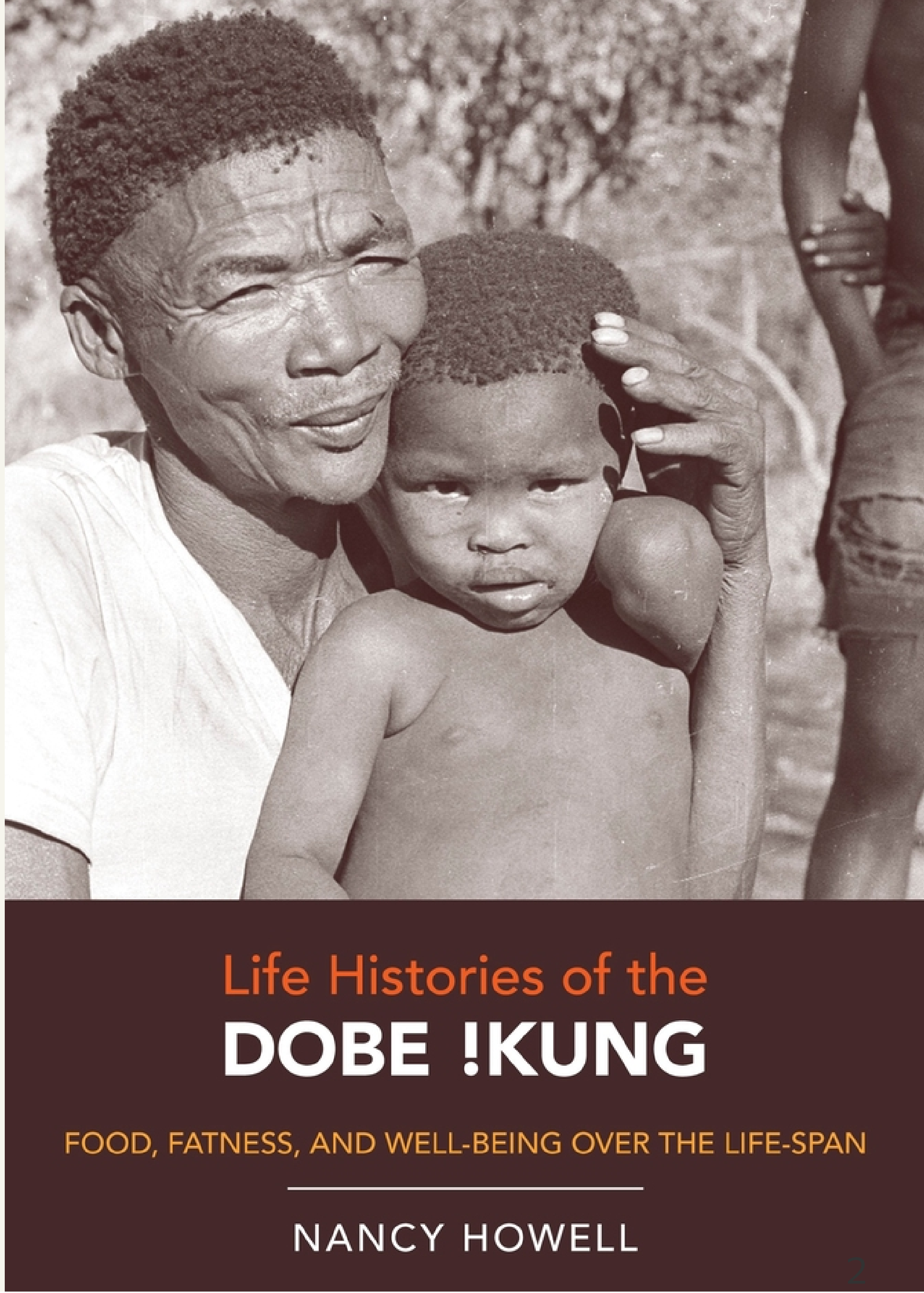
Bayesian regression

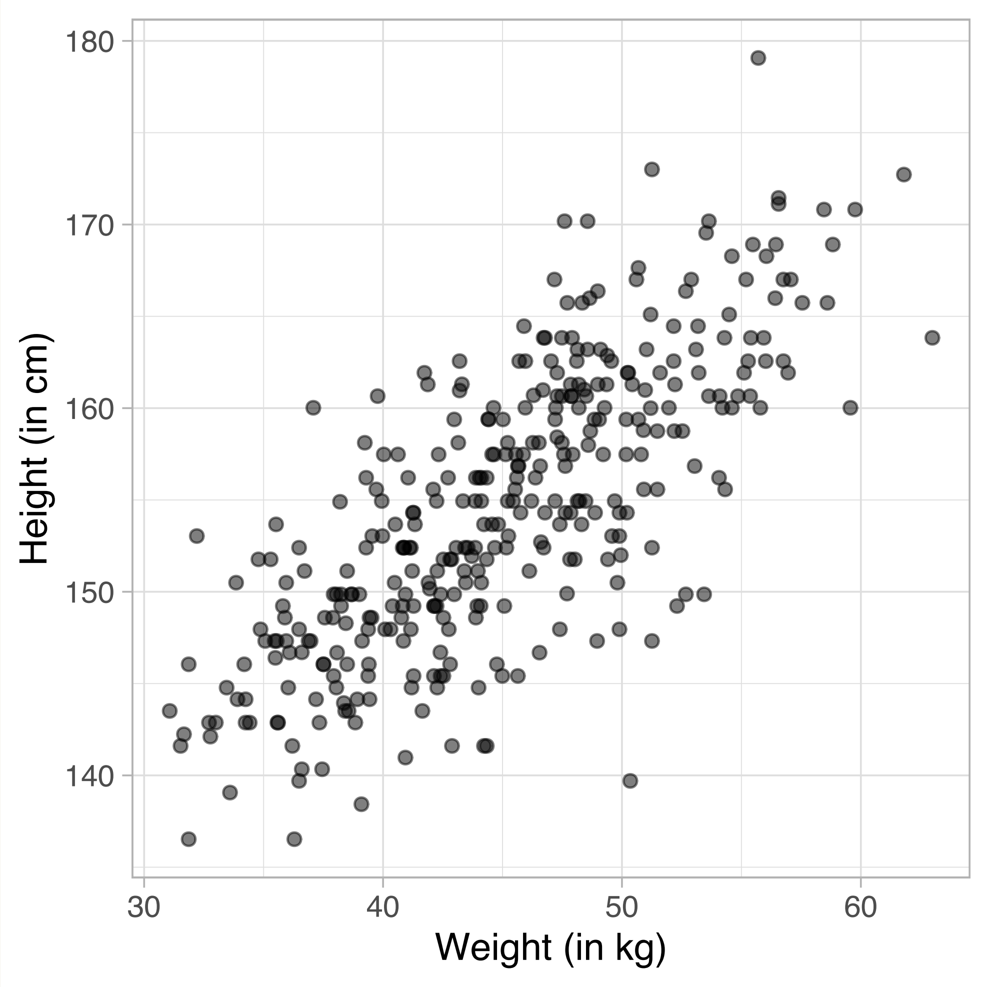
Stat 340: Bayesian Statistics

Example

- Partial census data for the Dobe area !Kung San, a foraging population
- Compiled from Nancy Howell's interviews

```
##              mean         sd      5.5%     94.5%       histogra
## height 154.59709  7.7423321 142.8750 167.00500
## weight  44.99049  6.4567081  35.1375  55.76588
## age     41.13849 15.9678551  20.0000  70.00000
## male     0.46875  0.4997328   0.0000   1.00000
```





How can we write a **Bayesian** model to relate weight and height of the Kalahari foragers?

