Package ‘uniformly’

July 31, 2018

Type Package
Title Uniform Sampling
Version 0.1.0
Date 2018-07-31
Author Stéphane Laurent
Maintainer Stéphane Laurent <laurent_step@yahoo.fr>
Description Uniform sampling on various geometric shapes, such as spheres, ellipsoids, simplices.
License GPL-3
Encoding UTF-8
LazyData true
URL https://github.com/stla/uniformly
BugReports https://github.com/stla/uniformly/issues
RoxygenNote 6.0.1
Imports stats, pgnorm
Suggests knitr, rmarkdown, geometry, rgl, misc3d, scatterplot3d
VignetteBuilder knitr
NeedsCompilation no
Repository CRAN
Date/Publication 2018-07-31 12:10:03 UTC

R topics documented:

  rphong_on_hemisphere ................................................. 2
  runif_cube ............................................................. 3
  runif_ellipsoid ......................................................... 3
  runif_in_annulus ....................................................... 4
  runif_in_pball ......................................................... 5
  runif_in_polygon ....................................................... 6
  runif_in_simplex ....................................................... 7
rphong_on_hemisphere

Sampling on Hemisphere

rphong_on_hemisphere

Description

Sampling on a hemisphere according to the Phong density (dimension 3).

Usage

```
    rphong_on_hemisphere(n, alpha = 0, r = 1)
```

Arguments

- **n**: number of simulations
- **alpha**: parameter of the Phong density, a positive number; 0 for uniform sampling (default)
- **r**: radius

Value

The simulations in a \( n \times 3 \) matrix.

Examples

```
    ## Not run: library(rgl)
    sims <- rphong_on_hemisphere(400, alpha=10)
    spheres3d(0, 0, 0, color="red", alpha=0.5)
    points3d(sims)
    ## End(Not run)
```
runif_cube

---

**runif_cube**  
*Uniform sampling on/in cube*

**Description**  
Uniform sampling on or in a cube (arbitrary dimension).

**Usage**  
```r
runif_in_cube(n, d, O = rep(0, d), r = 1)
runif_on_cube(n, d, O = rep(0, d), r = 1)
```

**Arguments**  
- `n`: number of simulations
- `d`: dimension
- `O`: center of the cube
- `r`: radius (half-side) of the cube

**Value**  
The simulations in an `n` times `d` matrix.

**Examples**  
```r
sims <- runif_on_cube(60, d=2)
plot(sims, xlim=c(-1,1), ylim=c(-1,1), pch=19, asp=1)
sims <- runif_in_cube(50, d=3)
library(scatterplot3d)
scatterplot3d(sims, pch=19, highlight.3d=TRUE, asp=1)
```

---

**runif_ellipsoid**  
*Uniform sampling on/in ellipsoid*

**Description**  
Uniform sampling on an ellipsoid or in an ellipsoid (arbitrary dimension).

**Usage**  
```r
runif_on_ellipsoid(n, A, r)
runif_in_ellipsoid(n, A, r)
```
runif_in_annulus

Arguments

- **n**: number of simulations
- **A**: symmetric positive-definite matrix defining the ellipsoid (see Details)
- **r**: "radius" (see Details)

Details

The ellipsoid is the set of vectors \( x \) satisfying \( t(x) A t(x) = r^2 \).

Value

The simulations in a matrix with \( n \) rows.

Examples

```r
A <- rbind(c(2,1),c(1,1))
r <- 2
sims <- runif_on_ellipsoid(30, A, r)
plot(sims, ylim=c(-2,2), ylim=c(-3,3), asp=1, pch=19)
sims <- runif_in_ellipsoid(100, A, r)
plot(sims, ylim=c(-2,2), ylim=c(-3,3), asp=1, pch=19)
```

# 3D example
```r
A <- matrix(c(5,1,1, 1,3,1, 1,1,1), ncol=3)
r <- 2
# draw the ellipsoid
library(misc3d)
x <- seq(-1, 1, len=50)
y <- seq(-1, 1, len=50)
z <- seq(-1, 1, len=50)
g <- as.matrix(expand.grid(x=x, y=y, z=z))
voxel <- array(apply(g, 1, function(v) t(v) A t(v)), dim=c(50,50,50))
isosurface <- computeContour3d(voxel, max(voxel), r^2, x=x, y=y, z=z)
drawScene3d(makeTriangles(isosurface, alpha=0.3))
```

# simulate and plot points on ellipsoid
```r
library(rgl)
sims <- runif_on_ellipsoid(200, A, r)
points3d(sims)
```

runif_in_annulus  
**Uniform sampling in an annulus**

Description

Uniform sampling in an annulus (dimension 2).

