

Introduction to Bayesian linear regression with brms

Stefano Coretta

18/01/2020

Road map

1. Basic concepts.

Road map

1. Basic concepts.
2. Choosing priors.

Road map

1. Basic concepts.
2. Choosing priors.
3. Model fitting with brms.

Road map

1. Basic concepts.
2. Choosing priors.
3. Model fitting with brms.
 - 3.1 Normal.

Road map

1. Basic concepts.
2. Choosing priors.
3. Model fitting with brms.
 - 3.1 Normal.
 - 3.2 Log-normal.

Road map

1. Basic concepts.
2. Choosing priors.
3. Model fitting with brms.
 - 3.1 Normal.
 - 3.2 Log-normal.
 - 3.3 Binomial (Bernoulli).

Road map

1. Basic concepts.
2. Choosing priors.
3. Model fitting with brms.
 - 3.1 Normal.
 - 3.2 Log-normal.
 - 3.3 Binomial (Bernoulli).
4. Bayesian inference.

BASIC CONCEPTS

Random variables

- We have a question about the world, so we collect data (sample from a population).

Random variables

- We have a question about the world, so we collect data (sample from a population).
 - $y = (y_1, y_2, y_3, y_4, \dots, y_n)$

Random variables

- We have a question about the world, so we collect data (sample from a population).
 - $y = (y_1, y_2, y_3, y_4, \dots, y_n)$
- We want to know how the data (the sample y) was generated.

Random variables

- We have a question about the world, so we collect data (sample from a population).
 - $y = (y_1, y_2, y_3, y_4, \dots, y_n)$
- We want to know how the data (the sample y) was generated.
- In probability theory, data is generated by a random variable Y .

- Y is a variable whose value is generated by a random event.

Random variables

- Y is a variable whose value is generated by a random event.
- Y is uncertain.

Random variables

- Y is a variable whose value is generated by a random event.
- Y is uncertain.
 - We can describe Y as a probability distribution.

Random variables

- Probability distribution.

Random variables

- Probability distribution.
 - A list of the values a random variable could take on along with their corresponding probability.

Random variables

- Probability distribution.
 - A list of the values a random variable could take on along with their corresponding probability.
- Probability distributions can be expressed by a set of parameters $\Theta = (\theta_1, \dots, \theta_n)$.

Random variables

- Probability distribution.
 - A list of the values a random variable could take on along with their corresponding probability.
- Probability distributions can be expressed by a set of parameters $\Theta = (\theta_1, \dots, \theta_n)$.
- Some probability distributions:

Random variables

- Probability distribution.
 - A list of the values a random variable could take on along with their corresponding probability.
- Probability distributions can be expressed by a set of parameters $\Theta = (\theta_1, \dots, \theta_n)$.
- Some probability distributions:
 - *Normal*(μ, σ),

Random variables

- Probability distribution.
 - A list of the values a random variable could take on along with their corresponding probability.
- Probability distributions can be expressed by a set of parameters $\Theta = (\theta_1, \dots, \theta_n)$.
- Some probability distributions:
 - *Normal*(μ, σ),
 - *Binomial*(n, p),

Random variables

- Probability distribution.
 - A list of the values a random variable could take on along with their corresponding probability.
- Probability distributions can be expressed by a set of parameters $\Theta = (\theta_1, \dots, \theta_n)$.
- Some probability distributions:
 - *Normal*(μ, σ),
 - *Binomial*(n, p),
 - *Poisson*(λ)

$$y_i \sim \textit{Normal}(\mu, \sigma)$$

Frequentist vs Bayesian view

- Parameters: $\mu, \sigma, p, \lambda, \dots$

Frequentist vs Bayesian view

- Parameters: $\mu, \sigma, p, \lambda, \dots$
- Frequentist view:

Frequentist vs Bayesian view

- Parameters: $\mu, \sigma, p, \lambda, \dots$
- Frequentist view:
 - The parameters are **fixed** (they are unknown but certain).

Frequentist vs Bayesian view

- Parameters: $\mu, \sigma, p, \lambda, \dots$
- Frequentist view:
 - The parameters are **fixed** (they are unknown but certain).
 - They take on a specific value.

Frequentist vs Bayesian view

- Parameters: $\mu, \sigma, p, \lambda, \dots$
- Frequentist view:
 - The parameters are **fixed** (they are unknown but certain).
 - They take on a specific value.
- Bayesian view:

Frequentist vs Bayesian view

- Parameters: $\mu, \sigma, p, \lambda, \dots$
- Frequentist view:
 - The parameters are **fixed** (they are unknown but certain).
 - They take on a specific value.
- Bayesian view:
 - The parameters are **random variables** (they are unknown and uncertain).

Frequentist vs Bayesian view

- Parameters: $\mu, \sigma, p, \lambda, \dots$
- Frequentist view:
 - The parameters are **fixed** (they are unknown but certain).
 - They take on a specific value.
- Bayesian view:
 - The parameters are **random variables** (they are unknown and uncertain).
 - We describe each parameter as a probability distribution, expressed by a set of **hyperparameters**.

- We want to estimate the parameters μ and σ from the data.

Priors

- We want to estimate the parameters μ and σ from the data.
- We can incorporate previous knowledge (belief) about the parameters using **priors** (*prior probability distributions*).

Priors

- We want to estimate the parameters μ and σ from the data.
- We can incorporate previous knowledge (belief) about the parameters using **priors** (*prior probability distributions*).
 - We specify the hyperparameters of the prior probability distributions.

