Hierarchical Models with the rstanarm and brms Packages

Ben Goodrich

May 02, 2022

Review Session for Final Exam

- Classes end tonight
- Review session Thursday, May 5th, from 10AM to 12PM in Hamilton 413 (across Amsterdam from the hospital)
- Final exam is Monday, May 9th, from 4:10PM to 7PM in IAB 403

What Are Hierarchical Models

- In Bayesian terms, a hierarchical model is nothing more than a model where the prior distribution of some parameter depends on another parameter
- We have already seen several examples:
  - Bowling: x_2 depends on $n = 10 - x_1$ and both depend on inability, θ
  - Linear models: $\sigma_Y \sim ?$ and $\forall n : \epsilon_n \sim \mathcal{N}(0, \sigma_Y)$
  - Splines: $\sigma_\beta \sim ?$ and $\forall k : \beta_k \sim \mathcal{N}(0, \sigma_\beta)$
- In other words, it is just another application of the rules of probability:

$$f(\boldsymbol{\theta}) = \int f(\boldsymbol{\theta}, \boldsymbol{\phi})\, d\phi_1 \dots d\phi_K = \int f(\boldsymbol{\theta} \mid \boldsymbol{\phi})\, f(\boldsymbol{\phi})\, d\phi_1 \dots d\phi_K$$

Cluster Sampling vs. Stratified Sampling

- For cluster random sampling, you
  - Sample J large units (such as schools) from their population
  - Sample N_j small units (such as students) from the j-th large unit
- If you replicate such a study, you get different realizations of the large units
- For stratified random sampling, you
  - Divide the population of large units into J mutually exclusive and exhaustive groups (like states), which are not random variables
  - Sample N_j small units (such as voters) from the j-th large unit
- If you replicate such a study, you would use the same large units and only get different realizations of the small units
- The difference between cluster and stratified random sampling is critical if you care about the distribution of an estimator across randomly-sampled datasets

Models with Group-Specific Intercepts

- Let α be the common intercept and $\boldsymbol{\beta}$ be the common coefficients while a_j is the deviation from the common intercept in the j-th group. Write a model as:

$$y_{ij} = \overbrace{\underbrace{\alpha + \sum_{k=1}^{K} \beta_k x_{ik}}_{\text{Frequentist } \boldsymbol{\mu}|\mathbf{x}} + a_j}^{\text{Bayesian } \boldsymbol{\mu}|\mathbf{x},j} + \epsilon_{ij} = \alpha + \sum_{k=1}^{K} \beta_k x_{ik} + \underbrace{a_j + \overbrace{\epsilon_{ij}}^{\text{Bayesian error}}}_{\text{Frequentist error}}$$

- The same holds in GLMs where $\eta_{ij} = \alpha + \sum_{k=1}^{K} \beta_k x_{ik} + a_j$ or $\eta_{ij} = \alpha + \sum_{k=1}^{K} \beta_k x_{ik}$ depending on if you are Bayesian or Frequentist

Models with Group-Specific Slopes and Intercepts

- Let α be the common intercept and $\boldsymbol{\beta}$ be the common coefficients while a_j is the deviation from the common intercept in the j-th group and $\mathbf{b}_j$ is the deviation from the common coefficients. Write the model as:

$$y_{ij} = \overbrace{\underbrace{\alpha + \sum_{k=1}^{K} \beta_k x_{ik}}_{\text{Frequentist } \boldsymbol{\mu}|\mathbf{x}} + a_j + \sum_{k=1}^{K} b_{jk} x_{ik}}^{\text{Bayesian } \boldsymbol{\mu}|\mathbf{x},j} + \epsilon_{ij} =$$

$$\alpha + \sum_{k=1}^{K} \beta_k x_{ik} + \underbrace{a_j + \sum_{k=1}^{K} b_{jk} x_{ik} + \overbrace{\epsilon_{ij}}^{\text{Bayesian error}}}_{\text{Frequentist error}}$$

