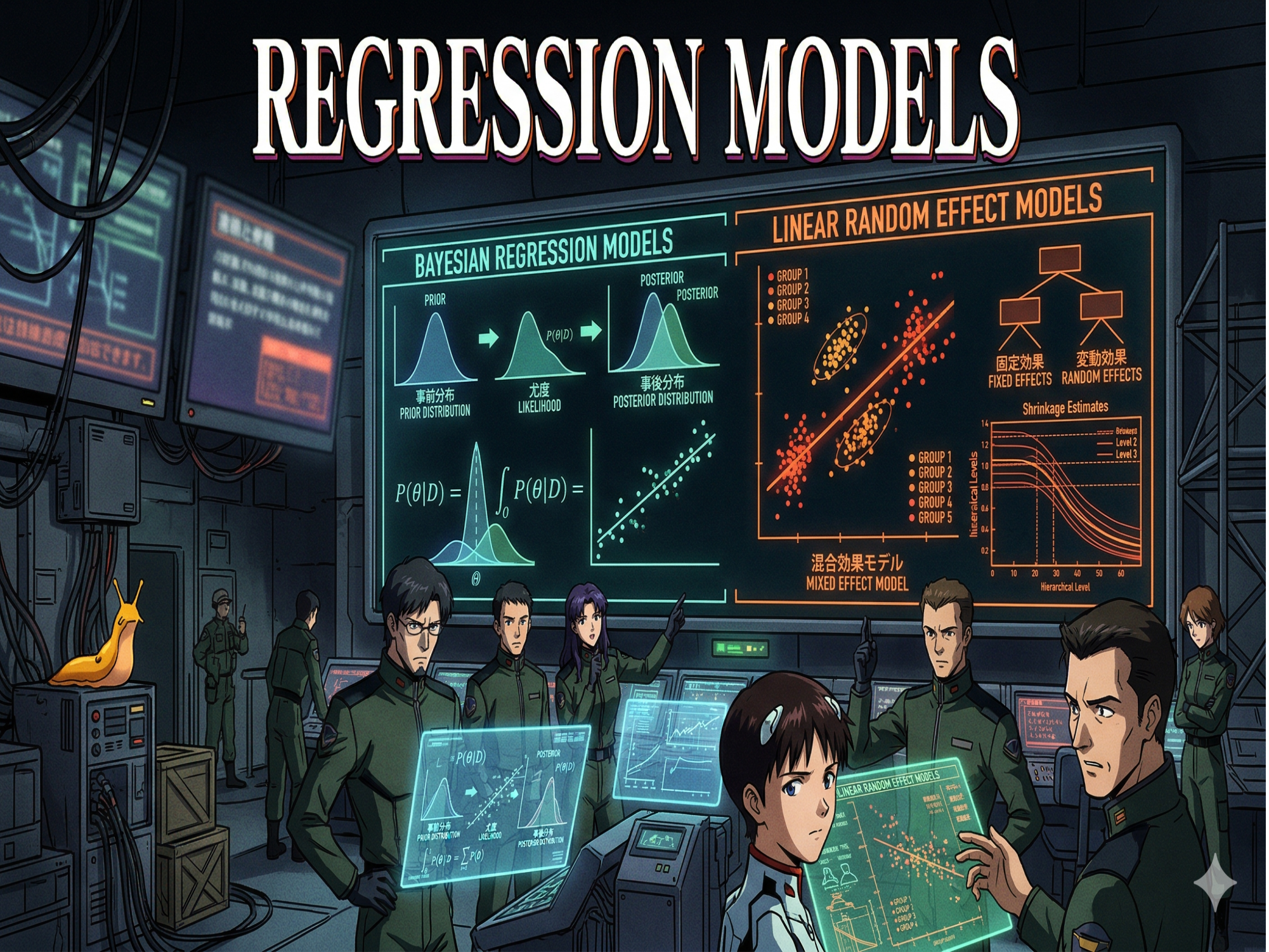
REGRESSION MODELS



Linear regression

- Denote $\boldsymbol{y} = (y_1, ..., y_n)$ the responses/outcomes.
- Denote $\boldsymbol{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_n)^T$ the $n \times p$ matrix of explanatory variables (potentially including the intercept).
- Consider the normal linear model with mean

$$\mathbb{E}(y_i|\boldsymbol{\beta}, \boldsymbol{X}_i) = \sum_{j=1}^{p} \beta_j x_{ij}$$

  and constant variance. Or equivalently in matrix notation that

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

  where $\boldsymbol{\epsilon} \sim N(0, \sigma \boldsymbol{I}_n)$.
- The maximum likelihood estimator of β is

$$\hat{\beta}_{MLE} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

Bayesian linear regression

- Treating $\boldsymbol{X}$ as fixed, consider the model

$$y_i|\boldsymbol{\beta}, \sigma \sim N(\boldsymbol{x}_i^T\boldsymbol{\beta}, \sigma^2), \quad i = 1, ..., n$$

- Assume a multivariate normal prior on β $\boldsymbol{\beta} \sim N(\boldsymbol{\mu}_0, V_0)$
- The full posterior distribution is

$$p(\boldsymbol{\beta}, \sigma^2|\boldsymbol{y}) \propto exp[-\frac{1}{2}\left((\boldsymbol{\beta}-\boldsymbol{\mu}_0)^T\mathbf{V}_0^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}_0) + \frac{1}{\sigma^2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})\right)]$$

- Expanding the quadratic terms, we get the posterior conditional

$$\begin{aligned}\boldsymbol{\beta}|\boldsymbol{y}, \sigma^2 &\sim N(\mu_n, V_n)\\ \mu_n &= V_n(V_0^{-1}\mu_0 + \frac{1}{\sigma^2}\boldsymbol{X}^T\boldsymbol{y})\\ V_n^{-1} &= V_0^{-1} + \frac{1}{\sigma^2}\boldsymbol{X}^T\boldsymbol{X}\end{aligned}$$

- In the limit as $V_0 \to \infty$, we have $V_n \to \sigma^2(\boldsymbol{X}^T\boldsymbol{X})^{-1}$ and $\mu_n \to \hat{\beta}_{MLE}$, which is the same as the frequentist sampling distribution of the MLE.

Choice of prior for β

- A conventional noninformative choice is $p(\boldsymbol{\beta}) \propto 1$.
- The normal prior leads to what is known as ridge regression. It allows us to make inference when $p > n$.
- When n is large and p is small, a relative diffuse prior typically works fine. Long tailed distributions (e.g., t) provides more robust inference for outliers.
- Many priors have been proposed to impose different properties (shrinkage, sparsity, group structures) on the regression coefficients when p is large.
- Be careful about the scale of X. Unlike the noninformative prior, the scale (and relative scares) of X matter in our models. E.g., $N(0, 1000)$ may not be diffuse at all if X are on the scale of 10^{-8}.
- It is usually worth rescaling the covariates to a scale that is easier to interpret. When there are multiple predictors, a common choice is to standardize columns of x to have zero mean and unit standard deviation

Conjugate prior on σ^2

- The conditionally conjugate prior for σ^2 is the familiar inverse Gamma prior, i.e., suppose

$$\sigma^2 \sim \text{Inv-Gamma}(\frac{v_0}{2}, \frac{v_0\sigma_0^2}{2})$$

- σ_0^2 is the prior guess of σ^2.
- v_0 is the strength of the prior information in units of sample size.
- The posterior is

$$\sigma^2|\boldsymbol{y}, \boldsymbol{\beta} \sim \text{Inv-Gamma}(\frac{v_0+n}{2}, \frac{v_0\sigma_0^2 + \sum_{i=1}^n (y_i - \boldsymbol{x}_i^T\boldsymbol{\beta})^2}{2})$$

- Similar to before, there are non-informative priors such as $p(\boldsymbol{\beta}, \sigma^2) \propto \sigma^{-2}$. See textbook for details.

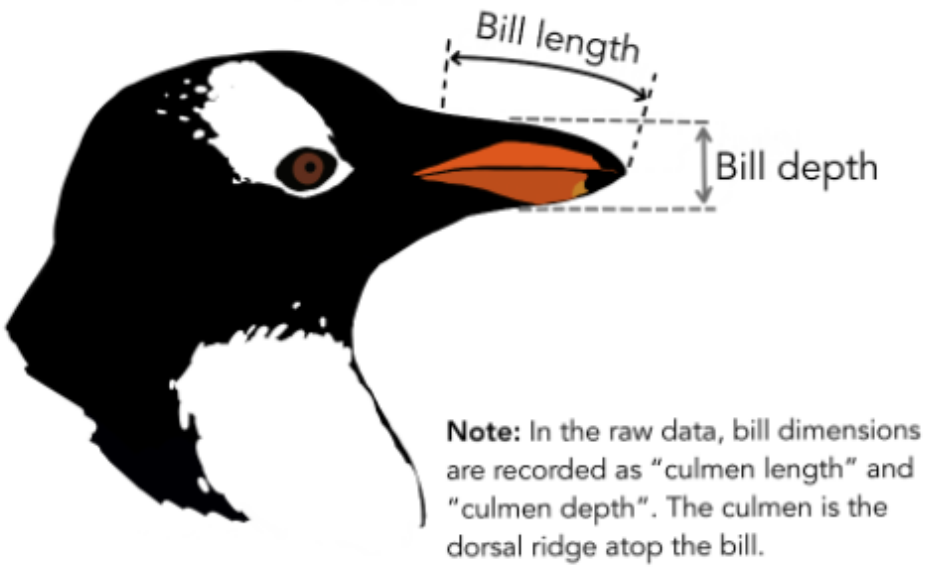
Posterior Predictive Distribution

- Suppose we are interested in making prediction of $\tilde{\boldsymbol{y}}$ given a new (or existing) set of $\tilde{\boldsymbol{X}}$.
- It is easy to draw posterior predictive samples of $\tilde{\boldsymbol{y}}$ from $\tilde{\boldsymbol{y}} \sim N(\tilde{\boldsymbol{X}}\beta, \sigma^2 \boldsymbol{I})$ with posterior draws of $(\boldsymbol{\beta}, \sigma^2)$.
- To gain more insight of the predictive uncertainty, consider marginalize out $\boldsymbol{\beta}$, then

$$
\begin{aligned}
\mathbb{E}(\tilde{\boldsymbol{y}}|\sigma^2, \boldsymbol{y}) &= \mathbb{E}(\mathbb{E}(\tilde{\boldsymbol{y}}|\boldsymbol{\beta}, \sigma^2, \boldsymbol{y})|\sigma^2, \boldsymbol{y}) \\
&= \tilde{\boldsymbol{X}}\mathbb{E}(\boldsymbol{\beta}|\sigma^2, \boldsymbol{y}) \\
\operatorname{var}(\tilde{\boldsymbol{y}}|\sigma^2, \boldsymbol{y}) &= \operatorname{var}(\mathbb{E}(\tilde{\boldsymbol{y}}|\boldsymbol{\beta}, \sigma^2, \boldsymbol{y})|\sigma^2, \boldsymbol{y}) + \mathbb{E}(\operatorname{var}(\tilde{\boldsymbol{y}}|\boldsymbol{\beta}, \sigma^2, \boldsymbol{y})|\sigma^2, \boldsymbol{y}) \\
&= \operatorname{var}(\tilde{\boldsymbol{X}}\boldsymbol{\beta}|\sigma^2, \boldsymbol{y}) + \mathbb{E}(\sigma^2 \boldsymbol{I}|\sigma^2, \boldsymbol{y}) \\
&= (\tilde{\boldsymbol{X}} \operatorname{var}(\boldsymbol{\beta}|\sigma^2, \boldsymbol{y})\tilde{\boldsymbol{X}}^T + I)\sigma^2
\end{aligned}
$$

Example: Palmer penguins

- As an illustration, let us look at the Palmer penguins dataset. It collects various measurements for 344 penguins in Palmer station in Antarctica.
- Let us build a linear model for the bill length.



