Normal sampling model

Roger Federer

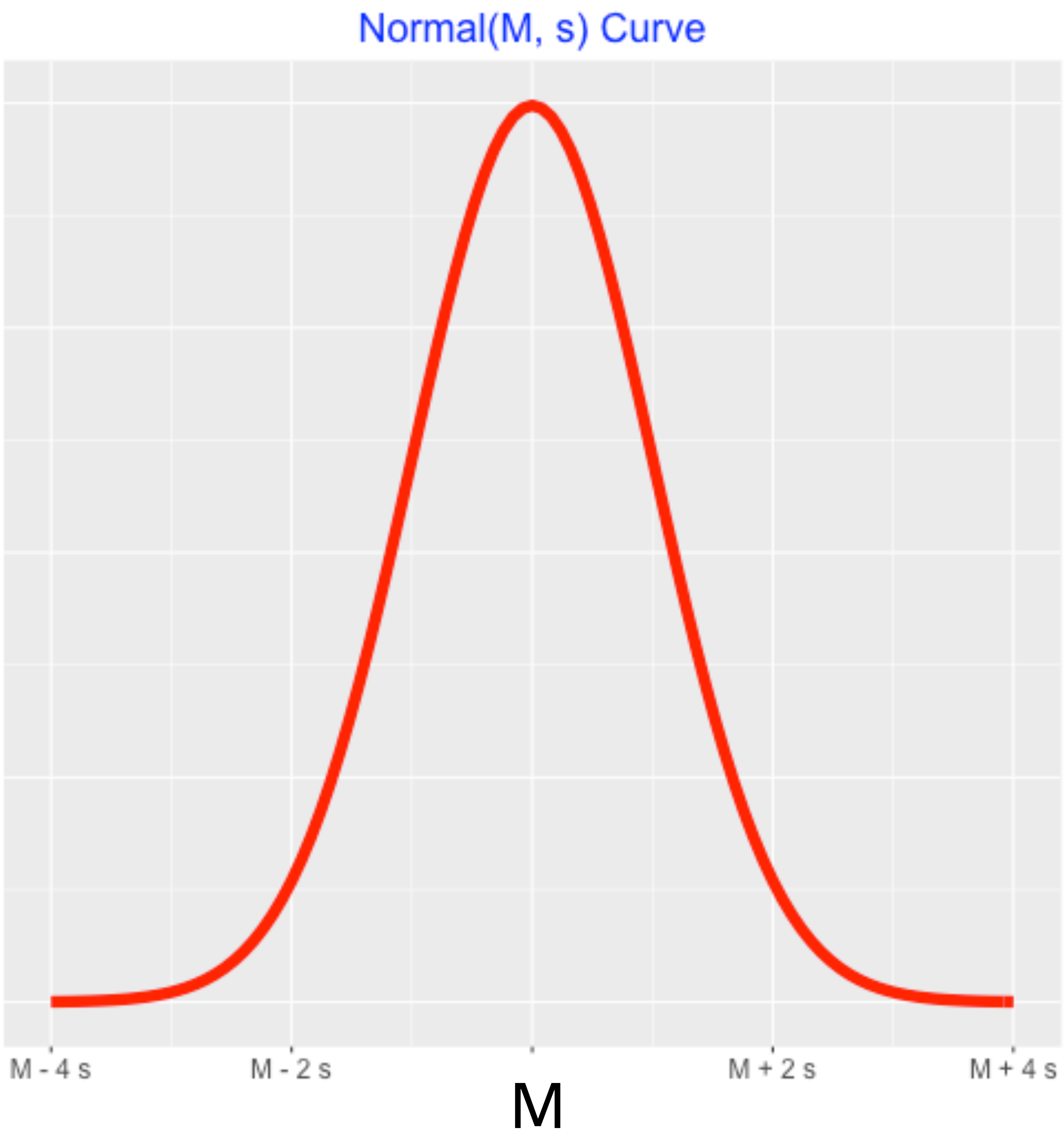
Serving in tennis

- Roger seems very “efficient” in his serving style
- Often wins service games quickly
- How long, on average, does it take Roger to serve?

Sampling model for time-to-serve

- Roger's time-to-serve measurements are normally distributed
- Mean **M** and standard deviation **s** (measured in seconds)
- Assume we know **s** = 4 seconds

Normal(M, s) model for time-to-serve

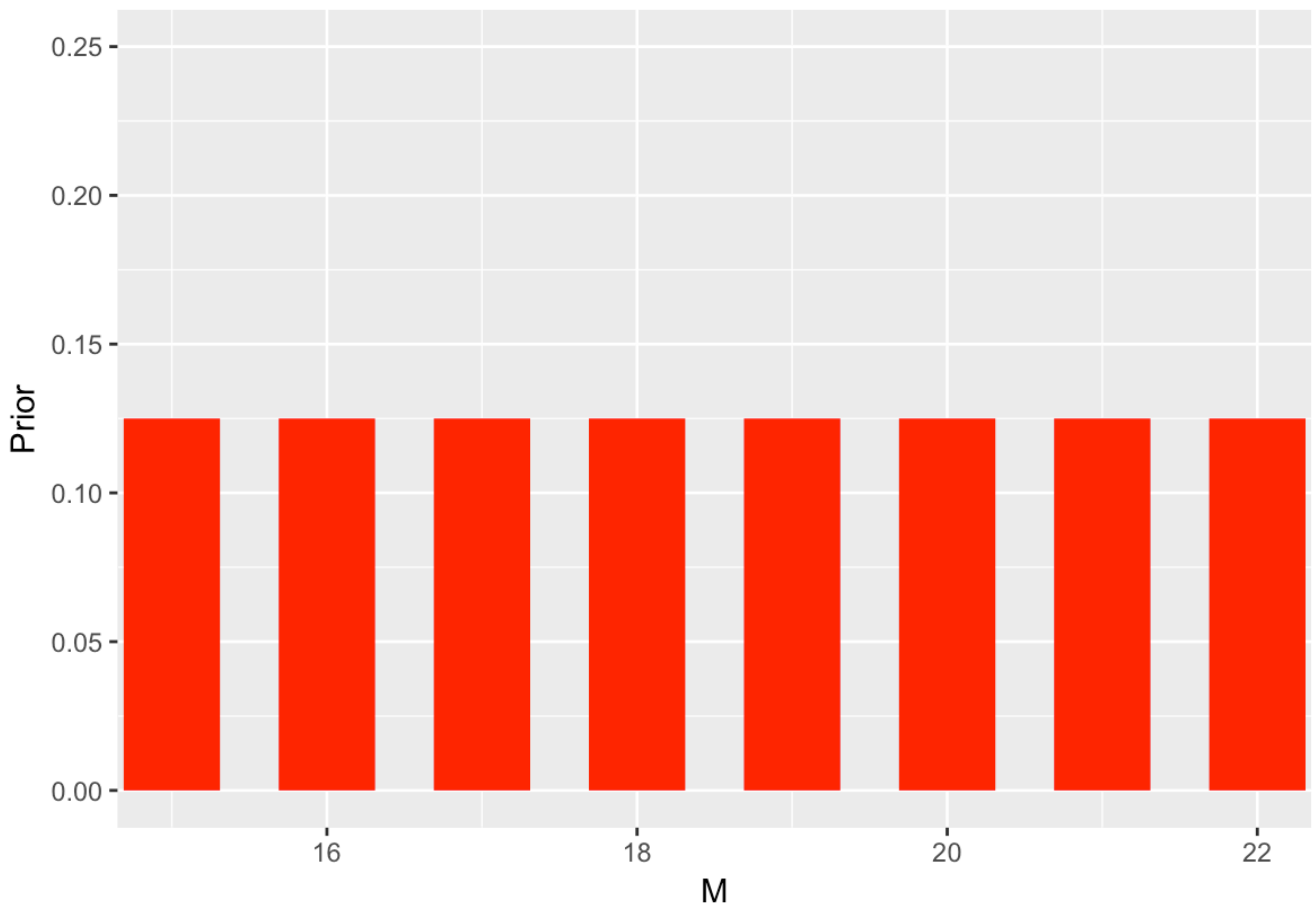


Prior beliefs about M

- Discrete prior (e.g. 15, 16, 17, ..., 22 seconds)
- No reason to prefer any of these values
- Assign M a uniform prior on these values

