Contents

simple linear model 1
likelihood 1
priors 1
posterior distribution 2
Gibbs sampling 2
R code 2

simple linear model

The simple linear regression model is

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

, where

- y_i is the i-th observation for the outcome variable y
- β_0 is the intercept
- x_i is the i-th observation for the predictor variable x
- β_1 is the coefficient
- ϵ_i is the intercept with a distribution $N(0, \sigma^2)$, or $N(0, 1/\tau)$ where $\tau = \frac{1}{\sigma^2}$

likelihood

Given a sample with n observations, the likelihood is

$$p(y|\beta_0, \beta_1, \tau) = \prod_{i=1}^{n} N(\beta_0 + \beta_1 x_i, 1/\tau) = \prod_{i=1}^{n} \sqrt{\frac{\tau}{2\pi}} e^{-\frac{(y-\beta_0-x_i\beta_1)^2\tau}{2}}$$

priors

In Bayesian statistics, the parameters also have a distribution, called priors. The prior distributions for our model are

$$\beta_0 \sim N(\mu_0, 1/\tau_0)$$

$$\beta_1 \sim N(\mu_1, 1/\tau_1)$$

$$\tau \sim \text{Gamma}(\alpha, \beta)$$

So the probability function of these prior distributions are

$$p(\beta_0|\mu_0, \tau_0) = \sqrt{\frac{\tau_0}{2\pi}} e^{-\frac{(\beta_0-\mu_0)^2\tau_0}{2}}$$
$$p(\beta_1|\mu_1, \tau_1) = \sqrt{\frac{\tau_1}{2\pi}} e^{-\frac{(\beta_1-\mu_1)^2\tau_1}{2}}$$

$$p(\tau|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \tau^{\alpha-1} e^{-\beta\tau}$$

The reason to choose these priors are due to computational simplicity ("*conjugate priors*"). There could be other ways to choose priors. The parameters $\mu_0, \tau_0, \mu_1, \tau_1, \alpha, \beta$ are called *hyperparameters*.

posterior distribution

Based on Bayes' formula, the posterior distribution is

$$p(\beta_0, \beta_1, \tau|y) \propto p(y|\beta_0, \beta_1, \tau)p(\beta_0)p(\beta_1)p(\tau)$$

Gibbs sampling

The joint posterior distribution sometimes can be hard to get. Gibbs sampling uses the conditional posterior distributions.

updates for β_0

$p(\beta_0|y, \beta_1, \tau) \propto p(y|\beta_0, \beta_1, \tau)p(\beta_0)$

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

$\propto exp[-\frac{\tau}{2}\sum(y_i - \beta_0 - \beta_1 x_i)^2 - \frac{\tau_0}{2}(\beta_0 - \mu_0)^2]$

$= exp[-\frac{1}{2}[(\tau_0 + \tau n)\beta_0^2 - 2(\tau_0\mu_0 + \tau\sum(y_i - \beta_1 x_i))\beta_0 + C]]$where C is a constant

$\sim N(\frac{\tau_0\mu_0+\tau\sum(y_i-\beta_1 x_i)}{\tau_0+\tau n}, \frac{1}{\tau_0+\tau n})$

updates for β_1

Similarly,

$p(\beta_1|y, \beta_0, \tau) \propto p(y|\beta_0, \beta_1, \tau)p(\beta_1)$

$\sim N(\frac{\tau_1\mu_1+\tau\sum(y_i-\beta_0)x_i}{\tau_1+\tau\sum x_i^2}, \frac{1}{\tau_1+\tau\sum x_i^2})$

updates for τ

$p(\tau|y, \beta_0, \beta_1) \propto p(y|\beta_0, \beta_1, \tau)p(\tau)$

$= \prod_{i=1}^n \sqrt{\frac{\tau}{2\pi}} e^{-\frac{(y_i-\beta_0-x_i\beta_1)^2\tau}{2}} \frac{\beta^\alpha}{\Gamma(\alpha)}\tau^{\alpha-1}e^{-\beta\tau}$

$\propto \tau^{n/2} \times e^{-\sum(y_i-\beta_0-\beta_1 x_i)^2\times\tau/2} \times \tau^{\alpha-1} \times e^{-\beta\tau}$

$= \tau^{n/2+\alpha-1}e^{-\tau(\frac{\sum(y-\beta_0-\beta_1 x_i)^2}{2}+\beta)}$

$\sim \text{Gamma}(n/2 + \alpha, \frac{\sum(y-\beta_0-\beta_1 x_i)^2}{2} + \beta)$