- And similarly for GLMs

Frequentist Estimation of Multilevel Models

- Frequentists assume that a_j and b_j deviate from the common parameters according to a (multivariate) normal distribution, whose (co)variances are common parameters to be estimated
- To Frequentists, a_j and b_j are not parameters because parameters must remained fixed in repeated sampling of observations from some population
- Since a_j and b_j are not parameters, they can't be "estimated" only "predicted"
- Since a_j and b_j aren't estimated, they must be integrated out of the likelihood function, leaving an integrated likelihood function of the common parameters
- After obtaining maximum likelihood estimates of the common parameters, each a_j and b_j can be predicted from the residuals via a regression
- Estimated standard errors produced by frequentist software are too small
- There are no standard errors, p-values, etc. for the a_j and b_j
- Maximum likelihood estimation often results in a corner solution

Data-Generating Processes for Multilevel Models

Bayesian

$$\sigma_a \sim ?$$
$$\sigma_b \sim ?$$
$$\rho \sim ?$$
$$\forall j : a_j, b_j \sim \mathcal{N}_2\left(\left(0,0\right)^\top, \sigma_a, \sigma_b, \rho\right)$$
$$\alpha \sim ?$$
$$\beta \sim ?$$
$$\forall i,j : \mu_{ij} = \alpha + \beta x_{ij} + a_j + b_j x_{ij}$$
$$\sigma_y \sim ?$$
$$\forall i,j : \epsilon_{ij} \sim \mathcal{N}\left(0, \sigma_y\right)$$
$$\forall i,j : y_{ij} \equiv \mu_{ij} + \epsilon_{ij}$$

Frequentist

α is given

β is given

$$\forall i,j : \mu_{ij} = \alpha + \beta x_{ij}$$

σ_a is given

σ_b is given

ρ is given

$$\forall j : a_j, b_j \sim \mathcal{N}_2\left(\left(0,0\right)^\top, \sigma_a, \sigma_b, \rho\right)$$

σ_y is given

$$\forall i,j : \epsilon_{ij} \sim \mathcal{N}\left(a_j + b_j x_n, \sigma_y\right)$$
$$\forall i,j : y_{ij} \equiv \mu_{ij} + \epsilon_{ij}$$

Table 2 from the lme4 Vignette (see also the FAQ)

Formula	Alternative	Meaning
`(1 | g)`	`1 + (1 | g)`	Random intercept with fixed mean
`0 + offset(o) + (1 | g)`	`-1 + offset(o) + (1 | g)`	Random intercept with *a priori* means
`(1 | g1/g2)`	`(1 | g1)+(1 | g1:g2)`	Intercept varying among `g1` and `g2` within `g1`
`(1 | g1)+(1 | g2)`	`1 + (1 | g1) + (1 | g2)`	Intercept varying among `g1` and `g2`
`x + (x | g)`	`1 + x + (1 + x | g)`	Correlated random intercept and slope
`x + (x || g)`	`1 + x + (1 | g) + (0 + x | g)`	Uncorrelated random intercept and slope

Table 2: Examples of the right-hand sides of mixed-effects model formulas. The names of grouping factors are denoted `g`, `g1`, and `g2`, and covariates and *a priori* known offsets as `x` and `o`.

lme4 syntax

Hierarchical Models in Psychology

- In political science and economics, the “big” units are often countries or sub-national political areas like states and the “small” units are people
- In psychology, the “big” units are often people and the “small” units are questions or outcomes on repeated tasks
- Hierarchical model syntax is like

```
y ~ x + (x | person) + (1 | question)
```

- Question of interest is how to predict `y` for a new “big” unit (person), as opposed to predicting how well an old “big” unit will answer a new “small” unit (question), but you could do either

Hierarchical Models in rstanarm (from this paper)

```
dat <- readr::read_csv("https://osf.io/5cg32/download")
```

```
post <-  stan_glmer(valence ~ arousal + (1 + arousal | PID), data = dat,
                    prior = normal(0, 1), prior_intercept = normal(50, 20),
                    prior_aux = exponential(rate = 0.1))
```

```
post
```

```
...
##             Median MAD_SD
## (Intercept) 29.6    5.4
## arousal      0.5    0.1
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 9.2    0.4
```

```
##
## Error terms:
##  Groups   Name        Std.Dev. Corr
##  PID      (Intercept) 20.48
##           arousal      0.24    -0.64
##  Residual              9.27
## Num. levels: PID 20
##
...
```

