

Package ‘ssEDA’

3 October 2019

Title Exploratory Data Analysis for Earthquake Data

Version 2.3-20

Date 2019-10-03

Author David Harte

Maintainer David Harte <d.s.harte@gmail.com>

Description Contains functions to do epicentral plots, magnitude-time plots, b-value plots, and various other empirical summaries.

Depends ssBase

Imports chron, maps

Suggests mapdata, ssNZ, ssPDE, ssSCEC, rggobi

LazyData yes

License GPL (>=2)

URL <http://www.statsresearch.co.nz/dsh/sslib/>

Additional_repositories <http://www.statsresearch.co.nz/dsh/sslib/r-repo>

R topics documented:

ssEDA-package	2
bvalue.contour	3
Change Log	4
depth.hist	8
dkagan	9
epicentres	10
epicentres.identify	14
est.kagan	15
freq.cusum	17
freq.magnitude	18
hemisphere	20
magnitude.contour	20
magnitude.convert	22
magnitude.cusum	23

magnitude.time	24
major	25
map1	29
multigraph	31
plot.subset	32
rotation	32
threeD	34
timeplot	36
worldLores	37

Index	38
--------------	-----------

ssEDA-package	<i>Overview of Package ssEDA</i>
---------------	----------------------------------

Description

The **ssEDA** package contains functions to do Exploratory Data Analysis. The package was originally ported to R by Ray Brownrigg in 2000.

Summary of Functions Within the Package

Functions are listed under the the groupings below.

Epicentral and Hypercentral Plots: See [epicentres](#), [epicentres.identify](#), [rotation](#), [threeD](#).

Spatial Magnitude Variation: See [bvalue.contour](#), [magnitude.contour](#).

Catalogue Completeness: See [freq.magnitude](#), [freq.cusum](#), [magnitude.cusum](#).

Other EDA: See [depth.hist](#), [magnitude.time](#), [timeplot](#), [plot.subset](#).

Maps: See [map1](#), [worldLores](#), [hemisphere](#). The **maps** package is required, and a higher resolution world map can be found in the **mapdata** package.

Miscellaneous: See [magnitude.convert](#), [major](#), [multigraph](#).

Kagan Distribution: See [dkagan](#), [pkagan](#), [qkagan](#), [rkagan](#), [est.kagan](#).

References

- Choi, E. & Hall, P.G. (1999). Nonparametric approach to analysis of space-time data on earthquake occurrences. *J. Comput. Graph. Statist.* **8**(4), 733–748. doi: [10.2307/1390824](#)
- Kagan, Y.Y. (1997). Seismic moment-frequency relation for shallow earthquakes: Regional comparison. *Journal of Geophysical Research* **102**, 2835–2852. doi: [10.1029/96JB03386](#)
- Kanamori, H. & Anderson, D.L. (1975). Theoretical basis of some empirical relations in seismology. *Bulletin of the Seismological Society of America* **65**(5), 1073–1095.
- Ruppert, D. & Wand, M.P. (1994). Multivariate locally weighted least squares regression. *Ann. Statist.* **22**(3), 1346–1370. doi: [10.1214/aos/1176325632](#)
- Utsu, T. (1965). A method for determining the value of b in a formula $\log n = a - bM$ showing the magnitude frequency relation for earthquakes. *Geophys. Bull. Hokkaido University* **13**, 99–103 (in Japanese). doi: [10.14943/gbhu.13.99](#)

Vere-Jones, D.; Robinson, R. & Yang, W. (2001). Remarks on the accelerated moment release model: problems of model formulation, simulation and estimation. *Geophysical Journal International* **144**, 517–531. doi: [10.1046/j.1365246x.2001.01348.x](https://doi.org/10.1046/j.1365246x.2001.01348.x)

bvalue.contour	<i>b-Value Contours at Specified Depth</i>
----------------	--

Description

This function calculates b-value contours at a pre-specified depth.

Usage

```
bvalue.contour(events, divs=c(15, 15, 15), h=c(0.5, 0.5, 15),
               threshold=30, depth=10)
```

Arguments

events	defines the events to be plotted. It is an object of class "subset", generally created by subsetcircle , subsetpolygon , subsetrect or subsetsphere .
divs	a vector of length three. It specifies the grid points where b-values are calculated. The first, second and third entries of this vector correspond to the number of divisions on the longitude, latitude and depth axes respectively.
h	a vector of length three. This specifies the bandwidths for the kernel function (see Details).
threshold	the minimum number of events needed to evaluate pointwise b-values (see Details).
depth	specifies the longitude-latitude plane where the b-value contours are plotted.

Details

The b-values are calculated in two steps. First, they are evaluated on a 3D grid using the maximum likelihood method (Utsu, 1965). The MLE estimate at (x, y, z) (corresponds to longitude, latitude and depth respectively) only utilises data that lie in

$$S = (x - \text{divs}[1], x + \text{divs}[1], y - \text{divs}[2], y + \text{divs}[2], z - \text{divs}[3], z + \text{divs}[3]).$$

However, if the number of points in S is less than that specified by the threshold parameter, the closest threshold number of points to (x, y, z) are used. The resulting b-value estimates are smoothed using local linear regression technique described in Ruppert and Wand (1994) using a standard trivariate normal kernel. The smoothness of the contours is controlled by the bandwidths of the kernel.

Value

NULL

Author(s)

Edwin Choi (ANU), 1997

References

Cited references are listed on the [ssEDA-package](#) manual page.

See Also

[magnitude.contour](#)

Examples

```
# This example requires the NZ catalogue

## Not run:
if (require(ssNZ)){

# Remember the parameters of the graphics device
par.reset <- par(no.readonly=TRUE)

# Make the Wellington Catalogue
as.catalogue(subsetrect(NZ, minlat=-42.2, maxlat=-40.5,
  minlong=173.6, maxlong=176.0, minday=julian(1,1,1978)),
  catname="Wellington")

b <- subsetrect(Wellington, maxdepth=100, minmag=3,
  minlat=-42.2, maxlat=-40.5, minlong=173.6, maxlong=176.0,
  minday=julian(1,1,1980), maxday=julian(1,1,2000))

par(pty="s")
# NOTE: bvalue.contour() will not work if called directly.
# It is blocked because it uses as.catalogue() which writes into
# the user's directory. A *temporary* fix is to write the function to
# your local directory first, then source that code:
dump("bvalue.contour", file="bvalue.contour.R")
source("bvalue.contour.R")
bvalue.contour(b, h=c(0.25, 0.2, 2), depth=10)
if (require(mapdata)) mpnm <- "nzHires" else mpnm <- "nz"
map(database=mpnm, add=TRUE, col="green3")
par(par.reset)
}

## End(Not run)
```

Description

This page contains a listing of recent changes made to functions, and known general problems.

Details

1. [hemisphere](#): An argument has been added called `filename`, with default value `"temp.ps"`, giving the name of the postscript file. (February 2003)
2. Function `time.plot` has been renamed to [timeplot](#) to stop confusion with methods and generic functions. (June 2003)
3. [hemisphere](#): Argument `filename` has been deleted. The plot is now written to the current device or a new window is opened. (June 2003)
4. [rotation](#): Argument `psname` has been deleted. The plot is now written to the current device or a new window is opened. (June 2003)
5. [epicentres](#): Argument `mapname` has been added. (July 2003)
6. [epicentres.identify](#): Default on argument `mapname` has been changed to `"world.lores"`. (July 2003)
7. [epicentres.identify](#): Changed example in documentation. (October 2003)
8. `world.lores` data object renamed to [worldLores](#) and implemented as `.R` form in `data/` directory rather than creating `.rda` file at install time. (October 2003)
9. `world.lores` map name in [epicentres.identify](#) renamed to [world](#) to match `maps_2.0-2`. (October 2003)
10. [epicentres](#): default value of argument `mapname` changed from `world2.lores` to [world2](#) to match `maps_2.0-2`. (November 2003)
11. [epicentres.identify](#): default value of argument `mapname` changed from `world` to [world2](#). (November 2003)
12. Minor documentation formatting changes, mainly to use `\dQuote` and `\pkg`. (January 2004)
13. [map1](#): New function. (February 2004)
14. [epicentres](#): now calls [map1](#) to plot map. Is now plotted to the current device, not necessarily a postscript file. Colours are now not defined using the current palette, but explicitly using character strings. (February 2004)
15. [worldLores](#), [hemisphere](#): Examples modified to make plotting region square. (February 2004)
16. [timeplot](#), [depth.hist](#): `ylim` extended so that tallest bar doesn't touch upper box boundary. (February 2004)
17. `texps1`: has been deleted. (February 2004)
18. [magnitude.contour](#), [bvalue.contour](#): Examples added, now plotted to current device rather than a postscript file. (February 2004)
19. [depth.hist](#), [freq.cusum](#), [magnitude.cusum](#), [magnitude.time](#), [multigraph](#), [plot.subset](#), [timeplot](#): The statement `if(length(dev.list()) == 0) X11()` has been removed, was required in S-PLUS. (February 2004)
20. [epicentres](#): Depth ranges added as a footnote in example plots. (February 2004)

