lecture 06

bayes

Bayesian inference; rejection algorithm; a Bayesian invariance property; censoring; duration models

Paper and Workshop

review

example exam questions

Define a *first difference.* Explain how and why the first difference unifies diverse statistical models. Use two diverse examples. Use words and/or equations, but be precise.

Explain the conceptual framework of {marginaleffects}. Identify the main functions and their key arguments.

For claims about the effect of X on Y, what is a *good default* quantity of interest? Defend your choices.

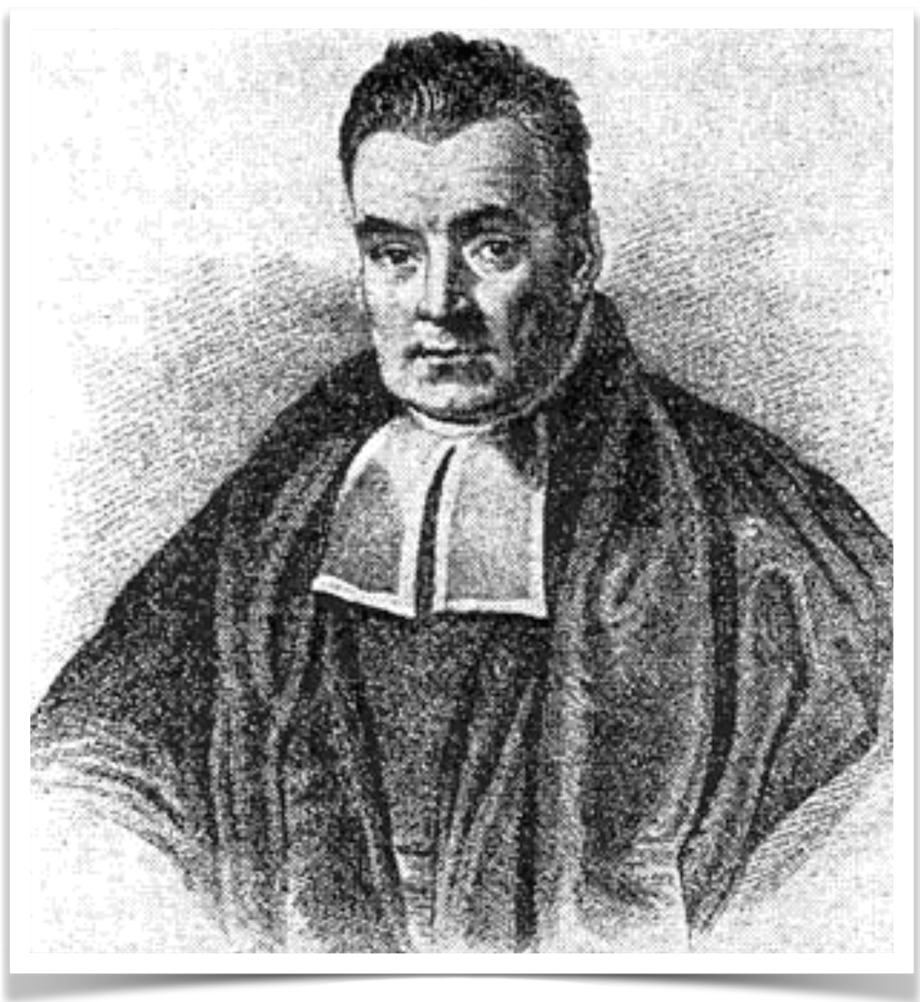
Bayesian Inference

Frequentism: a toolkit for evaluating estimators (e.g., bias, efficiency, coverage); ML offers some frequentist guarantees.

Bayesian inference: a method for finding the "correct" inference.

Bayesian Inference

(just Bayes' rule)



Bayesian Inference

(just Bayes' rule)

For continuous random variables [edit]

For two continuous random variables X and Y, Bayes' theorem may be analogously derived from the definition of conditional density:

$$f_{X|Y=y}(x) = \frac{f_{X,Y}(x,y)}{f_Y(y)}$$

$$f_{Y|X=x}(y) = \frac{f_{X,Y}(x,y)}{f_X(x)}$$

Therefore,

$$f_{X|Y=y}(x) = \frac{f_{Y|X=x}(y) f_X(x)}{f_Y(y)}.$$

IMPORTANT:
We have 3 different fs.

$$f(\theta \mid x) = \frac{f(x \mid \theta) \times f(\theta)}{\int_{-\infty}^{\infty} f(x \mid \theta) f(\theta) d\theta}$$

$$\boxed{f(\theta \mid x)} = \frac{\boxed{f(x \mid \theta)} \otimes \boxed{f(\theta)}}{\boxed{\int_{-\infty}^{\infty} f(x \mid \theta) f(\theta) d\theta}}$$

1. Choose a distribution for the data (i.e., "**the likelihood**" or model of the data).
2. Choose a **prior distribution** to describe your prior beliefs.
3. Update the prior distribution upon observing the data by computing the **posterior distribution**.

easy! (i) Multiply the likelihood and prior density.

HARD! (ii) Normalize, so that posterior integrates to one.

$$f(\theta \mid x) = \frac{f(x \mid \theta) \times f(\theta)}{\int_{-\infty}^{\infty} f(x \mid \theta) f(\theta) d\theta}$$

HARD!

Solutions

Conjugate Prior

Choose a likelihood and prior so that the posterior and prior are from the same family of probability distributions (e.g., both beta).

(restricts prior specification)

Unnormalized Posterior

$$f(\theta \mid x) \propto f(x \mid \theta) \times f(\theta)$$

(correct shape, but doesn't integrate to one)

Unnormalized Log-Posterior

$$\log f(\theta \mid x) \propto \log f(x \mid \theta) + \log f(\theta)$$

(turns out to be really handy!)

Simple Example

toothpaste cap problem

The Likelihood

$$f(x_i \mid \pi) = \pi^{x_i}(1-\pi)^{(1-x_i)}$$

$$f(x \mid \pi) = f(x_1 \mid \pi) \times \ldots \times f(x_N \mid \pi) = \prod_{i=1}^{N} f(x_i \mid \pi)$$

$$f(x \mid \pi) = \pi^{k}(1-\pi)^{(N-k)}, \text{where } k = \sum_{i=1}^{N} x_i$$

Recall we had N = 150 and k = 8 as an example dataset—we've got the likelihood already.

