Setting Priors in brms

Bayesian Mixed Effects Models with brms for Linguists

Workshop Materials

2025-11-11

Table of contents

1 1. Setting Priors in brms (20 min) 1

1.1 Default vs. Weakly Informative Priors 1

1.2 Default brms Priors 2

1.2.1 IMPORTANT: The Intercept prior ADAPTS to your data! 2

1.3 Reaction Time Data (log-transformed) 3

1.3.1 Check default priors 3

1.3.2 Set weakly informative priors for RTs 3

1.3.3 Why normal() instead of brms default student_t()? 3

1.3.4 Why these numbers for RT priors? 4

1.4 Grammaticality Judgments (binary) 5

1.4.1 Default brms priors for logistic regression 5

1.4.2 Check default priors and set custom priors 5

1.4.3 Why normal() vs student_t() for logistic regression? 6

1.4.4 Why these numbers for logistic regression priors? 6

1.4.5 Quick reference: Log-odds to probability 7

1 1. Setting Priors in brms (20 min)

1.1 Default vs. Weakly Informative Priors

brms uses weakly informative priors by default (not completely flat). However, for psycholinguistics, domain-specific priors are even better.

1.2 Default brms Priors

What you get if you don't specify:

- **Intercept**: `student_t(3, mean(y), 2.5)` - DATA-DEPENDENT! Centers at your data mean
  - df = 3: heavy tails (allows outliers)
  - location = mean(y): adapts to your data scale
  - scale = 2.5: wide spread around the mean
  - For RT data with mean(log_rt) = 6: allows roughly 150ms-1100ms range
- **b (slopes)**: `(flat)` - improper uniform prior over (-∞, +∞), any effect size equally likely
  - "flat" means no information, completely uninformative
  - Technically improper: doesn't integrate to 1 (not a true probability distribution)
  - **How does this work?** The prior drops out of Bayes' theorem:
    - posterior likelihood × prior
    - If prior is constant (flat), posterior likelihood
    - So you get the maximum likelihood estimate (MLE), just with uncertainty from MCMC
  - Only slopes (b) get flat priors - everything else is weakly informative!
- **sigma**: `student_t(3, 0, 2.5)` with lower bound 0 - weakly informative
  - Mean 2.5, allows reasonable residual variance
  - Heavy tails permit some flexibility for noisy data
- **sd**: `student_t(3, 0, 2.5)` with lower bound 0 - weakly informative
  - Same as sigma, for random effects standard deviations
  - Encourages moderate but not extreme between-subject/item variation
- **cor**: `lkj(1)` - uniform over correlation matrices (-1 to +1 equally likely)
  - = 1: no preference for any correlation value
  - Uniform from perfect negative (-1) to perfect positive (+1) correlation

1.2.1 IMPORTANT: The Intercept prior ADAPTS to your data!

- If mean(log_rt) = 6, you get `student_t(3, 6, 2.5)`
- If mean(log_rt) = 10, you get `student_t(3, 10, 2.5)`
- This is reasonable but not ideal - better to use domain knowledge!
- See `materials/scripts/getprior.R` for verification.

1.3 Reaction Time Data (log-transformed)

1.3.1 Check default priors

```
get_prior(log_rt ~ condition + (1 + condition | subject) + (1 | item),
          data = rt_data, family = gaussian())
```

1.3.2 Set weakly informative priors for RTs

```
rt_priors <- c(
  prior(normal(6, 1.5), class = Intercept),        # log(RT) around 400ms
  prior(normal(0, 0.5), class = b),                # effects usually < 150ms
  prior(exponential(1), class = sigma),            # residual SD
  prior(exponential(1), class = sd),               # random effects SD
  prior(lkj(2), class = cor)                       # correlation matrix
)

# Fit model with priors
fit_rt <- brm(log_rt ~ condition + (1 + condition | subject) + (1 | item),
              data = rt_data, family = gaussian(),
              prior = rt_priors,
              sample_prior = "yes")  # Important for prior checks!
```

1.3.3 Why normal() instead of brms default student_t()?

- brms uses `student_t(3, , 2.5)` - has heavier tails than normal
- Student-t with df=3 allows more extreme values (outlier-robust)
- `normal( , )` is more concentrated around the mean (more informative)
- For psycholinguistics: we typically WANT to down-weight extreme values
  - RTs of 5000ms are possible but unlikely - normal() makes them less probable
  - Effects of 500ms are possible but unlikely - normal() regularizes them
- Student-t is good for defaults (conservative), but normal is better when you have domain knowledge about plausible ranges

Compare the tails:

```
quantile(rnorm(10000, 0, 0.5), c(0.001, 0.999))  # Normal: ±1.55
quantile(rt(10000, 3) * 0.5, c(0.001, 0.999))     # Student-t(3): ±3.18
# → Student-t allows values 2x more extreme!
```

1.3.4 Why these numbers for RT priors?

1.3.4.1 `normal(6, 1.5)` for Intercept:

- Mean = 6 on log scale → exp(6) 403ms (typical RT)
- SD = 1.5 → 95% prior interval: 6 ± 2*1.5 = [3, 9] on log scale
- This translates to exp(3) to exp(9) = 20ms to 8100ms (very wide!)
- But 95% of prior mass is between exp(6-1.96*1.5) to exp(6+1.96*1.5) 150ms-1100ms
- **"Prior interval"** = range on the parameter scale (log-RT)
- **"Prior mass"** = what that means for the actual quantity (milliseconds)
- Allows flexibility but down-weights extreme RTs like 1ms or 10 seconds