Priors

- We want to estimate the parameters μ and σ from the data.
- We can incorporate previous knowledge (belief) about the parameters using **priors** (*prior probability distributions*).
 - We specify the hyperparameters of the prior probability distributions.
- Priors are chosen based on expert knowledge, previous studies, pilot data...

Priors

- We want to estimate the parameters μ and σ from the data.
- We can incorporate previous knowledge (belief) about the parameters using **priors** (*prior probability distributions*).
 - We specify the hyperparameters of the prior probability distributions.
- Priors are chosen based on expert knowledge, previous studies, pilot data...
 - Priors must **not** be chosen based on inspection of the data to be analysed.

$$\textit{observed data} \times \textit{prior belief} = \textit{posterior belief}$$

Bayesian belief update

<https://nanx.shinyapps.io/conjugate-normal-umkv/>

CHOOSING PRIORS

- Toy example with F1.

- Toy example with F1.
- Data:

- Toy example with F1.
- Data:
 - Italian.

- Toy example with F1.
- Data:
 - Italian.
 - 7 speakers.

- Toy example with F1.
- Data:
 - Italian.
 - 7 speakers.
 - CVCV words in frame sentence.

- Toy example with F1.
- Data:
 - Italian.
 - 7 speakers.
 - CVCV words in frame sentence.
 - C1 = /p/.

- Toy example with F1.
- Data:
 - Italian.
 - 7 speakers.
 - CVCV words in frame sentence.
 - C1 = /p/.
 - V1 = V2 = /a, o, u/.

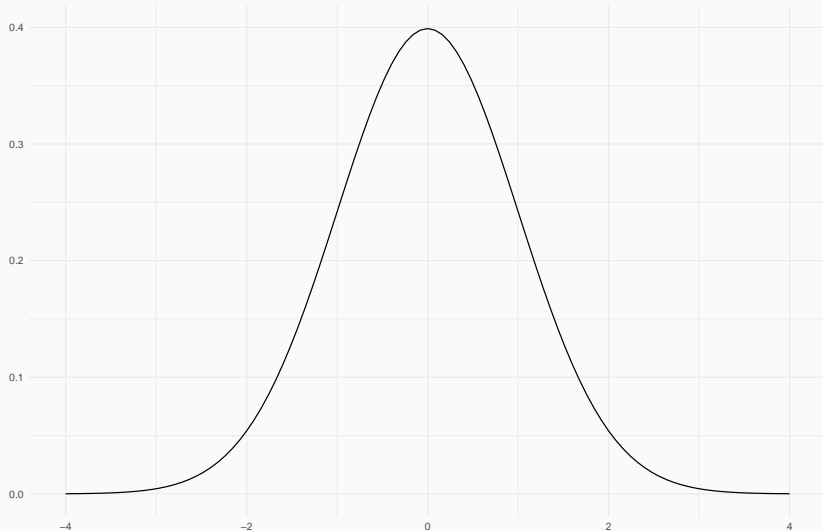
- Toy example with F1.
- Data:
 - Italian.
 - 7 speakers.
 - CVCV words in frame sentence.
 - C1 = /p/.
 - V1 = V2 = /a, o, u/.
 - C2 = /t, d, k, g/.

Formant values are roughly distributed according to a normal (Gaussian) distribution.

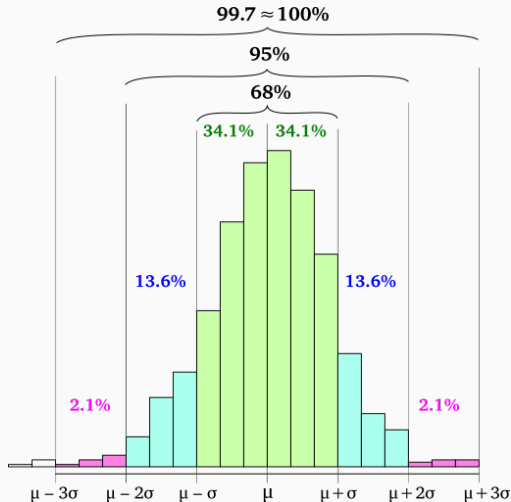
$$F1_i \sim \text{Normal}(\mu, \sigma)$$

Normal (Gaussian) distribution

Standard normal (Gaussian) distribution
mean = 0, SD = 1



Normal (Gaussian) distribution



Melikamp, https://commons.wikimedia.org/wiki/File:Empirical_rule_histogram.svg (CC BY-SA 4.0)

- *Normal*(μ, σ) has two parameters, μ and σ .

- $Normal(\mu, \sigma)$ has two parameters, μ and σ .
- μ and σ are random variables (unkown and uncertain).

- $Normal(\mu, \sigma)$ has two parameters, μ and σ .
- μ and σ are random variables (unknown and uncertain).
- We express these parameters as priors (probability distributions with hyperparameters).

$$F1_i \sim \text{Normal}(\mu, \sigma)$$

What prior for μ ?

$$F1_i \sim \text{Normal}(\mu, \sigma)$$

$$\mu \sim \text{Normal}(\mu_1, \sigma_1)$$

Prior for the mean

Normal(0, 500) as the prior for μ .

- This means that we believe μ to be between -1000 and +1000 at 95% confidence.