The Prior

the beta distribution

$$f(x) = \frac{1}{B(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

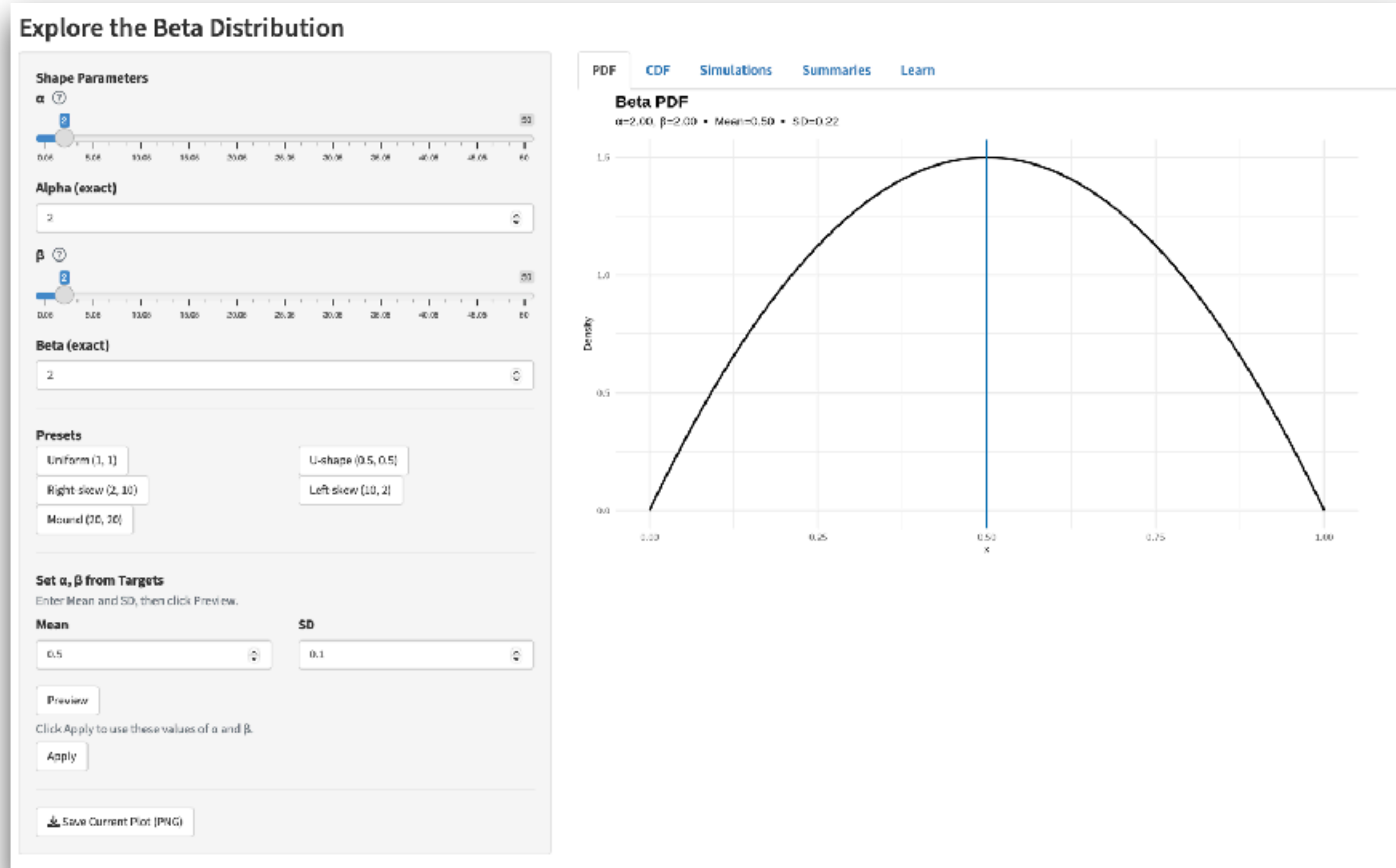
the prior
(which is beta)

$$f(\pi) = \frac{1}{B(\alpha^*,\beta^*)} \pi^{\alpha^*-1}(1-\pi)^{\beta^*-1}$$

two prior parameters

What prior parameters make sense
for the toothpaste cap problem?

https://carlislerainey.shinyapps.io/shiny-beta/



Understand your priors.

https://gist.github.com/carlislerainey/45414e0d9f22e4e1960449402e6a8048

Let’s plot our priors.

https://gist.github.com/carlislerainey/45414e0d9f22e4e1960449402e6a8048

(note: this code plots priors and posteriors—only run the prior part for now)

The Posterior

$$f(\theta \mid x) = \frac{f(x \mid \theta) \times f(\theta)}{\int_{-\infty}^{\infty} f(x \mid \theta) f(\theta) d\theta}$$

$$f(\pi \mid x) = \frac{\left[\pi^k (1-\pi)^{(N-k)}\right] \times \left[\frac{1}{B(\alpha^*, \beta^*)} \pi^{\alpha^* - 1} (1-\pi)^{\beta^* - 1}\right]}{C_1}$$

work on whiteboard

Some useful equations below

the posterior (where we're starting)

$$f(\pi \mid x) = \frac{\left[\pi^k (1-\pi)^{(N-k)}\right] \times \left[\frac{1}{B(\alpha^*,\beta^*)} \pi^{\alpha^*-1} (1-\pi)^{\beta^*-1}\right]}{C_1}$$

the posterior (where we're heading)

$$f(\pi \mid x) = \frac{\pi^{\overbrace{(\alpha^*+k)}^{\alpha'}-1} \times (1-\pi)^{\overbrace{[\beta^*+(N-k)]}^{\beta'}-1}}{B(\alpha^*+k, \beta^*+[N-k])}$$

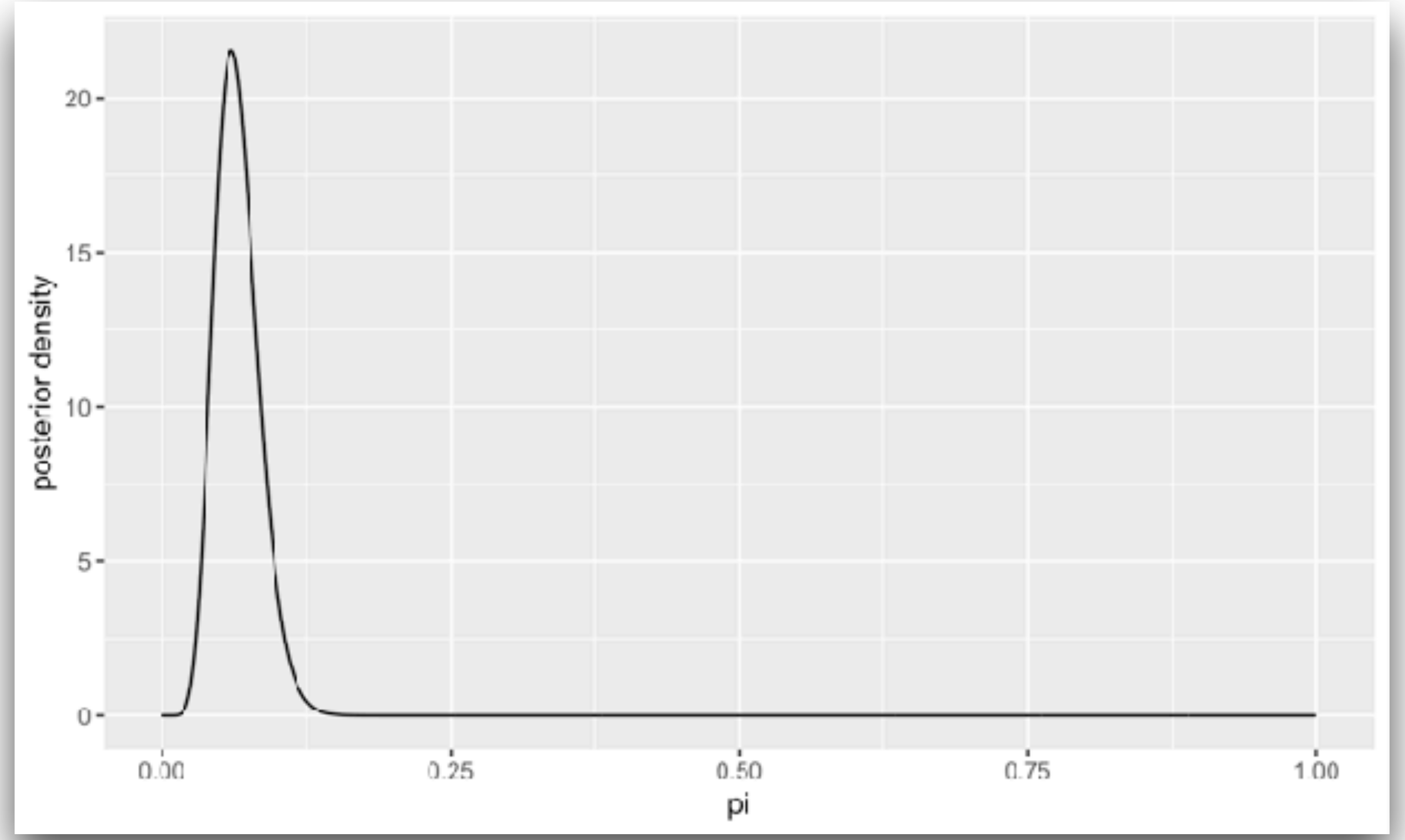
$$= \frac{\pi^{\alpha'-1} \times (1-\pi)^{\beta'-1}}{B(\alpha',\beta')}, \text{where } \alpha' = \alpha^* + k \text{ and } \beta' = \beta^* + [N-k]$$

the beta distribution

$$f(x) = \frac{1}{B(\alpha,\beta)} x^{\alpha-1} (1-x)^{\beta-1}$$

```r
# prior
alpha_star <- 3
beta_star <- 15

# data
N <- 150
k <- 8

# posterior
alpha_prime <- alpha_star + k
beta_prime <- beta_star + (N - k)
```

```r
# plot posterior pdf
ggplot() +
  xlim(0, 1) +
  stat_function(fun = dbeta, n = 1001,
                args = list(shape1 = alpha_prime,
                            shape2 = beta_prime)) +
  labs(x = "pi",
       y = "posterior density")
```



```
# find posterior mean
alpha_prime/(alpha_prime + beta_prime)
```

```
> # find posterior mean
> alpha_prime/(alpha_prime + beta_prime)
[1] 0.06547619
>
```

