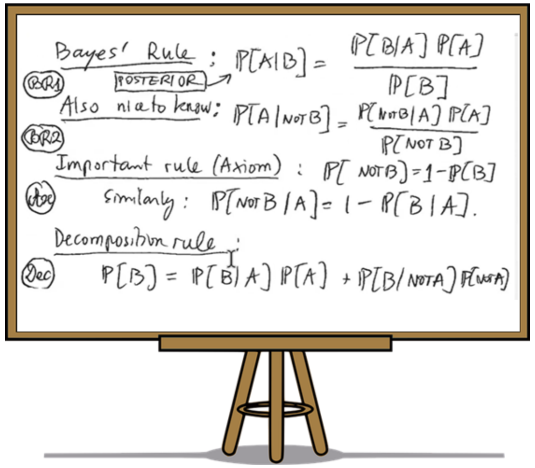
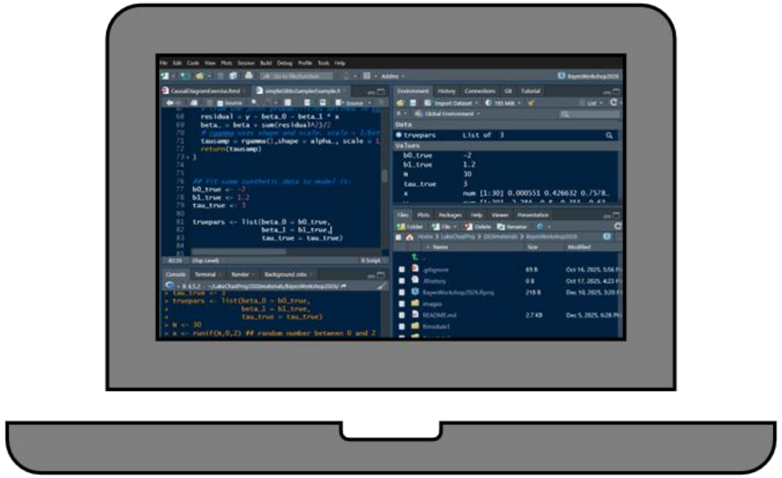
Bridging Theory and Practice Part 2: Intro to Bayesian computation

Katherine Muller



Theory

Bayes Rule!



Computation

Recap of Part 1

- Statistical inference involves estimating unknown or unobservable parameters from observable data (e.g., disease resistance from disease occurrence).

- Frequentist vs. Bayesian inference.

- Simple linear regression model for rainfed rice yield vs. rainfall.

- Without computer sampling, Bayesian inference can only be performed on **simple models** with **very restricted priors**.

Goals for Part 2

- Build your intuitive understanding about how Bayesian theory extends to Bayesian computation

- Gain familiarity with the basics of Bayesian computation using simple linear regression model

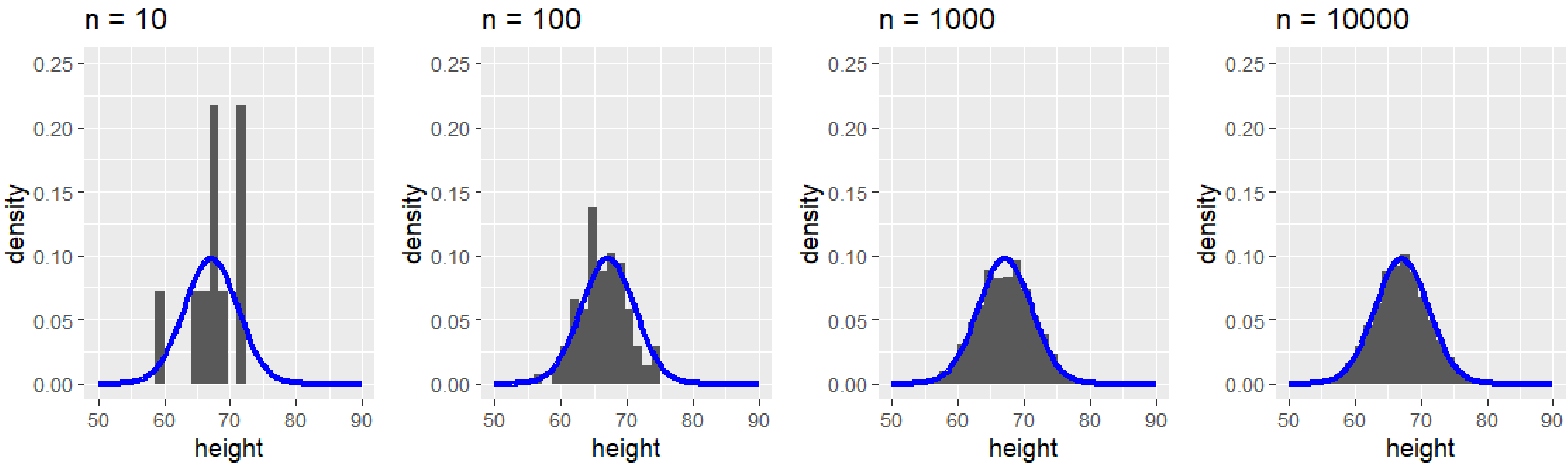
Computers turn an analytical task into a sampling task

$$\text{height} \sim \mathcal{N}(\mu, \sigma)$$

a.k.a. Law of Large Numbers

```
rnorm(n, mean, sd)
```

More samples → closer to theoretical distribution



Markov chain Monte Carlo (MCMC)

- Family of computer algorithms used for Bayesian data analysis
- Approximate theoretical distributions through iterative sampling
- Gibbs sampling is one type of MCMC algorithm