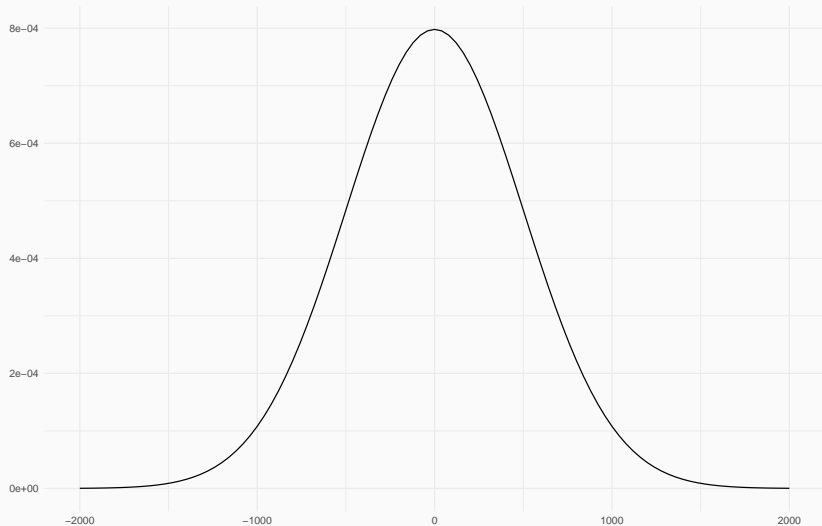
Normal(0, 500) as the prior for μ .

- This means that we believe μ to be between -1000 and +1000 at 95% confidence.
 - 95% credible interval (CI) = $[\mu_1 - 2\sigma_1, \mu_1 + 2\sigma_1]$.

Prior for the mean

Prior for μ

$\mu = 0, \sigma = 500$



Type of priors

- Informative and weakly informative priors.

Type of priors

- Informative and weakly informative priors.
- Uninformative or diffuse priors.

Type of priors

- Informative and weakly informative priors.
- Uninformative or diffuse priors.
 - Uniform distribution.

Type of priors

- Informative and weakly informative priors.
- Uninformative or diffuse priors.
 - Uniform distribution.
- Regularising priors.

$$F1_i \sim \text{Normal}(\mu, \sigma)$$

$$\mu \sim \text{Normal}(0, 500)$$

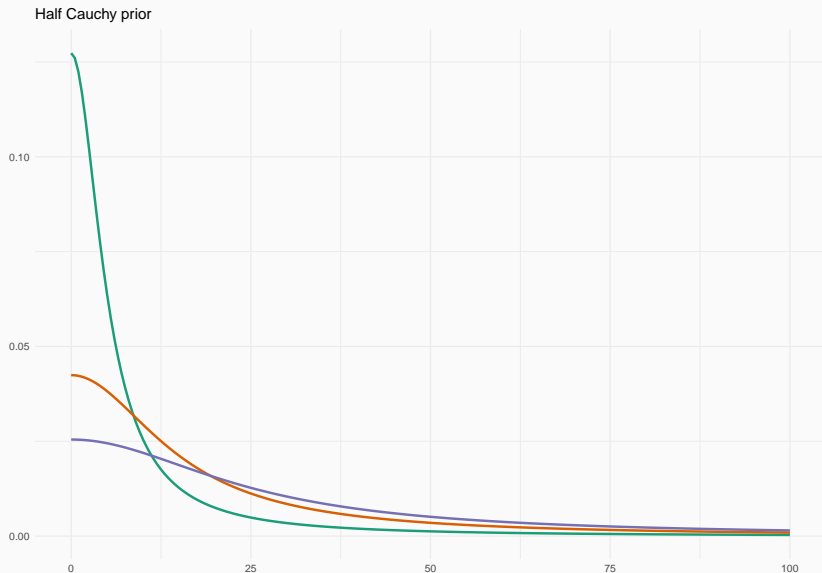
What about σ ?

$$Fl_i \sim Normal(\mu, \sigma)$$

$$\mu \sim Normal(0, 500)$$

$$\sigma \sim HalfCauchy(x_0, \gamma)$$

Prior for the standard deviation



Prior for the standard deviation

```
library(HDInterval)
inverseCDF(c(0.025, 0.975), phcauchy, 5)

## [1] 0.1964676 127.2584987

inverseCDF(c(0.025, 0.975), phcauchy, 15)

## [1] 0.5893621 381.7754937

inverseCDF(c(0.025, 0.975), phcauchy, 25)

## [1] 0.9822792 636.2924897
```

$$F1_i \sim \text{Normal}(\mu, \sigma)$$

$$\mu \sim \text{Normal}(0, 500)$$

$$\sigma \sim \text{HalfCauchy}(0, 25)$$

MODEL FITTING

- We have a model which incorporates our (vague) knowledge about F1 (through the priors for μ and σ).

- We have a model which incorporates our (vague) knowledge about F1 (through the priors for μ and σ).
- Now we want to obtain the **posterior distributions** of μ and σ .

- We have a model which incorporates our (vague) knowledge about F1 (through the priors for μ and σ).
- Now we want to obtain the **posterior distributions** of μ and σ .
 - The posterior distribution is the prior distribution *conditioned* on the data.

- We have a model which incorporates our (vague) knowledge about F1 (through the priors for μ and σ).
- Now we want to obtain the **posterior distributions** of μ and σ .
 - The posterior distribution is the prior distribution *conditioned* on the data.
- **brms** R package: Bayesian Regression Models using Stan (Bürkner, 2018).

Installation

Safe method:

- **Install Rstan first:** <https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started> (see *Installation of Rstan, Checking the C++ Toolchain, and Configuration of the C++ Toolchain*).

Installation

Safe method:

- **Install Rstan first:** <https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started> (see *Installation of Rstan*, *Checking the C++ Toolchain*, and *Configuration of the C++ Toolchain*).
 - Note that details in *Checking the C++ Toolchain* differ depending on your OS.

