APPLYING BAYESIAN METHODS

BAYESIAN STATISTICS FOR ECOLOGISTS

IGB 18. TO 26. NOVEMBER 2019

RECALL BAYES' THEOREM

- We have seen the likelihood, and can understand it as a distribution
- The prior is conceptually quite similar

$$\underbrace{pr(\theta|X)}_{\text{posterior probability}} = \frac{\overbrace{pr(X|\theta)}^{\text{likelihood}}\overbrace{pr(\theta)}^{\text{prior probability}}}{\underbrace{pr(X)}_{\text{normalising constant}}}$$

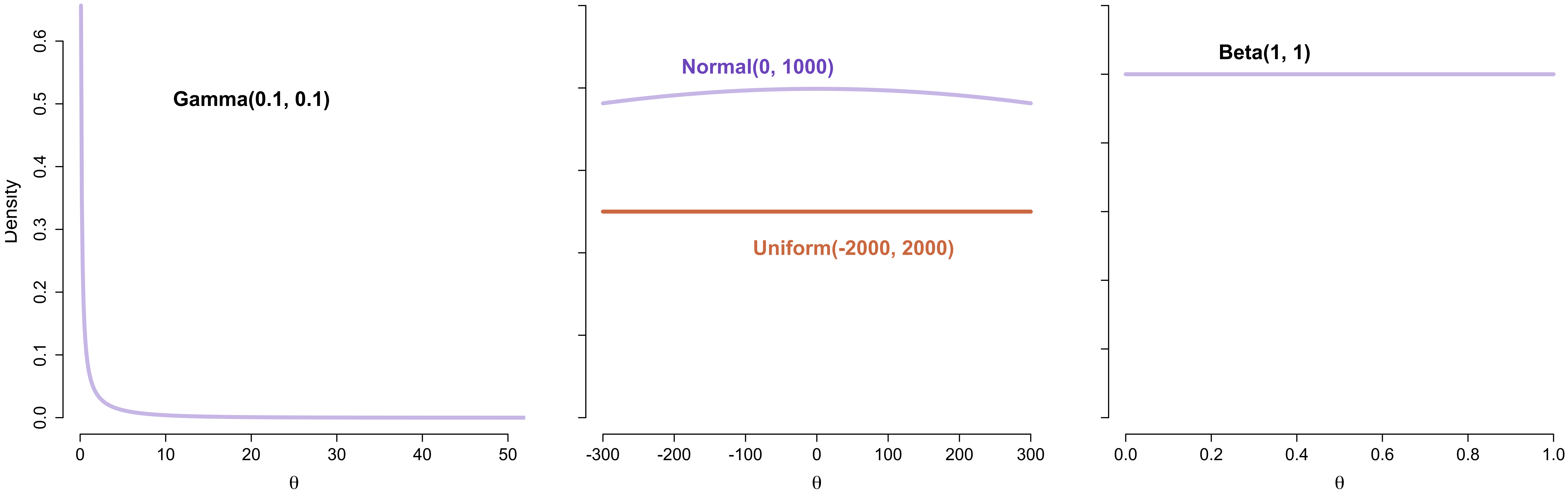
PRIOR DISTRIBUTIONS

- There is nothing special about the prior - could easily be called "Other information"
- Prior can take the same form as the information in the data, in which case, the likelihood times the prior looks just like evaluating the likelihood with new data + prior data (example coming)
- Often we know something, but the information is not as specific as the information in the data

TYPES OF PRIORS

- Priors come on a spectrum, from uninformative (sometimes called flat) to strongly informative
- The stronger the prior, the larger the role it takes in determining the shape of the posterior
- Priors can help a model when the signal in the data is weak (e.g., small sample size)

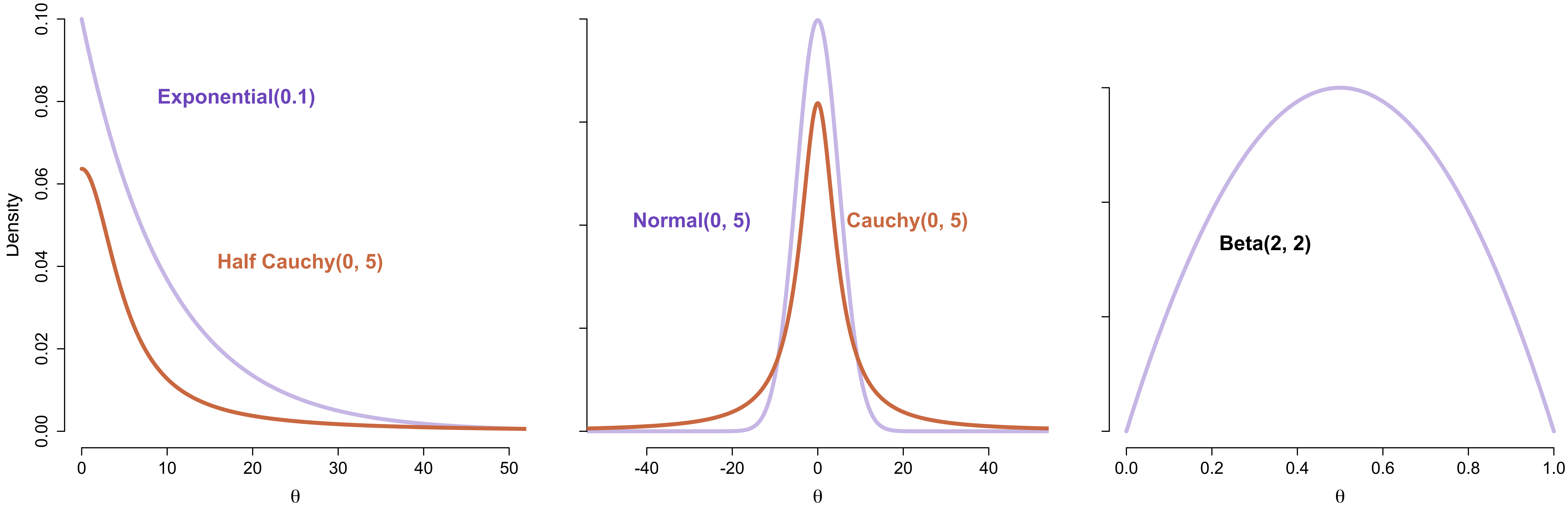
UNINFORMATIVE PRIORS



WEAKLY INFORMATIVE PRIORS

- Goal: regularise model (rule out unreasonable parameters) without ruling out sensible but extreme values
- Similar in concept to shrinkage in ridge regression/lasso
- Provide some information to the model to stabilise estimates and not spend too much time investigating the extremes of the posterior distribution
- Provide vague "meta-information" – e.g., the slope parameter of a logistic regression is usually < 10… Cauchy(0, 2.5) can accomplish this, as can Normal(0, 3)

WEAKLY INFORMATIVE PRIORS

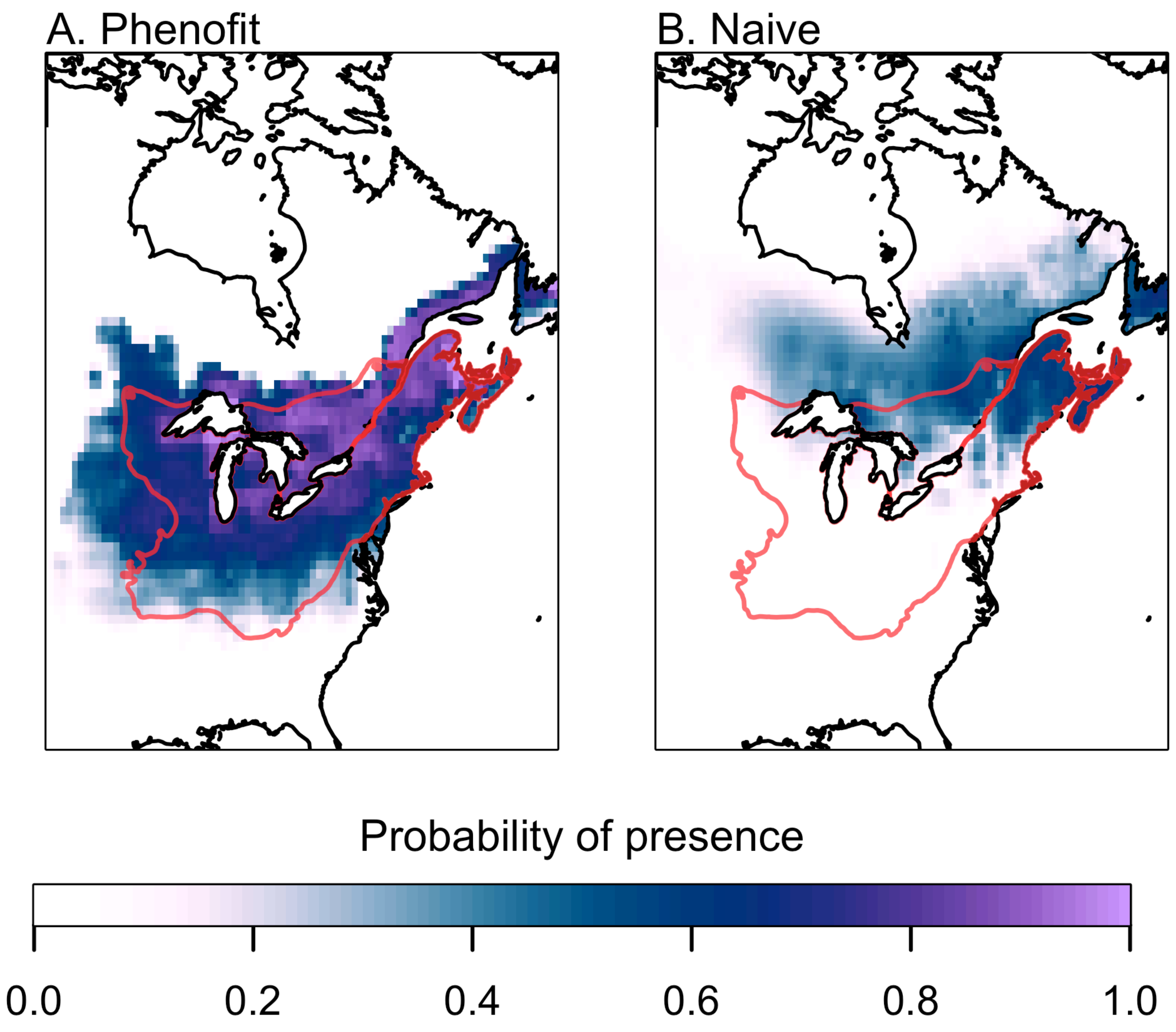


INFORMATIVE PRIORS

- Used when we have **specific information** about the problem and we wish to incorporate that into inference

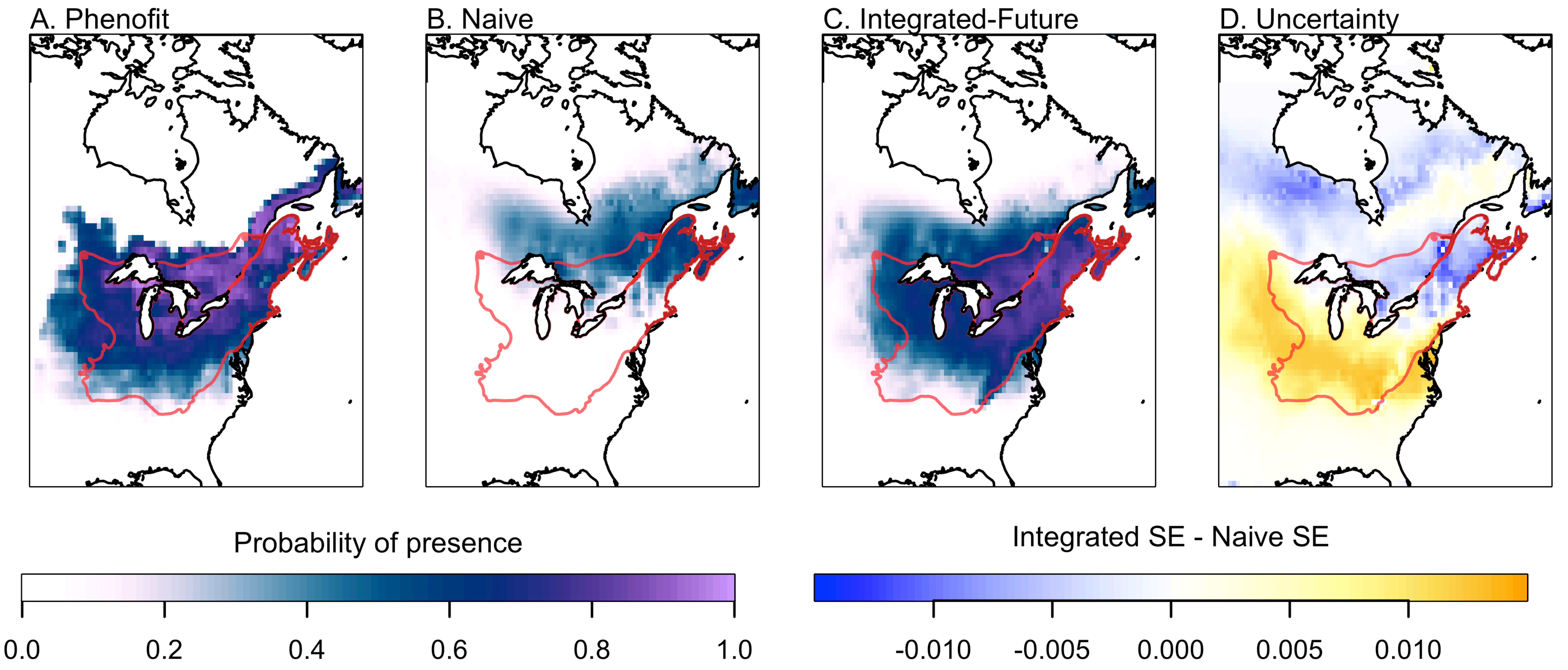
INFORMATIVE PRIORS

- Used when we have **specific information** about the problem and we wish to incorporate that into inference



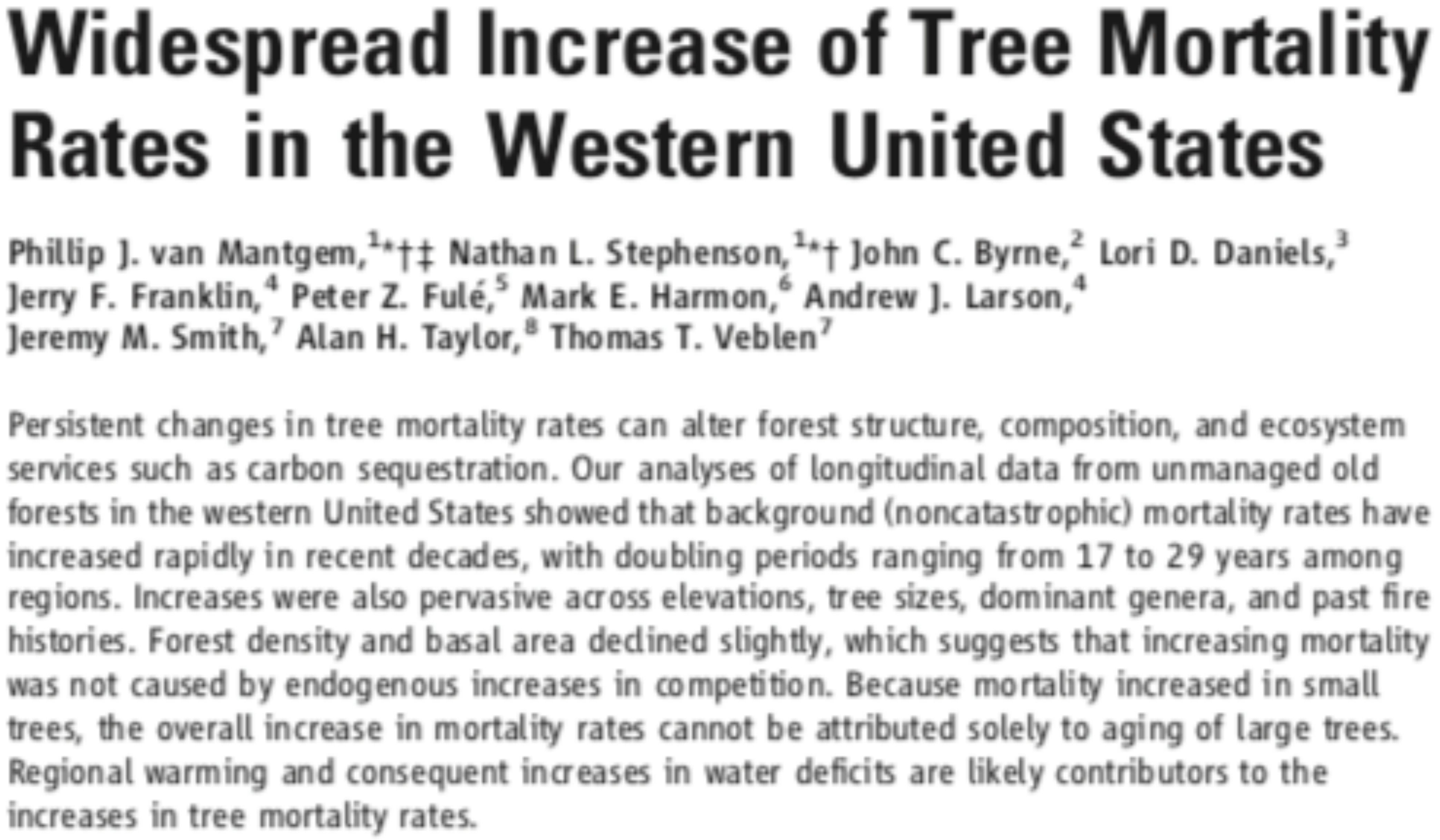
INFORMATIVE PRIORS

- Used when we have **specific information** about the problem and we wish to incorporate that into inference



