Mixture Models

EES 5891-03

Bayesian Statistical Methods

Jonathan Gilligan

Class #17: Thursday, October 27 2022

Poisson Models

Poisson Models

- *Binomial* model:
  - Number of events out of N trials, with probability p of an event in any trial.
    - Maximum number of events is N.
- *Poisson* model:
  - Limit of many trials with small p.
    - No upper limit, but the average is finite.
  - If a random event has a constant probability of happening in an interval of time
    - *Exponential* distribution describes time between events
    - *Poisson* distribution describes # events per interval
- Examples
  - Radioactive decay
  - Storms hitting a location
  - Eruptions in an active volcano (e.g., Etna, Mauna Loa)
  - Bicycle commuter traffic
  - Customers visiting a business
  - Incidence of cancer

Poisson Distribution

- Poisson distribution

$$P(k|\lambda) = \frac{\lambda^k e^{-\lambda}}{k!}$$

  - $\text{mean}(k) = \lambda$
  - $\text{standard deviation}(k) = \sqrt{\lambda}$

Overdispersed Data

- *Poisson* has one parameter λ, so *mean* and *standard deviation* are not independent.
  - mean = λ
  - standard deviation = $\sqrt{\lambda}$ = $\sqrt{\textbf{mean}}$
- What happens if standard deviation > $\sqrt{\textbf{mean}}$?
- *Gamma Poisson* model (also known as *Negative binomial* model)

$$\text{Gamma Poisson}(k|r,p) = \int_0^\infty \text{Poisson}(k|\lambda)\text{Gamma}\left(\lambda \middle| r, \frac{1-p}{p}\right) d\lambda$$

  - This is a combination of Poisson distributions for many values of λ
  - Example:
    - Cancer:
      - *Poisson* describes cancer incidence if everyone has the same risk
      - *Gamma Poisson* describes cancer incidence for a heterogeneous population where different people have different risks.

Examples of Gamma-Poisson/Negative Binomial models

- Hurricanes:
  - *Poisson* describes # hurricanes per year if all years are the same
  - *Gamma Poisson* accounts for climatic variation, such as El Niño, that affects hurricane frequency.
- Bicycle commuters crossing Brooklyn Bridge each day
  - *Poisson* describes # commuters per day if all commuters have the same probability of cycling each day
  - *Gamma Poisson*: Commuters are heterogeneous: different people have different probability of cycling.
    - Weather affects rates too.
      - *Gamma Poisson* accounts for this with random variation in λ
      - You could also explicitly model effect of weather on λ.
- Volcanic eruptions: Chih-Hsiang Ho. 1990. Bayesian analysis of volcanic eruptions. *J. Volcan. Geotherm. Res.* 43:91–98.
  - *Poisson* assumes equal likelihood of eruption each year
  - *Gamma-Poisson* accounts for unpredictable variation in activity over time

Mixture Models

Mixture Models

- Simple models assume a homogeneous population
  - Each individual, or each slice of time, is the same as all others
- Mixture models account for heterogeneous populations
  - The probability distribution for the whole population is a *mixture* of different distributions for the members of the population
- Common mixtures:
  - Overdispersed models (continuous or discrete)
    - *Gamma Poisson/Negative Binomial* (mixture of *Poisson* distributions with different λ)
    - *Beta Binomial* (mixture of *Binomial* distributions with different p)
    - *Student-t* (mixture of *Normal* distributions with different σ)
  - Zero-inflated models (continuous or discrete)
    - Mixture of individuals for whom effect is strictly zero with others for whom there is a distribution of effects

Mixture Models for Categorical Data

- Binomial describes 2 possible outcomes
- Multinomial (categorical) describes *k* possible outcomes
- Ordered categorical describes *k* outcomes that have a sequence or ranking order.
  - Mixtures of many binary logistic models
  - Use of *cumulative link* functions

Overdispersed Counts

Beta-Binomial Models

- Dig deeper into the graduate school admissions data

```
data(UCBadmit)
d <- UCBadmit %>%
  mutate(applicant.gender =
           ordered(applicant.gender,
                   levels = c("male", "female")),
         gid = as.integer(applicant.gender))
```

```
kable(d)
```