Let’s try it with your prior

https://gist.github.com/carlislerainey/5064ecde24ea764817da73b037e7a18b

Let's compare our inferences

https://gist.github.com/carlislerainey/155e2fee1c01620736120c9a8bbccd6d

A Few Conjugate Priors

Model of Data	Prior
Bernoulli	Beta
Poisson	Gamma
Normal	Normal; Inverse Gamma
Exponential	Gamma

Poisson: in the notes

exponential: an exercise

Rejection Algorithm

Observation: simulations from the posterior are as good as a closed-form solution.

Implication: we don't need to find the posterior; we need to find a way to ***simulate from*** the posterior.

```r
# prior parameters
alpha_prior <- 3
beta_prior <- 15

# data
k <- 8
N <- 150

# posterior parameters
alpha_posterior <- alpha_prior + k
beta_posterior <- beta_prior + N - k

# for compact calculations below
a1 <- alpha_posterior
b1 <- beta_posterior

# posterior simulation; trivial
n_sims <- 100000
pi_tilde <- rbeta(n_sims, shape1 = a1, shape2 = b1)

# posterior mean
a1 / (a1 + b1)  # closed form
```

```
[1] 0.06547619
```

```r
mean(pi_tilde)  # posterior sim
```

```
[1] 0.06543861
```

```r
# posterior sd
sqrt((a1 * b1) / ((a1 + b1)^2 * (a1 + b1 + 1))) # closed form
```

```
[1] 0.01902802
```

```r
sd(pi_tilde)  # posterior sim
```

```
[1] 0.01901066
```

```r
# posterior 5th percentile
qbeta(0.05, shape1 = a1, shape2 = b1)  # closed form
```

```
[1] 0.03737493
```

```r
quantile(pi_tilde, probs = 0.05)  # posterior sims
```

```
        5%
0.03742531
```

```r
# posterior 95th percentile
qbeta(0.95, shape1 = a1, shape2 = b1)  # closed form
```

```
[1] 0.09945329
```

```r
quantile(pi_tilde, probs = 0.95)  # posterior sims
```

```
       95%
0.09936871
```

But this idea is

even

more

powerful

than it initially seems

Surprising fact: It is easier to sample from a distribution than find the closed form.

How it is **even possible** to sample from distribution without knowing the closed form?

Example Algorithm: Rejection

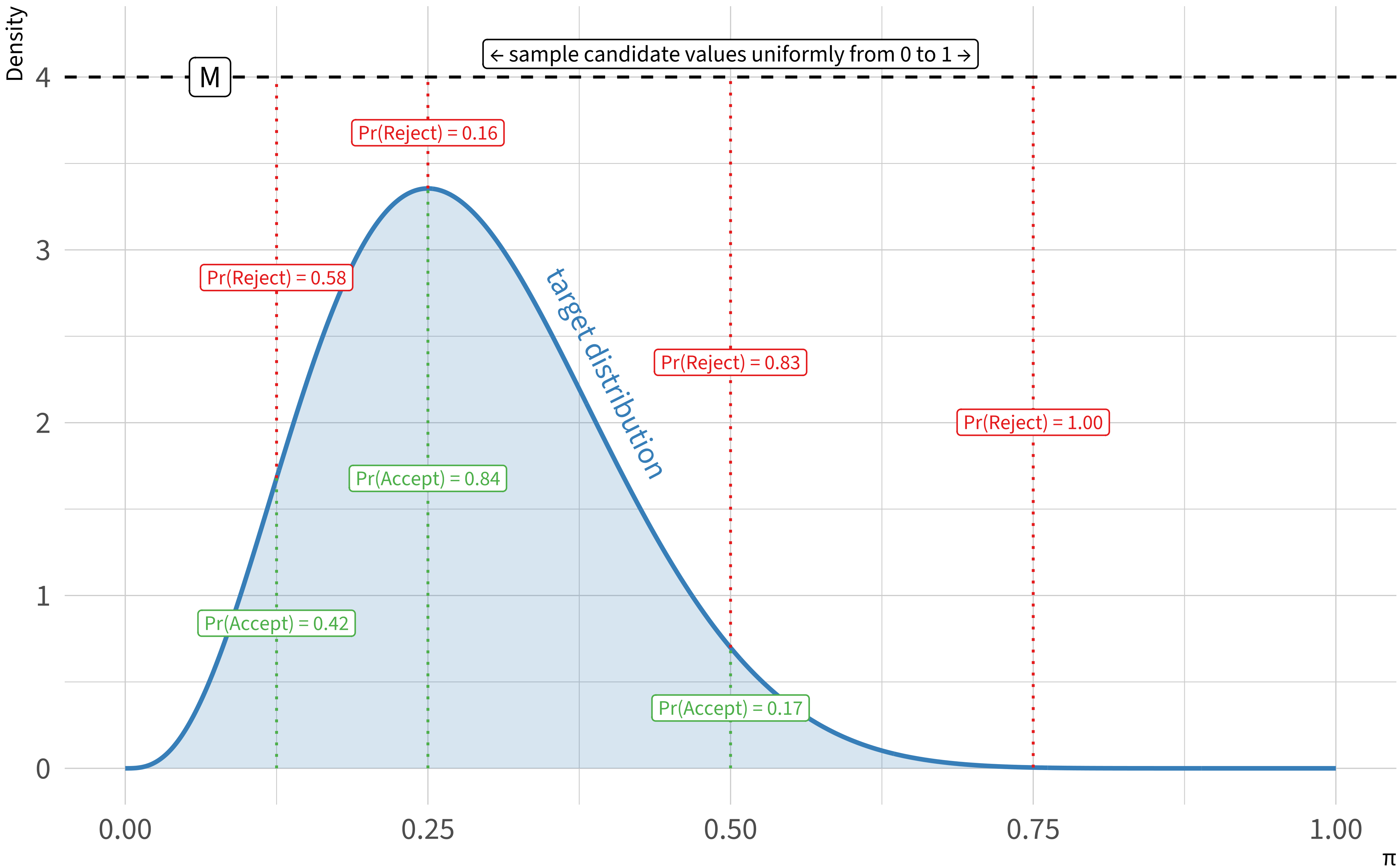
Example Targets:

Beta(4, 10)

Weird One! (Sawtooth Prior)

Illustrating the logic of the rejection algorithm

For a beta(4, 10) target distribution.



Algorithm: Rejection Sampling[a]

Inputs:

- The unnormalized posterior $f(\pi \mid y)$ on $[0, 1]$.
- Desired number of draws S.
- Envelope constant M such that $M > f(\pi)\ \forall \pi$.

Algorithm:

1. **Initialize:**Sets=1.