ADDING PRIOR INFORMATION: TREE MORTALITY

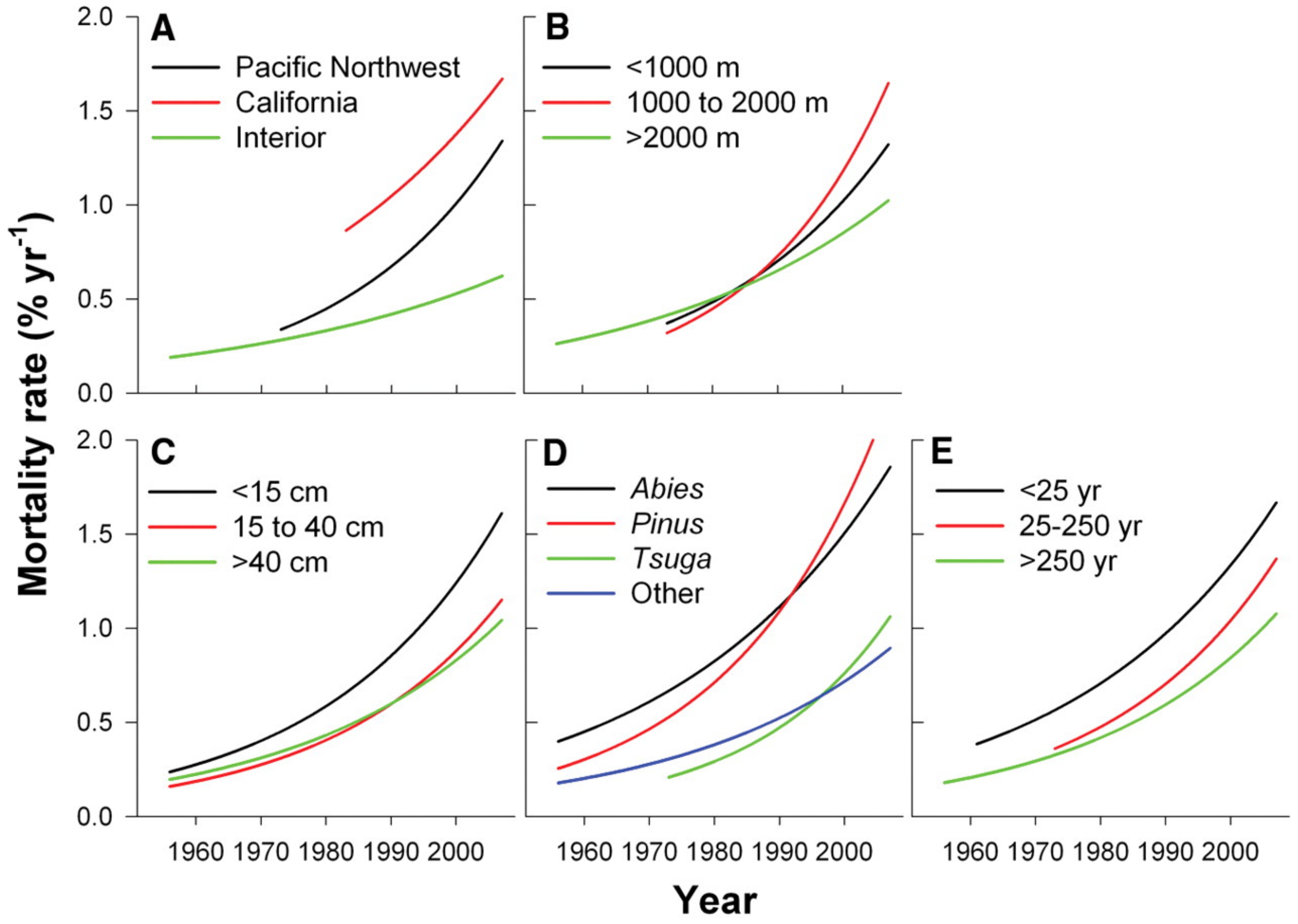
- Using the third data point from trees.rds
- 6 trees of 49 died
- We have prior information, average mortality prob of 0.087



Widespread Increase of Tree Mortality Rates in the Western United States

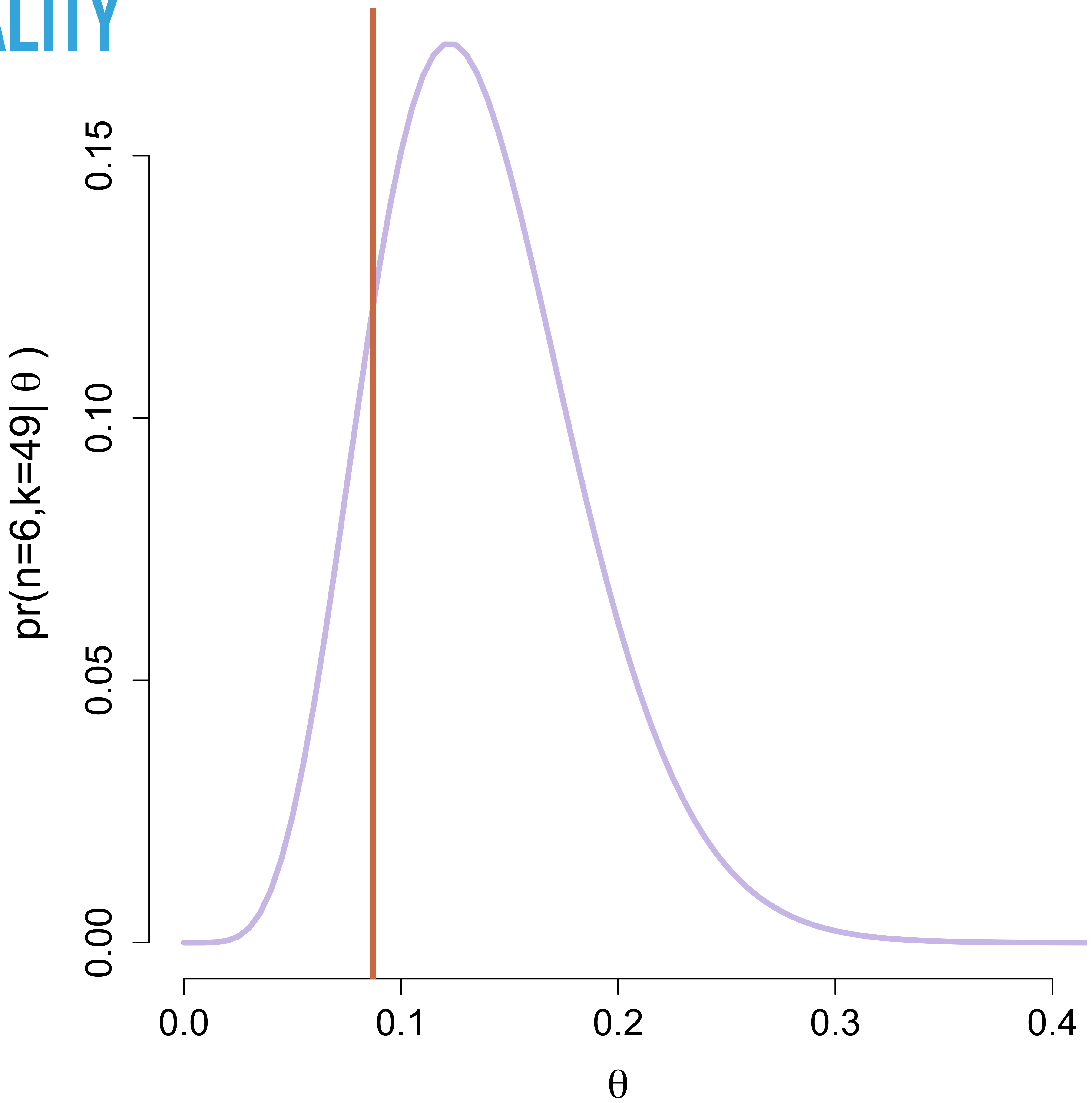
Phillip J. van Mantgem,[1*†‡] Nathan L. Stephenson,[1*†] John C. Byrne,[2] Lori D. Daniels,[3] Jerry F. Franklin,[4] Peter Z. Fulé,[5] Mark E. Harmon,[6] Andrew J. Larson,[4] Jeremy M. Smith,[7] Alan H. Taylor,[8] Thomas T. Veblen[7]

Persistent changes in tree mortality rates can alter forest structure, composition, and ecosystem services such as carbon sequestration. Our analyses of longitudinal data from unmanaged old forests in the western United States showed that background (noncatastrophic) mortality rates have increased rapidly in recent decades, with doubling periods ranging from 17 to 29 years among regions. Increases were also pervasive across elevations, tree sizes, dominant genera, and past fire histories. Forest density and basal area declined slightly, which suggests that increasing mortality was not caused by endogenous increases in competition. Because mortality increased in small trees, the overall increase in mortality rates cannot be attributed solely to aging of large trees. Regional warming and consequent increases in water deficits are likely contributors to the increases in tree mortality rates.



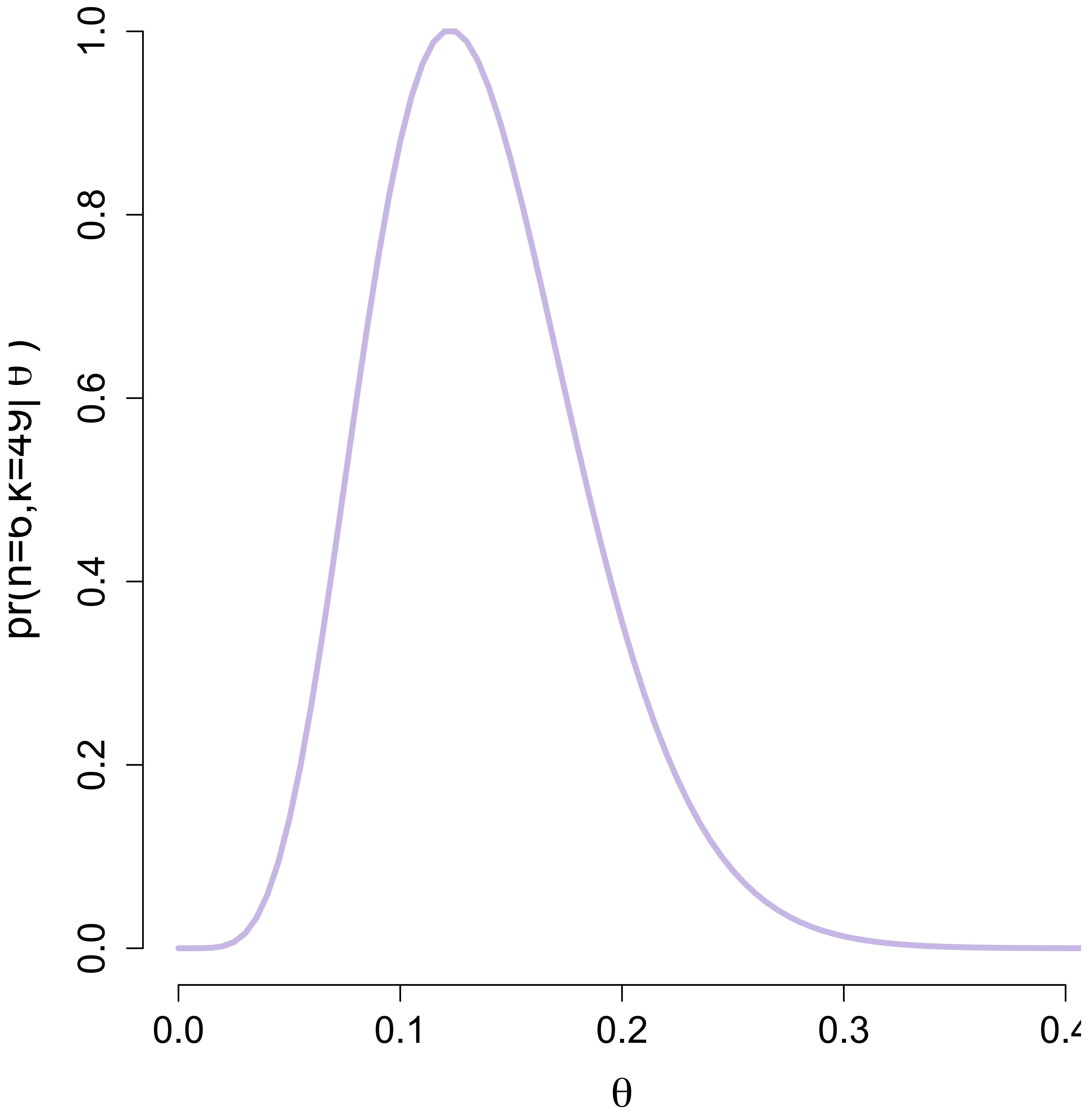
ADDING PRIOR INFORMATION: TREE MORTALITY

- Likelihood: dbinom(k = 6, n = 49)
- MLE: 6/49 = 0.122; prior mean: 0.087
- How to choose prior distribution?



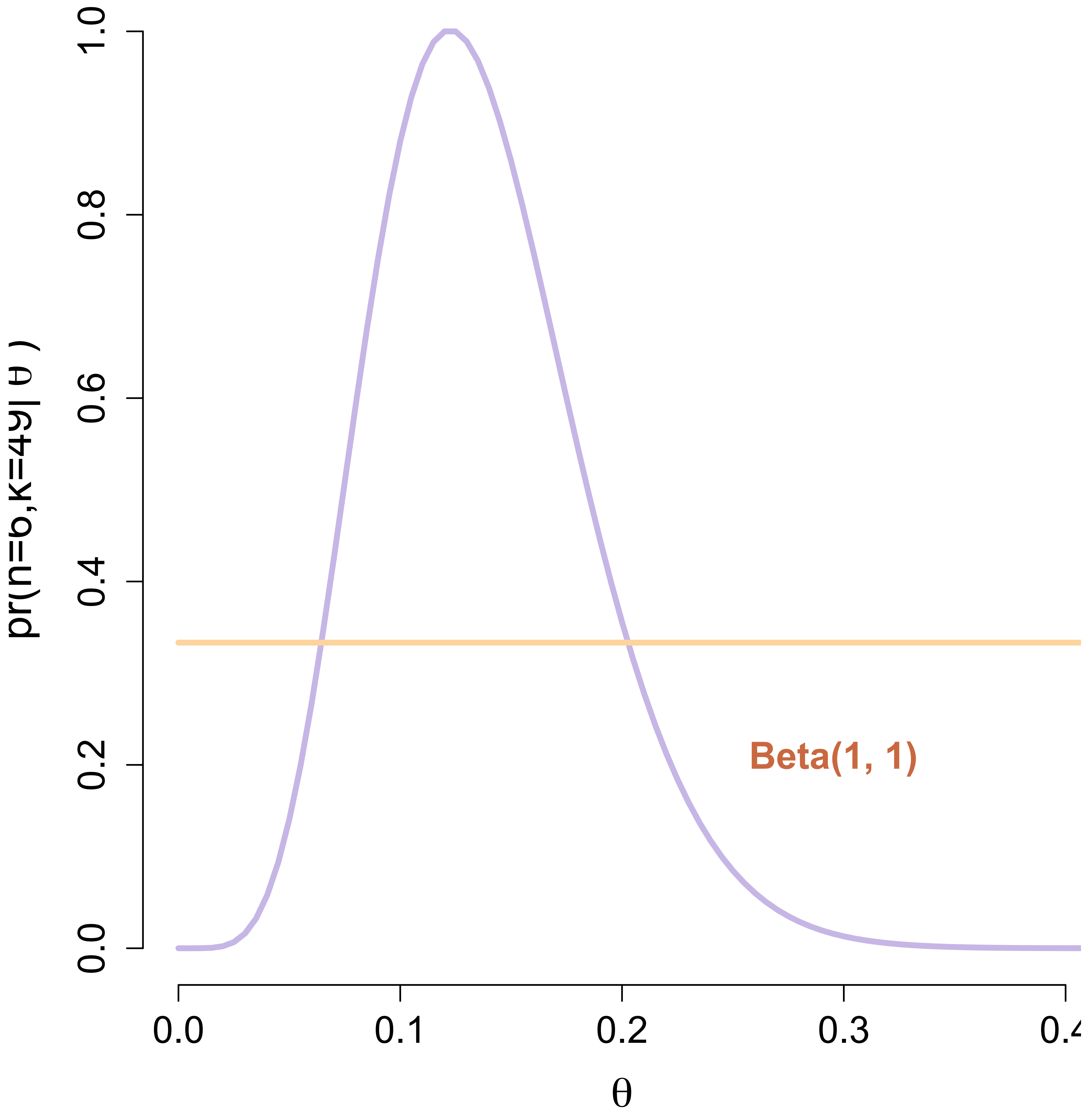
BETA PRIORS FOR BINOMIAL MODELS

- Beta parameters α and β can be interpreted as 1 + prior successes and failures
- Mean is just α/(α + β)
- How much do we trust the prior information?