21. `map1`: calls the function `map`, whose argument `color` has been changed to `col`. (February 2004)
22. `rotation`: x-axis label changed to make use of degree symbol. (March 2004)
23. `bvalue.contour`, `magnitude.contour`: map added to example plot. (March 2004)
24. `worldLores`: quotes added around package name in code, i.e. "ssEDA". (7 May 2004)
25. `epicentres`: function default values in arguments `magnitude` and `depth` changed from using $1/\theta$ to Inf . (7 May 2004)
26. `threeD`: a statement `require(xgobi)` has been added to the code. This requirement is then no longer a "package" requirement. (7 May 2004)
27. A statement `require(ssNZ)` has been added to the examples that use the NZ catalogue. Hence `ssNZ` is no longer a "package" requirement of `ssEDA`. (18 May 2004)
28. `epicentres`: code modified so that argument `events` can again be a catalogue. (9 Apr 2005)
29. `epicentres`: expression statement in examples with subtitle containing multiple inequalities changed to make syntactically compatible with recent R updates. (16 Apr 2005)
30. `epicentres`: Reference to postscript graphics device eliminated from manual page. (21 Apr 2005)
31. `worldLores`: now implemented directly as a `data()` call, rather than via `delay()`, which has been deprecated. (9 Jun 2005)
32. `freq.cusum`: replace `seq(begin, finish + 1) - 1900` with `formatC((seq(begin, finish + 1))%100, width=2, flag="0")`. (22 Dec 2005)
33. `magnitude.cusum`: replace `at.breaks - 1900` with `formatC(at.breaks%100, width=2, flag="0")`. (22 Dec 2005)
34. Package vignettes added. (22 Dec 2005)
35. Fix error in package vignettes. (04 May 2006)
36. `depth.hist`, `timeplot`, `freq.magnitude`: removed redundant arguments in call to `hist` when `plot==FALSE`. Caused warning messages. (30 Jan 2007)
37. `major`: new page containing a listing of major earthquake events. (14 May 2007)
38. The `if require(catalogue)` syntax in all examples has been changed to `require(catalogue)`. (15 May 2007)
39. The DESCRIPTION file has the following added: `Suggests: xgobi, mapdata, ssNZ, ssPDE, ssSCEC`. (05 Jul 2007)
40. All occurrences of `subset.circle`, `subset.polygon`, `subset.rect`, `subset.sphere` in the Examples have been changed to `subsetcircle`, `subsetpolygon`, `subsetrect`, `subsetsphere`. See Changes in `ssBase`. (8 Nov 2007)
41. `threeD`: modified to use the `ggobi` utility. See topic `threeD` for installation notes for `ggobi`. (21 Nov 2007)
42. Removal of LaTeX markups from DESCRIPTION file. (31 May 2009)
43. `timeplot`: complete rewrite of function. (11 Oct 2009)
44. `timeplot` and `depth.hist`: argument added to specify the colour of the bars. (13 Nov 2009)
45. `threeD`: more detailed explanation on the manual page about `GGobi` installation and also use of the GUI. (30 Apr 2010)

46. `threeD`: Incorrect angle of view in description on manual page . (1 May 2010)
47. `magnitude.time`: argument `ylim` added, with default value `range(magnitude) + c(-0.1, 0.1)`. Prior to this change, it was set as `range(magnitude) + c(-0.5, 0.5)`. (31 Aug 2010)
48. The `require(catalogue)` syntax in all examples has been changed back to `if (require(catalogue))`. Caused problems when generating Windows binary without catalogues. (31 Aug 2010)
49. `magnitude.time`: minor change to argument `ylim`: default action now determined by `plot`. (20 Sep 2010)
50. `rotation`: new arguments `ylim` and `zero`. (20 Sep 2010)
51. `freq.magnitude`: function has largely been rewritten. The original code written for S-Plus no longer did exactly what was required. (20 Sep 2010)
52. Add CITATION file. (24 Sep 2010)
53. Use higher resolution maps in examples if **mapdata** package available. (28 Sep 2010)
54. `magnitude.time`: new argument `xlim`. (2 Jan 2011)
55. `freq.cusum`: more explanation in documentation, mark both bounds of target interval on plot. (26 Mar 2011)
56. `magnitude.time`: add argument `axes`. (02 May 2011)
57. `threeD`: removed from package. The code and manual page can be found in directory `threeD` if you want to reinstate. (05 Oct 2011)
58. Implement NAMESPACE and remove file `/R/zzz.R`. (4 Nov 2011)
59. `freq.cusum`, `magnitude.cusum`: In call to function `axis`, change argument `lab` to `labels`. (03 Apr 2012)
60. `freq.magnitude`: Correct displayed equation on manual page (Details) for *b*-value (*n* missing in numerator). (28 May 2012)
61. `freq.magnitude`: Replace `y <-1 -cumsum(counts$intensities)*delta` with `y <-1 -cumsum(counts$density)*delta`. (14 Apr 2013)
62. `hemisphere`: Changed `worldLores` to `ssEDA::worldLores`, deleted `data(worldLores, package="ssEDA")`. (02 Jul 2015)
63. `map1`: Added argument `lwd`. (05 Sep 2016)
64. `epicentres`: Added arguments `maplwd` and `mapcol`. (05 Sep 2016)
65. Additions to file NAMESPACE. (08 Sep 2016)
66. `magnitude.time`: Added arguments `type` and `cex`. (16 Nov 2016)
67. `bvalue.contour`, `magnitude.contour`: Both have problems. Call `as.catalogue` which tries to write into user directory. Blocked now by R. Needs modification. See temporary work around in Examples. (19 Jun 2017)
68. Include overview topic `ssEDA-package`. (23 Aug 2017)
69. `threeD`: `rggobi::ggobi`. (11 Oct 2017)
70. `threeD`: `requireNamespace('rggobi')`, manual page updates. (13 Oct 2017)
71. **maps** changed to “Suggests”. (18 Oct 2017)
72. `epicentres.identify`: added arguments `pch` and `cex`. Note that the default symbol has changed to `pch=1`, the old default was `pch='.'`. (08 Dec 2017)

- 73. [magnitude.time](#): Added argument labels. (10 Jan 2018)
- 74. Package **maps** changed from “Suggests” to “Imports”, see files DESCRIPTION and NAMESPACE. (02 May 2018)
- 75. [map1](#): Problem with maps either over-running or not running up to the plot boundary. See comment in R/maps1.R. (04 May 2018)
- 76. Rebuild package in R version 3.6.1. (22 Sep 2019)
- 77. [epicentres](#): Added argument axes. (03 Oct 2019)

depth.hist

Depth Histogram

Description

Plots a histogram of the depth of selected events.

Usage

```
depth.hist(events, cumulative=FALSE, delta=NA, col="grey80")
```

Arguments

events	defines the events to be plotted. It is an object of class "subset", generally created by subsetcircle , subsetpolygon , subsetrect or subsetsphere .
cumulative	boolean value. If TRUE, the plot is a cumulative histogram. Default if FALSE.
delta	numeric, the bar width used in the histogram. By default, if the depth range is greater than 110 km, delta=10, else delta=1.
col	colour of the bars, default is "grey80".

See Also

[freq.cusum](#), [freq.magnitude](#), [magnitude.cusum](#), [magnitude.time](#), [epicentres](#), [timeplot](#), [multigraph](#)

Examples

```
data(NZ55)

depth.hist(subsetrect(NZ55, minmag=5.5))
```


dkagan

*Kagan Distribution***Description**

Density, cumulative probability, quantiles and random generation for the Kagan distribution.

