BRMS cheat sheet

Disclaimer: This document is not exhaustive and is intended to assist in selecting suitable priors and distributions for Bayesian analysis using brms (part 1), as well as guiding through a basic data analysis pipeline (part 2). It should be understood as a foundational resource to support your analysis rather than a comprehensive guide.

For those new to data analysis in R, I recommend consulting the freely available book "R for Data Science," accessible at: https://r4ds.hadley.nz.

Part 1: About brms

A. Understanding the Brms Formula:

Brms use the same syntax as other R packages such as lm or lme4, let's break through the classic formula using a fitted example from a daily week. This example examines the productivity (in %) of any given day given the quality of the coffee drunk during that same day.

Let's imagine we are studying the impact of coffee quality on our daily productivity (measured in %). We can use brms to model this relationship using the following formula:

Model: Productivity ~ 1 + coffee + (1 + coffee | researcher)

1. Dependent Variable: Productivity

This is the variable we're trying to predict, represented by the tilde (~).

2. Fixed Effects: 1+ coffee

This term captures the average effect of coffee quality on productivity. A positive coefficient here would suggest that better coffee generally leads to higher productivity.

Note: The intercept is implicit meaning that writing 1 + coffee or coffee equals the same. If you want to fit without an intercept you should write 0 + coffee.

3. Random Effects: (1 + coffee | researcher)

This part accounts for potential variations due to researchers:

3.1. 1 | researcher - This models a random intercept for each researcher. This allows for baseline productivity differences between researchers, independent of coffee quality.

3.2. coffee | researcher - This models a random slope for coffee quality within each researcher. This allows the effect of coffee to vary across researchers. Some researchers might be more sensitive to coffee quality than others in terms of their productivity boost.

B. What are Priors?

In Bayesian statistics, priors represent your initial beliefs about the possible values of model parameters before observing any data. They act as a regularization technique, preventing

extreme estimates and influencing the posterior distribution (distribution of parameters after considering the data).

Specifying Priors in brms:

Brms doesn't provide a single function for all priors. Instead, it offers various built-in functions specific to different prior distributions and parameter types. Here's a categorization of some commonly used priors:

1. Continuous Distributions:

- **Normal:** normal(location, scale) - Represents a normal distribution with specified mean (location) and standard deviation (scale). This is a common choice for most continuous parameters when you have some prior belief about their central tendency and spread.
    - **Half-Normal**: half_normal (0, scale) – Represents the absolute value transformation of a standard normal distribution. Unlike the normal distribution, the location of the half normal is always fixed at 0. This is often used for variance parameters as it reflects the variability in datasets without assuming a specific direction.
- **Student's t:** student_t(nu, location, scale) - Defines a t-distribution with degrees of freedom (nu) controlling the tail weight. Useful when you expect some heavier tails compared to a normal distribution, allowing for more extreme values.
- **Cauchy:** cauchy(location, scale) - Represents a Cauchy distribution with a center (location) and scale parameter. This prior allows for even heavier tails than the t-distribution, suitable for scenarios where very large or small effects are plausible.
- **Uniform:** uniform(lower, upper) - Sets a uniform distribution between a specified lower and upper bound. This is a non-informative prior, assigning equal probability to all values within the range. Use it cautiously as it doesn't convey any prior knowledge.
- **Gamma:** gamma(shape, rate) - This distribution is useful for modeling positive continuous variables that are skewed to the right (larger values are more likely). The shape parameter controls the skewness, and the rate parameter determines the scale.
- **Beta:** beta(alpha, beta) - This distribution is ideal for modeling proportions or probabilities between 0 and 1. The alpha and beta parameters control the shape of the distribution, influencing how heavily the probability mass is concentrated near 0, 1, or the center.
- **Exponential:** exponential(rate) - This distribution is suitable for modeling waiting times or durations until an event occurs. The rate parameter determines the decay rate of the probability density function. Higher rates indicate events happening more frequently.

2. Discrete Distributions:

- **Categorical:** categorical() or cat() - Specifies a prior distribution for categorical variables. This allows you to incorporate your beliefs about the relative probabilities of different categories.
- **Bernoulli:** bernoulli(p) - Defines a Bernoulli distribution for binary variables (success/failure). This prior lets you express your initial belief about the probability of success.
- **Poisson:** poisson(lambda) - Represents a Poisson distribution with a rate parameter (lambda) for count data. This prior reflects your prior knowledge about the expected number of events occurring in a fixed interval.
- **Binomial**: binomial(size, p) - This is a prior for binary data where each observation represents a fixed number of trials (size) with a probability of success (p) for each trial. This could be useful if you were measuring the number of successful completions out of a fixed number of tasks performed at different coffee quality levels.
- **Ordered categorical**: ordered() - This is a prior for categorical variables with a natural order between the categories. It allows you to model the relationship between the outcome and the ordered categories. This could be relevant if you have researcher sensitivity to coffee quality categorized as low, medium, and high, and you believe there's a progressive effect (e.g., low sensitivity shows the least impact, medium shows more, and high shows the strongest impact).
- **Negative binomial**: negative.binomial(size, mu) - This is a prior for count data similar to the Poisson distribution but allowing for more overdispersion (greater variance than expected under Poisson). This could be useful if you believe the variability in productivity across days might be higher than what a Poisson distribution could capture.
- **Zero-inflated distributions**: brms offers priors for zero-inflated distributions, where there's a higher probability of observing zero counts compared to a standard count distribution. These include:
    - **Zero-inflated Poisson (ZIP):** zeroinff(poisson(lambda)) - This models a process with two components: one generating zero counts and another generating counts following a Poisson distribution.
    - **Zero-inflated negative binomial (ZINB):** zeroinff(negative.binomial(size, mu)) - Similar to ZIP but with negative binomial for count data.