Accessor Functions (based on the lme4 package)

```
fixef(post) # posterior medians in rstanarm
```

```
## (Intercept)     arousal
##  29.5960196   0.5312932
```

```
cbind(b = head(ranef(post)$PID), total = head(coef(post)$PID))
```

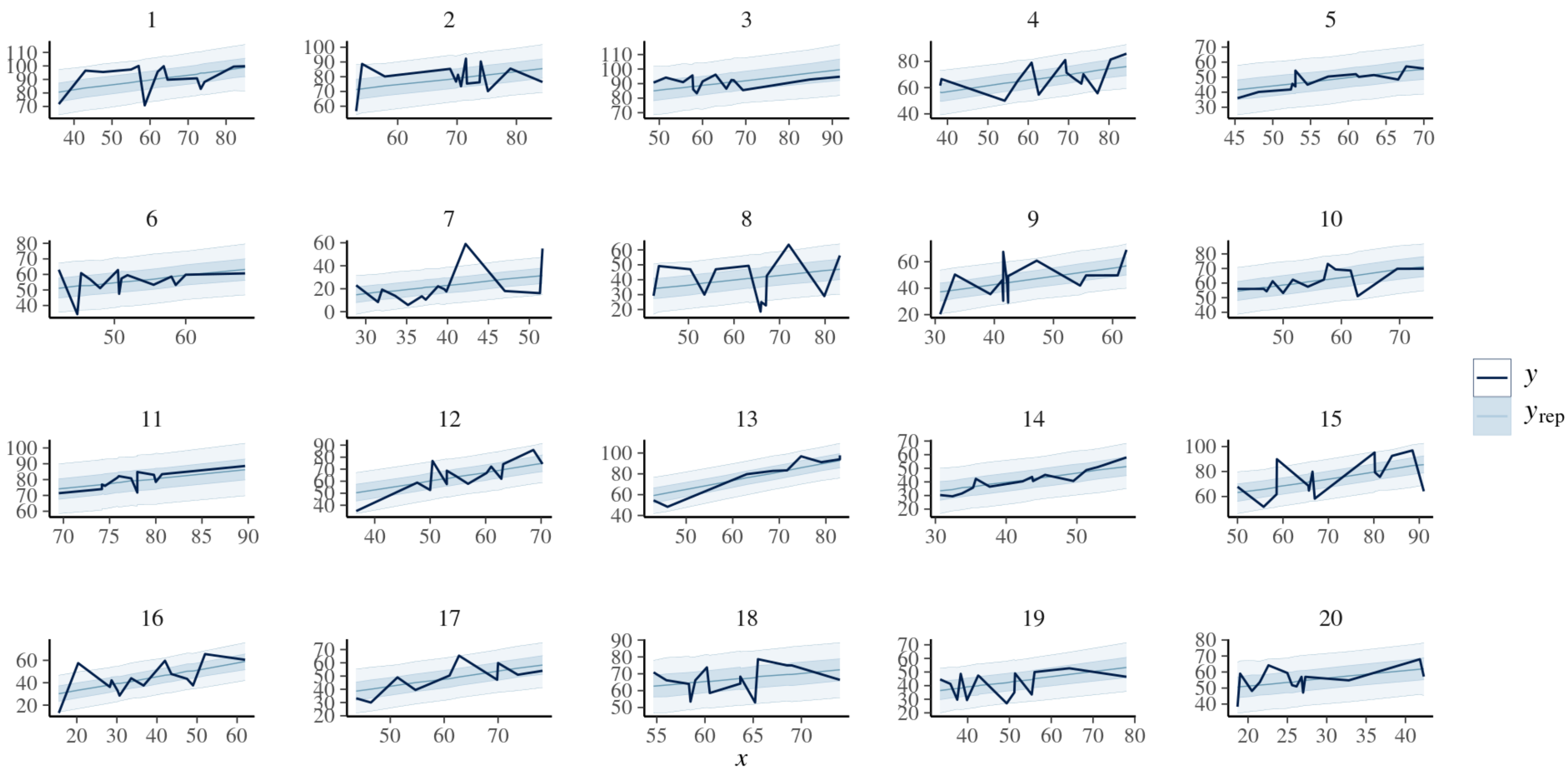
```
##   b.(Intercept)   b.arousal total.(Intercept) total.arousal
## 1     37.079597 -0.15583881          66.67562     0.3754544
## 2     17.180126 -0.07219256          46.77615     0.4591007
## 3     38.229115 -0.18936363          67.82513     0.3419296
## 4     10.159159 -0.10321874          39.75518     0.4280745
## 5    -13.916902  0.02652458          15.67912     0.5578178
## 6      1.370325 -0.05797636          30.96634     0.4733169
```

```
dim(as.matrix(post)) # 4000 x 46
```

```
## [1] 4000   46
```