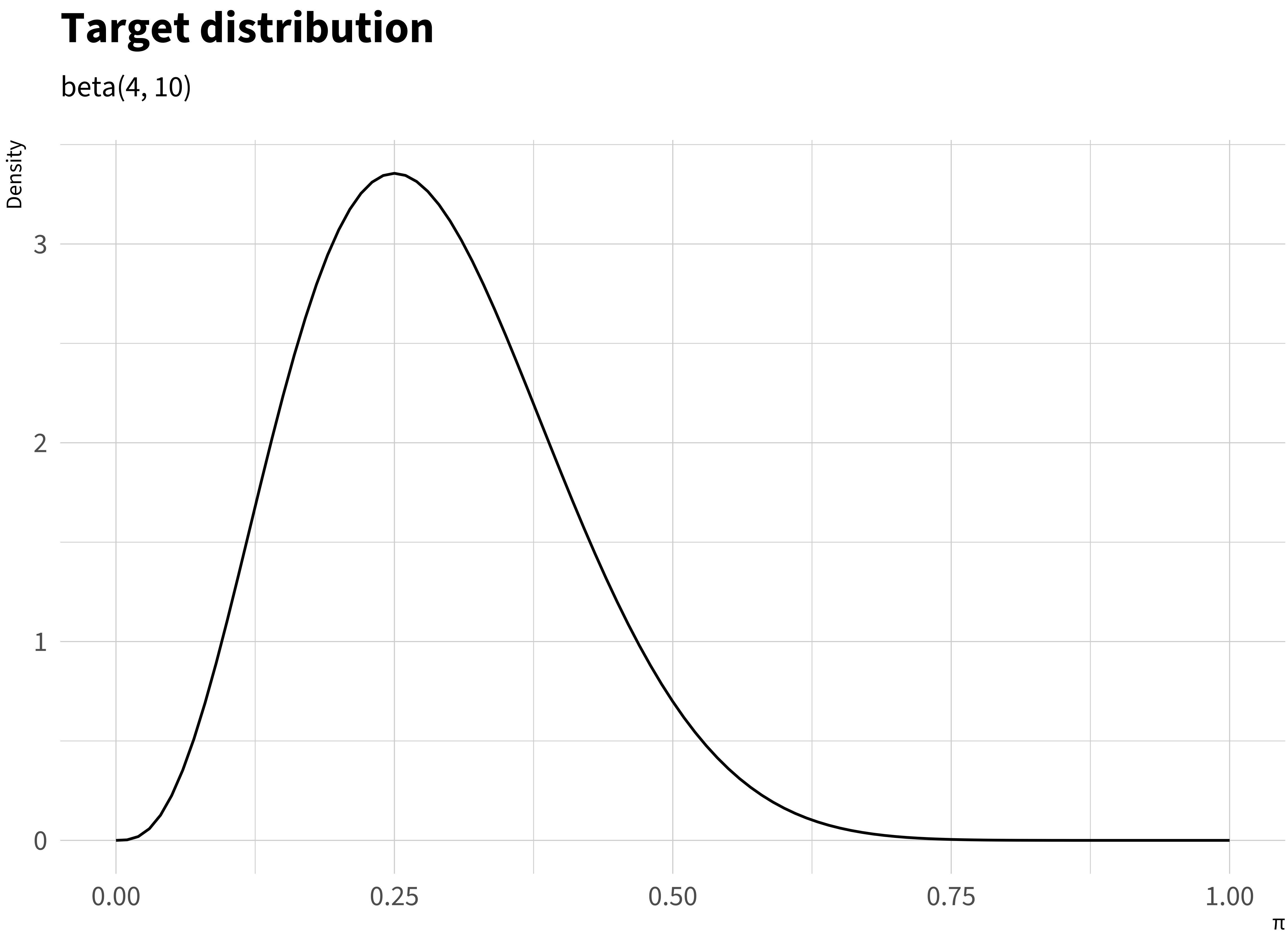
2. **Repeat until** $s = S$:
   (a) $z \sim \text{Uniform}(0, 1)$.
   (b) $u \sim \text{Uniform}(0, 1)$.
   (c) **Accept–reject step:**
   If $u \leq f(z)/M$, accept: set $\pi^{(s)} = z$, $s \leftarrow s + 1$.
   Otherwise, reject z and return to Step 2a.

Output: $\pi^{(1)}, \pi^{(2)}, \ldots, \pi^{(S)} \sim f(\pi \mid y)$.

[a] To make the rejection algorithm simple, I've written it to apply specifically to the posterior for the Bernoulli model, which has support $[0, 1]$. The target density doesn't need to be a posterior and can have support other than $[0, 1]$. The proposal distribution doesn't have to be uniform. The key is that M is larger than the maximum of the target distribution and draws are accepted with probability $f(z)/M$.

```r
rej <- function(f, S, M) {

  # create containers and initialize counters
  samples <- numeric(S)  # container to store samples
  rejects <- NULL  # container to track rejected values; for teaching; slow!
  s <- 1 # currently trying to take sample 1
  n_prop <- 0  # count proposals (for an acceptance-rate message)

  # so long as the current sample s is less
  #   than the desired samples S.
  #   do the following:
  while (s <= S) {

    # A: propose z ~ uniform(0,1)
    z <- runif(1)

    # B: draw u ~ uniform(0,1)
    u <- runif(1)

    # C: Accept or reject
    fz <- f(z) # compute once, for effeciency

    ## scenario 1: u <= f(z)/M  →  Accept
    if (u <= fz / M) {
      samples[s] <- z
      s <- s + 1
    }

    ## scenario 2: f(z) > M  →  shouldn't happen; error
    if (fz > M) stop("Stop: Envelope M is too small.")  # find appropriate M

    ## scenario 3: u > f(z)/M  →  Reject
    ##   tracking these values just for teaching and learning--not needed usually
    if (u > fz / M) {
      rejects <- c(rejects, z)
    }

    # track total proposals so far
    n_prop <- n_prop + 1
  }

  # return
  list(
    n_prop = n_prop,
    acc_rate = S / n_prop,
    samples = samples,
    rejects = rejects
  )
}
```