Example: Palmer penguins

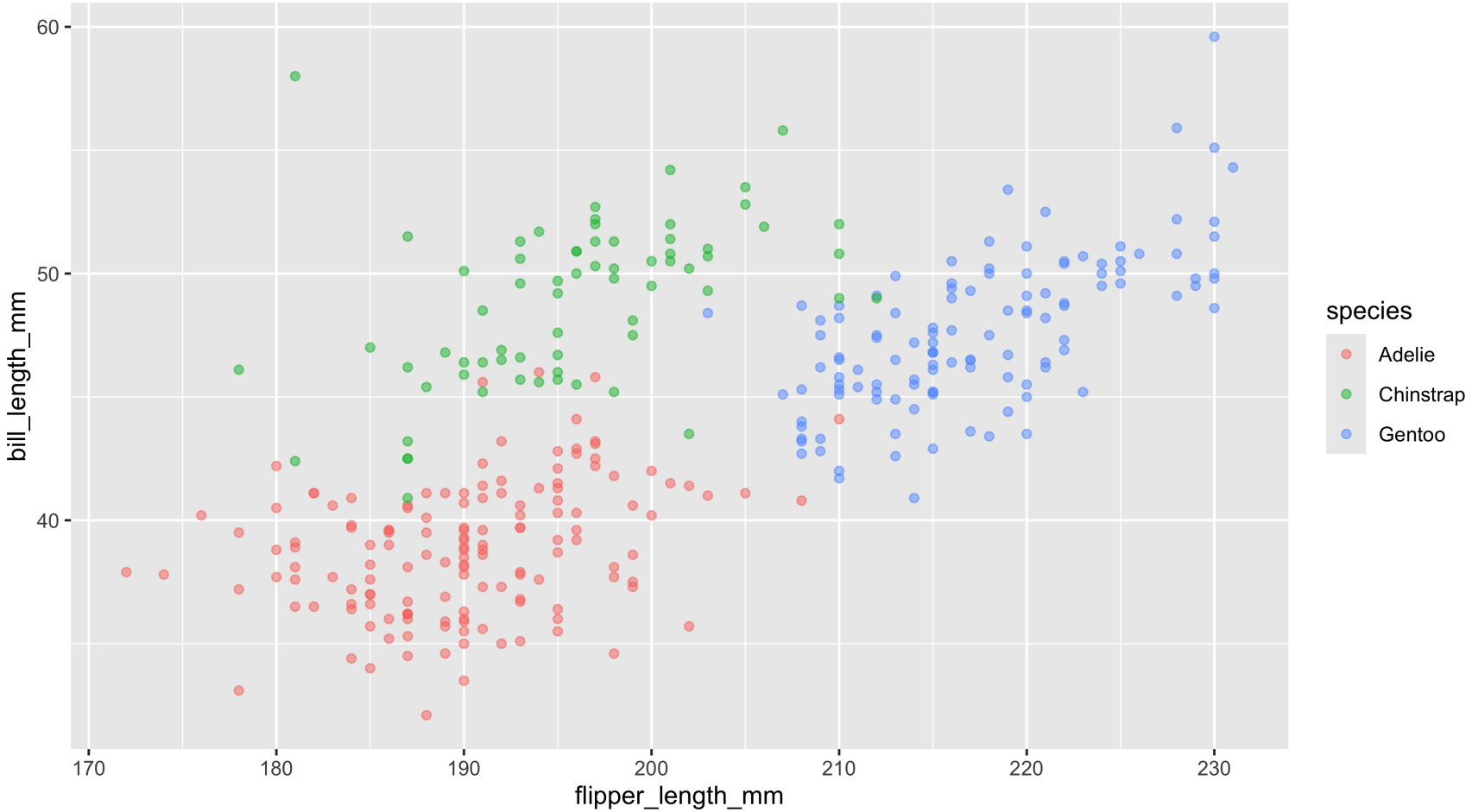
```
library(ggplot2)
library(patchwork)
library(kableExtra)
options(kable_styling_latex_options = "scale_down")

library(palmerpenguins)
kable(head(penguins)) %>% kable_styling()
```

species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
Adelie	Torgersen	39	19	181	3750	male	2007
Adelie	Torgersen	40	17	186	3800	female	2007
Adelie	Torgersen	40	18	195	3250	female	2007
Adelie	Torgersen	NA	NA	NA	NA	NA	2007
Adelie	Torgersen	37	19	193	3450	female	2007
Adelie	Torgersen	39	21	190	3650	male	2007

```
penguins <- subset(penguins, !is.na(flipper_length_mm) &
                             !is.na(bill_length_mm) &
                             !is.na(species) &
                             !is.na(sex))
```

Example: Palmer penguins

```
ggplot(penguins, aes(x = flipper_length_mm, y = bill_length_mm)) +
        geom_point(alpha = .5, aes(color = species))
```



Model 1: Linear regression, one covariate

First let us regress y only on the flipper length ignoring species difference. Here I used $p(\sigma^2) \propto 1/\sigma^2$, or equivalently setting $v_0 = 0$ and $\sigma_0^2 = 0$ so that I can reuse the previously derived posteriors.

```
library(bayesplot)
library(mvtnorm)

y <- penguins$bill_length_mm
x1 <- as.matrix(cbind(1,
                      penguins$flipper_length_mm))
# prior for beta
m <- rep(0, 2)
V <- diag(2) * 1000^2
# prior for sigma, improper
v0 <- 0
sigma2_0 <- 0
# initial value for sigma2
sigma2 <- 1
```

Model 1: Linear regression, one covariate

$$\beta|\boldsymbol{y}, \sigma^2 \sim N(\mu_n, V_n)$$

$$\mu_n = V_n(V_0^{-1}\mu_0 + \frac{1}{\sigma^2}\boldsymbol{X}^T\boldsymbol{y})$$

$$V_n^{-1} = V_0^{-1} + \frac{1}{\sigma^2}\boldsymbol{X}^T\boldsymbol{X}$$

$$\sigma^2|\boldsymbol{y}, \beta \sim \text{Inv-Gamma}(\frac{v_0 + n}{2}, \frac{v_0\sigma_0^2 + \sum_{i=1}^{n}(y_i - \boldsymbol{x}_i^T\boldsymbol{\beta})^2}{2})$$

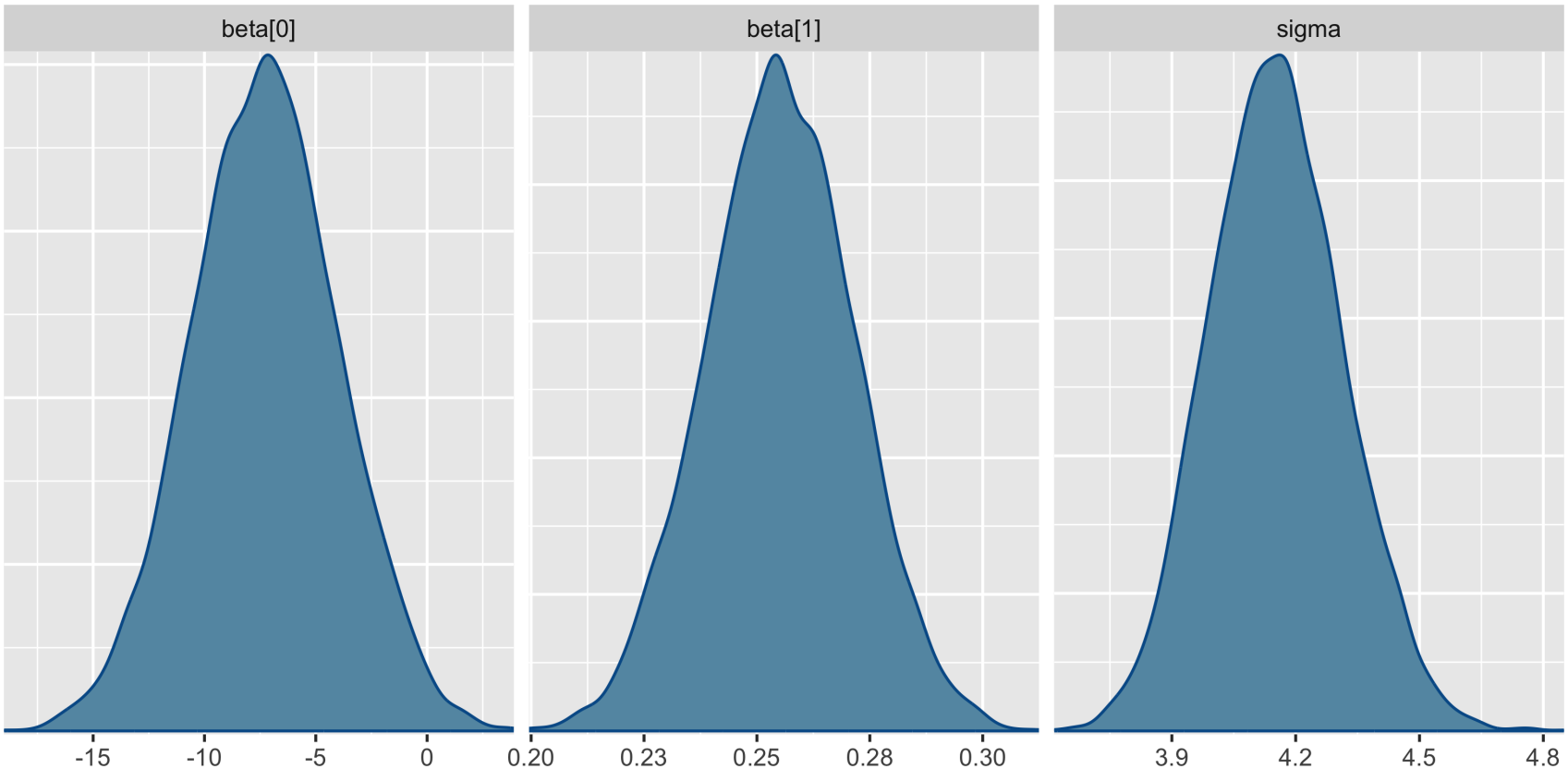
```
sample_beta <- function(x, y, m, V, sigma2){
    Vn <- solve(solve(V) + t(x) %*% x / sigma2)
    mn <- Vn %*% (solve(V) %*% m + t(x) %*% y / sigma2)
    beta <- rmvnorm(1, mn, Vn)
    return(beta)
}
sample_sigma2 <- function(x, y, beta, v0, s0){
    n <- length(y)
    sigma2 <- 1/rgamma(1, (v0+n)/2,
                       (v0*s0 + sum((y - x %*% t(beta))^2)/2))
    return(sigma2)
}
```

