Di(stribution)versity

How distributions for regression are parameterized in brms
Version 0.1

Holger Sennhenn-Reulen

June 30, 2020

Authorship Statement   This document, *Di(stribution)versity*, is intended for educational purposes. It depends on a vignette[1] by Paul-Christian Bürkner, the author of the brms package (Bürkner, 2017, 2018). This vignette introduces many (all?) of the response distributions that are implemented in brms. The complete structure of this document is a copy of this vignette, all of the equations have been completely copied from there – I just slightly changed notation a little here and there. My intention is to use this structure and these contents and add short R code chunks with which I show how the parameterisations in brms translate to base R (R Core Team, 2019) parameterisations, and how artificial data can be simulated for the respective distributions.

---

[1]`https://cran.r-project.org/web/packages/brms/vignettes/brms_distributions.html`

Contents

References 18

1 Notation

Throughout this document, the *response* or *outcome* variable will be denoted as y, a probability density function as p, and we will use μ to refer to the the parameter of the distribution which is usually in the main focus: the mean or *expectation* of the response or some closely related quantity.

All the following distributions are models for the unknown data-generating mechanism. There are distributions that are defined on only one parameter, θ, or on two or more parameters, $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots)^\top$. μ is one of those parameters.

With regression models, our aim is to quantify how those parameters vary if influencing variables x are varying. The quantity – which is the linear predictor η – that captures influencing variables x and their corresponding influences β is constructed to take on values on the real line. See `help(brmsformula)` for details on the possibilities with the use of *brms* for the construction of η.

The domain for any of those parameters θ is sometimes the real line, but most often, not. We therefore need a transformation function that links the parameter-specific linear predictor, η_θ, to the respective parameters of interest. This transformation function is either called the *link-function*, h_θ, if the focus is on the linear predictor:

$$\eta_\theta = h_\theta\left(\theta\right)$$

or the *response function*, g_θ – ie. $h_\theta^{-1} = g_\theta$ –, if the focus is on the response:

$$\theta = g_\theta\left(\eta_\theta\right).$$

2 Location shift models

For location shift models, y can be any real value.

2.1 Gaussian

The density of the *Gaussian* or *normal distribution* is given by:

$$p\left(y\right) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2\right),$$

where μ is the *expectation* and σ is the standard deviation of the *residuals*, $y - \mu$.

```
library("brms")
set.seed(123456789)
N <- 1000
mu <- 0.7
sigma <- 1.4
y <- rnorm(n = N, mean = mu, sd = sigma)
fit <- brm(y ~ 1, distribution = "gaussian", data = data.frame(y = y), seed = 123456789)
B <- as.matrix(fit)
str(B)
par(mfrow = c(1, 2))
boxplot(B[, "b_Intercept"])
abline(h = mu)
boxplot(B[, "sigma"])
abline(h = sigma)
```

2.2 Student-t

The density of the *Student* or *Student-t distribution* is given by:

$$p\left(y\right) = \frac{\Gamma((\nu+1)/2)}{\Gamma(\nu/2)} \frac{1}{\sqrt{\nu\pi}\sigma} \left(1 + \frac{1}{\nu}\left(\frac{y-\mu}{\sigma}\right)^2\right)^{-(\nu+1)/2}$$

μ is the *expectation*, σ is the standard deviation of the *residuals*, $y - \mu$, Γ denotes the *gamma function* and $\nu > 1$ is the *degrees of freedom*. The Student-t distribution is related to the Gaussian distribution: As $\nu \to \infty$, the Student-t becomes the Gaussian.

```
library("brms")
set.seed(123456789)
N <- 1000
mu <- -7
sigma <- 14
nu <- 9
y <- mu + sigma * rt(n = N, df = nu)
fit <- brm(y ~ 1, distribution = "student", data = data.frame(y = y), seed = 123456789)
B <- as.matrix(fit)
str(B)
par(mfrow = c(1, 3))
boxplot(B[, "b_Intercept"])
abline(h = mu)
boxplot(B[, "sigma"])
abline(h = sigma)
boxplot(B[, "nu"])
abline(h = nu)
```