Posterior Predictive Checks

```
pp_check(post, plotfun = "ribbon_grouped", x = dat$arousal, group = dat$PID)
```



Posterior Prediction

```
PPD <- posterior_predict(post) # of previous people
nd <- dat[dat$PID == 1, ]
nd$PID <- 0L # a new person
PPD_0 <- posterior_predict(post, newdata = nd) # 4000 x 14
```

- How is that even possible? For each of the S posterior draws, $\dots$

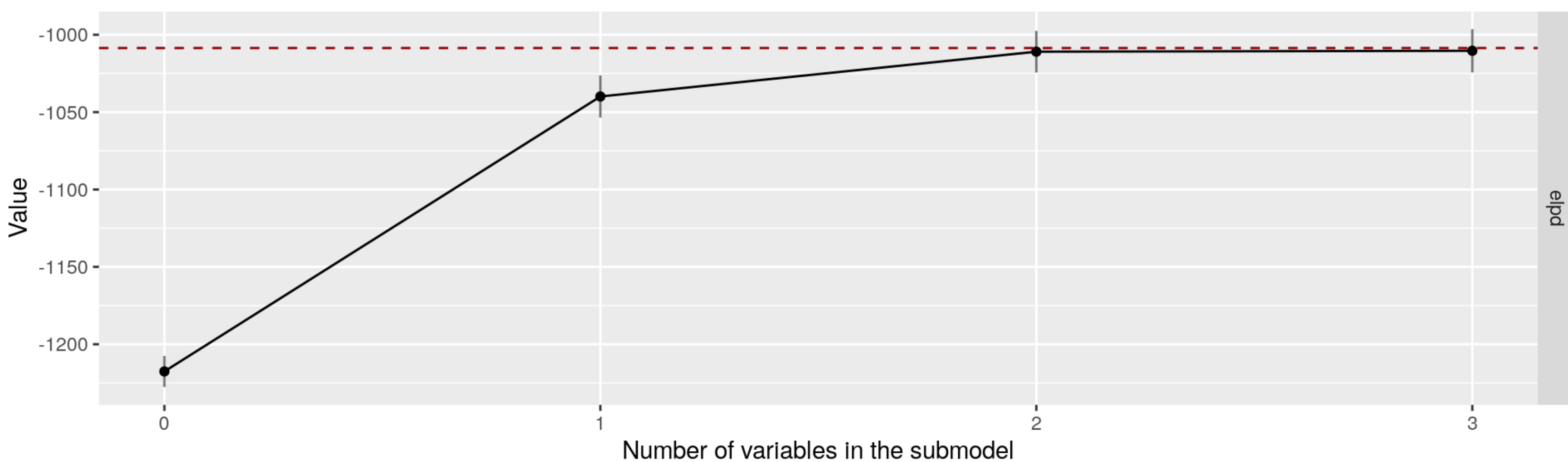
1. Draw a_0 and b_0 from a bivariate normal with means zero and covariance matrix Σ
2. Form $\boldsymbol{\mu}_0 \equiv \alpha + a_0 + (\beta + b_0)\,\mathbf{x}$
3. Draw each ϵ_t from a normal distribution with mean zero and standard deviation σ
4. Form $\mathbf{y}_0 = \boldsymbol{\mu}_0 + \boldsymbol{\epsilon}$