Model 1: Linear regression, one covariate

```
Nsim <- 10000
out1 <- matrix(NA, Nsim, dim(x1)[2] + 1)
colnames(out1) <- c("beta[0]", "beta[1]", "sigma")
loglik1 <- matrix(NA, Nsim, length(y))
for(i in 1:Nsim){
    beta <- sample_beta(x1, y, m, V, sigma2)
    sigma2 <- sample_sigma2(x1, y, beta, v0, sigma2_0)
    out1[i, ] <- c(beta, sqrt(sigma2))
    loglik1[i, ] <- dnorm(y, x1 %*% t(beta), sqrt(sigma2), log = TRUE)
}
```

Model 1: Linear regression, one covariate

```
mcmc_dens(out1[(Nsim/2):Nsim, ])
```



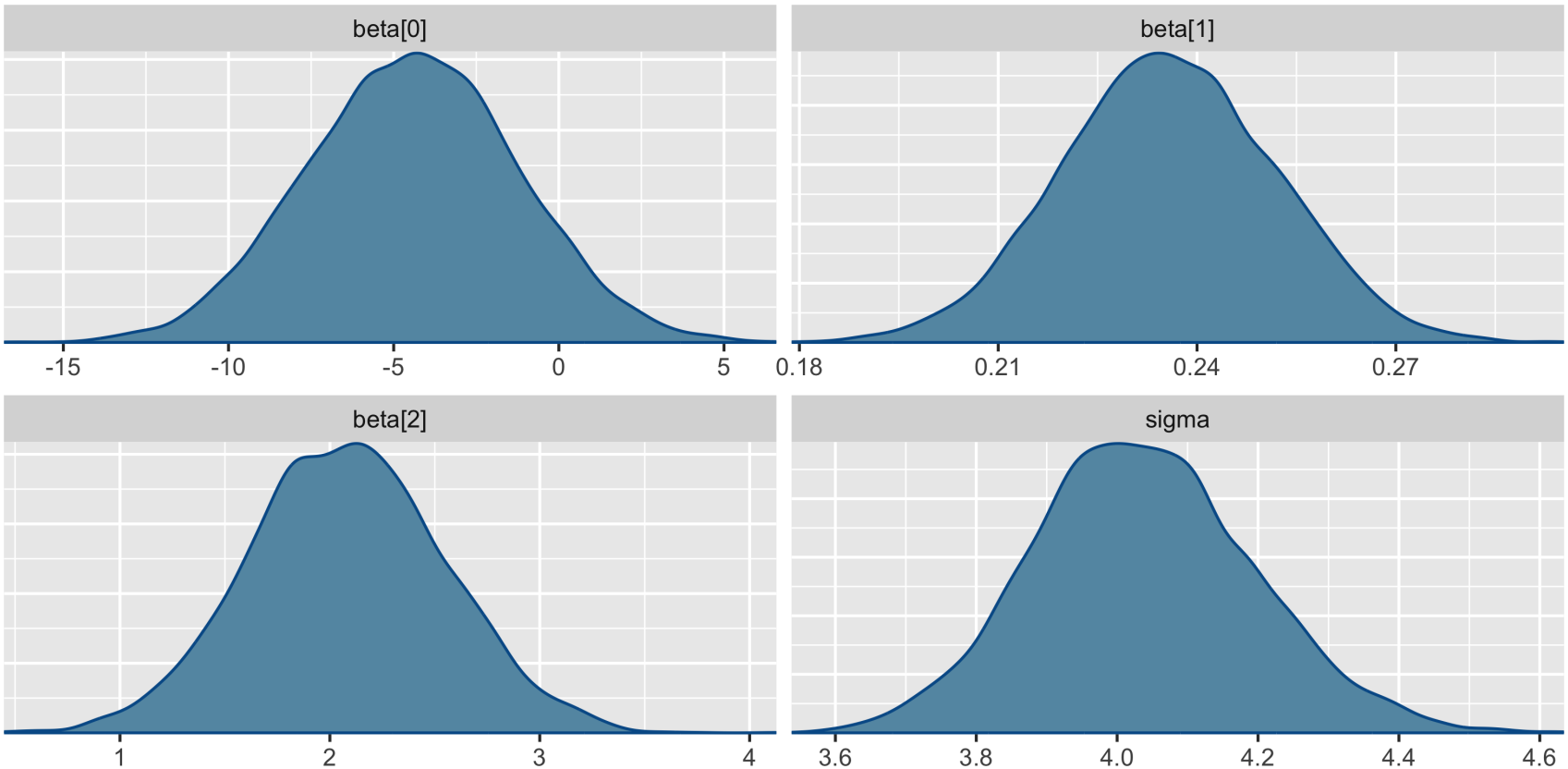
Model 2: Linear regression, two covariates

We now add one more covariate to the model.

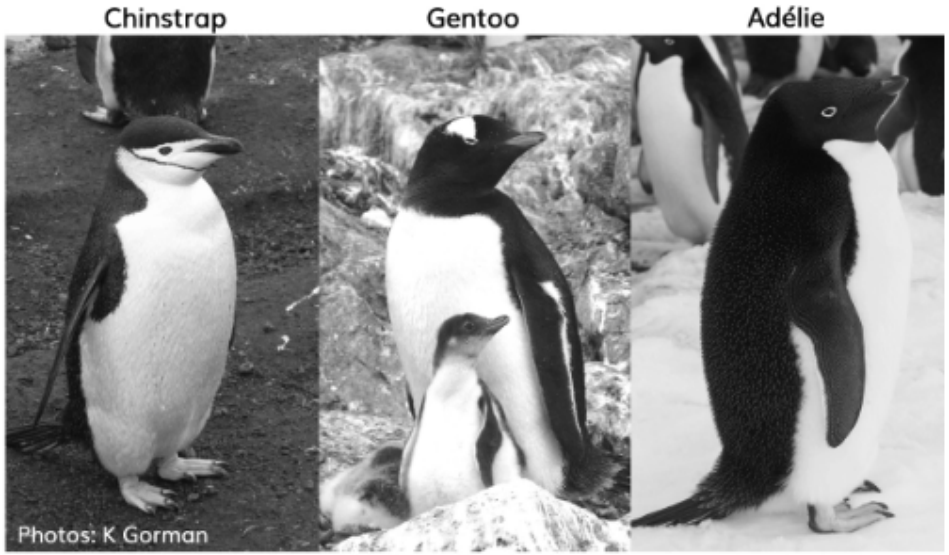
```
# Adding sex as covariate
x2 <- as.matrix(cbind(1,
                      penguins$flipper_length_mm,
                      as.numeric(penguins$sex == "male")))

out2 <- matrix(NA, Nsim, dim(x2)[2] + 1)
colnames(out2) <- c("beta[0]", "beta[1]", "beta[2]", "sigma")
loglik2 <- matrix(NA, Nsim, length(y))
# prior for beta
m <- rep(0, 3)
V <- diag(3) * 1000^2
for(i in 1:Nsim){
    beta <- sample_beta(x2, y, m, V, sigma2)
    sigma2 <- sample_sigma2(x2, y, beta, v0, sigma2_0)
    out2[i, ] <- c(beta, sqrt(sigma2))
    loglik2[i, ] <- dnorm(y, x2 %*% t(beta), sqrt(sigma2), log = TRUE)
}
```

Model 2: Linear regression, two covariates

```
mcmc_dens(out2[(Nsim/2):Nsim, ])
```



Example: Palmer penguins



Model 3: Linear regression, species fixed effect

We now add species-specific intercept to the model

```
# add fixed intercept for species
x.int <- model.matrix(~species - 1, data = penguins)
head(x.int)
```

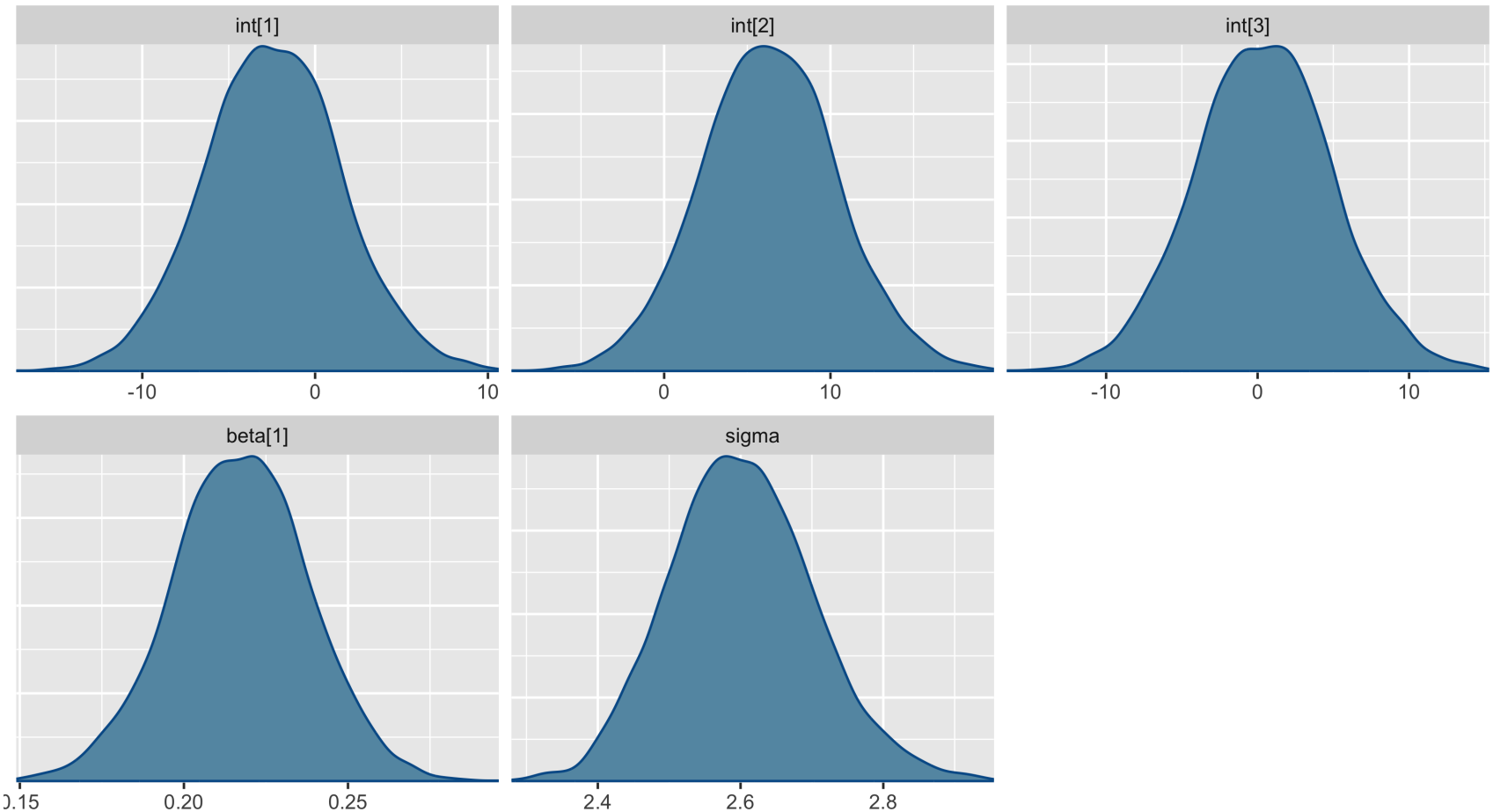
```
##   speciesAdelie speciesChinstrap speciesGentoo
## 1             1                0             0
## 2             1                0             0
## 3             1                0             0
## 4             1                0             0
## 5             1                0             0
## 6             1                0             0
```