R code

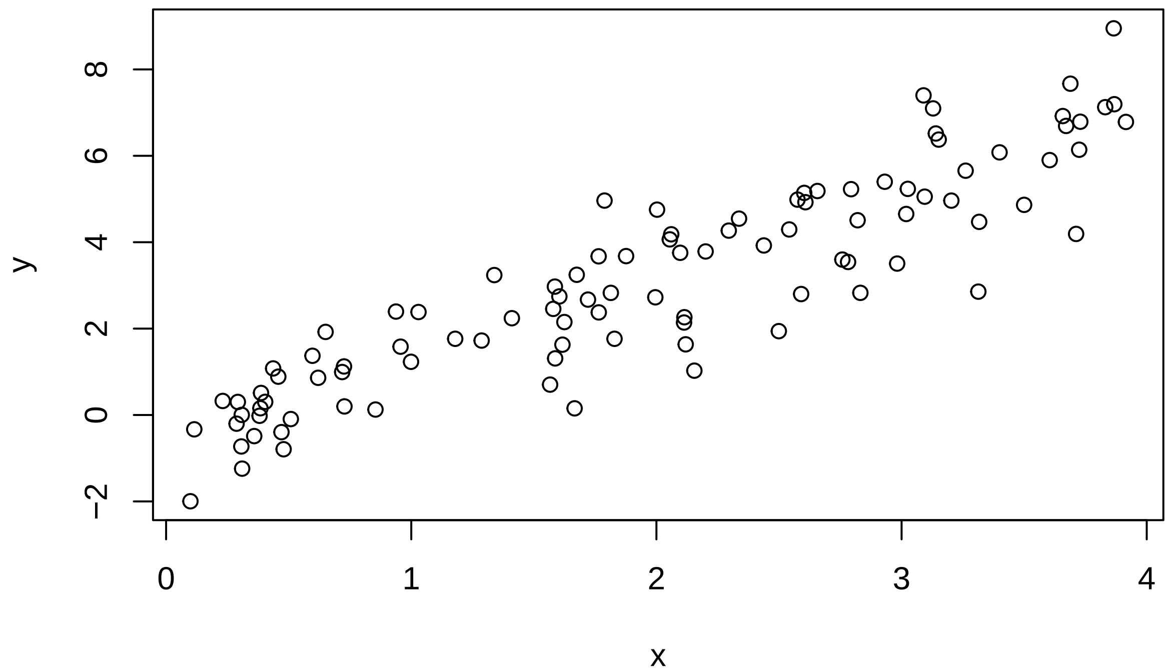
```r
### sampling beta_0
sample_beta_0 = function(y, x, beta_1, tau, mu_0, tau_0){
    n=length(y)
    precision = tau_0+tau*n
    mean = tau_0*mu_0 + tau*sum(y-beta_1*x)
    mean = mean/precision
    return(rnorm(1, mean=mean, sd=1/sqrt(precision)))
}
```

```r
### sampling beta_1
sample_beta_1 = function(y, x, beta_0, tau, mu_1, tau_1){
    n=length(y)
    precision = tau_1 + tau * sum(x^2)
    mean = tau_1*mu_1 + tau*sum((y-beta_0)*x)
    mean = mean/precision
    return(rnorm(1, mean=mean, sd=1/sqrt(precision)))
}

### sampling tau
sample_tau = function(y, x, beta_0, beta_1, alpha, beta) {
    n = length(y)
    alpha_new = alpha + n/2
    resid = y - beta_0 - beta_1*x
    beta_new = beta + sum(resid^2)/2
    return(rgamma(1, shape=alpha_new, rate=beta_new))
}

## Gibbs sampling
gibbs_sample = function(y, x, iters, init, hypers){
    beta_0 = init[["beta_0"]]
    beta_1 = init[["beta_1"]]
    tau = init[["tau"]]

    trace = c()
    for(i in 1:iters){
        beta_0 = sample_beta_0(y, x, beta_1, tau, hypers[["mu_0"]], hypers[["tau_0"]])
        beta_1 = sample_beta_1(y, x, beta_0, tau, hypers[["mu_1"]], hypers[["tau_1"]])
        tau = sample_tau(y, x, beta_0, beta_1, hypers[["alpha"]], hypers[["beta"]])
        trace = rbind(trace, data.frame(beta_0, beta_1, tau))
    }
    return(trace)
}

## simulation
beta_0_true = -1
beta_1_true = 2
tau_true = 1

n = 100
set.seed(422)
x = runif(n, min=0, max=4)
y = beta_0_true + beta_1_true*x + rnorm(n, 0, 1/sqrt(tau_true))
plot(x, y)
summary(lm(y~x))
```