Usage

```r
runif_in_annulus(n, a, r)
```
runif_in_pball

Arguments

- `n`: number of simulations
- `o`: center of the annulus
- `r1`: inner radius
- `r2`: outer radius

Value

The simulations in a \( n \times 2 \) matrix.

Examples

```r
sims <- runif_in_annulus(100, c(0,0), 1, 2)
plot(sims, xlim = c(-2,2), ylim = c(-2,2), asp = 1, pch = 19)
```

---

runif_in_pball

Uniform sampling in a p-ball

Description

Uniform sampling in a p-ball (arbitrary dimension).

Usage

```r
runif_in_pball(n, d, p, r = 1)
```

Arguments

- `n`: number of simulations
- `d`: dimension
- `p`: exponent in the p-norm, a positive number
- `r`: positive number, the radius

Value

The simulations in a \( n \times d \) matrix.

Examples

```r
sims <- runif_in_pball(500, d=2, p=1)
plot(sims, xlim=c(-1,1), ylim=c(-1,1), asp=1)
```
runif_in_polygon  Uniform sampling in a polygon

Description
Uniform sampling in a polygon (dimension 2).

Usage
runif_in_polygon(n, vertices, center = "centroid")

Arguments
n  number of simulations
vertices  two-columns matrix giving the vertices (rows); the vertices must be ordered (clockwise or counterclockwise)
center  a point with respect to which the polygon is star-shaped, or "centroid" (default) to take the centroid (see Details)

Details
This function works for a star-shaped polygon, that is, a polygon that contains a point from which the entire polygon boundary is visible. This point must be given in the center argument. If the polygon is convex, any point inside the polygon is suitable (thus the default option of the center argument is appropriate in this case).

Value
The simulations in a n times 2 matrix.

Examples
vs <- matrix(c(0.951056516295154, 0.309016994374947, 0.224513988289793, 0.309016994374947, -0.951056516295154, 0.309016994374948, -0.363271264002681, -0.118033988749895, 0.587785252292473, -0.809016994374948, -0.36327126400268, -0.118033988749895, 0, 1, 0, 1, -0.224513988289793, 0.309016994374947, -0.587785252292473, -0.809016994374947, 0, -0.381966011250105), ncol=2, byrow=TRUE)
sims <- runif_in_polygon(500, vs)
plot(sims, xlim=c(-1,1), ylim=c(-1,1), pch=19, asp=1)
runif_in_simplex  

Uniform sampling in a simplex

Description
Uniform sampling in a simplex (arbitrary dimension).

Usage
runif_in_simplex(n, simplex)

Arguments

n  number of simulations
simplex  a (d+1) times d matrix giving the vertices of the simplex (rows)

Value
The simulations in a n times d matrix.

Note
In dimension 3, you can use runif_in_tetrahedron instead.

Examples

simplex <- rbind(c(0,0,0), c(1,0,0), c(1,1,0), c(1,1,2))
sims <- runif_in_simplex(1000, simplex)
library(rgl)
points3d(sims)

runif_in_tetrahedron  

Uniform sampling in a tetrahedron

Description
Uniform sampling in a tetrahedron (in dimension 3).

Usage
runif_in_tetrahedron(n, v1, v2, v3, v4)

Arguments

n  number of simulations
v1, v2, v3, v4  vertices of the tetrahedron
runif_on_spherePatch

Uniform sampling on a spherical patch

Description

Uniform sampling on a spherical patch (in dimension 3).

Usage

runif_on_spherePatch(n, r = 1, phi1, phi2, theta1, theta2)

Arguments

- `n`: number of simulations
- `r`: radius
- `phi1, phi2`: numbers defining the latitudinal angle range
- `theta1, theta2`: numbers defining the longitudinal angle range

Details

A sphere patch is the part of the sphere whose polar angles theta and phi satisfy $0 \leq \theta_1 \leq \theta \leq \theta_2 \leq 2\pi$ and $0 \leq \phi_1 \leq \phi \leq \phi_2 \leq \pi$.

Value

The simulations in a $n \times 3$ matrix.

See Also

- `runif_on_stri` for sampling on a spherical triangle.
runif_on_stri

**Examples**

```r
# sampling on the first orthant:
sims <- runif_on_spherePatch(100, phi1=0, phi2=pi/2, theta1=0, theta2=pi/2)
## Not run: library(rgl)
spheres3d(0, 0, 0, color="red", alpha=0.5)
points3d(sims)
## End(Not run)
```

---

**runif_on_stri**  
*Uniform sampling on a spherical triangle*

**Description**

Uniform sampling on a spherical triangle (in dimension 3).