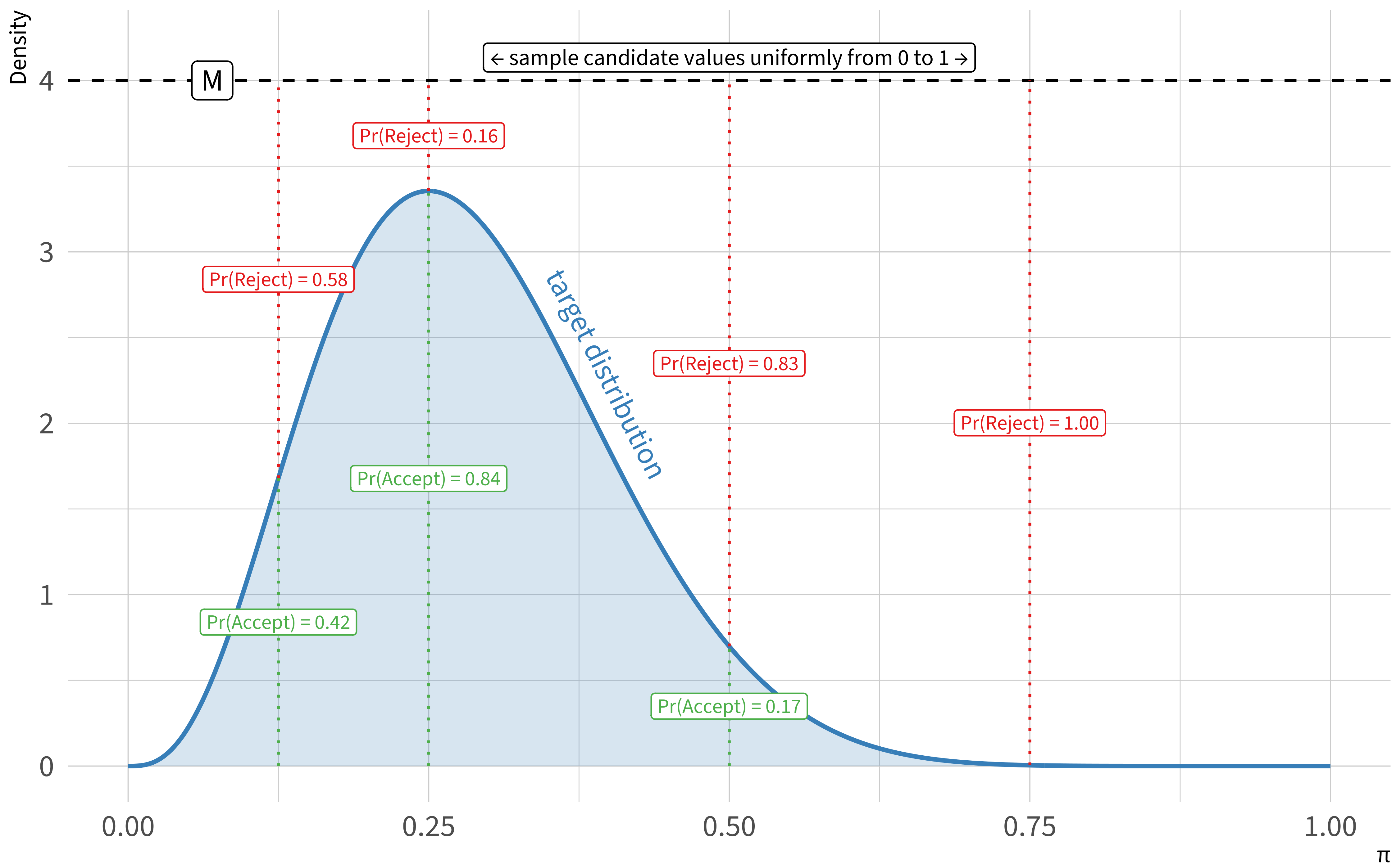
scenario 1 is the key

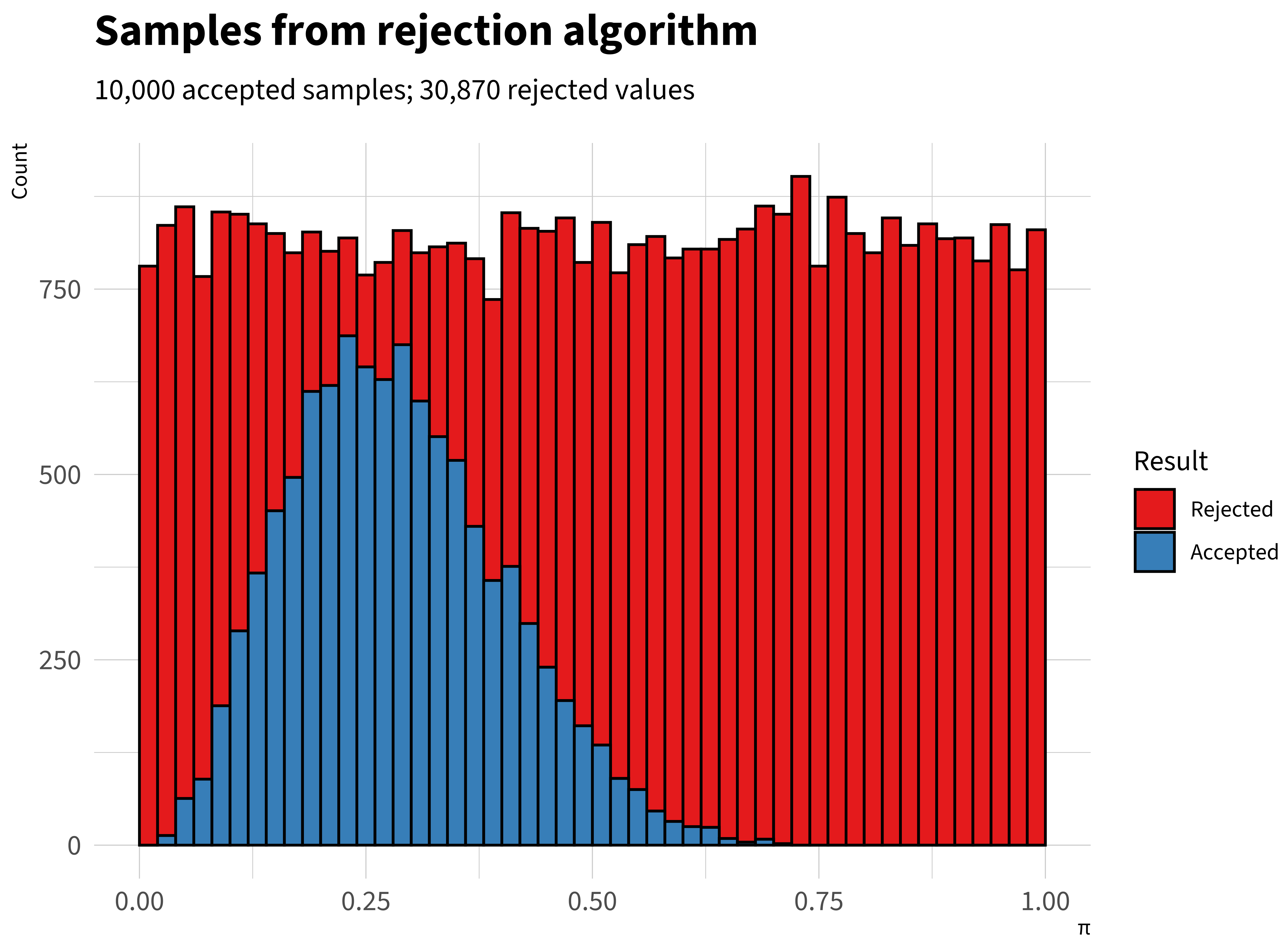


Target distribution
beta(4, 10)
Density
3
2
1
0
0.00
0.25
0.50
0.75
1.00
π

Illustrating the logic of the rejection algorithm

For a beta(4, 10) target distribution.




Samples from rejection algorithm
10,000 accepted samples; 30,870 rejected values
Count
750
500
250
0
0.00
0.25
0.50
0.75
1.00
π
Result
Rejected
Accepted

```
> # posterior mean
> 4/(4 + 10)  # closed-form
[1] 0.2857143
> mean(r$samples)  # simulation
[1] 0.2847601
```

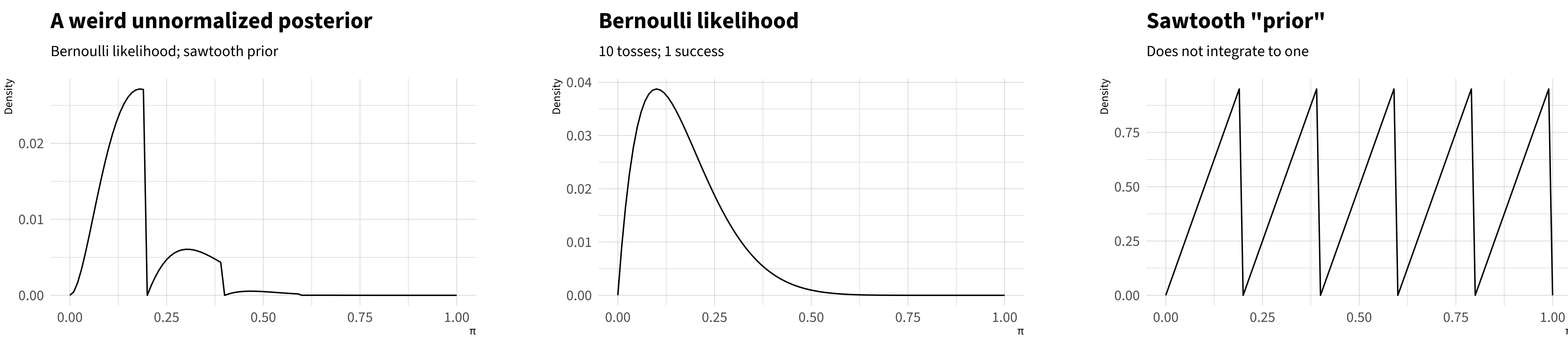
Let’s make it weird

```r
# unnormalized prior
prior_saw <- function(p, n_teeth = 5) {
  ((n_teeth*p) %% 1)
}
```

$$\text{posterior} = \frac{\text{Bernoulli likelihood} \times \dfrac{\text{sawtooth prior}}{\text{normalizing constant}}}{\text{normalizing constant}}$$

$$\text{posterior} \propto \text{Bernoulli likelihood} \times \text{unnormalized sawtooth prior}$$

$$\text{posterior} \propto \text{Bernoulli likelihood} \times \text{unnormalized sawtooth prior}$$



Find mean of posterior?!?

