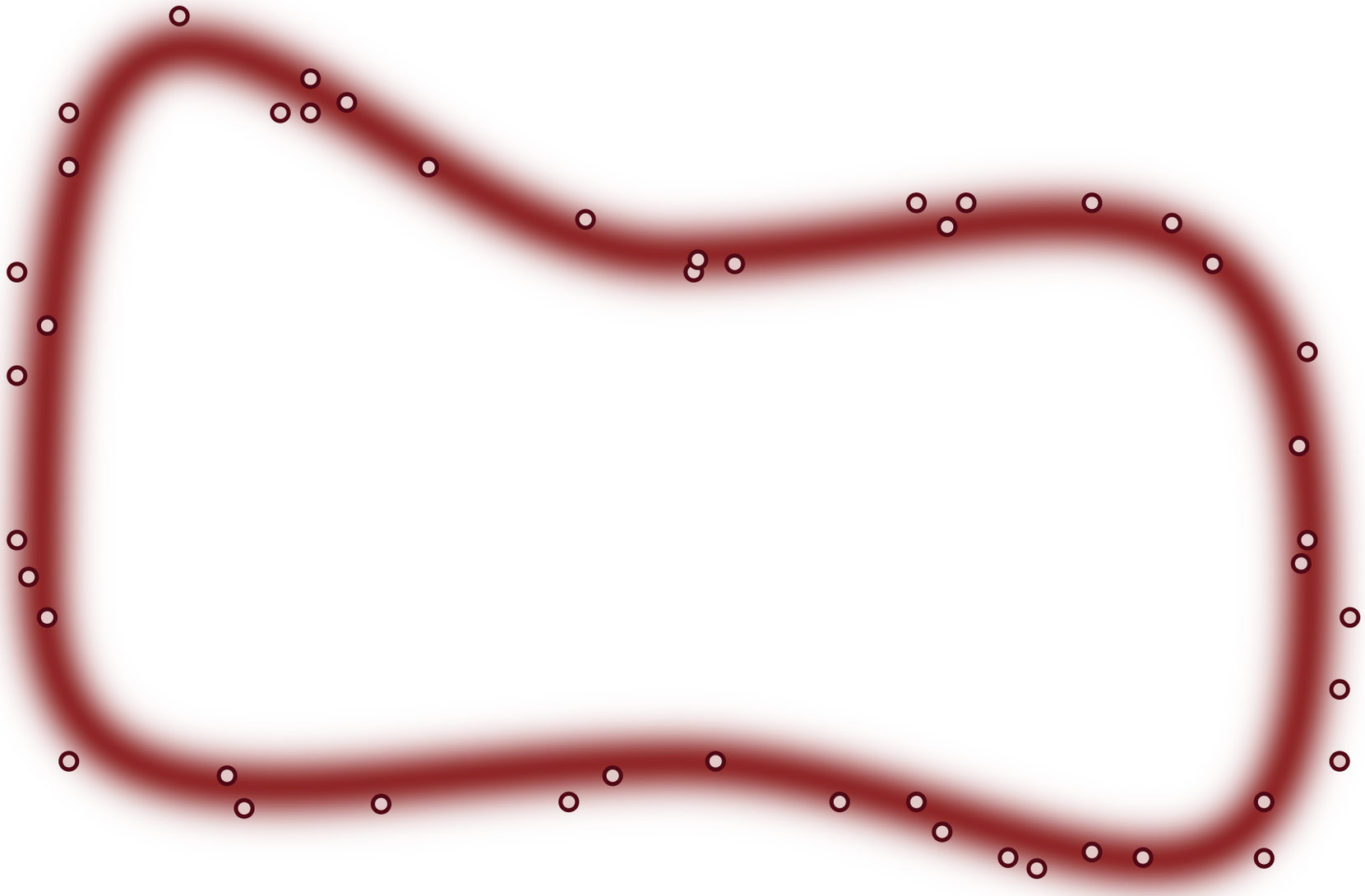
$$\mu = \int \mathrm{d}\theta \, \pi_S(\theta \mid \tilde{y}) \, \theta$$

$$\sigma^2 = \int \mathrm{d}\theta \, \pi_S(\theta \mid \tilde{y}) \, \left(\theta^2 - \mu^2\right)$$