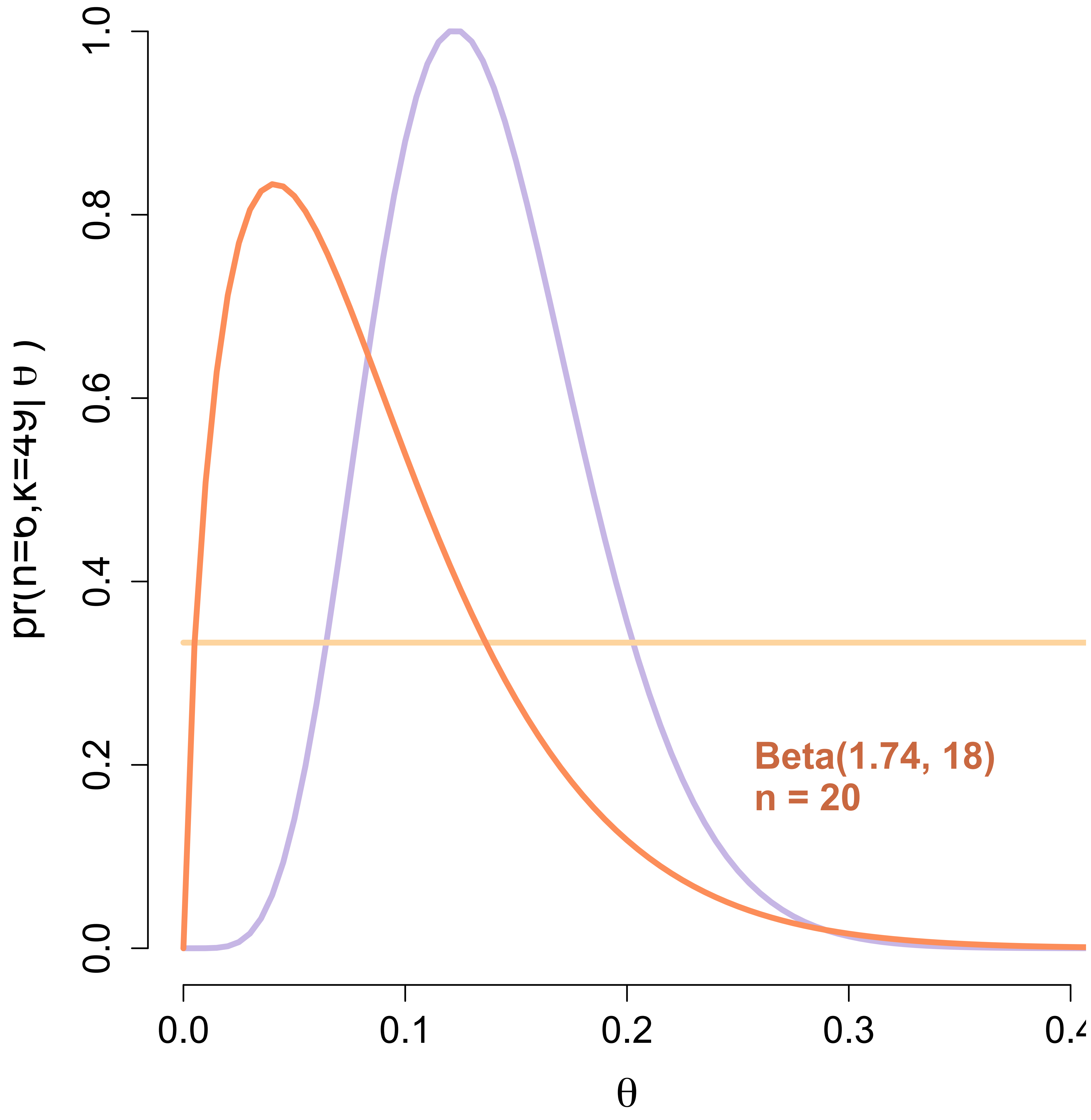
BETA PRIORS FOR BINOMIAL MODELS

- Beta parameters α and β can be interpreted as 1 + prior successes and failures
- Mean is just $\alpha/(\alpha + \beta)$
- How much do we trust the prior information?



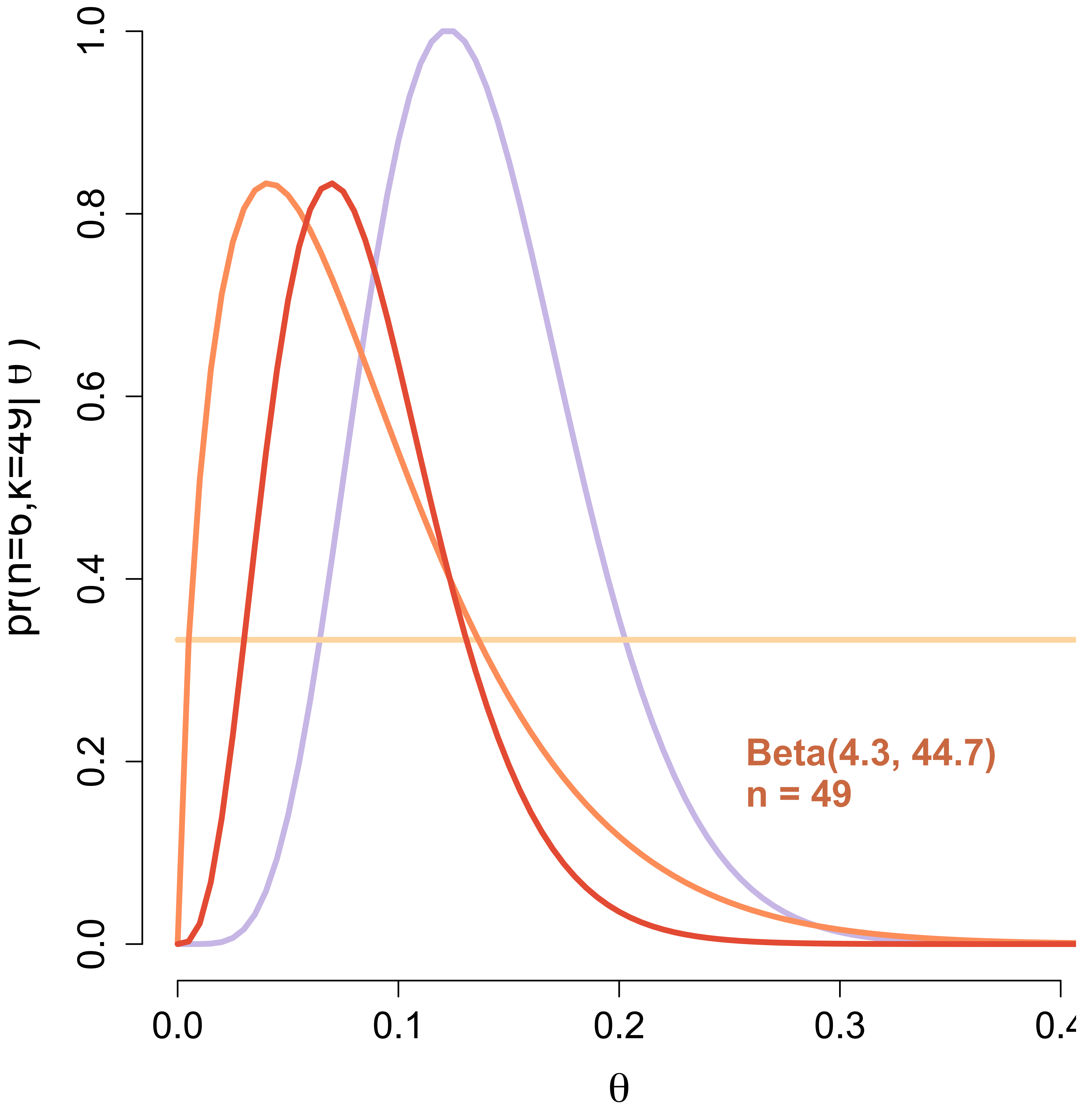
BETA PRIORS FOR BINOMIAL MODELS

- Beta parameters α and β can be interpreted as 1 + prior successes and failures
- Mean is just α/(α + β)
- How much do we trust the prior information?



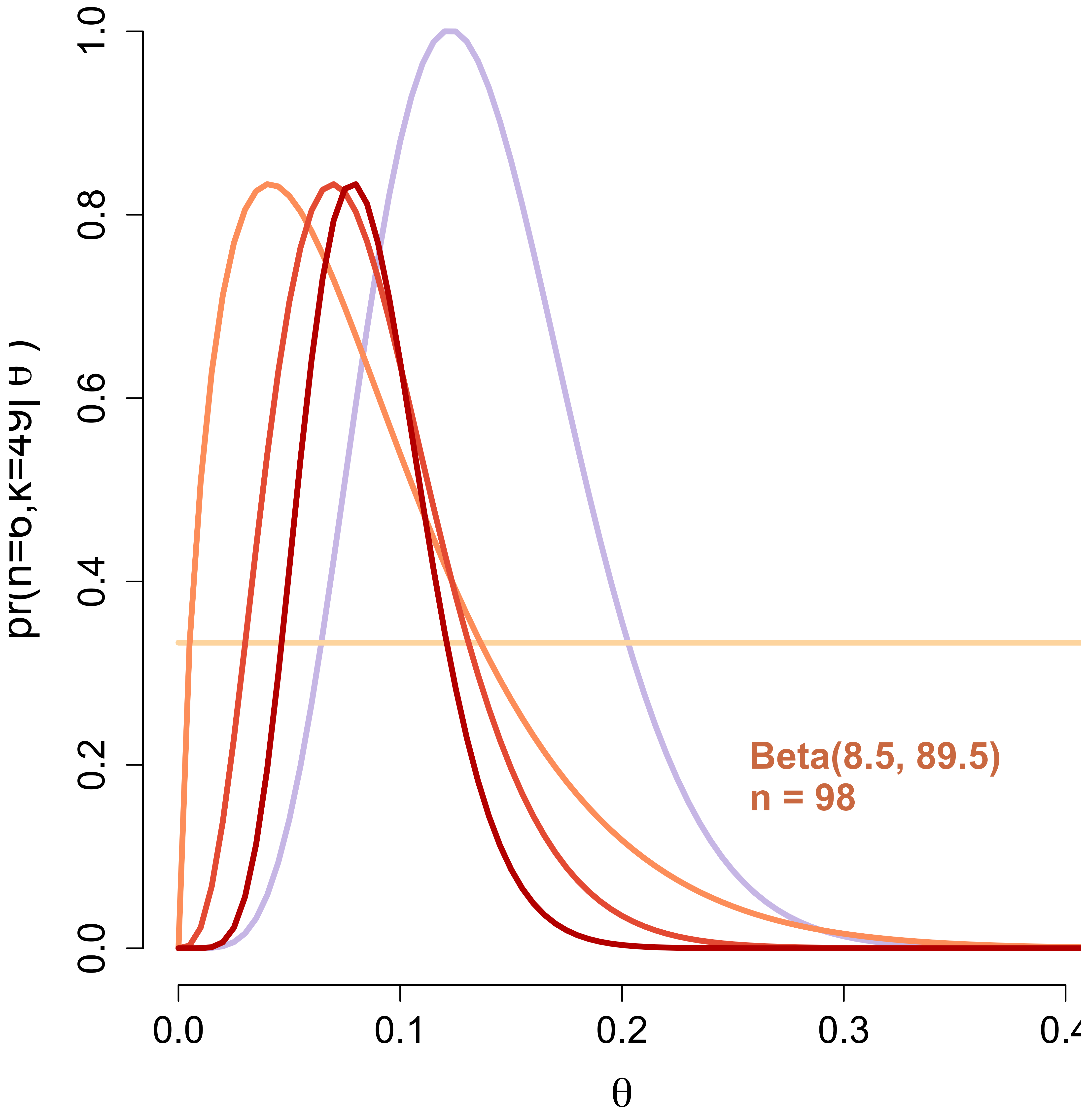
BETA PRIORS FOR BINOMIAL MODELS

- Beta parameters α and β can be interpreted as 1 + prior successes and failures
- Mean is just $\alpha/(\alpha + \beta)$
- How much do we trust the prior information?



BETA PRIORS FOR BINOMIAL MODELS

- Beta parameters α and β can be interpreted as 1 + prior successes and failures
- Mean is just $\alpha/(\alpha + \beta)$
- How much do we trust the prior information?