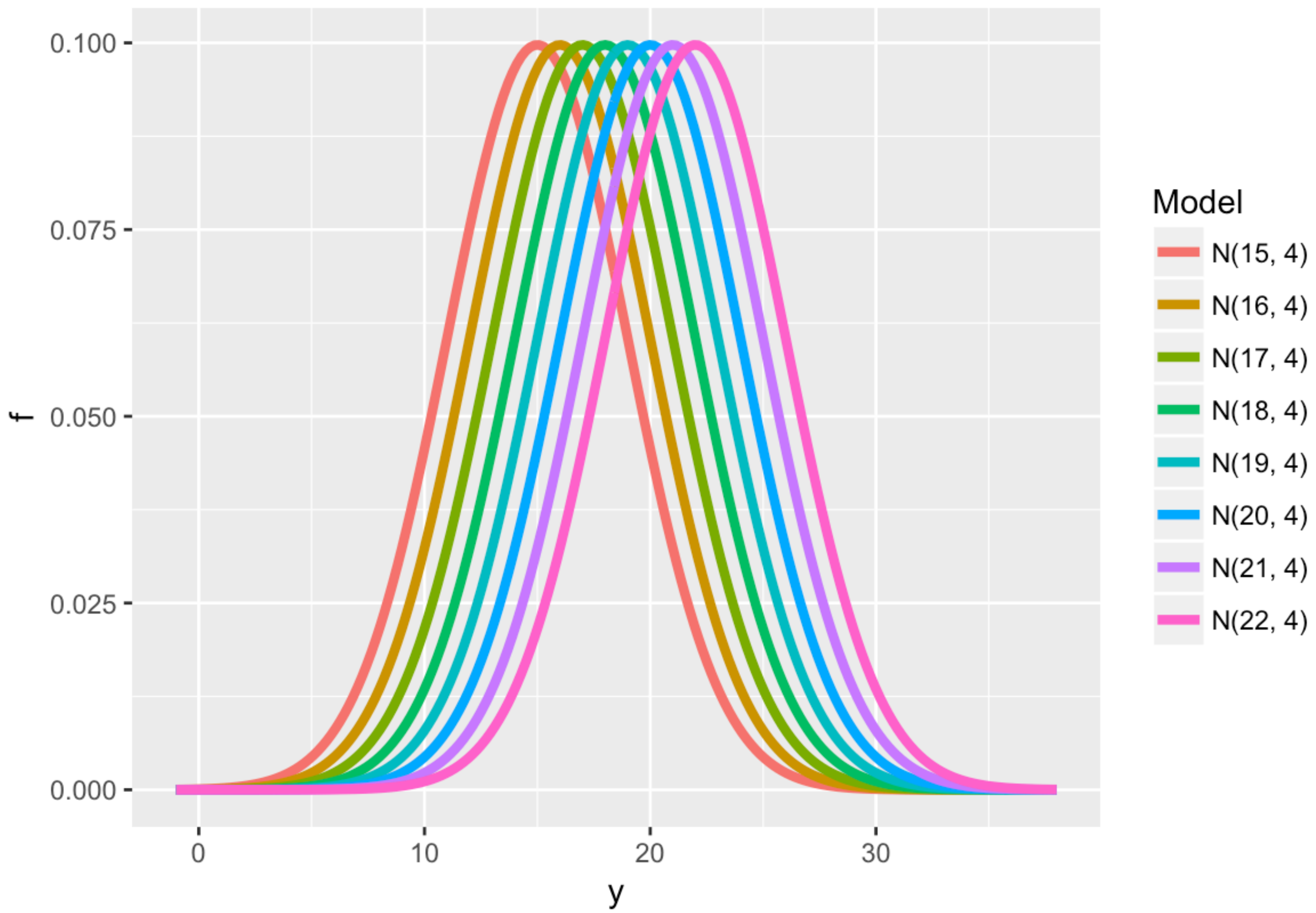
Graph of the prior

```
> library(TeachBayes)
> bayes_df <- data.frame(M = 15:22, Prior = rep(1/8, 8))
> prob_plot(bayes_df) + ylim(0, 0.25)
```



Deciding between 8 normal models

```
> library(TeachBayes)
> Models <- list(c(15, 4), c(16, 4), c(17, 4), c(18, 4),
                 c(19, 4), c(20, 4), c(21, 4), c(22, 4))
> many_normal_plots(Models)
```



Observe data

- Record the time-to-serve for 20 of Roger's serves
- Compute the mean value: $\bar{y} = 17.2$
- Associated standard error: $se = \frac{S}{\sqrt{n}} = \frac{4}{\sqrt{20}} = 0.89$

Note: standard error is another way to say standard deviation of the sample mean

Likelihood function

- Likelihood is sampling density of $\bar{y}$ — Normal(mean = M, sd = se)
- Substitute known values of $\bar{y}$ and standard deviation
- View as function of M

```
> bayes_df$Likelihood <- dnorm(17.2,
                               mean = bayes_df$M,
                               sd = 0.89)
```

Use bayesian_crank() to compute posterior

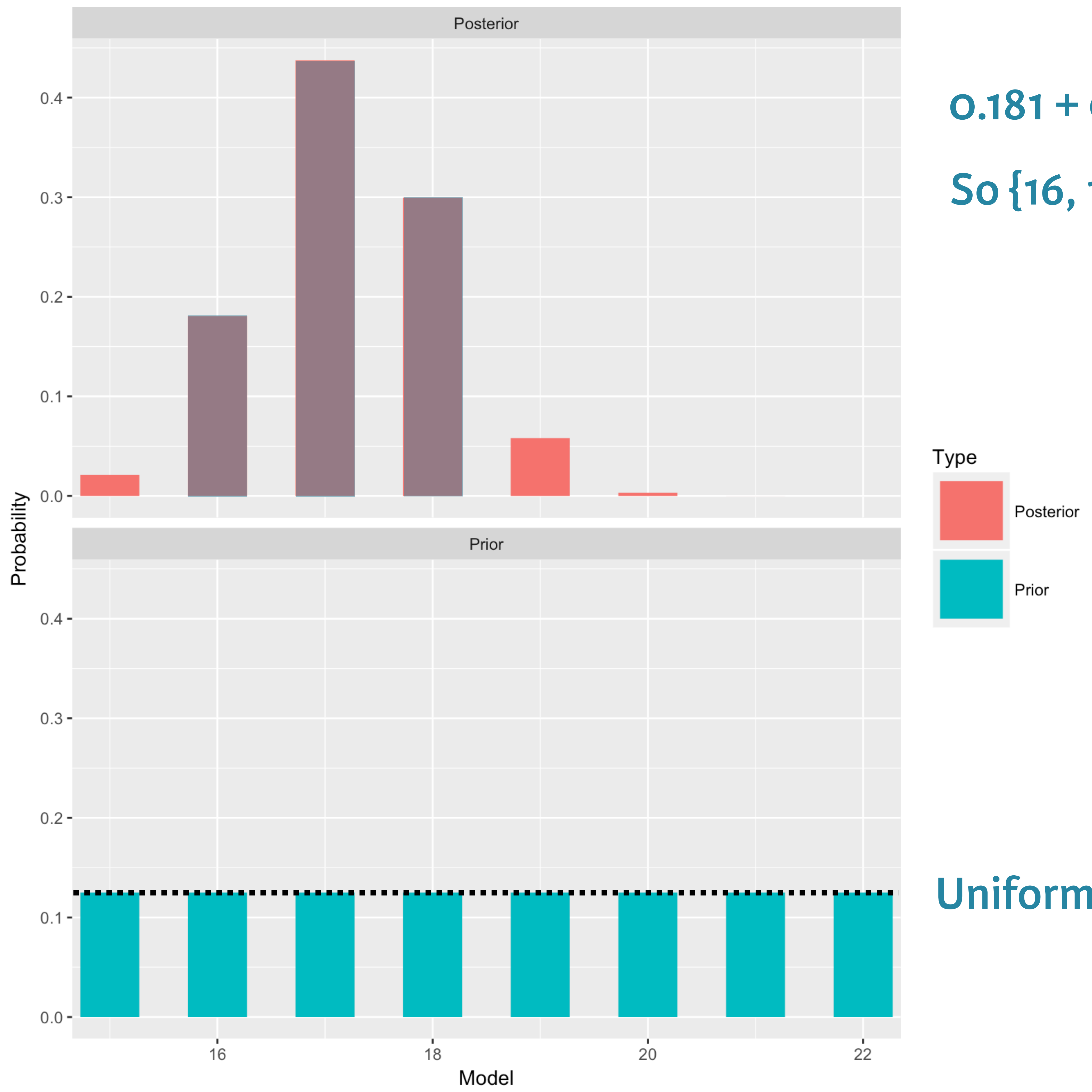
```
> library(TeachBayes)
> bayes_df <- bayesian_crank(bayes_df)
> round(bayes_df, 3)

   M Prior Likelihood Product Posterior
1 15 0.125      0.021   0.003     0.021
2 16 0.125      0.181   0.023     0.181
3 17 0.125      0.437   0.055     0.437
4 18 0.125      0.299   0.037     0.299
5 19 0.125      0.058   0.007     0.058
6 20 0.125      0.003   0.000     0.003
7 21 0.125      0.000   0.000     0.000
8 22 0.125      0.000   0.000     0.000
```