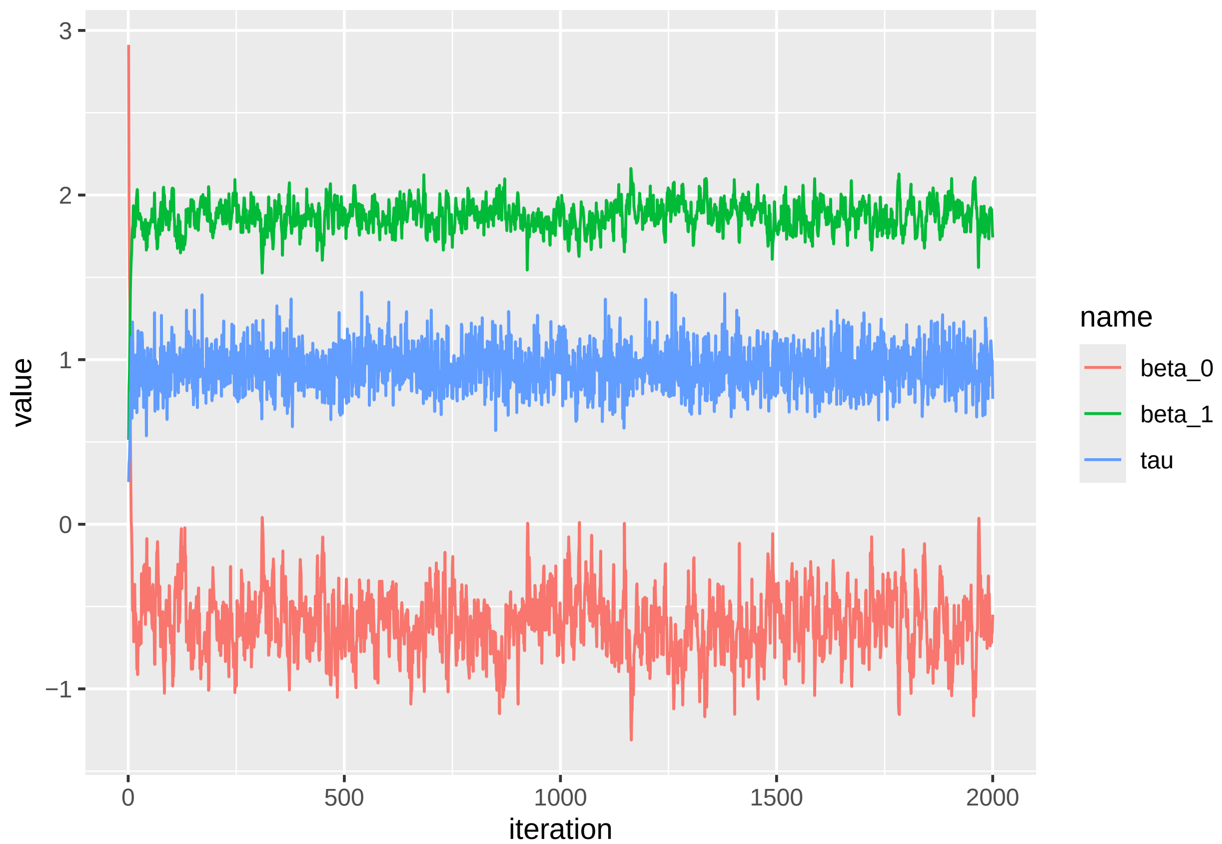
```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7786 -0.5421  0.1250  0.6312  2.2655
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.66967    0.20586  -3.253  0.00157 **
## x            1.90276    0.09073  20.971  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.034 on 98 degrees of freedom
## Multiple R-squared:  0.8178, Adjusted R-squared:  0.8159
## F-statistic: 439.8 on 1 and 98 DF,  p-value: < 2.2e-16
```

