Bayesian linear regression: Model fitting in JAGS

Stat 340: Bayesian Statistics

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

```r
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
)
```

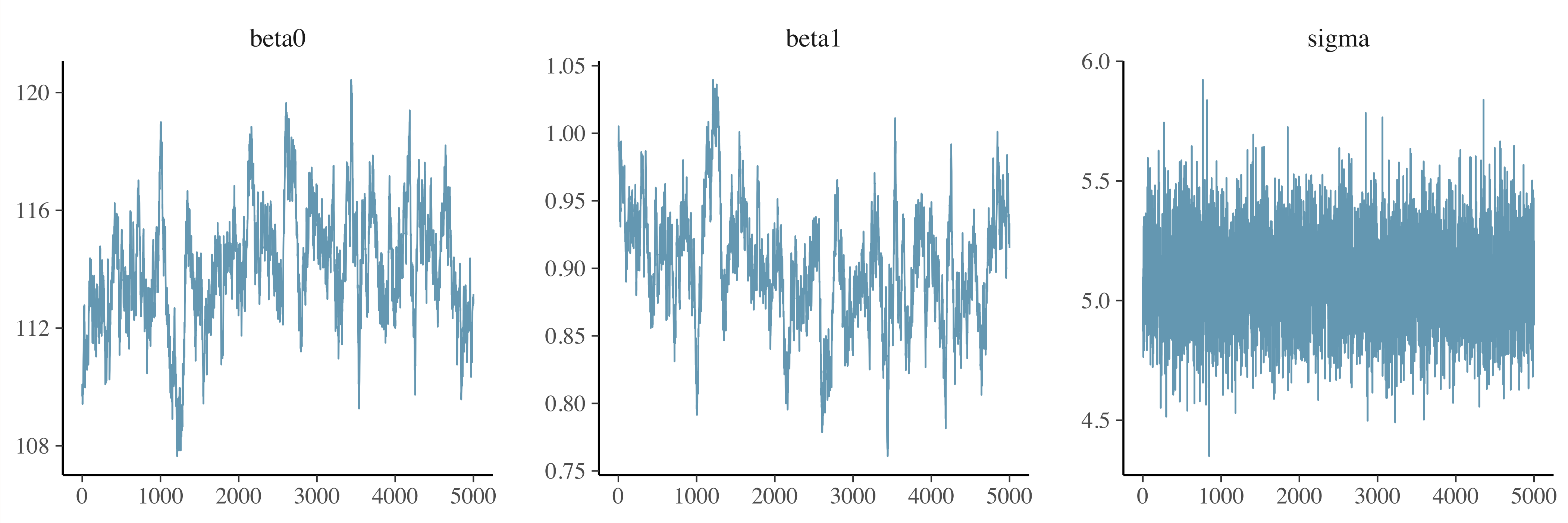
Summary of the fitted model

```
summary(posterior)
```

```
##              Lower95      Median     Upper95        Mean          SD         Mod
## beta0 109.9239345 114.0514287 117.8570568 114.0582551 1.96278394 113.890967
## beta1   0.8176516   0.9010266   0.9923402   0.9010779 0.04317723    0.901268
## sigma   4.7257285   5.0865511   5.4687390   5.0877078 0.19274417    5.080021
##                MCerr MC%ofSD SSeff         AC.10 psrf
## beta0 0.276830286    14.1    50 0.8109435560   NA
## beta1 0.006083780    14.1    50 0.8091597027   NA
## sigma 0.002725814     1.4  5000 0.0007790325   NA
```