Usage

```
dkagan(M, alpha=1, beta=0, gamma=7, M0=4, mag=TRUE, theta=0.75, phi=2.4)
pkagan(M, alpha=1, beta=0, gamma=7, M0=4, mag=TRUE, theta=0.75, phi=2.4)
qkagan(p, alpha=1, beta=0, gamma=7, M0=4, mag=TRUE, theta=0.75, phi=2.4)
rkagan(n, alpha=1, beta=0, gamma=7, M0=4, mag=TRUE, theta=0.75, phi=2.4)
```

Arguments

p	Vector of probabilities.
n	Non-negative sample size. If length(n) is greater than 1, then length(n) random variables are returned.
M	Vector of quantiles.
alpha	Parameter of Kagan distribution. Index of the power law.
beta	Parameter of Kagan distribution. Controls the lower end of the distribution.
gamma	Parameter of Kagan distribution. Controls the upper end of the distribution.
M0	Lowest magnitude under consideration.
mag	Flag indicating whether stress or magnitudes are to be used. Default is true where stresses are used.
theta	Parameter of <code>magnitude.convert</code> .
phi	Parameter of <code>magnitude.convert</code> .

Details

Elements of p or M that are missing will cause the corresponding elements of the result to be missing.

The variables in this basic form of the distribution correspond physically to moments (stress, benioff strain relief). The alternative form allows the parameters and variable to be specified in terms of magnitudes. Once the appropriate values are calculated, if need be, they can be transformed to magnitudes using the [magnitude.convert](#) function.

Value

Density (dkagan), probability (pkagan), quantile (qkagan) or random sample (rkagan) for the Kagan distribution. The cumulative density function is given by:

$$F(x) = 1 - \left\{ \left(1 + \frac{x}{\beta} \right)^{-\alpha} \exp \left(\frac{-x}{\delta} \right) \right\}$$

where x , β and δ are to be interpreted as stresses according to the relation:

$$S(M) = 10^{\phi + \theta M}.$$

Warning

The function `qkagan` uses a approximate numerical method (Newton-Raphson) to evaluate the quantile.

References

Cited references are listed on the [ssEDA-package](#) manual page.

See Also

[magnitude.convert](#)

Examples

```
# Examine the Gutenberg-Richter Law:
# Use default parameters

x <- seq(0, 10, length=1000)
plot(log(x), log(1-pkagan(x)), type="l")
```

epicentres

Epicentral Plot of Selected Events

Description

Plots earthquake epicentres selected from a given catalogue. The colour and size of each point can be made to represent the depth and the magnitude of the event, respectively.

Usage

```
epicentres(events, usr=NA, magnitude=c(-Inf, Inf), cex=NA,
            criteria=TRUE, depth=c(0, Inf),
            colours=c("red2", "yellow2", "green2", "cyan2", "blue2"),
            mapname="world2", mapcol="gray35", maplwd=1, axes=TRUE)
```

Arguments

events	defines the events to be plotted. It is an object of class "subset", "catalogue", or "matrix". If of class "subset", it will generally be created by subsetcircle , subsetpolygon , subsetrect or subsetsphere . If of class "matrix", it must contain the named columns: longitude, latitude, depth, and magnitude.
usr	a vector of length 4 than defines boundary positions of the axes, i.e. <code>c(xmin, xmax, ymin, ymax)</code> . The default is selected so that the extreme events are just included.

magnitude	vector of magnitudes, where <code>magnitude[i]</code> is less than or equal to <code>magnitude[i+1]</code> . If <code>length(magnitude)==2</code> , then events with magnitude greater than or equal to <code>magnitude[1]</code> and less than <code>magnitude[2]</code> are selected and events are plotted as points. If <code>length(magnitude)</code> is greater than 2, <code>magnitude</code> defines a sequence of intervals used to group events. Events are plotted as circles of increasing size by increasing magnitude. The size of the circle for each interval is determined by the <code>cex</code> parameter. Default is <code>c(-Inf, Inf)</code> .
cex	vector containing the sizes of the plotted points or circles. The length of the vector should correspond to the number of magnitude intervals defined by the <code>magnitude</code> vector. If <code>length(magnitude)==2</code> , then events will be plotted as points and <code>cex</code> will be a scalar denoting the size of the points. If <code>length(magnitude)</code> is greater than 2, then the events are plotted as circles of increasing radius for increasing magnitude. The circle radii are proportional to the values specified in the <code>cex</code> vector. Default values of <code>cex</code> are <code>seq(0.2, length(magnitude)-1, 0.5)</code> . These values are appropriate for plots with the order of a thousand points, but may be too small for 100 or less.
criteria	logical variable. If TRUE (default) the subsetting parameters are printed below the graph.
depth	vector of depths, where <code>depth[i]</code> is less than or equal to <code>depth[i+1]</code> . If <code>length(depth)==2</code> , then events with depth greater than or equal to <code>depth[1]</code> and less than <code>depth[2]</code> are selected. If <code>length(depth)</code> is greater than 2, <code>depth</code> defines a sequence of intervals, for which events are plotted in the colours specified by the parameter <code>colours</code> . Default is <code>c(0, Inf)</code> .
colours	vector of colours for the different depth intervals.
mapname	character string giving the map name. Low resolution maps contained in package maps are "nz" and "world2" (default). High resolution maps contained in the package mapdata are "nzHires", "world2Hires" and "chinaHires". The "2" on "world2" denotes the version where the longitudes are all positive.
mapcol	map outline colour.
maplwd	map line width.
axes	logical, if TRUE (default) plot axes are drawn.

Details

The plot is placed into the current graphics device. If a graphics device is not open, one of the default type is opened.

The plot uses a rectangular projection. The aspect of the plot (y/x ratio) is calculated within the function, and is set so that the use of the available area within the graphics device is maximised. The graphics device remains open at the end of the function execution so that additional annotation and points can be added to the plot, if required. The margins and aspect ratio will also remain in effect at the end of the execution of the function, and should be reset if multiple plots are to be written to the same output file; see "Examples" below.

See Also

[depth.hist](#), [freq.cusum](#), [freq.magnitude](#), [magnitude.cusum](#), [magnitude.time](#), [timeplot](#), [multigraph](#)

Examples

```
# Remember the parameters of the graphics device

par.reset <- par(no.readonly=TRUE)

#-----

# Here the data are contained in a matrix

data <- cbind(latitude=c( -41.70, -39.30, -40.50, -30.62, -28.42, -32.31),
                  longitude=c(172.20, 177.00, 175.50, 178.42, 179.97, 181.21),
                  depth=c(20, 30, 45, 12, 300, 339),
                  magnitude=c(7.8, 7.8, 7.6, 7.4, 7.4, 7.9))

epicentres(data, usr=c(172, 182, -42, -28), cex=2, mapname="nz")
par(par.reset)

#-----

# Here the data are contained in a catalogue

data(NZ55)

if (require(mapdata)) mpnm <- "nzHires" else mpnm <- "nz"
epicentres(NZ55, mapname=mpnm)
title(main="Some Large NZ Events")
par(par.reset)

#-----

# Requires the NZ Catalogue
if (require(ssNZ)){

# Make the Wellington Catalogue
as.catalogue(subsetrect(NZ, minlat=-42.2, maxlat=-40.5,
                      minlong=173.6, maxlong=176.0, minday=julian(1,1,1978)),
              catname="Wellington")

b <- subsetrect(Wellington, minlong=173.6, maxlong=176.0, minlat=-42.1,
                maxlat=-40.5, mindepth=36, minday=julian(1,1,1988),
                maxday=julian(1,1,1993))

# Note that usr specifies the area in the plot
if (require(mapdata)) mpnm <- "nzHires" else mpnm <- "nz"
epicentres(b, usr=c(b$minlong-0.01, b$maxlong+0.01,
                  b$minlat-0.005, b$maxlat+0.005),
            depth=c(36, 50, 70, 100, 150, Inf), criteria=FALSE,
            magnitude=c(2, 3, 4, 5, 6, Inf), mapname=mpnm)
title(sub=expression(paste("Depth (km): ", 36 <= {red < {50 <=
{yellow < {70 <= {green < {100 <= {cyan < {150 <= {blue <
infinity}}}}}}}})), line=3)
```

```

title(main="Wellington Catalogue:  Deep Events")
par(par.reset)
}

#-----

#   Requires the NZ Catalogue
if (require(ssNZ)){

b <- subsetrect(Wellington, minlong=173.6, maxlong=176.0, minlat=-42.1,
                 maxlat=-40.5, maxdepth=35.99, minday=julian(1,1,1988),
                 maxday=julian(1,1,1993))

#   Note that usr specifies the area in the plot
if (require(mapdata)) mpnm <- "nzHires" else mpnm <- "nz"
epicentres(b, usr=c(b$minlong-0.01, b$maxlong+0.01,
                   b$minlat-0.005, b$maxlat+0.005),
           depth=c(0, 10, 15, 20, 25, 36), criteria=FALSE,
           magnitude=c(2, 3, 4, 5, 6, Inf), mapname=mpnm)
title(sub=expression(paste("Depth (km):  ", 0 <= {red < {10 <=
{yellow < {15 <= {green < {20 <= {cyan < {25 <= {blue < 36}}}}}}}})),
      line=3)
title(main="Wellington Catalogue:  Shallow Events")
par(par.reset)
}

#-----

#   Requires the NZ Catalogue
if (require(ssNZ)){

b <- subsetrect(NZ, minlong=166, maxlong=180, minlat=-48,
                 maxlat=-34, minmag=3, minday=julian(1,1,1970),
                 maxday=julian(1,1,1993), mindepth=36)

if (require(mapdata)) mpnm <- "nzHires" else mpnm <- "nz"
epicentres(b, depth=c(36, 70, 100, 140, 200, Inf), criteria=FALSE,
           usr=c(b$minlong, b$maxlong, b$minlat, b$maxlat), mapname=mpnm)
title(sub=expression(paste("Depth (km):  ", 36 <= {red < {70 <=
{yellow < {100 <= {green < {140 <= {cyan < {200 <= {blue <
infinity}}}}}}}})), line=3, cex.sub=0.95)
title(main="NZ Catalogue:  Deep Events")
par(par.reset)
}

#-----

#   Requires the NZ Catalogue
if (require(ssNZ)){

b <- subsetcircle(NZ, centrelong=176.8, centrelat=-37.8,
                  minmag=3, minday=julian(1,1,1987),
                  maxday=julian(1,1,1988), maxradius=100)