Building our model

1. Specify the likelihood

2. Specify priors

3. Sample posterior

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

Building our model

1. Specify the likelihood

2. Specify priors

3. Sample posterior

$$P(\theta|D) = \frac{\boxed{P(D|\theta)}P(\theta)}{P(D)}$$

Choose a **likelihood function** that **represents our data-generating process**

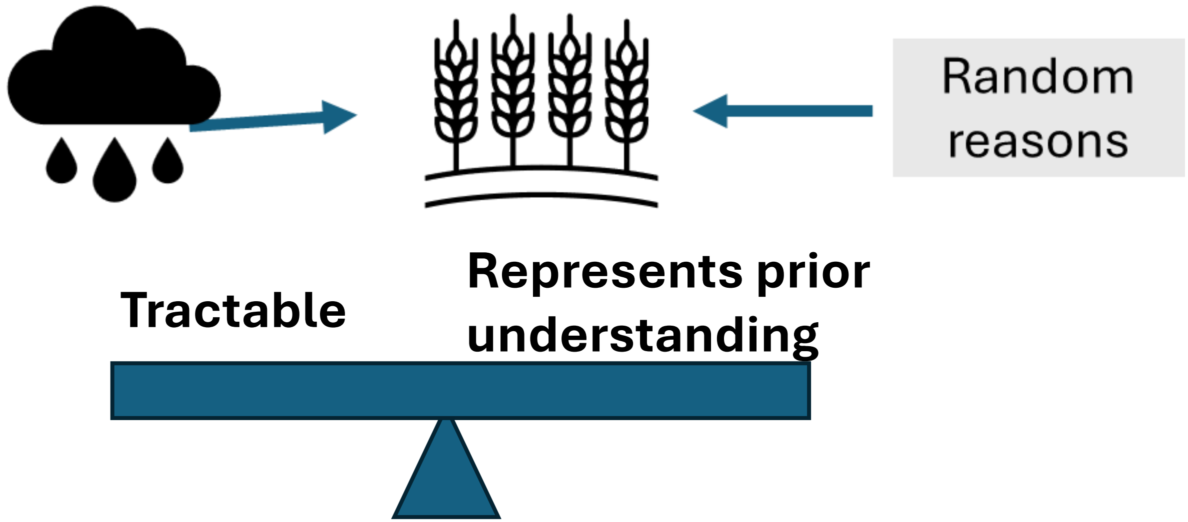
$$y_i \mid x_i, \beta_0, \beta_1, \tau \sim \mathcal{N}(\beta_0 + \beta_1 x_i, 1/\tau)$$

*Yield data (**y**) are generated on a normal distribution with the mean influenced by rainfall (**x**).*

*We can simulate **y** by sampling a normal distribution .*

```
b0  <- 0; b1  <- 1.2; tau <- 3 #”true” parameters
N <- 30 # number of observations of x and y
x <- runif(N,-1,-1) ## random number between -1 and 1
y <- rnorm(N, mean = b0 + b1*x, sd = 1/sqrt(tau))
```

Building our model



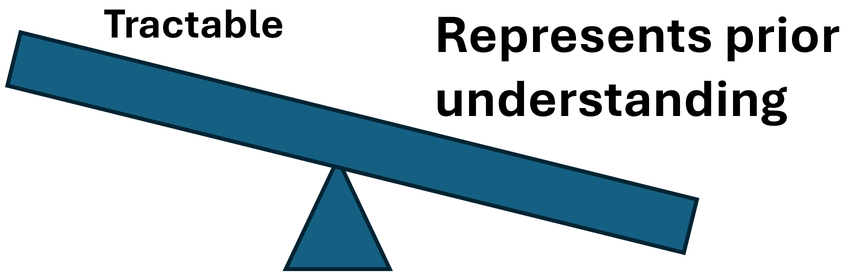
1. Specify the likelihood

2. Specify priors

3. Sample posterior

$$P(\theta|D) = \frac{P(D|\theta)\boxed{P(\theta)}}{P(D)}$$

Building our model



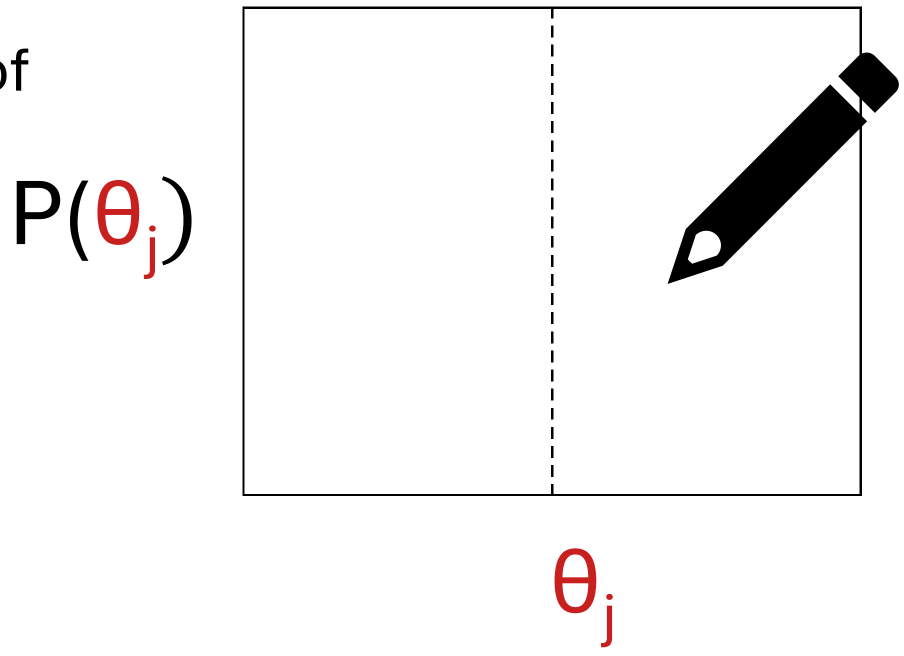
1. Specify the likelihood

2. Specify priors

3. Sample posterior

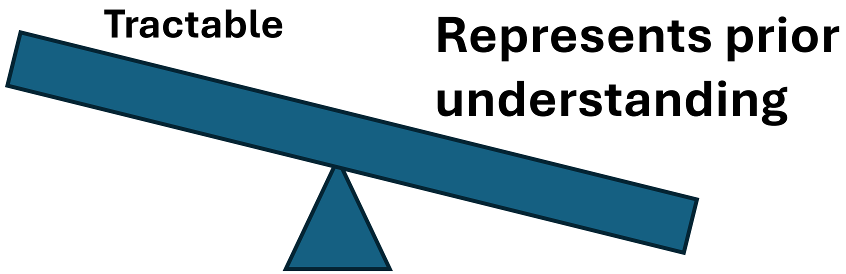
$$P(\boldsymbol{\theta}|\boldsymbol{D}) = \frac{P(\boldsymbol{D}\,|\boldsymbol{\theta})\boxed{P(\boldsymbol{\theta})}}{P(\boldsymbol{D})}$$