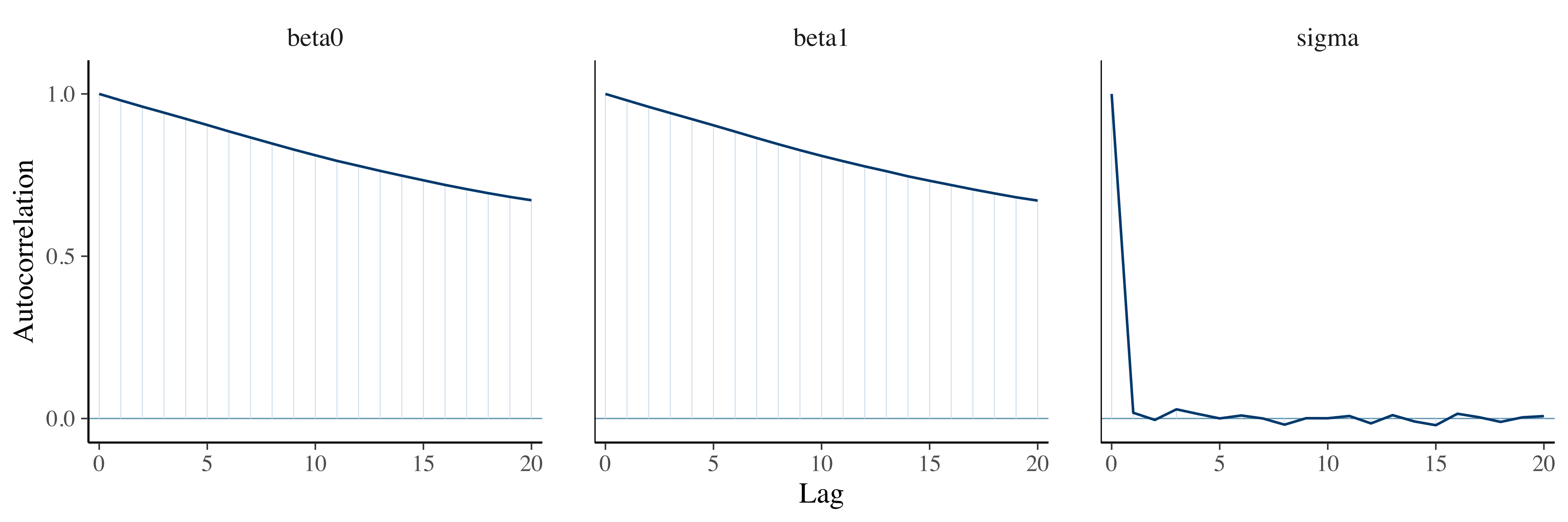
MCMC diagnostics

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



MCMC diagnostics

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

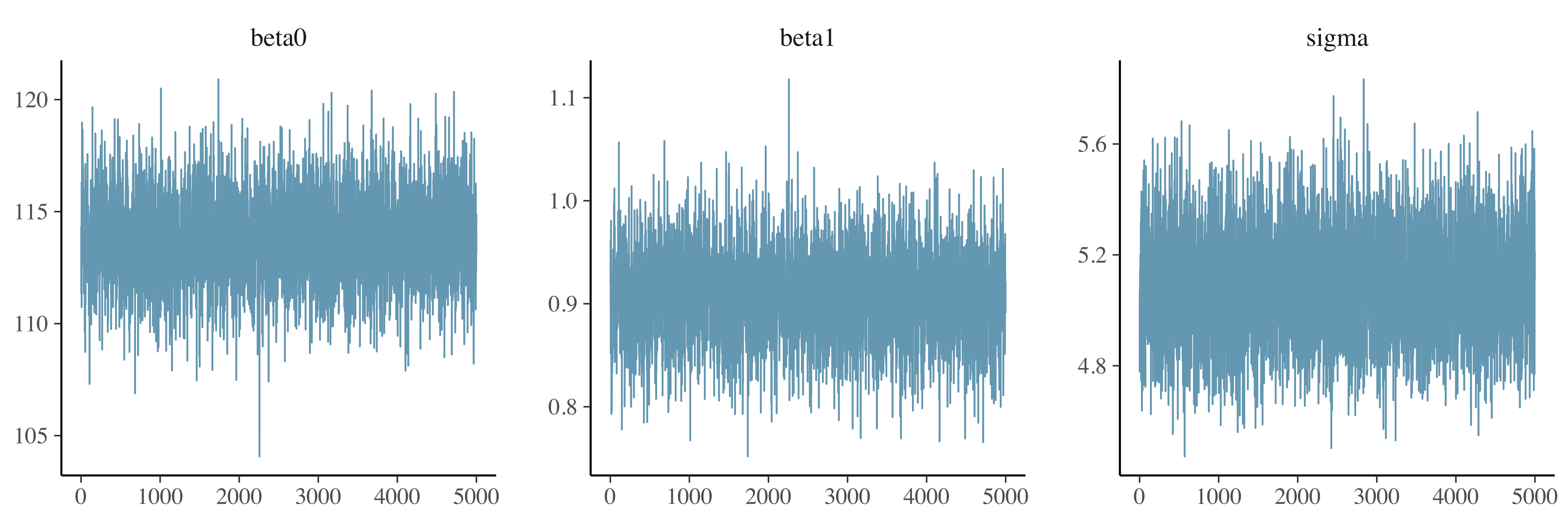


Setting `thin = 50`

```r
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
)
```

