STA 610L: Module 3.3

Bayesian linear mixed effects models

Dr. Olanrewaju Michael Akande

Linear mixed effects model

Recall the standard representation of the linear mixed effects model is

$$\begin{aligned} &Y_i = X_i\beta + Z_i b_i + \varepsilon_i; \quad i = 1, \ldots, m \\ &b_i \perp \varepsilon_i \quad\quad b_i \sim N_q(0, D) \quad\quad \varepsilon_i \sim N_{n_i}(0, R_i), \end{aligned}$$

where

- Y_i is a $n_i \times 1$ vector of outcomes for subject i
- X_i is a $n_i \times p$ design matrix of predictor variables corresponding to each outcome measurement occasion for subject i
- Z_i is a $n_i \times q$ design matrix corresponding to the random effects for subject i
- β is a $p \times 1$ vector of regression coefficients (fixed effects)
- b_i is a $q \times 1$ vector of random effects for subject i
- ε_i is a $n_i \times 1$ vector of errors for subject i

Bayesian inference for the linear mixed effects model

Given our discussions on how complicated specifying D and R_i can be, it will be very convenient to start with a simplified version of this model as we try to understand our options.

Specifically, we will start by assuming that X_i and Z_i are the same, and also that $R_i = \sigma^2 I_{n_i}$, so that we can focus on D.

Thus, we write

$$Y_{ij} = X_{ij}\beta_i + \varepsilon_{ij}, \qquad \beta_i = \theta + b_i,$$

where $\varepsilon_{ij} \overset{iid}{\sim} N(0, \sigma^2)$, $b_i \overset{iid}{\sim} N(0, D)$.

We can then write $\beta_i \mid \theta \sim N(\theta, D)$.

Here, $i = 1, \ldots, m$ index groups, with group i having n_i observations, so that the parameters θ are fixed effects and the parameters b_i are random effects.

Priors

We already know that a conditionally-conjugate prior specification for two of the parameters is given by

$$\theta \sim N(\mu_0, \Lambda_0),$$

$$\sigma^2 \sim \mathcal{IG}\left(\frac{\nu_0}{2}, \frac{\nu_0\sigma_0^2}{2}\right).$$

How about D?

One complication of course is that the D must be **positive definite and symmetric**.

Review: positive definite and symmetric

"Positive definite" means that for all $x \in \mathcal{R}^p$, $x^T D x > 0$.

Basically ensures that the diagonal elements of D (corresponding to the marginal variances) are positive.

Also, ensures that the correlation coefficients for each pair of variables are between -1 and 1.

Our prior for D should thus assign probability one to set of positive definite matrices.

Analogous to the univariate case, the inverse-Wishart distribution is the corresponding conditionally conjugate prior for D (multivariate generalization of the inverse-gamma).

The STA 360/601/602 Hoff textbook covers the construction of Wishart and inverse-Wishart random variables. We will skip the actual development.

Review: Inverse-Wishart distribution

A random variable $\Sigma \sim \text{IW}_p(\eta_0, \boldsymbol{S}_0)$, where Σ is positive definite and $p \times p$, has pdf

$$p(\Sigma) \;\propto\; |\Sigma|^{\frac{-(\eta_0+p+1)}{2}} \text{exp} \left\{ -\frac{1}{2} \text{tr}(\boldsymbol{S}_0 \Sigma^{-1}) \right\},$$

where

- $\eta_0 > p - 1$ is the "degrees of freedom", and
- $\boldsymbol{S}_0$ is a $p \times p$ positive definite matrix.

For this distribution, $\mathbb{E}[\Sigma] = \dfrac{1}{\eta_0 - p - 1} \boldsymbol{S}_0$, for $\eta_0 > p + 1$.

Hence, $\boldsymbol{S}_0$ is the scaled mean of the $\text{IW}_p(\eta_0, \boldsymbol{S}_0)$.

Review: Inverse-Wishart distribution

If we are very confident in a prior guess Σ_0, for Σ, then we might set

- η_0, the degrees of freedom to be very large, and
- $\boldsymbol{S}_0 = (\eta_0 - p - 1)\Sigma_0$.

In this case, $\mathbb{E}[\Sigma] = \dfrac{1}{\eta_0 - p - 1}\boldsymbol{S}_0 = \dfrac{1}{\eta_0 - p - 1}(\eta_0 - p - 1)\Sigma_0 = \Sigma_0$, and Σ is tightly (depending on the value of η_0) centered around Σ_0.

If we are not at all confident but we still have a prior guess Σ_0, we might set

- $\eta_0 = p + 2$, so that the $\mathbb{E}[\Sigma] = \dfrac{1}{\eta_0 - p - 1}\boldsymbol{S}_0$ is finite.
- $\boldsymbol{S}_0 = \Sigma_0$

Here, $\mathbb{E}[\Sigma] = \Sigma_0$ as before, but Σ is only loosely centered around Σ_0.

