

Package: RockFab (via r-universe)

September 10, 2024

Type Package

Title Rock Fabric and Strain Analysis Tools

Version 1.2.1

Date 2014-01-14

Author Jeffrey R. Webber

Maintainer Jeffrey R. Webber <jwebber@geo.umass.edu>

Description Provides functions to complete three-dimensional rock fabric and strain analyses following the Rf Phi, Fry, and normalized Fry methods. Also allows for plotting of results and interactive 3D visualization functionality.

License GPL

Imports rgl, EBImage

Depends methods

NeedsCompilation no

Repository CRAN

Date/Publication 2022-06-23 07:18:46 UTC

Contents

RockFab-package	2
ElliIn	3
ElliOut	4
EllipAxes	5
ElliView	6
FlinnPlot	6
FRY-class	7
FryData	8
FryNormData	9
FryObjective	10
FrySelect	11
NadaiPlot	12
ParEx	13

RfPhi	14
RFPHI-class	15
RfPhiParRead	16
StereoCirc	16
StereoPlane	17
StereoPlot	18
StereoPoint	19
StereoWeb	21

Index	22
--------------	-----------

RockFab-package	<i>Rock fabric and strain analysis tools.</i>
-----------------	---

Description

Provides functions to complete three-dimensional rock fabric and strain analyses following the Rf Phi, Fry, and normalized Fry methods. Also allows for plotting of results and interactive 3D visualization functionality.

Details

Package:	RockFab
Type:	Package
Version:	1.2
Date:	2014-01-14
License:	GPL

Author(s)

Jeffrey R. Webber

Maintainer: Jeffrey R. Webber <jwebber@geo.umass.edu>

ElliIn *Read output files generated from the program Ellipsoid 2003.*

Description

ElliIn reads output files generated from the program Ellipsoid 2003 in order to create stereonet, Nadai plots, and interactive 3D ellipsoid models.

Usage

```
ElliIn(elli.files)
```

Arguments

`elli.files` Path of Ellipsoid 2003 file (.elli) to be read into R.

Value

Returns a data frame object containing properties of fitted ellipsoids including the filename of the Ellipsoid 2003 file; X, Y, and Z axial lengths of the ellipsoids; X, Y, Z axial azimuths and inclinations; strike and dip of the XY plane; rake of lineation within XY plane; number of sections used to fit ellipsoid; incompatibility index; octahedral shear strain, and Lode parameter.

Author(s)

Jeffrey R. Webber

References

Launeau, P. and P. Robin (2005). Determination of fabric and strain ellipsoids from measured sectional ellipses implementation and applications. *Journal of structural geology* 27 (12), 2223–2233

See Also

[ElliOut](#)

ElliOut

Write results of sectional analyses to Ellipsoid 2003 compatible files.

Description

ElliOut writes an ASCII file formatted for use in the program Ellipsoid 2003 for ellipsoid fitting to sectional data.

Usage

```
ElliOut(my.results, setup.file, out.file, use.ratio = FALSE)
```

Arguments

<code>my.results</code>	An object or list of objects of class RFPHI or FRY.
<code>setup.file</code>	Location of setup.txt file containing orientation data corresponding to the sectional fabric or strain data.
<code>out.file</code>	File to write Ellipsoid 2003 file to.
<code>use.ratio</code>	Logical, if TRUE, Rs ratio is used instead of axial lengths. Best used when axial lengths between sectional analyses are not meaningful.

Value

File for use in Ellipsoid 2003.

Author(s)

Jeffrey R. Webber

References

Launeau, P. and P. Robin (2005). Determination of fabric and strain ellipsoids from measured sectional ellipses implementation and applications. *Journal of structural geology* 27 (12), 2223–2233

See Also

[ElliIn](#)

EllipAxes	<i>Calculate ellipsoid axial lengths based on octahedral shear strain and Lode parameter.</i>
-----------	---

Description

Function uses the octahedral shear strain and Lode parameter of a desired strain ellipsoid and returns the normalized axial lengths X Y and Z.

Usage

```
EllipAxes(es, nu)
```

Arguments

es	Octahedral shear strain. Values must be positive.
nu	Lode parameter. Values must be between -1 and 1.

Value

A numeric vector of length three with values returned in descending order (i.e. X, Y, and Z)

Note

Not used in RockFab scripts but can be useful for other endeavors.