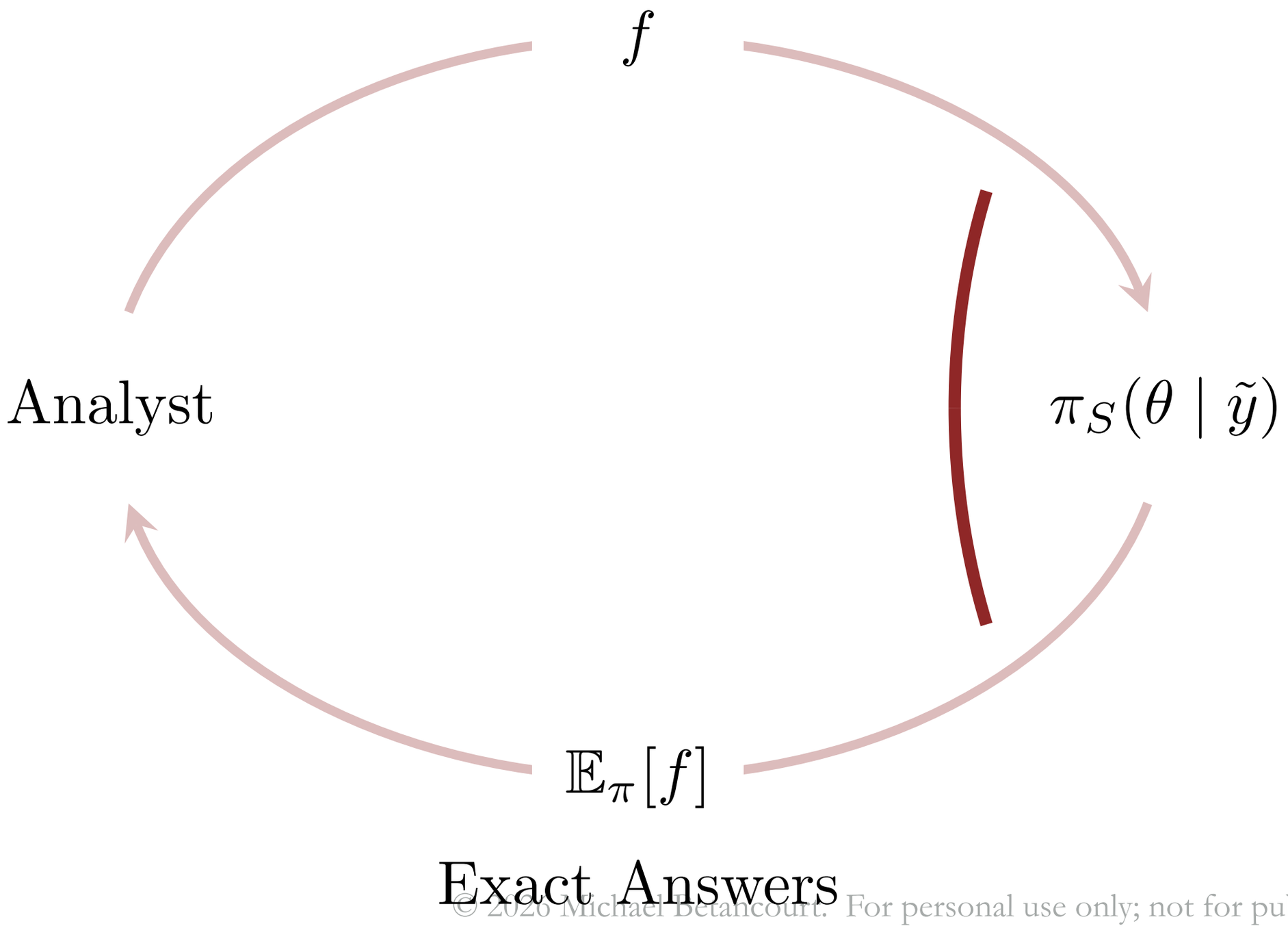
Expectations also include interval probabilities which we can use to construct histograms.



Probabilistic Computation



Well-defined inferential questions are coded as functions,
with the answers given by *posterior expectation values*.