Plausibility of parameter value



parameter value

Building our model

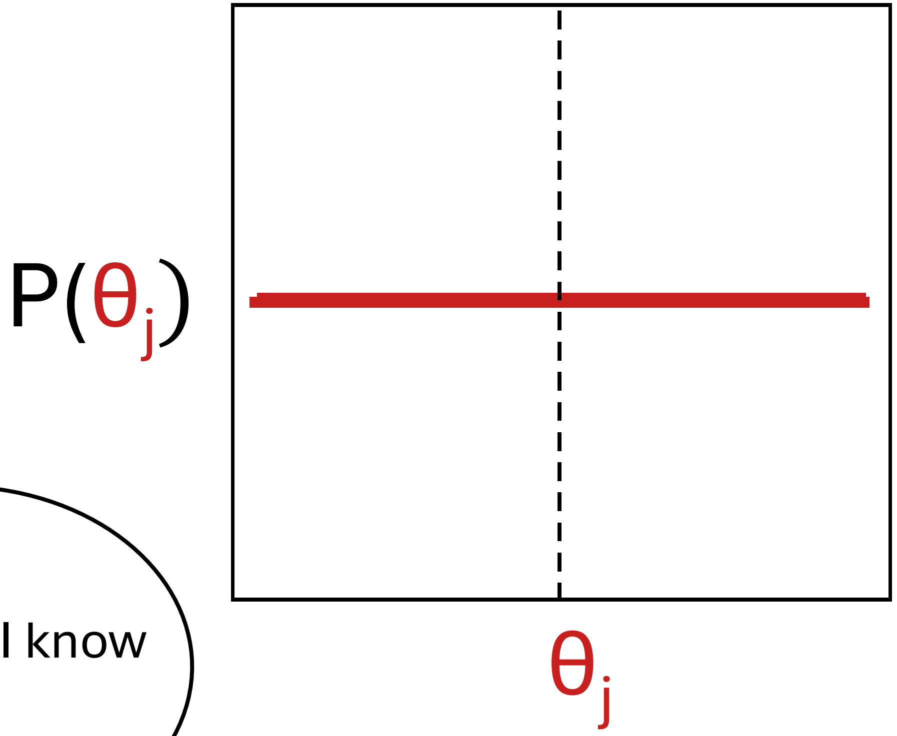


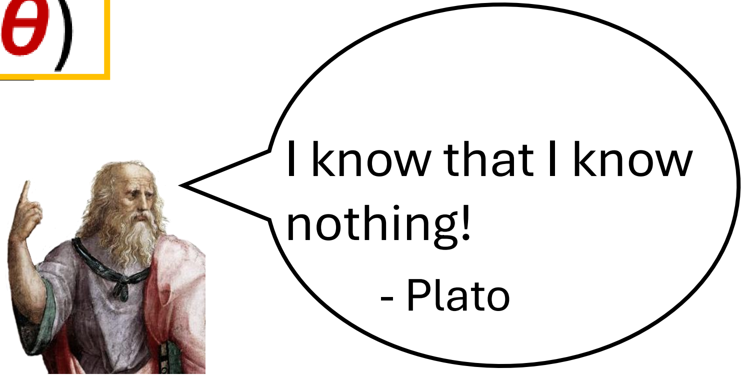
1. Specify the likelihood

2. Specify priors

3. Sample posterior

$$P(\boldsymbol{\theta}|\boldsymbol{D}) = \frac{P(\boldsymbol{D}\,|\boldsymbol{\theta})\boxed{P(\boldsymbol{\theta})}}{P(\boldsymbol{D})}$$