Author(s)

Jeffrey R. Webber

References

See for example: Ramsay, J. and M. Huber (1993). The techniques of modern structural geology.

Examples

```
es <- runif(min = 0, max = 3, n = 1)
nu <- runif(min = -1, max = 1, n = 1)

EllipAxes(es = es, nu = nu)
```

ElliView	<i>Interactive 3D fabric and strain ellipsoid visualization tool.</i>
----------	---

Description

Function deforms a unit sphere based on the parameters of the calculated fabric or strain ellipsoid and opens an rgl device to interactively view the result.

Usage

```
ElliView(elli.dat, out.file, obj.res = 3)
```

Arguments

elli.dat	Data frame object returned from ElliIn .
out.file	PLY file to save ellipsoid object to.
obj.res	Number of divisions to apply to spherical mesh. Larger numbers better approximate the ellipsoid surface but at the expense of file size and render time.

Value

PLY 3D object file.

Author(s)

Jeffrey R. Webber

See Also

[ElliIn](#)

FlinnPlot	<i>Function to plot a linear Flinn diagram from octahedral shear strain and Lode parameters of ellipsoids.</i>
-----------	--

Description

Function produces a standardized linear Flinn diagram from the octahedral shear strains and Lode parameters of fitted ellipsoids.

Usage

```
FlinnPlot(oss = 1, lp = 0, out.file, max.k = 5,  
plot.title = "Flinn diagram", labs)
```

Arguments

oss	A vector of octahedral shear strain values.
lp	A vector of Lode parameters
out.file	Path to PDF file for plotting.
max.k	Maximum x and y plotting limits. Either X/Y or Y/Z. Default 5 is good for moderate strains. Higher strains may need larger values.
plot.title	Title of the plot.
labs	Vector of labels of length equal to the length of oss and lp to add to plot.

Value

Creates a PDF of Flinn diagram.

Author(s)

Jeffrey R. Webber

References

Flinn, Derek. "On the deformation of the Funzie conglomerate, Fetlar, Shetland." *The Journal of Geology* (1956): 480-505.

See Also

[NadaiPlot](#)

FRY-class

Class "FRY"

Description

A class to hold data from sectional Fry analyses.

Objects from the Class

Objects can be created by calls of the form `new("FRY", ...)`.

Slots

sectionName: Object of class "character" defining the section ID string.

vectorMean: Object of class "numeric" for the calculated central void rake.

strainRatio: Object of class "numeric" for calculated strain ratio.

sampleSize: Object of class "numeric" containing the number of objects used in sectional analysis.

`rsAxes`: Object of class "numeric" with length of two containing the major and minor axial lengths of central void.

`meanObjectArea`: Object of class "numeric" the average object area.

`fryParams`: Object of class "data.frame" containing fry point coordinates, angle from origin, and distance from origin.

`voidScale`: Object of class "numeric" containing the distance from origin plot should be scaled.

Methods

No methods defined with class "FRY" in the signature.

Author(s)

Jeffrey R. Webber

See Also

[RFPHI](#)

Examples

```
showClass("FRY")
```

FryData

Determine Fry points and scale to central void region.

Description

Function takes locations of object centroids and calculates the parameters required to execute an interactive Fry plot. This function also determines a scale parameter that aids in data reduction.

Usage

```
FryData(object.data, pie.step = 5, expansion = 2, section.name)
```

Arguments

<code>object.data</code>	Data frame object returned from ParEx .
<code>pie.step</code>	Angular distance in degrees as a multiple of 180. This is used to divide the fry plot into n wedge shaped regions of distance <code>pie.step</code> where the closest point to the origin within each wedge is saved and the largest of these values is returned. This provides an estimate of the central void diameter.
<code>expansion</code>	An expansion factor to be multiplied to the central void diameter estimate in order to properly scale Fry plots.
<code>section.name</code>	The name of the sectional analysis.

Value

Returns an object of class [FRY](#).

Author(s)

Jeffrey R. Webber

See Also

[FryNormData](#), [RfPhi](#)

FryNormData

Determine normalized Fry points and scale to central void region.

Description

Function takes locations of object centroids and calculates the parameters required to execute an interactive Fry plot. This function also determines a scale parameter that aids in data reduction.