Observation-specific mean

We can adapt our normal model for the mean to use an observation-specific mean, μ_i:

Sampling model: $\quad Y_i|\mu_i, \sigma \stackrel{\text{ind}}{\sim} \mathcal{N}(\mu_i, \sigma)$

Now we need to link μ_i and x_i

A weakly informative prior

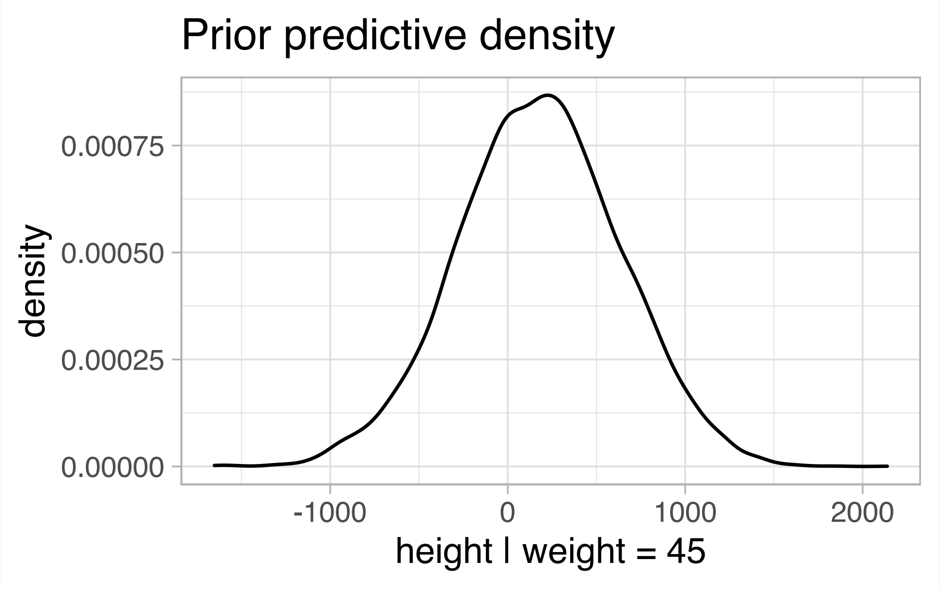
We may have limited prior information about the regression coefficients, β_0 and β_1, and/or the standard deviation σ

Assume independence: $\pi(\beta_0, \beta_1, \sigma) = \pi(\beta_0, \beta_1)\pi(\sigma)$

1. Assume β_0 and β_1 are independent
   - $\pi(\beta_0, \beta_1) = \pi(\beta_0)\pi(\beta_1)$
   - Assign a weakly informative prior each coefficient: $\beta_i \sim \mathcal{N}(m_i, s_i)$
   - example: $\mathcal{N}(0, 100)$

2. Assign a weakly informative prior to σ
   - example: $1\sigma^2 \sim \text{Gamma}(1, 1)$

Prior predictive as "sanity check"



Simulate the prior predictive:

1. Draw parameters from their prior distributions

2. Draw data from the sampling model plugging in these simulated parameters

Sampling from the prior predictive distribution

```
nsim <- 1e4                                               # no. of simulations

prior.sims <- data_frame(                                 # simulate from ind. priors
  beta0 = rnorm(nsim, 178, 100),
  beta1 = rnorm(nsim, 0, 10),
  sigma = runif(nsim, 0, 50)
)
```

```
weight.value <- 45                                        # condition on value of x
```

```
prior.pred <- prior.sims %>%
  mutate(
    mu = beta0 + beta1 * weight.value,                    # calculate E(y|x)
    height = rnorm(n(), mean = mu, sd = sigma)            # draw sample from normal
  )
```

Deriving the posterior

Sampling model: $Y_i | x_i, \beta_0, \beta_1, \sigma \sim \mathcal{N}(\beta_0 + \beta_1 x_i, \sigma)$

Joint likelihood:

$$
\begin{aligned}
L(\beta_0, \beta_1, \sigma) &= \prod_{i=1}^{n} \left[\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{1}{2\sigma^2}(y_i - \beta_0 - \beta_1 x_i)^2 \right\} \right] \\
&\propto \left(\frac{1}{\sigma^2} \right)^{n/2} \exp\left\{ -\frac{1/\sigma^2}{2} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2 \right\}
\end{aligned}
$$