ADDING PRIOR INFORMATION: TREE MORTALITY

- We will use Beta(1.8, 18.89)

$$pr(\hat{\theta}|k = 6, n = 49) = \frac{pr(n=6,k=49|\hat{\theta})pr(\hat{\theta})}{\int_0^1 pr(n=6,k=49|\theta)pr(\theta)}$$

binomial

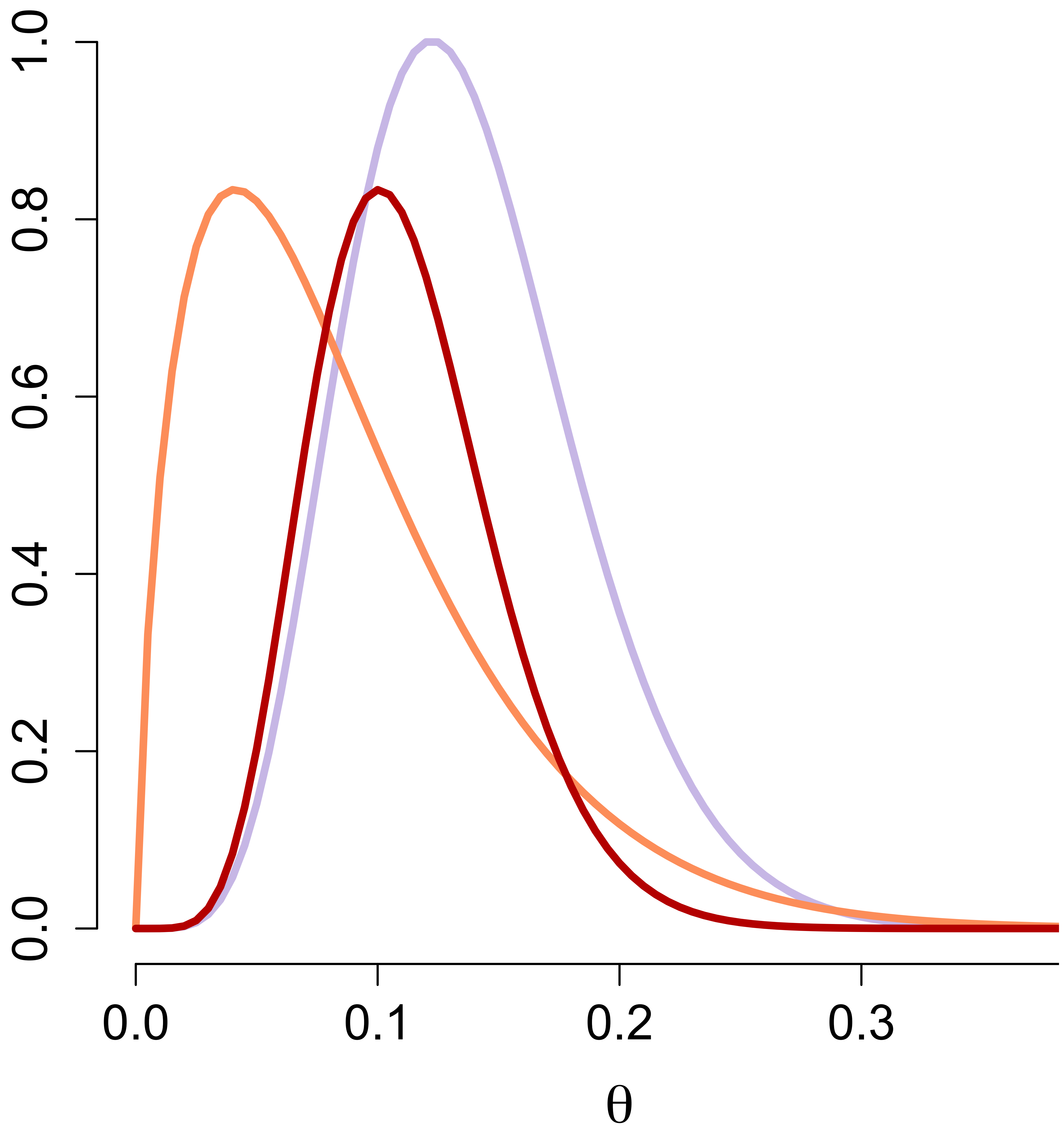
beta

$$= \frac{\underbrace{\binom{n}{k}\theta^k(1-\theta)^{n-k}}_{\text{binomial}}\underbrace{\frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{\mathrm{B}(\alpha,\beta)}}_{\text{beta}}}{\int_0^1 \binom{n}{k}\theta^k(1-\theta)^{n-k}\frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{\mathrm{B}(\alpha,\beta)}}$$

ADDING PRIOR INFORMATION: TREE MORTALITY

$$= \text{Beta}(\alpha + k, \beta + n - k)$$

- This relationship is called conjugacy
- A binomial likelihood with a beta prior produces a beta posterior
- Same relationship exists for poisson/gamma, normal/normal, and others
- Conjugacy allows us to solve Bayes' theorem analytically



NORMALISATION CONSTANT

- Normalisation ensures posterior integrates to one
- As a rule (with exceptions) we cannot compute this integral
- Because it is constant, we use a proportional form
- We have algorithms to sample from unknown distributions

$$pr(\theta|X) = \frac{pr(X|\theta)pr(\theta)}{\int pr(X|\theta)pr(\theta)d\theta}$$

NORMALISATION CONSTANT

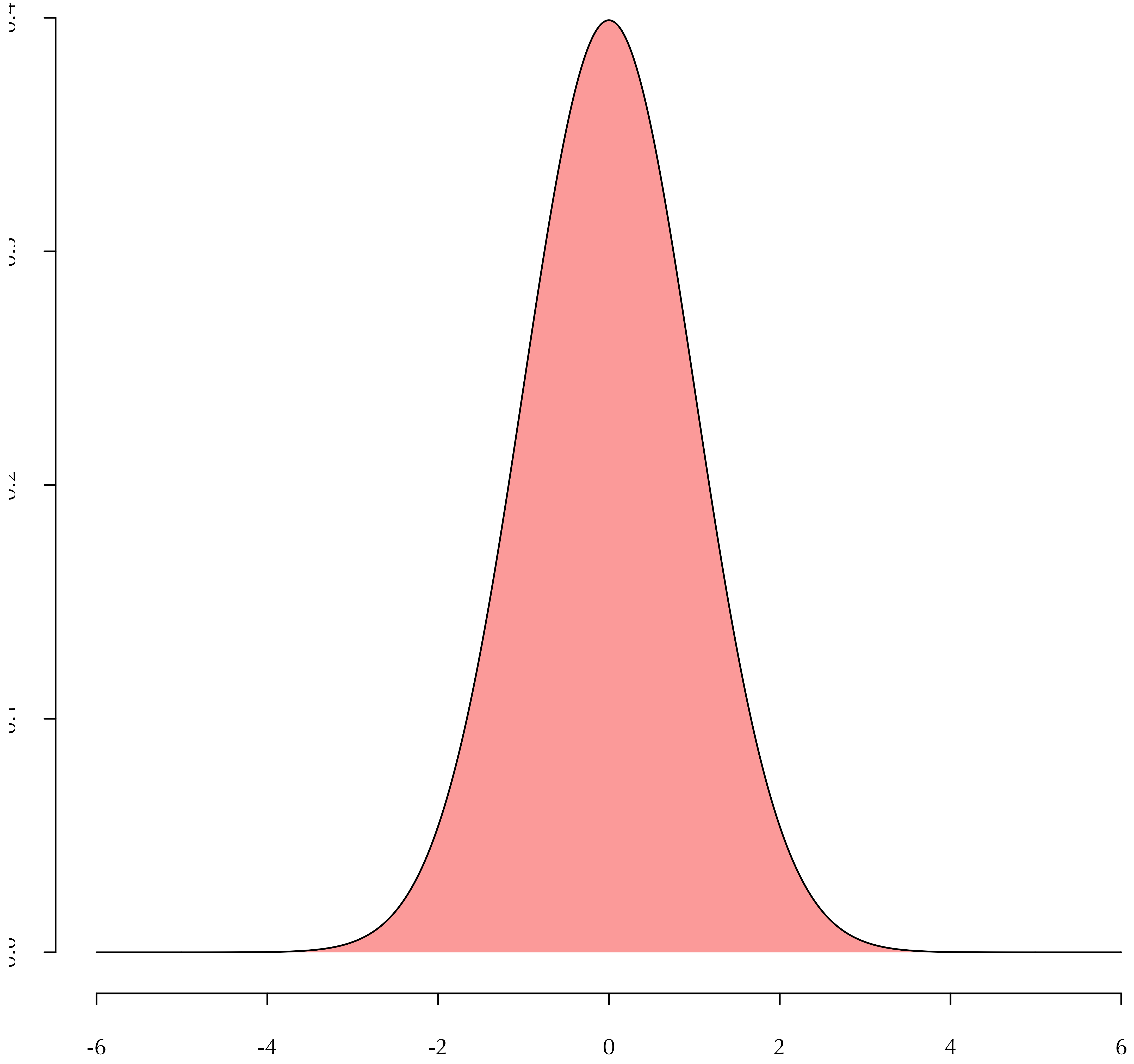
- Normalisation ensures posterior integrates to one
- As a rule (with exceptions) we cannot compute this integral
- Because it is constant, we use a proportional form
- We have algorithms to sample from unknown distributions

$$pr(\theta|X) = \frac{pr(X|\theta)pr(\theta)}{\int pr(X|\theta)pr(\theta)d\theta}$$

$$pr(\theta|X) \propto pr(X|\theta)pr\theta$$

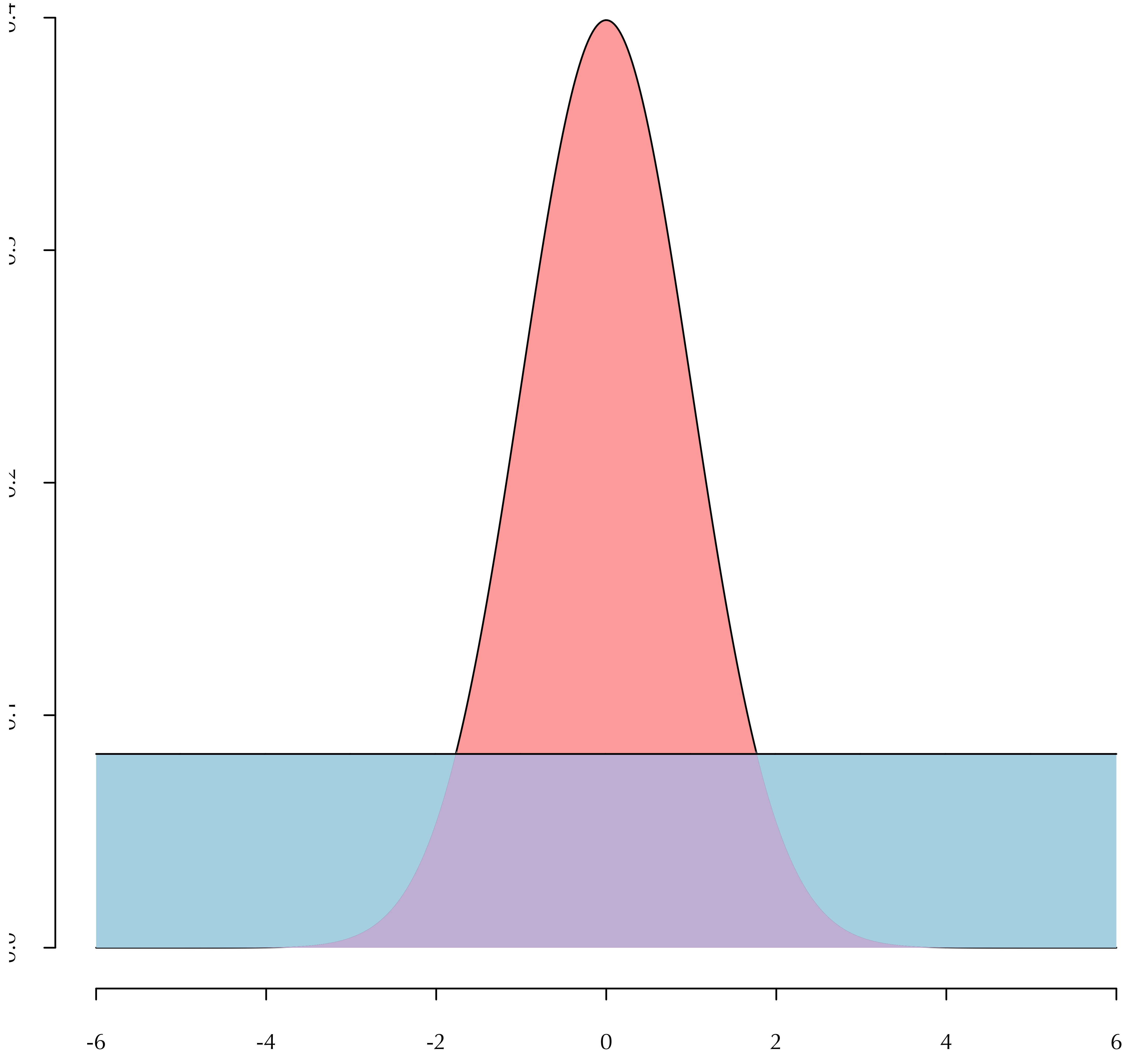
REJECTION SAMPLING

- Use an easily-sampled candidate distribution c(x) to sample from the difficult-to-sample target distribution t(x)
- Reject samples with probability proportional to difference between the distributions



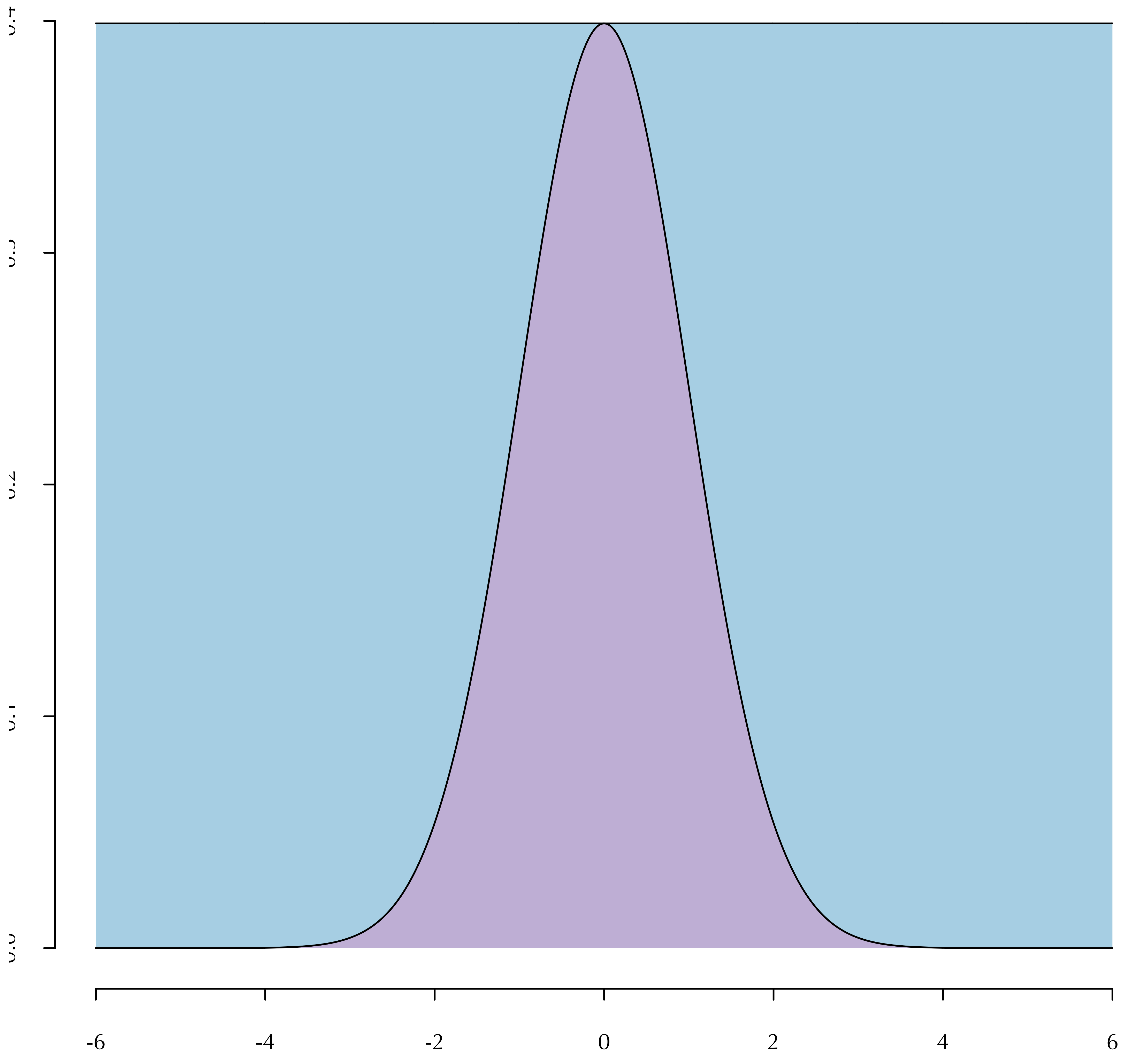
REJECTION SAMPLING

- Use an easily-sampled candidate distribution c(x) to sample from the difficult-to-sample target distribution t(x)
- Reject samples with probability proportional to difference between the distributions



REJECTION SAMPLING

- Use an easily-sampled candidate distribution c(x) to sample from the difficult-to-sample target distribution t(x)
- Reject samples with probability proportional to difference between the distributions