Projection Pursuit

- When you simplify a model by assuming some parameter “is” zero, you must propagate your uncertainty as to whether that parameter actually is zero to the other, nonzero parameters when estimating the ELPD

```
library(projpred); library(optimx); cvv <- cv_varsel(post)
```

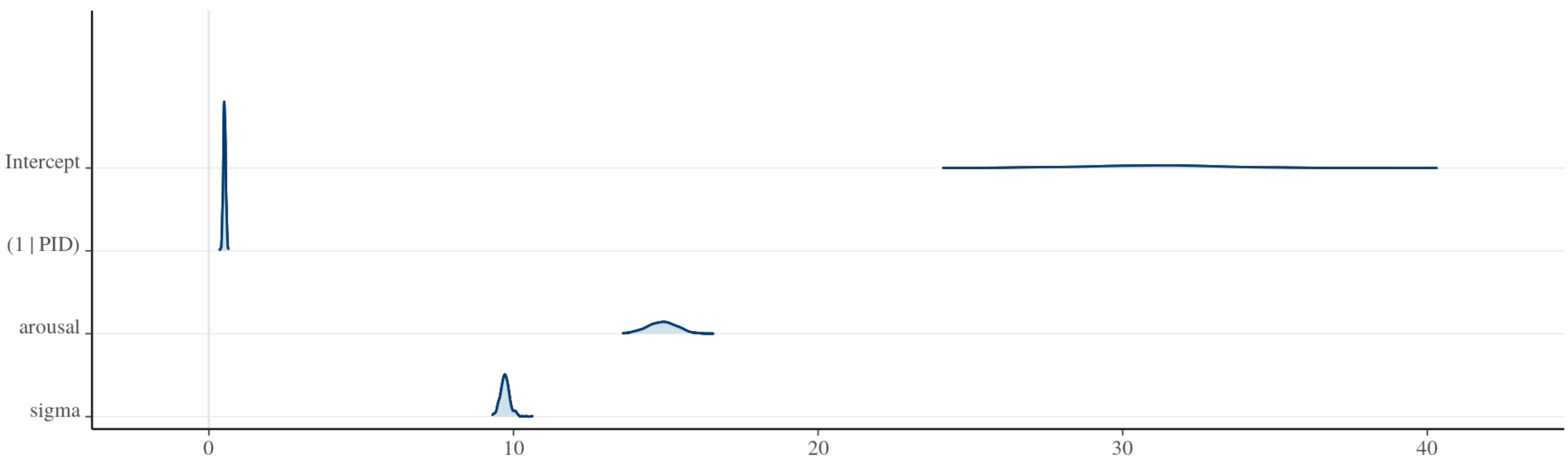
```
plot(cvv)
```



Finding the Optimal Submodel

```
summary(cvv); bayesplot::mcmc_areas_ridges(as.matrix(projpred::project(cvv, nterms = 2)))
```

```
##   size  solution_terms      elpd  elpd.se
## 2    0            <NA> -1217.564 10.09339
## 3    1        (1 | PID) -1039.894 13.69747
## 4    2          arousal -1010.938 13.47416
## 5    3 (arousal | PID) -1010.357 14.05326
```



Frequentist Multilevel Model Example

```
poll <- readRDS("GooglePoll.rds") # WantToWin is coded as 1 for Romney and 0 for Obama
poll$Income[poll$Income == "150,000+"] <- "100,000-149,999" # too few rich people
library(dplyr)
collapsed <- filter(poll, !is.na(WantToWin)) %>%
             group_by(Region, Gender, Urban_Density, Age, Income) %>%
             summarize(Romney = sum(grepl("Romney", WantToWin)), Obama = n() - Romney) %>%
             na.omit
```

```
lme4::glmer(cbind(Romney, Obama) ~ Gender + Urban_Density + Age + Income +
              (Gender + Urban_Density + Age + Income | Region),
             data = collapsed, family = binomial(link = "logit"))
```

```
## boundary (singular) fit: see ?isSingular
```