Joint posterior: $\pi(\beta_0, \beta_1, \sigma | \text{data}) \propto \pi(\beta_0, \beta_1, \sigma) L(\beta_0, \beta_1, \sigma)$

JAGS for Bayesian SLR

Write down the model string

```
slr_model <-"model {
## sampling model
for (i in 1:N){
y[i] ~ dnorm(beta0 + beta1 * x[i], invsigma2)
}

## priors
beta0 ~ dnorm(mu0, g0)
beta1 ~ dnorm(mu1, g1)
invsigma2 ~ dgamma(a, b)
sigma <- sqrt(pow(invsigma2, -1))
}"
```

JAGS for Bayesian SLR

Define the data and set prior parameters

```
the_data <- list(
  y = adults$height,    # response variable
  x = adults$weight,    # explanatory variable
  N = nrow(adults),     # sample size
  mu0 = 0,              # prior mean for beta0
  g0 = 0.0001,          # prior precision for beta0
  mu1 = 0,              # prior mean for beta1
  g1 = 0.0001,          # prior precision for beta1
  a = 1,                # prior shape for 1/sigma2
  b = 1                 # prior scale for 1/sigma2
)
```

JAGS for Bayesian SLR

Generate samples from the posterior

```
posterior <- run.jags(
  slr_model,
  data = the_data,
  n.chains = 1,
  monitor = c("beta0", "beta1", "sigma"),
  adapt = 1000,
  burnin = 5000,
  sample = 5000,
  silent.jags = TRUE
)
```

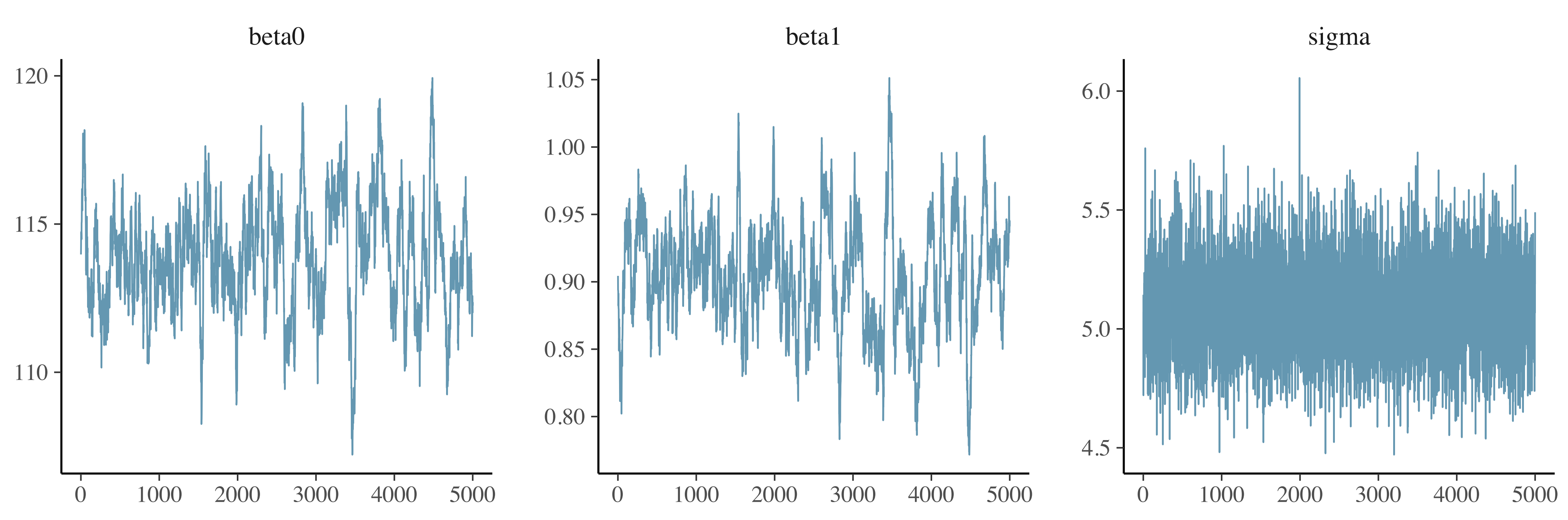
Summary of the fitted model

```
summary(posterior)
```

```
##             Lower95      Median     Upper95        Mean         SD        Mode
## beta0 110.185429 113.8831812 117.5070605 113.8926504 1.85060652 113.6446508
## beta1   0.823308   0.9048906   0.9842071   0.9046513 0.04065837   0.9064154
## sigma   4.692724   5.0789234   5.4558836   5.0853508 0.19425986   5.0721822
##               MCerr MC%ofSD SSeff        AC.10 psrf
## beta0 0.246236363    13.3    56  0.791740214   NA
## beta1 0.005487650    13.5    55  0.793173622   NA
## sigma 0.002747249     1.4  5000 -0.004470933   NA
```