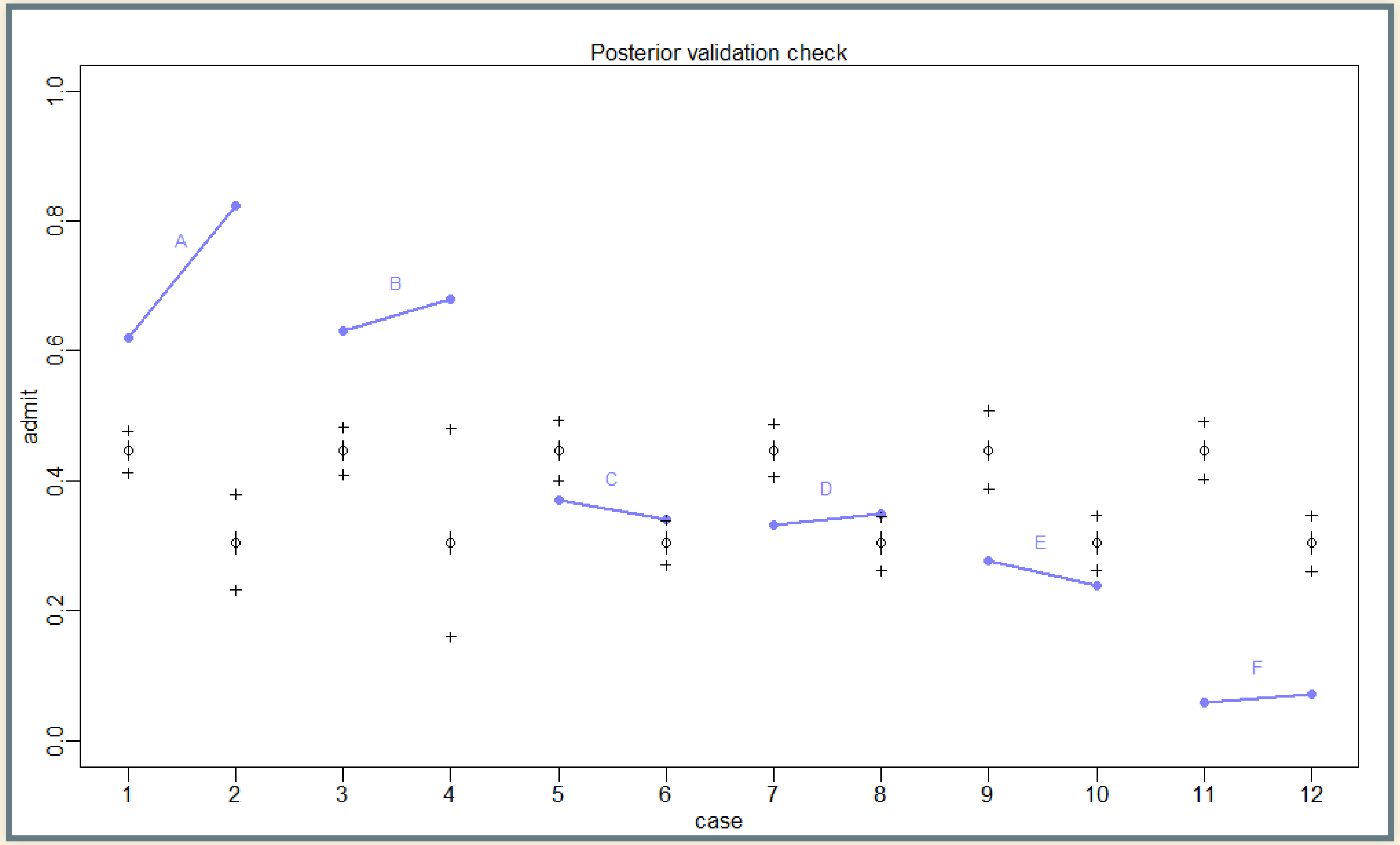
dept	applicant.gender	admit	reject	applications	gid
A	male	512	313	825	1
A	female	89	19	108	2
B	male	353	207	560	1
B	female	17	8	25	2
C	male	120	205	325	1
C	female	202	391	593	2
D	male	138	279	417	1
D	female	131	244	375	2
E	male	53	138	191	1
E	female	94	299	393	2
F	male	22	351	373	1
F	female	24	317	341	2

