Package: isocubes (via r-universe)

November 29, 2024

Type Package

Title Voxel Rendering with Isometric Cubes

Version 0.1.5.9009

Maintainer Mike Cheng <mikefc@coolbutuseless.com>

Description Visualising three-dimensional voxel data.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Imports grid

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

URL https://github.com/coolbutuseless/isocubes

BugReports https://github.com/coolbutuseless/isocubes/issues

VignetteBuilder knitr

Depends R (>= 2.10)

LazyData true

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

RemoteUrl https://github.com/coolbutuseless/isocubes

RemoteRef HEAD

RemoteSha 1a7cb3d4cc93679e2ef2136a793ccd18ea254b2f

Contents

`coords_heightmap` *Calculate isocubes coordinates from a height matrix*

Description

Calculate isocubes coordinates from a height matrix

Usage

```
coords_heightmap(
  mat,
  fill = NULL,
  scale = 1,
  flipx = FALSE,
  flipy = TRUE,
  ground = "xz",
  solid = TRUE,
  check_visibility = FALSE,
  verbose = FALSE
)
```

Arguments

`mat`	integer matrix. The matrix will be interpreted as cubes flat on the page, with the value in the matrix interpreted as the height above the page.
`fill`	matrix of colours the same dimensions as the `mat` argument. Default: NULL. If `fill` is not NULL, then a `fill` column will be included in the final returned coordinates.
`scale`	scale factor for values in matrix. Default = 1
`flipx`, `flipy`	Should the matrix be flipped in the horizontal/vertical directions (respectively)? Default: `flipx` = `FALSE`, `flipy` = `TRUE`. Note: `flipy` defaults to `TRUE` as matrices are indexed from the top-down, but the isometric coordinate space is increasing from the bottom up. Flipping the matrix vertically is usually what you want.
`ground`	Orientation of the ground plane. Default: 'xz'. Possible values 'xz', 'xy'
`solid`	Should the heightmap be made 'solid' i.e. without holes? default: TRUE. This can be an expensive operation in terms of both memory and CPU, but should be OK for simple examples. Set to FALSE if things take too long. This operation works by extruding cubes down from the top of the height map to the floor to ensure gaps do not appear when the slope is too great.
`check_visibility`	Should non-visible cubes be removed? Default: FALSE. If you plan on rotating or manipulating the returned coordinates then this should definitely by FALSE. If TRUE, then non-visible voxels will be entirely removed from the returned

	coordinates i.e. they will be missing if you change the rendering viewpoint from the default.
verbose	Be verbose? default: FALSE

Value

data.frame of isocube coordinates

Examples

```
# Plot the standard volcano
mat <- volcano
mat[seq(nrow(mat)),] <- mat[rev(seq(nrow(mat))),]
val <- as.vector(mat)
val <- round(255 * (val - min(val)) / diff(range(val)))
fill <- matrix("", nrow=nrow(mat), ncol=ncol(mat))
fill[] <- terrain.colors(256)[val + 1L]

coords <- coords_heightmap(mat - min(mat), fill = fill, scale = 0.3)
cubes  <- isocubesGrob(coords, size = 2)
grid::grid.draw(cubes)
```

---

isocubesGrob *Create a grob of isocubes*

Description

Create a grob of isocubes

Usage

```
isocubesGrob(
  coords,
  fill = NULL,
  fill_left = NULL,
  fill_right = NULL,
  intensity = c(1, 0.3, 0.7),
  size = 5,
  x = NULL,
  y = NULL,
  col = "black",
  default.units = "npc",
  default.units.cube = "mm",
  xyplane = "right",
  handedness = "left",
  verbosity = 0,
  ...
)
```

Arguments

coords	data.frame of x,y,z coordinates for the cubes (integer coordinates)
fill	fill colour for the top face of cube. Default: NULL will attempt to use the 'fill' colour in the coords data.frame, otherwise 'grey50'
fill_left, fill_right	fill colours for left and fight faces of cube.
intensity	c(1, 0.3, 0.6) Intensity shading for fill for the top, left and right faces respectively. Note: this setting has no effect on the shading of the left face if fill_left has been set explicitly by the user; same for the right face.
size	dimensions of cube i.e. the length of the vertical edge of the cube. Default: 5mm
x, y	the origin of the isometric coordinate system in 'snpc' coordinates. These values should be given as vanilla floating point values. Be default the origin is the middle bottom of the graphics device i.e. (x, y) = (0.5, 0)
col	Stroke colour for outline of cube faces. Default: black. If NA then no outlines will be drawn. If negative, then outline colour will be the same as the face colour.
default.units	Default unit for (x,y) position is 'npc'
default.units.cube	Default unit for size of a cube is 'mm'
xyplane	How is the xyplane oriented with respect to the unit isometric cube?. "left", "right", "top" (or "flat"). Default: "right"
handedness	How is the z-axis positioned with respect to the xy-plane? I.e. is this a right-handed or left-handed coordinate system? Default: "left"
verbosity	Verbosity level. Default: 0
...	other values passed to gpar() to set the graphical parameters e.g. lwd and col for the linewidth and colour of the outline stroke for each cube face.