Installation

Safe method:

- **Install Rstan first:** <https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started> (see *Installation of Rstan*, *Checking the C++ Toolchain*, and *Configuration of the C++ Toolchain*).
 - Note that details in *Checking the C++ Toolchain* differ depending on your OS.
- **Install brms:** <https://github.com/paul-buerkner/brms#how-do-i-install-brms>.

- Stan (Stan Development Team, 2017).

- Stan (Stan Development Team, 2017).
 - Statistical programming language written in C++.

- Stan (Stan Development Team, 2017).
 - Statistical programming language written in C++.
 - Fit Bayesian models (calculate posterior distributions).

- Stan (Stan Development Team, 2017).
 - Statistical programming language written in C++.
 - Fit Bayesian models (calculate posterior distributions).
- Calculation can be complex and/or impossible, so we take many samples from the data and from the possible parameter values to find the posterior distributions of the hyperparameters.

- Stan (Stan Development Team, 2017).
 - Statistical programming language written in C++.
 - Fit Bayesian models (calculate posterior distributions).
- Calculation can be complex and/or impossible, so we take many samples from the data and from the possible parameter values to find the posterior distributions of the hyperparameters.
 - Markov Chain Monte Carlo (MCMC) sampling using the No-U-Turn sampler (NUTS).

- brms translates R code into Stan code.

- brms translates R code into Stan code.
- Stan code is run in Stan via Rstan, an R interface to Stan.

- brms translates R code into Stan code.
- Stan code is run in Stan via Rstan, an R interface to Stan.
- **brm()** function from brms.

- brms translates R code into Stan code.
- Stan code is run in Stan via Rstan, an R interface to Stan.
- **brm()** function from brms.
 - lme4 syntax ($y \sim x + (1|w)$).

- brms translates R code into Stan code.
- Stan code is run in Stan via Rstan, an R interface to Stan.
- **brm()** function from brms.
 - lme4 syntax ($y \sim x + (1|w)$).
 - Creates a Stan model, which is compiled (in C++) and run.

```
library(brms)

f1 <- brm(
  <model_formula>,
  <family>,
  <prior>,
  <data>,
  chains = 4,
  iter = 2000
)
```

```
library(brms)

f1 <- brm(
  f1 ~ 1,
  family = gaussian(),
  <prior>,
  data = f_end,
  chains = 4,
  iter = 2000
)
```

Get prior

```
get_prior(  
  f1 ~ 1,  
  family = gaussian(),  
  data = f_end  
)
```

```
##               prior      class coef group resp dpar nlpar bound  
## 1 student_t(3, 495, 167) Intercept  
## 2  student_t(3, 0, 167)      sigma
```


Prior predictive checks

```
nsim <- 1000
nobs <- 1000

y <- matrix(rep(NA, nsim * nobs), ncol = nobs)

mu <- rnorm(nsim, 0, 500)
sigma <- rhcauchy(nsim, 25)

for (i in 1:nsim) {
  y[i,] <- rnorm(nobs, mean = mu[i], sd = sigma[i])
}
```

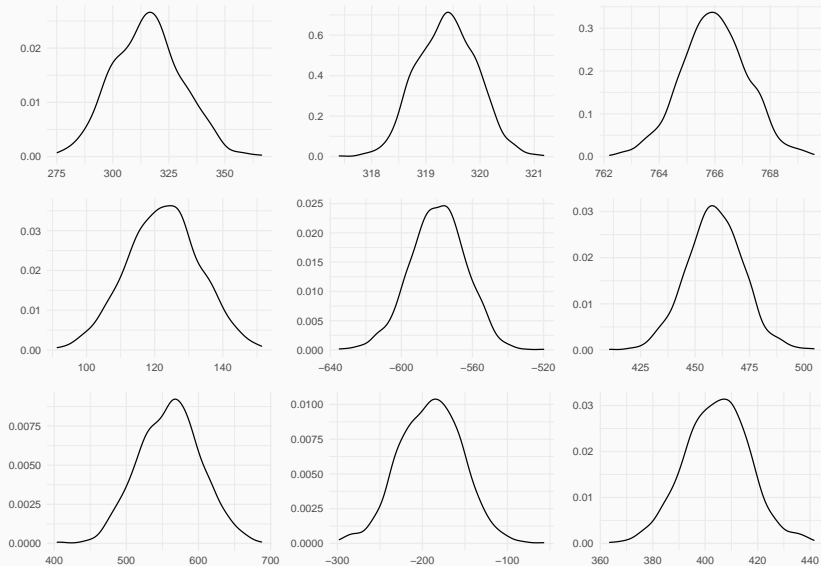
Prior predictive checks

```
rand_sample <- sample(1:nsim, 9, replace = FALSE)
plots <- list()

j = 1

for (i in rand_sample) {
  my_data <- enframe(y[i,], name = NULL)
  plots[[j]] <- ggplot(data = my_data) +
    aes(x = value) +
    geom_density() +
    labs(x = element_blank(), y = element_blank())
  j = j + 1
}
```

