

---

**brmp**

**Nov 18, 2020**



---

## Contents

---

|          |                           |           |
|----------|---------------------------|-----------|
| <b>1</b> | <b>brmp</b>               | <b>1</b>  |
| <b>2</b> | <b>Formula</b>            | <b>5</b>  |
| <b>3</b> | <b>Fit</b>                | <b>7</b>  |
| <b>4</b> | <b>Family</b>             | <b>9</b>  |
| <b>5</b> | <b>Priors</b>             | <b>11</b> |
| <b>6</b> | <b>Backends</b>           | <b>13</b> |
| <b>7</b> | <b>Indices and tables</b> | <b>15</b> |
|          | <b>Index</b>              | <b>17</b> |



`brmp.brm` (*formula\_str*, *df*, *family=None*, *priors=None*, *contrasts=None*)

Defines a model and encodes data in design matrices.

By default categorical columns are coded using dummy coding.

#### Parameters

- **formula\_str** (*str*) – An lme4 formula. e.g. `'y ~ 1 + x'`. See [Formula](#) for a description of the supported syntax.
- **df** (*pandas.DataFrame*) – A data frame containing columns for each of the variables in *formula\_str*.
- **family** (*brmp.family.Family*) – The model's response family.
- **priors** (*list*) – A list of [Prior](#) instances describing the model's priors.
- **contrasts** (*dict*) – A dictionary that optionally maps variable names to contrast matrices describing custom encodings of categorical variables. Each contrast matrix should be a `ndarray` of shape `(L, C)`, where `L` is the number of levels present in the categorical variable and `C` is the length of the desired encoding.

**Returns** A wrapper around the model description and the design matrices.

**Return type** *brmp.ModelAndData*

Example:

```
df = pd.DataFrame({'y': [1., 2.], 'x': [.5, 0.]})
model = brm('y ~ 1 + x', df)
```

**class** `brmp.ModelAndData` (*model*, *df*, *data*)

**data** = `None`

A dictionary with the encoded data as its values. Each value is a `ndarray`.

**fit** (*algo='nuts'*, *\*\*kwargs*)

Fits the wrapped model.

**Parameters**

- **algo** (*str*) – The algorithm used for inference, either 'nuts', 'svi' or 'prior'.
- **kwargs** – Inference algorithm-specific keyword arguments. See `nuts()`, `svi()` and `prior()` for details.

**Returns** A model fit.

**Return type** *brmp.fit.Fit*

Example:

```
fit = brm('y ~ x', df).fit()
```

**nuts** (*iter=10, warmup=None, num\_chains=1, seed=None, backend=<Backend name="NumPyro">*)  
Fit the model using NUTS.

**Parameters**

- **iter** (*int*) – The number of (post warm up) samples to take.
- **warmup** (*int*) – The number of warm up samples to take. Warm up samples are not included in the final model fit. Defaults to `iter / 2`.
- **num\_chains** (*int*) – The number of chains to run.
- **seed** (*int*) – Random seed.
- **backend** (*brmp.backend.Backend*) – The backend used to perform inference.

**Returns** A model fit.

**Return type** *brmp.fit.Fit*

Example:

```
fit = brm('y ~ x', df).nuts()
```

**svi** (*iter=10, num\_samples=10, seed=None, backend=<Backend name="Pyro">, \*\*kwargs*)  
Fit the model using stochastic variational inference.

**Parameters**

- **iter** (*int*) – The number of optimisation steps to take.
- **num\_samples** (*int*) – The number of samples to take from the variational posterior.
- **seed** (*int*) – Random seed.
- **backend** (*brmp.backend.Backend*) – The backend used to perform inference.

**Returns** A model fit.

**Return type** *brmp.fit.Fit*

Example:

```
fit = brm('y ~ x', df).svi()
```

**prior** (*num\_samples=10, seed=None, backend=<Backend name="Pyro">*)  
Sample from the prior.

**Parameters**

- **num\_samples** (*int*) – The number of samples to take.