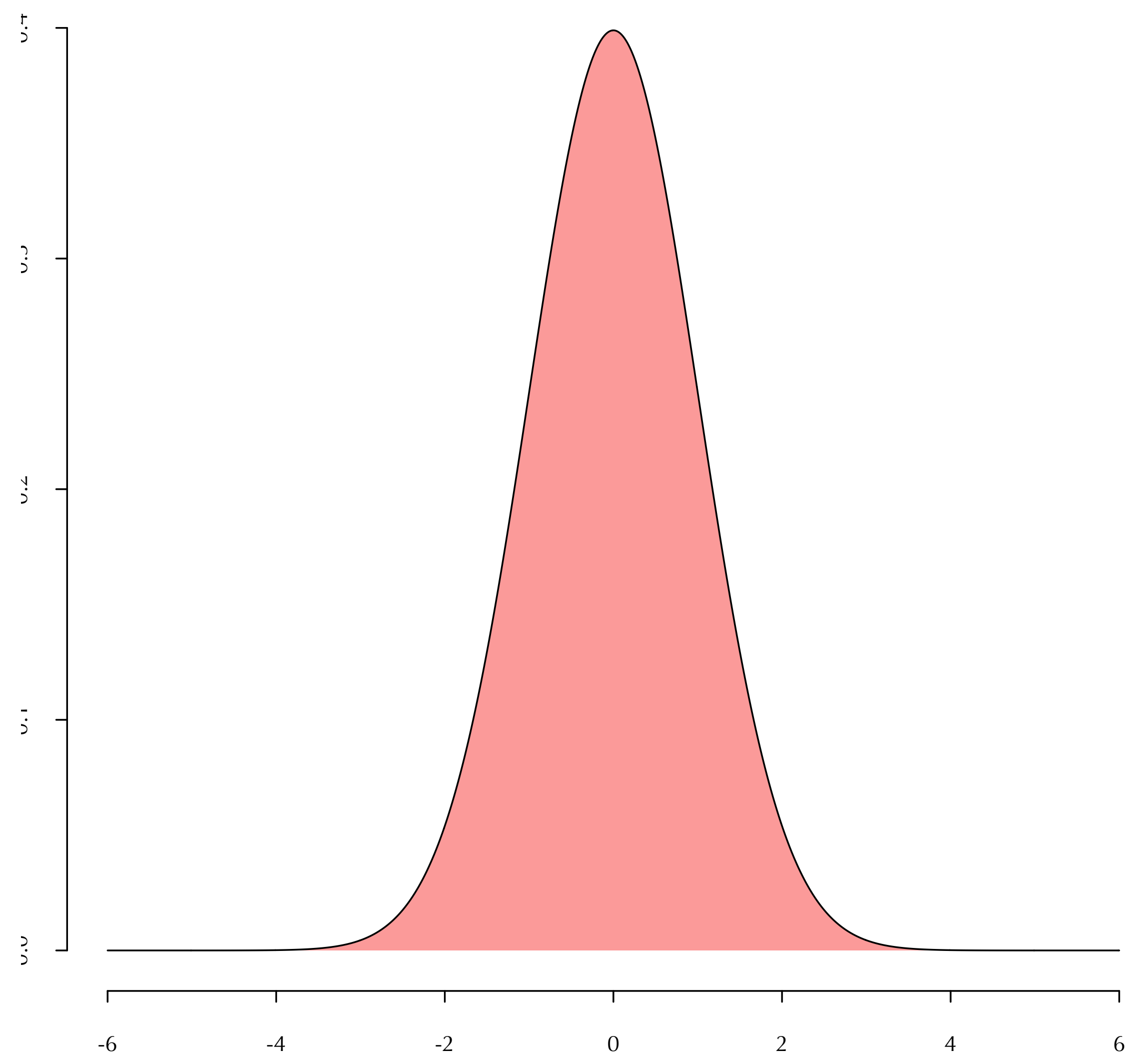
REJECTION SAMPLING PRACTISE

Algorithm

REJECTION SAMPLING PRACTISE

Algorithm

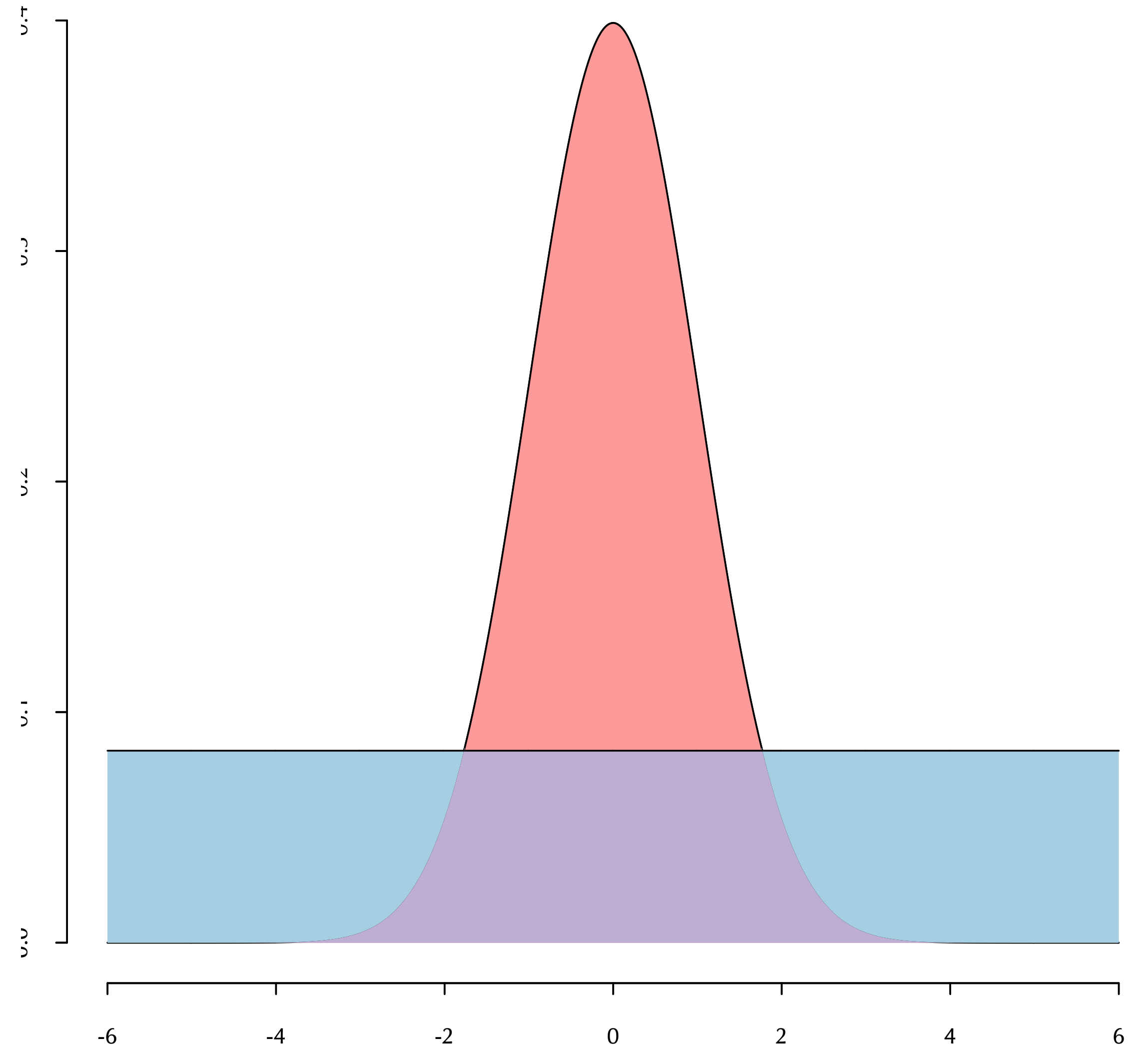
```
Define target distribution t(x)
```



REJECTION SAMPLING PRACTISE

Algorithm

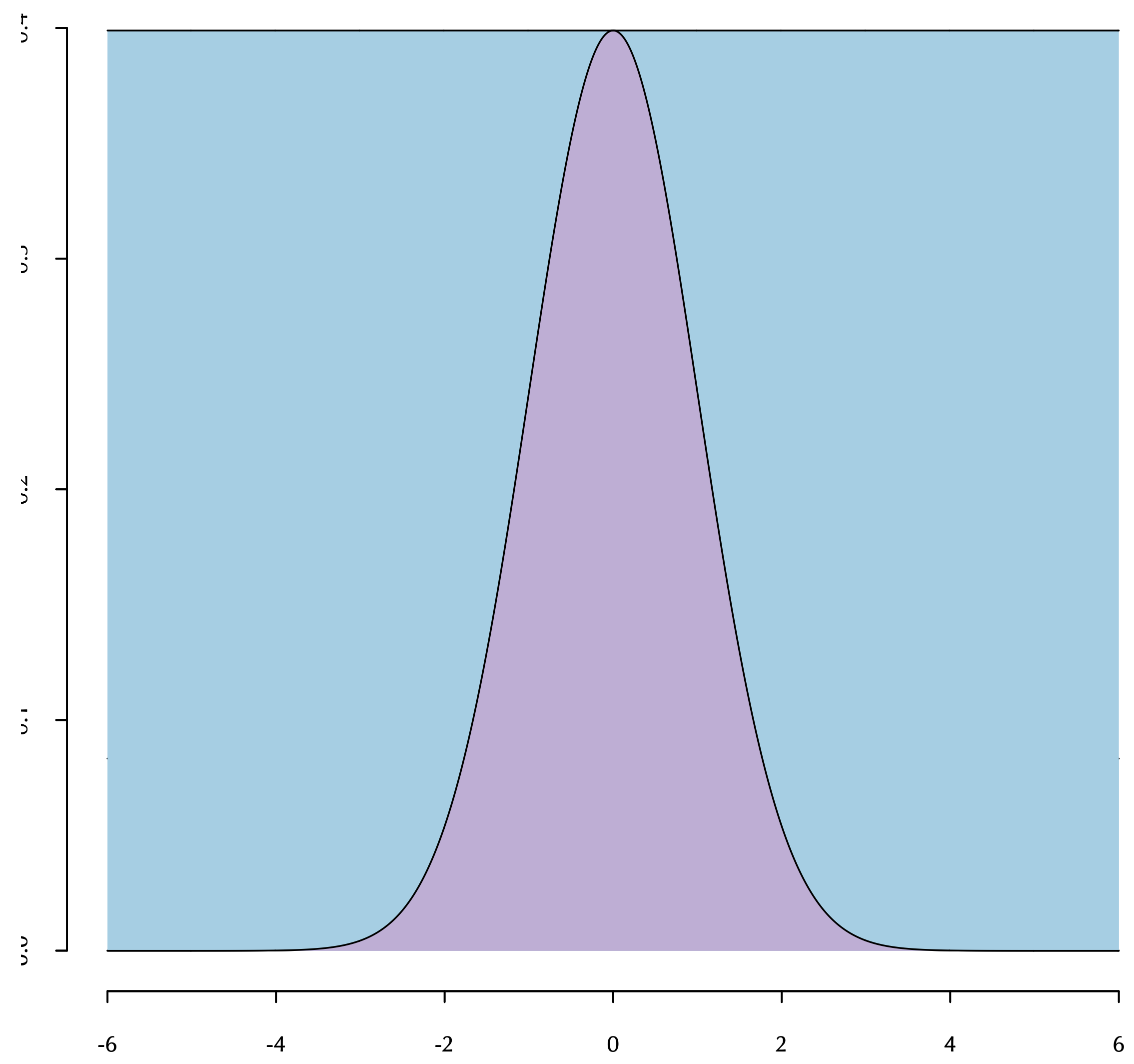
```
Define target distribution t(x)
Define sample distribution c(x)
```



REJECTION SAMPLING PRACTISE

Algorithm

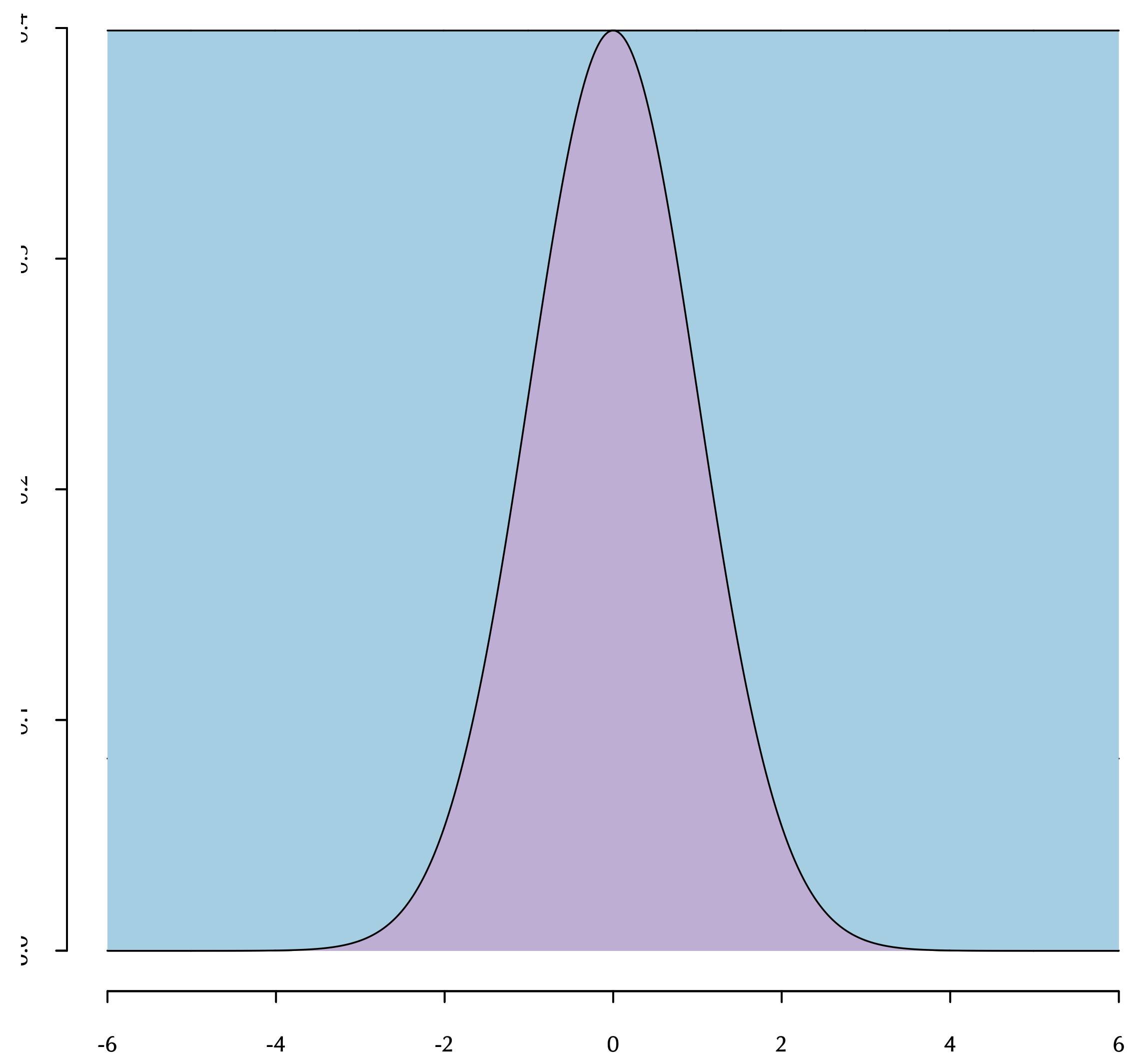
```
Define target distribution t(x)
Define sample distribution c(x)
Define constant M so that Mc(x) ≥ t(x) for all x
```



