Package: bayesplay (via r-universe)

September 13, 2024

Type Package

Title The Bayes Factor Playground

Version 0.9.4

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 (>= 3.0.0), covr, knitr, rmarkdown, markdown, vdiffr

VignetteBuilder knitr

RoxygenNote 7.2.3

Imports methods, gginnards, stats, ggplot2, rlang, future.apply

URL `https://github.com/bayesplay/bayesplay`

BugReports `https://github.com/bayesplay/bayesplay/issues`

Collate 'bayesplay-package.R' 'classes.r' 'extract_functions.R' 'helper_functions.R' 'likelihoods.R' 'plots.R' 'priors.r' 'utils.r' 'robustness.r'

Config/testthat/edition 3

Depends R (>= 4.1.0)

Repository https://bayesplay.r-universe.dev

RemoteUrl https://github.com/bayesplay/bayesplay

RemoteRef HEAD

RemoteSha af830a81edc1e988e7742fcf52c7d4189947e625

Contents

bfrr 2
extract_posterior 3
extract_predictions 4
integral 4
likelihood 5
names,bayesplay-method 7
plot 7
prior 8
sd_ratio 10
summary,bf-method 11
summary,robustness-method 11
visual_compare 12

Index **13**

`bfrr` *Perform a robustness analysis*

Description

Perform a robustness analysis by systematically varying the values of the alternative prior

Usage

```
bfrr(
  likelihood,
  alternative_prior,
  null_prior,
  parameters,
  steps = 100L,
  cutoff = 3L,
  multicore = TRUE
)
```

Arguments

`likelihood`	`likelihood` object representing the data
`alternative_prior`	`prior` object for the alternative prior
`null_prior`	`prior` object for the null prior
`parameters`	`list` of min and max values for each parameter to vary over.
`steps`	`numeric` the number of steps each parameter is varied over. (Default: 100 steps)
`cutoff`	Minimum Bayes factor value for evidence for the hypothesis (Default: 3)
`multicore`	Run robustness analysis across multiple cores (Default: TRUE if available)

Value

A `robustness` object

Examples

```
# define a likelihood
data_model <- likelihood(family = "normal", mean = 1, sd = 5)

# define the alternative prior
alternative_prior <- prior(family = "normal", mean = 5, sd = 4)

# define the null prior
null_prior <- prior(family = "point", point = 0L)

# set the parameters to vary and the range to vary them across
# vary the mean from -10 to 5
# vary the sd from 1 to 5
parameters <- list(mean = c(-10, 5), sd = c(1, 5))

# mark all Bayes factors larger/smaller than 3/.3 as evidence for the
# alternative / null
cutoff <- 3
bfrr(data_model, alternative_prior, null_prior, parameters, steps = 10,
  cutoff,
  multicore = FALSE
)
```

`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` *Compute integral*

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` *Specify a likelihood*

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` *Plot a bayesplay object*

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", ...)

## S3 method for class 'robustness'
plot(x, ...)
```

Arguments

`x`	a `robustness` 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` *Specify a prior*

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 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 Saveage-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` *Summarise a Bayes factor*

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

`summary,robustness-method` *Summarise a Robustness analysis*

Description

Provide a summary of a robustness analysis

Usage

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

Arguments

`object`	a `robustness` object

Value

No return, called for side effects

`visual_compare` *Visually compare two models*

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

bfrr, 2

extract_posterior, 3
extract_predictions, 4

integral, 4

likelihood, 5

names,bayesplay-method, 7

plot, 7
prior, 8

sd_ratio, 10
summary,bf-method, 11
summary,robustness-method, 11

visual_compare, 12