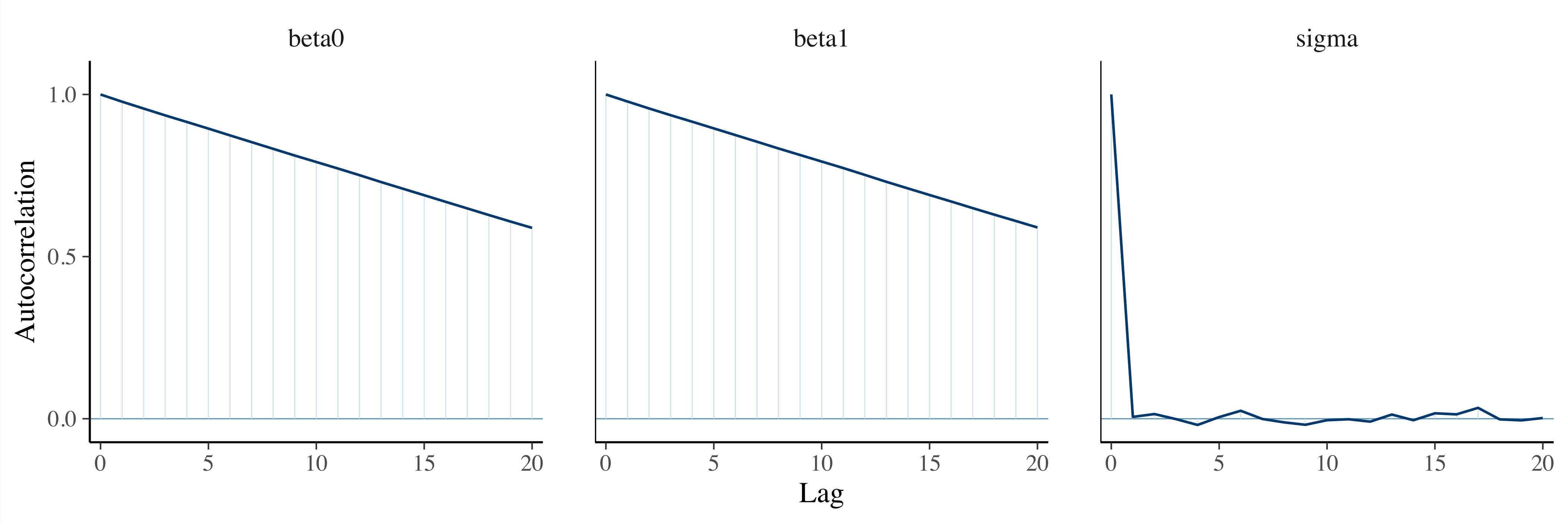
MCMC diagnostics

```
mcmc_trace(posterior$mcmc)
```