Find quantiles of posterior?!?

HOW?!?

```r
# unnormalized prior
prior_saw <- function(p, n_teeth = 5) {
  ((n_teeth*p) %% 1)
}

# likelihood (10 tosses; 1 success )
lik <- function(p) {
  p^1 * (1-p)^9
}

# unnormalized posterior
unnormalized_posterior <- function(p) {
  lik(p)*prior_saw(p)
}

# rejection algorithm
r <- rej(unnormalized_posterior, S = 10000, M = 0.03)
```

```
> r <- rej(unnormalized_posterior, S = 10000, M = 0.03)
💪 Successfully generated 10,000 samples! 🎉

✅ Accepted samples: 10,000
❌ Rejected samples: 58,263
% Acceptance rate: 15%
⏱ Total time: 7.4s
```

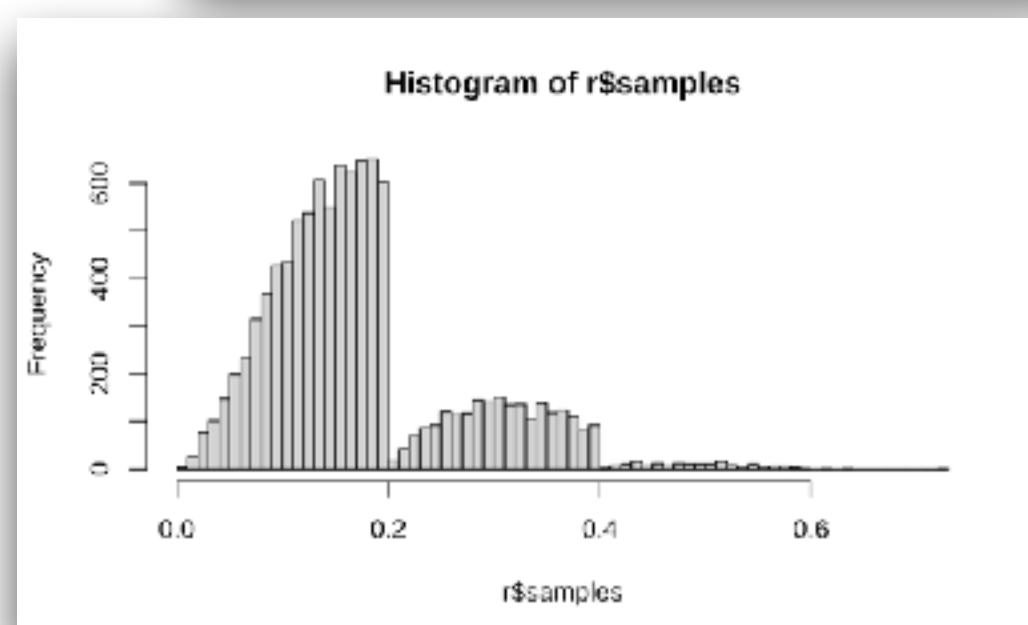
```r
# posterior mean
mean(r$samples)
```

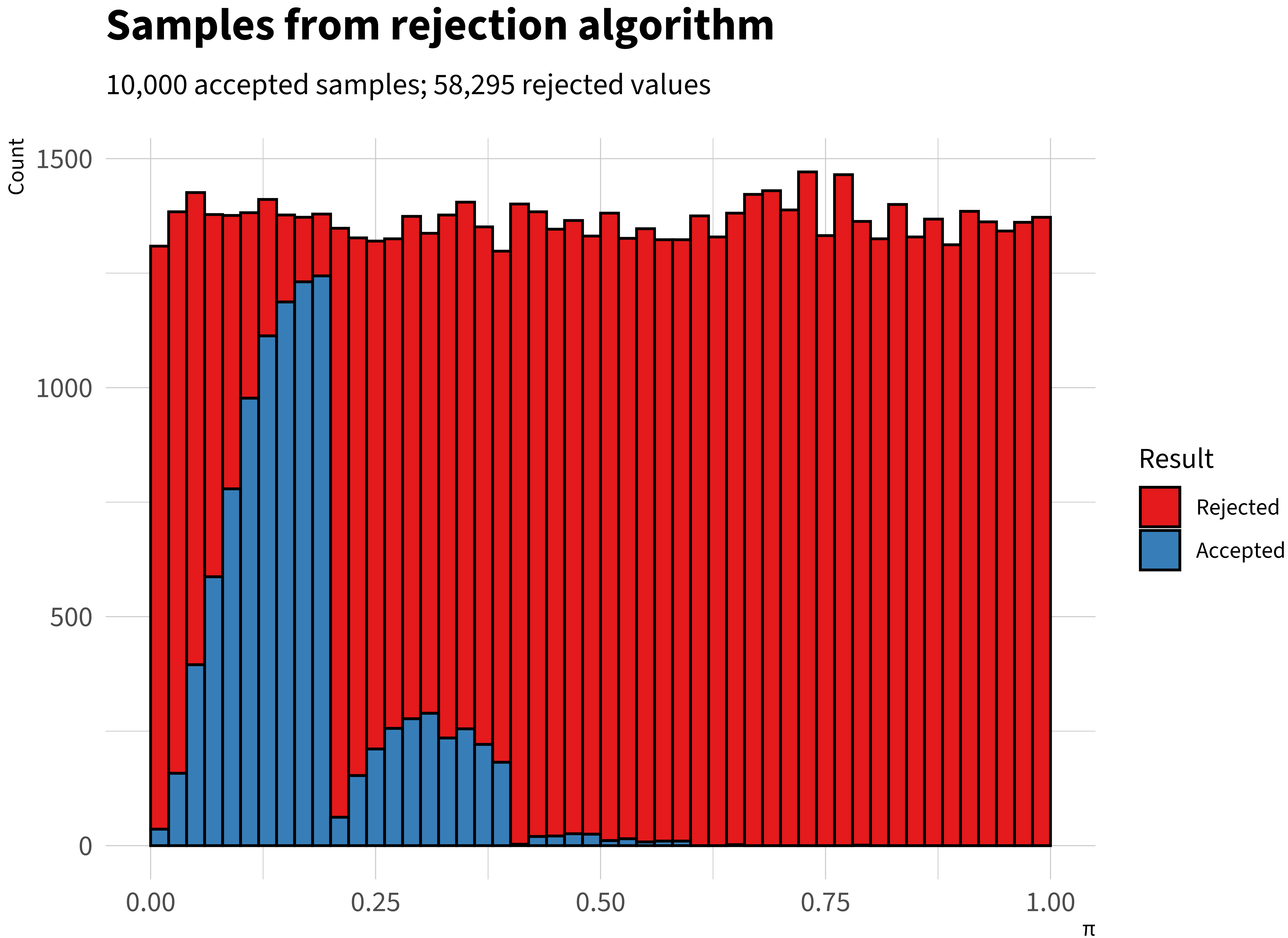
```
> # posterior mean
> mean(r$samples)
[1] 0.1759005
```

```r
# 90% credible interval
quantile(r$samples, probs = c(0.05, 0.95))
```

```
> # 90% credible interval
> quantile(r$samples, probs = c(0.05, 0.95))
        5%        95%
0.05770977 0.36608499
```

```r
# histogram
hist(r$samples, breaks = 100)
```




Samples from rejection algorithm
10,000 accepted samples; 58,295 rejected values
Count
1500
1000
500
0
0.00
0.25
0.50
0.75
1.00
π
Result
Rejected
Accepted

$$f(\theta \mid x) = \frac{f(x \mid \theta) \times f(\theta)}{\underbrace{\int_{-\infty}^{\infty} f(x \mid \theta) f(\theta) d\theta}_{\text{HARD!}}}$$

Notice how the rejection algorithm allowed us to completely skip the hard parts.