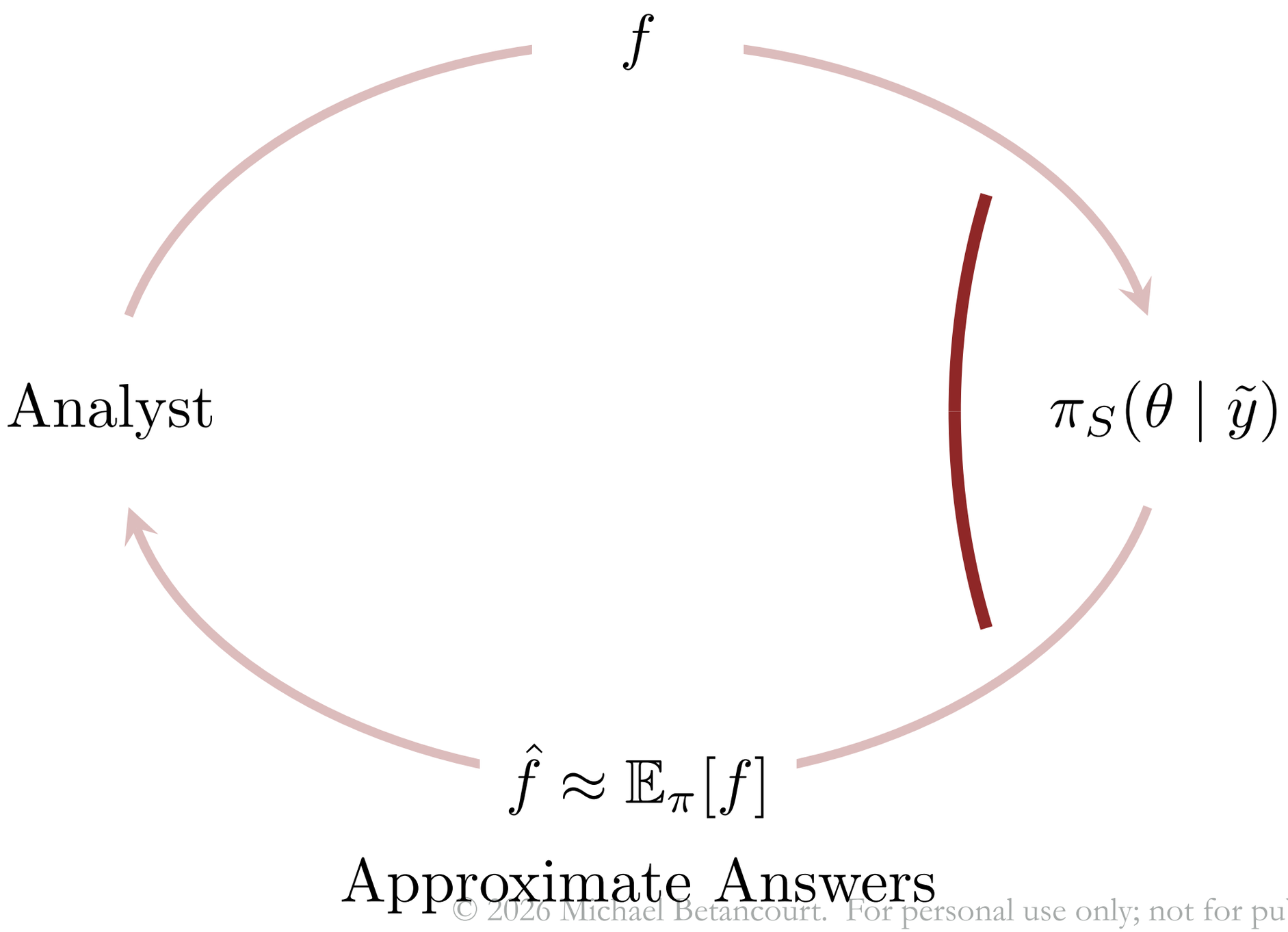
Well-defined inferential questions are coded as functions,
with the answers given by *posterior expectation values*.

$$\mathbb{E}_{\pi}[f] = \int \mathrm{d}\theta \, \pi_S(\theta \mid \tilde{y}) \, f(\theta)$$

In practice, we are typically limited to approximate expectation values, and hence approximate inferences.

$$\hat{f} \approx \int \mathrm{d}\theta \, \pi_S(\theta \mid \tilde{y}) \, f(\theta)$$

In practice, we are typically limited to approximate expectation values, and hence approximate inferences.