Review: Wishart distribution

Just as we had with the gamma and inverse-gamma relationship in the univariate case, we can also work in terms of the Wishart distribution (multivariate generalization of the gamma) instead.

The Wishart distribution provides a conditionally-conjugate prior for the precision matrix Σ^{-1} in a multivariate normal model.

Specifically, if $\Sigma \sim \mathrm{IW}_p(\eta_0, \boldsymbol{S}_0)$, then $\Phi = \Sigma^{-1} \sim \mathrm{W}_p(\eta_0, \boldsymbol{S}_0^{-1})$.

A random variable $\Phi \sim \mathrm{W}_p(\eta_0, \boldsymbol{S}_0^{-1})$, where Φ has dimension $(p \times p)$, has pdf

$$f(\Phi) \ \propto \ |\Phi|^{\frac{\eta_0 - p - 1}{2}} \exp\left\{-\frac{1}{2}\mathrm{tr}(\boldsymbol{S}_0 \Phi)\right\}.$$

Here, $\mathbb{E}[\Phi] = \eta_0 \boldsymbol{S}_0$.

Note that the STA 360/601/602 Hoff textbook writes the inverse-Wishart as $\mathrm{IW}_p(\eta_0, \boldsymbol{S}_0^{-1})$. I prefer $\mathrm{IW}_p(\eta_0, \boldsymbol{S}_0)$ instead. Feel free to use either notation but try not to get confused.

Back to the priors

For the full prior specification, we can then write

$$\theta \sim N(\mu_0, \Lambda_0),$$

$$D \sim \text{inverse-Wishart}(\eta_0, S_0),$$

and

$$\sigma^2 \sim \text{inverse-gamma}\left(\frac{\nu_0}{2}, \frac{\nu_0\sigma_0^2}{2}\right),$$

so that a simple Gibbs sampler can be used for posterior computation.

We will mostly rely on the brms package for simple specifications but it is relatively easy to write your own Gibbs sampler here.

Full conditionals

$$\beta_i \mid y_i, X_i, \theta, D, \sigma^2 \sim N(\mu_{\beta_i}, \Sigma_{\beta_i}),$$

where

$$\Sigma_{\beta_i} = \left(D^{-1} + \frac{1}{\sigma^2} X_i' X_i \right)^{-1},$$

and

$$\mu_{\beta_i} = \Sigma_{\beta_i} \left(D^{-1}\theta + \frac{1}{\sigma^2} X_i' y_i \right).$$

Full conditionals

$$\theta \mid \beta_1, \ldots, \beta_m, D \sim N(\mu_\theta, \Lambda_\theta),$$

where

$$\Lambda_\theta = \left(\Lambda_0^{-1} + mD^{-1}\right)^{-1},$$

$$\mu_\theta = \Lambda_\theta \left(\Lambda_0^{-1}\mu_0 + mD^{-1}\bar{\beta}\right),$$

and $\bar{\beta}$ is the vector average $\frac{1}{m}\sum \beta_i$.

Full conditionals

$$D \mid \theta, \beta_1, \ldots, \beta_m \sim \mathrm{IW}\left(\eta_D, S_D\right),$$

where

$$\eta_D = \eta_0 + m; \quad S_D = S_0 + S_\theta,$$

with

$$S_\theta = \sum_{i=1}^{m} (\beta_i - \theta)(\beta_i - \theta)'$$

Full conditionals

$$\sigma^2 \mid \beta_1, \ldots, \beta_m \sim \mathcal{IG}\left(\frac{\nu_n}{2}, \frac{\nu_n \sigma_n^2}{2}\right),$$

where

$$\nu_n = \nu_0 + \sum n_i; \quad \sigma_n^2 = \frac{1}{\nu_n}\left[\nu_0 \sigma_0^2 + SSR\right],$$

with

$$SSR = \sum_{i=1}^{m} \sum_{j=1}^{n_i} (y_{ij} - x_{ij}\beta_i)^2.$$

Motivation for other covariance priors

While the inverse Wishart is a nice prior for symmetric matrices, computation can be a challenge, expecially as the covariance matrix becomes large.

Why is modeling a covariance matrix difficult?

- number of parameters may be quite large
- matrix constrained to be nonnegative definite

Motivation for other covariance priors

Another down side of the Wishart is that we must use the same df for all elements, though in practice, we may have more information about some components than others.

For example, we may believe in advance that the regression coefficients for one predictor are fairly similar across groups, while we may have little knowledge about similarity of coefficients for another predictor.

It is essentially impossible to express these prior beliefs using the inverse Wishart.

Other covariance priors

A popular alternative approach is to decompose the covariance matrix Σ into a correlation matrix and a diagonal matrix of standard deviations:

$$\Sigma = \begin{pmatrix} \tau_1 & 0 & \cdots & 0 \\ 0 & \tau_2 & \cdots & 0 \\ 0 & \vdots & \cdots & \vdots \\ 0 & \cdots & \cdots & \tau_K \end{pmatrix} \Omega \begin{pmatrix} \tau_1 & 0 & \cdots & 0 \\ 0 & \tau_2 & \cdots & 0 \\ 0 & \vdots & \cdots & \vdots \\ 0 & \cdots & \cdots & \tau_K \end{pmatrix},$$

where $\tau_k = \sqrt{\Sigma_{k,k}}$ and $\Omega_{i,j} = \frac{\Sigma_{i,j}}{\tau_i \tau_j}$.