Bayesian Normal Linear Regression (Conjugate Prior)

$$\text{Likelihood:} \quad y \mid \beta, \sigma^2 \sim \mathcal{N}(X\beta,\ \sigma^2 I)$$

$$\text{Prior:} \quad \beta \mid \sigma^2 \sim \mathcal{N}(\beta^*,\ \sigma^2 V^*), \quad \sigma^2 \sim \text{Inv-Gamma}(a^*, b^*)$$

$$\text{Posterior:} \quad \beta \mid \sigma^2, y \sim \mathcal{N}(\beta',\ \sigma^2 V'), \quad \sigma^2 \mid y \sim \text{Inv-Gamma}(a', b')$$

$$\text{Updates:} \quad V' = \big((V^*)^{-1} + X^\top X\big)^{-1}, \qquad \beta' = V'\big((V^*)^{-1}\beta^* + X^\top y\big)$$

$$a' = a^* + \tfrac{n}{2}, \qquad b' = b^* + \tfrac{1}{2}\big(y^\top y + (\beta^*)^\top (V^*)^{-1}\beta^* - (\beta')^\top (V')^{-1}\beta'\big)$$

Parameters: $\sigma^2 =$ error variance, $V^*, V' =$ prior/posterior covariances, $a^*, a' =$ shape, $b^*, b' =$ scale.

Note: $\beta \mid y \sim t_{2a'}\Big(\beta',\ \frac{b'}{a'}\, V'\Big)$ (marginal posterior after integrating out σ^2).

A Bayesian Invariance Property

$$\hat{\tau}_{mean} \neq \tau\left(\hat{\theta}_{mean}\right)$$

The transformed posterior mean is *NOT* the posterior mean of the transformation.

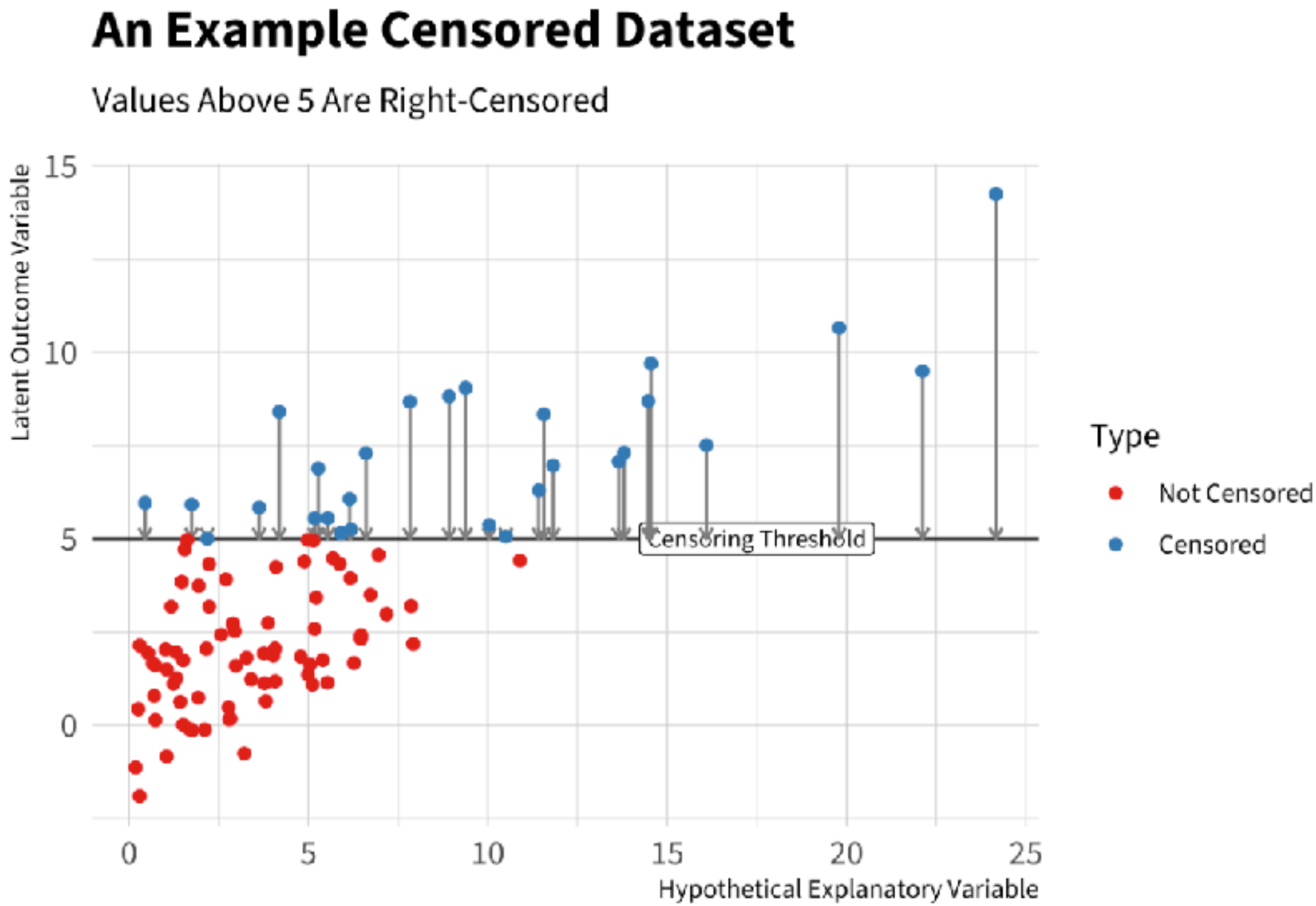
A Bayesian invariance property. Suppose $\{\tilde{\theta}^{(s)}\}_{s=1}^{S}$ are posterior simulations of θ. Let $\tau = \tau(\theta)$ be a quantity of interest for any function τ. Then posterior simulations of τ can be obtained by applying τ to each draw $\tilde{\theta}^{(s)}$ so that $\tilde{\tau}^{(s)} = \tau\left(\tilde{\theta}^{(s)}\right)$. Summaries of the posterior distribution of τ (e.g., mean, median, credible intervals) are obtained by summarizing the transformed draws $\{\tilde{\tau}^{(s)}\}_{s=1}^{S}$.