1.3.4.2 `normal(0, 0.5)` for effects (b):

- Mean = 0 (no assumption about direction)
- SD = 0.5 on log scale
- 95% prior interval: 0 ± 2*0.5 = [-1, 1] on log scale
- 95% prior mass: effects mostly between ±65% of baseline (multiplicative)
- Or approximately ±100-150ms for typical RTs around 400-600ms
- Regularizes extreme effect sizes (e.g., 500ms difference gets down-weighted)

1.3.4.3 `exponential(1)` for sigma (residual SD):

- Mean = 1, most mass near 0-2 (log scale)
- Translates to reasonable within-condition variability
- Penalizes very large residual variance

1.3.4.4 `exponential(1)` for sd (random effects SD):

- Mean = 1, encourages moderate between-subject/item variation
- Prevents overfitting with extreme random effect variance

1.3.4.5 `lkj(2)` for correlations:

- = 2: slight preference for correlations near 0 (skeptical of strong correlations)
- = 1: uniform (no preference)
- > 1: regularizing, prevents extreme correlations (±1)

1.4 Grammaticality Judgments (binary)

1.4.1 Default brms priors for logistic regression

What you get if you don't specify:

- **Intercept**: `student_t(3, 0, 2.5)` - FIXED at 0, NOT data-dependent!
  - df = 3: heavy tails
  - location = 0 on log-odds scale → 50% probability (plogis(0) = 0.5)
  - scale = 2.5: wide range on log-odds scale
  - Covers roughly 5%-95% accuracy range
  - Same prior whether your data has 30%, 70%, or 95% accuracy
- **b (slopes)**: `(flat)` - improper uniform, any effect size equally likely (same as RT models)
- **sd**: `student_t(3, 0, 2.5)` with lower bound 0 - weakly informative
  - Same as for Gaussian models, allows moderate random effects variation
- **cor**: `lkj(1)` - uniform over correlation matrices (-1 to +1 equally likely)
- **Note**: No sigma for binary data (bernoulli has no residual variance parameter)
  - Variance is determined by the probability: var = p(1-p)

KEY DIFFERENCE from Gaussian models: - Gaussian: Intercept prior adapts to mean(y) - Bernoulli: Intercept prior is ALWAYS `student_t(3, 0, 2.5)` - See `materials/scripts/getprior.R` for verification.

1.4.2 Check default priors and set custom priors

```r
# Check default priors
get_prior(correct ~ condition + (1 + condition | subject) + (1 | item),
          data = gram_data, family = bernoulli())

# Set priors for logistic regression
gram_priors <- c(
  prior(normal(0, 1.5), class = Intercept),        # log-odds scale
  prior(normal(0, 1), class = b),                  # effect sizes
  prior(exponential(1), class = sd),               # random effects SD
  prior(lkj(2), class = cor)
)

fit_gram <- brm(correct ~ condition + (1 + condition | subject) + (1 | item),
                data = gram_data, family = bernoulli(),
                prior = gram_priors,
                sample_prior = "yes")
```

1.4.3 Why normal() vs student_t() for logistic regression?

- Same reasoning as for Gaussian models:
- `student_t(3, 0, 2.5)` on log-odds scale allows extreme probabilities (1%, 99%)
- `normal(0, 1.5)` is more concentrated, regularizes toward middle ranges
- For grammaticality judgments: extreme accuracies (5%, 95%) are rare
- We want to down-weight implausible effect sizes

On log-odds scale, extreme values matter more:

```r
plogis(qnorm(0.999, 0, 1.5))    # Normal: 98.5% (plausible)
plogis(qt(0.999, 3) * 2.5)      # Student-t: 99.97% (implausible ceiling)
# → Student-t allows near-perfect performance too easily
```

1.4.4 Why these numbers for logistic regression priors?

1.4.4.1 `normal(0, 1.5)` for Intercept:

- Mean = 0 on log-odds scale → 50% probability (neutral)
- SD = 1.5
- 95% prior interval: 0 ± 2*1.5 = [-3, 3] on log-odds scale (approx, exact ±1.96*SD)
- 95% prior mass: intercept corresponds to ~5% to ~95% accuracy
- **"Prior interval"** = range on log-odds scale
- **"Prior mass"** = what that means for actual probabilities (%)

Check what this prior implies for probabilities:

```
plogis(qnorm(c(0.025, 0.5, 0.975), mean = 0, sd = 1.5))
# [1] 0.050 0.500 0.950  # i.e., 5% to 95% accuracy range
```

- Covers reasonable range for grammaticality judgments

1.4.4.2 `normal(0, 1)` for effects (b):

- Mean = 0 (no direction bias)
- SD = 1 on log-odds scale
- 95% prior interval: 0 ± 2*1 = [-2, 2] on log-odds scale
- 95% prior mass: effects change probability by ~12-88% range
- Example: if baseline is 50%, effect of +1 moves it to 73% (plogis(1) = 0.73)
- Regularizes implausibly large effects (e.g., moving from 10% to 99%)

1.4.5 Quick reference: Log-odds to probability

```
# Convert log-odds to probability
plogis(0) = 0.50   # log-odds 0 = 50%
plogis(1) = 0.73   # log-odds 1 = 73%
plogis(2) = 0.88   # log-odds 2 = 88%
plogis(-1) = 0.27  # log-odds -1 = 27%
```