

# Package ‘bayesplay’

May 7, 2026

**Type** Package

**Title** The Bayes Factor Playground

**Version** 0.9.3

**Description** A lightweight modelling syntax for defining likelihoods and priors and for computing Bayes factors for simple one parameter models. It includes functionality for computing and plotting priors, likelihoods, and model predictions. Additional functionality is included for computing and plotting posteriors.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Suggests** testthat, covr, knitr, rmarkdown, markdown, vdiff, ggplot2, patrick

**VignetteBuilder** knitr

**RoxygenNote** 7.2.3

**Imports** methods, gginnards, stats

**URL** <https://github.com/bayesplay/bayesplay>

**BugReports** <https://github.com/bayesplay/bayesplay/issues>

**NeedsCompilation** no

**Author** Lincoln John Colling [aut, cre] (ORCID:  
<<https://orcid.org/0000-0002-3572-7758>>)

**Maintainer** Lincoln John Colling <[lincoln@colling.net.nz](mailto:lincoln@colling.net.nz)>

**Repository** CRAN

**Date/Publication** 2023-04-13 12:10:02 UTC

## Contents

extract_posterior	2
extract_predictions	2
integral	3
likelihood	3

names,bayesplay-method . . . . .	5
plot . . . . .	6
prior . . . . .	6
sd_ratio . . . . .	8
summary,bf-method . . . . .	9
visual_compare . . . . .	9

## Index 11

extract\_posterior      *Extract the posterior*

### Description

Extract the posterior object from a product object

### Usage

```
extract_posterior(x)
```

### Arguments

x                      a product object

### Value

a posterior object

extract\_predictions      *Extract predictions*

### Description

Extract the marginal predictions over the prior

### Usage

```
extract_predictions(x)
```

### Arguments

x                      a product object

### Value

a prediction object

---

integral	<i>Compute integral</i>
----------	-------------------------

---

**Description**

Computes the definite integral of a product object over the range of the parameter

**Usage**

```
integral(obj)
```

**Arguments**

obj            a product object

**Value**

A numeric of the marginal likelihood

**Examples**

```
# define a likelihood
data_model <- likelihood(family = "normal", mean = 5.5, sd = 32.35)

# define a prior
prior_model <- prior(family = "normal", mean = 5.5, sd = 13.3)

# multiply the likelihood by the prior
model <- data_model * prior_model

# take the integral
integral(model)
```

---

likelihood	<i>Specify a likelihood</i>
------------	-----------------------------

---

**Description**

Define likelihoods using different different distribution families

**Usage**

```
likelihood(family, ...)
```

**Arguments**

family            the likelihood distribution (see details)  
...                see details

## Details

### Available distribution families:

The following distribution families can be used for the likelihood

- `normal` a normal distribution
- `student_t` a scaled and shifted t-distribution
- `noncentral_t` a noncentral t (for t statistic)
- `noncentral_d` a noncentral t (for one sample d)
- `noncentral_d2` a noncentral t (for independent samples d)
- `binomial` a binomial distribution

The parameters that need to be specified will be dependent on the family

### normal distribution:

When family is set to `normal` then the following parameters must be set

- `mean` mean of the normal likelihood
- `sd` standard deviation of the normal likelihood

### student\_t distribution:

When family is set to `student_t` then the following parameters may be set

- `mean` mean of the scaled and shifted t likelihood
- `sd` standard deviation of the scaled and shifted t likelihood
- `df` degrees of freedom

### noncentral\_t distribution:

When family is set to `noncentral_t` then the following parameters may be set

- `t` the t value of the data
- `df` degrees of freedom

### noncentral\_d distribution:

When family is set to `noncentral_d` then the following parameters may be set

- `d` the d (mean / sd) value of the data
- `n` the sample size

### noncentral\_d2 distribution:

When family is set to `noncentral_d2` then the following parameters may be set

- `d` the d (mean / `s_pooled`) value of the data
- `n1` the sample size of group 1
- `n2` the sample size of group 2

`s_pooled` is set as below:

$$s_{\text{pooled}} = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}}$$

### binomial distribution:

When the family is set to `binomial` then the following parameters may be set

- `successes` the number of successes
- `trials` the number of trials

**Value**

an object of class likelihood

**Examples**

```
# specify a normal likelihood
likelihood(family = "normal", mean = 5.5, sd = 32.35)

# specify a scaled and shifted t likelihood
likelihood(family = "student_t", mean = 5.5, sd = 32.35, df = 10)

# specify non-central t likelihood (t scaled)
likelihood(family = "noncentral_t", t = 10, df = 10)

# specify non-central t likelihood (d scaled)
likelihood(family = "noncentral_d", d = 10, n = 10)

# specify non-central t likelihood (independent samples d scaled)
likelihood(family = "noncentral_d2", d = 10, n1 = 10, n2 = 12)

# specify a binomial likelihood
likelihood(family = "binomial", successes = 2, trials = 10)
```

---

names,bayesplay-method

*Get names from data slot*

---

**Description**

Get names from data slot

**Usage**

```
## S4 method for signature 'bayesplay'
names(x)
```

**Arguments**

x                    a bayesplay object

**Value**

the field names from the data slot

---

plot	<i>Plot a bayesplay object</i>
------	--------------------------------

---

**Description**

Plots an object created by bayesplay

**Usage**

```
## S3 method for class 'prior'  
plot(x, ...)  
  
## S3 method for class 'posterior'  
plot(x, add_prior = FALSE, ...)  
  
## S3 method for class 'likelihood'  
plot(x, ...)  
  
## S3 method for class 'product'  
plot(x, ...)  
  
## S3 method for class 'prediction'  
plot(x, model_name = "model", ...)
```