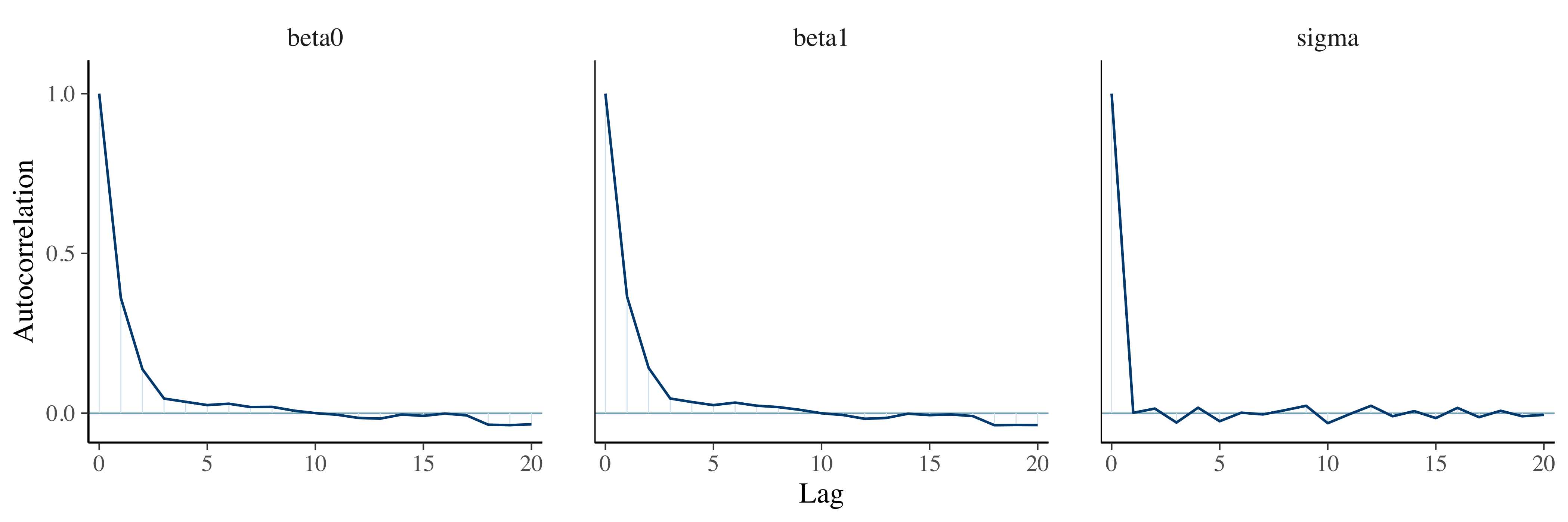
MCMC diagnostics

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



MCMC diagnostics

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



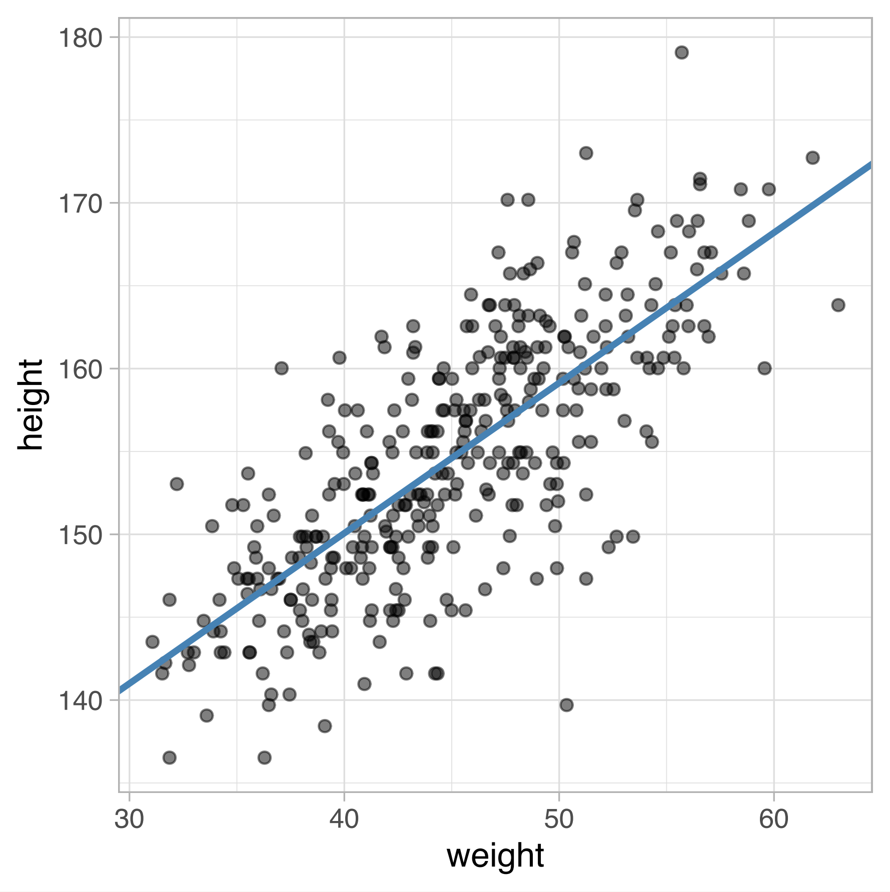
Summary of the fitted model

```
summary(posterior)
```

```
##               Lower95     Median     Upper95        Mean          SD        Mode
## beta0 110.1305805 113.816191 117.6676455 113.7985303 1.93775771 113.795387
## beta1   0.8258775   0.906227   