- For models that are more complicated than `(1 + x | g)`, the MLE of $\boldsymbol{\Sigma}$ — which is the covariance among the group-specific intercepts and slopes — usually implies that $\widehat{\boldsymbol{\Sigma}}^{-1}$ does not exist

Bayesian Version of the “Same” Model Works Fine

But needed a prior on a 13×13 variance-covariance matrix, $\mathbf{\Sigma}$

```
post_h <- stan_glmer(cbind(Romney, Obama) ~ Gender + Urban_Density + Age + Income +
                       (Gender + Urban_Density + Age + Income | Region),
                     data = collapsed, family = binomial(link = "logit"),
                     QR = TRUE, adapt_delta = 0.98, seed = 12345)
```

```
VarCorr(post_h) # posterior means of elements of Sigma
```

```
##  Groups Name                  Std.Dev. Corr
##  Region (Intercept)           0.171258
##         GenderMale            0.092365 -0.004
##         Urban_DensitySuburban 0.098756 -0.024 -0.009
##         Urban_DensityUrban    0.098112  0.029  0.003  0.065
##         Age25-34              0.108125  0.058 -0.009  0.003  0.012
##         Age35-44              0.123529  0.003  0.023  0.021  0.015  0.015
##         Age45-54              0.115544  0.082 -0.016 -0.006  0.020  0.054 -0.042
##         Age55-64              0.103972  0.010 -0.005 -0.013  0.006  0.030 -0.003  0.054
##         Age65+                0.110897  0.031 -0.015 -0.010  0.011  0.031 -0.016  0.045
##         Income25,000-49,999   0.114640 -0.083 -0.025 -0.041 -0.020 -0.012 -0.078  0.007
##         Income50,000-74,999   0.112559 -0.035  0.002  0.015 -0.004 -0.003  0.019 -0.005
##         Income75,000-99,999   0.129755 -0.036  0.030 -0.010  0.004 -0.026  0.033 -0.045
##         Income100,000-149,999 0.130075  0.022 -0.005  0.013 -0.008  0.008 -0.008  0.013
##
##
##
##
##
##
##
##
##
```

Poststratification

- Posterior distributions are conditional on the data you collected, which may or may not be a random sample or otherwise representative of a population, so how do you make principled claims about a population?
- Frequentists utilize weights while estimating parameters; Bayesians use (different) weights after estimating the parameters.

```
mu <- posterior_epred(post_h); dim(mu)
```

```
## [1] 4000  513
```

- Assume `shares` is the proportion of voters for each level of `Gender`, `Urban_Density`, `Age`, and `Income` crossed with `Region`

```
mu_US <- mu %*% shares # matrix-vector multiplication yields a vector of size 4000
```

- Now you have a posterior distribution for the proportion supporting Romney for the United States as a whole

PSISLOOCV (of a group, assuming that is sensible)

```
(loo_hier <- loo(post_h)) # 156 nominal parameters but much fewer effective parameters
```

```
...
##          Estimate   SE
## elpd_loo   -845.9 20.5
## p_loo        27.8  2.3
## looic      1691.9 41.1
## ------
## Monte Carlo SE of elpd_loo is 0.1.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
NA
NA
NA
NA
...
```

What Were the Priors?

```
prior_summary(post_h)
```

```
## Priors for model 'post_h'
## ------
## Intercept (after predictors centered)
##  ~ normal(location = 0, scale = 2.5)
##
## Coefficients (in Q-space)
##  ~ normal(location = [0,0,0,...], scale = [2.5,2.5,2.5,...])
##
## Covariance
##  ~ decov(reg. = 1, conc. = 1, shape = 1, scale = 1)
## ------
## See help('prior_summary.stanreg') for more details
```

What Is **`decov(1, 1, 1, 1)`**?