Value

grid grob object

Examples

```
coords <- sphere_coords
fill <- rainbow(nrow(coords))
iso <- isocubesGrob(coords, fill = fill, size = 2)
grid::grid.draw(iso)

coords <- organic_coords
iso <- isocubesGrob(coords, size = 2)
grid::grid.newpage()
grid::grid.draw(iso)
```

isolinesGrob *Isometrix grid of lines*

Description

Isometrix grid of lines

Usage

```
isolinesGrob(
  N = 50,
  size = 5,
  x = NULL,
  y = NULL,
  col = "black",
  default.units = "npc",
  default.units.cube = "mm",
  verbosity = 0,
  ...
)
```

Arguments

N	extents
size	dimensions of cube i.e. the length of the vertical edge of the cube. Default: 5mm
x, y	the origin of the isometric coordinate system in 'snpc' coordinates. These values should be given as vanilla floating point values. Be default the origin is the middle bottom of the graphics device i.e. (x, y) = (0.5, 0)
col	Stroke colour for outline of cube faces. Default: black. If NA then no outlines will be drawn. If negative, then outline colour will be the same as the face colour.
default.units	Default unit for (x,y) position is 'npc'
default.units.cube	Default unit for size of a cube is 'mm'
verbosity	Verbosity level. Default: 0
...	other values passed to gpar() to set the graphical parameters e.g. lwd and col for the linewidth and colour of the outline stroke for each cube face.

Value

isometric line grid

Examples

```
grid <- isolinesGrob()
grid::grid.draw(grid)
```

`isopointsGrob` *Isometric grid of points*

Description

Isometric grid of points

Usage

```
isopointsGrob(
  N = 50,
  size = 5,
  x = NULL,
  y = NULL,
  col = "black",
  pch = ".",
  default.units = "npc",
  default.units.cube = "mm",
  verbosity = 0,
  ...
)
```

Arguments

`N`	extents
`size`	dimensions of cube i.e. the length of the vertical edge of the cube. Default: 5mm
`x, y`	the origin of the isometric coordinate system in 'snpc' coordinates. These values should be given as vanilla floating point values. Be default the origin is the middle bottom of the graphics device i.e. `(x, y) = (0.5, 0)`
`col`	Stroke colour for outline of cube faces. Default: black. If `NA` then no outlines will be drawn. If negative, then outline colour will be the same as the face colour.
`pch`	plotting character. default '.'
`default.units`	Default unit for (x,y) position is 'npc'
`default.units.cube`	Default unit for size of a cube is 'mm'
`verbosity`	Verbosity level. Default: 0
`...`	other values passed to `gpar()` to set the graphical parameters e.g. `lwd` and `col` for the linewidth and colour of the outline stroke for each cube face.

Value

isometric point grid

Examples

```
grid <- isopointsGrob()
grid::grid.draw(grid)
```

`organic_coords` *Voxel coordinates for an organic shape*

Description

Voxel coordinates for an organic shape

Usage

```
organic_coords
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 14292 rows and 4 columns.

See Also

Other datasets: `r_coords`, `sphere_coords`

Examples

```
head(organic_coords)
cubes <- isocubesGrob(organic_coords, size = 2)
grid::grid.newpage()
grid::grid.draw(cubes)
```

`r_coords` *Voxel coordinates for a sphere*

Description

Voxel coordinates for a sphere

Usage

```
r_coords
```

Format

An object of class `data.frame` with 68 rows and 3 columns.

See Also

Other datasets: organic_coords, sphere_coords

Examples

```
head(r_coords)
cubes <- isocubesGrob(r_coords, size = 5, y = 0)
grid::grid.newpage()
grid::grid.draw(cubes)
```

---

sphere_coords *Voxel coordinates for a sphere*

Description

Voxel coordinates for a sphere

Usage

```
sphere_coords
```

Format

An object of class tbl_df (inherits from tbl, data.frame) with 17071 rows and 3 columns.

See Also

Other datasets: organic_coords, r_coords

Examples

```
head(sphere_coords)
cubes <- isocubesGrob(sphere_coords, size = 2)
grid::grid.newpage()
grid::grid.draw(cubes)
```

Index