Usage

```
FryNormData(object.data, pie.step = 5, expansion = 2, section.name)
```

Arguments

<code>object.data</code>	Data frame object returned from ParEx .
<code>pie.step</code>	Angular distance in degrees as a multiple of 180. This is used to divide the fry plot into n wedge shaped regions of distance <code>pie.step</code> where the closest point to the origin within each wedge is saved and the largest of these values is returned. This provides an estimate of the central void diameter.
<code>expansion</code>	An expansion factor to be multiplied to the central void diameter estimate in order to properly scale Fry plots.
<code>section.name</code>	The name of the sectional analysis.

Value

Returns an object of class [FRY](#).

Author(s)

Jeffrey R. Webber

See Also

[FryData](#), [RfPhi](#)

 FryObjective

Objective ellipse fitting to central void.

Description

Function selects nearest points to Fry plot origin in angular wedges and fits an ellipse to these points using a deterministic least squares method. The wedges are iteratively modified from the resultant ellipse to better define selection bins that are narrower near the apogee of the central void and wider near the perigee.

Usage

```
FryObjective(object.data, n.pass = 15, pie.step = 12,
  expansion = 1.5, pie.pts = 1, section.name, ave.piepts = FALSE,
  norm = TRUE)
```

Arguments

<code>object.data</code>	Data frame object returned from ParEx .
<code>n.pass</code>	Number of ellipse fitting iterations.
<code>pie.step</code>	Angular distance in degrees as a multiple of 360. This is initially used to divide the fry plot into <code>n</code> wedge shaped regions of distance <code>pie.step</code> where the closest <code>n pie.pts</code> to the origin within each wedge is saved. Used in initial ellipse fit.
<code>expansion</code>	An expansion factor to be multiplied to the central void diameter estimate in order to properly scale Fry plots.
<code>pie.pts</code>	The number of nearest points to the plot origin that are selected in each wedge bin.
<code>section.name</code>	The name of the sectional analysis.
<code>ave.piepts</code>	Logical, if TRUE, the coordinates of <code>n pie.pts</code> selected from each wedge bin are averaged prior to least squares ellipse fitting.
<code>norm</code>	Logical, if TRUE, a normalized Fry method is used.

Details

This function creates either a normalized or standard Fry plot data frame from the centroids of object traces and attempts to objectively fit an ellipse to the central void. This is done initially by determining the nearest `n pie.pts` to the origin within each angular bin of length `pie.step` in degrees. A deterministic least squares method of ellipse fitting (modified code written by Michael Bedward) is applied to these points in order to make an initial estimate of the central void. This initial estimate will likely underestimate the R_s ratio because a bias of more points near the perigees of the central void and fewer points near the apogees. The bins are modified based on the estimate ellipse so that so that a smaller angular distance is used near the apogees and a larger angular distance near the perigees. Points are selected from these bins are previously described and a new ellipse is fitted. This procedure is repeated `n.pass` times to return the parameters of the objectively fitted central void ellipse.

Value

Returns an object of class [FRY](#).

Author(s)

Jeffrey R. Webber

References

Deterministic least squares method of ellipse fitting function is modified from code written by Michael Bedward.

Halir, Radim, and Jan Flusser. "Numerically stable direct least squares fitting of ellipses." Proc. 6th International Conference in Central Europe on Computer Graphics and Visualization. WSCG. Vol. 98. 1998.

Fry, N. (1979) Random point distributions and strain measurement in rocks.

Erslev, E. (1988) Normalized center-to-center strain analysis of packed aggregates.

See Also

[RfPhi](#), [FryData](#), [FryNormData](#), [FrySelect](#)

FrySelect	<i>Interactive central void selection tool for the Fry and normalized Fry techniques.</i>
-----------	---

Description

Creates an interactive Fry or normalized plot that enables a user to select the parameters of the central void.

Usage

```
FrySelect(fry.data, max.dim, out.file, normalized = FALSE, select = TRUE)
```

Arguments

fry.data	An object of class FRY returned from FryData or FryNormData .
max.dim	Fry plot diameter.
out.file	Path to desired PDF file for results.
normalized	Logical, if TRUE, a normalized Fry plot is made.
select	Logical, if TRUE, manual selection is required. If FALSE, an object of FRY will be plotted.

Value

Function produces a standardized PDF file containing the results of the sectional analysis and returns an object of class `FRY` with all parameters of the sectional analyses.

Author(s)

Jeffrey R. Webber

References

- Fry, N. (1979) Random point distributions and strain measurement in rocks.
 Erslev, E. (1988) Normalized center-to-center strain analysis of packed aggregates.