```

```

if (require(mapdata)) mpnm <- "nzHires" else mpnm <- "nz"
epicentres(b, cex=1.3, mapname=mpnm)
title(main="Edgumbe Earthquake")
par(par.reset)
}

#-----

#   Requires the NZ Catalogue
if (require(ssNZ)){

b <- subsetcircle(NZ, centrelong=177.5, centrelat=-37,
                  minmag=3, minday=julian(1,1,1984),
                  maxday=julian(1,1,1986), maxradius=100)

if (require(mapdata)) mpnm <- "nzHires" else mpnm <- "nz"
epicentres(b, cex=1.5, mapname=mpnm)
title(main="Bay of Plenty Swarm")
par(par.reset)
}

#-----

#   Requires the NZ Catalogue
if (require(ssNZ)){

b <- subsetcircle(NZ, centrelong=176.4, centrelat=-40.4,
                  minmag=3, minday=julian(1,1,1990),
                  maxday=julian(1,1,1991), maxradius=50)

if (require(mapdata)) mpnm <- "nzHires" else mpnm <- "nz"
epicentres(b, cex=1.5, mapname=mpnm)
title(main="Weber (Dannevirke) Earthquake")
par(par.reset)
}

```

epicentres.identify *Identify Epicentre Outliers*

Description

Plots earthquake epicentres selected from a given catalogue. The points are plotted on the screen, and then may be identified.

Usage

```
epicentres.identify(events, mapname = "world2", criteria = TRUE, pch = 1, cex = 1)
```

Arguments

events	defines the events to be plotted. It is an object of class "subset", generally created by subsetcircle , subsetpolygon , subsetrect or subsetsphere .
mapname	character string giving the map name. Low resolution maps contained in package maps are "nz" and "world2" (default). High resolution maps contained in the package mapdata are "nzHires", "world2Hires" and "china". The "2" on "world2" denotes the version where the longitudes are all positive.
criteria	boolean variable. If TRUE (default) the subsetting parameters are printed below the graph.
pch	plotting character, either an integer specifying a symbol or a single character; see par for further details.
cex	size of plotted points.

Details

The points are plotted on an appropriate map, and points which are considered to be outliers are highlighted with larger circles. After the points have been plotted, the `identify` command is used for the user to identify points of interest. Use the left mouse button to select points (which are then identified on the map with their sequence number within the list of points plotted). Then use the middle or right button, *with the cursor still on the graphics window*, to terminate the identification process.

Value

A vector of indices (into the original catalogue) of the points identified.

See Also

[epicentres](#), [identify](#)

Examples

```
data(NZ55)

a <- subsetrect(NZ55, minmag=6.5)
b <- epicentres.identify(a, mapname = "nz", pch = 1)
if (length(b) > 0) print(NZ55[b,])
```

Description

Estimate different parameter or parameters of Kagan distribution for a given data set.

Usage

```
est.kagan(Data, alpha=1, beta=0, gamma=7, theta=0.75, phi=2.4, tol=10^-3,
          Mag=TRUE, deltam=2)
```

Arguments

Data	Vector of magnitude or stress.
alpha	Parameter of Kagan distribution. Index of the power law.
beta	Parameter of Kagan distribution. Control the lower turning point of distribution.
gamma	Parameter of Kagan distribution. Control the upper turning point of the distribution.
theta	Parameter of magnitude.convert .
phi	Parameter of magnitude.convert .
tol	minimum step length of Newton-Raphson algorithm.
Mag	Flag indicating whether stress or magnitude is to be used.
deltam	Range for scanning over beta parameter.

Details

Data is given in magnitudes or stress. Under the condition of data following Kagan distribution, parameters of distribution could be estimated, Newton-Raphson algorithm and maximum likelihood method are used here.

Value

Vector of estimated (or fixed) parameters alpha (α), beta (β) and gamma (γ) of the Kagan distribution, the value of M_0 (minimum magnitude) determined from the data and loglikelihood value that those parameters correspond to.

Warning

When the parameters of distribution are all unknown, the estimated results are sometimes not very accurate.

Author(s)

Wang Lifeng, 2001

References

Cited references are listed on the [ssEDA-package](#) manual page.

See Also

[dkagan](#)

Examples

```
estimate.alph <- NULL
for (i in 1:100)
{
  # follow Kagan distribution, using default parameters.
  stress <- rkagan(1000, mag = FALSE) # simulate data set of stress which
  # when alpha is unknown.
  alpha <- est.kagan(stress, alpha = NA, Mag = FALSE) # estimate alpha,
  estimate.alph <- rbind(estimate.alph, alpha)
}

# Get distribution of alpha estimated from the 100 samples. This
# way, we could know possible distance between estimated one
# and real one.
hist(estimate.alph[, 1], xlab="Alpha", ylab="Frequency", main="")
box()
```

freq.cusum

Frequency Cusum Plot

Description

Draws a reverse cusum of event frequencies over time.

Usage

```
freq.cusum(events, delta=1, trainyears=7)
```

Arguments

events	defines the events to be plotted. It is an object of class "subset", generally created by subsetcircle , subsetpolygon , subsetrect or subsetsphere .
delta	number of months in each counted interval. Default is 1.
trainyears	number of years in the training period.

Details

The target value is the estimated mean frequency in the training period. The training period consists of the most recent years in the dataset specified by the argument `trainyears`, i.e. number of years. Hence the number of events in other years is compared to that in the training period.

A cusum is a time series of the cumulative sum of deviations from the target value. Generally this cumulative sum starts at zero at the start (or minimum) time, and proceeds to the right. This is a reverse cusum. It starts at zero at the most recent time and works backwards in time. It will return to zero at the start of the training period. As one scans the plot to the left, a decreasing line indicates that there is a deficit of events relative to the target, and an increasing line indicates that there are more events, on average, than that predicted by the target interval.

See Also

[depth.hist](#), [freq.magnitude](#), [magnitude.cusum](#), [magnitude.time](#), [epicentres](#), [timeplot](#), [multigraph](#)

Examples

```
# This example requires the NZ catalogue
if (require(ssNZ)){

  b <- subsetrect(NZ, minday=julian(1,1,1964), maxday=julian(1,1,1994),
                 mindepth=40, maxdepth=120, minmag=4)

  freq.cusum(b)
}
```

freq.magnitude

Frequency Magnitude (Gutenberg-Richter) Plot

Description

Frequency-magnitude plot of the selected events. That is, for a given magnitude m (x axis), the base 10 logarithm of the proportion (or frequency) of events with magnitude greater than or equal to m is given on the y axis. The (negative) slope is commonly referred to as the b -value.