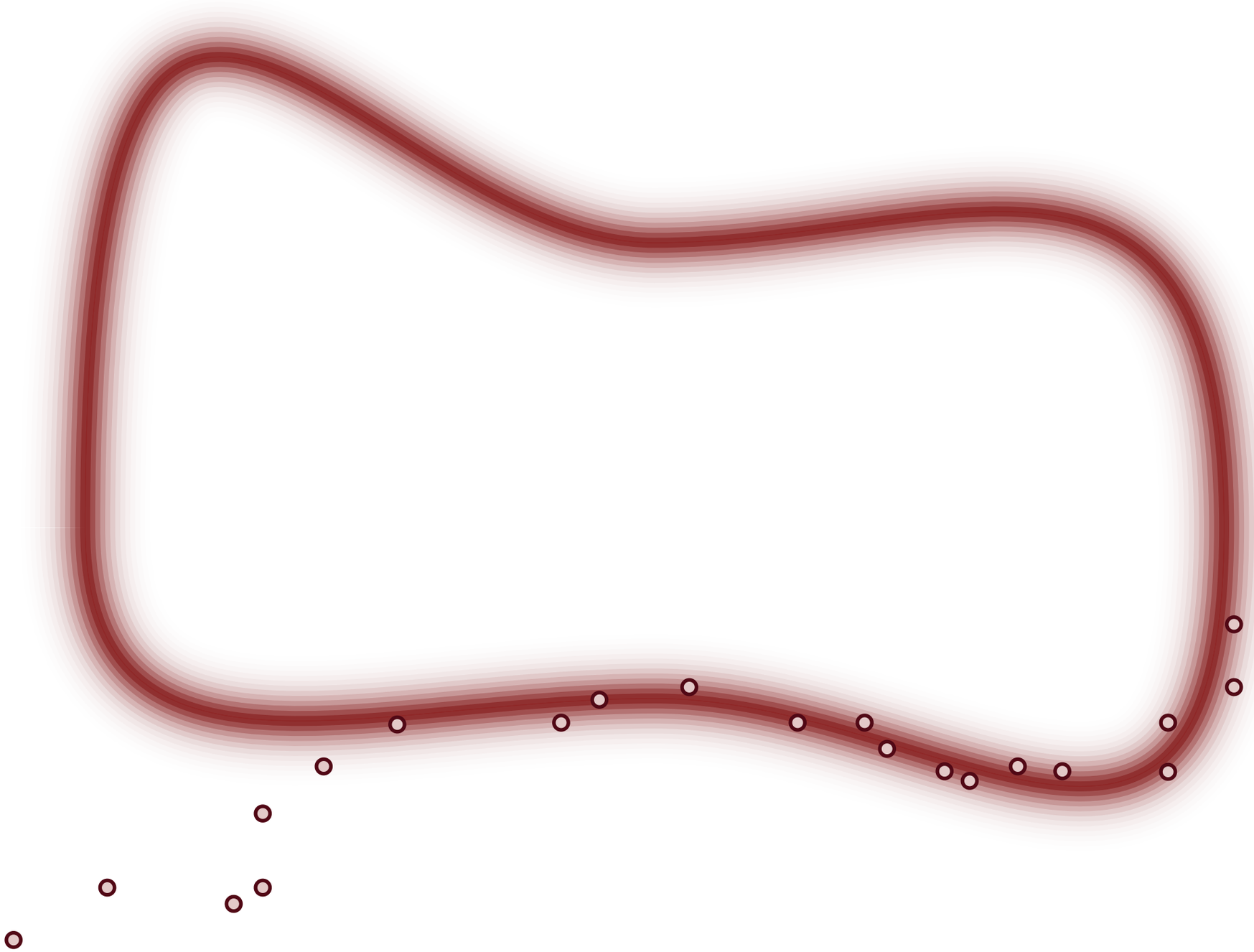
To implement Bayesian inference in practice we need to be able to well-approximate high-dimensional integrals.

$$\hat{f} \approx \int \mathrm{d}\theta \, \pi_{\mathcal{S}}(\theta \mid \tilde{y}) \, f(\theta)$$