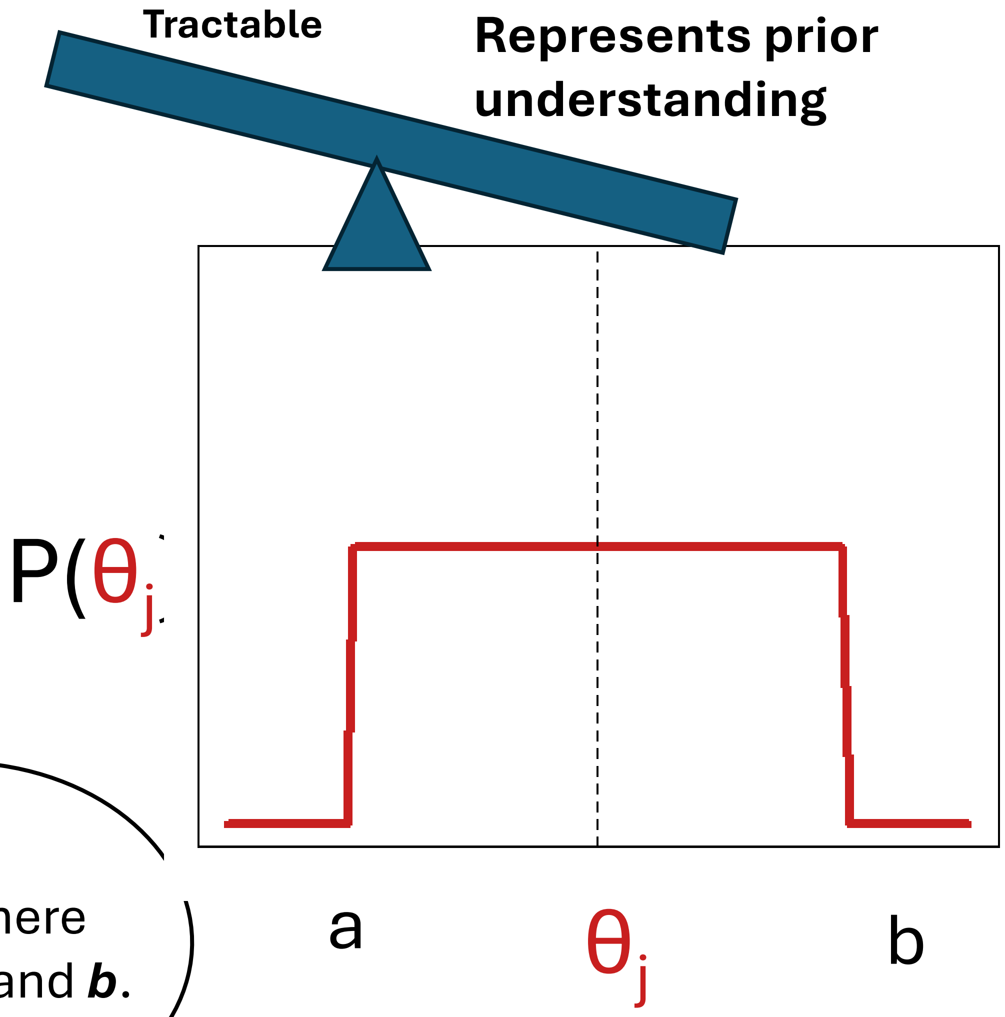
Building our model

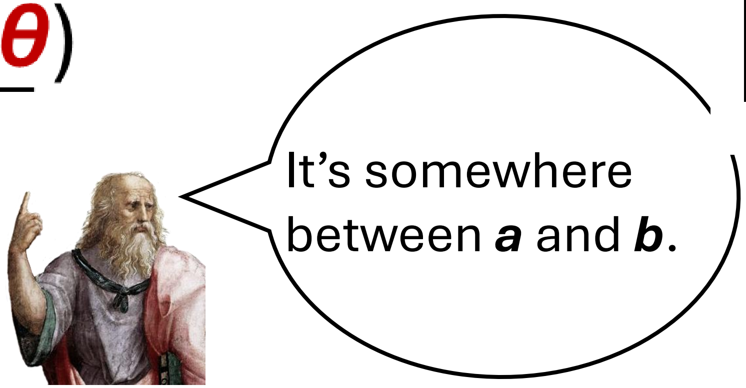
1. Specify the likelihood

2. Specify priors

3. Sample posterior

$$P(\boldsymbol{\theta}|\boldsymbol{D}) = \frac{P(\boldsymbol{D}\,|\boldsymbol{\theta})P(\boldsymbol{\theta})}{P(\boldsymbol{D})}$$





Building our model

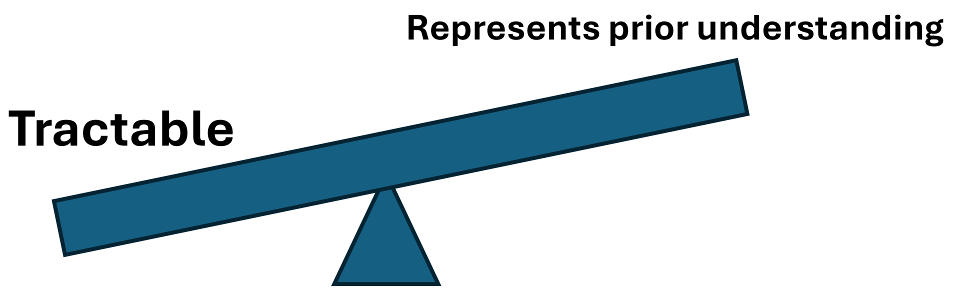
Data generating function

1. Specify the likelihood

2. Specify priors

3. Sample posterior

$$P(\boldsymbol{\theta}|\boldsymbol{D}) = \frac{P(\boldsymbol{D}\,|\boldsymbol{\theta})\boxed{P(\boldsymbol{\theta})}}{P(\boldsymbol{D})}$$



Choose prior functions that will combine with the likelihood function to **create posteriors we can easily sample**.

Parameter generating functions

$P(\boldsymbol{\theta})$	$P(\boldsymbol{\theta}\|\boldsymbol{D})$
$\theta_j \mid \theta_j^* \sim \underbrace{f(\theta_j^*)}$	$\theta_j \mid \theta_j^*, \boldsymbol{D} \sim \underbrace{f(\theta_j^*, \boldsymbol{D})}$

Same type of probability distribution

$\theta_j \sim \mathcal{N}(\mu_j, \tau_j^{-1})$ $\qquad$ $\theta_j \mid \boldsymbol{D} \sim \mathcal{N}(\mu^*_j, \tau^*_j{}^{-1})$

Conditionally conjugate prior

Conditionally conjugate priors for simple linear regression

Prior and conditional posterior distributions have the same form

P(**θ**)

$\beta_0 | \mu_0, \tau_0 \sim \mathcal{N}(\mu_0, {\tau_0}^{-1})$

$\beta_1 | \mu_1, \tau_1 \sim \mathcal{N}(\mu_1, {\tau_1}^{-1})$

$\tau \; | a, b \sim \mathrm{Gamma}(a, b)$

P(***D*** | **θ**)

$$p(\boldsymbol{y} | \boldsymbol{x}, \beta_0, \beta_1, \tau) = \prod_{\mathrm{i}=1}^{\mathrm{N}} \mathcal{N}(\beta_0 + \beta_1 \mathrm{x_i}, 1/\tau)$$

P(**θ**| ***D***)

$\beta_0 | \; \beta_1, \tau, \boldsymbol{y}, \boldsymbol{x} \sim \mathcal{N}({\mu_0}^*, ({\tau_0}^*)^{-1})$