MCMC diagnostics

```
mcmc_acf(posterior$mcmc)
```

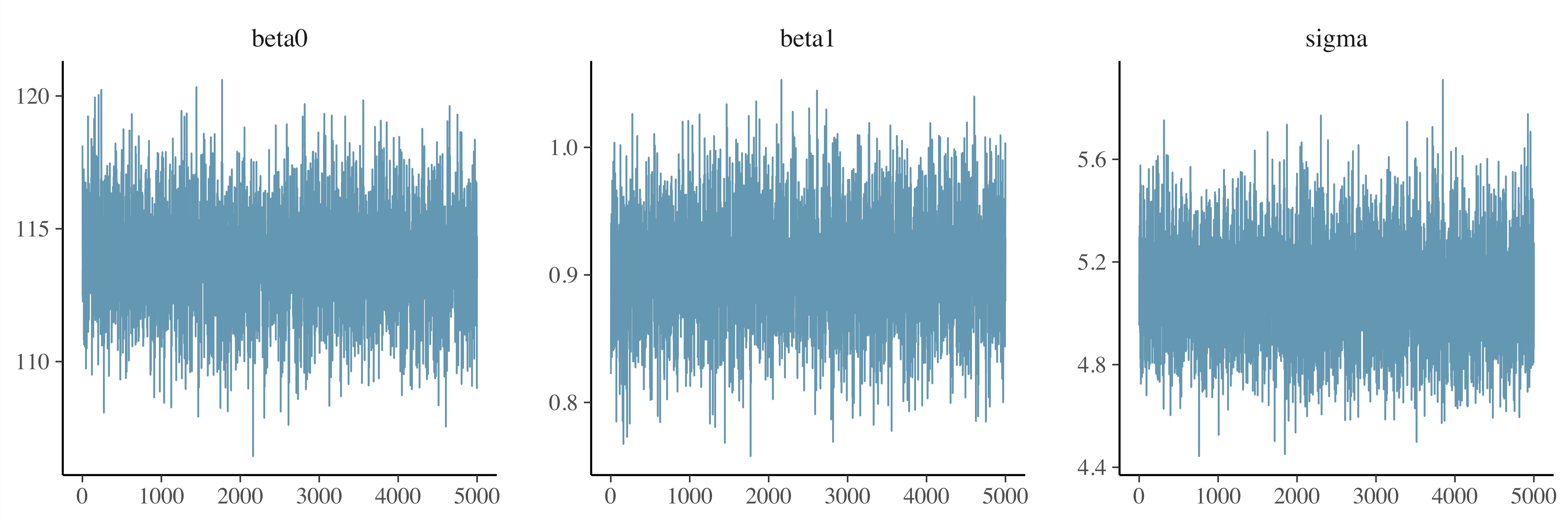


Setting `thin = 50`

```
posterior <- run.jags(
  slr_model,
  data = the_data,
  n.chains = 1,
  monitor = c("beta0", "beta1", "sigma"),
  adapt = 1000,
  burnin = 5000,
  sample = 5000,
  thin = 50,
  silent.jags = TRUE
)
```

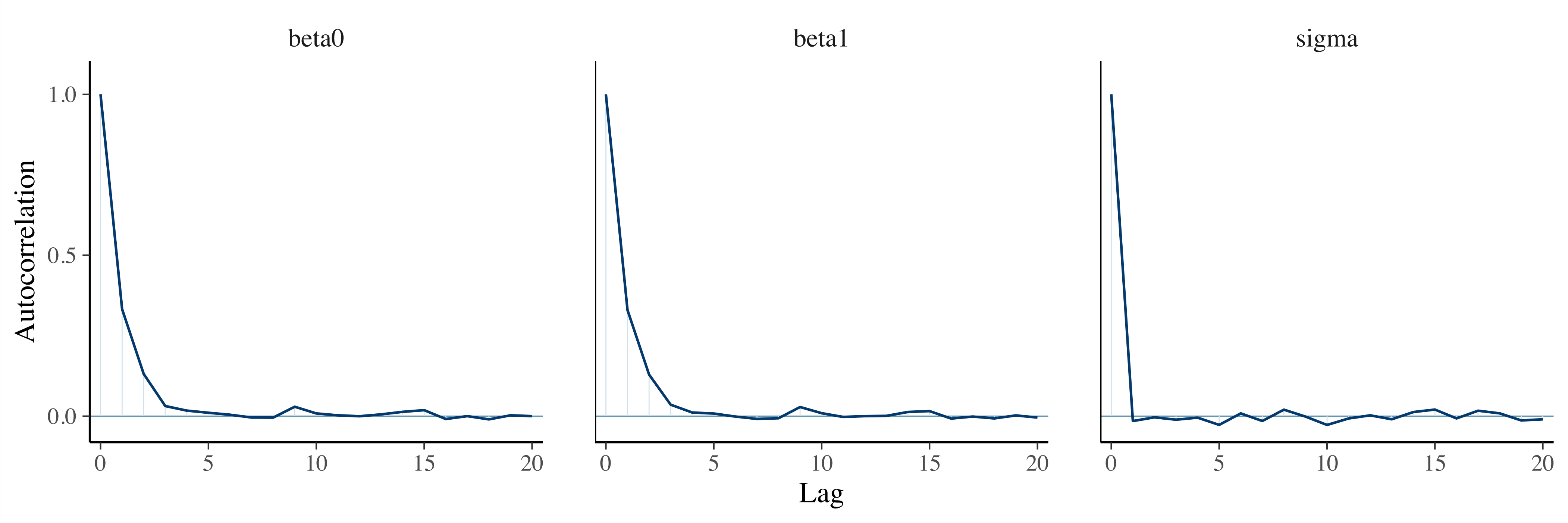
MCMC diagnostics

```
mcmc_trace(posterior$mcmc)
```