Usage

```
freq.magnitude(events, delta=0.1, bvalue=NA, estimate="ml",
               plot=TRUE, minmag=NA, freq=FALSE)
```

Arguments

events	defines the events to be plotted. It is an object of class "subset", generally created by subsetcircle , subsetpolygon , subsetrect or subsetsphere .
delta	the interval (bin) widths.
bvalue	numeric scalar. If NA (default), the maximum likelihood estimate is calculated and displayed on the graph. If not NA, a line with the given b -value is plotted.
estimate	determines the estimation method used for the b -value. It takes the values "ml" (default) or "ls", meaning maximum likelihood or least squares, respectively. See "Details" below.
plot	logical, default is TRUE. Determines if a plot is produced.
minmag	magnitude threshold. This is <i>not</i> for event subsetting, and only relates to the manner in which the b -value calculations are done at the lower distribution boundary. It <i>must</i> less than or equal to the minimum magnitude of the analysed events. If NULL it will default to events\$minmag. If this is $-\text{Inf}$, it will use the minimum magnitude from the selected events. See "Details" for further information.
freq	logical, default is FALSE. Plot on the vertical axis event frequencies if TRUE, or proportions if FALSE.

Details

The plot is constructed by initially binning the data. The first bin has a left boundary given by `minmag`. Intervals (bins) of width `delta` are then constructed so as to span all data up to and including the event with the largest magnitude. Further note that the intervals are *left closed*. Since the vertical axis represents the “Number of Events $\geq m$ ” when `freq==TRUE`, then when m equals `minmag`, the plotted point is the total number in the dataset. If the interval widths are sufficiently fine so that the last interval only contains a single event with the largest magnitude, then the number on the vertical scale will be one. When `freq==FALSE` these numbers will be converted into proportions. Note that the plotted points are located at the left limit of the series of intervals. This can be seen by setting `delta` to a sufficiently large value so that one has relatively few bins.

The argument `minmag` is quite important in the estimation of the b -value. For example, if we select events using one of the subset functions and set `minmag` to 4.0, all events with magnitude ≥ 4.0 will be selected. Now, say that the data have been rounded to one decimal place. Then a magnitude 4.0 event may have had a magnitude anywhere between 3.95 and 4.05, so the effective magnitude cut-off is really 3.95. In this situation, the `minmag` argument should be set to reflect this situation.

Let M_o be the value of `minmag` and M_i be the magnitude of the i th event where $i = 1, \dots, n$. Since the Gutenberg-Richter relationship is equivalent to the event magnitudes having an exponential distribution, the maximum likelihood estimate of the b -value is

$$\frac{n}{\log(10) \sum_{i=1}^n (M_i - M_o)}.$$

This method is used when `estimate="ml"` (default).

Another included method of estimation is simple unweighted least squares, `estimate="ls"`, applied directly to the cumulative binned counts in the plot. This method is *statistically invalid*, and is only included here for comparative purposes. Only the slope is estimated as the intercept is known and fixed according to the above argument.

Value

The estimated b -value is returned if the function call is assigned to an object, otherwise NULL.

See Also

`depth.hist`, `freq.cusum`, `magnitude.cusum`, `magnitude.time`, `epicentres`, `timeplot`, `multigraph`

Examples

```
# This example requires the NZ catalogue
if (require(ssNZ)){

  b <- subsetrect(NZ, minday=julian(1,1,2000), maxday=julian(1,1,2005),
                 mindepth=40, maxdepth=120, minmag=4)

  freq.magnitude(b)
}
```

hemisphere	<i>Map of the Hemisphere</i>
------------	------------------------------

Description

Map of the hemisphere about a given point on earths surface.

Usage

```
hemisphere(longitude, latitude, plot.title="Azimuths Projection")
```

Arguments

longitude	longitude of centre point.
latitude	latitude of centre point.
plot.title	plot title. Default is "Azimuths Projection".

See Also

[projection](#)

Examples

```
# make plotting area square shape
par.reset <- par(no.readonly=TRUE)
par(pin=rep(min(par())$pin, 2))

hemisphere(170,-42, plot.title="Hemisphere Centred on NZ")

hemisphere(350,42, plot.title="The Other Hemisphere")

par(par.reset)
```

magnitude.contour	<i>Mean Magnitude Contours</i>
-------------------	--------------------------------

Description

This function calculates mean magnitude contours at a pre-specified depth.

Usage

```
magnitude.contour(events, h=c(0.3, 0.2, 10), depth=10)
```

Arguments

events	defines the events to be plotted. It is an object of class "subset", generally created by subsetcircle , subsetpolygon , subsetrect or subsetsphere .
h	a vector of length three. It specifies the bandwidths for the kernel function (see Details).
depth	specifies the longitude-latitude plane where the b-value contours are plotted.

Details

The mean magnitude contours are produced using local linear regression technique described in Ruppert and Wand (1994) using a standard trivariate normal kernel. The smoothness of the contours is controlled by the bandwidths of the kernel.

Value

NULL

Author(s)

Edwin Choi (ANU), 1997

References

Cited references are listed on the [ssEDA-package](#) manual page.

See Also

[bvalue.contour](#)

Examples

```
## Not run:
# This example requires the NZ catalogue
if (require(ssNZ)){

# Remember the parameters of the graphics device
par.reset <- par(no.readonly=TRUE)

# Make the Wellington Catalogue
as.catalogue(subsetrect(NZ, minlat=-42.2, maxlat=-40.5,
                        minlong=173.6, maxlong=176.0, minday=julian(1,1,1978)),
             catname="Wellington")

b <- subsetrect(Wellington, maxdepth=100, minmag=3,
               minlat=-42.2, maxlat=-40.5, minlong=173.6, maxlong=176.0,
               minday=julian(1,1,1980), maxday=julian(1,1,2000))

par(pty="s")
# NOTE: magnitude.contour() will not work if called directly.
# It is blocked because it uses as.catalogue() which writes into
```

```
# the user's directory. A *temporary* fix is to write the function to
# your local directory first, then source that code:
dump("magnitude.contour", file="magnitude.contour.R")
source("magnitude.contour.R")
magnitude.contour(b, h=c(0.25, 0.2, 2), depth=10)
if (require(mapdata)) mpnm <- "nzHires" else mpnm <- "nz"
map(database=mpnm, add=TRUE, col="green3")
par(par.reset)
}

## End(Not run)
```

magnitude.convert	<i>Magnitude-Moment Conversion</i>
-------------------	------------------------------------

Description

Converts magnitudes to Benioff moments and vice versa.

Usage

```
magnitude.convert(m, phi=0.75, B=2.4, inverse=FALSE)
```

Arguments

m	A vector to which the magnitude.convert function is applied.
phi	parameter of the Benioff Moment.
B	parameter of the Benioff Moment.
inverse	logical. If inverse is FALSE, the Benioff Moment is returned. If inverse is TRUE, the magnitude is returned.

Value

The Benioff Moment is defined as:

$$S(m) = 10^{\phi m}.$$

The inverse is:

$$S(s)^{-1} = \frac{\log 10(s) - B}{\phi}.$$

Author(s)

Alistair Merrifield, 1998

References

Cited references are listed on the [ssEDA-package](#) manual page.

See Also

[pkagan](#), [qkagan](#), [rkagan](#), [dkagan](#)

Examples

```
magnitudes <- seq(0, 11, length=100)
moments <- magnitude.convert(magnitudes)
magnitudes <- magnitude.convert(moments, inverse=TRUE)
```

magnitude.cusum	<i>Cusum Magnitude Plot</i>
-----------------	-----------------------------

Description

Draws a cusum of event magnitudes over time. The target value is the estimated mean magnitude in the training period.

Usage

```
magnitude.cusum(events, trainyears=7, at.breaks=NA)
```

Arguments

events	defines the events to be plotted. It is an object of class "subset", generally created by subsetcircle , subsetpolygon , subsetrect or subsetsphere .
trainyears	length of the training period in years.
at.breaks	positions of labelled tick marks on the third (top) axis. Default is January of each year.

See Also

[depth.hist](#), [freq.cusum](#), [freq.magnitude](#), [magnitude.time](#), [epicentres](#), [timeplot](#), [multigraph](#)

Examples

```
# This example requires the NZ catalogue
if (require(ssNZ)){

  b <- subsetrect(NZ, minday=julian(1,1,1964), maxday=julian(1,1,1994),
                 mindepth=40, maxdepth=120, minmag=4)

  magnitude.cusum(b)
}
```

magnitude.time	<i>Magnitude Time Plot</i>
----------------	----------------------------

Description

Plots event magnitudes over time.