0.9911123   0.9067596 0.04255407   0.905762
## sigma   4.7092614   5.079929   5.4433240   5.0842297 0.18928022   5.077372
##                MCerr MC%ofSD SSeff          AC.500 psrf
## beta0 0.0399985656     2.1  2347  0.0001235812   NA
## beta1 0.0008823188     2.1  2326 -0.0002873793   NA
## sigma 0.0026768266     1.4  5000 -0.0313963340   NA
```

Plotting the fitted model

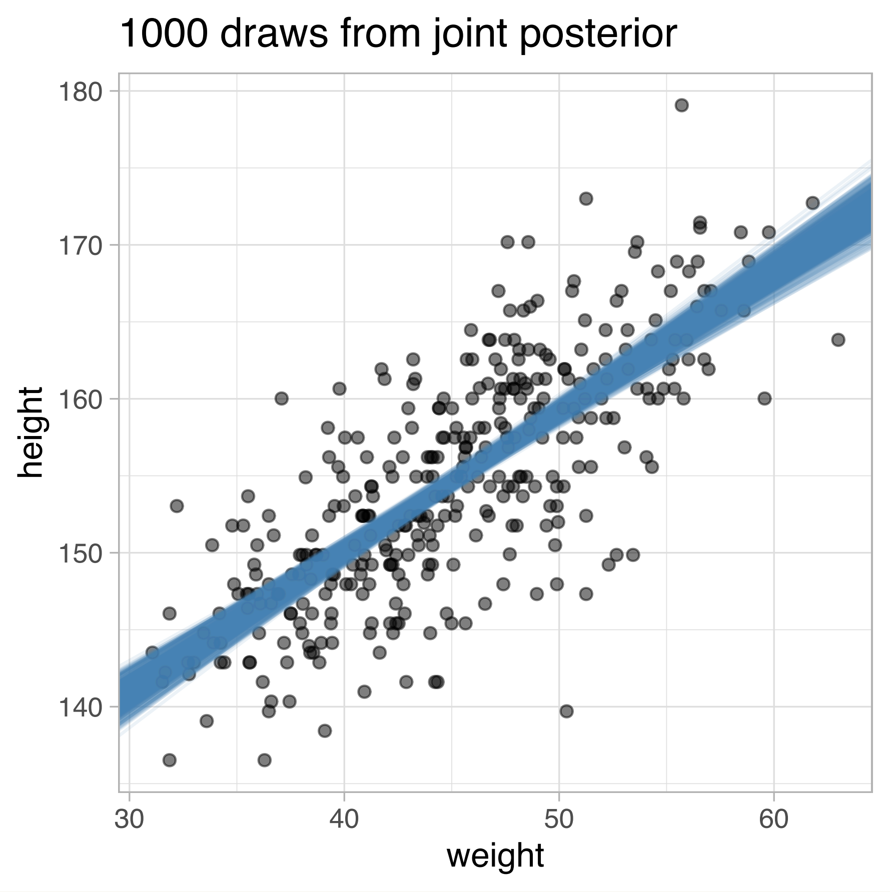
```
post_means <- apply(
  posterior$mcmc[[1]], 2, mean
)
```