Posterior values sum to 1

Multiply prior and likelihood

Normalize

Plot prior and posterior of M



0.181 + 0.435 + 0.299 = 0.915

So {16, 17, 18} is the 91.5% probability region

Uniform prior

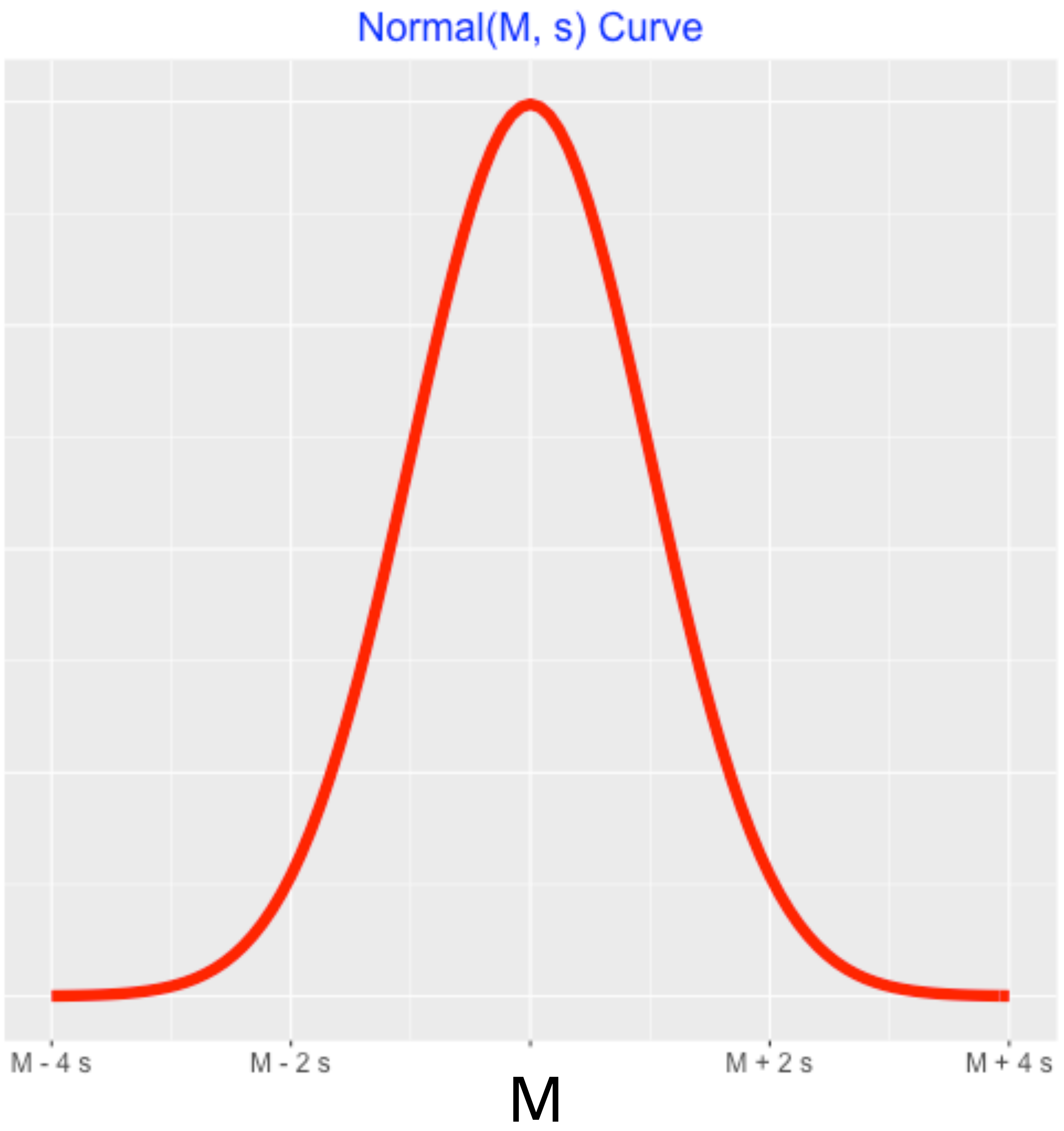
Let's practice!

Bayes with a continuous prior

Learning about time-to-serve

- Interested in how long Roger Federer takes to serve
- Assume times are normally distributed with mean M and standard deviation s (unit = seconds)
- Focus on the mean time-to-serve M

Normal sampling model

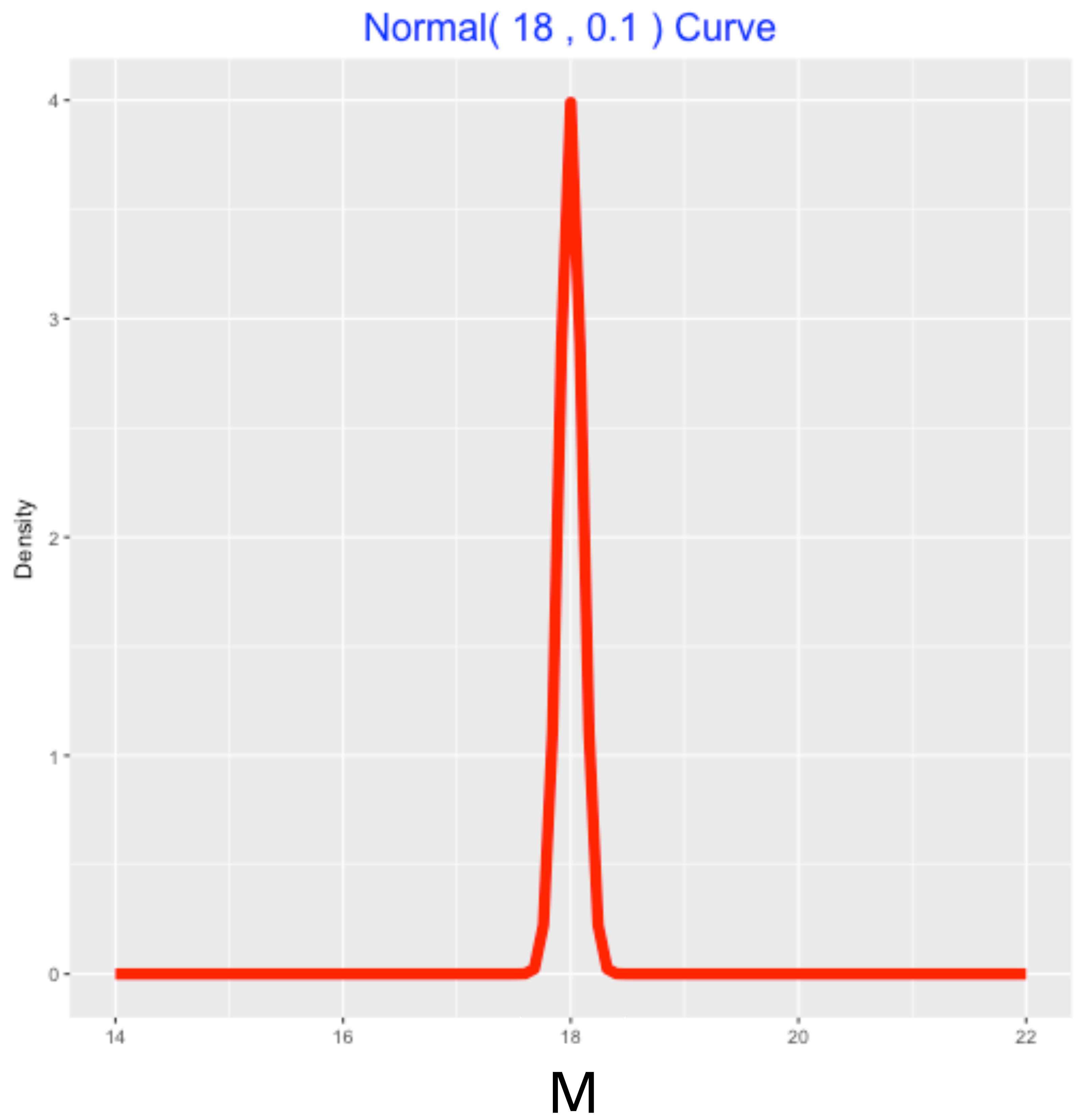


Prior beliefs about M

- Previously, assumed M was discrete
- Now, assume M is continuous
- Represent prior by a Normal(M_o, S_o) curve
  - M_o - best guess at M
  - S_o - standard deviation indicating how sure you are about your guess