Prior predictive checks



Set prior

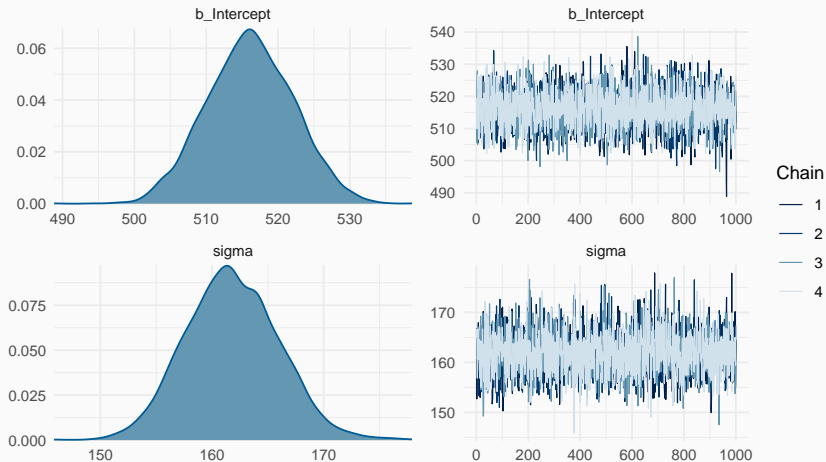
```
priors <- c(  
  prior(normal(0, 500), class = Intercept),  
  prior(cauchy(0, 25), class = sigma)  
)
```

Run the model

```
f1_bm <- brm(  
  f1 ~ 1,  
  family = gaussian(),  
  prior = priors,  
  data = f_end,  
  chains = 1,  
  iter = 2000,  
  file = "./cache/f1"  
)
```

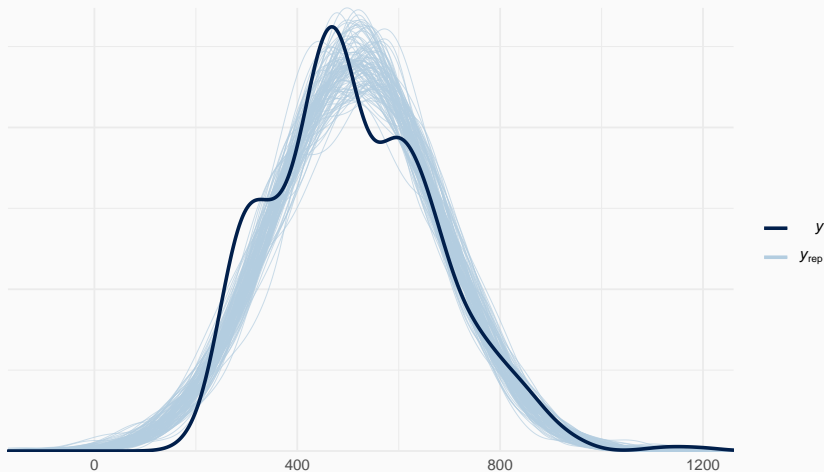
Plot model

```
plot(f1_bm, ask = FALSE)
```



Posterior predictive check

```
pp_check(f1_bm, nsamples = 100)
```



Model summary

f1_bm

```
## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: f1 ~ 1
## Data: f_end (Number of observations: 748)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##           total post-warmup samples = 4000
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept    516.14      6.01   504.35   527.81 1.00     3358     2498
##
## Family Specific Parameters:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma    161.81      4.18   153.95   170.03 1.00     3071     2682
##
## 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).
```


$$F1_i \sim \text{Normal}(\mu, \sigma)$$

$$\mu \sim \text{Normal}(0, 500)$$

$$\sigma \sim \text{HalfCauchy}(0, 25)$$

$$F1_i \sim \text{Normal}(\mu, \sigma)$$

$$\mu \sim \text{Normal}(0, 500)$$

$$\sigma \sim \text{HalfCauchy}(0, 25)$$

Let's add the predictor vowel.

Adding predictors

$$F1_i \sim \text{Normal}(\mu_i, \sigma)$$

$$\mu_i = \alpha + \beta_1 \times O_i + \beta_2 \times U_i$$

$$\alpha \sim \text{Normal}(\mu_1, \sigma_1) \text{ [vowel = /a/]}$$

$$\beta_1 \sim \text{Normal}(\mu_2, \sigma_2) \text{ [vowel = /o/]}$$

$$\beta_2 \sim \text{Normal}(\mu_3, \sigma_3) \text{ [vowel = /u/]}$$

$$\sigma \sim \text{HalfCauchy}(x_0, \gamma)$$

Adding predictors

$$F1_i \sim \text{Normal}(\mu_i, \sigma)$$

$$\mu_i = \alpha + \beta_1 \times O_i + \beta_2 \times U_i$$

$$\alpha \sim \text{Normal}(0, 500)$$

$$\beta_1 \sim \text{Normal}(0, 750)$$

$$\beta_2 \sim \text{Normal}(0, 750)$$

$$\sigma \sim \text{HalfCauchy}(0, 25)$$

Adding predictors

```
get_prior(  
  f1 ~ 1 + vowel,  
  family = gaussian(),  
  data = f_end  
)
```

```
##               prior      class  coef group resp dpar nlpar bound  
## 1                      b  
## 2                      b vowelo  
## 3                      b vowelu  
## 4 student_t(3, 495, 167) Intercept  
## 5  student_t(3, 0, 167)    sigma
```