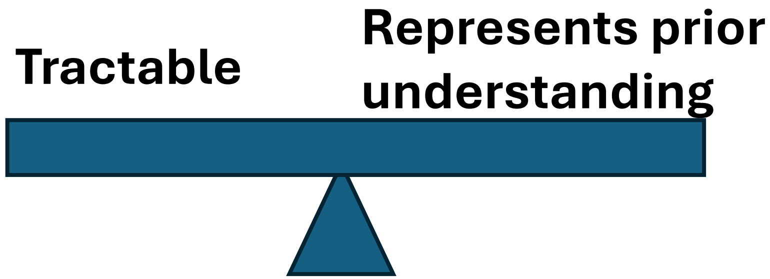
$\beta_1 | \; \beta_0, \tau, \boldsymbol{y}, \boldsymbol{x} \sim \mathcal{N}({\mu_1}^*, ({\tau_1}^*)^{-1})$

$\tau | \beta_0, \beta_1, \boldsymbol{y}, \boldsymbol{x} \sim \mathrm{Gamma}(\mathrm{a}^*, b^*)$

- Intuitive for Bayesian updating
- Can speed up processing
- Good default if you don't have strong prior understanding

- May not adequately model for prior understanding in all situations
- Not required for Bayesian computation

Building our model



1. Specify the likelihood

2. Specify priors

3. Sample posterior

$$P(\boldsymbol{\theta}|\boldsymbol{D}) = \frac{P(\boldsymbol{D}\,|\boldsymbol{\theta})\boxed{P(\boldsymbol{\theta})}}{P(\boldsymbol{D})}$$

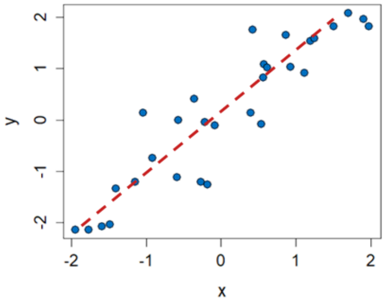
Use **tractable priors** (e.g., conditionally conjugate) and adjust **hyperparameters** to **reflect prior understanding**.

$$P(\boldsymbol{\theta}|\boldsymbol{D}) = \frac{P(\boldsymbol{D}|\boldsymbol{\theta})\boxed{P(\boldsymbol{\theta})}}{P(\boldsymbol{D})}$$

Simple linear regression:

$$\boldsymbol{y} = \beta_0 + \beta_1 \boldsymbol{x} + \boldsymbol{\varepsilon}$$

$$\boldsymbol{\varepsilon} \sim \mathcal{N}(0, 1/\tau)$$

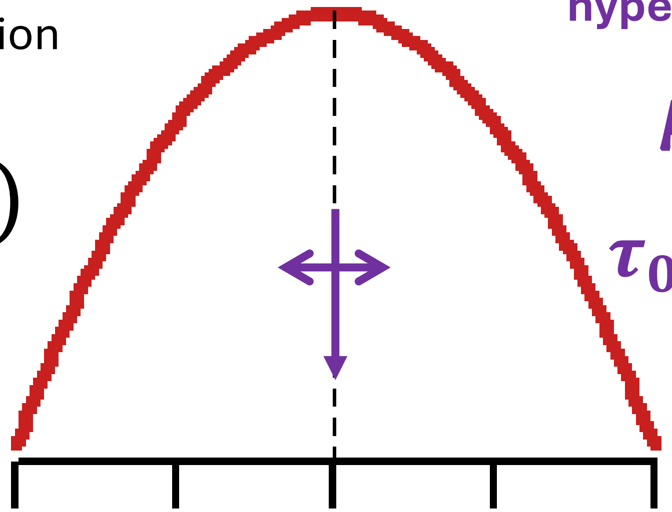


Intercept

Since we standardized **x** and **y**, the intercept is probably close to zero

Conjugate → Normal distribution

$P(\beta_0)$



β_0

hyperparameters

$$\boldsymbol{\mu_0} = \boldsymbol{0}$$

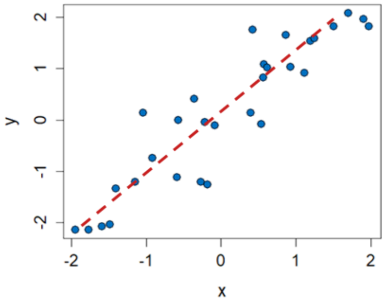
$$\boldsymbol{\tau_0}^{-1} =$$ *Some number expressing our uncertainty*

$$P(\boldsymbol{\theta}|\boldsymbol{D}) = \frac{P(\boldsymbol{D}|\boldsymbol{\theta})\boxed{P(\boldsymbol{\theta})}}{P(\boldsymbol{D})}$$

Simple linear regression:

$$\boldsymbol{y} = \beta_0 + \beta_1 \boldsymbol{x} + \boldsymbol{\varepsilon}$$

$$\boldsymbol{\varepsilon} \sim \mathcal{N}(0, 1/\tau)$$

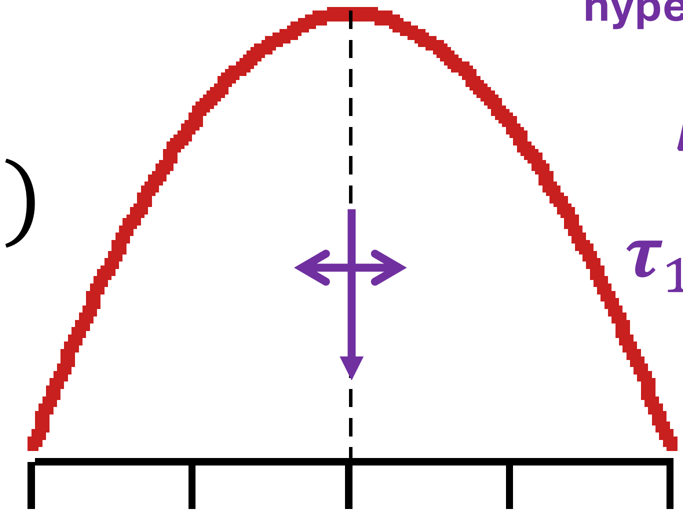


Slope

The slope of **x** vs. **y** could be negative, positive, or zero.