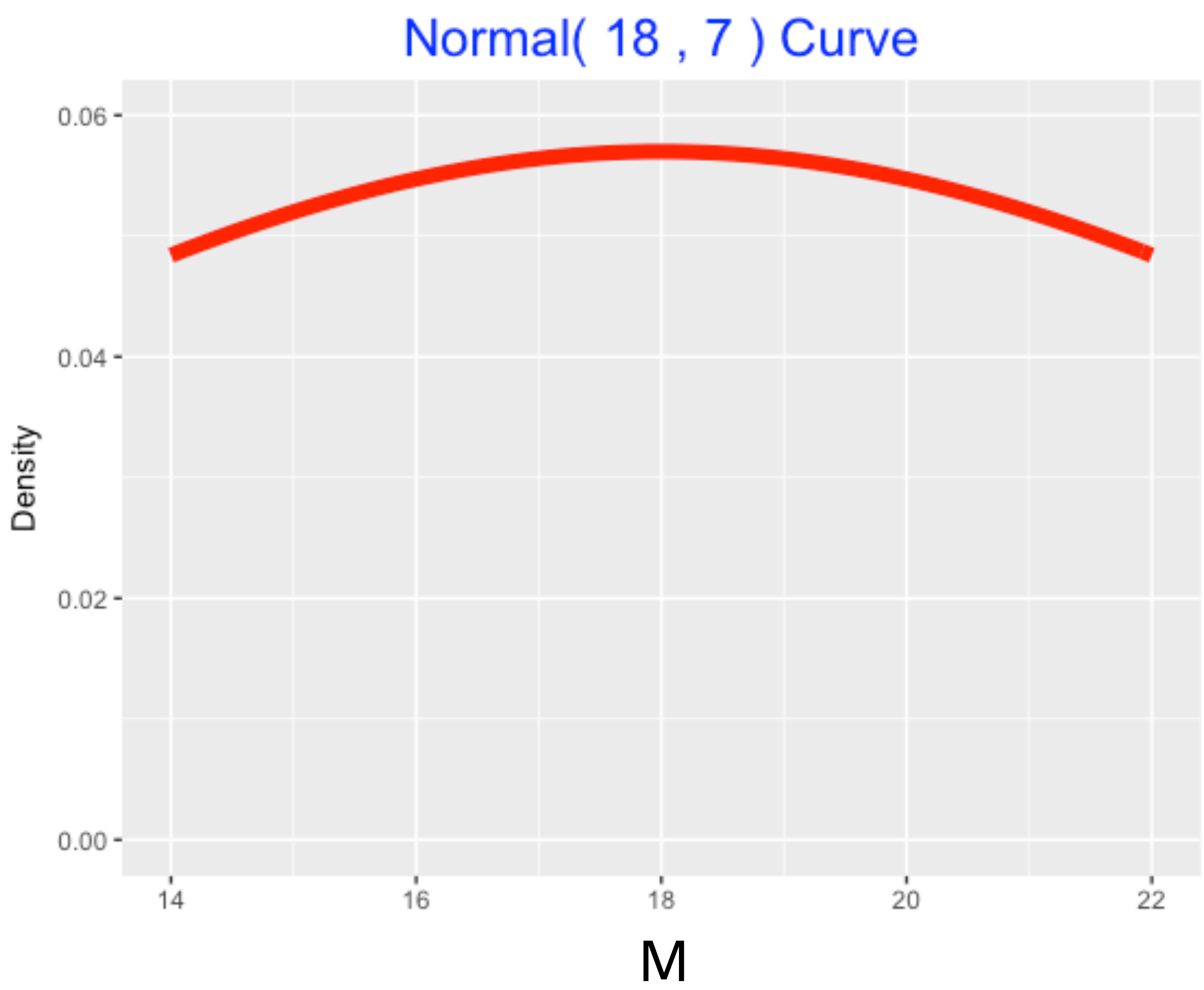
Joe's prior

- **Joe**: *"I believe strongly M is close to 18 seconds."*
- He uses $M_0 = 18$ and a small value for S_0



Sue's prior

- **Sue**: *"Maybe M is close to 18 seconds, but I really don't know much about serving times in tennis."*
- Uses $M_0 = 18$ but chooses a large value for S_0



Normal prior by 2 quantiles

- Remember a quantile is a value of M such that the chance of being smaller than the value is a given probability
- Think about 2 quantiles of M:
  - 0.50 quantile for M is 18 seconds
  - 0.90 quantile for M is 20 seconds

normal.select()

```
> library(TeachBayes)
> normal.select(list(x = 18, p = 0.5),
                list(x = 20, p = 0.9))
$mu
[1] 18

$sigma
[1] 1.560608
```