3. Special Priors:

- **Linear Regression Coefficients:** b() or bf() - Specifies priors for coefficients in linear regression models, including the intercept. This allows you to incorporate your prior beliefs about the strength and direction of relationships between variables.

- **Intercept Prior:** prior_intercept() or pi() - Sets a prior distribution specifically for the intercept term. This can be useful if you have specific knowledge about the overall average effect in the model.
- **Slope Prior:** prior_slope() or ps() - Defines a prior distribution for slope coefficients. This allows you to focus your prior beliefs on the direction and strength of the relationships between specific predictor variables and the outcome.
- **LKJ Correlation Matrix:** lkj(nu) - Represents a Lewandowski, Kurowicki, and Joe (LKJ) correlation matrix prior for multivariate models. This is useful when you have multiple correlated outcomes and want to incorporate prior knowledge about the correlation structure.

C. What are Probability Distributions?

Probability distributions describe the likelihood of different outcomes or events occurring within a given set of possible outcomes. These distributions provide a structured way to understand and quantify uncertainty in various scenarios. Just as in Bayesian statistics where priors represent initial beliefs, in probability theory, distributions serve as a foundational framework for expressing uncertainty. They encapsulate our understanding of how likely different outcomes are, given certain conditions or assumptions.

Specify probabilities distributions in brms:

In brms, distributions play a fundamental role in specifying the likelihood function of the observed data and in modeling the uncertainty associated with the response variable. It allows to choose from a wide range of probability distributions to accommodate different types of data and modeling assumptions.

Probability distributions	Type	Shape	Suitable for Data	Parameters
Gaussian (normal)	Continuous	Bell-shaped	Continuous data with normal distribution	Mean (center), Standard deviation (spread)
Student (t)	Continuous	Bell-shaped with heavier tails than Gaussian	Continuous data with potential outliers	Degrees of freedom (nu), location (center), scale
Binomial	Discrete	0 or 1	Binary data (success/failure)	Probability of success (p)
Bernoulli	Discrete	0 or 1	Special case of binomial with fixed number of trials (n=1)	Probability of success (p)
Beta-Binomial	Discrete	Between 0 and 1	Proportion data where success probability can vary within groups	Probability of success (p), number of trials (n), shape parameters (alpha, beta)

Poisson	Discrete	Non-negative integers	Count data (events occurring in a fixed interval)	Rate (lambda)
Negative Binomial	Discrete	Non-negative integers	Count data with overdispersion (more variance than Poisson)	Rate (lambda), size parameter (phi)
Geometric	Discrete	0, 1, 2, ...	Count data representing number of trials before first success	Probability of success (p)
Gamma	Continuous	Positive real numbers	Continuous data with skewed distribution (positive values)	Shape (k), scale
Skew-Normal	Continuous	Bell-shaped with possible skew	Continuous data with potential asymmetry	Location (center), scale, skew
Lognormal	Continuous	Log-normally distributed data (positive values after taking the log)	Continuous data with skewed distribution (positive values)	Location (center), scale
Shifted Lognormal	Continuous	Positive real numbers after adding a constant	Similar to lognormal but allows for negative values after transformation	Location (center), scale, shift
Ex-Gaussian	Continuous	Asymmetric with long right tail	Data with long response times or delays	Location (center), scale, shape
Wiener (Normal Error)	Continuous	Continuous data with normal errors	Similar to Gaussian but focuses on error terms	Mean (center), standard deviation (spread)
Inverse Gaussian	Continuous	Distribution with two modes (peaks)	Data with positive values and potential peak at zero	Shape (lambda), scale
Survival Families			Survival analysis (time-to-event data)	
Weibull	Continuous		Time-to-event data with monotonic hazard	Shape (k), scale
Exponential	Continuous		Time-to-event data with constant hazard	Rate (lambda)
Cox Proportional Hazards (cox)	Not directly a distribution		Proportional hazards models in survival analysis	Baseline hazard function, coefficients for predictors

D. Supplementary material:

- https://michael-franke.github.io/Bayesian-Regression/ : very intuitive tutorial
- https://f0nzie.github.io/statistical_rethinking-rsuite/ : the best tutorial probably but also the longest
- https://rawgit.com/nicebread/BFDA/master/package/doc/BFDA_manual.html : how to compute power analyses using Bayesians stats.
- https://towardsdatascience.com/a-bayesian-approach-to-linear-mixed-models-lmm-in-r-python-b2f1378c3ac8: tutorial for fitting hierarchical models.
- http://singmann.org/wiener-model-analysis-with-brms-part-ii/ tutorial about model diagnostics

Part 2: Tutorial

In this tutorial we will try to answer a simple question: does the quality of the coffee available at the office affect the daily productivity of the researchers in the lab?