Model 3: Linear regression, species fixed effect

```
x3 <- as.matrix(cbind(x.int,
                      penguins$flipper_length_mm))
out3 <- matrix(NA, Nsim, dim(x3)[2] + 1)
loglik3 <- matrix(NA, Nsim, length(y))
colnames(out3) <- c("int[1]", "int[2]", "int[3]", "beta[1]",  "sigma")

# prior for beta
m <- rep(0, dim(x3)[2])
V <- diag(dim(x3)[2]) * 1000^2
for(i in 1:Nsim){
    beta <- sample_beta(x3, y, m, V, sigma2)
    sigma2 <- sample_sigma2(x3, y, beta, v0, sigma2_0)
    out3[i, ] <- c(beta, sqrt(sigma2))
    loglik3[i, ] <- dnorm(y, x3 %*% t(beta), sqrt(sigma2), log = TRUE)
}
```

Model 3: Linear regression, species fixed effect

```
mcmc_dens(out3[(Nsim/2):Nsim, ])
```



Hierarchical regression

- Consider group $i = 1, ..., M$, observations $j = 1, ..., n_i$ in each group with $n = n_1 + ... + n_M$.
- A simple regression model with one covariate is

$$y_{ij} = \beta_0 + \beta_1 x_{ij} + \epsilon_{ij}$$

This is equivalent to

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where $\boldsymbol{y} = (y_{11}, ..., y_{1n_1}, ..., y_{M1}, ...y_{Mn_M})^T$ and

$$\boldsymbol{X} = [\mathbf{1}_{n\times 1}, \boldsymbol{x}], \quad \boldsymbol{\beta} = (\beta_0, \beta_1)^T$$

Hierarchical regression

- What if different groups have different intercept?

$$y_{ij} = \beta_{0i} + \beta_1 x_{ij} + \epsilon_{ij}$$

We can again write it into the matrix form by creating a new design matrix

$$\boldsymbol{y} = \tilde{\boldsymbol{X}}\tilde{\boldsymbol{\beta}} + \boldsymbol{\epsilon}$$

where $\boldsymbol{y} = (y_{11}, ..., y_{1n_1}, ..., y_{M1}, ...y_{Mn_M})^T$ and

$$\tilde{\boldsymbol{X}} = \begin{pmatrix} \mathbf{1}_{n_1 \times 1} & & \boldsymbol{x}_1 \\ & \ddots & \vdots \\ & & \mathbf{1}_{n_M \times 1} \quad \boldsymbol{x}_M \end{pmatrix}, \quad \tilde{\boldsymbol{\beta}} = \begin{pmatrix} \beta_{01} \\ \vdots \\ \beta_{0M} \\ \beta_1 \end{pmatrix}$$

- Note $\tilde{\boldsymbol{X}}$ is a $n \times (M+1)$ matrix

Hierarchical regression

- What if different groups have different intercept and slope?

$$y_{ij} = \beta_{0i} + \beta_{1i} x_{ij} + \epsilon_{ij}$$

We can again write it into the matrix form by creating a new design matrix

$$\boldsymbol{y} = \tilde{\boldsymbol{X}} \tilde{\boldsymbol{\beta}} + \boldsymbol{\epsilon}$$

where $\boldsymbol{y} = (y_{11}, ..., y_{1n_1}, ..., y_{M1}, ...y_{Mn_M})^T$ and

$$\tilde{\boldsymbol{X}} = \begin{pmatrix} \mathbf{1}_{n_1 \times 1} & & \boldsymbol{x}_1 & \\ & \ddots & & \ddots \\ & \mathbf{1}_{n_M \times 1} & & \boldsymbol{x}_M \end{pmatrix}, \quad \tilde{\boldsymbol{\beta}} = \begin{pmatrix} \beta_{01} \\ \vdots \\ \beta_{0M} \\ \beta_{11} \\ \vdots \\ \beta_{1M} \end{pmatrix}$$

- Note $\tilde{\boldsymbol{X}}$ is a $n \times (2M)$ matrix

Fixed and random effects

- In the previous three models, there are two types of regression coefficients, some are applied to all observations e.g., β_0, β_1, and some are applied to only some observations, e.g., β_{0i}, β_{1i}.
- For the latter case, we can consider that the group of parameters come from a common distribution, e.g.,

$$\beta_{0i} \sim N(\mu_0, \tau_0^2)$$

  And we can estimate τ_0^2 from the data. This is usually called *random effect*. The former case is usually referred to as *fixed effect*.
- The term *fixed effect* and *random effect* comes from the Frequentist literature. 'Random' means the parameter is considered random in Frequentist setting, as drawing from a population of such effects.
- They make less difference in Bayesian inference...

Random intercept model

- Returning to the earlier example, the random intercept model can be completed with

$$
\begin{aligned}
y_{ij} &= \beta_{0i} + \beta_1 x_{ij} + \epsilon_{ij} \\
\beta_{0i} &\sim N(\mu_0, \tau_0^2) \\
\beta_1 &\sim N(\mu_1, \tau_1^2) \\
\mu_0, \tau_0^2 &\sim p(\mu_0, \tau_0^2)
\end{aligned}
$$

- We typically do not put hyperpriors on the green terms, why?
- Note the prior in matrix form is

$$
\tilde{\boldsymbol{\beta}} = \begin{pmatrix} \beta_{01} \\ \vdots \\ \beta_{0M} \\ \beta_1 \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_0 \\ \vdots \\ \mu_0 \\ \mu_1 \end{pmatrix}, \begin{pmatrix} \tau_0^2 & & & \\ & \ddots & & \\ & & \tau_0^2 & \\ & & & \tau_1^2 \end{pmatrix} \right)
$$

Random slope model

- Returning to the earlier example, the random slope model can be completed with

$$
\begin{aligned}
y_{ij} &= \beta_{0i} + \beta_1 x_{ij} + \epsilon_{ij} \\
\beta_{0i} &\sim N(\mu_0, \tau_0^2) \\
\beta_{1i} &\sim N(\mu_1, \tau_1^2) \\
\mu_0, \tau_0^2 &\sim p(\mu_0, \tau_0^2) \\
\mu_1, \tau_1^2 &\sim p(\mu_1, \tau_1^2)
\end{aligned}
$$

- Note the prior in matrix form is

$$
\tilde{\boldsymbol{\beta}} = \begin{pmatrix} \boldsymbol{\beta}_0 \\ \boldsymbol{\beta}_1 \end{pmatrix} \sim N(\begin{pmatrix} \boldsymbol{\mu}_0 \\ \boldsymbol{\mu}_1 \end{pmatrix}, \begin{pmatrix} \tau_0^2 \boldsymbol{I}_M & \\ & \tau_1^2 \boldsymbol{I}_M \end{pmatrix})
$$

Choice of prior on random effects

- The somewhat 'default' prior for the mean and variance parameters are

$$\mu_0 \sim N(0, \phi_0)$$

$$\tau_0^2 \sim \text{Inv-Gamma}(a_0, b_0)$$

- A large ϕ_0 (or even $p(\mu_0) \propto 1$) is usually reasonable, as it can be considered as the prior for a fixed effect.

- Recall that if $x|\sigma^2 \sim N(0, \sigma^2)$ and $\sigma^2 \sim \text{Inv-Gamma}(v/2, vs^2/2)$, then $x \sim t_v(0, s^2)$, i.e., $x/s \sim t_v$.

- Thus we can understand the effect of priors on τ_0^2 by its induced prior on μ_0. if we want the 95% prior probability for the intercept to be between $\pm E$ from their mean, and fix $v = 1$, we get $s = \frac{E}{t_{1,0.975}}$ and the prior on τ_0 is Inv-Gamma$(\frac{1}{2}, \frac{E^2}{2t_{1,0.975}^2})$.

- For example, when $E = 0.2$, the corresponding prior is Inv-Gamma$(0.5, 0.000124)$. Similar reasoning can be made for τ_1^2.

- For non-conjugate choices, see Simpson et al (2014) for penalising complexity priors.

Linear mixed model: implementation

- We now have seen that the different hierarchical regression models can be written in the same format of

$$\boldsymbol{y} \sim N(\tilde{\boldsymbol{X}}\tilde{\boldsymbol{\beta}}, \sigma^2 \boldsymbol{I})$$
$$\tilde{\boldsymbol{\beta}} \sim N(\boldsymbol{m}, \boldsymbol{D})$$

with different specification of $\boldsymbol{D}$. Take the random slope model with one covariate for example, $\boldsymbol{y} = (y_{11}, ..., y_{1n_1}, ..., y_{M1}, ...y_{Mn_M})^T$ and

$$\tilde{\boldsymbol{X}} = \begin{pmatrix} \mathbf{1}_{n_1 \times 1} & & & \boldsymbol{x}_1 & & \\ & \ddots & & & \ddots & \\ & & \mathbf{1}_{n_M \times 1} & & & \boldsymbol{x}_M \end{pmatrix}, \quad \tilde{\boldsymbol{\beta}} = \begin{pmatrix} \beta_{01} \\ \vdots \\ \beta_{0M} \\ \beta_{11} \\ \vdots \\ \beta_{1M} \end{pmatrix}$$

$$\boldsymbol{m} = \begin{pmatrix} \boldsymbol{\mu}_0 \\ \boldsymbol{\mu}_1 \end{pmatrix}, \quad \boldsymbol{D} = \begin{pmatrix} \tau_0^2 \boldsymbol{I}_M & \\ & \tau_1^2 \boldsymbol{I}_M \end{pmatrix}$$
$$\mu_0, \mu_1 \sim N(0, \phi)$$
$$\tau_j^2 \sim \text{Inv-Gamma}(a_j, b_j), \quad j = 1, 2$$

Linear mixed model: implementation

1. Sample $\tilde{\beta}|\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{X}}, \boldsymbol{D}, \sigma^2$ following the same step in the standard regression case.