See Also

[RfPhi](#)

NadaiPlot

Nadai plotting tool for fabric and strain data.

Description

Function creates a standardized Nadai plot to better display calculated octahedral shear strain and Lode parameters of fabric and strain ellipsoids.

Usage

```
NadaiPlot(oss = 1, lp = 0, out.file,
  oss.int = 1, max.oss = 3,
  plot.title = "Nadai plot", labs)
```

Arguments

<code>oss</code>	Octahedral shear strain.
<code>lp</code>	Lode parameter.
<code>out.file</code>	Path to PDF file for plotting.
<code>oss.int</code>	Interval used between arcs of equal octahedral shear strain in plot.
<code>max.oss</code>	Maximum octahedral shear strain to be plotted at top of graph (used to scale the plot). Value must be an integer.
<code>plot.title</code>	Title of the plot.
<code>labs</code>	Vector of labels of length equal to the length of <code>oss</code> and <code>lp</code> to add to plot.

Value

Creates PDF of Nadai plot.

Author(s)

Jeffrey R. Webber

References

Nadai, A. (1963) Theory of fracture and flow of solids.

See Also

[FlinnPlot](#)

ParEx	<i>Parameter extraction tool to obtain location, eccentricity, and orientation of objects from sectional image traces.</i>
-------	--

Description

Function reads in an image file as accepted by the `readImage` function from the `EImage` package and uses routines therein to obtain location, eccentricity, and orientation data.

Usage

```
ParEx(img.file, out.file, section.title)
```

Arguments

<code>img.file</code>	Path to image file of sectional object traces.
<code>out.file</code>	Path of PDF file to plot fitted object ellipses.
<code>section.title</code>	Title of PDF plot.

Value

Function creates a PDF plot of fitted ellipses and returns a data frame object containing the parameters of each object.

Note

All image analysis routines are obtained from the `EImage` package

Author(s)

Jeffrey R. Webber

References

Pau, G., A. Oles, M. Smith, O. Sklyar, and W. Huber. (2013) `EImage`: Image processing toolbox for R.

RfPhi	<i>Determine Rs and corresponding parameters following the RfPhi method</i>
-------	---

Description

This function applies the RfPhi technique to object data following a Chi squared statistical test for the most uniform distribution of object orientations.

Usage

```
RfPhi(my.par, out.file, section.title, weight.vec = TRUE, unit.area = "px")
```

Arguments

<code>my.par</code>	A data frame object returned by ParEx or RfPhiParRead .
<code>out.file</code>	Path to PDF file where a standardized plot of the sectional results is made.
<code>section.title</code>	Title of the sectional analysis.
<code>weight.vec</code>	Logical, if TRUE an eccentricity weighting is applied such that more elliptical objects influence the vector mean stronger than objects with more circular shapes.
<code>unit.area</code>	The unit of measure in which the mean object area is calculated. If, for example, the input data is returned from ParEx the unit is in pixels; however, if manual data is used, the unit used to measure axial lengths should be used.

Value

Function creates a standardized PDF file containing an RfPhi plot, sectional Rs ellipse, Chi square plot, and a list of calculated properties. These properties are also returned via an object of class [RFPHI](#).

Author(s)

Jeffrey R. Webber

References

Dunnet, D. (1969). A technique of finite strain analysis using elliptical particles.
 Ramsay, J. (1969). Folding and fracturing of rocks.
 Lisle, R. (1985). Geological strain analysis, a manual for the rf/phi method.
 Mulchrone, K. and P. Meere (2001). A windows program for the analysis of tectonic strain using deformed elliptical markers.
 Mookerjee, M. and S. Nickleach (2011). Three-dimensional strain analysis using Mathematica.

See Also

[FryData](#), [FryNormData](#), [FrySelect](#)

RFPHI-class

Class "RFPHI"

Description

Class to hold basic RfPhi data.

Objects from the Class

Objects can be created by calls of the form `new("RFPHI", ...)`.

Slots

`sectionName`: Object of class "character" defining the section ID string.

`vectorMean`: Object of class "numeric" containing the calculated vector mean of object long axes.

`harmonicMean`: Object of class "numeric" containing the harmonic mean of object axial ratios.

`strainRatio`: Object of class "numeric" containing the calculated sectional strain or fabric ratio.