Other covariance priors

This separation strategy yields nice interpretations for components, as researchers are often more used to thinking of the standard deviations and correlations than of covariances.

Typically, the priors on τ_k are assumed to be independent of the prior on Ω, though this could be incorporated through a prior on $\Omega \mid \tau$.

Other covariance priors

In this parameterization, any reasonable prior for scale parameters can be given to the components of the scale vector τ.

Popular choices include half-Cauchy or half-normal distributions, but log normal or inverse gamma priors might also be used.

This approach is particularly attractive relative to the inverse Wishart, which requires us to use the same df for all elements, though in practice, we may wish to have more flexibility in dealing with tails of individual variance components.

LKJ PRIOR

A nice choice for the correlation matrix is the LKJ (Lewandowski-Kurowicka-Joe) prior, which is like an extension of the beta distribution.

The LKJ distribution is commonly used for positive definite correlation matrices, or equivalently for their Cholesky factors.

This prior is

$$\mathrm{LkjCorr}(\Omega \mid \eta) \propto |\Omega|^{\eta-1},$$

which for $\eta = 1$ is the joint uniform distribution (note the marginals here are not uniform but favor more mass around 0).

For $\eta > 1$, the density concentrates increasing mass around the identiy (favoring lower correlation), and for $\eta < 1$, mass is increasingly spread towards more extreme values.

For more information on the LKJ prior, see here.

Short activity!

Plot the LKJ density for a given correlation (unnormalized is ok) for a variety of values of the shape parameter η (positive scalar).

You may find this link quite useful along with instructions for installing the rethinking package.

Example: coffee robot

We use an example from McElreath's book *Statistical Rethinking* about a coffee robot.

While these are simulated data, they provide an interesting application as well as great code should you need to simulate hierarchical data in the future!

Suppose we have a coffee-making robot that moves among cafes to order coffee and record the wait time.

The robot also records the time of day of the visit because the average wait time in the morning tends to be longer than in the afternoon due to the fact that the cafes are busier in the mornings.

The robot learns more efficiently about wait times when it pools information across different cafes.

Example: coffee robot

- We can use varying intercepts to pool information across coffee shops.
- Coffee shops vary in average wait times due to a number of factors (e.g., barista skill, number of baristas).
- Coffee shops also vary in differences between morning and afternoon.
- Varying intercepts for cafes and "slopes" for the afternoon effect make for a reasonable model.
- In this example we focus on the cafe as a grouping factor.

Example: coffee robot

Model:

$$y_{ij} = \beta_{0,i} + \beta_{1,i} A_{ij} + \varepsilon_{ij}$$

$$\beta_{0,i} = \beta_0 + b_{0,i} \quad \beta_{1,i} = \beta_1 + b_{1,i}$$

$$\varepsilon_i \overset{iid}{\sim} N(0, \sigma^2 I) \quad \perp \quad b_i \overset{iid}{\sim} N(0, D), \quad D = \begin{pmatrix} \tau_0 & 0 \\ 0 & \tau_1 \end{pmatrix} \Upsilon \begin{pmatrix} \tau_0 & 0 \\ 0 & \tau_1 \end{pmatrix}$$

Example: coffee robot

Priors:

- $\beta_0 \sim N(0, 10) \qquad \beta_1 \sim N(0, 10)$
- $\sigma \sim \text{Half Cauchy}(0, 1)$
- $\tau_0 \sim \text{Half Cauchy}(0, 1) \quad \tau_1 \sim \text{Half Cauchy}(0, 1)$
- $\Upsilon = \begin{pmatrix} 1 & \upsilon \\ \upsilon & 1 \end{pmatrix} \sim LKJcorr(2)$

Data

```
#library(tidyverse)
#library(brms)
#example from McElreath with thanks to Solomon Kurz for the BRMS translation
a       <-  3.5  # average morning wait time
b       <- -1    # average difference afternoon wait time
sigma_a <-  1    # std dev in intercepts
sigma_b <-  0.5  # std dev in slopes
rho     <- -0.7   # correlation between intercepts and slopes

# combine the terms above
mu     <- c(a, b)
sigmas <- c(sigma_a, sigma_b)          # standard deviations
rho    <- matrix(c(1, rho,             # correlation matrix
                   rho, 1), nrow = 2)

# now matrix multiply to get covariance matrix
sigma <- diag(sigmas) %*% rho %*% diag(sigmas)
```