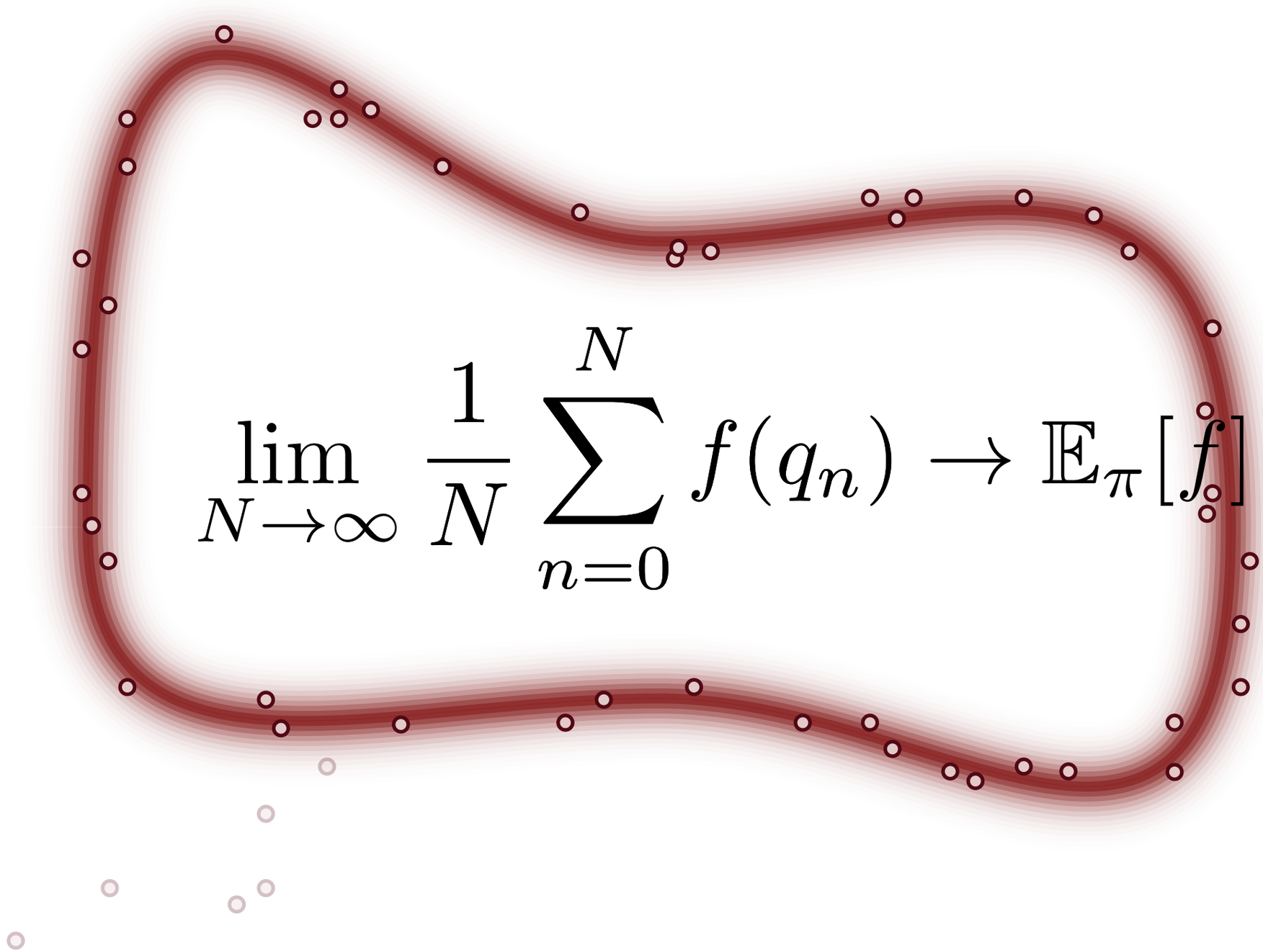
To implement Bayesian inference in practice we need to be able to well-approximate high-dimensional integrals.

$$\hat{f} \approx \int \mathrm{d}q \, \pi(q) \, f(q)$$

Markov chain Monte Carlo uses *Markov chains* to quantify the important regions of a posterior distribution.