2.3 Skew Normal

The density of the *skew normal* distribution is given by:

$$p\left(y\right) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{1}{2}\left(\frac{y-\xi}{\omega}\right)^2\right)\left(1 + \text{erf}\left(\alpha\left(\frac{y-\xi}{\omega\sqrt{2}}\right)\right)\right)$$

where ξ is the location parameter, ω is the positive scale parameter, α the skewness parameter, and erf denotes the error function of the gaussian distribution. To parameterize the skew-normal distribution in terms of the mean μ and standard deviation σ, ω and ξ are computed as:

$$\omega = \frac{\sigma}{\sqrt{1 - \frac{2}{\pi}\frac{\alpha^2}{1+\alpha^2}}},$$

and

$$\xi = \mu - \omega\frac{\alpha}{\sqrt{1+\alpha^2}}\sqrt{\frac{2}{\pi}}.$$

If $\alpha = 0$, the skew-normal distribution becomes the gaussian distribution.

There is no skew normal distribution implemented in base R, but it is in Stan (Stan Development Team, 2019):

```
library("rstan")
prg <- '
data{
  int N;
  real mu;
  real<lower=0> sigma;
  real alpha;
  }
parameters {}
model {}
generated quantities{
  real y_sim[N];
  for (n in 1:N)
    y_sim[n] = skew_normal_rng(mu, sigma, alpha);
  }'
N <- 1000
mu <- -7
sigma <- 14
alpha <- 4
## auxiliary parameter to calculate expected value and scale of the skew normal:
delta <- alpha / sqrt(1 + alpha^2)
## expected value of the skew_normal:
skew_normal_mu <- mu + sigma * delta * sqrt(2/pi)
## scale of the skew_normal:
skew_normal_scale <- sqrt(sigma^2 * (1 - 2 * delta^2 / pi))
## use stan as a random number generator:
sim <- stan(model_code = prg, algorithm = "Fixed_param", chains = 1, iter = 1,
            data = list(N = N, mu = mu, sigma = sigma, alpha = alpha), seed = 123456789)
Y <- as.matrix(sim)
y <- Y[1, 1:N]
hist(y, freq = F)
## density of the skew normal (source: wikipedia)
d_skew_normal <- function(x, mu, sigma, alpha) {
  x <- (x - mu)/sigma
  return(2 * dnorm(x = x) * pnorm(q = alpha * x) / sigma)
}
x <- seq(min(y), max(y), length.out = 200)
# sum(diff(x)[1]*d_skew_normal(x, mu = mu, sigma = sigma, alpha = alpha)) ## Check prob. density
lines(x, d_skew_normal(x, mu = mu, sigma = sigma, alpha = alpha))
library("brms")
## from 'make_stancode(y ~ 1, distribution = "skew_normal", data = data.frame(y = y))':
## // use efficient skew-normal parameterization
## delta = alpha / sqrt(1 + alpha^2);
```

```
## omega = sigma / sqrt(1 - sqrt_2_div_pi^2 * delta^2);
## for (n in 1:N) {
##   mu[n] = mu[n] - omega * delta * sqrt_2_div_pi;
## }
## ...
## target += skew_normal_lpdf(Y | mu, omega, alpha);
fit <- brm(y ~ 1, distribution = "skew_normal", data = data.frame(y = y), seed = 123456789)
summary(fit)
B <- as.matrix(fit)
str(B)
par(mfrow = c(1, 3))
boxplot(B[, "b_Intercept"])
abline(h = skew_normal_mu)
abline(h = mean(y), lty = 2)
boxplot(B[, "sigma"])
abline(h = skew_normal_scale)
abline(h = sd(y), lty = 2)
boxplot(B[, "alpha"])
abline(h = alpha)
abline(h = mean((y - mean(y))^3)/sd(y)^3)
```

3 Binary and count data models

3.1 Binomial

The density of the *binomial* distribution is given by:

$$p\left(y\right) = \binom{N}{y}\mu^y(1-\mu)^{N-y}$$

where N is the number of trials and $y \in \{0, ..., N\}$. When all N are 1 (i.e., $y \in \{0, 1\}$), the *Bernoulli* distribution for binary data arises.