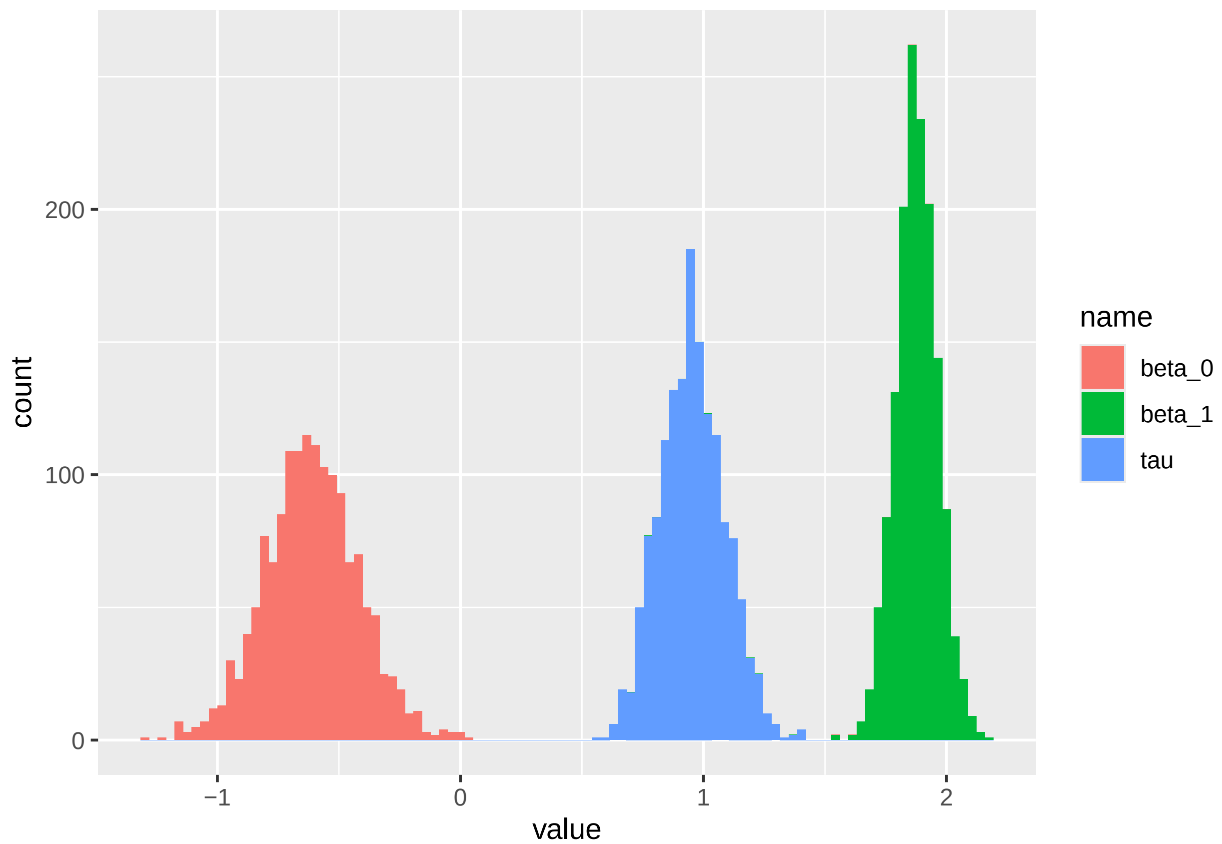
```
## specify initial values
init = list("beta_0"=0,
        "beta_1"=0,
        "tau"=2)

## specify hyper parameters
hypers = list("mu_0"=0,
         "tau_0"=1,
         "mu_1"=0,
         "tau_1"=1,
         "alpha"=2,
         "beta"=1)

## run
iters = 2000
trace = gibbs_sample(y, x, iters, init, hypers)

## trace plot (the first file iterations with fluctuating estimates are called "burn-in" period)
suppressWarnings(suppressMessages(library(tidyverse)))
```




```
trace %>% mutate(iteration = row_number()) %>% pivot_longer(-"iteration") %>%
    ggplot() + geom_path(aes(x=iteration, y=value, colour=name))
```




```r
## distribution
trace_burnt = trace[501:nrow(trace),]
trace_burnt %>% mutate(iteration = row_number()) %>% pivot_longer(-"iteration") %>%
    ggplot() + geom_histogram(aes(x=value, fill=name), bins=100)
```




```
apply(trace_burnt,2,mean)
```

```
##     beta_0     beta_1       tau
## -0.6122425  1.8765900  0.9541811
```

```
apply(trace_burnt,2,sd)
```

```
##     beta_0     beta_1        tau
## 0.19359102 0.08565184 0.13367839
```