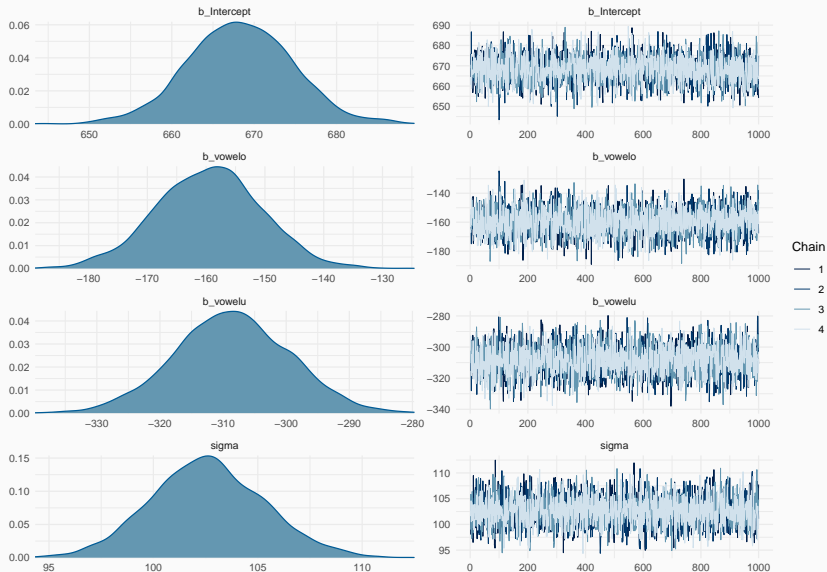
Adding predictors

```
priors <- c(  
  prior(normal(0, 500), class = Intercept),  
  prior(cauchy(0, 25), class = sigma),  
  prior(normal(0, 750), class = b, coef = "vowel_o"),  
  prior(normal(0, 750), class = b, coef = "vowel_u")  
)
```

Adding predictors

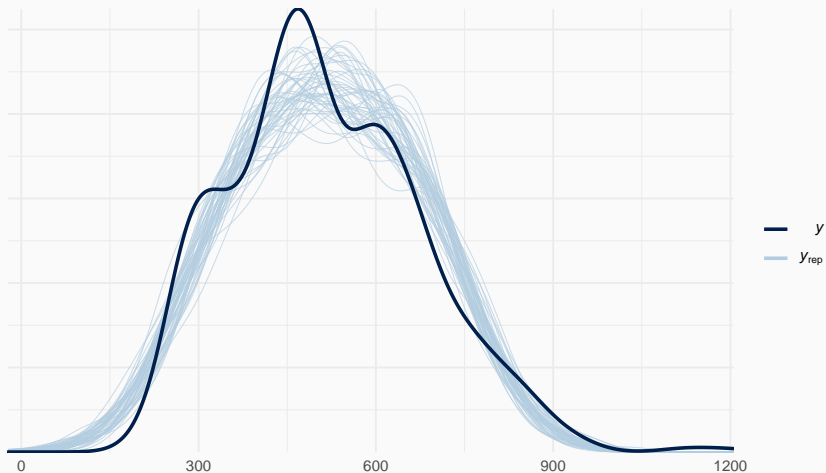
```
f1_2 <- brm(  
  f1 ~ 1 + vowel,  
  family = gaussian(),  
  prior = priors,  
  data = f_end,  
  chains = 4,  
  iter = 2000,  
  file = "./cache/f1_2"  
)
```

Adding predictors



Adding predictors

```
pp_check(f1_2, nsamples = 50)
```

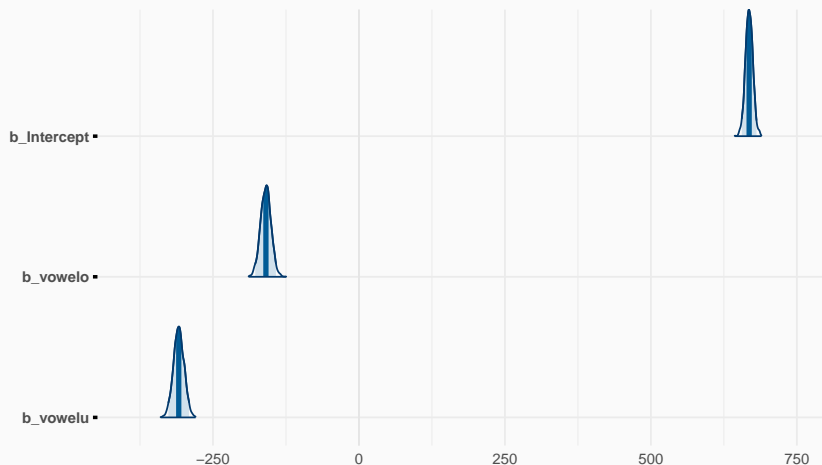


Adding predictors

```
## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: f1 ~ 1 + vowel
## Data: f_end (Number of observations: 748)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept    668.30      6.38   655.69   680.85 1.00     3249     2620
## vowel0       -159.54      8.94  -177.58  -142.24 1.00     3355     2903
## vowel1       -308.57      9.06  -326.33  -291.04 1.00     3281     2734
##
## Family Specific Parameters:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma    102.60      2.70    97.38   108.10 1.00     3470     2818
##
## 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).
```

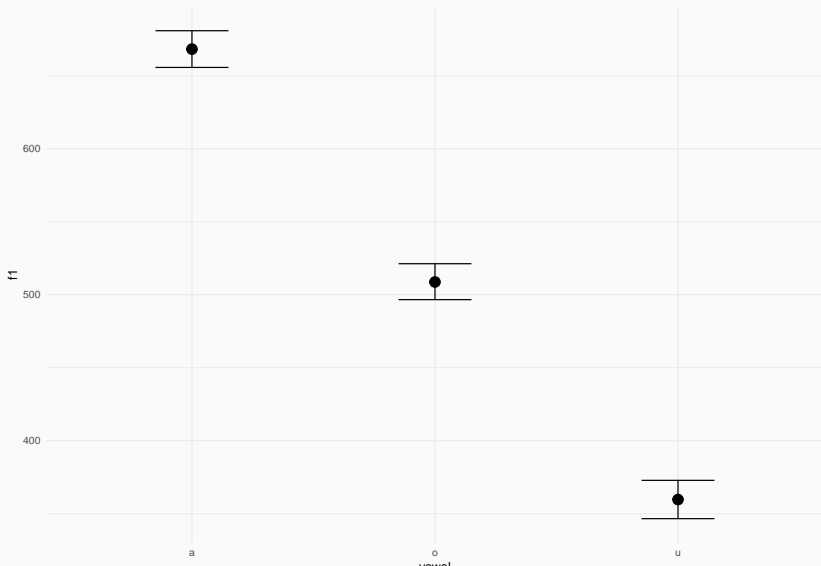
Adding predictors

```
mcmc_areas(f1_2, regex_pars = "b_", prob = 0.95)
```