- `decov` = Decomposition of Covariance
- `reg.` is the regularization parameter in the LKJ prior on the correlation matrix
- `conc.` is the concentration parameter in the Dirichlet prior on the variance components
- `shape` and `scale` pertain to the Gamma prior on multiplier for the variance components
- You usually do not need to change these defaults to get good results

McElreath / Kotz Example

```
library(brms)
funding <-
  tibble(
    discipline   = rep(c("Chemical sciences", "Physical sciences", "Physics", "Humanities",
                         "Technical sciences",  "Interdisciplinary", "Earth/life sciences",
                         "Social sciences", "Medical sciences"),
                     each = 2),
    gender       = rep(c("m", "f"), times = 9),
    applications = c(83, 39, 135, 39, 67, 9, 230, 166, 189, 62, 105, 78, 156, 126, 425, 409, 24
    awards       = c(22, 10, 26, 9, 18, 2, 33, 32, 30, 13, 12, 17, 38, 18, 65, 47, 46, 29),
    rejects      = c(61, 29, 109, 30, 49, 7, 197, 134, 159, 49, 93, 61, 118, 108, 360, 362, 19
    male         = ifelse(gender == "f", 0, 1) %>% as.integer()
  )

b13.bonus_2 <-
  brm(awards | trials(applications) ~ 1 + male + (1 + male | discipline),
      data = funding, family = binomial, control = list(adapt_delta = 0.92),
      prior = c(prior(normal(0, 4), class = Intercept), prior(normal(0, 4), class = b),
                prior(cauchy(0, 1), class = sd), prior(lkj(4), class = cor)))
```

```
## Compiling Stan program...
```

```
## Start sampling
```

Overall Results

```
b13.bonus_2
```

```
##  Family: binomial 
##   Links: mu = logit 
## Formula: awards | trials(applications) ~ 1 + male + (1 + male | discipline) 
##    Data: funding (Number of observations: 18) 
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
## 
## Group-Level Effects: 
## ~discipline (Number of levels: 9) 
##                       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)             0.28      0.14     0.05     0.60 1.00     1050     1166
## sd(male)                  0.33      0.17     0.04     0.72 1.00      899      885
## cor(Intercept,male)      -0.18      0.31    -0.72     0.47 1.00     2078     2742
## 
## Population-Level Effects: 
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept    -1.62      0.14    -1.88    -1.32 1.00     1927     2257
## male          0.15      0.17    -0.20     0.49 1.00     2189     1912
## 
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Disclipline Specific Results

```r
print(coef(b13.bonus_2),
      digits = 3)
```

```
## $discipline
## , , Intercept
##
##                       Estimate Est.Error  Q2.5  Q97.5
## Chemical sciences        -1.41     0.260 -1.86 -0.859
## Earth/life sciences      -1.65     0.181 -2.02 -1.305
## Humanities               -1.59     0.168 -1.89 -1.251
## Interdisciplinary        -1.59     0.213 -1.98 -1.153
## Medical sciences         -1.87     0.158 -2.21 -1.590
## Physical sciences        -1.53     0.230 -1.94 -1.038
## Physics                  -1.50     0.283 -2.01 -0.915
## Social sciences          -1.92     0.139 -2.20 -1.660
## Technical sciences       -1.58     0.211 -1.97 -1.139
##
```

```
## , , male
##
##                       Estimate Est.Error    Q2.5 Q97.5
## Chemical sciences       0.2503     0.283 -0.3114 0.819
## Earth/life sciences     0.3931     0.243 -0.0250 0.900
## Humanities             -0.0947     0.234 -0.5913 0.311
## Interdisciplinary      -0.1677     0.315 -0.8632 0.328
## Medical sciences        0.3396     0.211 -0.0267 0.792
## Physical sciences       0.0964     0.259 -0.4642 0.578
## Physics                 0.3037     0.306 -0.2534 0.941
## Social sciences         0.2023     0.171 -0.1252 0.559
## Technical sciences     -0.0169     0.246 -0.5620 0.420
```

- In light of the considerable uncertainty for a department, these data are consistent with both discrimination and no discrimination by sex