2. Sample

$$\mu_0|\{\beta_{0i}\}, \tau_0 \sim N(\frac{\sum_{i=1}^m \beta_{0i}/\tau^2}{1/\phi^2 + m/\tau_0^2}, \frac{1}{1/\phi^2 + m/\tau_0^2})$$

$$\mu_1|\{\beta_{1i}\}, \tau_1 \sim N(\frac{\sum_{i=1}^m \beta_{1i}/\tau^2}{1/\phi^2 + m/\tau_1^2}, \frac{1}{1/\phi^2 + m/\tau_1^2})$$

3. Sample

$$\tau_0^2|\{\beta_{0i}\}, \mu_0 \sim \text{Inv-Gamma}(a_0 + \frac{m}{2}, b_0 + \frac{1}{2}\sum_{i=1}^m (\beta_{0i} - \mu_0)^2)$$

$$\tau_1^2|\{\beta_{1i}\}, \mu_1 \sim \text{Inv-Gamma}(a_0 + \frac{m}{2}, b_1 + \frac{1}{2}\sum_{i=1}^m (\beta_{1i} - \mu_1)^2)$$

Similarly, for the random intercept only model, you only need to remove some of the steps here.

Model 4: Linear mixed model, species random effect

Let y_{ij} denote the bill length of the j-th penguin from the i-th species, and x_{ij1} and x_{ij2}denote the flipper length and sex (male = 1, female = 0) of the j-th penguin from the i-th species. Consider

$$y_{ij} = \beta_{0i} + \beta_1 x_{ij1} + \beta_2 x_{ij2} + \epsilon_{ij}$$
$$\beta_{0i} \sim N(\mu_0, \tau_0^2)$$

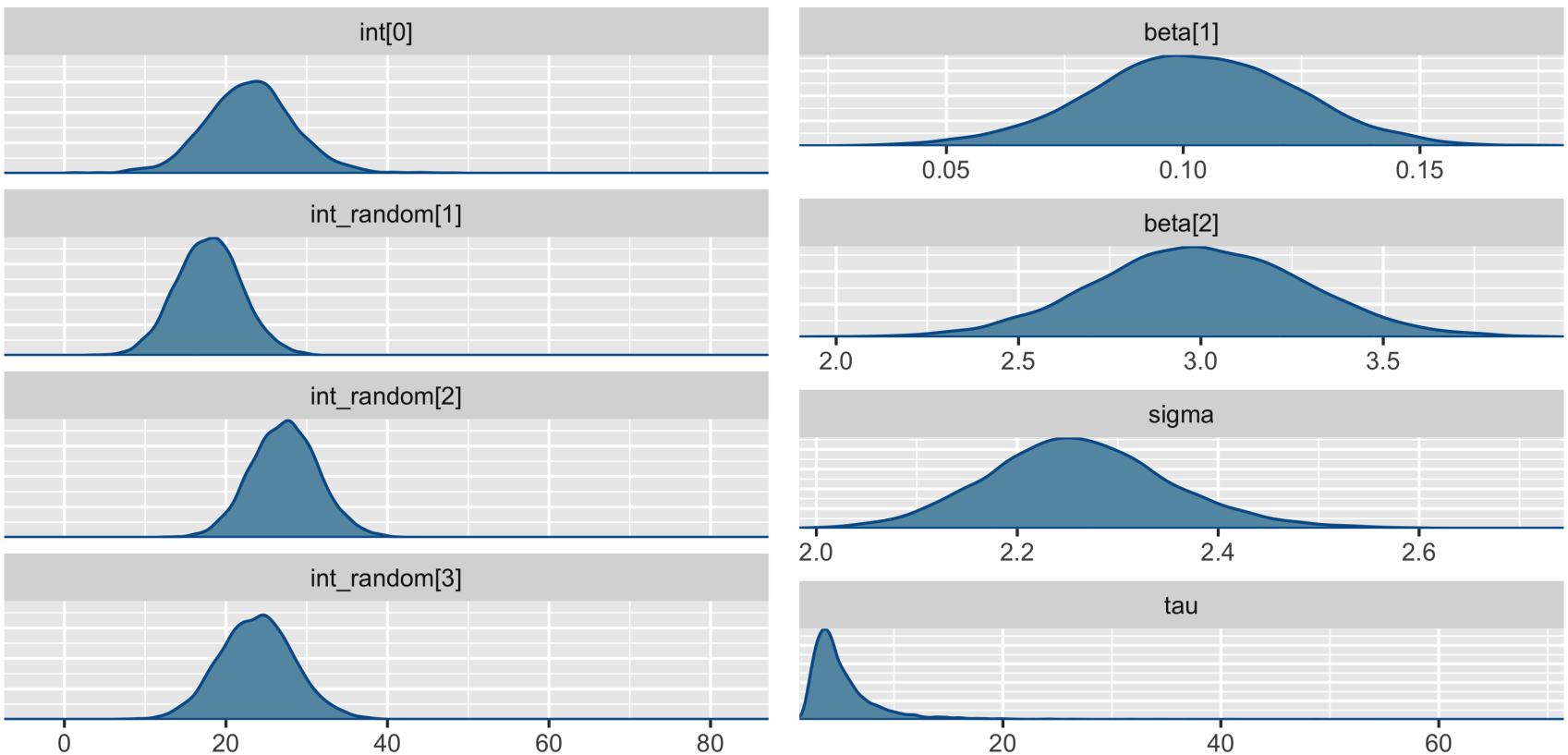
```
update_mu <- function(beta_r, tau2, phi2){
    m <- length(beta_r)
    V <- 1/(1/phi2 + m/tau2)
    mu <- rnorm(1, sum(beta_r)/tau2 * V, sqrt(V))
    return(mu)
}
update_tau2 <- function(a, b, beta_r, mu){
    m <- length(beta_r)
    tau2 <- 1/rgamma(1, a + m/2, b + sum((beta_r - mu)^2)/2)
    return(tau2)
}
```

Model 4: Linear mixed model, species random effect

```
x4 <- as.matrix(cbind(x.int,
                      penguins$flipper_length_mm,
                      as.numeric(penguins$sex == "male")))
out4 <- matrix(NA, Nsim, 8)
colnames(out4) <- c("int[0]", "int_random[1]", "int_random[2]",
        "int_random[3]", "beta[1]", "beta[2]", "sigma", "tau")
loglik4 <- matrix(NA, Nsim, length(y))

b <- 15^2 / 2 / qt(0.975, df = 1)^2
m <- rep(0, dim(x4)[2])
V <- diag(c(1, 1, 1, 1000^2, 1000^2))
tau2 <- 1
for(i in 1:Nsim){
    beta <- sample_beta(x4, y, m, V, sigma2)
    sigma2 <- sample_sigma2(x4, y, beta, v0, sigma2_0)
    mu0 <- update_mu(beta[1:3], tau2, 1000^2)
    m[1:3] <- mu0
    tau2 <- update_tau2(a = 0.5, b = b, beta[1:3], mu0)
    V <- diag(c(rep(tau2, 3), 1000^2, 1000^2))
    out4[i, ] <- c(mu0, beta, sqrt(sigma2), sqrt(tau2))
    loglik4[i, ] <- dnorm(y, x4 %*% t(beta), sqrt(sigma2), log = TRUE)
}
```

Model 4: Linear mixed model, species random effect

```
g1 <- mcmc_dens(out4[(Nsim/2):Nsim, ],
        pars = c("int[0]", "int_random[1]", "int_random[2]", "int_random[3]"),
        facet_args = list(scales = 'fixed', ncol = 1))
g2 <- mcmc_dens(out4[(Nsim/2):Nsim, ],
     pars = c("beta[1]", "beta[2]", "sigma", "tau"),
     facet_args = list(ncol = 1))
g1 + g2
```



Model 4: Stan implementation

```
library(rstan)
m1 <- " data {
          int N;            // number of observations
          int M;            // number of groups
          int ID[N];        // index of groups
          vector[N] x1;      // covariate
          vector[N] x2;      // covariate
          vector[N] y;       // outcome
        }
        parameters {
          vector[2] beta;  // fixed effects
          vector[M] mu;    // random effects
          real mu0;       // mean of random intercepts
          real<lower=0> sigma;  // sd of y
          real<lower=0> tau2;  // sd of mu
        }
        transformed parameters{
            real<lower=0> tau = sqrt(tau2);
        }
        model {
          real yhat;
          beta[1] ~ normal(0, 1000); beta[2] ~ normal(0, 1000);
          mu0 ~ normal(0, 1000);
          for(i in 1:M) mu[i] ~ normal(mu0, sqrt(tau2));
          // uniform priors on sigma, inv gamma prior on tau2
          tau2 ~ inv_gamma(0.5, 0.6968203);
          for(i in 1:N){
            yhat= mu[ID[i]] + beta[1] * x1[i] + beta[2] * x2[i];
            y[i] ~ normal(yhat, sigma);
          }
        }"
```