Data

```r
# how many cafes would you like?
n_cafes <- 20

set.seed(13)  # used to replicate example
vary_effects <-
  MASS::mvrnorm(n_cafes, mu, sigma) %>%
  data.frame() %>%
  set_names("a_cafe", "b_cafe")

head(vary_effects)
```

```
##     a_cafe     b_cafe
## 1 2.917639 -0.8649154
## 2 3.552770 -1.6814372
## 3 1.694390 -0.4168858
## 4 3.442417 -0.6011724
## 5 2.289988 -0.7461953
## 6 3.069283 -0.8839639
```

Data

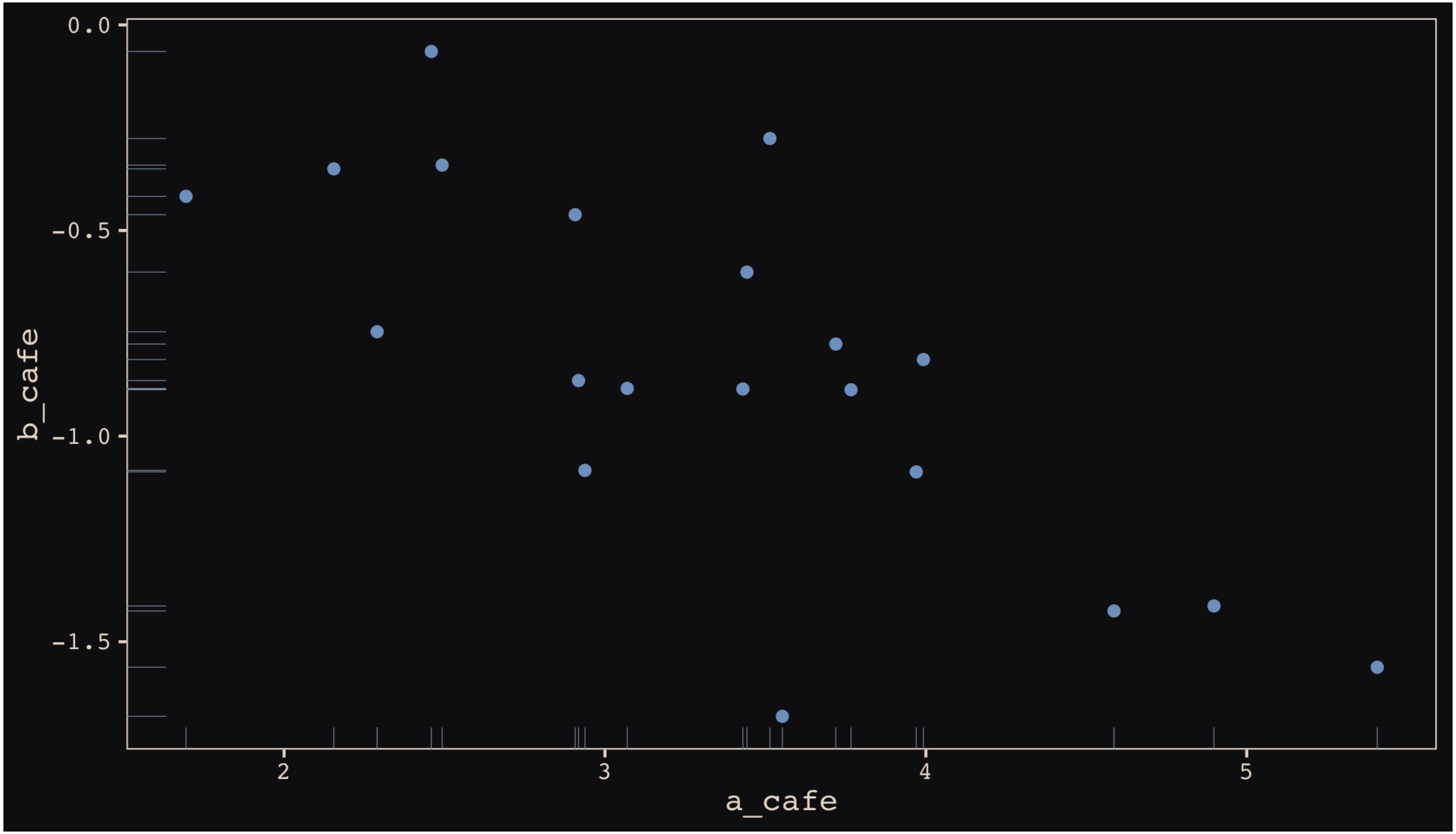
OK, so now we've simulated the cafe-specific intercepts and slopes!

This next block of code adds a pretty set of colors.

```r
#plot of cafe-specific intercepts and slopes
vary_effects %>%
  ggplot(aes(x = a_cafe, y = b_cafe)) +
  geom_point(color = "#80A0C7") +
  geom_rug(color = "#8B9DAF", size = 1/7) +
  theme_pearl_earring
```

Data

Here we see a negative correlation in our intercepts and slopes. Remember these are the "true" parameters rather than our data.