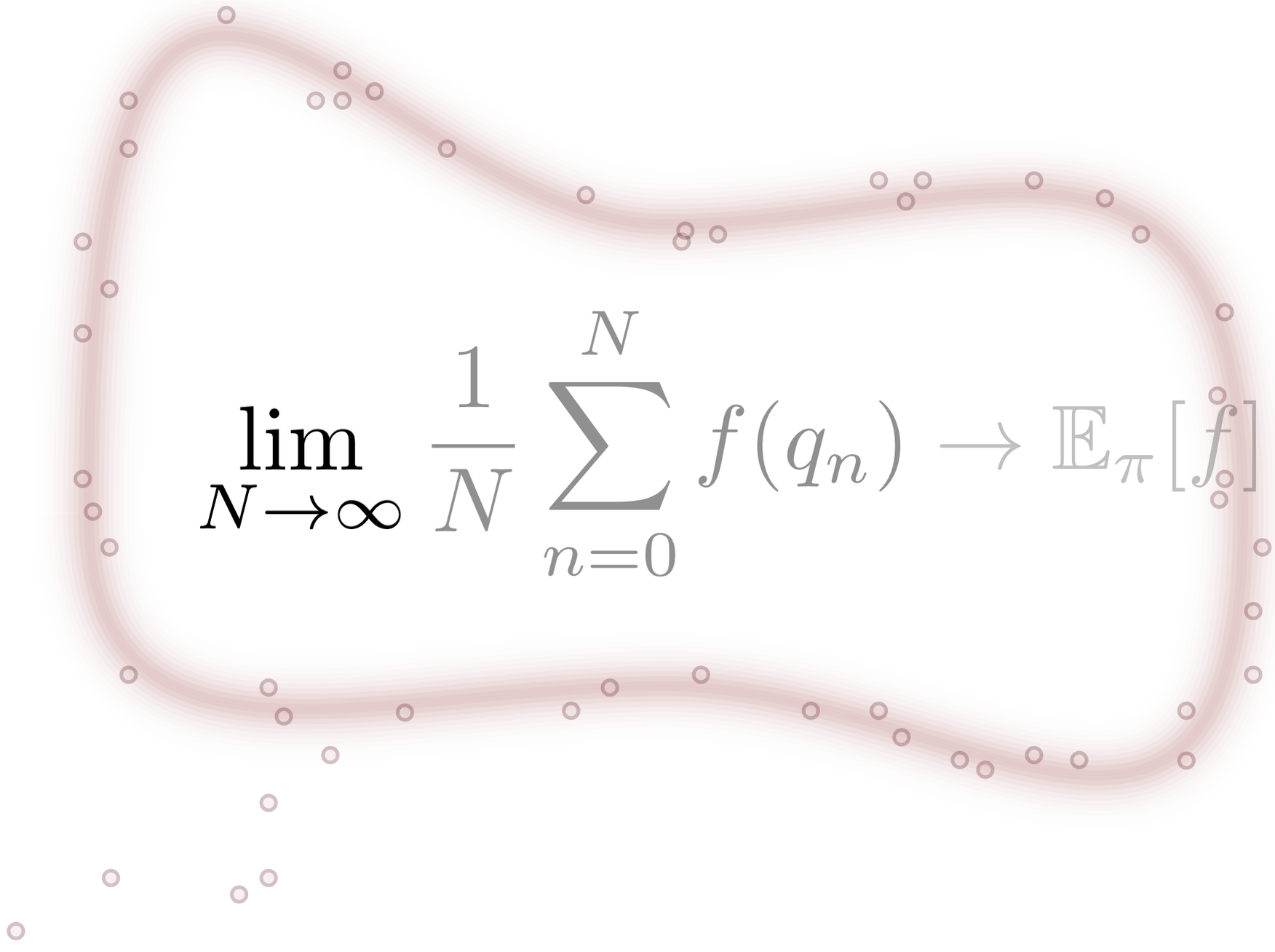


Averaging over a Markov chain defines an asymptotically consistent *Markov Chain Monte Carlo estimator* for integrals.