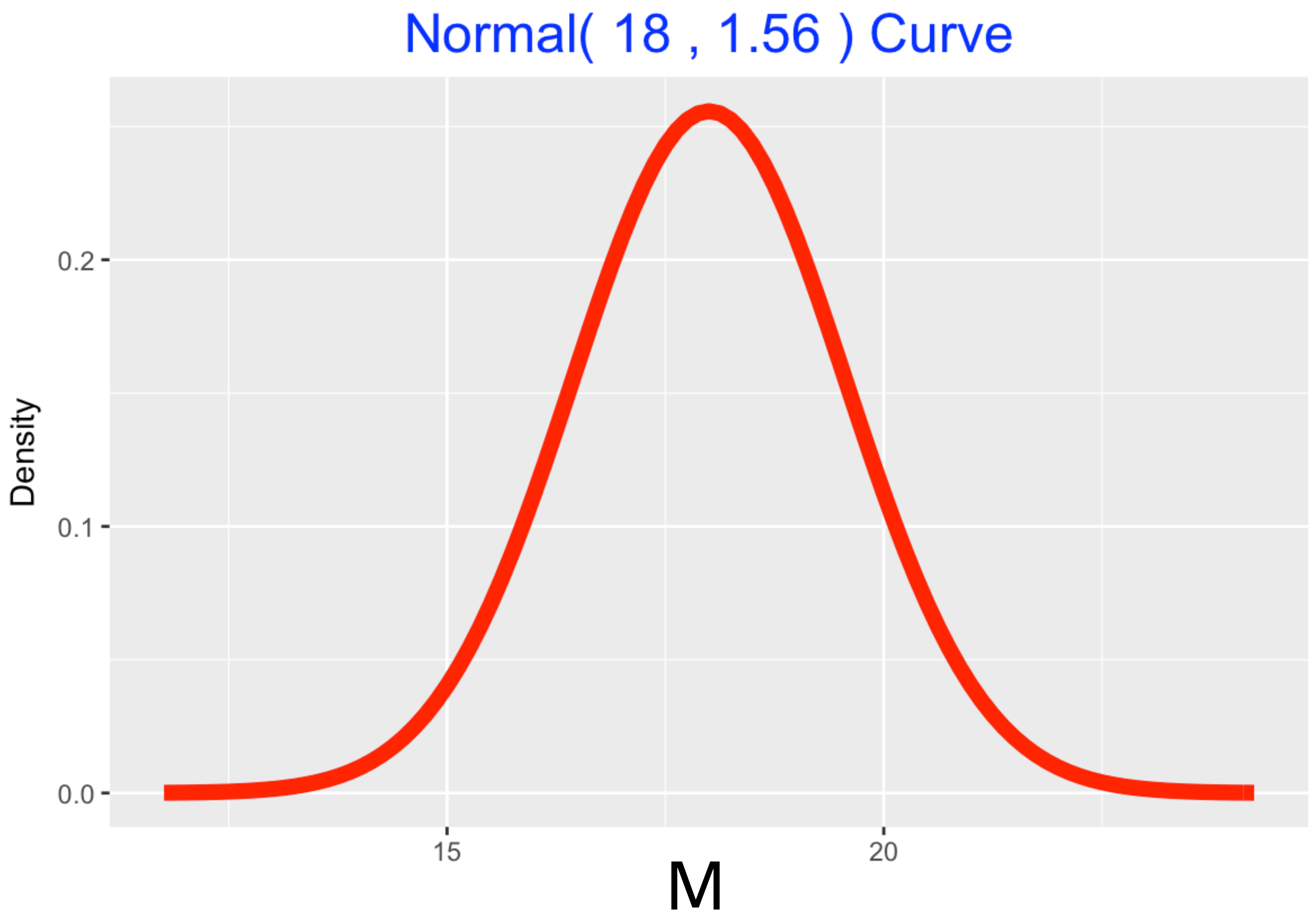
Input quantiles of normal prior

Mean

Standard deviation

My prior for M

```
> library(TeachBayes)
> normal_draw(c(18, 1.56))
```

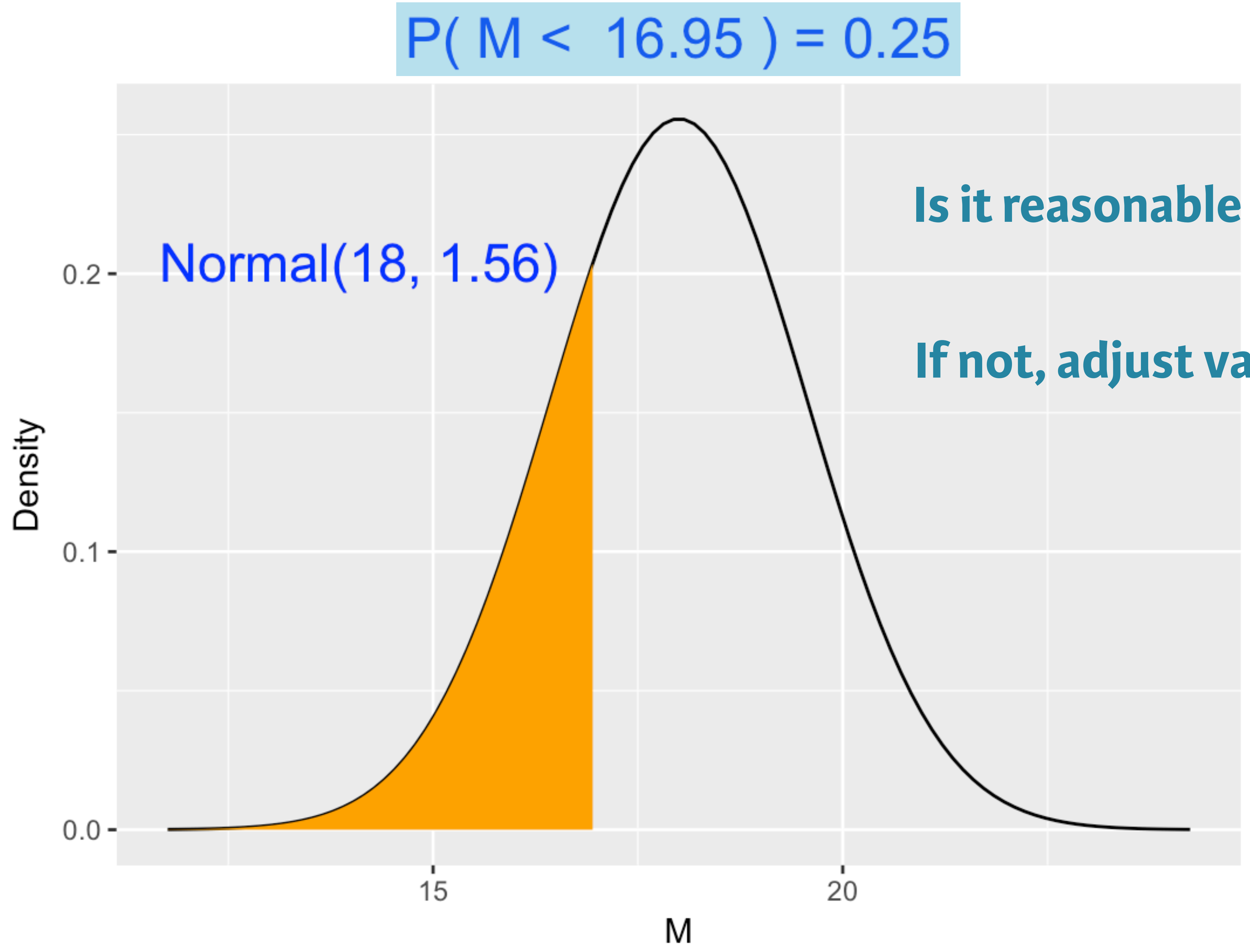


Any prior is an approximation to my opinion

- Prior is approximate
- Check if prior makes sense:
  - `normal_area()`
  - `normal_percentile()`
  - `normal_interval()`

Find 0.25 quantile of my prior

```
> library(TeachBayes)
> normal_quantile(0.25, c(18, 1.56))
```



Is it reasonable that $P(M < 16.95) = 0.25$?

If not, adjust values M_0 and S_0

Let's practice!

Updating the normal prior

Learning about time-to-serve

- Interested in how long Roger Federer takes to serve
- My prior for M is Normal(18, 1.56) (unit = seconds)

Observe data

- Record the time-to-serve for 20 of Roger's serves
  - Compute the mean value: $\bar{y} = 17.2$
  - Associated standard error: $se = \frac{S}{\sqrt{n}} = \frac{4}{\sqrt{20}} = 0.89$

```
> # Store likelihood in bayes_df:
> bayes_df$Likelihood <- dnorm(17.2, mean = bayes_df$M, sd = 0.89)
```

Posterior of M?

- Posterior = Prior x Likelihood
- Posterior = Normal(M, 18, 1.56) x Normal(17.2, M, 0.89)
- Posterior curve turns out to be Normal!

Finding posterior distribution

$$Precision = \frac{1}{SD^2}$$

Useful for combining information in the prior and the data

Source	Mean	Precision	Stand_Dev
Prior			
Data			
Posterior			

Step 1: Input known information

Source	Mean	Precision	Stand_Dev
Prior	18		1.56
Data	17.2		0.89
Posterior			

Step 2: Compute precisions

```
> Precisions <- 1 / c(1.56, 0.89)^2
```

Source	Mean	Precision	Stand_Dev
Prior	18	0.41	1.56
Data	17.2	1.26	0.89
Posterior			

Step 3: Compute posterior precision and SD

```
> Post_Precision <- sum(Precisions)
> Post_SD <- 1 / sqrt(Post_Precision)
```

Source	Mean	Precision	Stand_Dev
Prior	18	0.41	1.56
Data	17.2	1.26	0.89
Posterior		1.67	0.77

Step 4: Compute posterior mean

```
> Post_mean <- weighted.mean(x = c(18, 17.2), w = c(0.41, 1.26))
```

Source	Mean	Precision	Stand_Dev
Prior	18	0.41	1.56
Data	17.2	1.26	0.89
Posterior	17.4	1.67	0.77

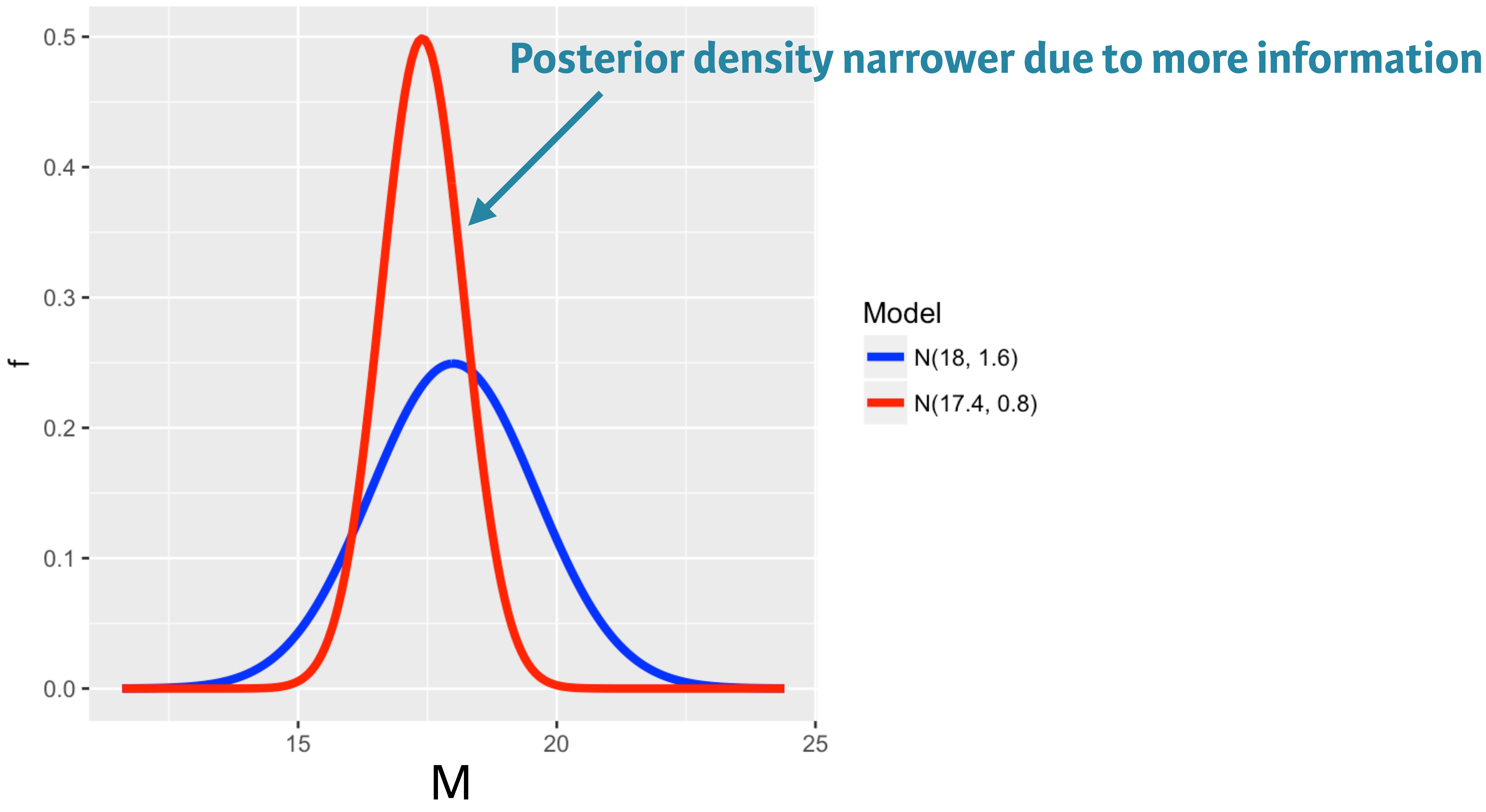
Use normal_update() function

- Input prior vector (mean, sd) and data vector (mean, sd)
- Outputs posterior mean and standard deviation

```
> library(TeachBayes)
> normal_update(c(18, 1.56), c(17.2, 0.89))
[1] 17.3964473  0.7730412

> normal_update(c(18, 1.56), c(17.2, 0.89), teach = TRUE)

       Type     Mean Precision Stand_Dev
1     Prior 18.00000 0.4109139 1.5600000
2      Data 17.20000 1.2624669 0.8900000
3 Posterior 17.39645 1.6733807 0.7730412
```

Compare prior and posterior curves for M

```
> library(TeachBayes)
> many_normal_plots(list(c(18, 1.56), c(17.4, 0.77)))
```



Testing problem

- Someone claims that Roger takes on average *at least* 19 seconds to serve
- Is this reasonable?