**Usage**

```r
runif_on_stri(n, r = 1, v1, v2, v3)
```

**Arguments**

- `n`: number of simulations
- `r`: radius
- `v1, v2, v3`: vertices

**Value**

The simulations in a n times 3 matrix.

**Examples**

```r
# sampling on the first orthant:
sims <- runif_on_stri(100, v1=c(1,0,0), v2=c(0,1,0), v3=c(0,0,1))
## Not run: library(rgl)
spheres3d(0, 0, 0, color="red", alpha=0.5)
points3d(sims)
## End(Not run)
```
### runif_sphere

**Description**

Uniform sampling on a sphere or in a sphere, in arbitrary dimension.

**Usage**

```r
runif_on_sphere(n, d, r = 1)
runif_in_sphere(n, d, r = 1)
```

**Arguments**

- `n`: number of simulations
- `d`: dimension of the space
- `r`: radius of the sphere

**Value**

The simulations in a `n` times `d` matrix.

**Examples**

```r
sims <- runif_on_sphere(20, d=2)
pplot(sims, xlim=c(-1,1), ylim=c(-1,1), asp=1, pch=19)
sims <- runif_in_sphere(100, d=2)
pplot(sims, xlim=c(-1,1), ylim=c(-1,1), asp=1, pch=19)
```

### runif_triangle

**Description**

Uniform sampling on or in a triangle (dimension 2).

**Usage**

```r
runif_in_triangle(n, v1, v2, v3)
runif_on_triangle(n, v1, v2, v3)
```
runif_unitSimplex

Arguments

- **n**: number of simulations
- **v1, vR, vS**: vertices of the triangle

Value

The simulations in a \( n \times 2 \) matrix.

Examples

```r
sims <- runif_on_triangle(SP, c(P,P), c(1,P), c(P,1))
plot(sims, xlim=c(P,1), ylim=c(P,1), pch=19)
sims <- runif_in_triangle(1PP, c(P,P), c(1,P), c(P,1))
plot(sims, xlim=c(P,1), ylim=c(P,1), pch=19)
```

runif_unitSimplex  
Uniform sampling on/in a unit simplex

Description

Uniform sampling on or in a unit simplex (arbitrary dimension).

Usage

```r
runif_on_unitSimplex(n, d)
runif_in_unitSimplex(n, d)
```

Arguments

- **n**: number of simulations
- **d**: dimension of the space

Value

The simulations in a \( n \times d \) matrix.

See Also

- `runif_in_tetrahedron` for sampling in an arbitrary tetrahedron in dimension 3; `runif_in_simplex` for sampling in an arbitrary simplex.

Examples

```r
library(rgl)
sims <- runif_on_unitSimplex(300, d=3)
points3d(sims)
```
**surface_sphere**  
*Sphere surface*

**Description**  
Surface of a d-dimensional sphere.

**Usage**  
`surface_sphere(d, r = 1)`

**Arguments**  
- `d`: dimension of the space  
- `r`: radius of the sphere

**Value**  
The surface of the sphere of radius `r` in the d-dimensional space.

**Examples**
```r  
r <- 2  
surface_sphere(3, r)  
4*pi*r^2  
# perimeter of the unit circle:  
surface_sphere(2)```

---

**surface_spherePatch**  
*Sphere patch surface*

**Description**  
Surface of a sphere patch.

**Usage**  
`surface_spherePatch(r, phi1, phi2, theta1, theta2)`

**Arguments**  
- `r`: radius  
- `phi1`, `phi2`: numbers defining the latitudinal angle range  
- `theta1`, `theta2`: numbers defining the longitudinal angle range
Details

A sphere patch is the part of the sphere whose polar angles theta and phi satisfy \(0 \leq \theta_1 \leq \theta \leq \theta_2 \leq 2\pi\) and \(0 \leq \phi_1 \leq \phi \leq \phi_2 \leq \pi\).

Value

The surface of the sphere patch.

See Also

surface_stri for the surface of a spherical triangle.

Examples

C surface of the first orthant:
surface_spherePatch(r=1, phi1=0, phi2=pi/2, theta1=0, theta2=pi/2)
surface_stri(r=1, c(1,0,0), c(0,1,0), c(0,0,1))
surface_triangle  Triangle surface

Description

Surface of a triangle.

Usage

surface_triangle(v1, v2, v3)

Arguments

v1, v2, v3  vertices of the triangle

Value

The surface of the triangle with vertices v1, v2, v3.

Examples

surface_triangle(c(0,0), c(0,1), c(1,0))

volume_ellipsoid  Ellipsoid volume

Description

Volume of an ellipsoid (arbitrary dimension).

Usage

volume_ellipsoid(A, r)

Arguments

A  symmetric positive-definite matrix defining the ellipsoid (see Details)
r  "radius" (see Details)

Details

The (boundary of the) ellipsoid is the set of vectors x satisfying t(x) %*% A %*% x == r^2.

Value

The volume of the ellipsoid.
### Examples