MCMC diagnostics

```
mcmc_acf(posterior$mcmc)
```



Summary of the fitted model

```
summary(posterior)
```

```
##              Lower95      Median     Upper95        Mean          SD         Mod
## beta0 110.1454786 113.8402826 117.6245171 113.8453309 1.90365131 113.681470
## beta1   0.8258936   0.9058944   0.9889891   0.9058291 0.04190476   0.908775
## sigma   4.7025891   5.0730595   5.4504958   5.0810728 0.19099318   5.094771
##              MCerr MC%ofSD SSeff       AC.500 psrf
## beta0 0.0381309057     2.0  2492  0.008593566   NA
## beta1 0.0008545736     2.0  2405  0.009331956   NA
## sigma 0.0027010514     1.4  5000 -0.027179755   NA
```

Plotting the fitted model

```
post_means <- apply(
  posterior$mcmc[[1]], 2, mean
)
```

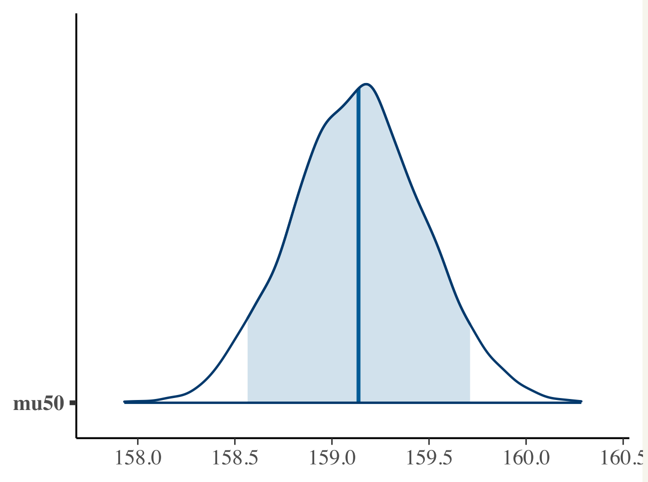
Sampling from the joint posterior

```
post_draws <- as.data.frame(
  posterior$mcmc[[1]]
)
head(post_draws)
```

```
##            beta0     beta1    sigma
## 6001 118.1336 0.8222446 5.299407
## 6051 114.7289 0.8955723 4.953780
## 6101 112.5141 0.9357981 5.153145
## 6151 113.4607 0.9216525 5.020973
## 6201 112.2582 0.9389457 5.214484
## 6251 112.8237 0.9406860 4.993891
```

Generating mean responses

```r
mu_at_50 <- with(post_draws,
                 beta0 + beta1 * 50)
```




```r
quantile(mu_at_50, probs = c(0.05, 0.95))
##       5%      95%
## 158.5656 159.7114
```