Usage

```
magnitude.time(events, ylim=NULL, xlim=NULL, axes=TRUE, type="h", cex=1,
               labels=TRUE)
```

Arguments

events	defines the events to be plotted. It is an object of class "subset", generally created by subsetcircle , subsetpolygon , subsetrect or subsetsphere .
xlim	a vector giving the horizontal axis (time) limits. The default uses the values deduced from object events.
ylim	a vector giving the vertical axis (magnitude) limits. The default uses the value given by plot .
axes	logical, default is TRUE. Determines whether axes and tick marks are added.
type	character, default is "h", see information in par .
cex	Determines size of plotted symbols.
labels	logical, should the axes be labelled.

See Also

[depth.hist](#), [freq.cusum](#), [freq.magnitude](#), [magnitude.cusum](#), [epicentres](#), [timeplot](#), [multigraph](#)

Examples

```
# This example requires the NZ catalogue
if (require(ssNZ)){

# Make the Wellington Catalogue
as.catalogue(subsetrect(NZ, minlat=-42.2, maxlat=-40.5,
                      minlong=173.6, maxlong=176.0, minday=julian(1,1,1978)),
             catname="Wellington")

# Cape Palliser Earthquake Sequence
b <- subsetcircle(Wellington, centrelong=175.5, centrelat=-41.65,
                 maxradius=20, minday=julian(1,1,1990),
                 maxday=julian(1,1,1993), maxdepth=40)

# Showing all selected events
magnitude.time(b)
```



```

title(main="Cape Palliser Earthquake Sequence")

# Showing events with M >= 2.0
magnitude.time(b, ylim=c(1.9, 5.5))
title(main="Cape Palliser Earthquake Sequence")
}

```

major

Major Earthquake Events

Description

This page contains a listing of major earthquake events. They all [require](#) certain catalogues to be installed.

Examples

```

# Remember the parameters of the graphics device
# Run this first, else the plots will get progressively smaller

par.reset <- par(no.readonly=TRUE)

#-----
event <- "Sumatra (Nias) Earthquake - 28 March 2005"

if (require(ssPDE)){
if (require(mapdata)) mpnm <- "world2Hires" else mpnm <- "world2"

usr <- c(92, 104, -5, 7)
a <- subsetrect(PDE, minday=julian(3, 1, 2005),
                maxday=julian(8, 1, 2005), minlong=usr[1],
                maxlong=usr[2], minlat=usr[3], maxlat=usr[4],
                minmag=4)
epicentres(a, usr=usr, mapname=mpnm, magnitude=c(4, 5, 6, 7, 8),
            cex=c(0.2, 1, 2, 4), criteria=FALSE)
title(main=event)
par(par.reset)
magnitude.time(a)
title(main=event)
}

#-----
event <- "Phuket Thailand Earthquake - 26 Dec 2004"

if (require(ssPDE)){
if (require(mapdata)) mpnm <- "world2Hires" else mpnm <- "world2"

usr <- c(89, 105, 0, 16)
a <- subsetrect(PDE, minday=julian(12, 1, 2004),
                maxday=julian(3, 1, 2005), minlong=usr[1],

```

```

        maxlong=usr[2], minlat=usr[3], maxlat=usr[4],
        minmag=5)
epicentres(a, usr=usr, mapname=mpnm, magnitude=c(5, 6, 7, 8),
        cex=c(0.2, 1, 4), criteria=FALSE)
title(main=event)
par(par.reset)
magnitude.time(a)
title(main=event)
}

#-----
event <- "Parkfield California Earthquake - 28 Sept 2004"

if (require(ssSCEC)){
if (require(mapdata)) mpnm <- "world2Hires" else mpnm <- "world2"

usr <- c(238, 241, 35, 37)
a <- subsetrect(SCEC, minday=julian(9, 1, 2004),
        maxday=julian(1, 1, 2005), minlong=usr[1],
        maxlong=usr[2], minlat=usr[3], maxlat=usr[4],
        minmag=2)
epicentres(a, usr=usr, mapname=mpnm, magnitude=c(2, 3, 4, 5, 6, 7),
        cex=c(0.2, 0.5, 1, 3, 5), criteria=FALSE)
title(main=event)
par(par.reset)
magnitude.time(a)
title(main=event)
}

#-----
event <- "Alaska (Denali) Earthquake - 3 November 2002"

if (require(ssPDE)){
if (require(mapdata)) mpnm <- "world2Hires" else mpnm <- "world2"

usr <- c(200, 220, 55, 66)
a <- subsetrect(PDE, minday=julian(11, 1, 2002),
        maxday=julian(7, 1, 2003), minlong=usr[1],
        maxlong=usr[2], minlat=usr[3], maxlat=usr[4],
        minmag=4)
epicentres(a, usr=usr, mapname=mpnm, magnitude=c(4, 5, 6, 7, 8),
        cex=c(0.2, 1, 2, 4), criteria=FALSE)
title(main=event)
par(par.reset)
magnitude.time(a)
title(main=event)
}

#-----
event <- "Hector Mine California Earthquake - 16 October 1999"

if (require(ssSCEC)){
if (require(mapdata)) mpnm <- "world2Hires" else mpnm <- "world2"

```

```

usr <- c(240, 246, 32, 38)
a <- subsetrect(SCEC, minday=julian(10, 1, 1999),
               maxday=julian(7, 1, 2000), minlong=usr[1],
               maxlong=usr[2], minlat=usr[3], maxlat=usr[4],
               minmag=3)
epicentres(a, usr=usr, mapname=mpnm, magnitude=c(3, 5, 6, 7, 8),
           cex=c(0.2, 1, 2, 4), criteria=FALSE)
title(main=event)
par(par.reset)
magnitude.time(a)
title(main=event)
}

#-----
event <- "Kobe Japan Earthquake - 17 January 1995"

if (require(ssPDE)){
if (require(mapdata)) mpnm <- "world2Hires" else mpnm <- "world2"

usr <- c(133, 137, 32, 36)
a <- subsetrect(PDE, minday=julian(1, 10, 1995),
               maxday=julian(12, 31, 1995), minlong=usr[1],
               maxlong=usr[2], minlat=usr[3], maxlat=usr[4],
               minmag=4)
epicentres(a, usr=usr, mapname=mpnm, magnitude=c(4, 5, 6, 7),
           cex=c(1, 2, 3), criteria=FALSE)
title(main=event)
par(par.reset)
magnitude.time(a)
title(main=event)
}

#-----
event <- "Northridge California Earthquake - 17 January 1994"

if (require(ssSCEC)){
if (require(mapdata)) mpnm <- "world2Hires" else mpnm <- "world2"

usr <- c(240, 245, 33, 36)
a <- subsetrect(SCEC, minday=julian(1, 1, 1994),
               maxday=julian(7, 1, 1994), minlong=usr[1],
               maxlong=usr[2], minlat=usr[3], maxlat=usr[4],
               minmag=3)
epicentres(a, usr=usr, mapname=mpnm, magnitude=c(3, 5, 6, 7),
           cex=c(0.2, 1, 4), criteria=FALSE)
title(main=event)
par(par.reset)
magnitude.time(a)
title(main=event)
}

#-----

```

```

event <- "Landers California Earthquake - 28 June 1992"

if (require(ssSCEC)){
if (require(mapdata)) mpnm <- "world2Hires" else mpnm <- "world2"

usr <- c(240, 246, 32, 38)
a <- subsetrect(SCEC, minday=julian(6, 1, 1992),
                maxday=julian(1, 1, 1993), minlong=usr[1],
                maxlong=usr[2], minlat=usr[3], maxlat=usr[4],
                minmag=3)
epicentres(a, usr=usr, mapname=mpnm, magnitude=c(3, 5, 6, 7, 8),
            cex=c(0.2, 1, 2, 4), criteria=FALSE)
title(main=event)
par(par.reset)
magnitude.time(a)
title(main=event)
}

#-----
event <- "Loma Prieta Earthquake - 18 October 1989"

if (require(ssPDE)){
if (require(mapdata)) mpnm <- "world2Hires" else mpnm <- "world2"

usr <- c(237, 239, 36.5, 38.5)
a <- subsetrect(PDE, minday=julian(10, 1, 1989),
                maxday=julian(7, 1, 1990), minlong=usr[1],
                maxlong=usr[2], minlat=usr[3], maxlat=usr[4],
                minmag=3)
epicentres(a, usr=usr, mapname=mpnm, magnitude=c(3, 5, 6, 7),
            cex=c(0.2, 1, 4), criteria=FALSE)
title(main=event)
par(par.reset)
magnitude.time(a)
title(main=event)
}

#-----
event <- "Edgecumbe NZ Earthquake - 2 March 1987"

if (require(ssNZ)){
if (require(mapdata)) mpnm <- "nzHires" else mpnm <- "nz"

usr <- c(176, 178, -38.5, -36.5)
a <- subsetrect(NZ, minday=julian(1, 1, 1987),
                maxday=julian(1, 1, 1988), minlong=usr[1],
                maxlong=usr[2], minlat=usr[3], maxlat=usr[4],
                minmag=3)
epicentres(a, usr=usr, mapname=mpnm, magnitude=c(3, 5, 6, 7),
            cex=c(0.2, 1, 4), criteria=FALSE)
title(main=event)
par(par.reset)
magnitude.time(a)