- **seed** (*int*) – Random seed.
- **backend** (`brmp.backend.Backend`) – The backend used to perform inference.

**Returns** A model fit.

**Return type** *brmp.fit.Fit*

Example:

```
fit = brm('y ~ x', df).prior()
```



**class** brmp.formula.**Formula** (*response, terms, groups*)  
 Represents an lme4 formula.

|   |                                                                                                                                                              |
|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ~ | A valid formula contains exactly one occurrence of ~ . The LHS gives the name of the scalar response variable. The RHS describes the structure of the model. |
| + | A combination of terms.                                                                                                                                      |
| : | An interaction between two terms. Can also appear on the RHS of   or    to specify grouping by multiple factors.                                             |
|   | Introduces a group-level term. (i.e. random effect.)                                                                                                         |
|   | Introduces a group-level term, omitting modelling of group-level correlations.                                                                               |
| 1 | Intercept term. Note that intercept terms are not added automatically.                                                                                       |

The following examples are all parsed as valid formulae:

```
y ~ x
y ~ 1 + x
y ~ 1 + x1:x2
y ~ 1 + x1 + (1 + x2 | a)
y ~ 1 + x1 + (1 + x2 || a:b)
```



```
class brmp.fit.Fit(*args, **kwargs)
```

**fitted** (*what*='expectation', *data*=None, *seed*=None)

Produces predictions from the fitted model.

Predicted values are computed for each sample collected during inference, and for each row in the data set.

### Parameters

- **what** (*str*) – The value to predict. Valid arguments and their effect are described below:

|               |                                                                                                                                                |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| 'expectation' | Computes the expected value of the response distribution.                                                                                      |
| 'sample'      | Draws a sample from the response distribution.                                                                                                 |
| 'response'    | Computes the output of the model followed by any inverse link function. i.e. The value of the location parameter of the response distribution. |
| 'linear'      | Computes the output of the model prior to the application of any inverse link function.                                                        |

- **data** (*pandas.DataFrame*) – The data from which to compute the predictions. When omitted, the data on which the model was fit is used.
- **seed** (*int*) – Random seed. Used only when 'sample' is given as the 'what' argument.

**Returns** An array with shape (*S*, *N*). Where *S* is the number of samples taken during inference and *N* is the number of rows in the data set used for prediction.

**Return type** *numpy.ndarray*

**marginals** (*qs*=[0.025, 0.25, 0.5, 0.75, 0.975])

Produces a table containing statistics of the marginal distributions of the parameters of the fitted model.

**Parameters** **qs** (*list*) – A list of quantiles to include in the output.

**Returns** A table of marginal statistics.

**Return type** brmp.fit.ArrReprWrapper

Example:

```
fit = brm('y ~ x', df).fit()
print(fit.marginals())
```

| #       | mean | sd   | 2.5%  | 25%  | 50%  | 75%  | 97.5% | n_eff | r_hat |
|---------|------|------|-------|------|------|------|-------|-------|-------|
| # b_x   | 0.42 | 0.33 | -0.11 | 0.14 | 0.48 | 0.65 | 0.88  | 5.18  | 1.00  |
| # sigma | 0.78 | 0.28 | 0.48  | 0.61 | 0.68 | 0.87 | 1.32  | 5.28  | 1.10  |

**get\_scalar\_param**(name, preserve\_chains=False)

Extracts the values sampled for a single parameter from a model fit.

**Parameters**

- **name** (*str*) – The name of a parameter of the model. Valid names are those shown in the output of `marginals()`.
- **preserve\_chains** (*bool*) – Whether to group samples by the MCMC chain on which they were collected.

**Returns** An array with shape  $(S,)$  when `preserve_chains=False`,  $(C, S)$  otherwise. Where  $S$  is the number of samples taken during inference, and  $C$  is the number of MCMC chains run.