Conjugate → Normal distribution

$P(\beta_1)$



β_1

hyperparameters

$$\boldsymbol{\mu_1} = \mathbf{0}$$

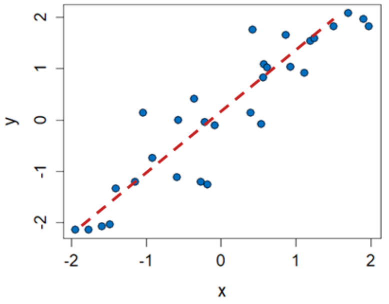
$$\boldsymbol{\tau_1}^{-1} = $$ *Some number expressing our uncertainty*

$$P(\boldsymbol{\theta}|\boldsymbol{D}) = \frac{P(\boldsymbol{D}\,|\boldsymbol{\theta})\boxed{P(\boldsymbol{\theta})}}{P(\boldsymbol{D})}$$

Simple linear regression:

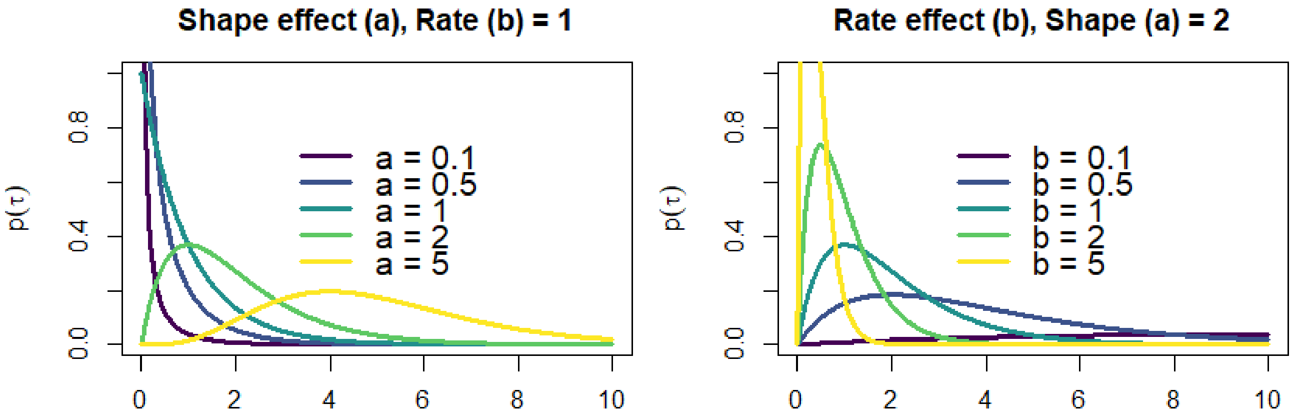
$$\boldsymbol{y} = \beta_0 + \beta_1 \boldsymbol{x} + \boldsymbol{\varepsilon}$$

$$\boldsymbol{\varepsilon} \sim \mathcal{N}(0, 1/\tau)$$

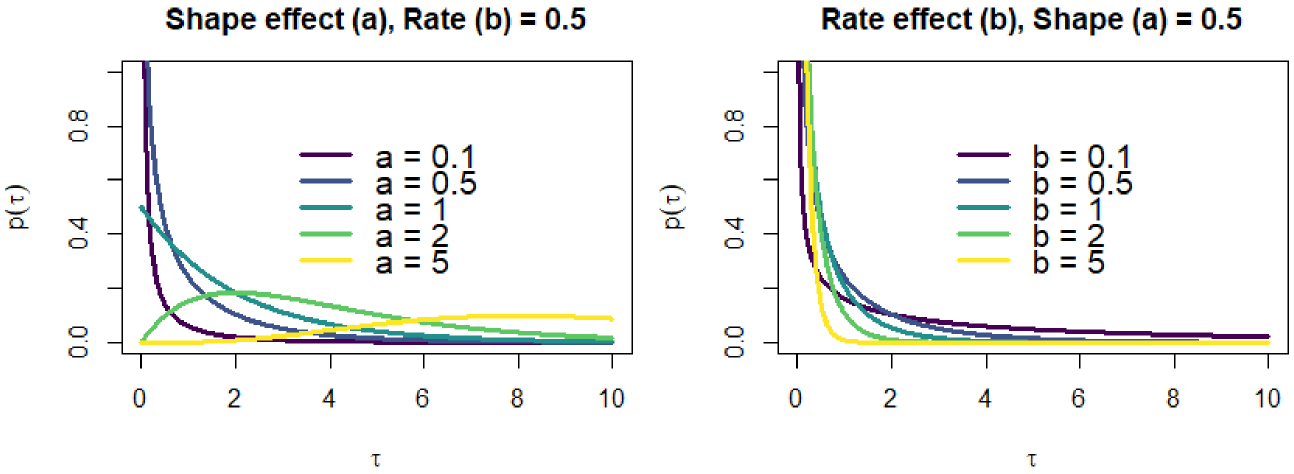


Precision

Precision is a number greater than zero. Smaller precision means bigger variance.