```

```

title(main=event)
}

#-----
event <- "Tangshan China Earthquake - 26 July 1976"

if (require(ssPDE)){
if (require(mapdata)) mpnm <- "world2Hires" else mpnm <- "world2"

usr <- c(116, 120, 38, 41)
a <- subsetrect(PDE, minday=julian(7, 1, 1976),
               maxday=julian(1, 1, 1977), minlong=usr[1],
               maxlong=usr[2], minlat=usr[3], maxlat=usr[4],
               minmag=4)
epicentres(a, usr=usr, mapname=mpnm, magnitude=c(4, 5, 6, 7, 8),
           cex=c(0.2, 1, 2, 4), criteria=FALSE)
title(main=event)
par(par.reset)
magnitude.time(a)
title(main=event)
as.catalogue(a, catname="temp")
print(temp)
}

```

map1

Draw Geographical Map

Description

Draws a geographical map with a rectangular projection. The user specifies the exact boundaries (latitude and longitude) of the map. The map is plotted to the current graphics device. If one is not open, a device of the default type will be opened.

Usage

```
map1(mapname, usr, axes = TRUE, reset = TRUE, col = "gray35", lwd = 1)
```

Arguments

mapname	character string giving the map name. Low resolution maps contained in package maps are "nz" and "world2" (default). High resolution maps contained in the package mapdata are "nzHires", "world2Hires" and "chinaHires". The "2" on "world2" denotes the version where the longitudes are all positive.
usr	a vector of length 4 than defines boundary positions (longitudes and latitudes) of the axes, i.e. c(xmin, xmax, ymin, ymax).
axes	logical variable. If TRUE (default), axes are added to the map, FALSE otherwise.
reset	logical variable. If TRUE (default), the graphics device parameters are reset to their initial values, and FALSE will leave them the same as those used to plot the current map.

col	the colour of the map outline, specified either as a character string or the number representing the required colour in current palette.
lwd	the line width of the map outline.

Details

It is assumed that generally the user will want to add further features to the map. In this situation, the argument `reset` should be set to `FALSE`. In this situation the sizes of the margins and area within the axes will remain the same.

If one subsequently wants to put a different plot onto the graphics device, then the graphics parameters ([par](#)) would need to be reset. See Examples below.

Author(s)

David Harte, 2004

See Also

[map](#), [epicentres](#)

Examples

```
par.reset <- par(no.readonly=TRUE)

if (require(mapdata)) mpnm <- "nzHires" else mpnm <- "nz"
map1(mapname=mpnm, usr=c(166, 179, -48, -34), reset=FALSE)
title(main="New Zealand")
par(par.reset)

if (require(mapdata)) mpnm <- "world2Hires" else mpnm <- "world2"
map1(mapname=mpnm, usr=c(220, 320, 10, 80), reset=FALSE)
title(main="North America")
par(par.reset)

if (require(mapdata)) mpnm <- "world2Hires" else mpnm <- "world2"
map1(mapname=mpnm, usr=c(90, 170, -20, 30), reset=FALSE)
title(main="South-East Asia")
par(par.reset)

if (require(mapdata)) mpnm <- "worldHires" else mpnm <- "world"
map1(mapname=mpnm, usr=c(-12, 40, 34, 65), reset=FALSE)
title(main="Europe")
par(par.reset)

if (require(mapdata)) mpnm <- "worldHires" else mpnm <- "world"
map1(mapname=mpnm, usr=c(-20, 60, -40, 45), reset=FALSE)
title(main="Africa")
par(par.reset)
```

multigraph

Plot Multiple Graphs

Description

Graph of multiple plots of events satisfying selection criteria.

Usage

```
multigraph(events, plots, ncols=1, title="", criteria=TRUE)
```

Arguments

events	defines the events to be plotted. It is an object of class "subset", generally created by subsetcircle , subsetpolygon , subsetrect or subsetsphere .
plots	list object of plot functions. Plot functions that can be included are those with the single events argument. Some of these are: <code>depth.hist</code> , <code>freq.cusum</code> , <code>freq.magnitude</code> , <code>magnitude.cusum</code> , <code>magnitude.time</code> , <code>timeplot</code> .
ncols	number of columns. Graphs are laid out in a matrix format, positions being filled row by row.
title	overall title for the collection of all graphs.
criteria	logical, default is TRUE. Determines whether the subsetting criteria is written at the bottom of the page.

See Also

[depth.hist](#), [freq.cusum](#), [freq.magnitude](#), [magnitude.cusum](#), [magnitude.time](#), [epicentres](#), [timeplot](#)

Examples

```
# This example requires the NZ catalogue
if (require(ssNZ)){

  b <- subsetrect(NZ, minday=julian(1,1,1964), maxday=julian(1,1,1994),
                 mindepth=40, maxdepth=120, minmag=4)

  multigraph(b,list(depth.hist, timeplot, magnitude.time,
                    freq.magnitude, freq.cusum, magnitude.cusum),
             ncols=3, title="Catalogue Completeness Analysis")
}
```

plot.subset

Method for Generic Function Plot

Description

Plots four summary graphs for the selected subset: frequency-magnitude plot, depth histogram, counts by year, and magnitude-time plot.

Usage

```
## S3 method for class 'subset'
plot(x, ...)
```

Arguments

x defines the events to be plotted. It is an object with class "subset" created by [subsetcircle](#), [subsetpolygon](#), [subsetrect](#) or [subsetsphere](#).

... other options for plotting "subset" objects.

Value

NULL

See Also

[subsetcircle](#), [subsetpolygon](#), [subsetrect](#), [subsetsphere](#)

Examples

```
data(NZ55)
a <- subsetrect(NZ55, minday=julian(1,1,1970))
plot(a)
```

rotation

Rotates Events to View Plate Boundary

Description

Plots a cross section of depth for the selected events. Depth is on the vertical axis, and the view is theta degrees west of north.

Usage

```
rotation(events, theta=0, km=TRUE, criteria=TRUE, zero=NULL,
          ylim=NULL)
```


Arguments

events	defines the events to be plotted. It is an object of class "subset", generally created by subsetcircle , subsetpolygon , subsetrect or subsetsphere .
theta	number of degrees of the direction of view from north, positive to the west. For example, theta=0 is viewing to the north (default) and for theta=-45 one would be viewing towards the NE.
km	if TRUE units of kilometres are used on the horizontal scale, if FALSE units of degrees are used.
criteria	boolean variable. If TRUE (default) the subsetting parameters are printed below the graph.
zero	a vector of length 2 containing a longitude and latitude. This could represent the location of an event of interest. The distances on the transformed scale (horizontal axis), when km == TRUE, are relative to the transformation of this point. The default is that "zero" is the mid-point of the range.
ylim	a vector giving the vertical axis (depth) limits. The default uses the value given by plot .

See Also

[depth.hist](#), [epicentres](#), [freq.cusum](#), [freq.magnitude](#), [magnitude.cusum](#),
[magnitude.time](#), [timeplot](#), [multigraph](#)

Examples

```
# These examples require the NZ catalogue
if (require(ssNZ)){
  b <- subsetrect(NZ, minlong=170, maxlong=180, minlat=-43,
                 maxlat=-35, minmag=3, minday=julian(1,1,2000),
                 maxday=julian(1,1,2005))

  rotation(b, theta=-47, ylim=c(-350, 0), zero=c(180, -41))
  title(main="NZ Plate Boundary")
}

#-----

# Make the Wellington Catalogue
if (require(ssNZ)){
  as.catalogue(subsetrect(NZ, minlat=-42.2, maxlat=-40.5,
                        minlong=173.6, maxlong=176.0, minday=julian(1,1,1978)),
              catname="Wellington")

  b <- subsetrect(Wellington, minlong=173.6, maxlong=176.1, minlat=-42.0,
                 maxlat=-40.3, minmag=2, minday=julian(1,1,1978),
                 maxday=julian(1,1,1992), maxdepth=200)

  rotation(b, theta=-40, km=FALSE)
  title(main="Wellington Plate Boundary")
}
```

threeD

*Dynamic 3D Plot of Earthquake Hypocentres***Description**

Dynamic three dimensional plot of earthquake hypocentres (longitude, latitude and depth) locations.