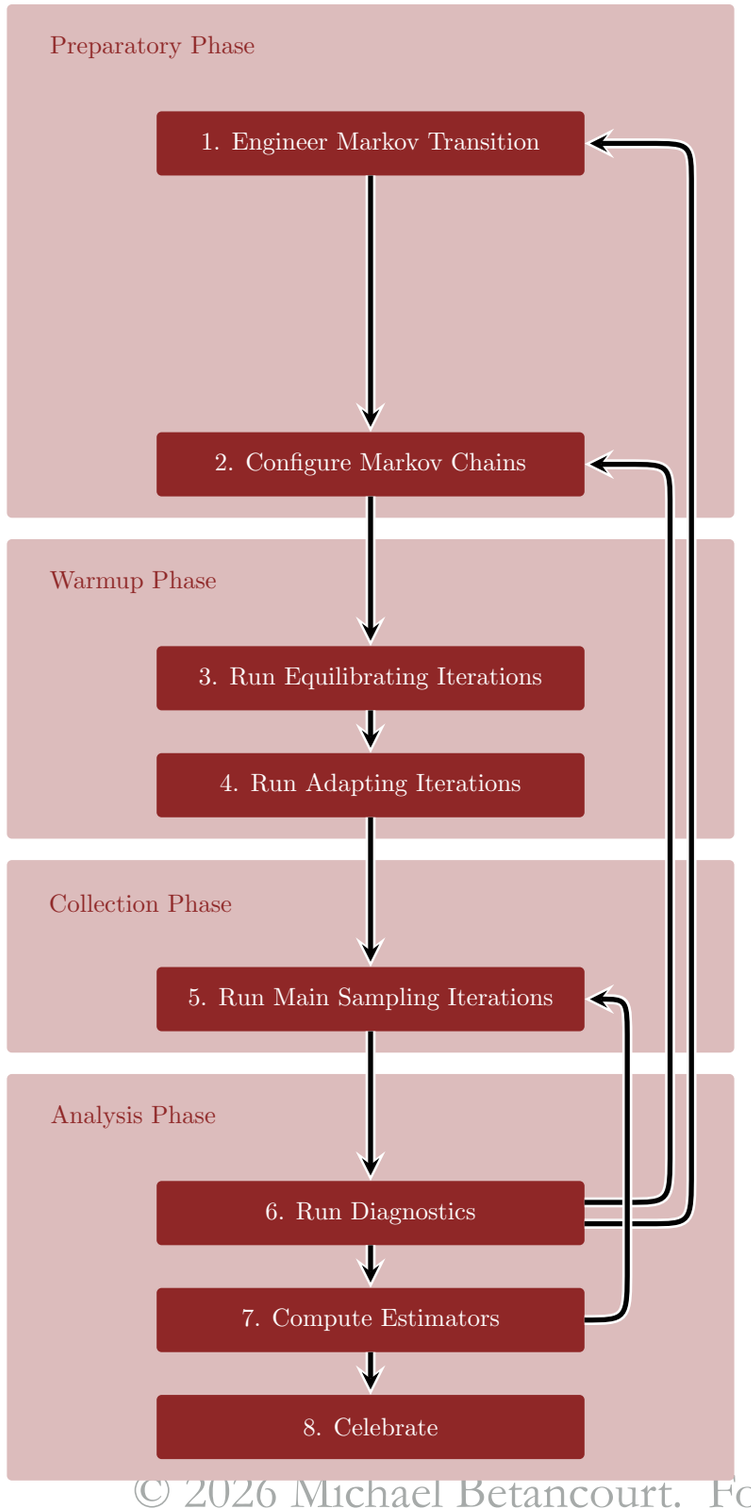
$$\lim_{N \to \infty} \frac{1}{N} \sum_{n=0}^{N} f(q_n) \to \mathbb{E}_{\pi}[f]$$

Unfortunately this asymptotic consistency makes no guarantees on the *finite time* behavior of the estimators.



$$\lim_{N \to \infty} \frac{1}{N} \sum_{n=0}^{N} f(q_n) \to \mathbb{E}_{\pi}[f]$$

In practice we rely on empirical diagnostics to determine when estimators accurately quantify expectation values.



The Stan Probabilistic Programming Language

A Stan program defines data variables, model configuration variables, and a joint density relating them.

$$\pi(y, \theta)$$

More precisely, it defines a *log* density function, which is more numerically stable and computationally convenient.

$$\begin{aligned} \log \pi(y, \theta) = &\quad \log \pi(y \mid \theta) \\ &+ \log \pi(\theta) \\ &+ \text{const} \end{aligned}$$

Log joint density functions can often be constructed incrementally from each individual contribution.

$$\log \pi(y, \theta) = \sum_{n} \log \pi(y_n \mid \theta) + \sum_{m} \log \pi(\theta_m) + \text{const}$$

Consider, for example, a straightforward model with a single parameter and N component observations.

$$\pi(y, \theta) = \prod_{n=1}^{N} \text{normal}(y_n \mid \theta, 1)$$
$$\times \text{normal}(\theta \mid 0, 1)$$

Consider, for example, a straightforward model with a single parameter and N component observations.

$$\log \pi(y, \theta) = \sum_{n=1}^{N} \log \circ \text{normal}(y_n \mid \theta, 1)$$
$$+ \log \circ \text{normal}(\theta \mid 0, 1)$$

A Stan program uses *programming blocks* to specify the structure of a full Bayesian model.

```
data {
  int N;
  array[N] real y;
}
```

$$\log \pi(y, \theta)$$

A Stan program uses *programming blocks* to specify the structure of a full Bayesian model.

```
data {
  int N;
  array[N] real y;
}

parameters {
  real theta;
}
```

$$\log \pi(y, \theta)$$

$$\log \pi(y, \theta)$$

A Stan program uses *programming blocks* to specify the structure of a full Bayesian model.

```
data {
  int N;
  array[N] real y;
}

parameters {
  real theta;
}

model {
  target += normal_lpdf(theta | 0, 1);
  for (n in 1:N)
    target += normal_lpdf(y[n] | theta, 1);
}
```