`indexSymmetry`: Object of class "numeric" containing the index of symmetry based on the vector and harmonic means.

`sampleSize`: Object of class "numeric" containing the number of objects used in the analysis.

`meanObjectArea`: Object of class "numeric" containing the average object area based on fitted object ellipses.

`rsAxes`: Object of class "numeric" of length two containing the axial lengths of the calculated sectional strain or fabric ellipse.

`chiSquare`: Object of class "data.frame" containing the results of the chi squared statistical test for uniform distribution at each increment of anti-strain.

Methods

No methods defined with class "RFPHI" in the signature.

Author(s)

Jeffrey R. Webber

See Also

[FRY](#)

Examples

```
showClass("RFPHI")
```

RfPhiParRead *Read manual data collected from the RfPhi method.*

Description

This function creates a data frame object typically constructed from sectional image data when manual Rf/Phi data is used.

Usage

```
RfPhiParRead(rfphi.file)
```

Arguments

`rfphi.file` A tab delineated text file containing the length of the long axis, short axis, and the orientation (rake) of objects measured in the field using a convention of rake measured positive clockwise from the direction of strike following right hand rule. Text file header must be used and should be defined as follows: l.axis
s.axis phi

Value

Returns a data frame object for use as input in [RfPhi](#).

Author(s)

Jeffrey R. Webber

See Also

[RfPhi](#)

StereoCirc *Primitive circle plotter for stereonet.*

Description

This function simply adds a primitive circle to an existing plot and is used in coordination with other stereonet plotting tools.

Usage

```
StereoCirc(n.seg = 360)
```

Arguments

`n.seg` The number of segments to use in circle plot. Defaults to 360.

Value

Adds primitive circle to open plot.

Author(s)

Jeffrey R. Webber

References

Modified from the package RFOC by Jonathan M. Lees.

See Also

[Stereoplot](#), [Stereoweb](#), [Stereoplane](#), [Stereopoint](#)

Examples

```
#Plot random plane and lineation
strike <- runif(min = 0, max = 360, n = 1)
dip <- runif(min = 0, max = 90, n = 1)
trend <- runif(min = 0, max = 360, n = 1)
plunge <- runif(min = 0, max = 90, n = 1)

if(interactive()){
  Stereoplot(my.title = "Example Stereonet")
  Stereoweb()
  Stereocirc()
  Stereopoint(my.az = trend, my.inc = plunge)
  Stereoplane(my.strike = strike, my.dip = dip)
}
```

Stereoplane

Add plane data to stereonet.

Description

Function adds a great circle to existing plot based on the strike and dip of a plane.

Usage

```
Stereoplane(my.strike = 0, my.dip = 45, my.color = "black")
```

Arguments

<code>my.strike</code>	The strike of the plane to plot as a great circle.
<code>my.dip</code>	The dip of the plane to plot as a great circle.
<code>my.color</code>	The line color used in the plot.

Value

Adds great circle of plane to existing plot.

Author(s)

Jeffrey R. Webber

References

Modified from the package RFOC by Jonathan M. Lees.

See Also

[StereoPlot](#), [StereoWeb](#), [StereoCirc](#), [StereoPoint](#)

Examples

```
#Plot random plane and lineation
strike <- runif(min = 0, max = 360, n = 1)
dip <- runif(min = 0, max = 90, n = 1)
trend <- runif(min = 0, max = 360, n = 1)
plunge <- runif(min = 0, max = 90, n = 1)

if(interactive()){
  StereoPlot(my.title = "Example Stereonet")
  StereoWeb()
  StereoCirc()
  StereoPoint(my.az = trend, my.inc = plunge)
  StereoPlane(my.strike = strike, my.dip = dip)
}
```

StereoPlot

Setup plot extent for stereonet.

Description

Creates a standardized plot with predetermined extents and margins to used with other stereonet tools.

Usage

```
StereoPlot(my.title = "Stereonet", new = TRUE, pdf.file)
```

Arguments

<code>my.title</code>	Title of the stereonet plot.
<code>new</code>	Logical, if FALSE, new plot will open in active window.
<code>pdf.file</code>	The filename to plot stereonet as a pdf. If missing, <code>windows()</code> is used.

Value

Sets up a plot for stereographic projections.

Author(s)

Jeffrey R. Webber

References

Modified from the package RFOC by Jonathan M. Lees.