Sampling from the joint posterior

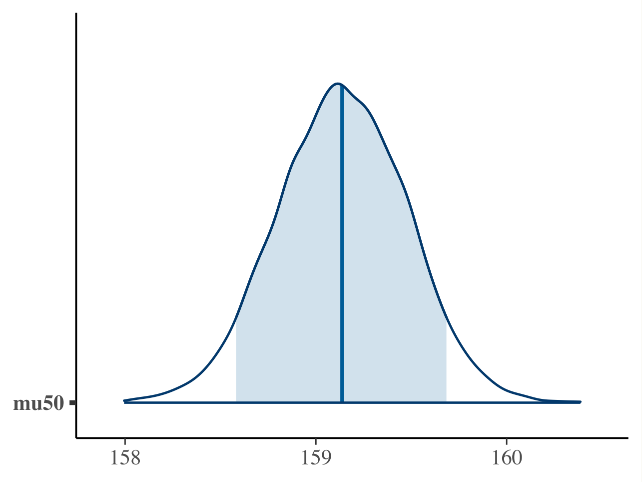
```
post_draws <- as.data.frame(
  posterior$mcmc[[1]]
)
head(post_draws)
```

```
##         beta0     beta1    sigma
## 6001 113.4758 0.9198960 5.075723
## 6051 113.3982 0.9271029 4.778472
## 6101 111.3489 0.9610665 5.002218
## 6151 114.3422 0.8982338 4.991962
## 6201 112.7927 0.9214061 4.954671
## 6251 116.3060 0.8514970 5.146747
```