A. Creating data

To begin, let's generate some data.

```r
# Set the seed for reproducibility
set.seed(123)

# Define the researchers and days
researchers <- c('Helene','Mrittika','Marius','Valerie')
days <- c('Lundi','Mardi','Mercredi','Jeudi','Vendredi')

# Generate random productivity between 50% and 90%
productivity <- rnorm(length(researchers) * length(days), mean = 70, sd = 10)

# Simulate different coffee quality levels (1- low, 5 - high)
coffee_quality <- sample(1:5, length(researchers) * length(days), replace = TRUE)

# Introduce some researcher effect on baseline productivity
researcher_effect <- rnorm(length(researchers), mean = 0, sd = 2)
productivity <- productivity + rep(researcher_effect, each = length(days))

# Simulate the effect of coffee quality on productivity (stronger for some researchers)
researcher_sensitivity <- rnorm(length(researchers), mean = 0.5, sd = 0.2)
productivity <- productivity + coffee_quality * rep(researcher_sensitivity, each = length(days))

# Create a data frame with researcher and day names
data <- data.frame(researcher = rep(researchers, times = length(days)),
                   day = rep(days, length.out = length(researchers) * length(days)),
                   productivity = productivity,
                   coffee_quality = coffee_quality)

# Print a glimpse of the data
print(data)
```

Now, we have a data frame that contains 20 rows and 4 columns which depicts the productivity of each researchers given the day and the quality of the coffee during a normal week.

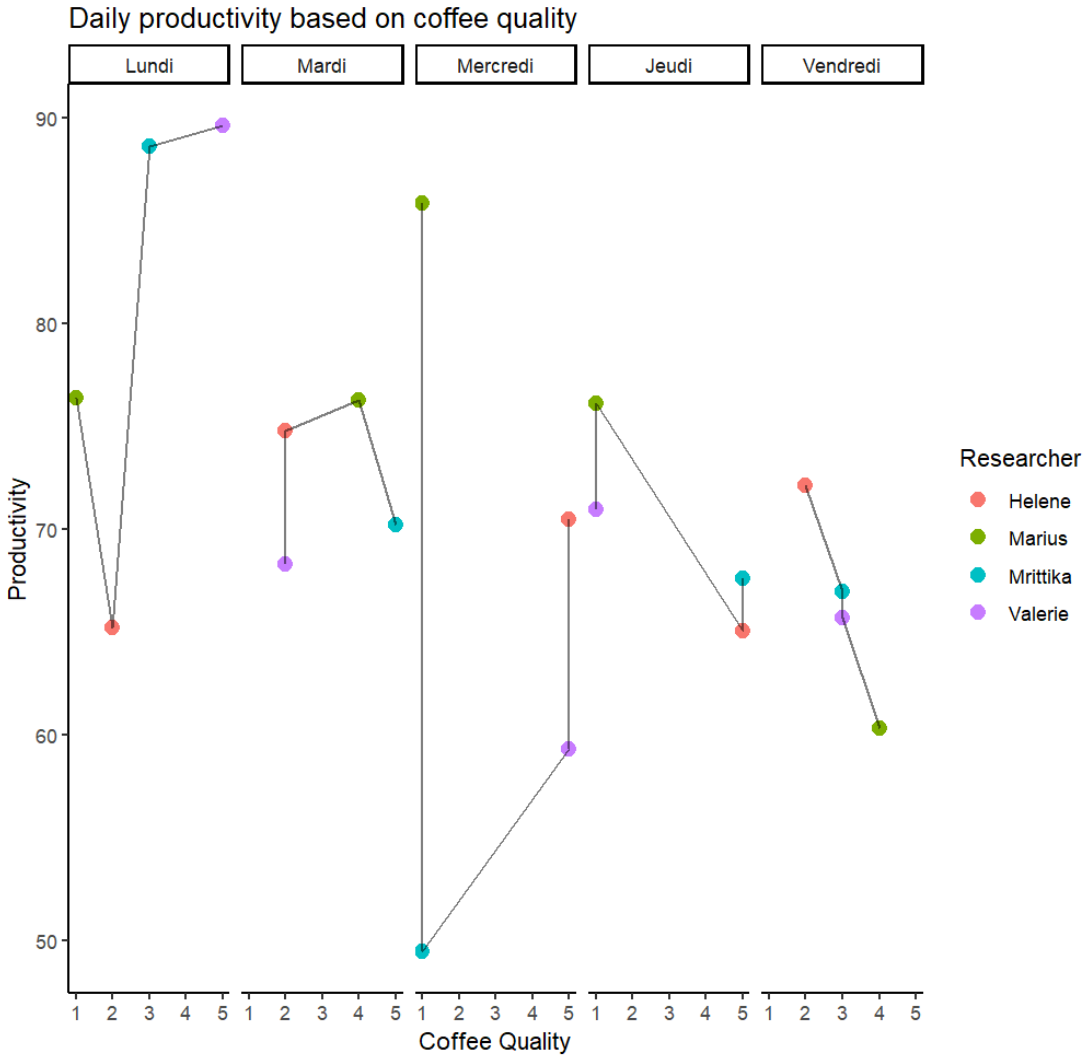
	researcher	day	productivity	coffee_quality
1	Helene	Lundi	65.20707	2
2	Mrittika	Mardi	70.18198	5
3	Marius	Mercredi	85.84160	1
4	Valerie	Jeudi	70.95960	1
5	Helene	Vendredi	72.10470	2
6	Mrittika	Lundi	88.57793	3
7	Marius	Mardi	76.29233	4
8	Valerie	Mercredi	59.28846	5
9	Helene	Jeudi	65.07054	5
10	Mrittika	Vendredi	66.97066	3
11	Marius	Lundi	76.37308	1
12	Valerie	Mardi	68.31731	2
13	Helene	Mercredi	70.48762	5
14	Mrittika	Jeudi	67.58673	5
15	Marius	Vendredi	60.33458	4
16	Valerie	Lundi	89.62572	5
17	Helene	Mardi	74.75499	2
18	Mrittika	Mercredi	49.45028	1
19	Marius	Jeudi	76.13001	1
20	Valerie	Vendredi	65.70861	3

We can also decide to summarize the data to check that every researcher has a mean productivity that isn't too different regardless of the coffee quality:

```
# A tibble: 4 × 2
  researcher mean_productivity
  <chr>                  <dbl>
1 Helene                  69.5
2 Marius                  75.0
3 Mrittika                68.6
4 Valerie                 70.8
```

We can use this information to set the prior of the intercept using the mean and SD of what we can see here.