See Also

[Stereopoint](#), [Stereopoint](#), [Stereopoint](#), [Stereopoint](#)

Examples

```
#Plot random plane and lineation
strike <- runif(min = 0, max = 360, n = 1)
dip <- runif(min = 0, max = 90, n = 1)
trend <- runif(min = 0, max = 360, n = 1)
plunge <- runif(min = 0, max = 90, n = 1)

if(interactive()){
  Stereoplot(my.title = "Example Stereonet")
  Stereopoint()
  Stereopoint()
  Stereopoint(my.az = trend, my.inc = plunge)
  Stereopoint(my.strike = strike, my.dip = dip)
}
```

Stereopoint

Add line data to stereonet.

Description

Adds point to stereonet based on the trend and plunge of linear data.

Usage

```
Stereopoint(my.az = 90, my.inc = 45,
  my.color = "black", my.pch = 19,
  my.size = 0.25, my.label)
```

Arguments

<code>my.az</code>	Lineation trend as an azimuth.
<code>my.inc</code>	Lineation plunge or inclination.
<code>my.color</code>	Point color.
<code>my.pch</code>	pch value. See <code>?par</code>
<code>my.size</code>	Size of points. Calls <code>cex(my.size)</code> . See <code>?par</code>
<code>my.label</code>	Label to add next to point

Value

Adds point to stereonet based on the trend and plunge of linear data.

Author(s)

Jeffrey R. Webber

References

Modified from the package RFOC by Jonathan M. Lees.

See Also

[StereoPlot](#), [StereoWeb](#), [StereoPlane](#), [StereoCirc](#)

Examples

```
#Plot random plane and lineation
strike <- runif(min = 0, max = 360, n = 1)
dip <- runif(min = 0, max = 90, n = 1)
trend <- runif(min = 0, max = 360, n = 1)
plunge <- runif(min = 0, max = 90, n = 1)

if(interactive()){
  StereoPlot(my.title = "Example Stereonet")
  StereoWeb()
  StereoCirc()
  StereoPoint(my.az = trend, my.inc = plunge)
  StereoPlane(my.strike = strike, my.dip = dip)
}
```

StereoWeb

Equal area stereonet grid plotter.

Description

Creates an equal-area net to existing stereographic projection.

Usage

```
StereoWeb()
```

Value

Plots an equal-area net to existing stereographic projection with 10 degree cells.

Author(s)

Jeffrey R. Webber

References

Modified from the package RFOC by Jonathan M. Lees.

See Also

[StereoPlot](#), [StereoCirc](#), [StereoPlane](#), [StereoPoint](#)

Examples

```
#Plot random plane and lineation
strike <- runif(min = 0, max = 360, n = 1)
dip <- runif(min = 0, max = 90, n = 1)
trend <- runif(min = 0, max = 360, n = 1)
plunge <- runif(min = 0, max = 90, n = 1)

if(interactive()){
  StereoPlot(my.title = "Example Stereonet")
  StereoWeb()
  StereoCirc()
  StereoPoint(my.az = trend, my.inc = plunge)
  StereoPlane(my.strike = strike, my.dip = dip)
}
```

Index

* classes

FRY-class, [7](#)
RFPHI-class, [15](#)

ElliIn, [3](#), [4](#), [6](#)
ElliOut, [3](#), [4](#)
EllipAxes, [5](#)
ElliView, [6](#)

FlinnPlot, [6](#), [13](#)
FRY, [9](#), [11](#), [12](#), [15](#)
FRY-class, [7](#)
FryData, [8](#), [9](#), [11](#), [14](#)
FryNormData, [9](#), [9](#), [11](#), [14](#)
FryObjective, [10](#)
FrySelect, [11](#), [11](#), [14](#)

NadaiPlot, [7](#), [12](#)

ParEx, [8–10](#), [13](#), [14](#)

RFPHI, [8](#), [14](#)
RfPhi, [9](#), [11](#), [12](#), [14](#), [16](#)
RFPHI-class, [15](#)
RfPhiParRead, [14](#), [16](#)
RockFab (RockFab-package), [2](#)
RockFab-package, [2](#)

StereoCirc, [16](#), [18–21](#)
StereoPlane, [17](#), [17](#), [19–21](#)
StereoPlot, [17](#), [18](#), [18](#), [20](#), [21](#)
StereoPoint, [17–19](#), [19](#), [21](#)
StereoWeb, [17–20](#), [21](#)