```r
# dimension 2 (area), with diagonal matrix A
A <- diag(c(2,3))
r <- 2
volume_ellipsoid(A, r)
pi * r^2 / sqrt(A[1,1]*A[2,2])
```

---

**volume_pball**  
*p-ball volume*

---

**Description**

Euclidean volume of a p-ball (arbitrary dimension).

**Usage**

```r
volume_pball(d, p, r = 1)
```

**Arguments**

- `d`: dimension
- `p`: exponent in the p-norm, a positive number
- `r`: radius of the ball

**Value**

The volume of the p-ball with radius `r`.

**Examples**

```r
volume_pball(d=4, p=2, r=2)
volume_sphere(d=4, r=2)
```

---

**volume_simplex**  
*Simplex volume*

---

**Description**

Volume of a simplex (arbitrary dimension).

**Usage**

```r
volume_simplex(simplex)
```
Arguments

simplex  a (d+1) times d matrix giving the vertices of the simplex (rows)

Value

The volume of the simplex.

Examples

set.seed(666)
simplex <- matrix(rnorm(4*3), nrow=4, ncol=3)
volume_simplex(simplex)
volume_tetrahedron(simplex[1,], simplex[2,], simplex[3,], simplex[4,])

dimension of the space
radius of the sphere

Value

The volume of the sphere with radius r in the d-dimensional space.

Examples

r <- 2
volume_sphere(3, r)
4/3*pi*r^3
### volume_tetrahedron

**Tetrahedron volume**

**Description**

Volume of a tetrahedron (dimension 3).

**Usage**

```r
volume_tetrahedron(v1, vR, vS, vT)
```

**Arguments**

- `v1`, `vR`, `vS`, `vT` vertices of the tetrahedron

**Value**

The volume of the tetrahedron.

**See Also**

- `volume_simplex` for the volume of a simplex in arbitrary dimension.

**Examples**

```r
v1 <- c(0,0,0); v2 <- c(1,0,0); v3 <- c(0,1,0); v4 <- c(0,0,1)
volume_tetrahedron(v1, v2, v3, v4)
volume_unitsimplex(3)
```

---

### volume_unitsimplex

**Unit simplex volume**

**Description**

Volume of the unit simplex (arbitrary dimension).

**Usage**

```r
volume_unitsimplex(d)
```

**Arguments**

- `d` dimension of the space

**Value**

The volume of the unit simplex in the space of dimension `d`. 
See Also

volume_simplex for the volume of an arbitrary simplex.
Index

rphong_on_hemisphere, 2
runif_cube, 3
runif_ellipsoid, 3
runif_in_annulus, 4
runif_in_cube (runif_cube), 3
runif_in_ellipsoid (runif_ellipsoid), 3
runif_in_pball, 5
runif_in_polygon, 6
runif_in_simplex, 7, 8, 11
runif_in_sphere (runif_sphere), 10
runif_in_tetrahedron, 7, 7, 11
runif_in_triangle (runif_triangle), 10
runif_in_unitSimplex
  (runif_unitSimplex), 11
runif_on_cube (runif_cube), 3
runif_on_ellipsoid (runif_ellipsoid), 3
runif_on_sphere (runif_sphere), 10
runif_on_spherePatch, 8
runif_on_stri, 8, 9
runif_on_triangle (runif_triangle), 10
runif_on_unitSimplex
  (runif_unitSimplex), 11
runif_sphere, 10
runif_triangle, 10
runif_unitSimplex, 11

surface_sphere, 12
surface_spherePatch, 12
surface_stri, 13, 13
surface_triangle, 14

volume_ellipsoid, 14
volume_pball, 15
volume_simplex, 15, 17, 18
volume_sphere, 16
volume_tetrahedron, 17
volume_unitSimplex, 17