Transform simulations *before* summarizing.

```r
# posterior simulation
pi_tilde <- rbeta(10000, shape1 = alpha_prime, shape2 = beta_prime)
oof_tilde <- (1 - pi_tilde)/pi_tilde

mean(oof_tilde)  # correct
(1 - mean(pi_tilde))/mean(pi_tilde)  # NOT correct
```

```
> mean(oof_tilde)  # correct
[1] 15.56664
> (1 - mean(pi_tilde))/mean(pi_tilde)  # NOT correct
[1] 14.15623
```

Censoring



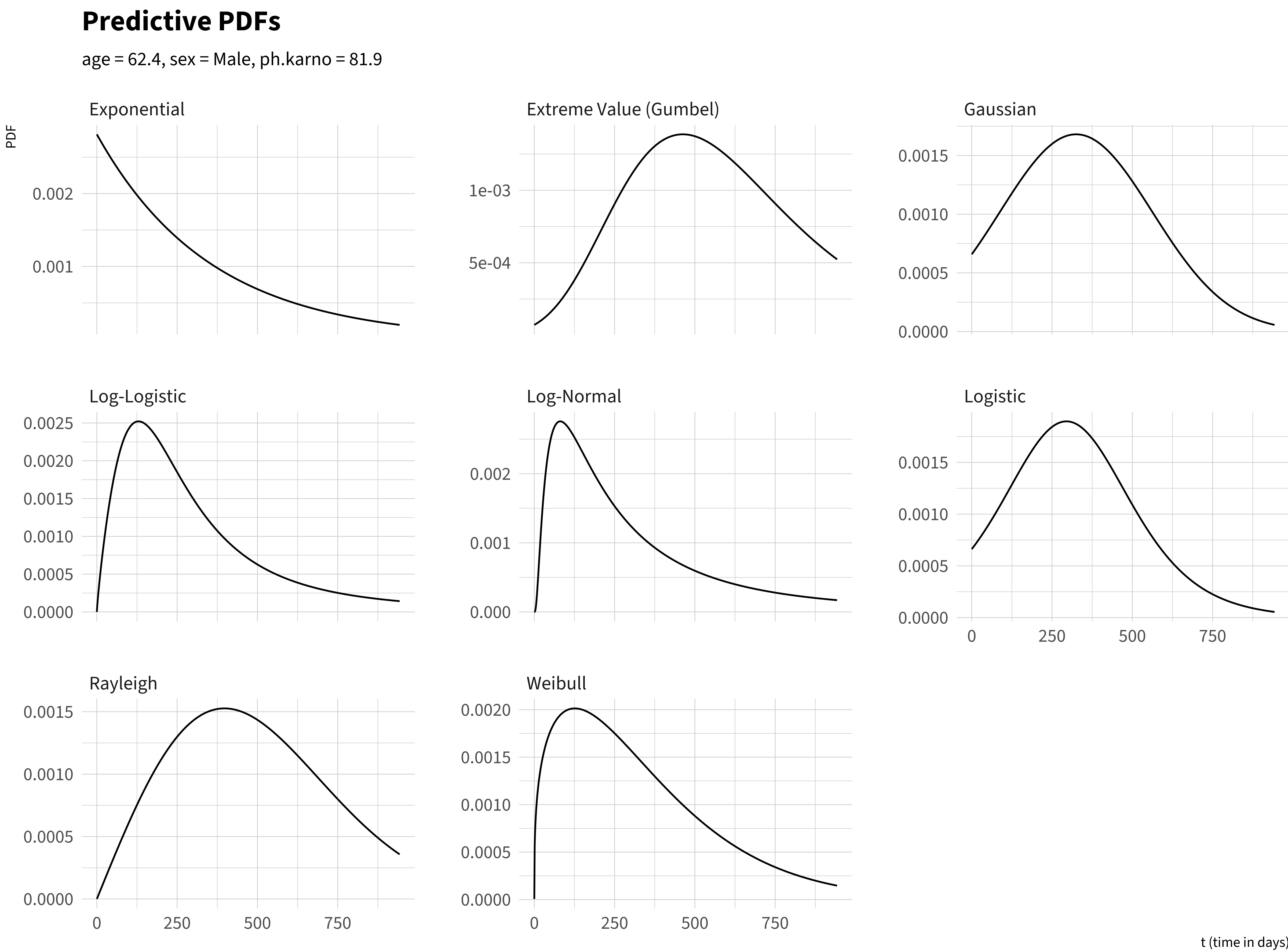
An Example Censored Dataset
Values Above 5 Are Right-Censored
Latent Outcome Variable
15
10
5
0
Censoring Threshold
0
5
10
15
20
25
Hypothetical Explanatory Variable
Type
Not Censored
Censored

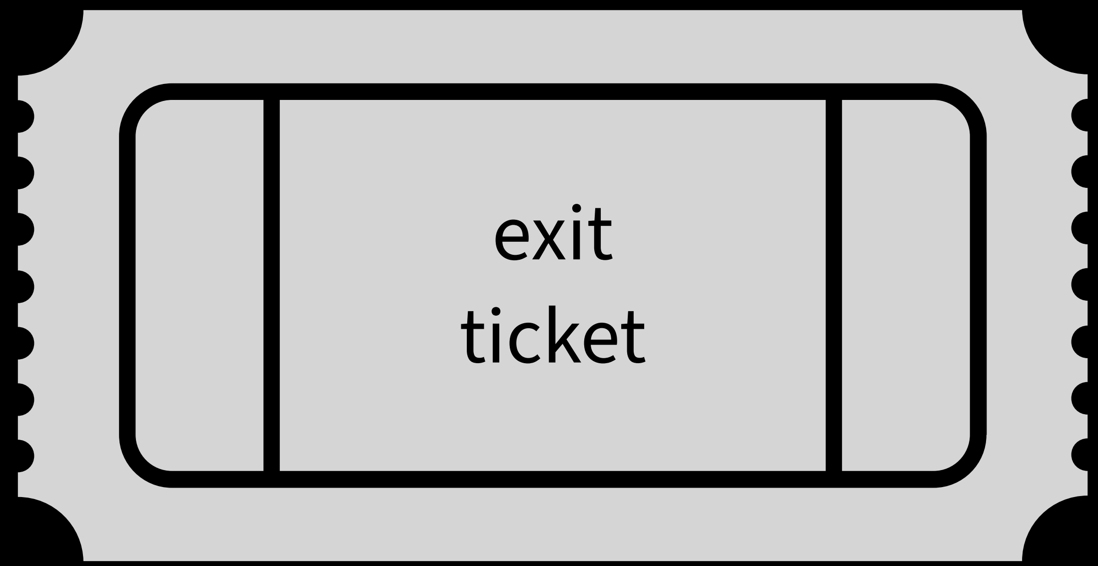
$$
\begin{array}{ccccccccccc}
y_1 = 2 & ; & y_2 > 3 & ; & y_3 = 1 & ; & y_4 < 1 & ; & y_5 \in (1,3) & ; & \cdots \\
f(2) & \times & \Pr(Y > 3) & \times & f(1) & \times & \Pr(Y < 1) & \times & \Pr(1 < Y < 3) & \times & \cdots \\
f(2) & \times & 1 - F(3) & \times & f(1) & \times & F(1) & \times & F(3) - F(1) & \times & \cdots
\end{array}
$$

Duration Models

```r
# load packages
library(survival)

# load data
canc <- survival::cancer |>
  mutate(sex = case_when(sex == 1 ~ "Male",
                         sex == 2 ~ "Female"))

# model
f <- Surv(time, status) ~ age + sex + ph.karno

# Exponential
fit_exp <- survreg(f, data = canc, dist = "exp")

# Log-Normal
fit_ln <- survreg(f, data = canc, dist = "lognormal")

# Weibull
fit_wei <- survreg(f, data = canc, dist = "weibull")

# Rayleigh
fit_ray <- survreg(f, data = canc, dist = "rayleigh")

# Extreme Value (Gumbel)
fit_extr <- survreg(f, data = canc, dist = "extreme")

# Gaussian (Normal)
fit_gaus <- survreg(f, data = canc, dist = "gaussian")

# Logistic
fit_logis <- survreg(f, data = canc, dist = "logistic")

# Log-Logistic
fit_llogis <- survreg(f, data = canc, dist = "loglogistic")
```

Model	k	BIC	Post. Prob.
Weibull	5	2304.508	75.3%
Log-Logistic	5	2306.753	24.5%
Exponential	4	2317.108	0.1%
Log-Normal	5	2325.959	0.0%
Rayleigh	4	2354.469	0.0%
Logistic	5	2358.144	0.0%
Gaussian	5	2363.521	0.0%
Extreme Value (Gumbel)	5	2443.159	0.0%


Predictive PDFs
age = 62.4, sex = Male, ph.karno = 81.9
PDF
Exponential
Extreme Value (Gumbel)
Gaussian
Log-Logistic
Log-Normal
Logistic
Rayleigh
Weibull
t (time in days)



List three important ideas from today's class. For each, briefly connect it to one or more ideas from last week.