Data

```r
n_visits <- 10
sigma    <-  0.5  # std dev within cafes

set.seed(13)  # used to replicate example
d <-
  vary_effects %>%
  mutate(cafe      = 1:n_cafes) %>%
  expand(nesting(cafe, a_cafe, b_cafe), visit = 1:n_visits) %>%
  mutate(afternoon = rep(0:1, times = n() / 2)) %>%
  mutate(mu        = a_cafe + b_cafe * afternoon) %>%
  mutate(wait      = rnorm(n = n(), mean = mu, sd = sigma))
d %>%
  head()
```

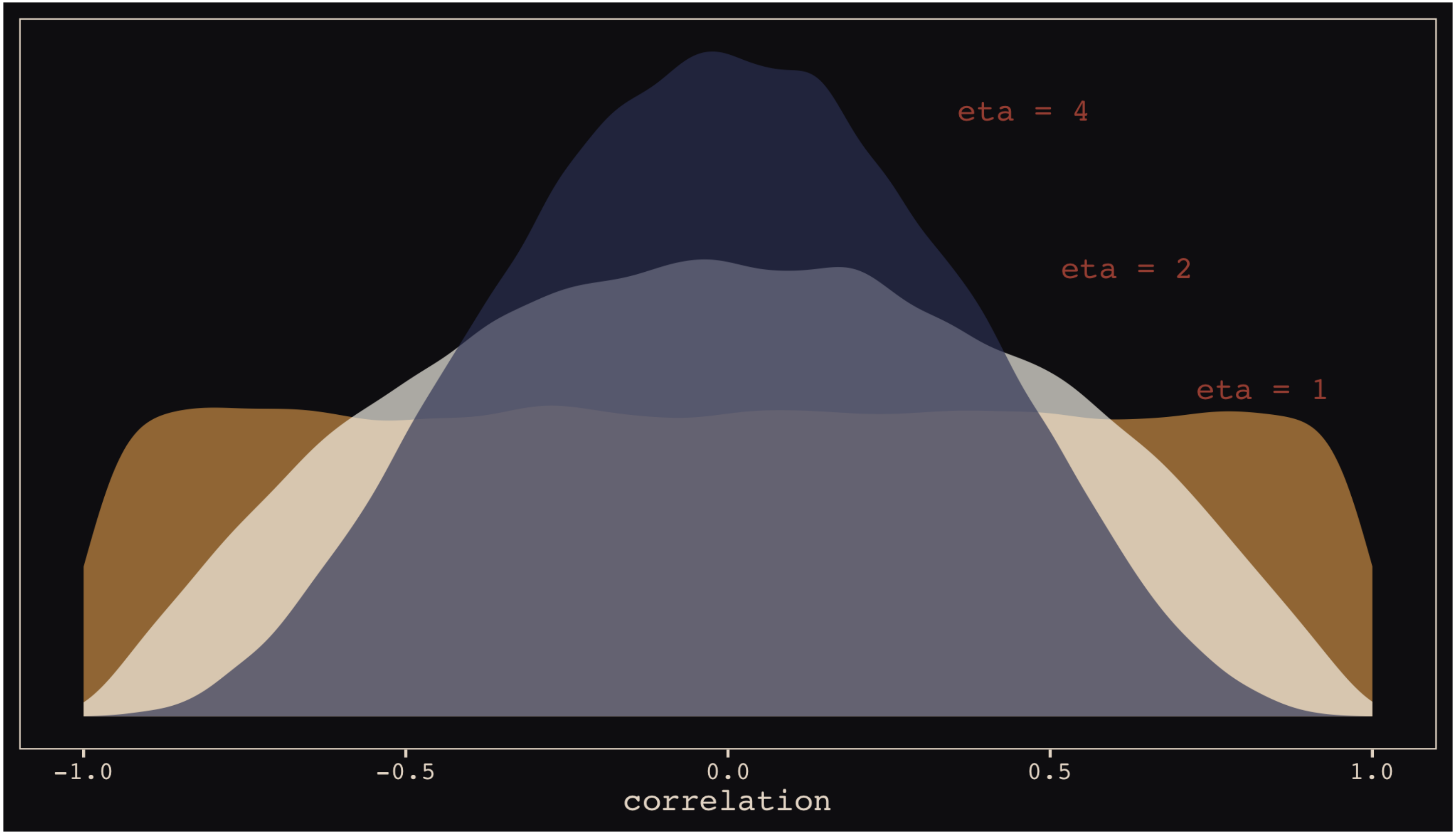
```
## # A tibble: 6 x 7
##    cafe a_cafe b_cafe visit afternoon    mu  wait
##   <int>  <dbl>  <dbl> <int>     <int> <dbl> <dbl>
## 1     1   2.92 -0.865     1         0  2.92  3.19
## 2     1   2.92 -0.865     2         1  2.05  1.91
## 3     1   2.92 -0.865     3         0  2.92  3.81
## 4     1   2.92 -0.865     4         1  2.05  2.15
## 5     1   2.92 -0.865     5         0  2.92  3.49
## 6     1   2.92 -0.865     6         1  2.05  2.26
```