**Return type** `numpy.ndarray`

```
class brmp.family.Family (name, params, support, link)
```

```
brmp.family.Normal = Normal()
```

**Parameters**

- **mu** – mean
- **sigma** – standard deviation

```
brmp.family.Bernoulli = Bernoulli()
```

**Parameters** **probs** – success probability

```
brmp.family.Cauchy = Cauchy()
```

**Parameters**

- **loc** – location
- **scale** – scale

```
brmp.family.LKJ = LKJ()
```

**Parameters** **eta** – shape

```
brmp.family.Binomial = Binomial()
```

**Parameters**

- **num\_trials** – number of trials
- **probs** – success probability

```
brmp.family.HalfNormal = HalfNormal()
```

**Parameters** **sigma** – scale

```
brmp.family.StudentT = StudentT()
```

**Parameters**

- **df** – degrees of freedom

- **loc** – location
- **scale** – scale

## Priors

**class** brmp.priors.**Prior** (*path*, *prior*)

A **Prior** instance associates a prior distribution with one or more parameters of a model. One or more such instances may be passed to `brm()` to override its default choice of priors.

The parameters of the model to which a prior should be applied are specified using a path. The following examples illustrate how this works:

| Path                     | Selected Parameters                                                           |
|--------------------------|-------------------------------------------------------------------------------|
| ('b',)                   | All population level coefficients                                             |
| ('b', 'intercept')       | The population level intercept                                                |
| ('b', 'x')               | The population level coefficient x                                            |
| ('sd',)                  | All standard deviations in all groups                                         |
| ('sd', 'a')              | All standard deviations in the group which groups by column a                 |
| ('sd', 'a:b')            | All standard deviations in the group which groups by columns a and b          |
| ('sd', 'a', 'intercept') | The standard deviation of the intercept in the group which groups by column a |
| ('cor',)                 | All correlation matrices                                                      |
| ('cor', 'a')             | The correlation matrix of the group which groups by column a                  |
| ('cor', 'a:b')           | The correlation matrix of the group which groups by columns a and b           |
| ('resp', `sigma`)        | The sigma parameter of the response distribution                              |

Example:

```
Prior(('b', 'intercept'), Normal(0., 1.))
```

### Parameters

- **path** (*tuple*) – A path describing one or more parameters of the model.
- **prior** (`brmp.family.Family`) – A prior distribution, given as a *Family* with all of its parameters specified.



## CHAPTER 6

---

### Backends

---

```
class brmp.backend.Backend
brmp.numpyro_backend.backend = <Backend name="NumPyro">
brmp.pyro_backend.backend = <Backend name="Pyro">
```



## CHAPTER 7

---

### Indices and tables

---

- `genindex`
- `search`



## B

Backend (*class in brmp.backend*), 13  
backend (*in module brmp.numpyro\_backend*), 13  
backend (*in module brmp.pyro\_backend*), 13  
Bernoulli (*in module brmp.family*), 9  
Binomial (*in module brmp.family*), 9  
brm() (*in module brmp*), 1

## C

Cauchy (*in module brmp.family*), 9

## D

data (*brmp.ModelAndData attribute*), 1

## F

Family (*class in brmp.family*), 9  
Fit (*class in brmp.fit*), 7  
fit() (*brmp.ModelAndData method*), 1  
fitted() (*brmp.fit.Fit method*), 7  
Formula (*class in brmp.formula*), 5

## G

get\_scalar\_param() (*brmp.fit.Fit method*), 8

## H

HalfNormal (*in module brmp.family*), 9

## L

LKJ (*in module brmp.family*), 9

## M

marginals() (*brmp.fit.Fit method*), 7  
ModelAndData (*class in brmp*), 1

## N

Normal (*in module brmp.family*), 9  
nuts() (*brmp.ModelAndData method*), 2

## P

Prior (*class in brmp.priors*), 11  
prior() (*brmp.ModelAndData method*), 2

## S

StudentT (*in module brmp.family*), 9  
svi() (*brmp.ModelAndData method*), 2