REJECTION SAMPLING PRACTISE

Algorithm

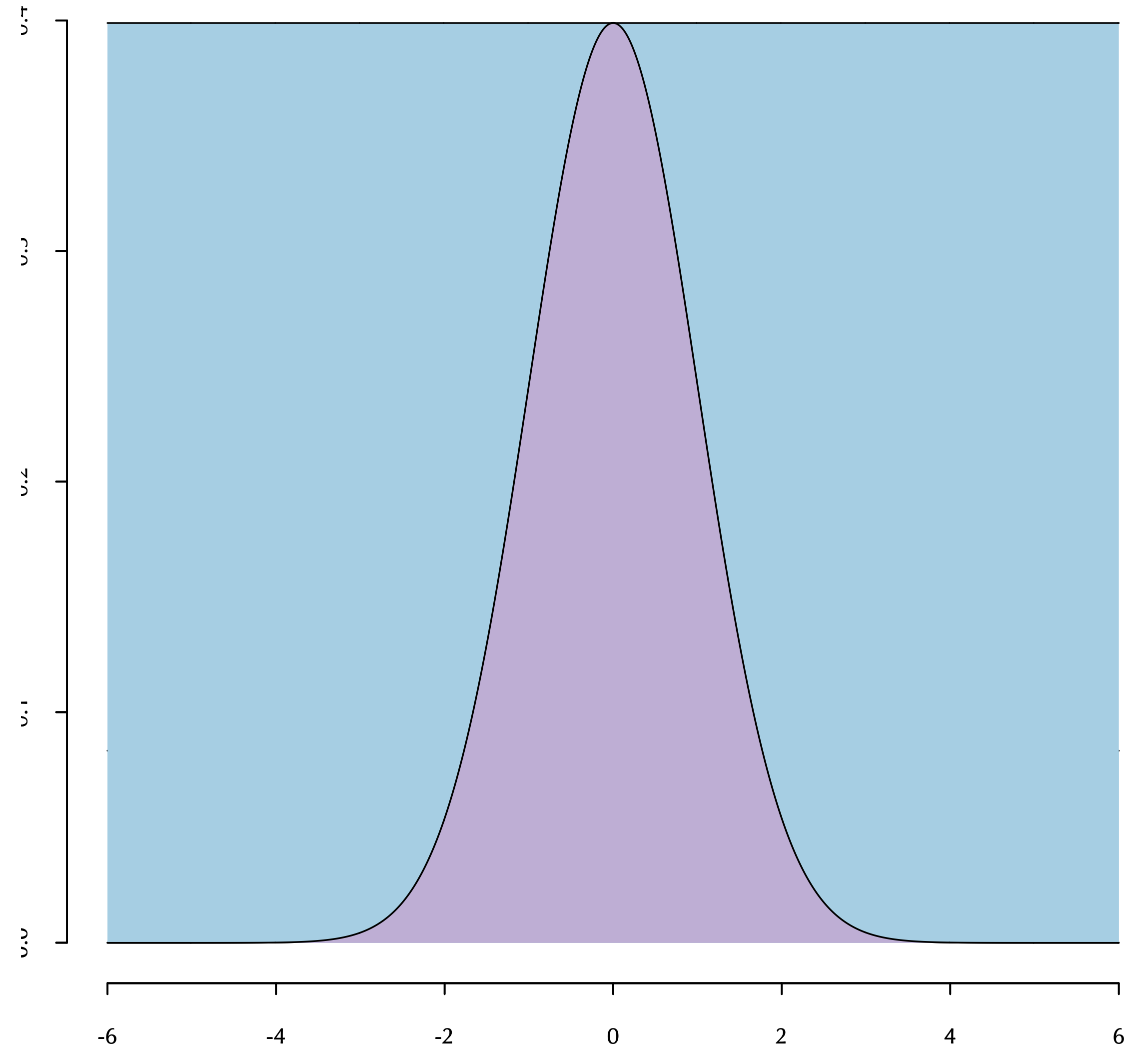
```
Define target distribution t(x)
Define sample distribution c(x)
Define constant M so that Mc(x) ≥ t(x) for all x
REPEAT UNTIL X is accepted
```



REJECTION SAMPLING PRACTISE

Algorithm

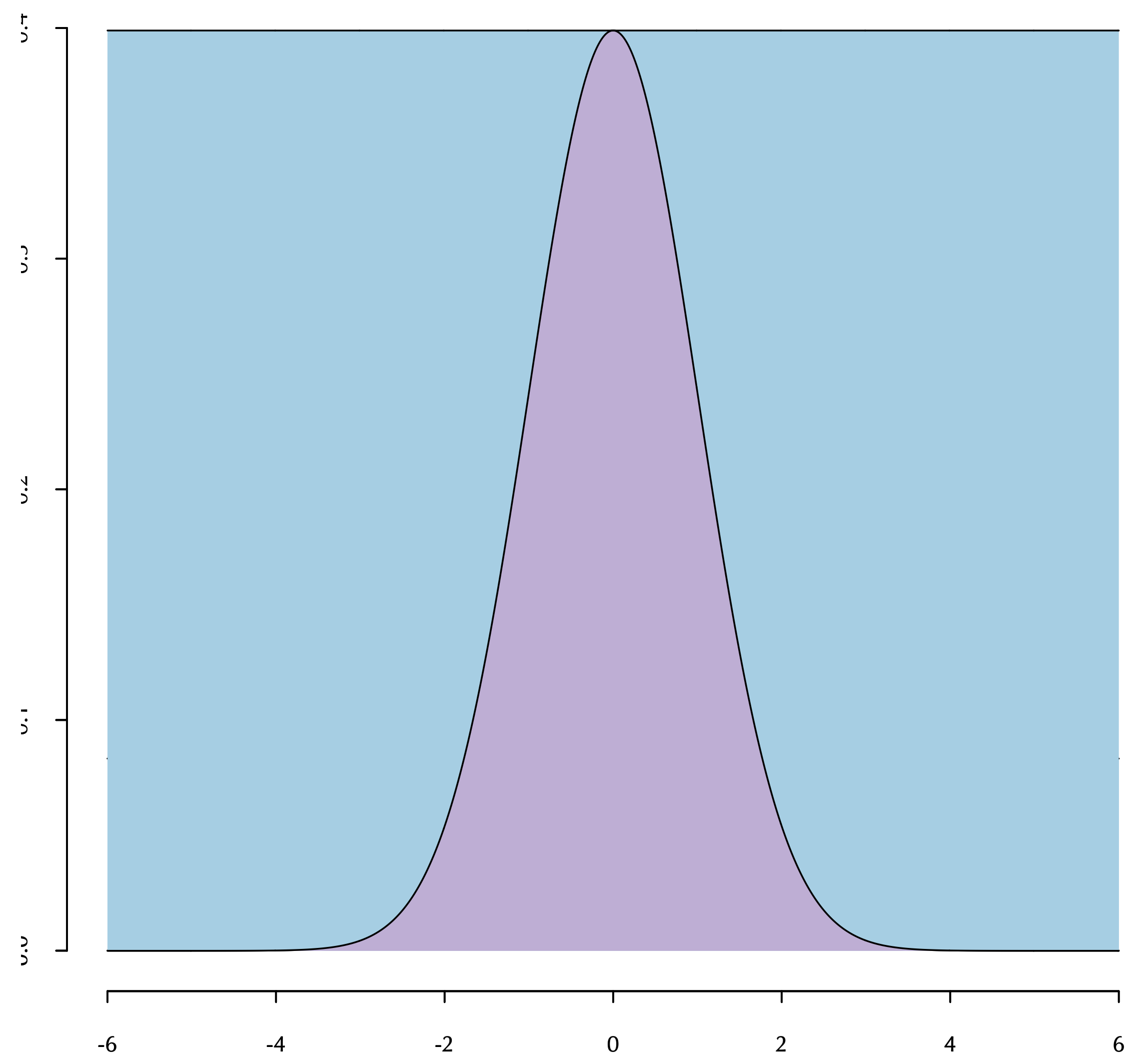
```
Define target distribution t(x)
Define sample distribution c(x)
Define constant M so that Mc(x) ≥ t(x) for all x
REPEAT UNTIL X is accepted
  Draw sample X from c(x)
```



REJECTION SAMPLING PRACTISE

Algorithm

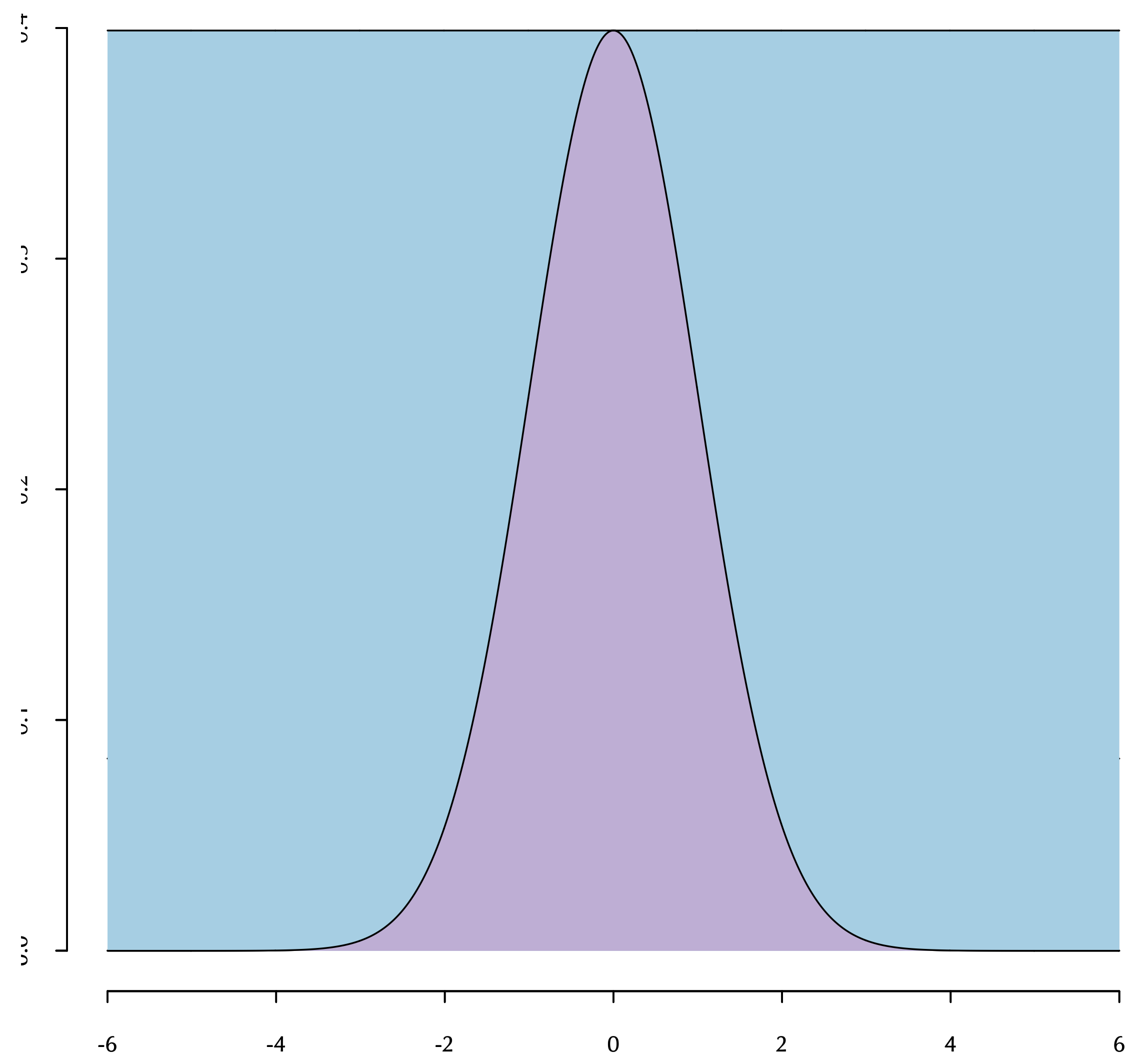
```
Define target distribution t(x)
Define sample distribution c(x)
Define constant M so that Mc(x) ≥ t(x) for all x
REPEAT UNTIL X is accepted
  Draw sample X from c(x)
  Calculate acceptance probability p = t(X) / (Mc(X))
```



REJECTION SAMPLING PRACTISE

Algorithm

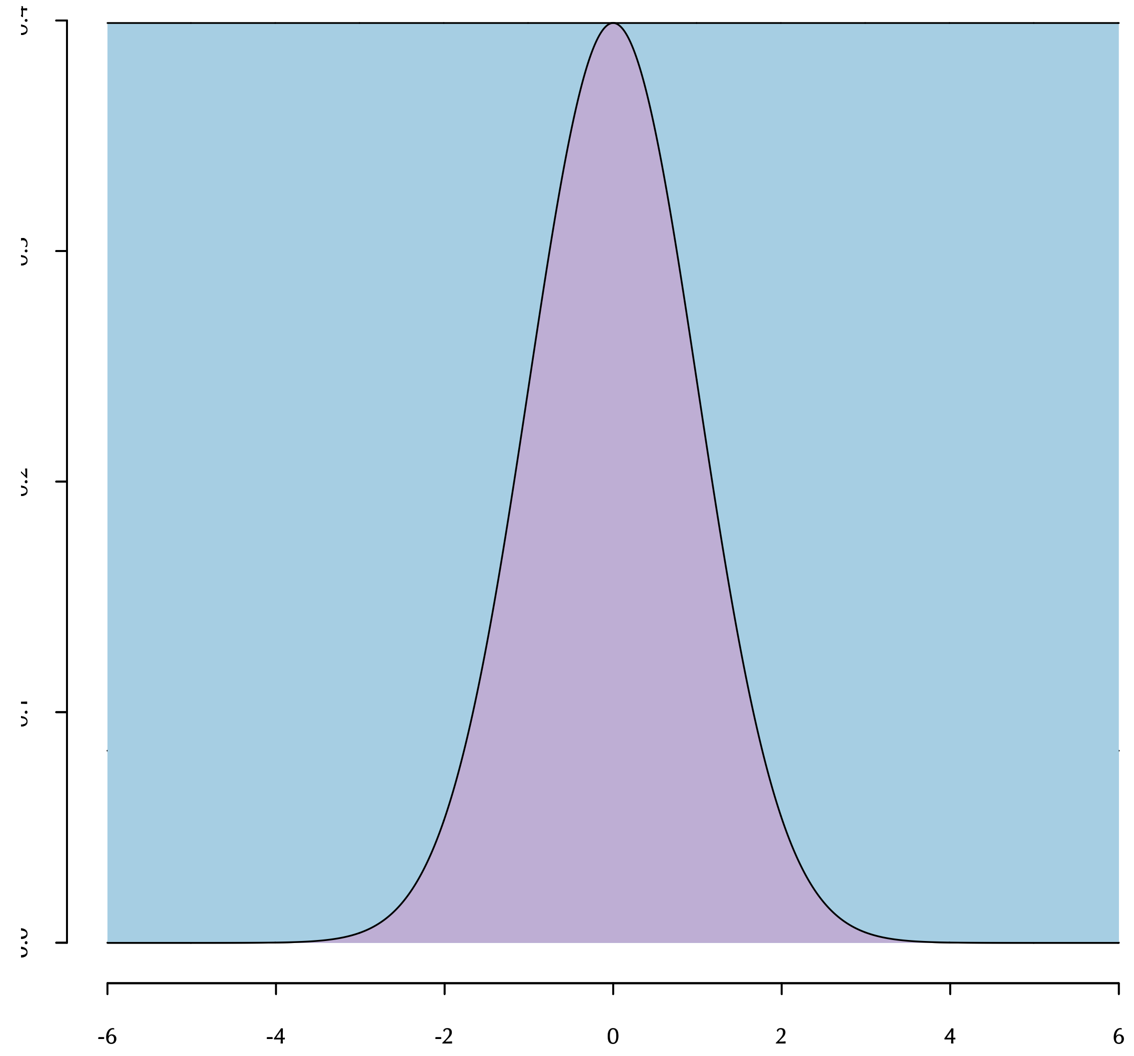
```
Define target distribution t(x)
Define sample distribution c(x)
Define constant M so that Mc(x) ≥ t(x) for all x
REPEAT UNTIL X is accepted
  Draw sample X from c(x)
  Calculate acceptance probability p = t(X) / (Mc(X))
  Draw test value P from random uniform on (0,1)
```