Adding predictors

```
conditional_effects(f1_2)
```



Random effects

$$F1_i \sim \text{Normal}(\mu_i, \sigma)$$

$$\mu_i = \alpha + \alpha_{\text{speaker}[i]} + (\beta_1 + \beta_{1\text{speaker}[i]}) \times O_i + (\beta_2 + \beta_{2\text{speaker}[i]}) \times U_i$$

$$\begin{bmatrix} \alpha_{\text{speaker}[i]} \\ \beta_{1\text{speaker}[i]} \\ \beta_{2\text{speaker}[i]} \end{bmatrix} \sim \text{MVNormal}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, S\right)$$

Random effects

$$\alpha \sim \text{Normal}(0, 500)$$

$$\alpha_{\text{speaker}} \sim \text{Normal}(0, \sigma_{\text{speaker}})$$

$$\beta_1 \sim \text{Normal}(0, 750)$$

$$\beta_2 \sim \text{Normal}(0, 750)$$

$$\sigma_{\alpha_{\text{speaker}}} \sim \text{Normal}(0, \sigma_{\text{speaker}})$$

$$\sigma_{\beta_1_{\text{speaker}}} \sim \text{HalfCauchy}(0, 15)$$

$$\sigma_{\beta_2_{\text{speaker}}} \sim \text{HalfCauchy}(0, 15)$$

$$\sigma_{\text{speaker}} \sim \text{HalfCauchy}(0, 15)$$

$$\sigma \sim \text{HalfCauchy}(0, 15)$$

Random effects

```
get_prior(  
  f1 ~ 1 + vowel + (1 + vowel | speaker),  
  family = gaussian(),  
  data = f_end  
)
```

```
##           prior      class      coef  group resp dpar nlpar bound  
## 1                b  
## 2                b  vowel0  
## 3                b  vowel1  
## 4      lkj(1)      cor  
## 5                cor      speaker  
## 6 student_t(3, 495, 167) Intercept  
## 7   student_t(3, 0, 167)      sd  
## 8                sd      speaker  
## 9                sd Intercept speaker  
## 10               sd  vowel0 speaker  
## 11               sd  vowel1 speaker  
## 12 student_t(3, 0, 167)  sigma
```

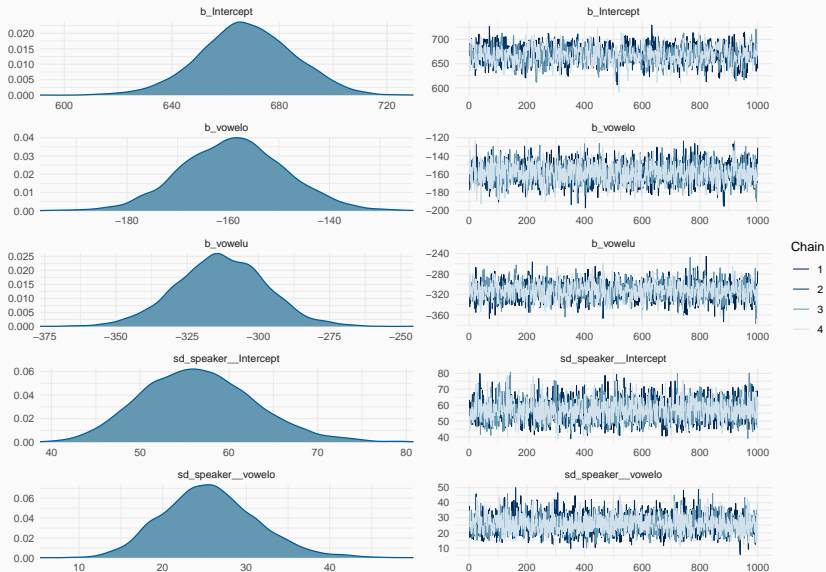
Random effects

```
priors <- c(  
  prior(normal(0, 500), class = Intercept),  
  prior(cauchy(0, 15), class = sigma),  
  prior(normal(0, 750), class = b, coef = "vowel_o"),  
  prior(normal(0, 750), class = b, coef = "vowel_u"),  
  prior(normal(0, 15), class = sd),  
  prior(lkj(2), class = cor)  
)
```

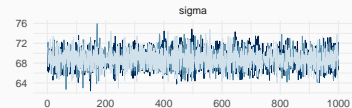
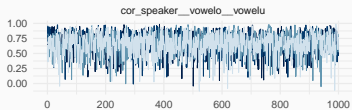
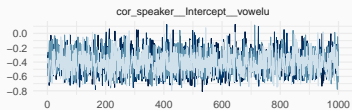
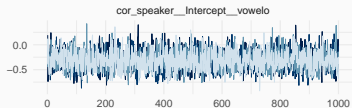
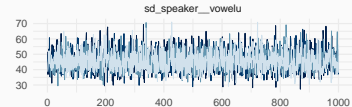
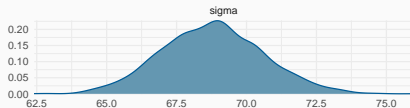
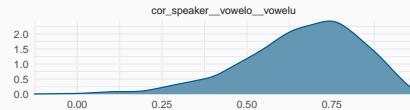
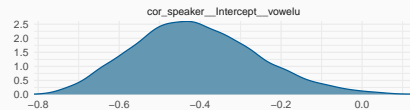
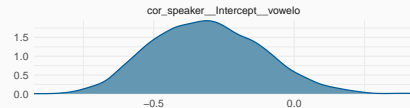
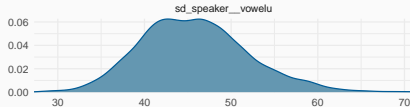