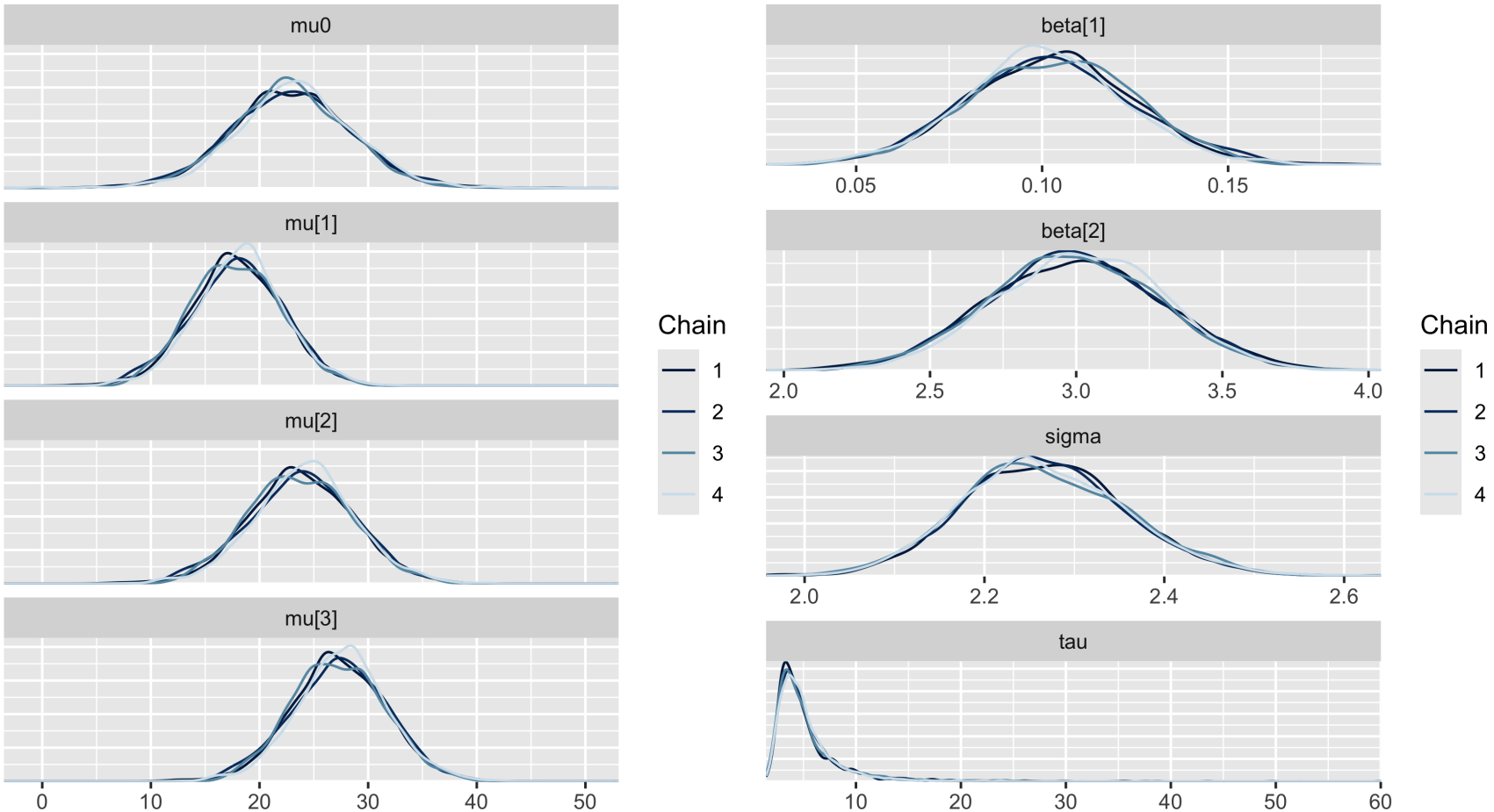
Model 4: Stan implementation

```
standata <- list(N = dim(penguins)[1],
                 M = 3,
                 ID = match(penguins$species, unique(penguins$species)),
                 x1 = penguins$flipper_length_mm,
                 x2 = as.numeric(penguins$sex == "male"),
                 y = penguins$bill_length_mm)
fit.stan <- stan(model_code = m1,
                     data = standata,
                     iter=4000, chains = 4)
```

```
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 6.5e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.65 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 4000 [  0%]  (Warmup)
## Chain 1: Iteration:  400 / 4000 [ 10%]  (Warmup)
## Chain 1: Iteration:  800 / 4000 [ 20%]  (Warmup)
## Chain 1: Iteration: 1200 / 4000 [ 30%]  (Warmup)
## Chain 1: Iteration: 1600 / 4000 [ 40%]  (Warmup)
## Chain 1: Iteration: 2000 / 4000 [ 50%]  (Warmup)
## Chain 1: Iteration: 2001 / 4000 [ 50%]  (Sampling)
## Chain 1: Iteration: 2400 / 4000 [ 60%]  (Sampling)
## Chain 1: Iteration: 2800 / 4000 [ 70%]  (Sampling)
## Chain 1: Iteration: 3200 / 4000 [ 80%]  (Sampling)
## Chain 1: Iteration: 3600 / 4000 [ 90%]  (Sampling)
## Chain 1: Iteration: 4000 / 4000 [100%]  (Sampling)
## Chain 1:
```

Model 4: Stan implementation

```
g1 <- mcmc_dens_overlay(fit.stan,
            pars = c("mu0", "mu[1]", "mu[2]", "mu[3]"),
            facet_args = list(scales = 'fixed', ncol = 1))
g2 <- mcmc_dens_overlay(fit.stan,
            pars = c("beta[1]", "beta[2]", "sigma", "tau"),
            facet_args = list(ncol = 1))
g1 + g2
```



Model 4: Stan implementation

```
summary(fit.stan)$summary
```

```
##           mean se_mean     sd     2.5%      25%    50%      75%    97.5% n_eff
## beta[1]    0.1 0.00057  0.022    0.058    0.087    0.1    0.12    0.15  1488
## beta[2]    3.0 0.00522  0.290    2.432    2.802    3.0    3.20    3.56  3082
## mu[1]     18.0 0.10706  4.144    9.793   15.217   18.0   20.73   26.14  1498
## mu[2]     23.9 0.12282  4.746   14.470   20.739   23.9   27.05   33.26  1493
## mu[3]     27.3 0.11053  4.283   18.898   24.503   27.3   30.21   35.76  1501
## mu0       23.0 0.13777  5.578   11.913   19.458   23.0   26.49   33.84  1639
## sigma      2.3 0.00127  0.091    2.098    2.204    2.3    2.33    2.45  5070
## tau2      39.3 1.89490 90.778    4.868   11.120   19.3   36.43  195.91  2295
## tau        5.3 0.07413  3.370    2.206    3.335    4.4    6.04   14.00  2067
## lp__    -445.3 0.04686  2.163 -450.558 -446.443 -444.9 -443.70 -442.13  2130
##         Rhat
## beta[1]    1
## beta[2]    1
## mu[1]      1
## mu[2]      1
## mu[3]      1
## mu0        1
## sigma      1
## tau2       1
## tau        1
## lp__       1
```

Model checking/criticism

- We have already fitted four regression models to a simple problem. Usually many probabilistic model provide reasonable fit to the data with varying levels of complexity. It is crucial to check your model and assess whether it is adequate.
- Sensibility: Does my model make sense? Does it align with domain knowledge or common sense?
  - Does my prior make sense? e.g., evaluating prior predictive distribution.
  - Does my data model make sense? e.g., does it capture relevant features of the data?
  - Does my inference procedure leads to reasonble results?
- Does the model fit the data adequatly?
  - We will focus on the posterior predictive check (Ch 6)
- Does the model overfit the data? How do we compare multiple models that fit the data reasonably well?
  - We will focus on various information criterion (Ch 7)

Posterior predictive check

- The main idea of posterior predictive check is to simulate replicate y^{rep} from the posterior distribution.

$$p(\boldsymbol{y}^{rep}|\boldsymbol{y}) = \int p(\boldsymbol{y}^{rep}|\boldsymbol{y},\theta)p(\theta|\boldsymbol{y})d\theta$$

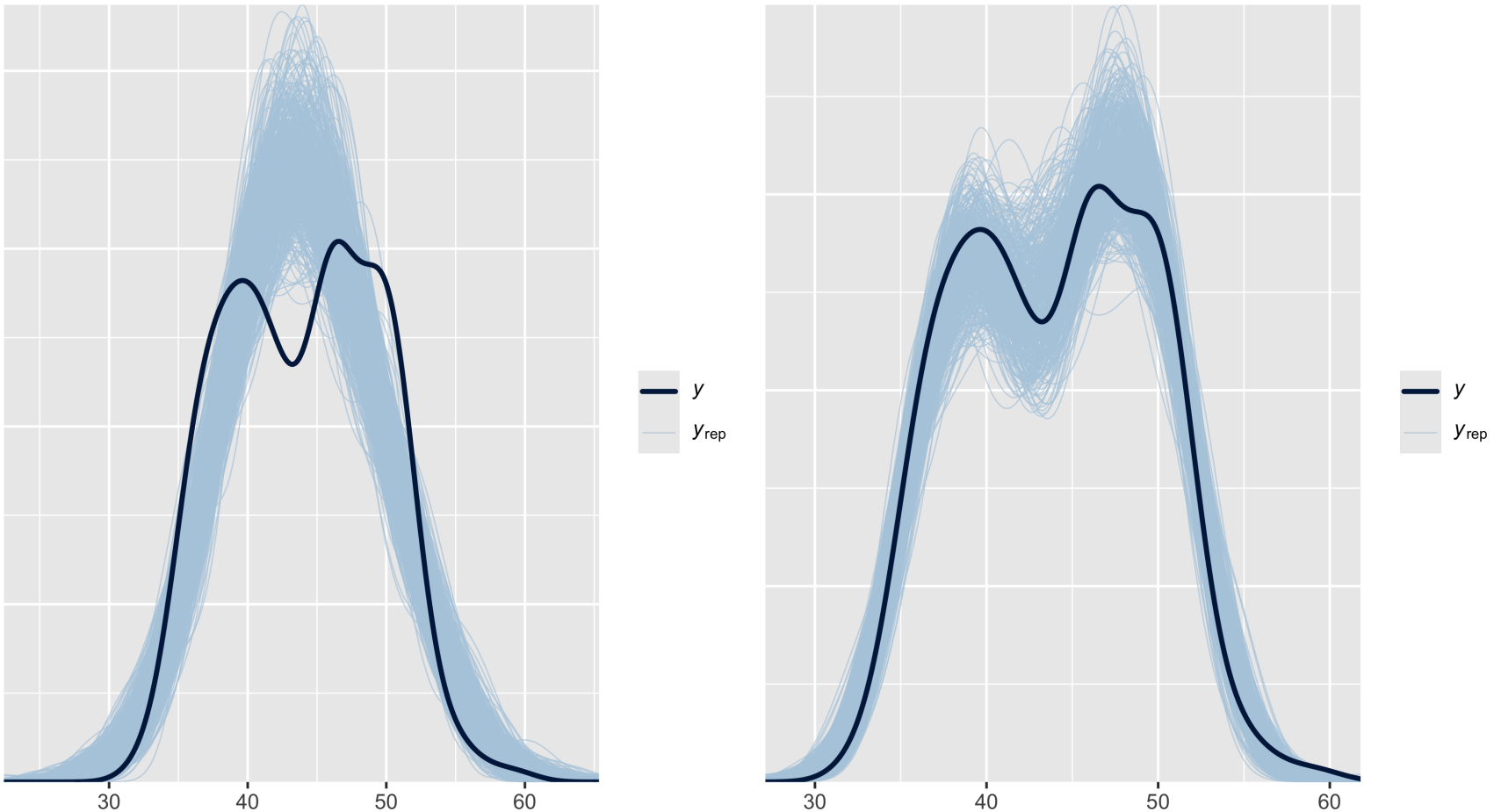
  - Draw $\theta^{(s)}$ from posterior $p(\theta|\boldsymbol{y})$.
  - Draw $\boldsymbol{y}^{rep(s)}|\theta^{(s)}$ from $p(\boldsymbol{y}|\theta)$.
- Define a test statistic $T(\boldsymbol{y})$ (or more generally $T(\boldsymbol{y},\theta)$).
  - E.g., do we want the predictive distribution of $\boldsymbol{y}|\theta_j$ or $(\boldsymbol{y},\theta_{J+1})|\phi$?
- Compute the distribution of $T(\boldsymbol{y}^{rep(s)})$, and compare with the observed $T(\boldsymbol{y})$.
- Similarly, we can assess whether the prior is too extreme by replacing the posterior distribution $p(\theta|\boldsymbol{y})$ with the prior $p(\theta)$, as we have implicitly carried out before. This is called prior predictive check.