Prior

First, let's look at that prior for Υ.

```r
#library(rethinking)
n_sim <- 1e5

set.seed(13)
r_1 <-
  rlkjcorr(n_sim, K = 2, eta = 1) %>%
  as_tibble()

r_2 <-
  rlkjcorr(n_sim, K = 2, eta = 2) %>%
  as_tibble()

r_4 <-
  rlkjcorr(n_sim, K = 2, eta = 4) %>%
  as_tibble()

ggplot(data = r_1, aes(x = V2)) +
  geom_density(color = "transparent", fill = "#DCA258", alpha = 2/3) +
  geom_density(data = r_2,
               color = "transparent", fill = "#FCF9F0", alpha = 2/3) +
  geom_density(data = r_4,
               color = "transparent", fill = "#394165", alpha = 2/3) +
  geom_text(data = tibble(x     = c(.83, .62, .46),
                          y     = c(.54, .74, 1),
                          label = c("eta = 1", "eta = 2", "eta = 4")),
            aes(x = x, y = y, label = label),
            color = "#A65141", family = "Courier") +
  scale_y_continuous(NULL, breaks = NULL) +
  xlab("correlation") +
  theme_pearl_earring
```

Prior



Model

Now we switch to the brms package and fit the model.

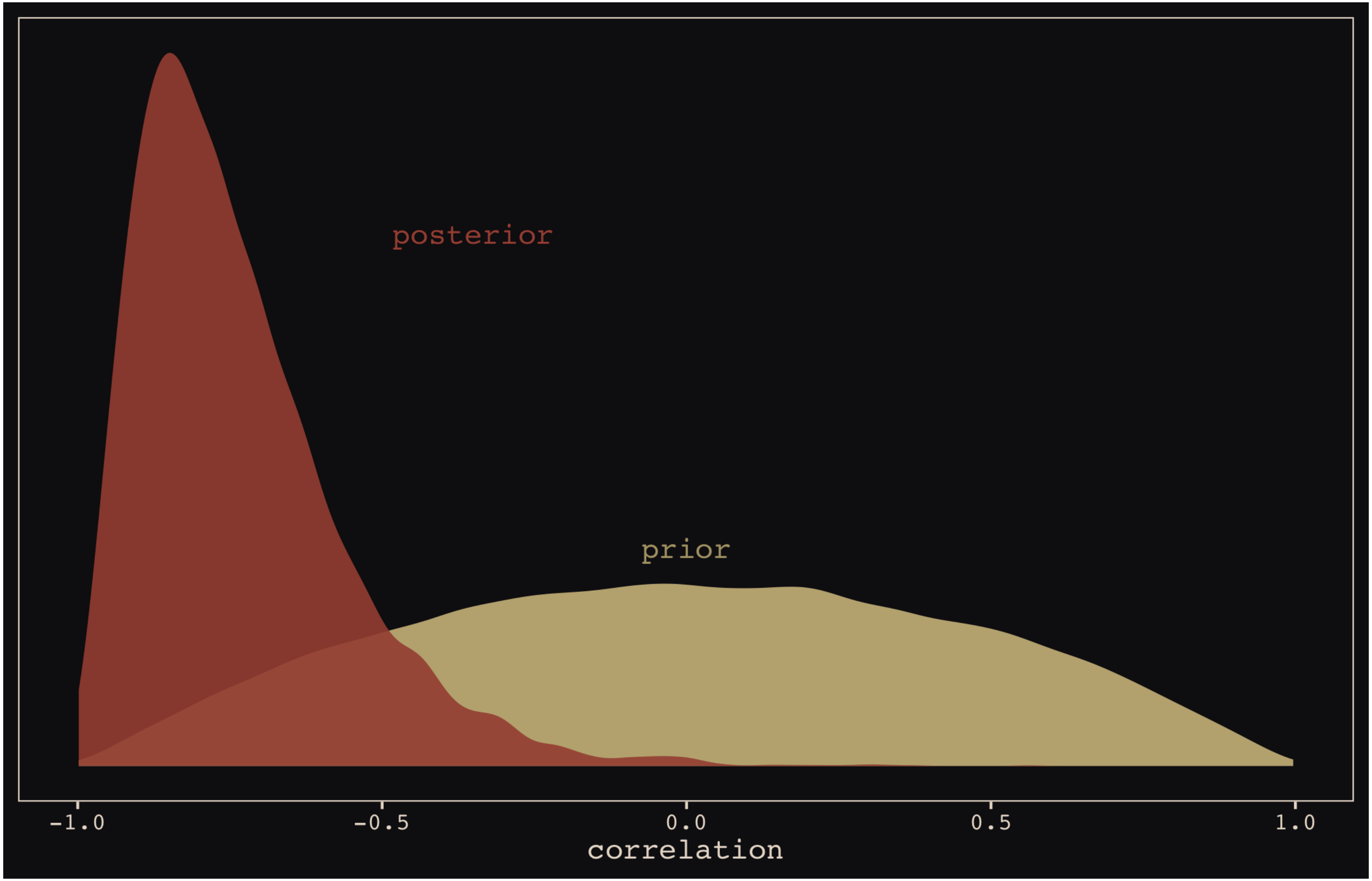
```r
detach(package:rethinking, unload = T)
#library(brms)

 b13.1 <-
  brm(data = d, family = gaussian,
      wait ~ 1 + afternoon + (1 + afternoon | cafe),
      prior = c(prior(normal(0, 10), class = Intercept),
                prior(normal(0, 10), class = b),
                prior(cauchy(0, 1), class = sd),
                prior(cauchy(0, 1), class = sigma),
                prior(lkj(2), class = cor)),
      iter = 5000, warmup = 2000, chains = 2, cores = 2,
      seed = 13)
```