Model that Ignores Department

```
dat_list <- select(d, admit, applications, gid)
```

```
mdl_gs <- ulam(
  alist(
    admit ~ dbinom(applications, p),
    logit(p) <- a[gid],
    a[gid] ~ dnorm(0, 1.5)
  ), data = dat_list, chains = 4, cores = 4)
```

Posterior Validation Check



- The data are overdispersed, compared to the model
  - Binomial distribution: mean = Np, standard deviation = $\sqrt{Np(1-p)}$
  - Model:

```
predicted_draws(mdl_gs, newdata = dat_list) %>%
  group_by(gid) %>%
  summarize(mean = mean(.prediction / applications),
            sd   =   sd(.prediction / applications))
        %>%
  kable(digits = 2)
```

gid	mean	sd
1	0.45	0.03
2	0.30	0.05

  - Data:

```
dat_list %>%
  group_by(gid) %>%
  summarize(mean = mean(admit / applications),
            sd   =   sd(admit / applications)) %>%
  kable(digits = 2)
```

gid	mean	sd

gid	mean	sd
1	0.38	0.22
2	0.42	0.28

- Standard deviation in the data is 5–7 times greater than the model predicts.

Beta Binomial Model

- Our model uses a binomial likelihood

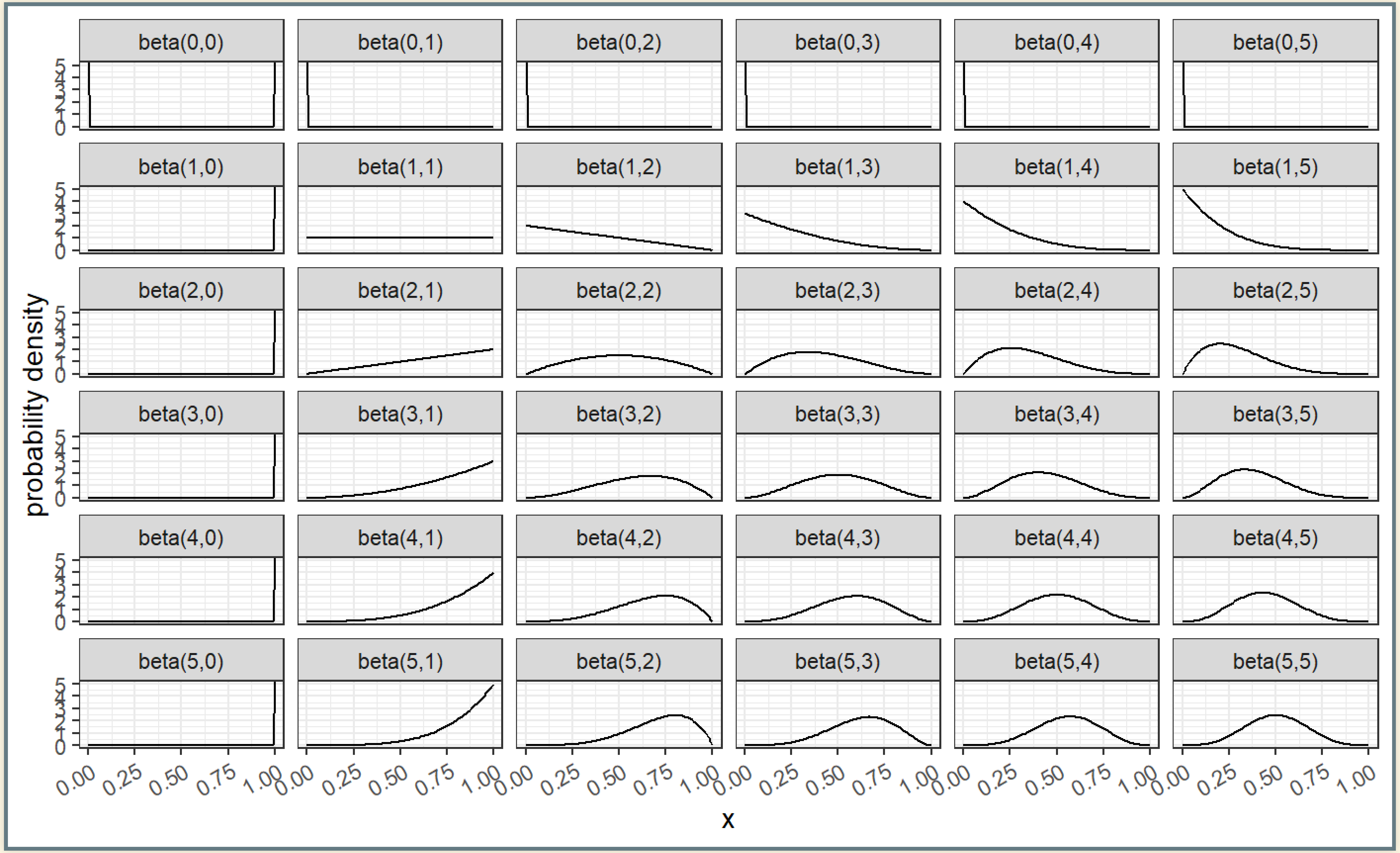
$$A_i \text{ Binomial}(N_i, p_{\text{gender}_i}),$$