If we want to visualize the full dataset, we could get something like this:



B. Write the model using brms.

```r
model_fit <- brm(productivity ~ 1 + coffee_quality + (1 + coffee_quality|researcher),
                 data = data,
                 prior = c(set_prior("normal(70,7)", class = "Intercept"),
                           set_prior("normal(0, 1)", class = "b"),
                           set_prior("normal(10.3,2)", class = "sigma")),
                 family = gaussian(),
                 chains = 4, cores = parallel::detectCores(),
                 warmup = 1000,iter = 2000)
```

For the prior, as we don't really know a lot about the effect of coffee on productivity, we use a weakly informative prior to avoid biasing the results towards specific values. We therefore say to the model that there is no effect of the coffee on productivity (mean of 0) but with some variations might still be expected (sd of 1). The intercept is based on the mean_productivity that we can get Another option is to use the get_priors() function so we can visualize what brms is actually using and then refine our own. Here is what it looks like:

```
> get_prior(productivity ~ 1 + coffee_quality + (1 + coffee_quality|researcher),
+           data = data)
                  prior     class           coef      group resp dpar nlpar lb ub       source
                 (flat)         b                                                      default
                 (flat)         b coffee_quality                                 (vectorized)
                 lkj(1)       cor                                                      default
                 lkj(1)       cor                researcher                    (vectorized)
 student_t(3, 70.3, 7.7) Intercept                                                      default
    student_t(3, 0, 7.7)        sd                                           0         default
    student_t(3, 0, 7.7)        sd                researcher                 0    (vectorized)
    student_t(3, 0, 7.7)        sd coffee_quality researcher                 0    (vectorized)
    student_t(3, 0, 7.7)        sd      Intercept researcher                 0    (vectorized)
    student_t(3, 0, 7.7)     sigma                                           0         default
```

Regarding the probability distribution, we opted for a gaussian at is it suitable for our continuous productivity data.

C. ~~Interpreting the results~~ Model diagnostics

Before going further, we should look at the summary of the model. The idea is first to check whether everything went right before interpreting anything.

These are two ways to have a look at the output. While summary shows everything, posterior summary shows only the estimates.

```r
summary(model_fit)

posterior_summary(x=model_fit, probs=c(0.025, 0.975), pars = c("^b_", "sigma"))
```

```
 Family: gaussian
  Links: mu = identity; sigma = identity
Formula: productivity ~ 1 + coffee_quality + (1 + coffee_quality | researcher)
   Data: data (Number of observations: 20)
  Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup draws = 4000

Multilevel Hyperparameters:
~researcher (Number of levels: 4)
                            Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sd(Intercept)                   3.99      3.55     0.13    12.93 1.00     1899     1784
sd(coffee_quality)              1.28      1.19     0.04     4.45 1.00     1773     1653
cor(Intercept,coffee_quality)  -0.18      0.58    -0.98     0.92 1.00     2562     2007

Regression Coefficients:
              Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept        71.01      3.91    63.48    78.43 1.00     2497     1935
coffee_quality   -0.07      0.88    -1.82     1.66 1.00     3998     2548

Further Distributional Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma    10.44      1.34     7.99    13.27 1.00     3476     2666

Draws were sampled 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).
Warning message:
There were 26 divergent transitions after warmup. Increasing adapt_delta above 0.8 may hel
g/misc/warnings.html#divergent-transitions-after-warmup
>
> posterior_summary(x=model_fit, probs=c(0.025, 0.975), pars = c("^b_", "sigma"))
                    Estimate Est.Error      Q2.5     Q97.5
b_Intercept      71.01459818  3.913956 63.476156 78.430934
b_coffee_quality -0.06639076  0.878724 -1.822233  1.659893
sigma            10.43873675  1.335920  7.987791 13.270493
```

After running our model and obtaining the summary, the next step is to check some key diagnostics.

Rhat (Potential Scale Reduction Factor):

Rhat helps assess if the MCMC simulation has converged. An Rhat of 1 indicates good mixing and convergence. An Rhat > 1.1 suggests convergence issues, and more iterations or model adjustments may be needed.

ESS (Effective Sample Size):

ESS measures the quality of the MCMC sample. A high ESS indicates effectively independent samples, while a low ESS indicates high autocorrelation and unreliable estimates. In brms, ESS is divided into:

- Bulk-ESS: Effective sample size in the central region of the posterior.
- Tail-ESS: Effective sample size in the extreme regions of the posterior.