3.2 Poisson

For $y \in \mathbb{N}_0$, the density of the *Poisson* distribution is given by

$$p\left(y\right) = \frac{\mu^y}{y!}\exp(-\mu)$$

3.3 Negative Binomial

The density of the *negbinomial* (negative binomial) distribution is

$$p\left(y\right) = \binom{y + \phi - 1}{y}\left(\frac{\mu}{\mu + \phi}\right)^y\left(\frac{\phi}{\mu + \phi}\right)^\phi$$

where ϕ is a positive precision parameter. For $\phi \to \infty$, the negative binomial distribution becomes the poisson distribution.

3.4 Geometric

The density of the *geometric* distribution arises if ϕ is set to 1.

3.5 Discrete Weibull

This distribution was a comment and not visible in the vignette on June 30, 2020. I need to check why? Maybe the discrete Weibull is not implemented (yet/anymore/…)?
The density of the *discrete Weibull* distribution is

$$p\left(y\right) = \mu^{y^\alpha} - \mu^{(y+1)^\alpha}$$

with location parameter $\mu \in [0, 1]$ and positive shape parameter α.

4 Survival models

With survival models we mean all models that are defined on the positive reals only, that is $y \in \mathbb{R}^+$.

4.1 Lognormal

The density of the *lognormal* distribution is given by:

$$p\left(y\right) = \frac{1}{\sqrt{2\pi}\sigma x} \exp\left(-\frac{1}{2}\left(\frac{\log(y) - \mu}{\sigma}\right)^2\right),$$

where σ is the residual standard deviation on the log-scale.

4.2 Gamma

The density of the *Gamma* distribution is given by:

$$p\left(y\right) = \frac{(\alpha/\mu)^\alpha}{\Gamma(\alpha)} y^{\alpha-1} \exp\left(-\frac{\alpha y}{\mu}\right),$$

where α is a positive shape parameter.

4.3 Weibull

The density of the *Weibull* distribution is given by

$$p\left(y\right) = \frac{\alpha}{s}\left(\frac{y}{s}\right)^{\alpha-1} \exp\left(-\left(\frac{y}{s}\right)^\alpha\right)$$

where α is again a positive shape parameter and $s = \mu/\Gamma(1 + 1/\alpha)$ is the scale parameter to that μ is the mean of the distribution.

4.4 Exponential

The *exponential* distribution arises if α is set to 1 for either the gamma or Weibull distribution.

4.5 Inverse Gaussian

The density of the *inverse Gaussian* distribution is given by:

$$p\left(y\right) = \left(\frac{\alpha}{2\pi y^3}\right)^{1/2} \exp\left(\frac{-\alpha(y - \mu)^2}{2\mu^2 y}\right)$$

where α is a positive shape parameter.

5 Extreme value models

Modeling extremes requires special distributions.

5.1 Frechet

One may use the *Weibull* distribution (see above) or the *Frechet* distribution with density:

$$p\left(y\right) = \frac{\nu}{s}\left(\frac{y}{s}\right)^{-1-\nu}\exp\left(-\left(\frac{y}{s}\right)^{-\nu}\right)$$

where $s = \mu/\Gamma(1 - 1/\nu)$ is a positive scale parameter and $\nu > 1$ is a shape parameter so that μ predicts the mean of the Frechet distribution.

5.2 Generalized Extreme Value

A generalization of both distributions is the generalized extreme value distribution (distribution *gen_extreme_value*) with density:

$$p\left(y\right) = \frac{1}{\sigma}t(y)^{-1-1/\xi}\exp(-t(y)),$$

where:

$$t(y) = \left(1 + \xi\left(\frac{y-\mu}{\sigma}\right)\right)^{-1/\xi}$$

with positive scale parameter σ and shape parameter ξ.

6 Response time models

6.1 Exponentially Modified Gaussian

One distribution that is especially suited to model reaction times is the *exgaussian* ('exponentially modified Gaussian') distribution. Its density is given by:

$$p\left(y\right) = \frac{1}{2\beta} \exp\left(\frac{1}{2\beta}\left(2\xi + \sigma^2/\beta - 2y\right)\right) \text{erfc}\left(\frac{\xi + \sigma^2/\beta - y}{\sqrt{2}\sigma}\right)$$

where β is the scale (inverse rate) of the exponential component, ξ is the mean of the Gaussian component, σ is the standard deviation of the Gaussian component, and erfc is the complementary error function. We parameterize $\mu = \xi + \beta$ so that the main predictor term equals the mean of the distribution.