  where A is the # of admissions, N is the # of applications, and p is the probability of getting admitted.
- On Tuesday we developed an alternative model in which p varied from department to department.
- But if we don't know who applied to which department, we can account for this variation by assuming that p varies randomly from one student to the next.
  - p must lie in the range [0,1], so we assume that the probabilities p are described by a *beta* distribution.
- Beta Binomial Distribution:

$$\text{BetaBinomial}(N, \bar{p}, \theta)$$

  - $\bar{p}$ is the average probability
  - θ is the *dispersion* (amount of variation in p).

Beta Distributions



- We saw this in chapter 2, as the posteriors for the fraction of water covering the Earth.

Beta Binomial Model

- New model:

$$
\begin{aligned}
A &\sim \text{BetaBinomial}(N, \bar{p}, \theta) \\
\text{logit}(\bar{p}) &= \alpha_{\text{gender}} \\
\alpha_{\text{gender}} &\sim \text{Normal}(0, 1.5) \\
\theta &= \phi + 2 \\
\phi &\sim \text{Exponential}(1)
\end{aligned}
$$

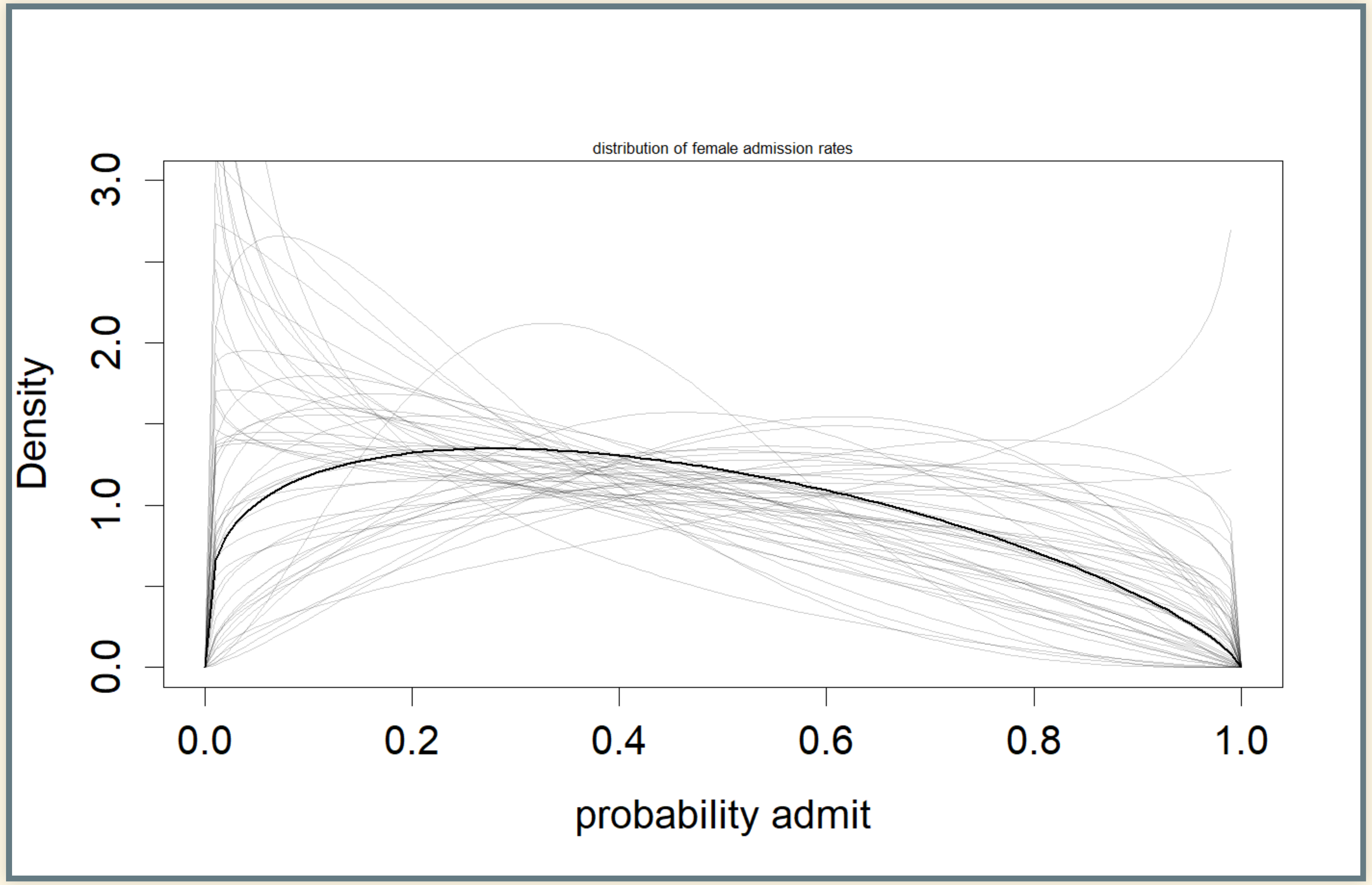
  - Trick: The dispersion θ is related to the sum of the parameters for *beta*, so we want it to be at least 2 (flat, uniform), so we don't pile up probability at 0 or 1.