Tricky—hyperparameter choices affect posterior sampling



Conjugate → Gamma distribution

a = shape

b = rate

Building our model

$$\boldsymbol{y} = \beta_0 + \beta_1 \boldsymbol{x} + \boldsymbol{\varepsilon}$$
$$\boldsymbol{\varepsilon} \sim \mathcal{N}(0, 1/\tau)$$

1. Specify the likelihood
2. Specify priors
3. Sample posterior

$$\boxed{P(\boldsymbol{\theta}|\boldsymbol{D})} = \frac{P(\boldsymbol{D}\,|\boldsymbol{\theta})P(\boldsymbol{\theta})}{P(\boldsymbol{D})}$$

Normalizing constant

Sample **one parameter at a time** conditional on the **data** and **other parameters**

Full conditional posterior

	Likelihood	Prior
$p(\beta_0 \mid \beta_1, \tau, \boldsymbol{y}, \boldsymbol{x}) \propto$	$p(\boldsymbol{y}, \boldsymbol{x} \mid \beta_0, \beta_1, \tau)$	$p(\beta_0)$
$p(\beta_1 \mid \beta_0, \tau, \boldsymbol{y}, \boldsymbol{x}) \propto$	$p(\boldsymbol{y}, \boldsymbol{x} \mid \beta_0, \beta_1, \tau)$	$p(\beta_1)$
$p(\tau \mid \beta_0, \beta_1, \boldsymbol{y}, \boldsymbol{x}) \propto$	$p(\boldsymbol{y}, \boldsymbol{x} \mid \beta_0, \beta_1, \tau)$	$p(\tau)$

Gibbs sampling!

Gibbs sampling

- A **computer algorithm** used to **approximate the full posterior distribution** of one or more parameters

- Works by **iteratively sampling** the **full conditional distributions** of each parameter

- One member of the broader family of **Markov Chain Monte Carlo (MCMC)** methods

To build our Gibbs sampler, we need...

1. Data on rainfall and rice yield.

2. A data-generating model

$$\boldsymbol{y} = \beta_0 + \beta_1 \boldsymbol{x} + \boldsymbol{\varepsilon}$$
$$\boldsymbol{\varepsilon} \sim \mathcal{N}(0, 1/\tau)$$

3. A likelihood function representing our data-generating model

$$p(\boldsymbol{y}, \boldsymbol{x} | \beta_0, \beta_1, \tau) = \prod_{i=1}^{N} \mathcal{N}(\beta_0 + \beta_1 x_i, 1/\tau)$$

4. Prior distributions and hyperparameters for each parameter

intercept $p(\beta_0) = \mathcal{N}(\mu_0, {\tau_0}^{-1})$

slope $p(\beta_1) = \mathcal{N}(\mu_1, {\tau_1}^{-1})$

precision $p(\tau) = \mathrm{Gamma}(a, b)$

Hyperparameters $\mu_0 = 0$ $\tau_0 = 1$
$\mu_1 = 0$ $\tau_1 = 1$
$a = 2$ $b = 1$

5. Full conditional posterior distributions for each parameter → **Calculate using Bayes' Rule!**

Full conditional posteriors via Bayes' Rule

Intercept β_0

Likelihood

Prior

$$p(\beta_0 | \beta_1, \tau, \boldsymbol{y}, \boldsymbol{x}) \propto p(\boldsymbol{y}, \boldsymbol{x} | \beta_0, \beta_1, \tau)\ p(\beta_0)$$

$$= \prod_{i=1}^{N} \mathcal{N}(\beta_0 + \beta_1 x_i, 1/\tau)\ \mathcal{N}(\mu_0, {\tau_0}^{-1})$$

$$= \prod_{i=1}^{N} \sqrt{\frac{\tau}{2\pi}}\ e^{-\frac{\tau}{2}(y_i - (\beta_0 + \beta_1 x_i)^2} \sqrt{\frac{\tau_0}{2\pi}}\ e^{-\frac{\tau_0}{2}(\beta_0 - \mu_0)^2}$$

*Some clever rearranging **

$$\beta_0 | \beta_1, \tau, \boldsymbol{y}, \boldsymbol{x} \sim \mathcal{N}\left(\frac{\tau_0 \mu_0 + \tau \sum_{i=1}^{N}(y_i - \beta_1 x_i)}{\tau_0 + \tau N}, (\tau_0 + \tau)^{-1}\right)$$

Normal probability density function

$$z | \mu, \tau \sim \mathcal{N}(\mu, \tau^{-1})$$

$$p(z|\mu, \tau) = \mathcal{N}(\mu, \tau^{-1})$$

$$p(z|\mu, \tau) = \sqrt{\frac{\tau}{2\pi}}\ e^{-\frac{\tau}{2}(z - \mu)^2}$$

** Read this for a full breakdown of the steps* https://stmorse.github.io/journal/gibbs.html

Full conditional posteriors via Bayes' Rule

Slope β_1

$$p(\beta_1 | \beta_0, \tau, \boldsymbol{y}, \boldsymbol{x}) \propto p(\boldsymbol{y}, \boldsymbol{x} | \beta_0, \beta_1, \tau)\, p(\beta_1)$$

$$= \prod_{i=1}^{N} \mathcal{N}(\beta_0 + \beta_1 x_i, 1/\tau)\, \mathcal{N}(\mu_1, {\tau_1}^{-1})$$

$$\underbrace{= \prod_{i=1}^{N} \sqrt{\frac{\tau}{2\pi}}\, e^{-\frac{\tau}{2}(y_i - (\beta_0 + \beta_1 x_i))^2} \sqrt{\frac{\tau_1}{2\pi}}\, e^{-\frac{\tau_1}{2}(\beta_1 - \mu_1)^2}}_{\text{Some clever rearranging *}}$$

$$\beta_1 | \beta_0, \tau, \boldsymbol{y}, \boldsymbol{x} \sim \mathcal{N}\left(\frac{\tau_1 \mu_1 + \tau \sum_{i=1}^{N} (y_i - \beta_0) x_i}{\tau_1 + \tau \sum_{i=1}^{N} x_i^2}, \left(\tau_1 + \tau \sum_{i=1}^{N} x_i^2\right)^{-1}\right)$$