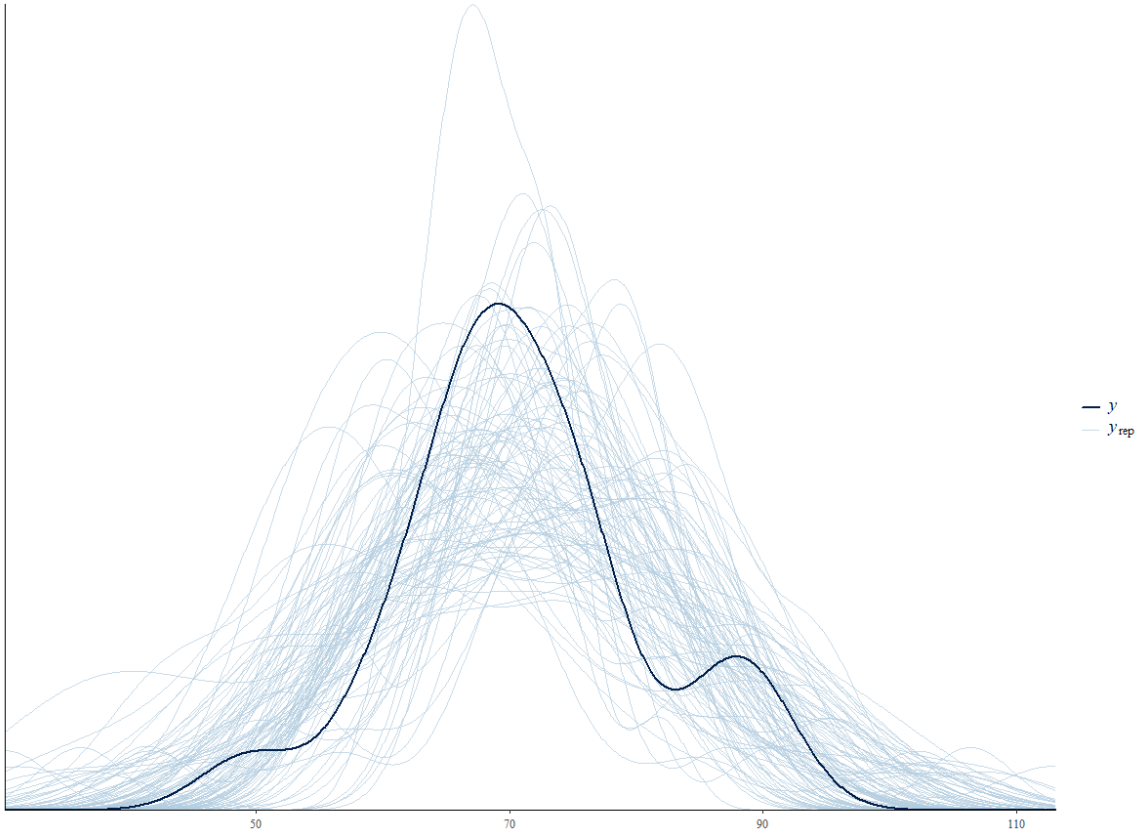
Divergent Transitions:

Divergent transitions indicate numerical instability in the sampler. According to Stan's guidelines, post-warmup divergences compromise estimate validity. In our case, there were 26 divergent transitions after warmup. Fortunately, this can be solved by using adapt_delta.

Adapt delta:

The adapt_delta parameter adjusts the target acceptance probability for the sampler steps. Increasing adapt_delta reduces divergences by making smaller, more careful steps but can also increase computation time. Balancing adapt_delta is key: too high a value can slow down sampling significantly without much benefit.

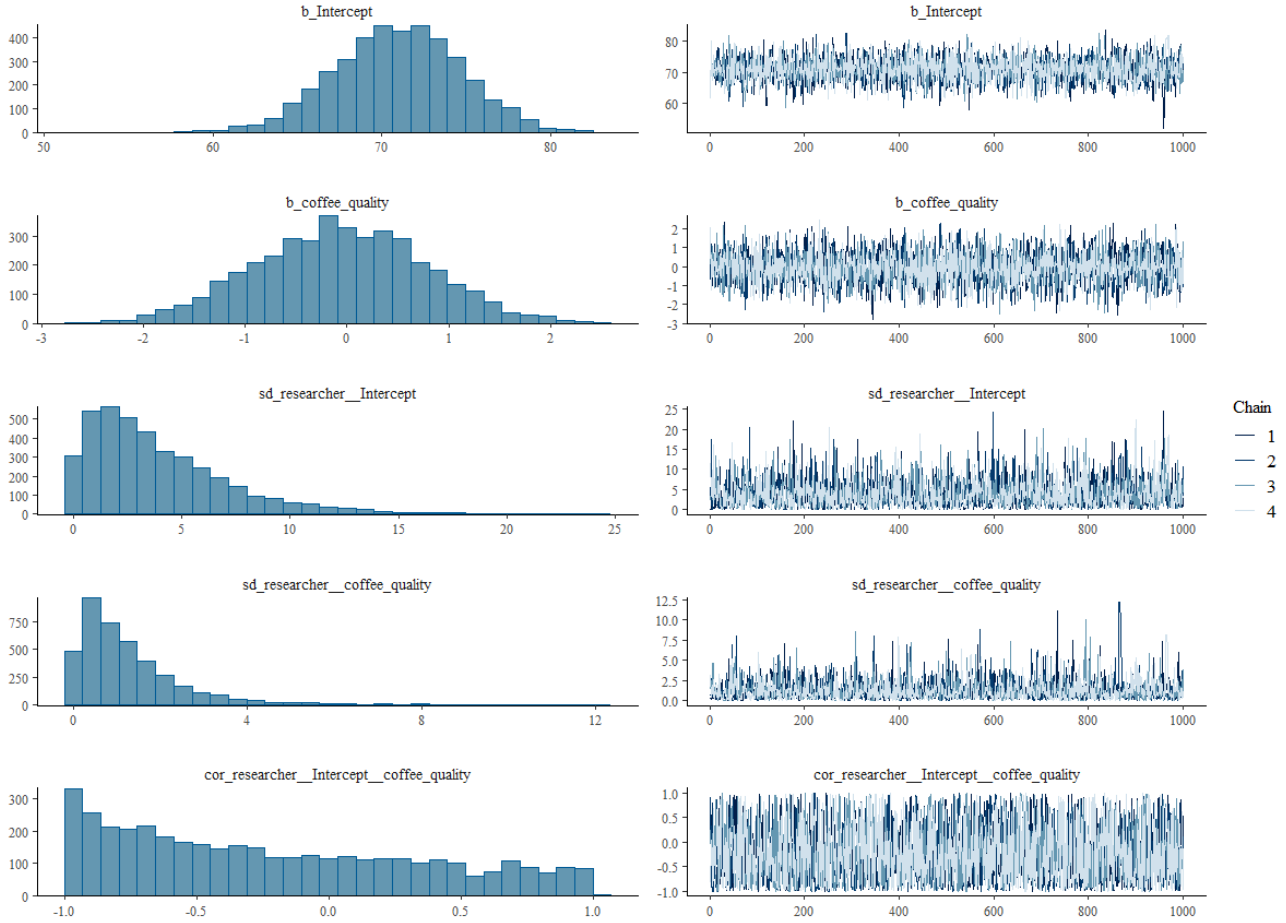
Another important step is to look graphically at the posterior distribution. You can do it by using "pp_check()". It would result in a graph like this one:



Finally, it is always recommended to check the behavior of the chains. While you can already get an insight from the Rhat and ESS, it never hurts to also visualize it. You can visually inspect selected parameters or all of them.

```
# Visually inspect the chain behavior of a few semi_randomly selected parameters
pars <- variables(model_fit)
pars_sel <- c(sample(pars[1:10], 3), sample(pars[-(1:10)], 3))
plot(model_fit, variable = pars_sel, N = 6,
     ask = FALSE, exact_match = TRUE, newpage = TRUE, plot = TRUE)

# directly plot all the parameters
plot(model_fit)
```



D. Interpreting the results

This are the results now that we have used adapt_delta = 0.95.

```
 Family: gaussian
  Links: mu = identity; sigma = identity
Formula: productivity ~ 1 + coffee_quality + (1 + coffee_quality | researcher)
   Data: data (Number of observations: 20)
  Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup draws = 4000

Multilevel Hyperparameters:
~researcher (Number of levels: 4)
                          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sd(Intercept)                 3.93      3.31     0.15    12.56 1.00     1865     1871
sd(coffee_quality)            1.27      1.25     0.04     4.58 1.00     1853     1508
cor(Intercept,coffee_quality)    -0.22      0.58    -0.98     0.92 1.00     3239     2486

Regression Coefficients:
             Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept       70.96      3.75    63.49    78.12 1.00     3383     2493
coffee_quality    -0.06      0.83    -1.69     1.54 1.00     4289     2894

Further Distributional Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma    10.37      1.27     8.09    13.08 1.00     4291     3060

Draws were sampled 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).
```

How can we interpret these results?

A. The correlation between the intercept and the slope for coffee quality is estimated to be -0.22. This suggests a moderate negative correlation between the

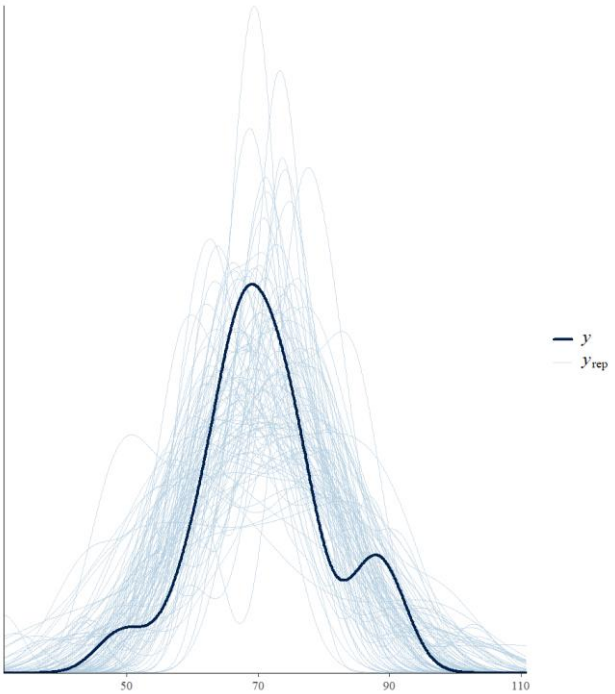
baseline productivity and the effect of coffee quality on productivity, indicating that as coffee quality increases, the effect on productivity decreases.

B. The fixed effect intercept is estimated at 70.96 This suggests that, on average, the daily productivity is at 70.96% when coffee quality is at the reference level.
C. The coefficient for coffee quality is -0.06, with a 95% confidence interval from -1.69 to 1.54. This indicates that for each increase in coffee quality, the productivity decreases by 0.06 units, on average. However, the wide confidence interval suggests a high degree of uncertainty around this estimate.
D. The estimate for the residual standard deviation (sigma) is 10.37. This suggests that the variability in productivity around the model's predictions is moderate.
E. The potential scale reduction factor (Rhat) is 1 for all parameters, indicating that the chains have converged well and that the posterior distributions are not overdispersed.
F. Bulk_ESS and Tail_ESS: These are effective sample size measures, with high values for most parameters, suggesting that the MCMC sampling was efficient.