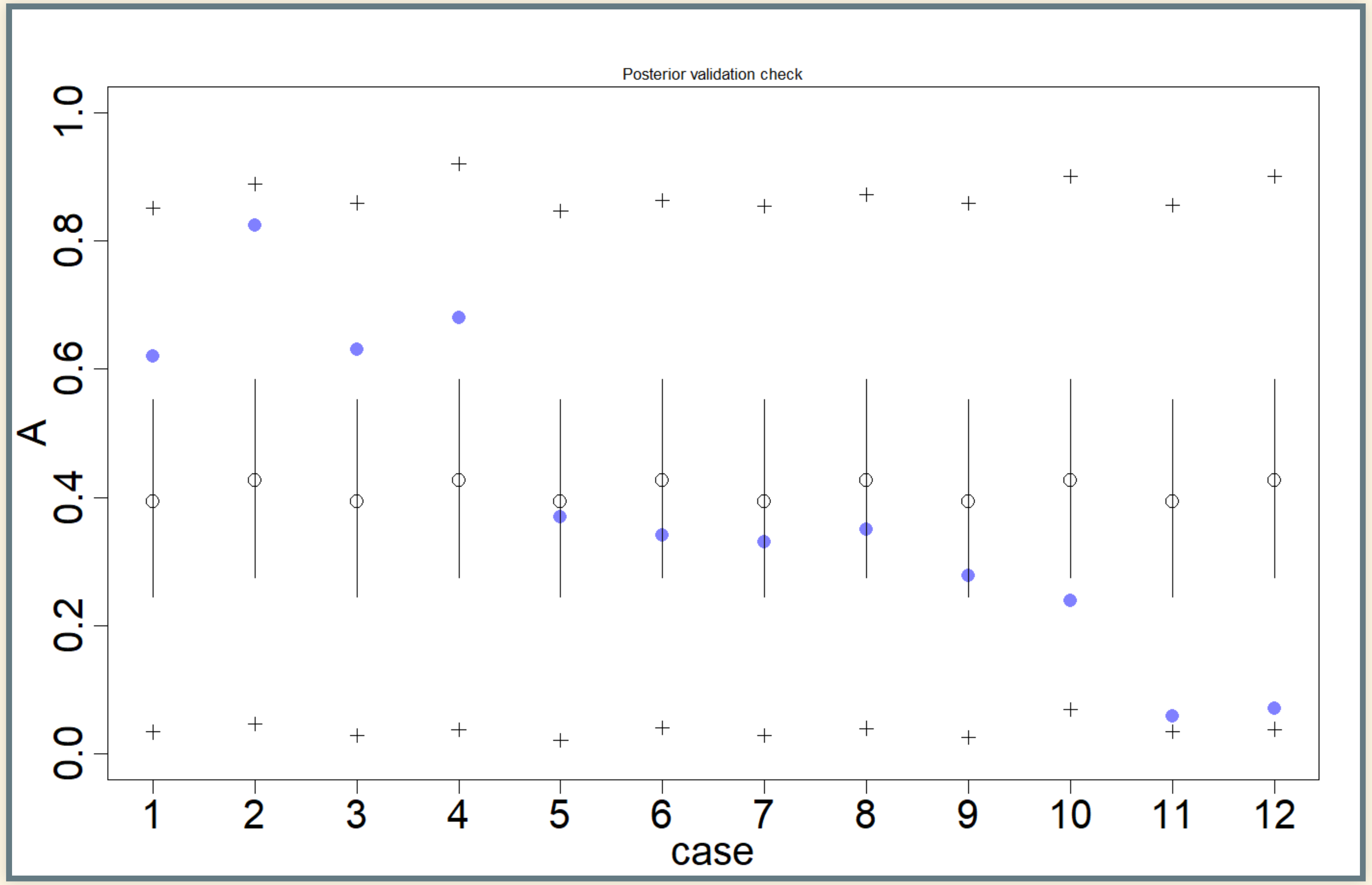
```
dat_list <- select(d, A = admit, N = applications, gid)

mdl_gs_bb <- ulam(
    alist(
        A ~ dbetabinom(N, pbar, theta),
        logit(pbar) <- a[gid],
        a[gid] ~ dnorm(0, 1.5),
        transpars> theta <<- phi + 2.0,
        phi ~ dexp(1)
    ), data =dat_list, chains = 4, cores = 4 )
```

```
post <- extract.samples(mdl_gs_bb)
post$da <- post$a[,1] - post$a[,2]
precis_show(precis(mdl_gs_bb, depth = 2, digits = 2))
```

```
##          mean   sd  5.5% 94.5% n_eff Rhat4
## a[1]    -0.45 0.42 -1.13  0.21  1294     1
## a[2]    -0.31 0.42 -0.97  0.34  1156     1
## phi      1.01 0.78  0.09  2.40  1610     1
## theta    3.01 0.78  2.09  4.40  1610     1
```