Normal probability density function

$$z | \mu, \tau \sim \mathcal{N}(\mu, \tau^{-1})$$

$$p(z | \mu, \tau) = \mathcal{N}(\mu, \tau^{-1})$$

$$p(z | \mu, \tau) = \sqrt{\frac{\tau}{2\pi}}\, e^{-\frac{\tau}{2}(z - \mu)^2}$$

** Read this for a full breakdown of the steps* https://stmorse.github.io/journal/gibbs.html

Full conditional posteriors via Bayes' Rule

Precision τ

$$p(\tau|\beta_0, \beta_1, \boldsymbol{y}, \boldsymbol{x}) \propto p(\boldsymbol{y}, \boldsymbol{x}|\beta_0, \beta_1, \tau)\, p(\tau)$$

$$= \prod_{i=1}^{N} \mathcal{N}(\beta_0 + \beta_1 x_i, 1/\tau)\, \mathrm{Gamma}(a, b)$$

$$= \prod_{i=1}^{N} \sqrt{\frac{\tau}{2\pi}}\, e^{-\frac{\tau}{2}(y_i - (\beta_0 + \beta_1 x_i)^2}\, \frac{b^a}{\Gamma(a)} \tau^{a-1} e^{-b\tau}$$

*Some clever rearranging **

$$\tau|\beta_0, \beta_1, \boldsymbol{y}, \boldsymbol{x} \sim \mathrm{Gamma}(a + \frac{N}{2}, b + \frac{1}{2}\sum_{i=1}^{N}(y_i - (\beta_0 + \beta_1 x_i))^2)$$

Gamma probability density function

$$x\,|a, b \sim \mathrm{Gamma}(a, b)$$

$$p(x|a, b) = \mathrm{Gamma}(a, b)$$

$$p(x|a, b) = \frac{b^a}{\Gamma(a)} x^{a-1} e^{-bx}$$

** Read this for a full breakdown of the steps* https://stmorse.github.io/journal/gibbs.html

We can code the full conditional posterior for a single parameter as a sampling function in R!

Intercept $\beta_0 | \beta_1, \tau, \boldsymbol{y}, \boldsymbol{x} \sim \mathcal{N}\left(\frac{\tau_0\mu_0 + \tau \sum_{i=1}^{N}(y_i - \beta_1 x_i)}{\tau_0 + \tau N}, (\tau_0 + \tau)^{-1}\right)$

```
sample_b0 <- function(x,y, beta_1, tau, mu_0, tau_0){
       N <- length(x)
       new_precision <-  tau_0 + tau * N
       new_mean <- (tau_0 * mu_0 + tau * sum(y - beta_1 * x))/new_precision
       b0samp <- rnorm(1,new_mean, 1/sqrt(new_precision))
       return(b0samp)
}
```

Generates one value of β_0 *by* ***sampling on a normal distribution*** *with mean and sd* ***calculated from input values*** *for the* ***data****,* ***parameters****, and* ***hyperparameters***

We can code the full conditional posterior for a single parameter as a sampling function in R!

Slope $$\beta_1 | \beta_0, \tau, \boldsymbol{y}, \boldsymbol{x} \sim \mathcal{N}\left(\frac{\tau_1\mu_1 + \tau\sum_{i=1}^{N}(y_i - \beta_0)x_i}{\tau_1 + \tau\sum_{i=1}^{N} x_i^2}, \left(\tau_1 + \tau\sum_{i=1}^{N} x_i^2\right)^{-1}\right)$$

```
sample_b1 <- function(x,y, beta_0, tau, mu_1, tau_1){
      N <- length(x)
      new_precision <-  tau_0 + tau * sum(x^2)
      new_mean <- (tau_1 * mu_1 + tau * sum(x*(y - beta_0)))/new_precision
      b1samp <- rnorm(1,new_mean, 1/sqrt(new_precision))
      return(b1samp)
}
```

Generates one value of β_1 *by* ***sampling on a normal distribution*** *with mean and sd* ***calculated from input values*** *for the* ***data****,* ***parameters****, and* ***hyperparameters***

We can code the full conditional posterior for a single parameter as a sampling function in R!

Precision $\tau | \beta_0, \beta_1, \boldsymbol{y}, \boldsymbol{x} \sim \text{Gamma}(a + \frac{N}{2}, b + \frac{1}{2}\sum_{i=1}^{N}(y_i - (\beta_0 + \beta_1 x_i))^2)$

```
sample_tau <- function(x,y, beta_0, beta_1, a, b){
        N <- length(x)
        new_shape <- a + N/2
        residual <- y - beta_0 - beta_1 * x
        new_rate <- b + sum(residual^2)/2
        tausamp <- rgamma(1, shape = new_shape, rate = new_rate)
        return(tausamp)
}
```

Generates one value of τ *by* ***sampling on a normal distribution*** *with shape and rate* ***calculated from input values*** *for the* ***data****,* ***parameters****, and* ***hyperparameters***

Gibbs sampling algorithm

1. Set a starting value for each parameter

Initialization (iteration 0):

$\beta_0^{(0)} = 0$,$\beta_1^{(0)} = 0$,$\tau^{(0)} = 1$

2. Sample each parameters using input data and previous values of other parameters

Iteration 1:

Update intercept: $\beta_0^{(1)} = \mathit{sample\_b0}\left(\beta_1^{(0)}, \tau^{(0)}, \mathbf{x}, \mathbf{y}\right)$

Update slope: $\beta_1^{(1)} = \mathit{sample\_b1}\left(\beta_0^{(1)}, \tau^{(0)}, \mathbf{x}, \mathbf{y}\right)$

Update precision: $\tau^{(1)} = \mathit{sample\_tau}\left(\beta_0^{(1)}, \beta_1^{(1)}, \mathbf{x}, \mathbf{y}\right)$

3. Repeat for many iterations

Iteration t:

Update intercept: $\beta_0^{(t)} = \mathit{sample\_b0}\left(\beta_1^{(t-1)}, \tau^{(t-1)}, \mathbf{x}, \mathbf{y}\right)$

Update slope: $\beta_1^{(t)} = \mathit{sample\_b1}\left(\beta_0^{(t)}, \tau^{(t-1)}, \mathbf{x}, \mathbf{y}\right)$

Update precision: $\tau^{(t)} = \mathit{sample\_tau}\left(\beta_0^{(t)}, \beta_1^{(t)}, \mathbf{x}, \mathbf{y}\right)$

Next: Look through the Gibbs sampler code and exercises in R.

Runnable code:
GibbsSampler.Rmd

Formatted output:
GibbsSampler.html