6.2 Shifted Lognormal

Another distribution well suited for modeling response times is the *shifted_lognormal* distribution. It's density equals that of the *lognormal* distribution except that the whole distribution is shifted to the right by a positive parameter called ndt (for consistency with the *wiener* diffusion model explained below).

6.3 Wiener Diffusion Model

A distribution concerned with the combined modeling of reaction times and corresponding binary responses is the *wiener* diffusion model. It has four model parameters each with a natural interpretation. The parameter $\alpha > 0$ describes the separation between two boundaries of the diffusion process, $\tau > 0$ describes the non-decision time (e.g., due to image or motor processing), $\beta \in [0, 1]$ describes the initial bias in favor of the upper alternative, and $\delta \in \mathbb{R}$ describes the drift rate to the boundaries (a positive value indicates a drift towards to upper boundary). The density for the reaction time at the upper boundary is given by:

$$p\left(y\right) = \frac{\alpha}{(y-\tau)^3/2} \exp\left(-\delta\alpha\beta - \frac{\delta^2(y-\tau)}{2}\right) \sum_{k=-\infty}^{\infty} (2k+\beta)\phi\left(\frac{2k+\alpha\beta}{\sqrt{y-\tau}}\right)$$

where $\phi(x)$ denotes the standard normal density function. The density at the lower boundary can be obtained by substituting $1-\beta$ for β and $-\delta$ for δ in the above equation. In brms the parameters α, τ, and β are modeled as auxiliary parameters named bs ('boundary separation'), ndt ('non-decision time'), and bias respectively, whereas the drift rate δ is modeled via the ordinary model formula that is as $\delta = \mu$.

7 Quantile regression

Quantile regression is implemented via distribution *asym_laplace* (asymmetric Laplace distribution) with density:

$$p\left(y\right) = \frac{p(1-p)}{\sigma} \exp\left(-\rho_p\left(\frac{y-\mu}{\sigma}\right)\right)$$

where ρ_p is given by $\rho_p(x) = x(p - I_{x<0})$ and I_A is the indicator function of set A. The parameter σ is a positive scale parameter and p is the `quantile` parameter taking on values in $(0,1)$. For this distribution, we have $P(Y < g(\eta)) = p$. Thus, quantile regression can be performed by fixing p to the quantile to interest.

8 Probability models

8.1 Beta

The density of the *Beta* distribution for $y \in (0, 1)$ is given by:

$$p\left(y\right) = \frac{y^{\mu\phi-1}(1-y)^{(1-\mu)\phi-1}}{B(\mu\phi, (1-\mu)\phi)}$$

where B is the beta function and ϕ is a positive precision parameter.

8.2 Dirichlet

A multivariate generalization of the *Beta* distribution is the *dirichlet* distribution with density:

$$p\left(y\right) = \frac{1}{B((\mu_1, \ldots, \mu_K)\phi)} \prod_{k=1}^{K} y_k^{\mu_k\phi-1}.$$

The *dirichlet* distribution is only implemented with the multivariate logit link function so that

$$\mu_j = \frac{\exp(\eta_j)}{\sum_{k=1}^{K} \exp(\eta_k)}$$

For reasons of identifiability, η_1 is set to 0.

9 Circular models

9.1 Von Mises

The density of the *von_mises* distribution for $y \in (-\pi, \pi)$ is given by:

$$p\left(y\right) = \frac{\exp(\kappa \cos(y - \mu))}{2\pi I_0(\kappa)}$$

where I_0 is the modified Bessel function of order 0 and κ is a positive precision parameter.

10 Ordinal and categorical models

For ordinal and categorical models, y is one of the categories $1, ..., K$. The intercepts of ordinal models are called thresholds and are denoted as τ_k, with $k \in \{1, ..., K-1\}$, whereas η does not contain a fixed effects intercept. Note that the applied link functions h are technically distribution functions $\mathbb{R} \to [0, 1]$.