Example: Posterior predictive check

```
sim_new <- function(x, beta, sigma2){
    nsim <- dim(beta)[1]
    n <- dim(x)[1]
    draws <- matrix(NA, nsim, n)
    for(i in 1:nsim){
        draws[i, ] <- rnorm(n, x %*% beta[i, ], sigma2[i])
    }
    return(draws)
}

toplot <- (Nsim - 500) : Nsim
yrep1 <- sim_new(x1, out1[toplot, 1:2], out1[toplot, 3])
yrep4 <- sim_new(x4, out4[toplot, 2:6], out4[toplot, 7])
```

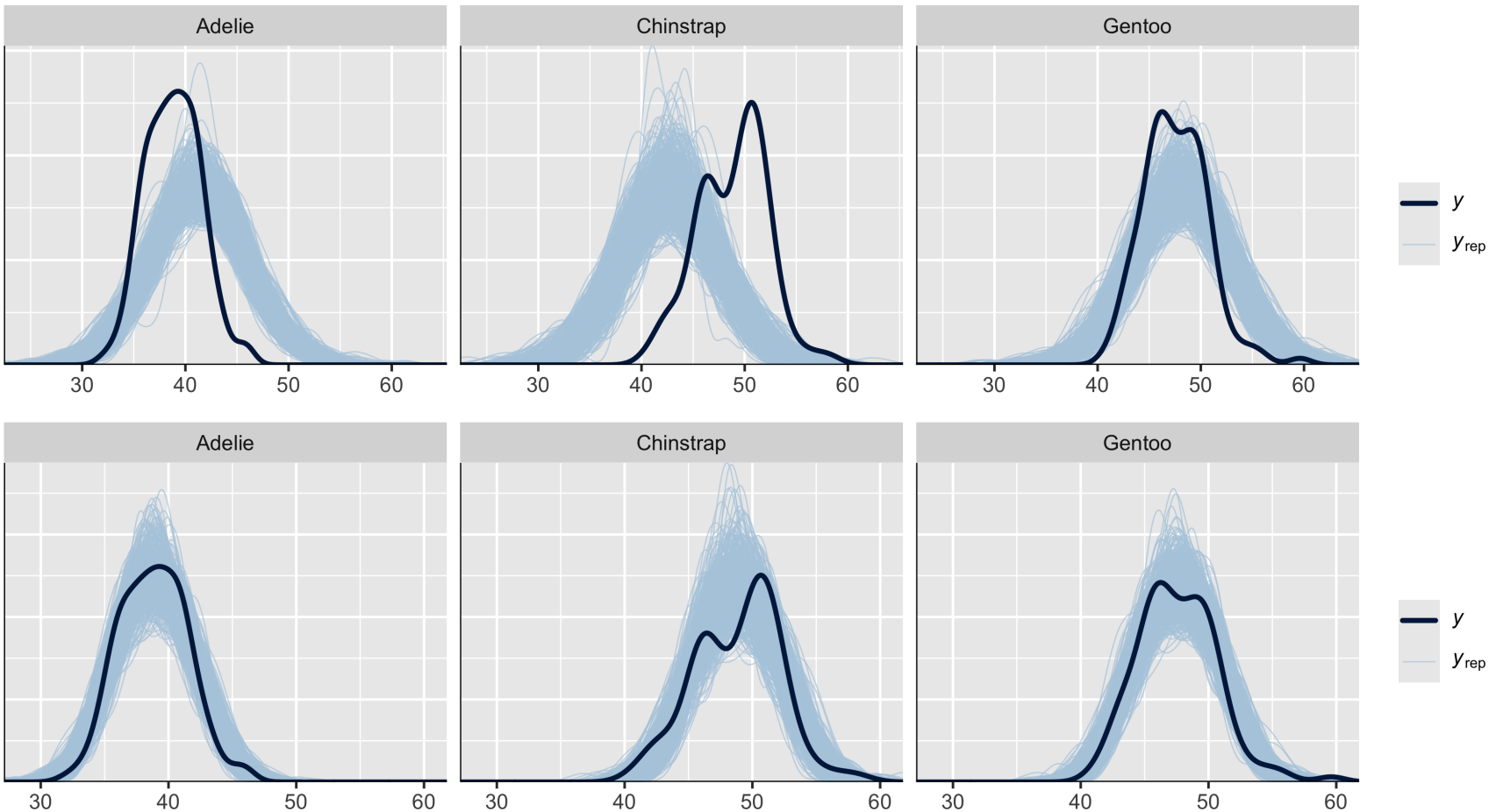
Example: Posterior predictive check

```
g1 <- ppc_dens_overlay(y, yrep1)
g2 <- ppc_dens_overlay(y, yrep4)
g1 + g2
```



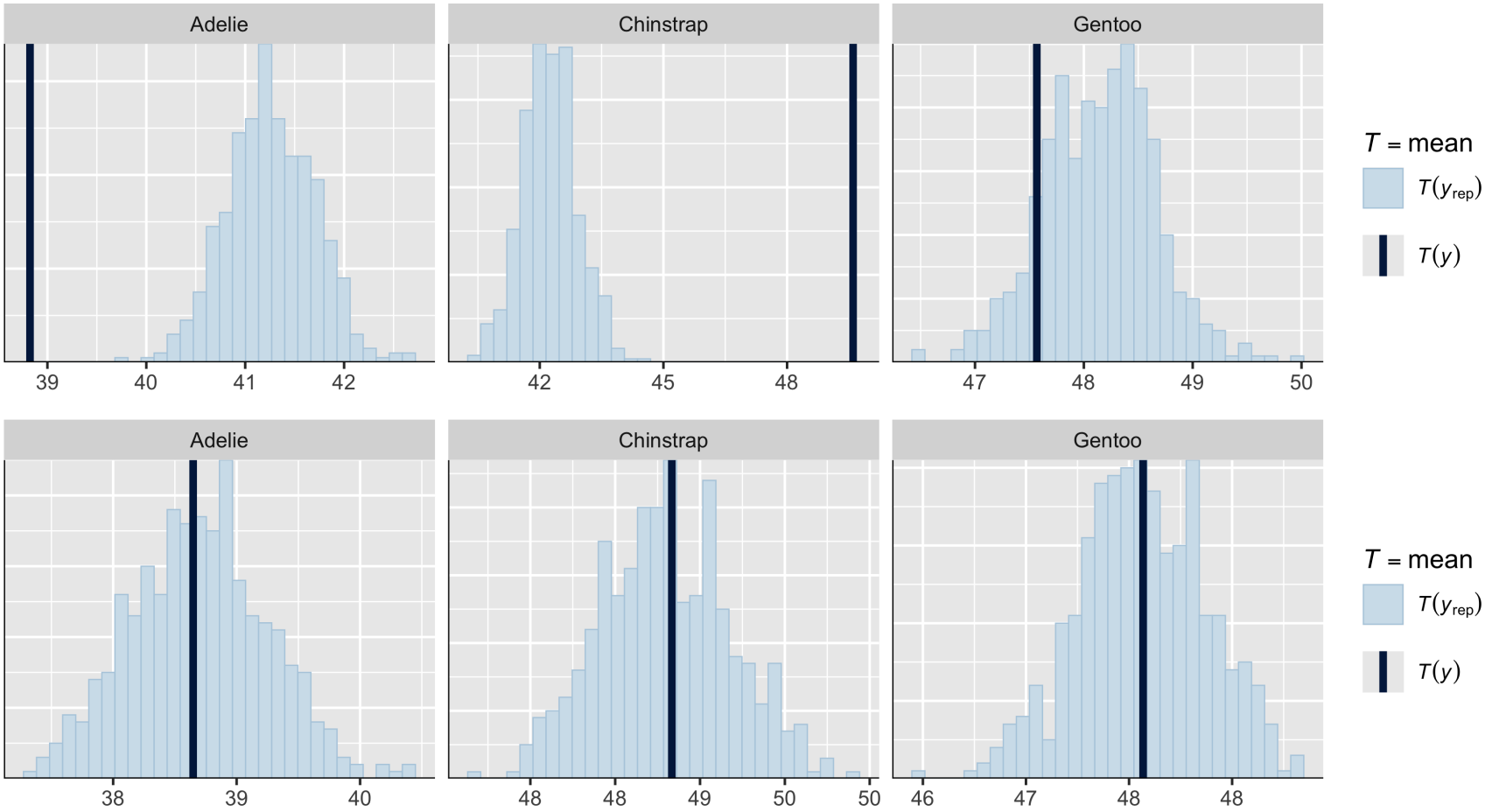
Example: Posterior predictive check

```
g1 <- ppc_dens_overlay_grouped(y, yrep1, group = penguins$species)
g2 <- ppc_dens_overlay_grouped(y, yrep4, group = penguins$species)
g1 / g2
```



Example: Posterior predictive check

```
g1 <- ppc_stat_grouped(y, yrep1, group = penguins$species,
                       stat = c("mean"))
g2 <- ppc_stat_grouped(y, yrep4, group = penguins$species,
                       stat = c("mean"))
g1 / g2
```




```
##################################################
## Different model comparison tools
```

Out-of-sample prediction

- We now turn to the predictive performance as a measure to evaluate and compare multiple models.
- Consider data $y_1, ..., y_n$ from a model $p(\boldsymbol{y}|\theta) = \prod_i^n p(y_i|\theta)$.
- The *expected* log pointwise predictive density for new data points $\tilde{y}_1, ..., \tilde{y}_n$ is

$$elppd = \mathbb{E}(\log p(\tilde{\boldsymbol{y}}|\boldsymbol{y})) = \sum_i \int \log p(\tilde{y}_i|\boldsymbol{y}) f(\tilde{y}_i) d\tilde{y}_i$$

  where $f(\tilde{y}_i)$ is the distribution representing the true data generating process for $\tilde{\boldsymbol{y}}$.
- A slightly different accuracy measure to $elppd$ is to further conditioning on a point estimate $\hat{\theta}$,

$$elppd_{\hat{\theta}} = \mathbb{E}(\log p(\tilde{\boldsymbol{y}}|\hat{\theta})) = \sum_i \int \log p(\tilde{y}_i|\hat{\theta}) f(\tilde{y}_i) d\tilde{y}_i$$

  where $\hat{\theta}$ is obtained by fitting a model on $\boldsymbol{y}$.
- We want a model to have high $elppd$. Unfortunately, $f(\tilde{y}_i)$ is unknown.

In-sample prediction

- A related quantity that we can compute is the log pointwise predictive density

$$lppd = \sum_i \log p(y_i|\boldsymbol{y}) = \sum_i \log \int p(y_i|\theta)p(\theta|\boldsymbol{y})d\theta = \sum_i \log \mathbb{E}_{\theta|\boldsymbol{y}}(p(y_i|\theta))$$

  This can be approximated with S posterior draws by

$$\widehat{lppd} = \sum_i \log(\frac{1}{S}\sum_{s=1}^{S} p(y_i|\theta^s))$$

- The $lppd$ evaluates the density on the data used to fit the model, thus is an overestimate of the elppd.

AIC

- Consider a regression model with k covariates and known σ^2,

$$y_i = \sum_{j=1}^{k} \theta_j x_{ij} + \epsilon_{ij}$$

  with the variance of noise term known.

