Package: BrownianMotion (via r-universe)

November 5, 2024

Type Package

Title Brownian Motion Simulations

Version 0.3.0

Description An implementation of algorithms for simulation of (multivariate) Brownian motion trajectories, including simple unconditional simulation, Brownian bridges, as well as layered and localised Brownian motion simulators.

License GPL-2

Depends R (>= 4.1.0),

Imports checkmate, dplyr, ggplot2, glue, graphics, grDevices, patchwork, rlang, Rmpfr, statmod, stats, tibble, utils

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

Suggests knitr, rmarkdown, sf

VignetteBuilder knitr

Config/pak/sysreqs libgmp3-dev libmpfr-dev

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

RemoteUrl https://github.com/louisaslett/BrownianMotion

RemoteRef HEAD

RemoteSha 75fe105b4e01da33bcfb73e6bffe484f704c8f5a

Contents

`add.labels` *Manually add labels after simulation*

Description

Description of adding labels

Usage

```
add.labels(bm, t, label = names(t))
```

Arguments

`bm`	a Brownian motion object from which simulation should continue. Note the object is updated in place
`t`	vector of fixed times of the Brownian motion.
`label`	vector of length 1 or the length of t indicating the introduction of user specified labels (if any). By default, uses any names in the vector of times.

`add.points` *Manually specify the value of a point in the skeleton*

Description

Note: Only valid for time points that don't lie within existing layers. Cannot be an existing time within the skeleton (in this case you should delete then add the point).

Usage

```
add.points(bm, t, W_t, label = names(t))
```

Arguments

bm	a Brownian motion object from which simulation should continue. Note the object is updated in place
t	vector of fixed times of the Brownian motion. Defaults to value of 0.
W_t	matrix of size the length of t (rows) x dimension (column). If dim = 1, a vector of the length of t is acceptable. If dim > 1 and the length of t = 1, then a vector of length 1 is acceptable (corresponding to the same value at every co-ordinate), or length d is acceptable (corresponding to the d components). Defaults to 0 for every component.
label	vector of length 1 or the length of t indicating the introduction of user specified labels (if any). By default, uses any names in the vector of times.

---

add.segment *Create / move a segment*

Description

Create / move a segment

Usage

```
add.segment(bm, l = -Inf, r = Inf, W_t = NULL, delta = NULL, label = NULL)
```

Arguments

bm	a Brownian motion object from which simulation should continue. Note the object is updated in place
l	closed left end of interval to add a segment. By default, '-Inf' which results in the introduction of a segment by shifting the entire path up to 'r'
r	open right end of interval to add a segment. By default, 'Inf' which results in the introduction of a segment by shifting the entire path from 'l'
W_t	vector of length the dimension of the Brownian motion specifying the coordinate of the Brownian motion at the time l (specified by the user). A vector of length 1 is acceptable (corresponding to the same value at every co-ordinate). Defaults to NULL, meaning this argument is ignored in favor of delta.
delta	vector of length the dimension of the Brownian motion specifying the shift of the Brownian motion at the time l (specified by the user). A vector of length 1 is acceptable (corresponding to the same shift for every co-ordinate). Defaults to NULL, meaning this argument is ignored in favor of W_t.
label	vector of length 1 or the length of t indicating the introduction of user specified labels (if any). By default, uses any names in the vector of times.

concat.bm *Copy / Concatenate Brownian motion objects*

Description

Copy / Concatenate Brownian motion objects

Usage

```
concat.bm(..., t0 = NULL)
```

Arguments

...	multiple Brownian motion objects which will be concatenated together in the order provided into a new BrownianMotion object. Note the object is updated in place
t0	scalar representing the time the new Brownian motion should be initialised. Defaults to value of initial time of the first Brownian motion object to be concatenated.

create.bm *Create a Brownian motion object*

Description

Creates an R environment to contain the trajectories and layer information of a Brownian motion.

Usage

```
create.bm(
  t = 0,
  W_t = 0,
  dim = 1,
  cov = 1,
  refine = TRUE,
  mult = 1,
  prefer = "bessel"
)
```

Arguments

t	vector of fixed times of the Brownian motion. Defaults to value of 0.
W_t	matrix of size the length of t (rows) x dimension (column). If dim = 1, a vector of the length of t is acceptable. If dim > 1 and the length of t = 1, then a vector of length 1 is acceptable (corresponding to the same value at every co-ordinate), or length d is acceptable (corresponding to the d components). Defaults to 0 for every component.
dim	scalar corresponding to the dimension of the Brownian motion.
cov	covariance matrix of the Brownian motion. If scalar corresponds to a multiple of the identity matrix. If matrix should be of size dim x dim.
refine	indicates whether refinement should be by default carried out. If of length 1, then this specifies refinement for every dimension. If of length dim, then this specifies refinement for each dimension (in order).
mult	set the default mult parameter to be used for any layer operations on this path, including for the level of refinement if refine=TRUE.
prefer	indicates whether there is a preference for "bessel" or "intersection" layers where possible. If of length 1, then this specifies the preference for every dimension. If of length dim, then this specifies preference for each dimension (in order). create.bm(....., refine = TRUE) create.bm(....., refine = FALSE) create.bm(....., refine = list(spatial = TRUE, temporal = FALSE))

---

delete.labels *Manually delete labels after simulation*

Description

Description of deleting labels

Usage

```
delete.labels(bm, t = NA, label = NA)
```

Arguments

bm	a Brownian motion object from which simulation should continue. Note the object is updated in place
t	vector of fixed times of the Brownian motion.
label	vector of length 1 or the length of t indicating the introduction of user specified labels (if any). By default, uses any names in the vector of times.

delete.skeleton *Eliminate parts of path, layer or both from the skeleton*

Description

Eliminate parts of path, layer or both from the skeleton

Usage

```
delete.skeleton(bm, l = -Inf, r = Inf, type = "all")
```

Arguments

bm	a Brownian motion object from which simulation should continue. Note the object is updated in place
l	closed left end of interval to delete. By default, '-Inf' which results in left truncation of the path up to 'r'
r	open right end of interval to delete. By default, 'Inf' which results in right truncation of the path from 'l'
type	one of '"all"' or '"layer"' to specify whether to delete both path observations and layers, or just layers in the open interval '(l,r)'

first.passage *Find a first passage time and state*

Description

Finds the first passage time of a Brownian motion object to a specified lower, upper, or lower and upper limit.