Classical approach

Test the hypothesis $H : M \geq 19\ seconds$

Test statistic: $Z = \frac{\bar{y} - 19}{se}$

```
> # Compute Z score
> (z <- (17.2 - 19) / 0.89)
[1] -2.022472

> # Compute p-value
> (p_value <- pnorm(z))
[1] 0.02156381
```

< 0.05 so reject H_0

Bayesian approach

Current belief is represented by posterior for M: Normal(17.4, 0.77)

```
> # Compute Prob(M >= 19)
> 1 - pnorm(19, 17.4, 0.77)
[1] 0.01885827
```

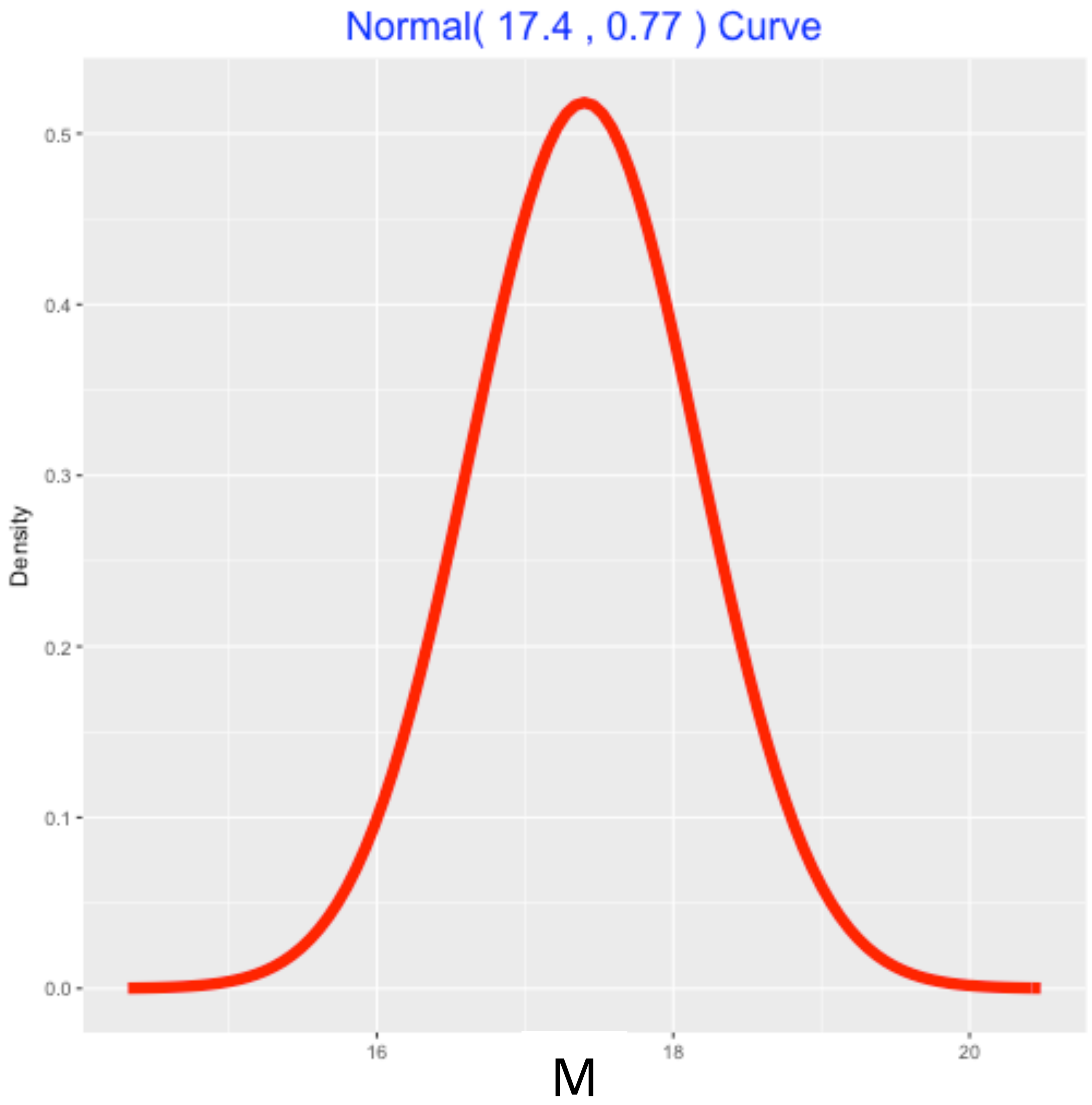
This is small, so conclude this claim is unlikely to be true

Let's practice!

Simulation

Learning about time-to-serve

- Interested in mean time M that Roger Federer takes to serve
- My posterior for M (mean serving time) is Normal(17.4, 0.77)



Posterior simulation

- One way to learn about M is by simulation
- Take a large sample from the posterior density
- Summarize the sample of simulated draws

```
> M_sim <- rnorm(1000, mean = 17.4, sd = 0.77)
```

Inference: A probability interval

- Suppose I want a 80% probability interval for M
- Take 0.10 and 0.90 sample quantiles of `M_sim`

```
> (Q <- round(quantile(M_sim, c(0.10, 0.90)), 1))
10%  90%
16.4 18.4
```

So Prob(16.4 < M < 18.4) = 0.80

Prediction

- Instead of learning about Roger's mean time-to-serve **M**, suppose I want to predict the time **y** that he takes on his next serve

- Interested in the **predictive density** of y, the distribution of times-to-serve **in the future**

Simulating predictive distribution

- Suppose we knew Roger's mean time-to-serve was exactly M = 17 seconds
- Then we could obtain 1000 (future) times-to-serve y by simulating from the sampling N(17, 4) density

```
> y_sim <- rnorm(1000, mean = 17, sd = 4)
```

- But we don't know the exact value of M

Add uncertainty about M

- We don't know the exact value of M
- But my current beliefs about M are summarized with the N(17.4, 0.77) posterior
- Use simulations from the posterior *as a first step* to obtain simulations of future time-to-serve

Idea

- To simulate one value from the predictive distribution:
  - Simulate a value M from N(17.4, 0.77) posterior
  - Simulate a value y from sampling density N(M, 4.0)

Illustration — step 1

Here's the first simulated M (and normal curve):