- AIC (an Information criterion or Akaike information criterion) estimates $elppd_{\hat{\theta}}$ for a point estimate of parameter, typically the maximum likelihood estimator, by

$$elppd_{\hat{\theta}}^{AIC} = \mathbb{E}(\log p(\tilde{\boldsymbol{y}}|\hat{\theta})) \approx \sum_{i=1}^{N} \log p(y_i|\hat{\theta}) - k$$

  where the expectation is taken over $p(\tilde{y})$ and k is the dimension of θ.

- The approximation comes from the asymptotic normal posterior distribution as $n \to \infty$.

- More commonly, AIC is defined as this quantity multiplied by -2, thus we prefer model with smaller AIC

$$AIC = -2 \log p(\boldsymbol{y}|\hat{\theta}) + 2k$$

DIC

- AIC is difficult to use for hierarchical models, as the number of parameters (or degree of freedom) is not straightforward to determine.
- Deviance information criterion (DIC) is a Bayesian variation of the AIC that estimates $elppd_{\hat{\theta}}$ at the posterior mean. Let $\hat{\theta}$ be the posterior mean of θ, we can approxiamte $elppd$ by

$$\widehat{elppd}^{DIC} = \log p(\boldsymbol{y}|\hat{\theta}) - p_{DIC}$$

  where p_{DIC} is called the effective number of parameters

$$p_{DIC} = 2(\log p(\boldsymbol{y}|\hat{\theta}) - \mathbb{E}_{\theta|\boldsymbol{y}}(\log p(\boldsymbol{y}|\theta)))$$

  which can be computed by

$$2(\log p(\boldsymbol{y}|\hat{\theta}) - \frac{1}{S}\sum_{s=1}^{S}(\log p(\boldsymbol{y}|\theta^s)))$$

- p_{DIC} is defined in this way based on the asymptotic normal posterior distribution and a few other approximation steps.
- The DIC is defined again by multiplying -2 to the $elppd$ estimate

$$DIC = -2\log p(\boldsymbol{y}|\hat{\theta}) + 2p_{DIC}$$

WAIC

- Watanabe-Akaike information criterion, or widely applicable information criterion, starts with $lppd$ instead of a point estimate.
- The adjustment is defined by

$$p_{WAIC} = 2\sum_i (\log(\mathbb{E}_{\theta|\boldsymbol{y}} p(y_i|\theta)) - \mathbb{E}_{\theta|\boldsymbol{y}} \log p(y_i|\theta))$$

- Intuitively, it measures the gap between the expected predictive density and the average predictive density over posterior draws. The difference is positive (Jensen inequality). If the posterior draws of the log likelihood is highly variable, the difference is larger
- An alternative formula is

$$p_{WAIC} = \sum_{i=1}^{n} \text{var}_{\theta|\boldsymbol{y}}(\log p(y_i|\theta))$$

- The WAIC is defined as

$$WAIC = -2\sum_i (\log(\mathbb{E}_{\theta|\boldsymbol{y}} p(y_i|\theta)) + 2p_{WAIC}$$

WAIC: equivalence

- Let $\ell(\theta) = \ln p(y_i|\theta)$. Perform a second-order Taylor expansion of the likelihood gives

$$e^{\ell} \approx e^{\bar{\ell}} + e^{\bar{\ell}}(\ell - \bar{\ell}) + \frac{1}{2}e^{\bar{\ell}}(\ell - \bar{\ell})^2$$

- Taking the posterior expectation on both sides:

$$E[e^{\ell}] \approx e^{\bar{\ell}} + e^{\bar{\ell}} \underbrace{E[\ell - \bar{\ell}]}_{0} + \frac{1}{2}e^{\bar{\ell}} \underbrace{E[(\ell - \bar{\ell})^2]}_{\text{var}(\ell)}$$

$$E[e^{\ell}] \approx e^{\bar{\ell}} \left(1 + \frac{1}{2}\text{var}(\ell)\right)$$

- Approximate with $\ln(1 + x) \approx x$ for small x,

$$\ln E[e^{\ell}] \approx \bar{\ell} + \ln \left(1 + \frac{1}{2}\text{var}(\ell)\right) \approx E[\ln p] + \frac{1}{2}\text{var}(\ln p)$$

Example: DIC and WAIC

```r
# DIC and WAIC, loglik is Nsim by N matrix
get_DIC <- function(x, y, beta, sigma2, loglik){
    dic <- sum(dnorm(y, x %*% beta, sigma2, log = TRUE))
    pd <- 2 * (dic - mean(apply(loglik, 1, sum)))
    dic <- -2 * dic + 2 * pd
    return(dic)
}

get_WAIC_1 <- function(loglik){
    waic <- sum(log(apply(exp(loglik), 2, mean)))
    pd <- 2 * sum(log(apply(exp(loglik), 2, mean)) -
                  apply(loglik, 2, mean))
    waic <- -2 * waic + 2 * pd
    return(waic)
}
get_WAIC_2 <- function(loglik){
    waic <- -2 * sum(log(apply(exp(loglik), 2, mean)))
    pd <- sum(apply(loglik, 2, var))
    waic <- waic + 2 * pd
    return(waic)
}
```

Example: DIC and WAIC

```
getIC <- function(x, y, beta, sigma2, loglik){
    c(get_DIC(x, y, beta, sigma2, loglik),
        get_WAIC_1(loglik),
        get_WAIC_2(loglik))
}

half <- (Nsim/2) : Nsim
metrics <- rbind(
    getIC(x1, y, apply(out1[half, ], 2, mean)[1:2],
                    mean(out1[(Nsim/2):Nsim, 3]^2),
                    loglik1[half, ]),
    getIC(x2, y, apply(out2[half, ], 2, mean)[1:3],
                    mean(out2[(Nsim/2):Nsim, 4]^2),
                    loglik2[half, ]),
    getIC(x3, y, apply(out3[half, ], 2, mean)[1:4],
                    mean(out3[(Nsim/2):Nsim, 5]^2),
                    loglik3[half, ]),
    getIC(x4, y, apply(out4[half, ], 2, mean)[2:6],
                    mean(out4[(Nsim/2):Nsim, 7]^2),
                    loglik4[half, ]))

metrics <- data.frame(metrics)
colnames(metrics) <- c("DIC", "WAIC 1", "WAIC 2")
rownames(metrics) <- c("Model 1", "Model 2", "Model 3", "Model 4")
```

Leave-one-out cross validation

- The various information criterion aims to approximate the out-of-sample prediction error using various asymptotic approximations.
- A more direct approach to evaluate prediction error is via cross validation. That is, we fit the model on a subset of data, predict the hold-out observations and evaluate the predictive density on the hold-out observations.
- leave-one-out (LOO) cross validation is a special case of the procedure, consider

$$lppd_{loo} = \sum_i \log p(y_i|\boldsymbol{y}_{-i}) = \sum_i \int p(y_i|\theta)p(\theta|\boldsymbol{y}_{-i})d\theta$$

- It turns out in order to compute $lppd_{loo}$, we do not need to refit the model n times. Instead, $p_{y_i|\boldsymbol{y}_{-i}}$ can be approximated using $p(y_i|\boldsymbol{y})$ directly.

Leave-one-out cross validation: importance sampling revisited

- Importance sampling: To approximate $\mathbb{E}(g(\theta)|y) = \int g(\theta)p(\theta|y)d\theta$, we can use a density $h(\theta)$ that is easy to sample from. Then

$$\mathbb{E}(g(\theta)|y) = \int \frac{g(\theta)}{h(\theta)} p(\theta|y)h(\theta)d\theta \approx \frac{1}{M}\sum_{m=1}^{M} g(\theta^{(m)})w(\theta^{(m)})$$

  where $\theta^{(m)} \sim H(\theta)$ and $w(\theta) = \frac{p(\theta|y)}{h(\theta)}$ is called the importance weight. Since we usually know $p(\theta|y)$ up to a proportional constant, then

$$\mathbb{E}(g(\theta)|y) \approx \frac{\sum_{m=1}^{M} g(\theta^{(m)})w(\theta^{(m)})}{\sum_{m=1}^{M} w(\theta^{(m)})}$$

- We want to compute $\int p(y_i|\theta)p(\theta|\boldsymbol{y}_{-i})d\theta$, and we can use the importance weights

$$w_i(\theta) = \frac{p(\theta|\boldsymbol{y}_{-i})}{p(\theta|\boldsymbol{y})} \propto \frac{p(\boldsymbol{y}_{-i}|\theta)p(\theta)}{p(\boldsymbol{y}|\theta)p(\theta)} = \frac{1}{p(y_i|\theta)}$$

  under the model where $p(\boldsymbol{y}|\theta) = \prod_i p(y_i|\theta)$.

- WAIC is asymptotically equivalent to LOO

Leave-one-out cross validation: importance sampling revisited

- Put it together, with every draw θ^s from the posterior $p(\theta|\boldsymbol{y})$, the importance weights are

$$w_i(\theta^s) = \propto \frac{1}{p(y_i|\theta^s)}$$

and the importance sampling LOO predictive distribution (IS-LOO) is

$$p(y_i|y_{-i}) \approx \frac{\sum_s w_i(\theta^s)p(y_i|\theta^s)}{\sum_s w_i(\theta^s)} = \frac{1}{\frac{1}{S}\sum_s \frac{1}{p(y_i|\theta^s)}}$$

which can then be added up to compute $lppd_{loo}$. Multiplying it by -2 leads to the same deviance scale as DIC and WAIC.

- The importance sampling procedure can be improved with an additional smoothing step of the weights (Pareto smoothed importance sampling (PSIS)).

- $lppd_{loo}$ also have other names in the literature, e.g., conditional predictive ordinates (CPO).

- WAIC is asymptotically equivalent to $-2lppd_{loo}$.

Example: DIC and WAIC and LOO

```
library(loo)
metrics$loo <- c(
    loo(loglik1[half, ])$estimates["looic", "Estimate"],
    loo(loglik2[half, ])$estimates["looic", "Estimate"],
    loo(loglik3[half, ])$estimates["looic", "Estimate"],
    loo(loglik4[half, ])$estimates["looic", "Estimate"])
kable(metrics) %>% kable_styling()
```

	DIC	WAIC 1	WAIC 2	loo
Model 1	1258	1896	1897	1897
Model 2	1257	1878	1879	1879
Model 3	1229	1588	1588	1588
Model 4	1212	1495	1496	1496

On model checking and comparison

- The model checking and comparison tools are more generally useful for other non-linear models as well.
- What we covered here are only some of the ‘simplest‘ and more widely useful tools.
- I highly recommend the Bayesian workflow paper for a much fuller picture on what to consider when developing/applying Bayesian models:
- *Gelman, Andrew, Aki Vehtari, Daniel Simpson, Charles C. Margossian, Bob Carpenter, Yuling Yao, Lauren Kennedy, Jonah Gabry, Paul-Christian Bürkner, and Martin Modrák. "Bayesian workflow." arXiv preprint arXiv:2011.01808 (2020).*