REJECTION SAMPLING PRACTISE

Algorithm

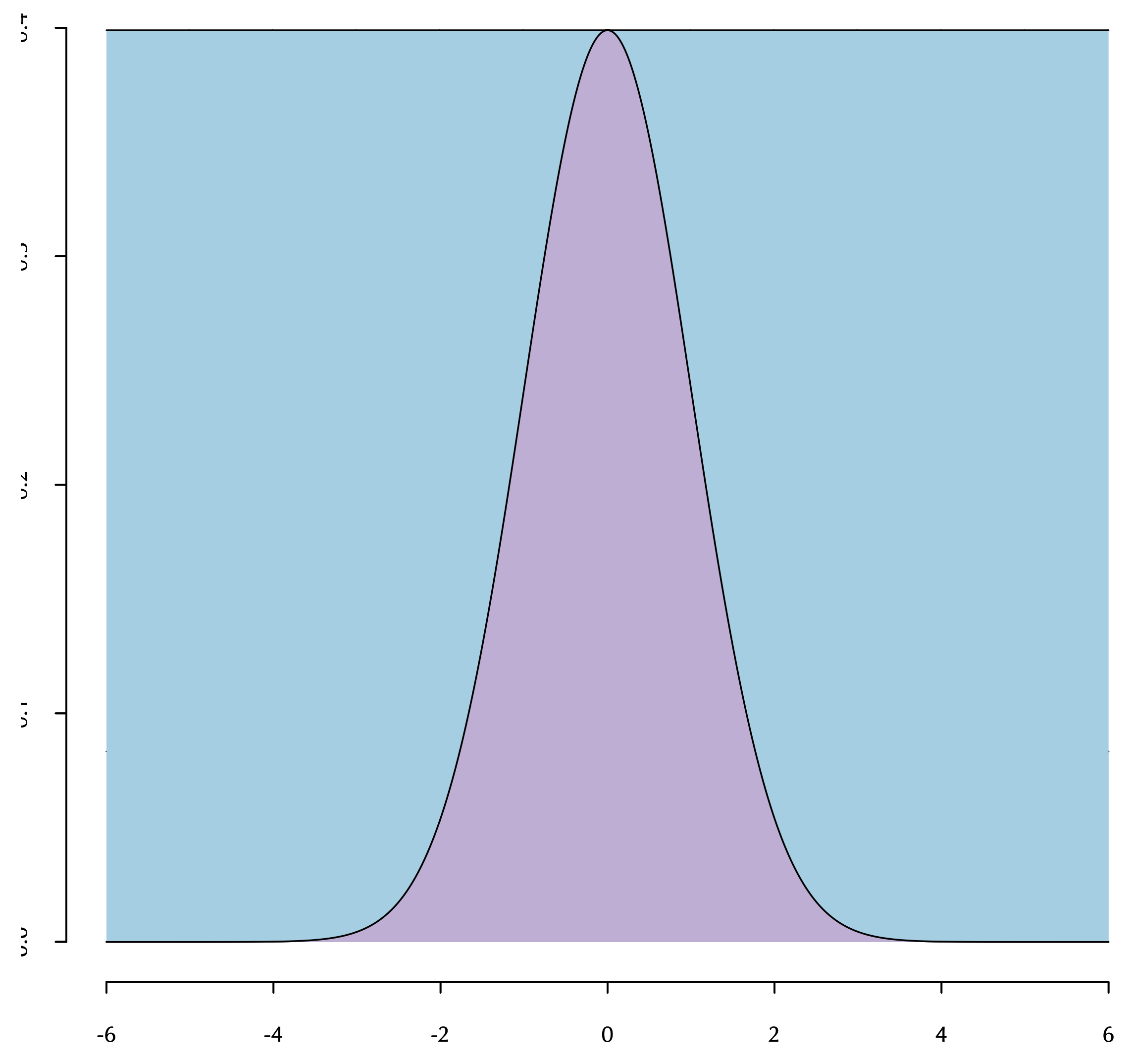
```
Define target distribution t(x)
Define sample distribution c(x)
Define constant M so that Mc(x) ≥ t(x) for all x
REPEAT UNTIL X is accepted
  Draw sample X from c(x)
  Calculate acceptance probability p = t(X) / (Mc(X))
  Draw test value P from random uniform on (0,1)
  IF P < p
```



REJECTION SAMPLING PRACTISE

Algorithm

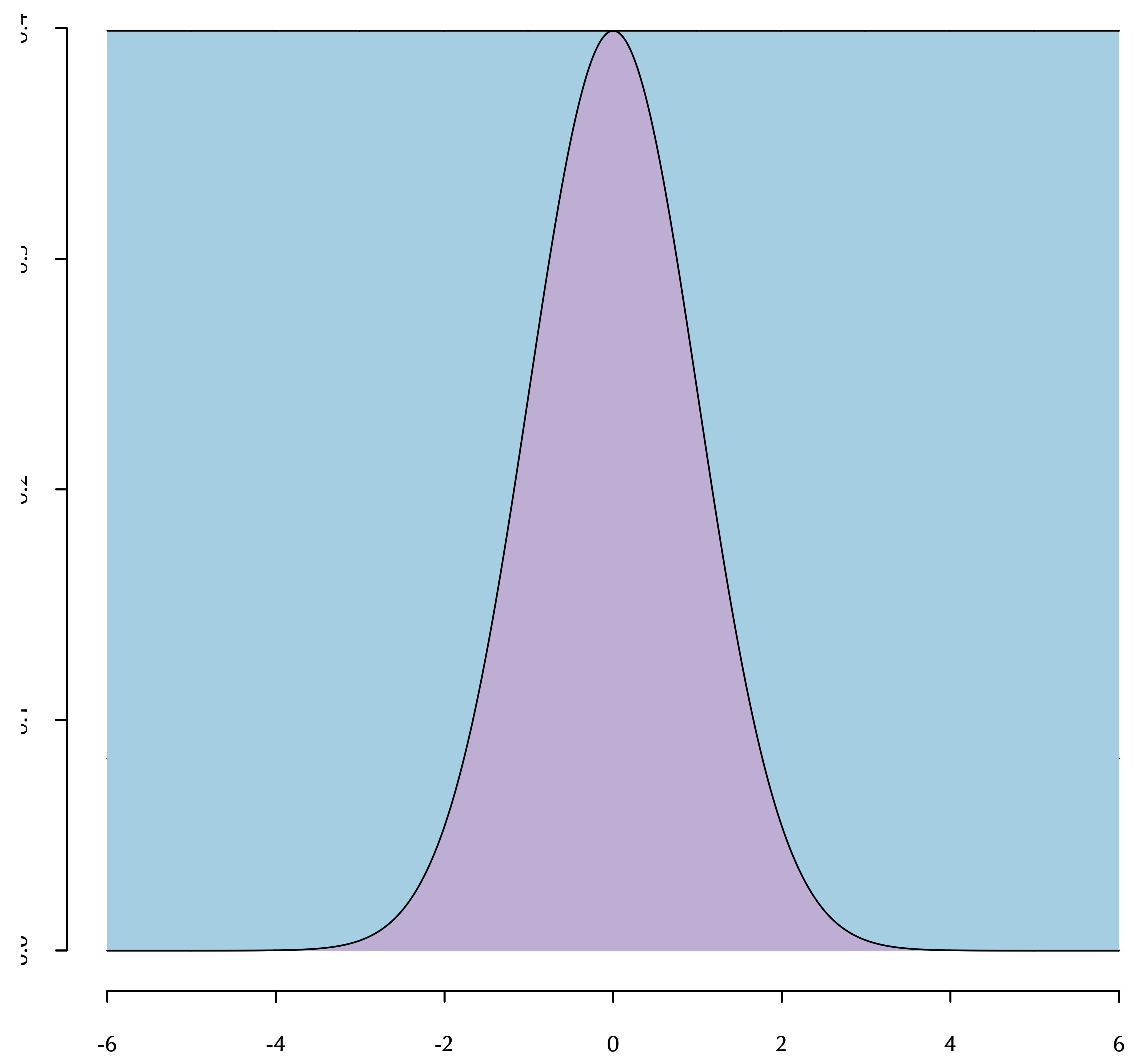
```
Define target distribution t(x)
Define sample distribution c(x)
Define constant M so that Mc(x) ≥ t(x) for all x
REPEAT UNTIL X is accepted
  Draw sample X from c(x)
  Calculate acceptance probability p = t(X) / (Mc(X))
  Draw test value P from random uniform on (0,1)
  IF P < p
    accept X
```



REJECTION SAMPLING PRACTISE

Algorithm

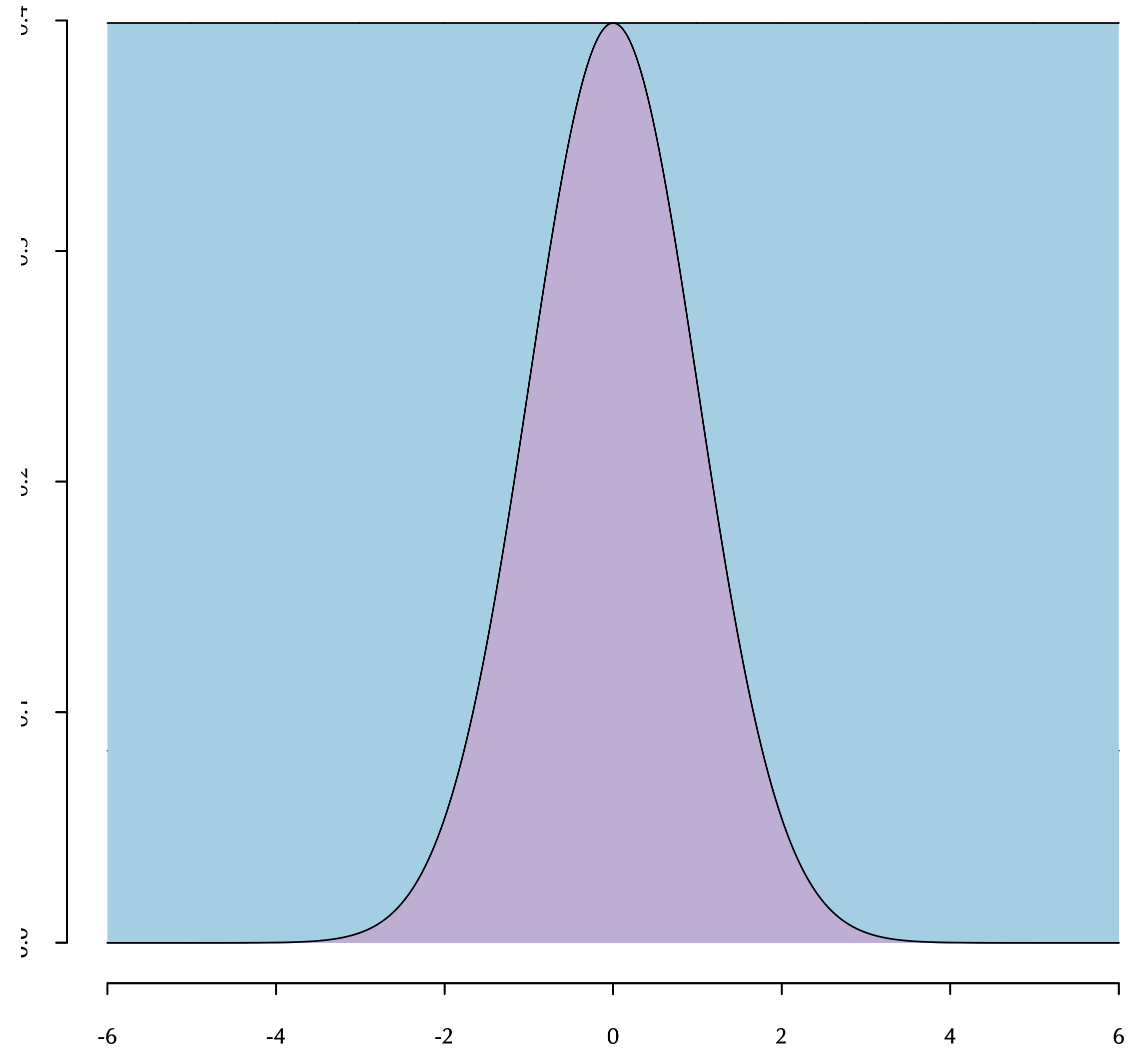
```
Define target distribution t(x)
Define sample distribution c(x)
Define constant M so that Mc(x) ≥ t(x) for all x
REPEAT UNTIL X is accepted
  Draw sample X from c(x)
  Calculate acceptance probability p = t(X) / (Mc(X))
  Draw test value P from random uniform on (0,1)
  IF P < p
    accept X
  ELSE
```



REJECTION SAMPLING PRACTISE

Algorithm

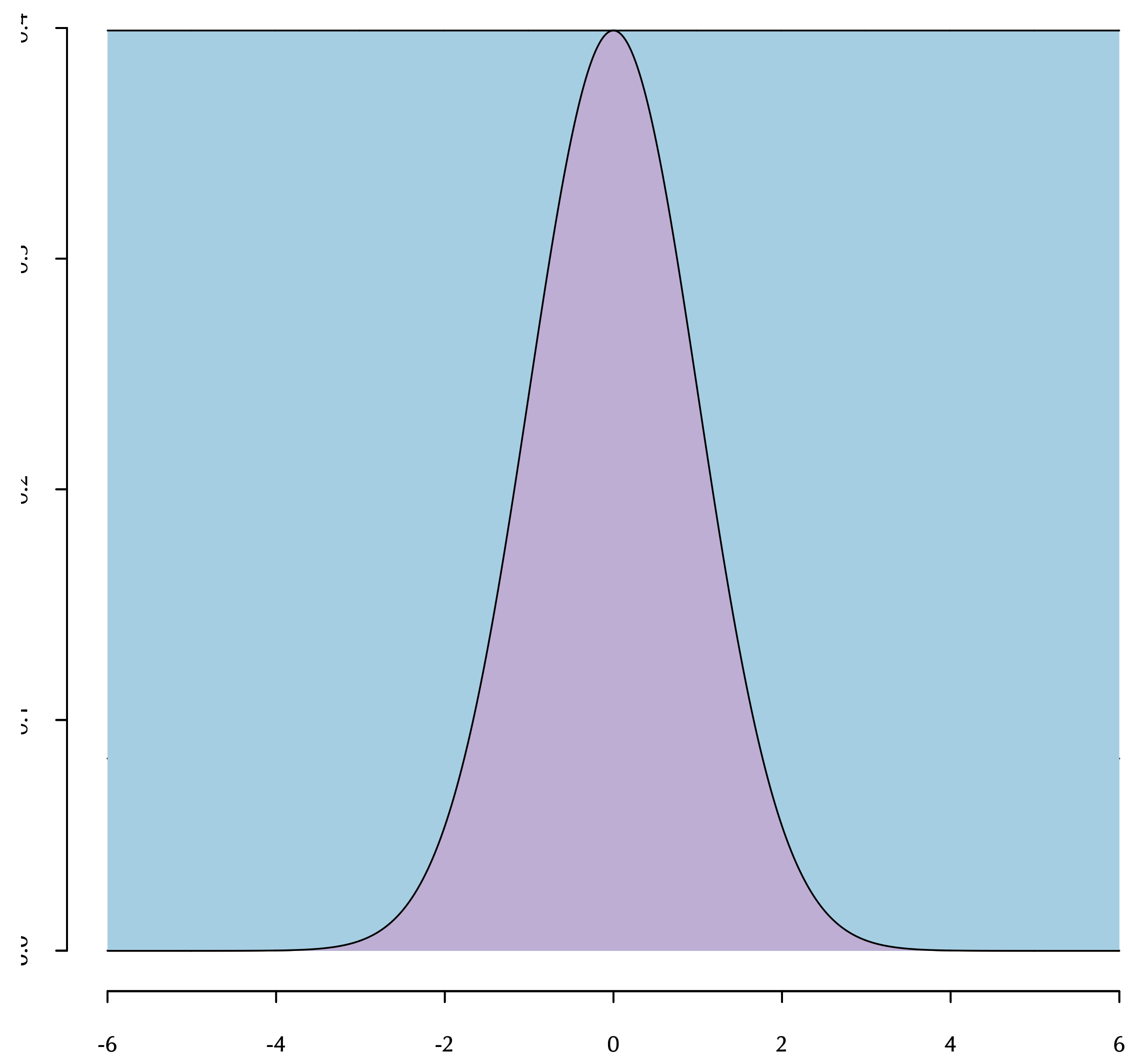
```
Define target distribution t(x)
Define sample distribution c(x)
Define constant M so that Mc(x) ≥ t(x) for all x
REPEAT UNTIL X is accepted
  Draw sample X from c(x)
  Calculate acceptance probability p = t(X) / (Mc(X))
  Draw test value P from random uniform on (0,1)
  IF P < p
    accept X
  ELSE
    reject X
```



REJECTION SAMPLING PRACTISE

Algorithm

```
Define target distribution t(x)
Define sample distribution c(x)
Define constant M so that Mc(x) ≥ t(x) for all x
REPEAT UNTIL X is accepted
  Draw sample X from c(x)
  Calculate acceptance probability p = t(X) / (Mc(X))
  Draw test value P from random uniform on (0,1)
  IF P < p
    accept X
  ELSE
    reject X
```



- Implement the algorithm to sample from a normal distribution with $\mu=4$ and $\sigma=2$
- Limit the candidate to Uniform(-100, 100)
- Take 1000 samples and evaluate; how does speed compare to rnorm?