E. Checking the model's predictions

A way to examine the predictions capacity of the model is to do the posterior predictive checking (PPC). It compares the observed data with the data sampled from the posterior distribution. If the model is good, it should generate data that looks like the observed data.

```
pp_check(object = model_fit, nsamples = 1e2)
```



In this graph we can see in blue to sample observation while the original ones are in black.

F. Comparing models

At this stage, we might want to check whether our model is the one that explains the data best. Hence, we will try another model and then compare them.

```
model_fit2 <- brm(formula = productivity ~ 1 + coffee_quality + (1|researcher),
                  data = data,
                  prior = c(set_prior("normal(70,7)", class = "Intercept"),
                            set_prior("normal(0, 1)", class = "b"),
                            set_prior("normal(10.3,2)", class = "sigma")),
                  family = gaussian(),
                  chains = 4, cores = parallel::detectCores(),
                  warmup = 1000,iter = 2000)
```

In this new model, we decided to only have a random intercept per researcher, which means that we believe that that each researcher will react the same way on the quality of the coffee.

We will also try another model to see whether the day of the week might be as equally important as the quality of the coffee to predict daily_productivity.

```
model_fit3 <- brm(formula = productivity ~ 1 + day + coffee_quality + (1 + coffee_quality|researcher),
                  data = data,
                  prior = c(set_prior("normal(70,7)", class = "Intercept"),
                            set_prior("normal(0, 1)", class = "b"),
                            set_prior("normal(10.3,2)", class = "sigma")),
                  family = gaussian(),
                  chains = 4, cores = parallel::detectCores(),
                  control = list(adapt_delta = 0.95),
                  warmup = 2000,iter = 4000)
```

In this other model we are also considering that the days of the week might have an impact on the productivity as well. Therefore, we are adding a fixed and a random effect of day. This is because it might be possible that some researchers will be more motivated on Mondays than Thursdays for example.

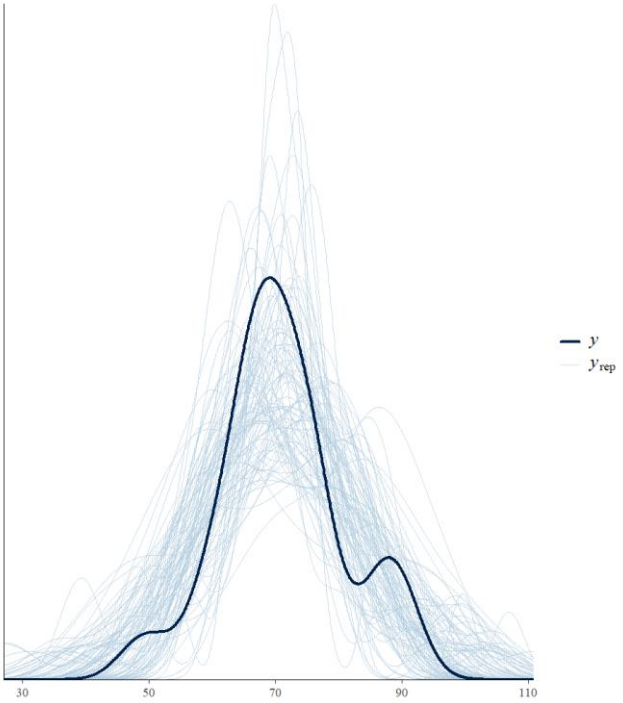
Now that we have three different models, let's see which one is considered best using the Watanabe-Akaike Information Criterion (WAIC). The WAIC is an estimate of out-of-sample prediction error. Lower values indicate a model with better predictive performance.

```
# We will use the WAIC to compare the models, we therefore need to add it to each model
# so we can compare them afterwards
model_fit <- add_criterion(model_fit, "waic")
model_fit2 <- add_criterion(model_fit2,"waic")
model_fit3 <- add_criterion(model_fit3,"waic")

comparaison_model <- loo_compare(model_fit, model_fit2, model_fit3, criterion = "waic")
print(comparaison_model, digits = 2, simplify = FALSE)
```

```
           elpd_diff se_diff elpd_waic se_elpd_waic p_waic se_p_waic waic   se_waic
model_fit3   0.00      0.00  -73.52      2.40         6.78   1.57    147.03   4.80
model_fit2  -2.68      1.21  -76.20      3.52         2.67   0.89    152.40   7.04
model_fit   -3.41      1.17  -76.92      3.46         3.43   1.05    153.85   6.92
```

Apparently, the best model, according to the WAIC, is there 3rd one. It might still be useful to have a look at its posterior distribution:



As confirmed by the WAIC and the PPC, the 3rd model seems to fit the data better. Let's have a look at its summary:

```
 Family: gaussian
  Links: mu = identity; sigma = identity
Formula: productivity ~ 1 + day + coffee_quality + (1 + day + coffee_quality | researcher)
   Data: data (Number of observations: 20)
  Draws: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
         total post-warmup draws = 8000

Multilevel Hyperparameters:
~researcher (Number of levels: 4)
                            Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sd(Intercept)                   3.75      3.34     0.14    11.99 1.00     4606     3919
sd(dayLundi)                    9.45      6.35     0.50    24.94 1.00     2545     2470
sd(dayMardi)                    4.21      3.60     0.19    13.40 1.00     6364     4147
sd(dayMercredi)                 9.35      6.14     0.58    23.66 1.00     3138     3329
sd(dayVendredi)                 4.75      4.02     0.20    14.74 1.00     5772     3878
sd(coffee_quality)              1.25      1.24     0.04     4.57 1.00     4067     3785
cor(Intercept,dayLundi)        -0.06      0.37    -0.73     0.65 1.00     6248     5639
cor(Intercept,dayMardi)        -0.01      0.38    -0.71     0.70 1.00    10351     5281
cor(dayLundi,dayMardi)          0.00      0.37    -0.70     0.70 1.00     9389     4926
cor(Intercept,dayMercredi)      0.04      0.37    -0.67     0.72 1.00     6993     5951
cor(dayLundi,dayMercredi)      -0.15      0.36    -0.77     0.58 1.00     5918     5936
cor(dayMardi,dayMercredi)       0.03      0.38    -0.70     0.73 1.00     5721     6659
cor(Intercept,dayVendredi)     -0.04      0.39    -0.74     0.70 1.00    10215     5239
cor(dayLundi,dayVendredi)      -0.04      0.37    -0.72     0.67 1.00     8524     6057
cor(dayMardi,dayVendredi)       0.01      0.38    -0.70     0.72 1.00     6516     6162
cor(dayMercredi,dayVendredi)    0.02      0.37    -0.69     0.70 1.00     6460     6008
cor(Intercept,coffee_quality)  -0.08      0.39    -0.77     0.67 1.00     9047     5413
cor(dayLundi,coffee_quality)    0.00      0.38    -0.71     0.72 1.00     7998     6152
cor(dayMardi,coffee_quality)   -0.03      0.38    -0.73     0.69 1.00     6552     6410
cor(dayMercredi,coffee_quality) -0.04     0.38    -0.72     0.67 1.00     6749     6252
cor(dayVendredi,coffee_quality) -0.02     0.38    -0.72     0.69 1.00     5489     5922
```

```
Regression Coefficients:
              Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept        71.36      3.80    63.62    78.69 1.00     6987     5817
dayLundi          0.23      0.99    -1.71     2.16 1.00    10519     5907
dayMardi          0.09      0.98    -1.86     1.99 1.00     9700     6069
dayMercredi      -0.09      1.00    -2.05     1.85 1.00    12216     5902
dayVendredi      -0.18      0.99    -2.07     1.77 1.00    12392     5787
coffee_quality   -0.21      0.84    -1.83     1.48 1.00     8869     5748

Further Distributional Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma     8.69      1.85     5.07    12.35 1.00     3003     3787

Draws were sampled 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).
```

How can we interpret these results?

A. The correlation between the intercept and the slope for coffee quality is estimated to be -0.08. This suggests a small negative correlation between the baseline productivity and the effect of coffee quality on productivity, indicating that as coffee quality increases, the effect on productivity decreases.
B. The fixed effect intercept is estimated at 71.36. This suggests that, on average, the daily productivity is at 71.36% when coffee quality is at the reference level.
C. The coefficient for coffee quality is -0.21, with a 95% confidence interval from -1.83 to 1.48 This indicates that for each increase in coffee quality, the productivity decreases by 0.21 units, on average. However, the wide confidence interval suggests a high degree of uncertainty around this estimate.
D. The estimate for the residual standard deviation (sigma) is 8.69. This suggests that the variability in productivity around the model's predictions is small.
E. The potential scale reduction factor (Rhat) is 1 for all parameters, indicating that the chains have fully converged and that the posterior distributions are not overdispersed.
F. Bulk_ESS and Tail_ESS: These are effective sample size measures, with high values for most parameters, suggesting that the MCMC sampling was efficient.

In order to answer our main question, we need to check the credible intervals (CI). In Bayesian analysis, they represent a range of values within which the true parameter value is believed to lie with a specified probability, given the observed data. A parameter effect is considered statistically significant if the credible interval for the parameter does not include zero. This is because a credible interval that excludes zero suggests that there is a high probability that the parameter has a meaningful effect. Coming back to our question, does coffee quality influence daily productivity? Based on the credible intervals provided for the model, we can see that the 95% credible interval for coffee_quality ranges from -1.83 to 1.48. This interval includes zero, indicating that the effect of coffee quality on daily productivity is not statistically significant. Therefore, we do not have strong evidence to conclude that coffee quality has a meaningful impact on daily productivity based on this model. Note that the CIs for all the days of the week also include zero, hence we do not have strong evidence as well to conclude that they have a meaningful impact on daily productivity.

If we wanted to write these results following the APA guidelines, we could write something like this:

"*A Bayesian hierarchical model was applied to assess the impact of coffee quality and the day of the week on productivity. The 95% credible interval for coffee quality (-1.83 to 1.48) included zero, suggesting a non-significant effect on daily productivity. Likewise, credible intervals for each weekday also spanned zero, indicating no discernible influence on productivity throughout the week. Thus, based on this analysis, there is insufficient evidence to assert that either coffee quality or specific weekdays significantly affect daily productivity.*"

In case you want to go a step further and perform hypothesis testing for your model parameters (for example, is the interaction between days and coffee_quality < 0), I would recommend using brms::hypothesis. Here is a description of this function from the package's ressources: "*hypothesis computes an evidence ratio for each hypothesis. For a one-sided hypothesis, this is just the posterior probability under the hypothesis against its alternative. That is, when the hypothesis is of the form a > b, the evidence ratio is the ratio of the posterior probability of a > b and the posterior probability of a < b. In this example, values greater than one indicate that the evidence in favor of a > b is larger than evidence in favor of a < b. For a two-sided hypothesis, the evidence ratio is a Bayes factor between the hypothesis and its alternative computed via the Savage-Dickey density ratio method. That is the posterior density at the point of interest divided by the prior density at that point. Values greater than one indicate that evidence in favor of the point hypothesis has increased after seeing the data.*"