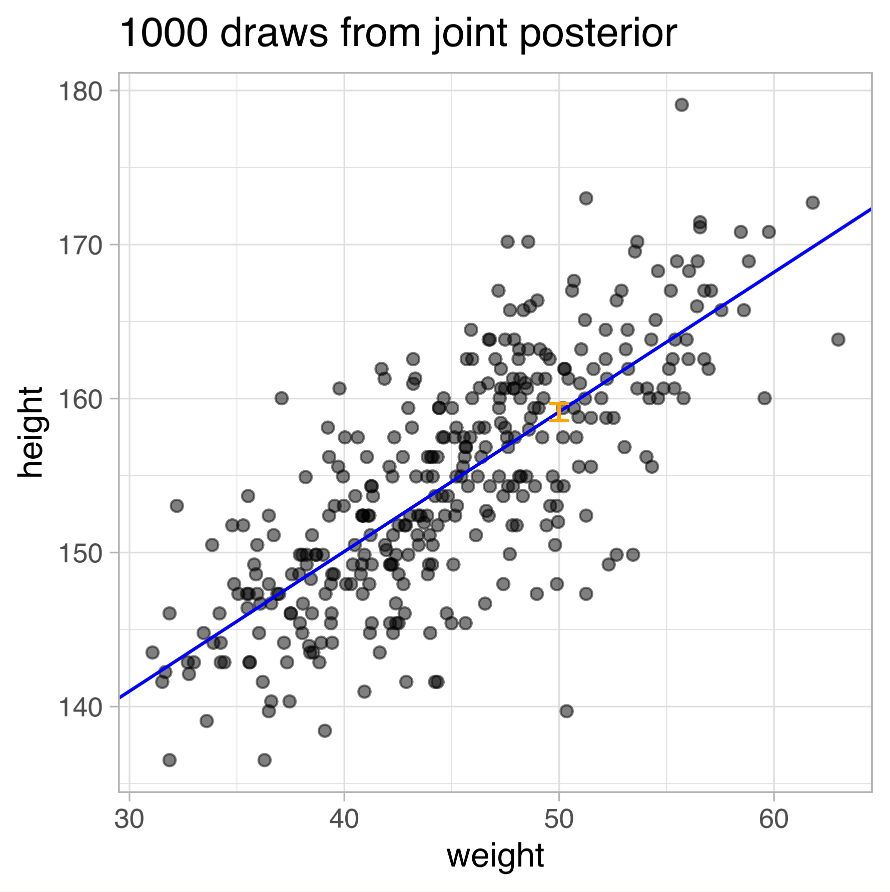


Generating mean responses

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

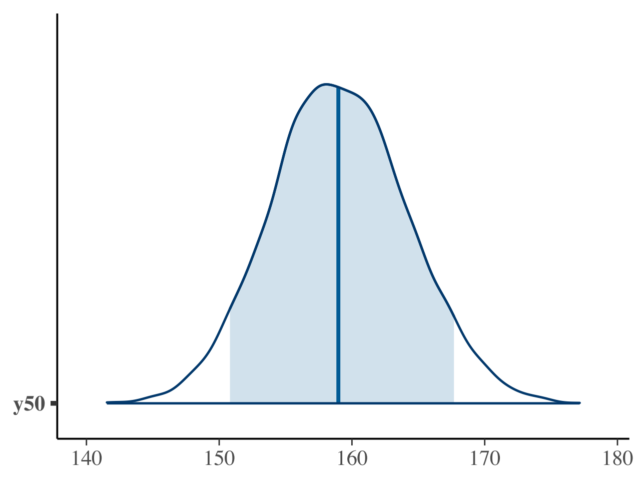


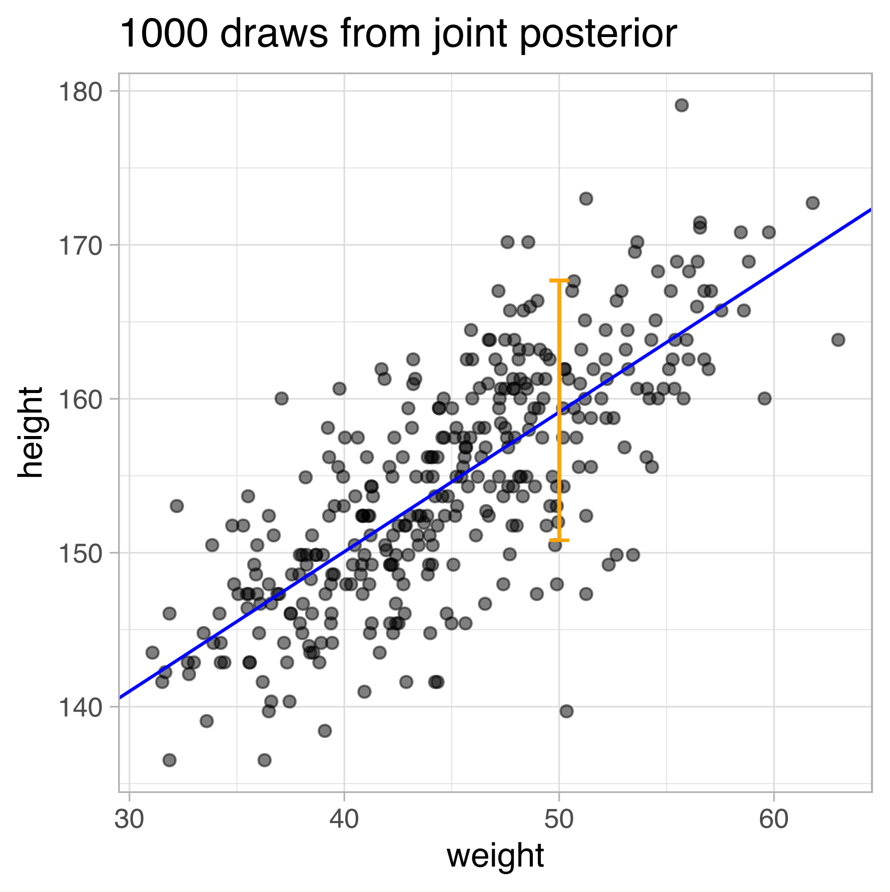

```
quantile(mu_at_50, probs = c(0.05, 0.95))
##        5%      95%
## 158.5815 159.6842
```



Generating predicted responses

```
mu_at_50 <- with(post_draws,
                 beta0 + beta1 * 50)
S <- length(mu_at_50)
y_at_50 <- rnorm(S,
                 mu_at_50,
                 post_draws$sigma)
```





Your turn

Work through the Bayesian regression handout with your neighbor(s).

Think about what ideas in simple linear regression are still unclear.