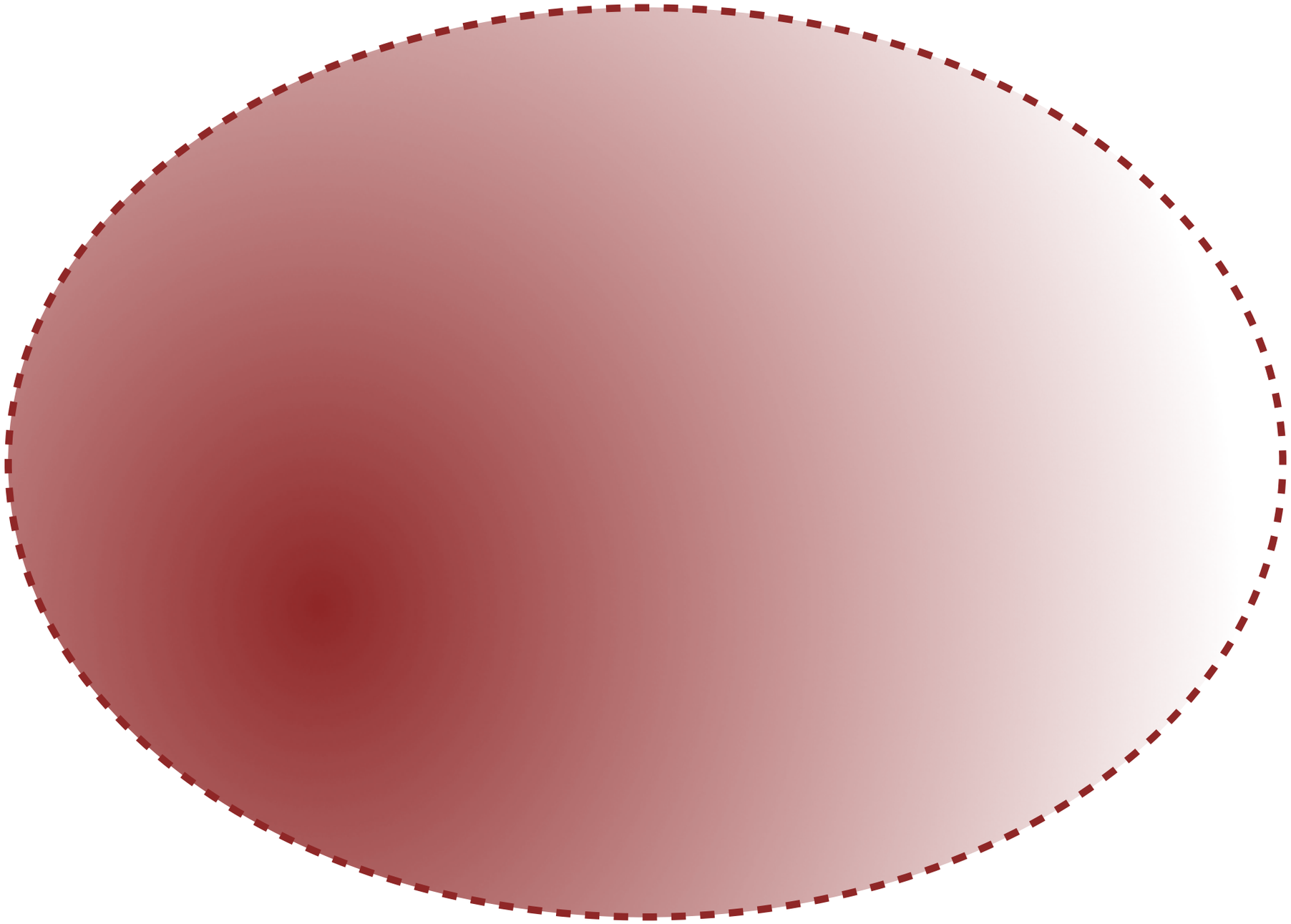
$$\log \pi(y, \theta)$$

$$\log \pi(y, \theta)$$

$$\log \pi(y, \theta)$$

Once the data and parameter variables have been specified, the log joint density is defined incrementally.

```
model {
  target += normal_lpdf(theta | 0, 1);
  for (n in 1:N)
    target += normal_lpdf(y[n] | theta, 1);
}
```



Exercises Material

part1/data/historical_data.json
part1/data/customer_segmentation.json