Posterior Checks



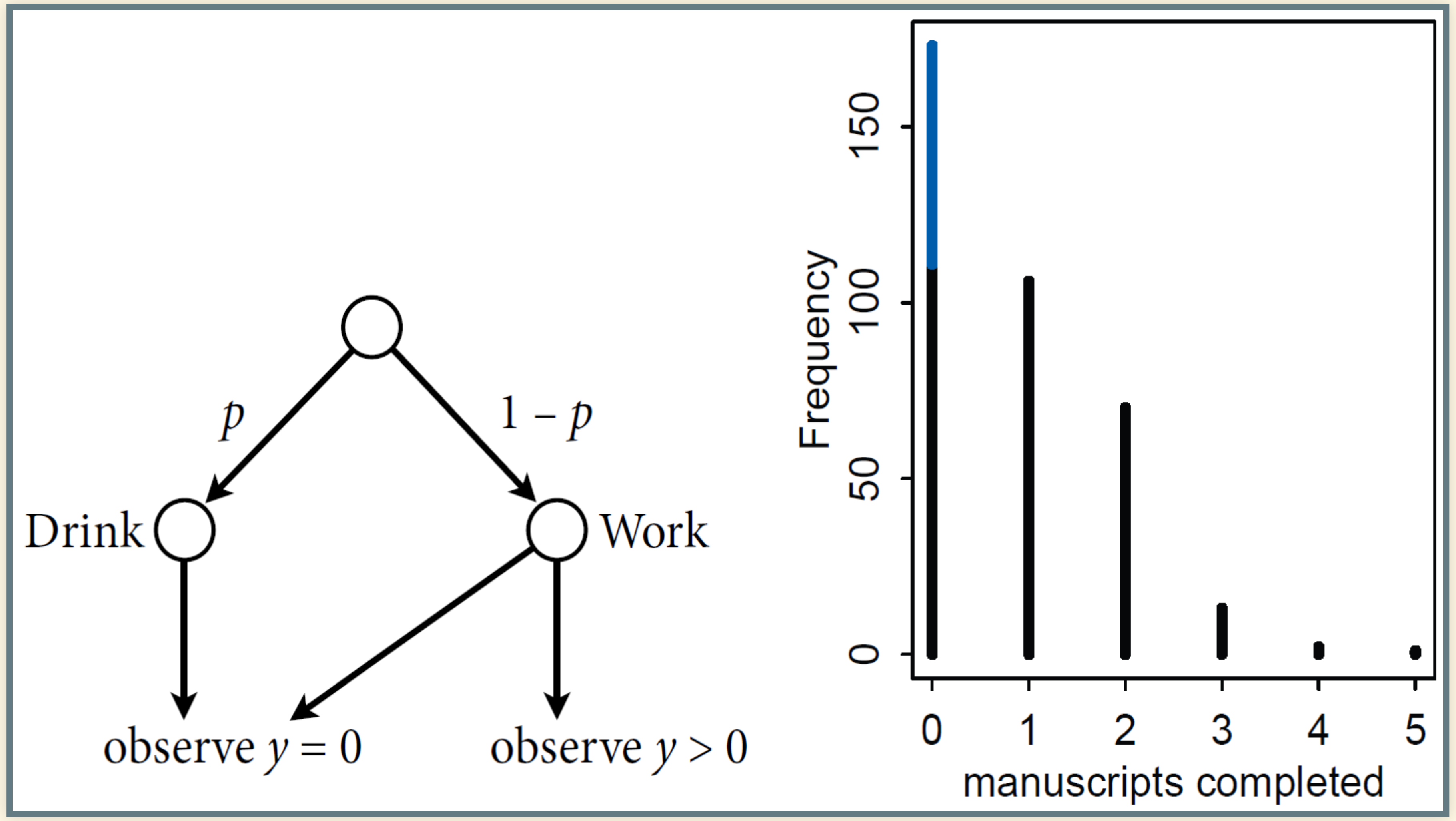


- Warning: Be very cautious using WAIC or PSIS with mixture models. It is not straightforward to interpret what the information criteria mean.

Zero-Inflated Models

Zero-Inflated Models

- This is like an extreme version of overdispersion.
  - Some individuals have zero probability of producing an event, and others have some probabilty distribution.
- Example: Monks and Manuscripts
  - Monks in the monastery have a probability p to spend the day drinking instead of working on manuscripts. When they do work on manuscripts, there is a Poisson distribution of manuscripts completed in a day, with mean λ.



- The histogram of manuscripts completed per day looks just like a Poisson distribution, but with an extra bit added to zero.