**Arguments**

x	a prediction object
...	arguments passed to methods
add_prior	set to TRUE to add prior to the posterior plot
model_name	name of the model

**Value**

a ggplot2 object

---

prior	<i>Specify a prior</i>
-------	------------------------

---

**Description**

Define priors using different different distribution families

**Usage**

```
prior(family, ...)
```

**Arguments**

family	the prior distribution (see details)
...	see details

**Details****Available distribution families:**

The following distributions families can be used for the prior

- normal a normal distribution
- student\_t a scaled and shifted t-distribution
- cauchy a Cauchy distribution
- uniform a uniform distribution
- point a point
- beta a beta distribution The parameters that need to be specified will be dependent on the family

**Normal distribution:**

When family is set to normal then the following parameters may be set

- mean mean of the normal prior
- sd standard deviation of the normal prior
- range (optional) a vector specifying the parameter range

**Student t distribution:**

When family is set to student\_t then the following parameters may be set

- mean mean of the scaled and shifted t prior
- sd standard deviation of the scaled and shifted t prior
- df degrees of freedom of the scaled and shifted t prior
- range (optional) a vector specifying the parameter range

**Cauchy distribution:**

When family is set to cauchy then the following parameters may be set

- location the centre of the Cauchy distribution (default: 0)
- scale the scale of the Cauchy distribution
- range (optional) a vector specifying the parameter range

**Uniform distribution:**

When family is set to uniform then the following parameters must be set

- min the lower bound
- max the upper bound

**Point:**

When family is set to point then the following parameters may be set

- point the location of the point prior (default: 0)

**Beta:**

When family is set to beta then the following parameters may be set

- alpha the first shape parameter
- beta the second shape parameter

**Value**

an object of class prior

**Examples**

```
# specify a normal prior
prior(family = "normal", mean = 0, sd = 13.3)

# specify a half-normal (range 0 to Infinity) prior
prior(family = "normal", mean = 0, sd = 13.3, range = c(0, Inf))

# specify a student t prior
prior(family = "student_t", mean = 0, sd = 13.3, df = 79)

# specify a truncated t prior
prior(family = "student_t", mean = 0, sd = 13.3, df = 79, range = c(-40, 40))

# specify a cauchy prior
prior(family = "cauchy", location = 0, scale = .707)

# specify a half cauchy prior
prior(family = "cauchy", location = 0, scale = 1, range = c(-Inf, 0))

# specify a uniform prior
prior(family = "uniform", min = 0, max = 20)

# specify a point prior
prior(family = "point", point = 0)

# specify a beta prior
prior(family = "beta", alpha = 2.5, beta = 3.8)
```

---

sd\_ratio

*Compute the Savage-Dickey density ratio*

---

**Description**

Computes the Savage-Dickey density ratio from a posterior object at a specified point

**Usage**

```
sd_ratio(x, point)
```

**Arguments**

x	a posterior object
point	the point at which to evaluate the Savage-Dickey ratio

**Value**

A numeric of the Savage-Dickey density ratio

**Examples**

```
# define a likelihood
data_model <- likelihood(family = "normal", mean = 5.5, sd = 32.35)

# define a prior
prior_model <- prior(family = "normal", mean = 5.5, sd = 13.3)

model <- extract_posterior(data_model * prior_model)

# compute the Savage-Dickey density ratio at 0
sd_ratio(model, 0)
```

---

summary,bf-method	<i>Summarise a Bayes factor</i>
-------------------	---------------------------------

---

**Description**

Provide a verbal summary of a Bayes factor and the level of evidence

**Usage**

```
## S4 method for signature 'bf'
summary(object)
```

**Arguments**

object            a bf object

**Value**

No return, called for side effects

---

visual_compare	<i>Visually compare two models</i>
----------------	------------------------------------

---

**Description**

Visually compare two models

**Usage**

```
visual_compare(model1, model2, ratio = FALSE)
```

**Arguments**

model1            a predictive object  
model2            a predictive object  
ratio              show ratio rather than comparison (default: FALSE)

**Value**

A ggplot2 object

**Examples**

```
# define two models
data_model <- likelihood(family = "normal", .5, 1)
h0_mod <- prior(family = "point", point = 0)
h1_mod <- prior(family = "normal", mean = 0, sd = 10)
m0 <- extract_predictions(data_model * h0_mod)
m1 <- extract_predictions(data_model * h1_mod)

# visually compare the model
visual_compare(m0, m1)
# plot the ratio of the two model predictions
visual_compare(m0, m1, ratio = TRUE)
```

# Index

`extract_posterior`, [2](#)  
`extract_predictions`, [2](#)  
  
`integral`, [3](#)  
  
`likelihood`, [3](#)  
  
`names`, `bayesplay-method`, [5](#)  
  
`plot`, [6](#)  
`prior`, [6](#)  
  
`sd_ratio`, [8](#)  
`summary`, `bf-method`, [9](#)  
  
`visual_compare`, [9](#)