Posterior summaries

Let's compare posterior correlation of random effects to the prior.

```
post <- posterior_samples(b13.1)

post %>%
  ggplot(aes(x = cor_cafe__Intercept__afternoon)) +
  geom_density(data = r_2, aes(x = V2),
               color = "transparent", fill = "#EEDA9D", alpha = 3/4) +
  geom_density(color = "transparent", fill = "#A65141", alpha = 9/10) +
  annotate("text", label = "posterior",
           x = -0.35, y = 2.2,
           color = "#A65141", family = "Courier") +
  annotate("text", label = "prior",
           x = 0, y = 0.9,
           color = "#EEDA9D", alpha = 2/3, family = "Courier") +
  scale_y_continuous(NULL, breaks = NULL) +
  xlab("correlation") +
  theme_pearl_earring
```

Posterior summaries



Posterior summaries

It takes a lot of code to generate the following figures, which illustrate shrinkage in this model.

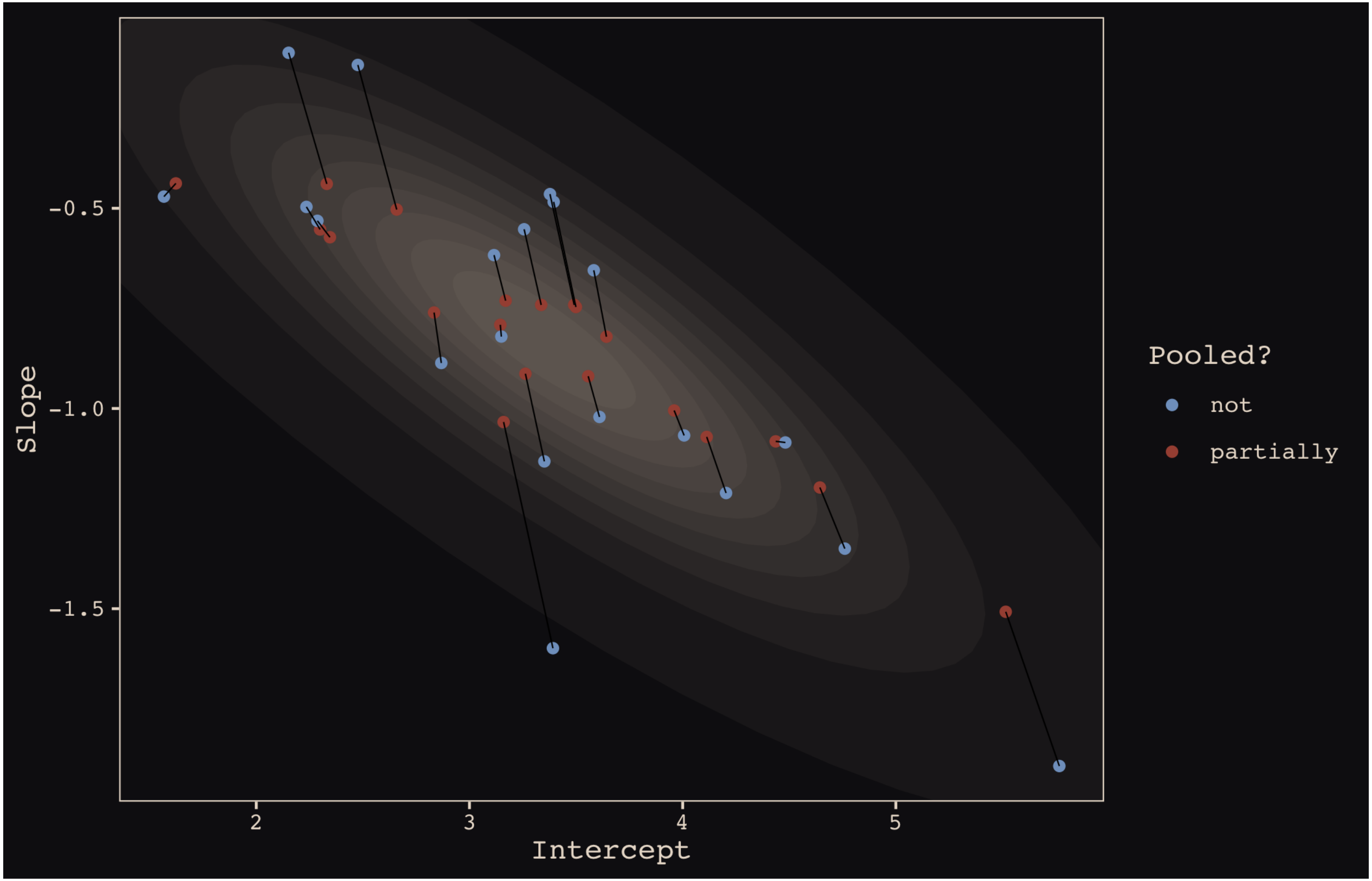
If you're interested, let me know and I can make it available to you, or the McElreath book, or Solomon's website.