Usage

```
first.passage(
  bm,
  l = NULL,
  u = NULL,
  delta.l = NULL,
  delta.u = NULL,
  delta = NULL,
  label = NULL
)
```

Arguments

bm	a Brownian motion object (of dimension 1) from which simulation should continue. Note the object is updated in place.
l	scalar giving lower bound for first passage.
u	scalar giving upper bound for first passage.
delta.l	provide bounds via the negative change in Brownian motion. May be specified together with delta.u to provide asymmetric first passage delta.
delta.u	provide bounds via the positive change in Brownian motion. May be specified together with delta.l to provide asymmetric first passage delta.
delta	scalar providing the bounds by way of the positive/negative change in the Brownian motion.
label	a user specified label for the first passage time created by this layer (optional).

Details

This function updates a Brownian motion object adding the first passage time to the specified lower (l), upper (u), or lower and upper limits. The limits can also be specfied via delta which places the limits as the current endpoint state +/- delta. Note that simulation of the first passage implicitly imposes constraints on the underlying Brownian motion between the endpoint and the first passage time.

In particular, in the case that the lower and upper limits are asymmetric around the endpoint, there may be a recursive simulation of nested symmetric first passages between the endpoint and the ultimate first passage, resulting in implicit imposition of nested constraints. When plotting a Brownian motion these are visible as dashed red lines nested within the solid red line constraint imposed by the overall first passage simulation.

Value

The Brownian motion object which was passed in argument bm is updated in place and returned, enabling chaining of commands with dplyr (and other) style pipes.

---

layers *Construct new layers*

Description

TODO UPDATE ME Performs unbiased simulation of the smallest specified layers which contain the Brownian motion between two known sample points by retrospective Bernoulli sampling and inversion sampling.

Usage

```
layers(
  bm,
  s,
  t,
  type = bm$prefer,
  refine = bm$refine,
  mult = bm$mult,
  prefer = bm$prefer,
  label = c(names(s), names(t))
)
```

Arguments

bm	a Brownian motion object from which simulation should continue. Note the object is updated in place
s	left hand time point
t	right hand time point
type	type specifies whether to carry out "bessel" or "intersection" or "localised" simulation. If of length 1, then this specifies the type for every dimension. If of length dim, then this specifies for each dimension the type. By default this matches the setting for preference specified on the creation of the Brownian motion object.
refine	indicates whether refinement should be by default carried out. If of length 1, then this specifies refinement for every dimension. If of length dim, then this specifies refinement for each dimension (in order). By default uses the setting on creation of the Brownian motion object.
mult	the default layer size is `sqrt(t-s)`. You can scale this by specifying the `mult` argument which will result in layer sizes of `mult*sqrt(t-s)`. By default uses the setting on creation of the Brownian motion object.
prefer	indicates whether there is a preference for "bessel" or "intersection" layers where possible. If of length 1, then this specifies the preference for every dimension. If of length dim, then this specifies preference for each dimension (in order). By default uses the setting on creation of the Brownian motion object.
label	vector of length 1 or the length of t indicating the introduction of user specified labels (if any). By default, uses any names in the vector of times.

Value

The Brownian motion object which was passed in argument `bm` is updated in place and returned, enabling chaining of commands with dplyr (and other) style pipes.

`refine` *Refine an intersection layer*

Description

Refine an intersection layer

Usage

```
refine(bm, s, t, mult = bm$mult)
```

Arguments

`bm`	a Brownian motion object from which simulation should continue. Note the object is updated in place
`s`	left hand time point
`t`	right hand time point
`mult`	the default layer size is `sqrt(t-s)`. You can scale this by specifying the `mult` argument which will result in layer sizes of `mult*sqrt(t-s)`. By default uses the setting on creation of the Brownian motion object.

`sim` *Simulate Brownian motion at a target time*

Description

Simulates Brownian motion at chosen times, automatically applying the correct algorithm to condition on the current known (ie previously simulated) states of the trajectory.

Usage

```
sim(
  bm,
  t,
  refine = bm$refine,
  mult = bm$mult,
  prefer = bm$prefer,
  label = names(t)
)
```

Arguments

bm	a Brownian motion object from which simulation should continue. Note the object is updated in place.
t	a vector of times to simulate at.
refine	whether to automatically refine layers where the simulated time results in bisection of an existing layer. Defaults to the option specified at the time of creation of bm.
mult	the mult parameter to be passed through to any layer operations including for the level of refinement if refine=TRUE. Defaults to the option specified at the time of creation of bm.
prefer	indicate the preferred layer type which should arise after conditional simulation bisects an existing layer (this will be respected where possible, but it not always achievable). Defaults to the option specified at the time of creation of bm.
label	vector of length 1 or the length of t indicating the introduction of user specified labels (if any). By default, uses any names in the vector of times.

Value

the Brownian motion object which was passed in argument bm is updated in place and returned, enabling chaining of commands with dplyr (and other) style pipes.

Index