MARKOV CHAIN MONTE CARLO (MCMC)

- Rejection sampling is inefficient, even though samples are independent, often waste a lot of time throwing away work
- Especially true for multivariate problems
- Instead we use Markov chains
- The state of the chain (i.e., the parameter value) at time t+1 depends on the state at time t and some probability of transitioning to a new state

MARKOV CHAIN MONTE CARLO (MCMC)

- When chains run a long time, they converge to a stationary distribution; this is the posterior distribution of the parameter
- Biggest assumption: positive recurrence. You must be able to reach any state from any other state in a "reasonable" amount of time

METROPOLIS-HASTINGS

- Simplest version of MCMC, especially for univariate problems
- Similar to rejection sampling, but each iteration generates a sample from the posterior
- Samples are autocorrelated, so effective sample size much smaller than the number of iterations

METROPOLIS HASTINGS ALGORITHM

Algorithm

```
define target distribution f(x)
define candidate distribution j(x|y)
define acceptance probability:
  r(x,y) = ( f(x) * j(y|x) ) / ( f(y) * j(x|y) )
define Markov chain array X of length N
choose starting value X[0]

FOR t in 1 to N
  sample x_cand from j(x|X[t-1])
  set r_cand = r(x_cand, X[t-1])
  sample U from uniform(0,1)
  IF U < r
    X[t] = x_cand
  ELSE
    X[t] = X[t-1]
```

METROPOLIS-HASTINGS

$$r = \frac{f(X_{cand})j(X_{t-1} \mid X_{cand})}{f(X_{t-1})j(X_{cand} \mid X_{t-1})}$$

- This is the acceptance probability
- Note that if f(x) is a posterior distribution, the normalisation constant cancels!

METROPOLIS-HASTINGS

- What to use for proposal distribution j(x|y)?
- Very flexible, but must allow for positive recurrence of the chain
- A good starting point is the normal distribution with mean equal to the previous value and some standard deviation
- This distribution is symmetric; j(x|y) = j(y|x), thus:

$$r = \frac{f(X_{cand})}{f(X_{t-1})}$$

METROPOLIS-HASTINGS

- The MH algorithm needs tuning; the standard deviation of the proposal distribution $j(x|y)$ will need to be adjusted
- The ideal acceptance rate converges to 0.234 as the number of parameters grows large; between 0.2 and 0.6 is probably ok
- A simple way to implement tuning…

METROPOLIS AUTOMATIC TUNING

Algorithm

```
set sigma = 1.0
for i in 1 to N #large, approx 2000-5000
  run one MH iteration
  IF x_cand is accepted
    sigma = sigma * 1.01
  ELSE
    sigma = sigma / 1.01
discard first N MCMC results as tuning values
fix sigma at the last value from above
if acceptance rate is acceptable,
    proceed with MH algorithm as before
```

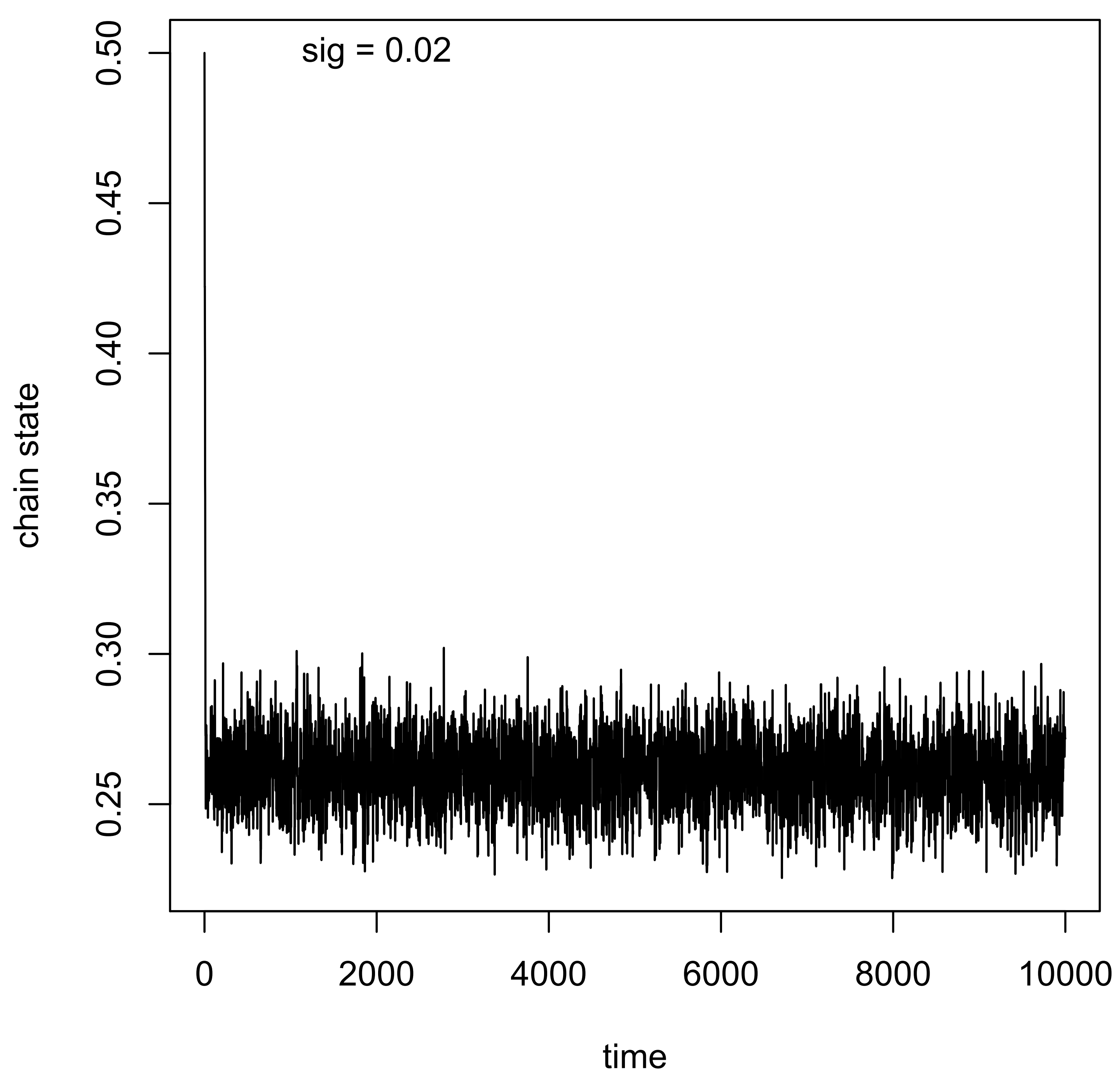
ADAPTIVE METROPOLIS PRACTISE

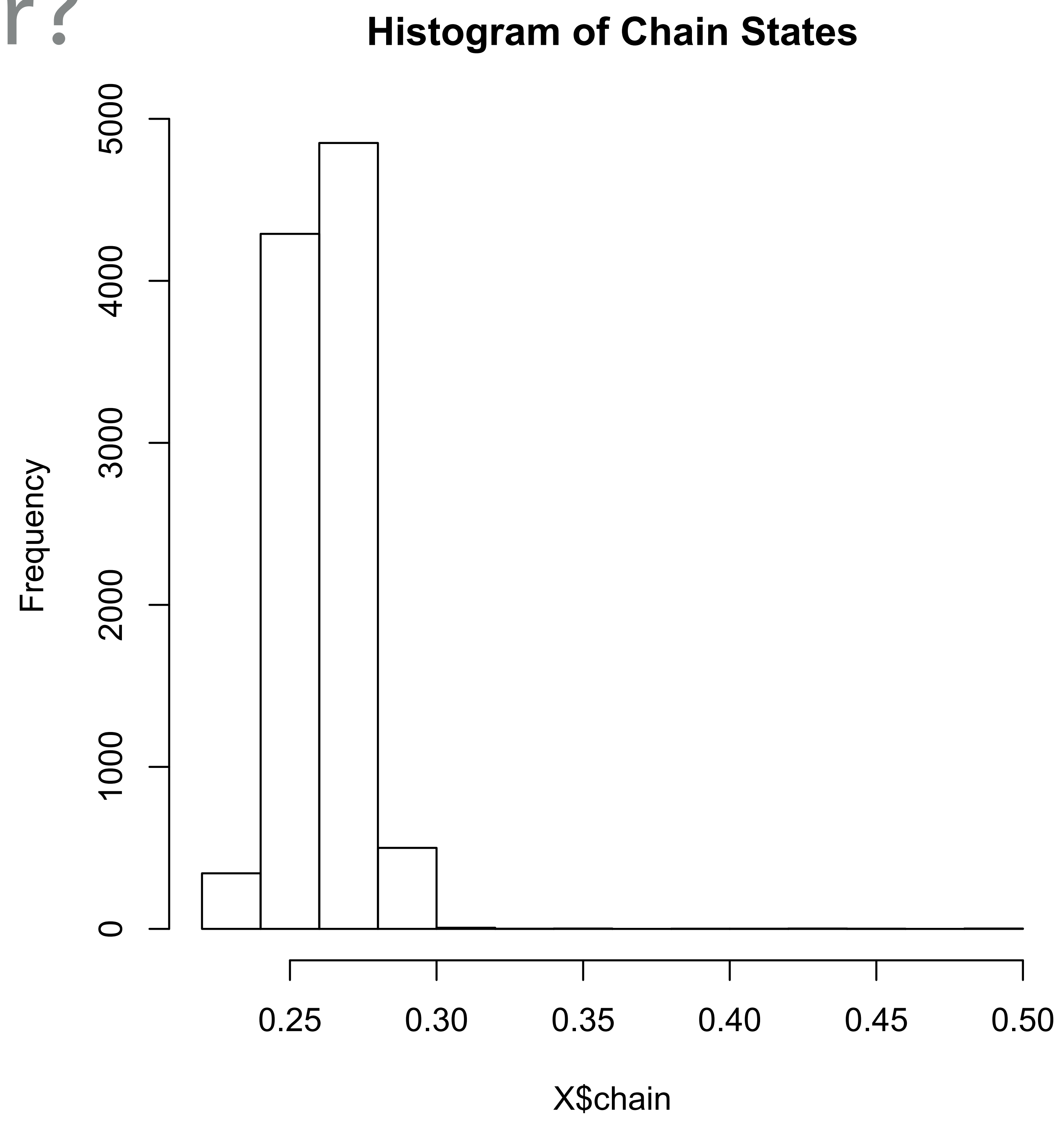
```r
library(data.table)
trees <- readRDS("data/trees.rds")
dat <- trees[grepl("TSU-CAN", species) & year == 2005 & n > 0]

lik_func <- function(theta, n, k) {

}

```

- Come back to the trees
- Use your likelihood function from earlier
- Incorporate a weak prior and return the log (proportional) posterior probability
- Implement the MH algorithm with tuning to sample the posterior for mortality rate
- What happens if you use the informative prior that comes with the data?

ADAPTIVE METROPOLIS PRACTISE

- Implement the MH algorithm with tuning to sample the posterior for mortality rate
- What is the effect of changing the prior?





MULTIVARIATE PROBLEMS

- If we have two parameters, we now want to estimate the joint posterior
  $pr(\theta_1,\theta_2|X)$
- Many potential algorithms to sample from this joint distribution
- One of the most common is Gibbs Sampling, a multivariate generalisation of Metropolis

GIBBS SAMPLING

- The problem is the proposal. To compute the acceptance probability:

$$r_{\theta_1,\theta_2} = \frac{pr(\theta_{1,pr}, \theta_{2,pr}|X)}{pr(\theta_1, \theta_2|X)}$$

- We must be able to make a multivariate proposal
- Gibbs instead samples from **conditional** distributions
- Propose each parameter in turn (which has its own sampler), accept/reject, then move to next parameter
- Suffers from high autocorrelation if we use Metropolis as the sampler

A SIMPLE MULTIVARIATE PROBLEM

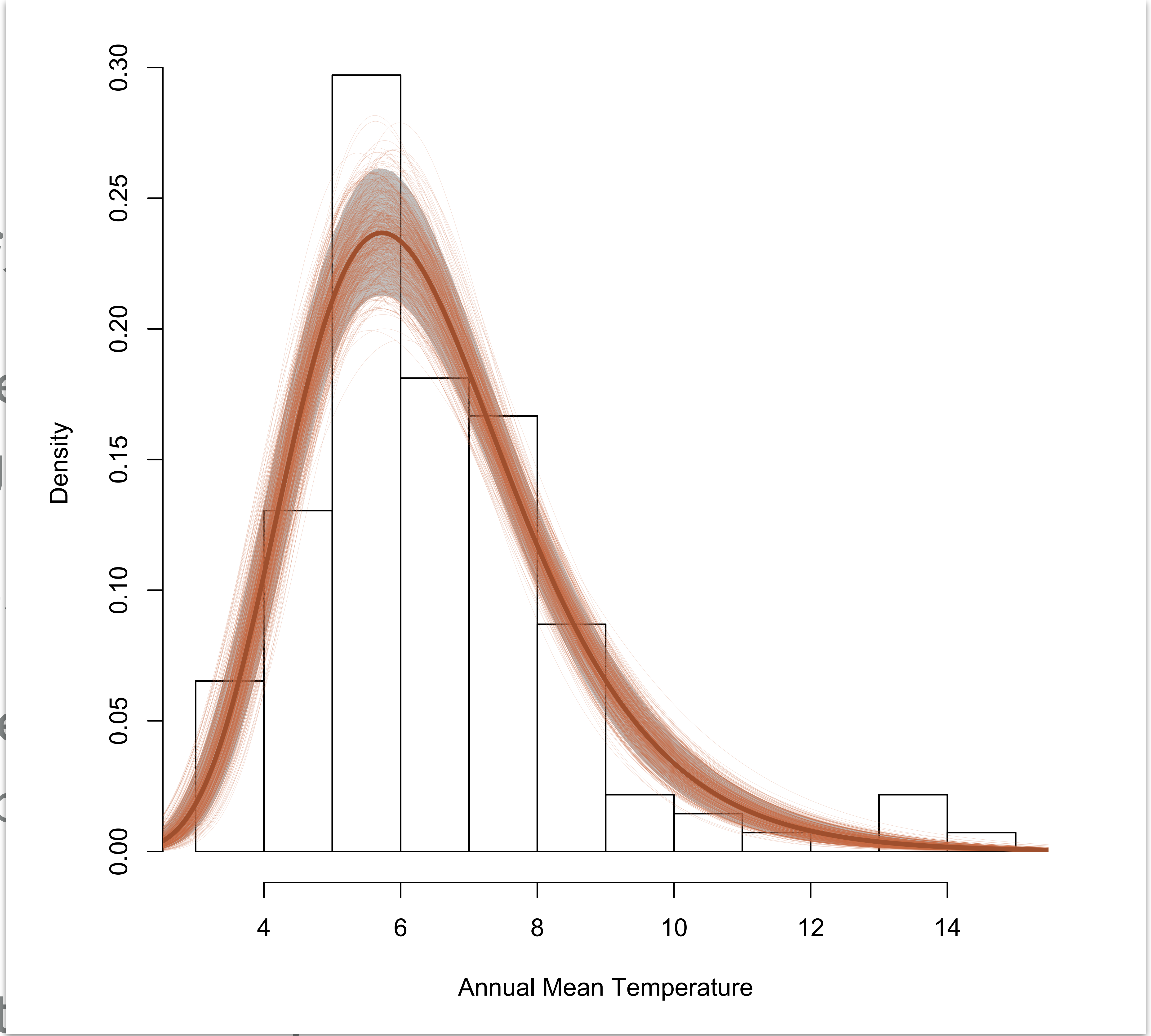
- Use the same data for *Tsuga canadensis*
- This time we will estimate the parameters of a distribution describing mean annual temperature in plots containing *T. canadensis*.
- Examine the histogram of temperatures. What distribution is appropriate?
- Use the `mcmc` package in R. Be sure to examine the help. Write a likelihood and priors for the parameters of your distribution estimate them with the `metrop()` function.
- How closely does your distribution match the original data?

A SIMPLE MULTIVARIATE PROBLEM

- Use the same data for *Tsuga canadensis*
- This time we will estimate the parameters of a distribution describing mean annual temperature in plots containing *T. canadensis*.
- Examine the histogram of temperatures. What distribution is appropriate?
- Use the `mcmc` package in R. Be sure to examine the help. Write a likelihood and priors for the parameters of your distribution estimate them with the `metrop()` function.
- How closely does your distribution match the original data?

A SIMPLE MULTIVARIATE PROBLEM

- Use the same data for *Tsuga canadensi*
- This time we will estimate the paramete annual temperature in plots containing
- Examine the histogram of temperature
- Use the `mcmc` package in R. Be sure to e priors for the parameters of your distrib function.
- How closely does your distribution mat



OTHER METHODS TO MAKE POSTERIOR INFERENCE

- Many MCMC variations, many of which are modifications to Metropolis
- Maximum *a posteriori* estimation (MAP): Bayesian equivalent of MLE
    - Can use any optimisation algorithm
    - Laplace approximation is frequently used
- Hamiltonian Monte Carlo
- We will focus Stan (which is a variety of HMC)