These figures examine random intercepts vs random slopes as well as the morning and afternoon wait times on the original scale (minutes).

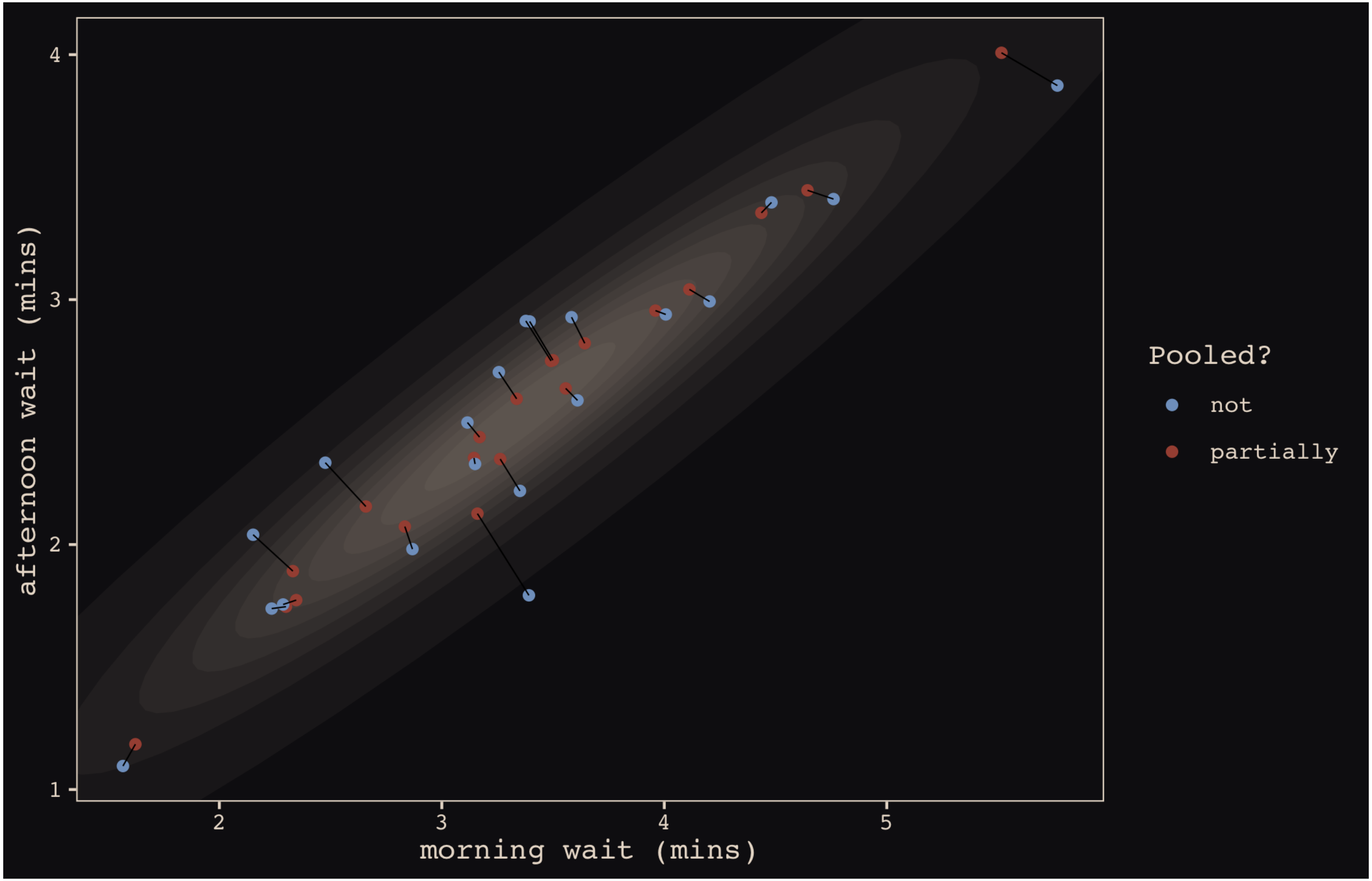
- Blue dot: unpooled estimate
- Red dot: pooled estimate

Note shrinkage is toward the center of the ellipse.

Posterior summaries



Posterior summaries



What's next?

Move on to the readings for the next module!

STA 610L