Writing the Model in R

- Generate data

```r
prob_drink <- 0.2 # 20% of days
rate_work <- 1    # average 1 manuscript per day
N <- 365

set.seed(736)
drink <- rbinom( N , 1 , prob_drink )
y <- (1-drink)*rpois( N , rate_work )
```

- Make the model

```r
mdl_zinf <- ulam(
    alist(
        y ~ dzipois(p, lambda),
        logit(p) <- ap,
        log(lambda) <- al,
        ap ~ dnorm(-1.5, 1 ),
        al ~ dnorm(1 , 0.5 )
    ), data=list(y = y), chains = 4, cores = 4)
```

- Analyze the results

```r
precis_show(precis( mdl_zinf, digits = 2))
```

```
##      mean   sd  5.5% 94.5% n_eff Rhat4
## ap -1.43 0.38 -2.06 -0.95   690  1.01
## al  0.09 0.08 -0.05  0.22   656  1.00
```

```r
post <- extract.samples(mdl_zinf)
mean(inv_logit( post$ap)) %>% round(2) # probability of
        drinking
```

```
## [1] 0.2
```

```r
# rate of finishing manuscripts when not drinking
mean(exp( post$al)) %>% round(2)
```

```
## [1] 1.09
```

Ordered Categorical Models

Categorical Models

- Multinomial data
  - Predict what subject high-school students will major in when they go to college
    - For simplicity, assume no double-majors
  - For N students, and K possible subjects to major in,

$$\Pr(y_1, y_2, \ldots, y_K | N, p_1, p_2, \ldots, p_K) = \frac{N!}{\prod_i y_i!} \prod_{i=1}^{K} p_i^{y_i}$$

  - Constraint:

$$\sum_{i=1}^{K} p_i = 1$$

    - *Softmax* (*multinomial logit*) link function ensures the probabilities add up to 1.

Ordered Categorical Outcome Variables

- Analyze survey data with Likert responses (scale of 1–5, 1–7, etc.) Predict Likert response, based on demographic data
  - “How much do you approve of Joe Biden?”
    1. Strongly disapprove
    2. Somewhat disapprove
    3. Neither approve nor disapprove
    4. Somewhat approve
    5. Strongly approve
  - **Warning:** It is not legitimate to just assume that Likert scores are like a *metric* variable with a numerical value.
    - Is the difference between 1 and 2 the same as between 3 and 4?
  - Instead analyze outcome variable as categorical, but with the constraint that it’s ordered (1 comes before 2, which comes before 3, …).

Modeling Ordered Categorical Outcomes

- Ordered logit function: K logit functions $f_1, f_2, \ldots, f_K$, where each one is centered on a *cutpoint* $C_1, C_2, \ldots, C_K$.

Boxcar Problem

A boxcar is rolling down some train tracks out of control. Five people are in the way Dennis can pull a lever and send the boxcar down a different track, which will save the five people but kill one other person.

- Three important psychological principles:
  1. *Action principle:** Harm caused by action is worse than harm caused by failing to act.
  2. **Intention principle:** Harm intended as the means to a goal is worse than harm as an unindended side-effect
  3. **Contact principle:** Using physical contact to cause harm is worse than causing equivalent harm without physical contact.
- Experiment: Ask people the question with different prompts that introduce one or more of these principles.
  - Response is a number 1–7 indicating how morally permissible it is to throw the switch.

Model to predict people's answers

- Set up the data

```
data(Trolley)
d <- Trolley
dat <- list(
  R = d$response,
  A = d$action,
  I = d$intention,
  C = d$contact )
```

- The model

```
mdl_boxcar <- ulam(
  alist(
      R ~ dordlogit(phi, cutpoints),
      phi <- bA * A + bC * C + BI * I,
      BI <- bI + bIA * A + bIC * C ,
      c(bA,bI,bC,bIA,bIC) ~ dnorm(0, 0.5),
      cutpoints ~ dnorm(0, 1.5)
  ), data=dat, chains=4, cores=4)
```

  - Note `c(bA,bI,bC,...)` instead of one line for each parameter.
  - `cutpoints ~ dnorm(0, 1.5)` automatically samples the right number of cutpoints but it runs into trouble because `dnorm` does not guarantee that the cutpoints are in the right order.