Package: funcgeo (via r-universe)

November 14, 2024

Type Package

Title Functional Geometry in R

Version 0.3.0

Description Peter Henderson's Functional Geometry in R.

License MIT + file LICENSE

Imports grid

Encoding UTF-8

RoxygenNote 7.3.1

URL https://github.com/mhenderson/funcgeo

BugReports https://github.com/mhenderson/funcgeo/issues

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

RemoteUrl https://github.com/MHenderson/funcgeo

RemoteRef v0.3.0

RemoteSha d1d65e36bc74db6b6bfbacea73845d23535e03ea

Contents

above *Superimpose two pictures, one above the other.*

Description

'above' returns a new picture made by placing the first input picture above the second input picture. The user can also provide arguments 'm' and 'n' in which case the pictures will be placed so that they occupy the output picture in a m:n ratio.

Usage

```
above(p, q, m = 1, n = 1)
```

Arguments

p	A 'grid::grob' picture
q	Another 'grid::grob' picture
m	An integer
n	Another integer

Value

A new 'grid::grob' picture obtained by superimposing p and q, one above the other.

beside *Superimpose two pictures side-by-side.*

Description

'beside' returns a new picture made by placing the first input picture beside the second input picture. The user can also provide arguments 'm' and 'n' in which case the pictures will be placed so that they occupy the output picture in a m:n ratio.

Usage

```
beside(p, q, m = 1, n = 1)
```

Arguments

p	A 'grid::grob' picture
q	Another 'grid::grob' picture
m	An integer
n	Another integer

Value

A new 'grid::grob' picture obtained by superimposing p and q, one beside the other.

cycle *Create a 2 by 2 grid pattern of rotated pictures*

Description

'cycle' returns a 2 by 2 grid pattern of four pictures, each obtained from the input 'grid::grob' picture 'p'. In the top left position of the grid is 'p', in the bottom-right position is the picture 'p' rotated by 90 degrees. In the bottom-left position 'p' is rotated through 180 degrees and in the top-right position 'p' is rotated through 270 degrees.

Usage

```
cycle(p)
```

Arguments

p	A 'grid::grob' for the top-left position

Value

A 'grid::grob' made up of rotations of the four input pictures arranged in a 2 by 2 grid pattern.

flip *Flip a picture*

Description

'flip' returns a new picture made by flipping the input picture along its vertical centre axis.

Usage

```
flip(g)
```

Arguments

g	A 'grid::grob' picture

Value

A flipped 'grid::grob' picture

nonet *Create a 3 by 3 grid pattern of pictures*

Description

'nonet' returns a 3 by 3 grid pattern made up of the nine input 'grid::grob' pictures: 'a', 's', 'd', 'f', 'g', 'h', 'j', 'k', 'l'. In the final grid 'a' is in the top-left position, 's' is in the top-middle position, 'd' is in the top-right position, 'f' is in the middle-left position, 'g' is in the centre position, 'h' is in the middle-right position, 'j' is in the bottom-left position, 'k' is in the bottom-middle position and 'l' is in the bottom-right position.

Usage

```
nonet(a, s, d, f, g, h, j, k, l)
```

Arguments

a	A 'grid::grob' for the top-left position
s	A 'grid::grob' for the top-middle position
d	A 'grid::grob' for the top-right position
f	A 'grid::grob' for the middle-left position
g	A 'grid::grob' for the centre position
h	A 'grid::grob' for the middle-right position
j	A 'grid::grob' for the bottom-left position
k	A 'grid::grob' for the bottom-middle position
l	A 'grid::grob' for the bottom-right position

Value

A 'grid::grob' made up of the nine input pictures arranged in a 3 by 3 grid pattern.

quartet *Create a 2 by 2 grid pattern of pictures*

Description

'quartet' returns a 2 by 2 grid pattern made up of the four input 'grid::grob' pictures: 'p', 'q', 'r' and 's'. In the final grid 'p' is in the top-left position, 'q' is in the top-right position, 'r' is in the bottom-left position and 's' is in the top-right position.

Usage

```
quartet(p, q, r, s)
```

Arguments

p	A ‘grid::grob‘ for the top-left position
q	A ‘grid::grob‘ for the top-right position
r	A ‘grid::grob‘ for the bottom-left position
s	A ‘grid::grob‘ for the bottom-right position

Value

A ‘grid::grob‘ made up of the four input pictures arranged in a 2 by 2 grid pattern.

---

rot *Rotate a picture*

Description

‘rot‘ returns a new picture, made by rotating the input picture. If no angle is specified then the new picture will be rotated clockwise by 90 degrees. A different angle can be specified with the ‘angle‘ argument.

Usage

```
rot(g, angle = 90)
```

Arguments

g	A ‘grid::grob‘ picture
angle	An angle

Value

A rotated ‘grid::grob‘ picture

Index