Random effects

```
f1_3 <- brm(  
  f1 ~ 1 + vowel + (1 + vowel | speaker),  
  family = gaussian(),  
  prior = priors,  
  data = f_end,  
  chains = 4,  
  iter = 2000,  
  file = "./cache/f1_3"  
)
```

Random effects



Random effects



Chain

- 1
- 2
- 3
- 4

Random effects

```
## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: f1 ~ 1 + vowel + (1 + vowel | speaker)
## Data: f_end (Number of observations: 748)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##           total post-warmup samples = 4000
##
## Group-Level Effects:
## ~speaker (Number of levels: 11)
##
```

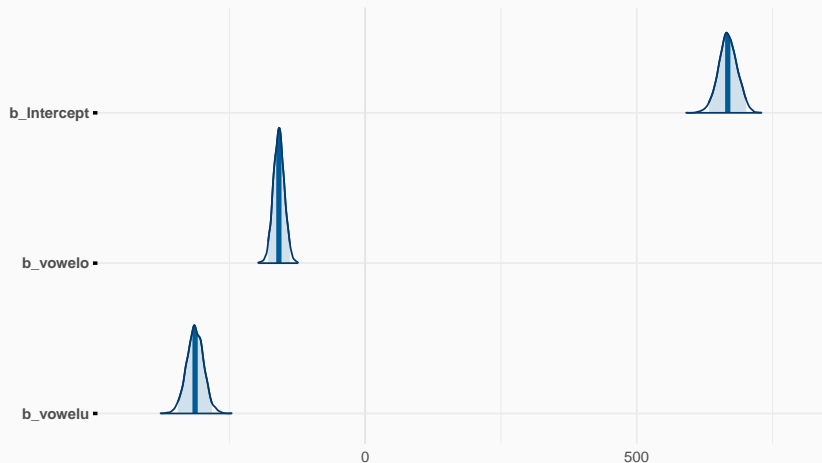
	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS
## sd(Intercept)	56.58	6.35	45.13	69.91	1.00	2264
## sd(vowelo)	25.69	5.68	15.55	37.80	1.00	2533
## sd(vowelu)	45.76	6.19	34.82	59.09	1.00	3639
## cor(Intercept,vowelo)	-0.31	0.19	-0.66	0.08	1.00	3262
## cor(Intercept,vowelu)	-0.41	0.15	-0.68	-0.08	1.00	3024
## cor(vowelo,vowelu)	0.66	0.17	0.26	0.93	1.00	2047

Random effects

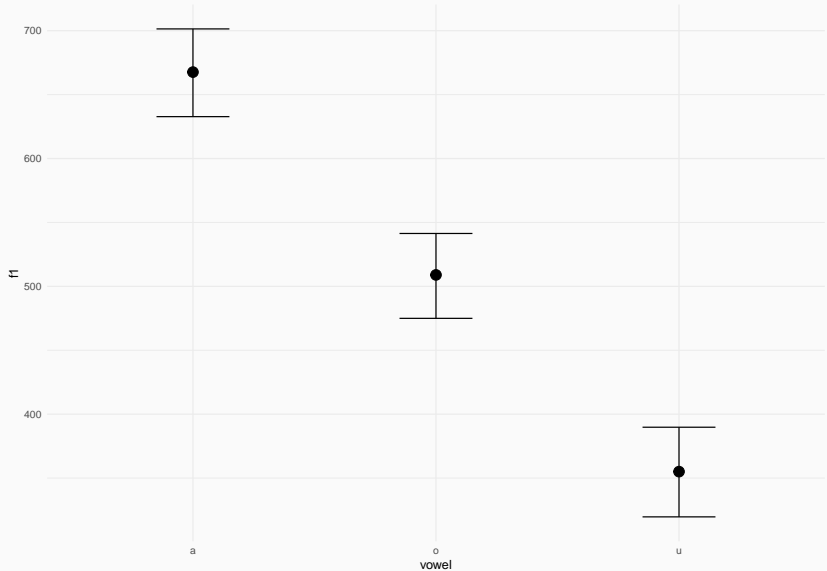
```
## sd(Intercept)          2910
## sd(vowelo)             2159
## sd(vowelu)             3039
## cor(Intercept,vowelo)  3191
## cor(Intercept,vowelu)  3185
## cor(vowelo,vowelu)     2510
##
## Population-Level Effects:
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept    667.84    17.44   632.77   701.41 1.00    1033    1386
## vowelo      -158.88    10.29  -178.49  -138.61 1.00    2297    2295
## vowelu       -313.03    15.87  -344.81  -282.04 1.00    1848    2364
```

Random effects

```
mcmc_areas(f1_3, regex_pars = "b_", prob = 0.95)
```



Random effects



References

- Bürkner, Paul-Christian. 2018. Advanced Bayesian multilevel modeling with the R package brms. *The R Journal* 10(1). 395–411.
- Stan Development Team. 2017. Stan: A C++ library for probability and sampling, version 2.14.0. <http://mc-stan.org/>.