10.1 Cumulative

The density of the *cumulative* distribution (implementing the most basic ordinal model) is given by:

$$p(y) = g(\tau_{y+1} - \eta) - g(\tau_y - \eta)$$

10.2 Stopping and continuation ratio

The densities of the *sratio* (stopping ratio) and *cratio* (continuation ratio) distributions are given by:

$$p(y) = g(\tau_{y+1} - \eta) \prod_{k=1}^{y} (1 - g(\tau_k - \eta))$$

and

$$p(y) = (1 - g(\eta - \tau_{y+1})) \prod_{k=1}^{y} g(\eta - \tau_k)$$

respectively. Note that both distributions are equivalent for symmetric link functions such as logit or probit.

10.3 Adjacent category

The density of the *acat* (adjacent category) distribution is given by

$$p(y) = \frac{\prod_{k=1}^{y} g(\eta - \tau_k) \prod_{k=y+1}^{K} (1 - g(\eta - \tau_k))}{\sum_{k=0}^{K} \prod_{j=1}^{k} g(\eta - \tau_j) \prod_{j=k+1}^{K} (1 - g(\eta - \tau_j))}$$

For the logit link, this can be simplified to

$$p(y) = \frac{\exp\left(\sum_{k=1}^{y} (\eta - \tau_k)\right)}{\sum_{k=0}^{K} \exp\left(\sum_{j=1}^{k} (\eta - \tau_j)\right)}$$

The linear predictor η can be generalized to also depend on the category k for a subset of predictors. This leads to category specific effects (for details on how to specify them see `help(brm)`). Note that *cumulative* and *sratio* models use $\tau - \eta$, whereas *cratio* and *acat* use $\eta - \tau$. This is done to ensure that larger values of η increase the probability of higher response categories.

10.4 Categorical

The *categorical* distribution is currently only implemented with the multivariate logit link function and has density:

$$p(y) = \mu_y = \frac{\exp(\eta_y)}{\sum_{k=1}^{K} \exp(\eta_k)}$$

Note that η does also depend on the category k. For reasons of identifiability, η_1 is set to 0.

10.5 Multinomial

A generalization of the *categorical* distribution to more than one trial is the *multinomial* distribution with density

$$p\left(y\right) = \binom{N}{y_1, y_2, \ldots, y_K} \prod_{k=1}^{K} \mu_k^{y_k}$$

where, for each category, μ_k is estimated via the multivariate logit link function shown above.

11 Zero-Inflated and Hurdle Models

Zero-inflated and *hurdle* distributions extend existing distributions by adding special processes for responses that are zero.

11.1 Zero-Inflated

The density of a *zero-inflated* distribution is given by:

$$p_z(y) = z + (1-z)p(0) \quad \text{if } y = 0 \qquad p_z(y) = (1-z)p(y) \quad \text{if } y > 0$$

where z denotes the zero-inflation probability. Currently implemented distributions are *zero_inflated_poisson*, *zero_inflated_binomial*, *zero_inflated_negbinomial*, and *zero_inflated_beta*.

11.2 Hurdle

The density of a *hurdle* distribution is given by:

$$f_z(y) = z \quad \text{if } y = 0 \qquad f_z(y) = (1-z)p(y)/(1-f(0)) \quad \text{if } y > 0$$

Currently implemented distributions are *hurdle_poisson*, *hurdle_negbinomial*, *hurdle_gamma*, and *hurdle_lognormal*.

11.3 Zero-One-Inflated

The density of a *zero-one-inflated* distribution is given by

$$f_{\alpha,\gamma}(y) = \alpha(1-\gamma) \quad \text{if } y = 0 \qquad f_{\alpha,\gamma}(y) = \alpha\gamma \quad \text{if } y = 1 \qquad f_{\alpha,\gamma}(y) = (1-\alpha)p(y) \quad \text{if } y \notin \{0,1\}$$

where α is the zero-one-inflation probability (i.e. the probability that zero or one occurs) and γ is the conditional one-inflation probability (i.e. the probability that one occurs rather than zero). Currently implemented distributions are *zero_one_inflated_beta*.

References

Bürkner, P.-C. (2017). brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, 80(1):1–28.

Bürkner, P.-C. (2018). Advanced Bayesian multilevel modeling with the R package brms. *The R Journal*, 10(1):395–411.

R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Stan Development Team (2019). Stan: A C++ library for probability and sampling, version 2.19.2. http://mc-stan.org/.