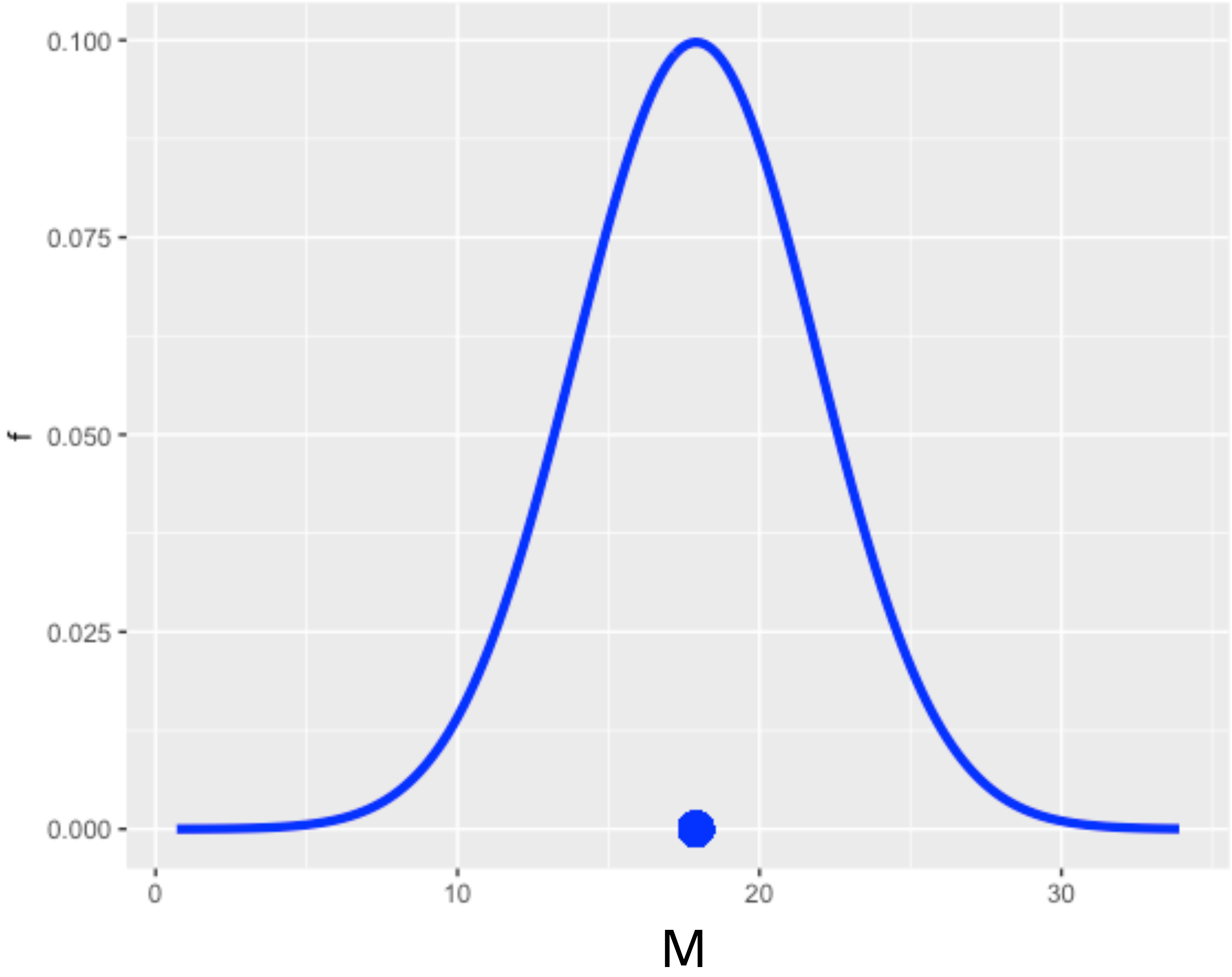


Illustration — step 2

Simulating another M:

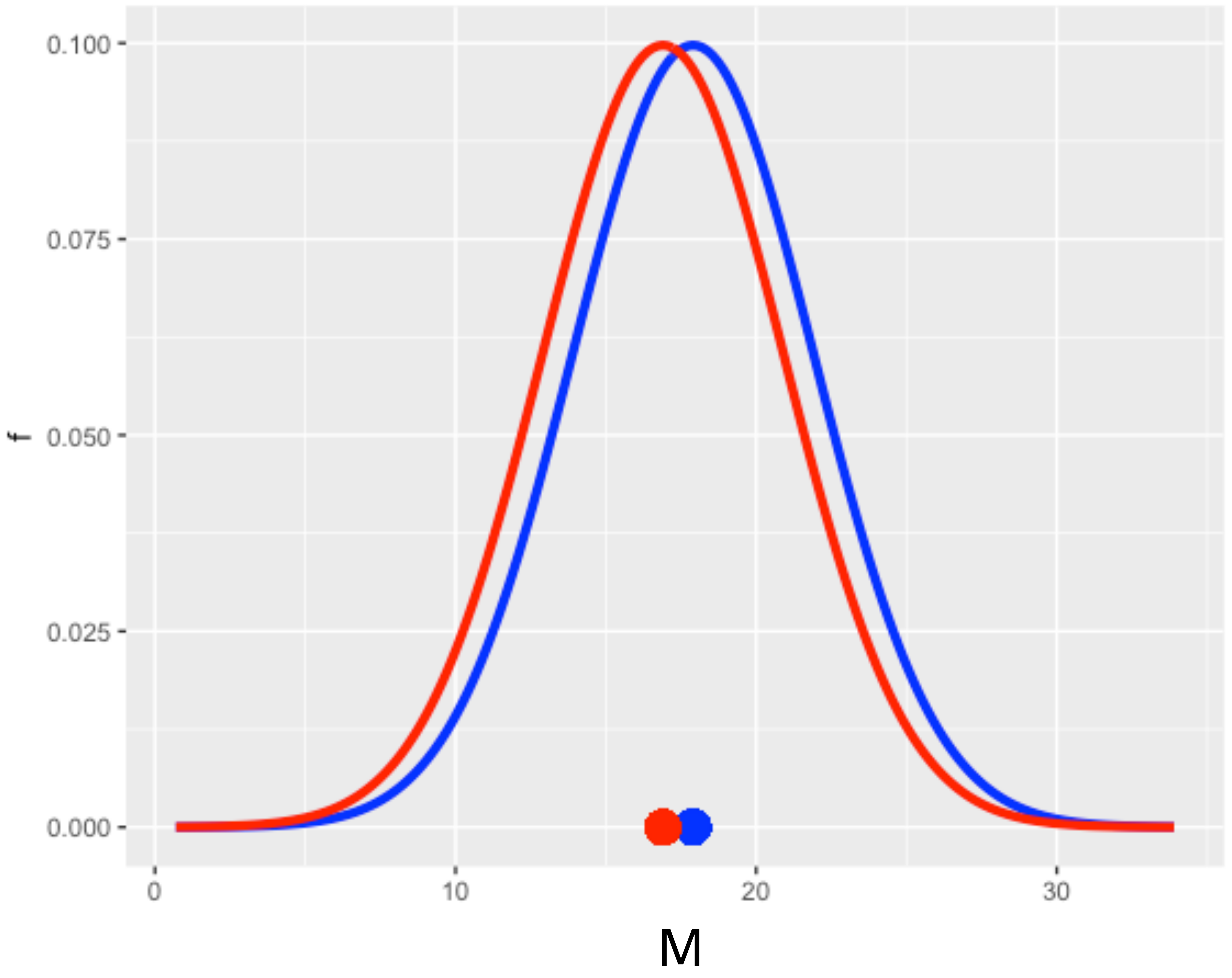


Illustration — step 3

Five simulated M's:

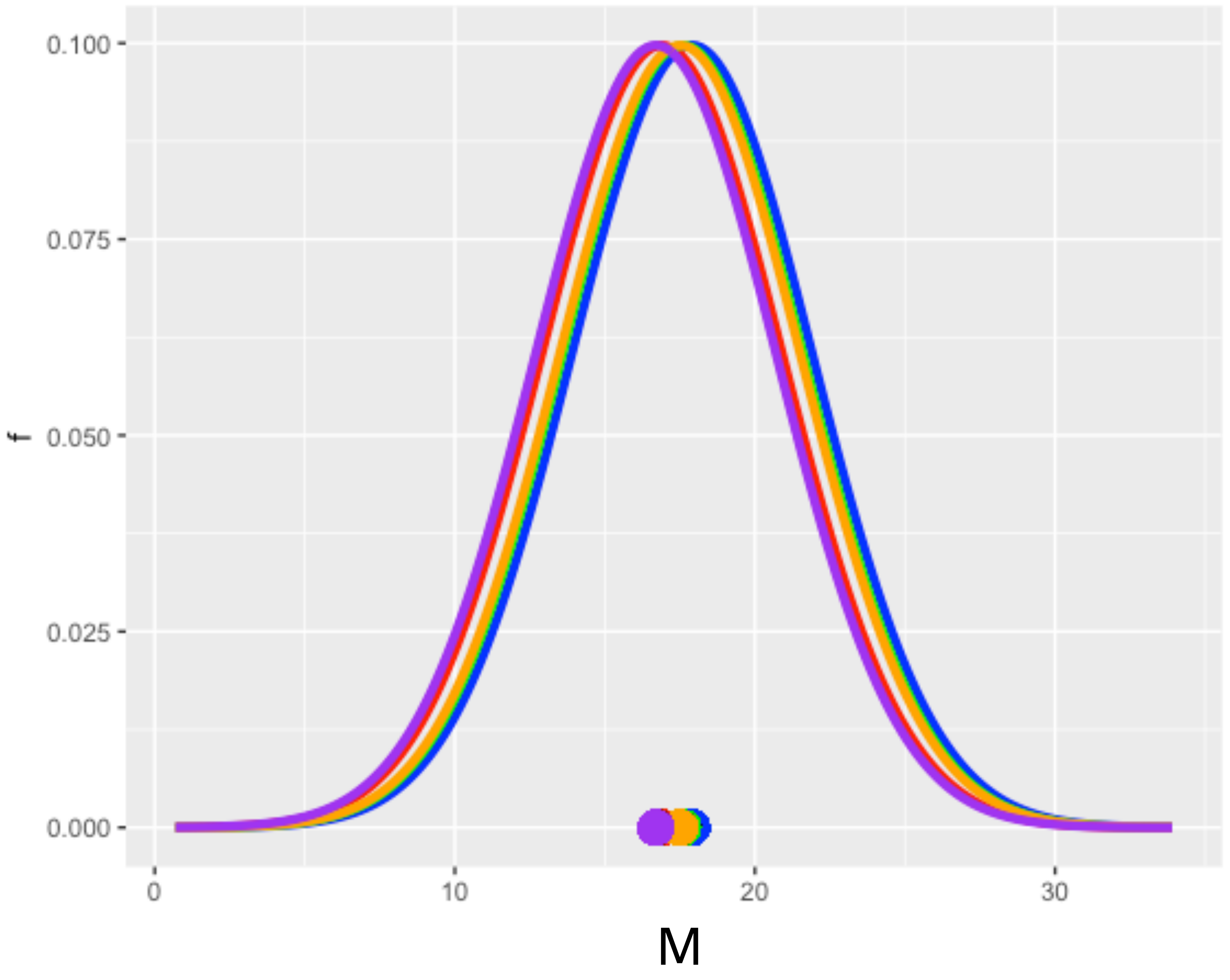
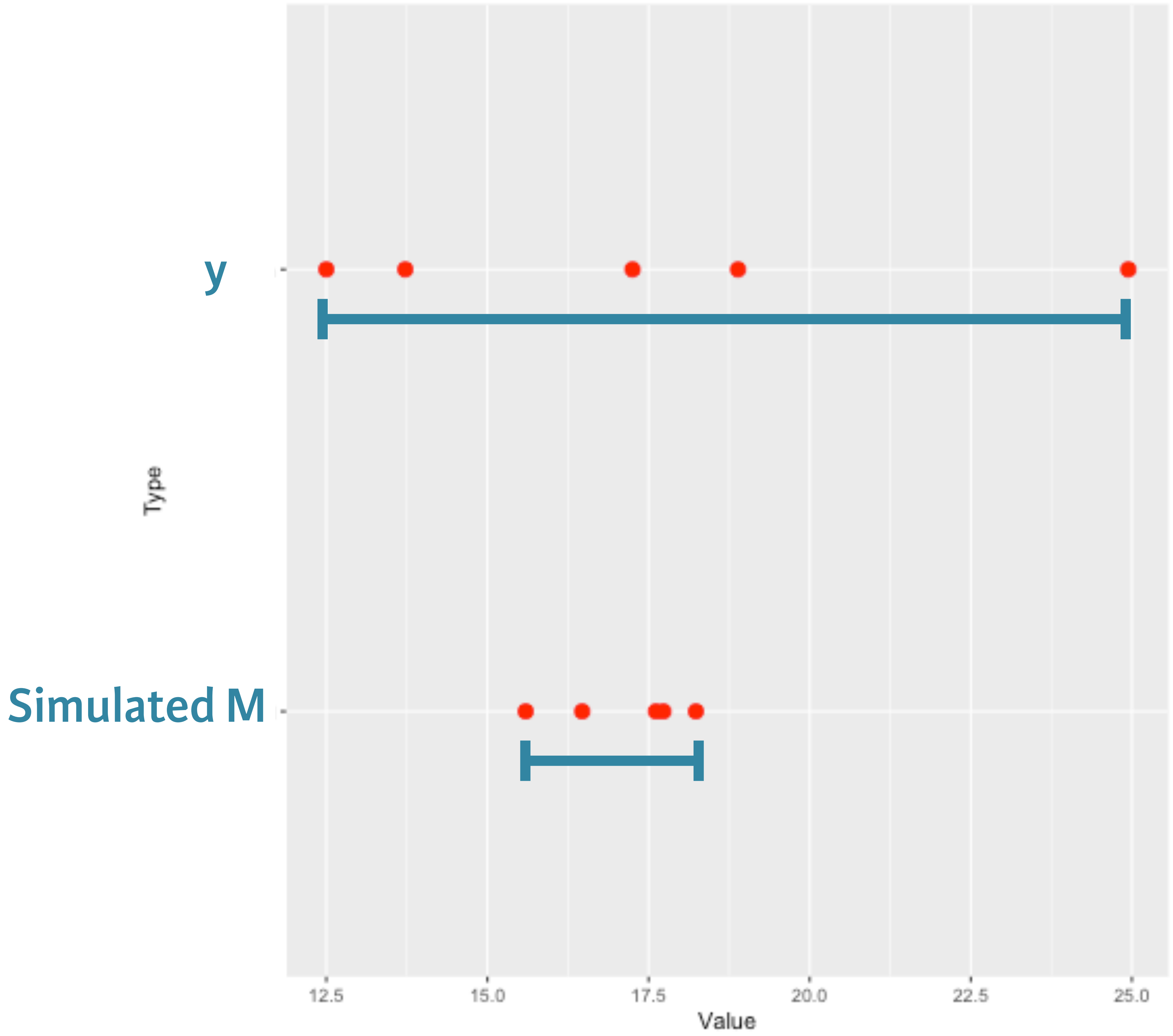


Illustration — step 4

For each Normal(M, S) curve, simulate y:



1000 values from predictive density

```
> # Simulate from posterior of M
> M_sim <- rnorm(1000, mean = 17.4, sd = 0.77)

> # Simulate from predictive density
> y_sim <- rnorm(1000, mean = M_sim, sd = 4)
```

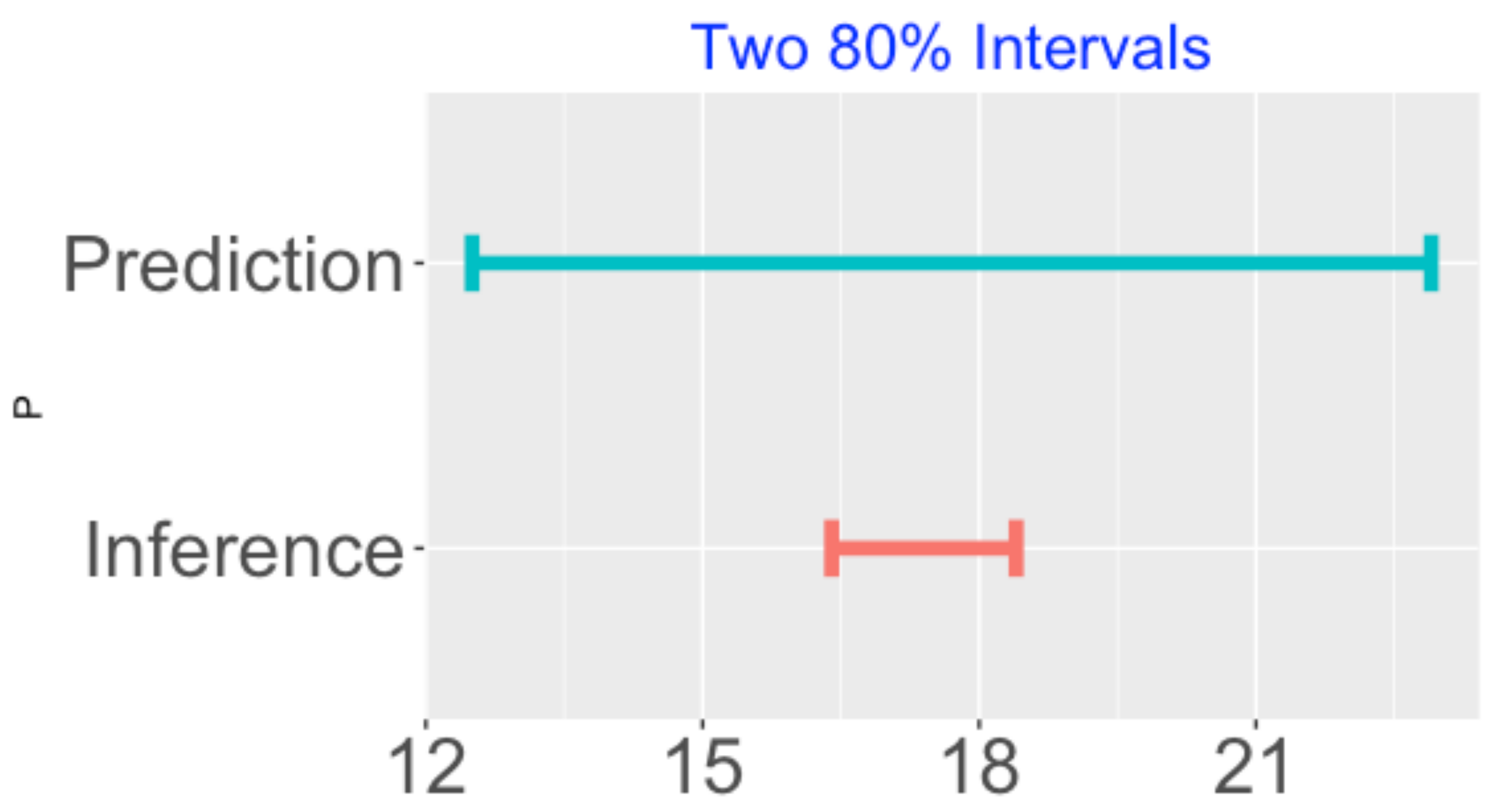
Summarize y_sim to predict future time-to-serve

80% predictive interval

```
> (Q <- round(quantile(y_sim, c(0.10, 0.90)), 1))
10%  90%
12.5 22.9
```

Compare two 80% intervals

- The predictive interval for y is much wider than the probability interval for M:



- Why? The prediction of y has two sources of uncertainty:
  - Don't know M (inference)
  - Don't know y given M (sampling)

Let's practice!