Usage

```
threeD(events)
```

Arguments

events	defines the events to be plotted. It is an object of class "subset", generally created by subsetcircle , subsetpolygon , subsetrect or subsetsphere .
--------	---

Details

Produces an interactive plot of the earthquake hypocentres with a GUI control panel. We briefly explain its operation using the example given in the "Examples" section below.

Do the initial standard plot of latitude vs depth. Events close to the surface are at the top, and deep events are at the bottom. Note the lines of shallow events. They are at 5km, 12km and 33km, and represent events where the depth is somewhat uncertain.

Running the command `threeD(a)`, a graph will appear with a separate GUI control panel. The windows can be enlarged by stretching in the usual way. Initially click "View" on the Control Panel then select "XY Plot"; on the graphics window, click "Options", and tick "Show Axes". On the right of the control panel are the three spatial variables in the dataset. Click "X" next to longitude and "Y" next to latitude. This gives an epicentral plot of the events. Now click "Y" next to depth. This will produce the same plot done initially.

Now we will rotate this plot horizontally while keeping the shallow events at the top and deep at the bottom. Click "View" then "Rotation". It rotates things in a rather random manner. Notice the "Clock" in the bottom left which indicates the view perspective. To gain control, click "Pause". Now click "Manip" and then the one handed clock face next to "depth" in the control panel. Under "Manual manipulation" select "Angular". On the graph use the mouse to make the hand on the "Clock" representing "depth" point upwards. Now select "Radial" under "Manual manipulation" and make the "depth" hand on the clock as long as it will go (out to edge of circle). You can determine the correct point when you start to see the events at 5km, 12km and 33km nicely lined up. You may need to iterate back and forth between "Radial" and "Angular" doing fine adjustments. Then set "Manual manipulation" on "Horiz", and then drag the pointer over the graph. This will keep the depth fixed but rotate the latitude and longitude.

Rotate the points until the clock hand representing longitude points to the right and is about twice as long as that representing latitude pointing to the left. Drag back and forth near this point until the points line up as close as possible. You should observe a relatively narrow band of point locations initially curving slightly to the left, and then almost straight down. These are the earthquake event locations on/close to the tectonic plate boundary of the Pacific Plate (right) subducting the

Australian Plate (left) beneath the North Island of New Zealand. The view perspective is looking through the Earth's crust in the direction of the tectonic boundary, roughly in a NE direction (towards Tonga). By clicking "Interaction" and then "Scale", you can zoom and pan over the graph. Other more general features of the GUI controls are explained in the GGobi Manual (<http://www.ggobi.org/docs/manual.pdf>).

Installing the rggobi Package

This function uses the R function `ggobi` to display the data using a dynamic graphics interface. The R function `ggobi` is contained in the `rggobi` package. The `rggobi` package requires the R package `RGtk2`. Installation of some system software that the `rggobi` and `RGtk2` packages require is not very straight forward. If your distribution of Linux offers a compiled binary for the two R packages `rggobi` and `RGtk2`, then installing these will automatically take care of the other system software dependencies.

The `rggobi` package requires installation of the *GGobi* system software (external to R). The *GGobi* system software can be downloaded from <http://www.ggobi.org>. Further, the R package `RGtk2` package has various GTK system requirements.

In Debian Jessie (Linux): initially install the following Debian packages:

```
sudo apt-get install libgtk2.0-dev libxml2-dev ggobi
```

The packages `libgtk2.0-dev` and `libxml2-dev` are required to compile the R packages `RGtk2` and `rggobi`, respectively. Then within R

```
install.packages("rggobi")
```

This will also install the dependency `RGtk2`.

The Debian Jessie repository does contain a binary package `r-cran-rgtk2` (compiled prior to Jessie's release). Using this produced dependency incompatibilities with `rggobi`. Sometimes problematic system dependencies (e.g. wrong versions) in Debian can be overcome by downloading the package source code, and then compiling it locally

```
sudo apt-get --compile source r-cran-rgtk2
```

Windows 10: initially install the R package `RGtk2`:

```
install.packages("RGtk2", type="win.binary")
```

This will also download and install system software from http://ftp.gnome.org/pub/gnome/binaries/win64/gtk+/2.22/gtk+-bundle_2.22.1-20101229_win64.zip. Once completed, quit R, then restart R, and check that the package will load:

```
library(RGtk2)
```

Then quit R. Next download the *GGobi* system software (exe file) from <http://www.ggobi.org/downloads/>. Install by executing the downloaded exe file, then in R:

```
install.packages("rggobi", type="win.binary")
```

Then quit R. Restart R, and check that it loads the package:

```
library(rggobi)
```

See Also[rotation](#)**Examples**

```
## Not run:
if (require(ssNZ) & require(rggobi)){
  #   This example also requires the NZ catalogue

  a <- subsetrect(NZ, minlat=-42, maxlat=-37, minmag=4,
                 minlong=171.5, maxlong=181, maxdepth=400,
                 minday=julian(1,1,1995),
                 maxday=julian(1,1,2005))

  plot(NZ$longitude[a$indices], -NZ$depth[a$indices])

  threeD(a)
}

## End(Not run)
```

timeplot

*Plots Event Frequencies by Time***Description**

Plots a histogram of the number of selected events by time interval, either months or years.

Usage

```
timeplot(events, yearly=TRUE, smoothline=FALSE, ymax=max(y)*1.05,
         col=c("grey80", "grey50"))
```

Arguments

events	defines the events to be plotted. It is an object of class "subset", generally created by subsetcircle , subsetpolygon , subsetrect or subsetsphere .
yearly	logical. Should each bar represent annual TRUE or monthly FALSE counts. Default is TRUE.
smoothline	logical. Plot a smoothed line over the counts. Default is FALSE. The smoothing is done using the function supsmu .
ymax	numeric. Sets the upper limit of the <i>y</i> axis for the bar plot, default is about 5% greater than the tallest bar.
col	colour of the bars. If annual, then only the first colour is used. If monthly, then each year alternates between the two colours. Default is <code>c("grey80", "grey50")</code> .

See Also

[depth.hist](#), [freq.cusum](#), [freq.magnitude](#), [magnitude.cusum](#), [magnitude.time](#), [epicentres](#), [rotation](#), [multigraph](#)

Examples

```
# This example requires the NZ catalogue
if (require(ssNZ)){

  b <- subsetrect(NZ, minday=julian(1,1,1961), minmag=4,
                 minlat=-50, maxlat=-33.5, minlong=165, maxlong=180)
  timeplot(b)
  title(main=expression(paste("Events in NZ With ", M[L] >= 4)))
}
```

worldLores

Low Resolution World Map

Description

This list object has components `longitude` and `latitude` which provide a low resolution version of the world map. It is used by the function [hemisphere](#).

Usage

```
worldLores
```

Examples

```
# make plotting area square shape
par.reset <- par(no.readonly=TRUE)
par(pin=rep(min(par())$pin, 2))

hemisphere(longitude=270, latitude=0, plot.title="Western Hemisphere")
hemisphere(longitude=90, latitude=0, plot.title="Eastern Hemisphere")

par(par.reset)
```

Index

- *Topic **datasets**
 - worldLores, [37](#)
- *Topic **distribution**
 - dkagan, [9](#)
 - est.kagan, [15](#)
- *Topic **documentation**
 - Change Log, [4](#)
 - major, [25](#)
 - ssEDA-package, [2](#)
- *Topic **dynamic**
 - threeD, [34](#)
- *Topic **hplot**
 - bvalue.contour, [3](#)
 - depth.hist, [8](#)
 - epicentres, [10](#)
 - freq.cusum, [17](#)
 - freq.magnitude, [18](#)
 - hemisphere, [20](#)
 - magnitude.contour, [20](#)
 - magnitude.cusum, [23](#)
 - magnitude.time, [24](#)
 - map1, [29](#)
 - multigraph, [31](#)
 - plot.subset, [32](#)
 - rotation, [32](#)
 - timeplot, [36](#)
- *Topic **iplot**
 - epicentres.identify, [14](#)
- *Topic **manip**
 - magnitude.convert, [22](#)
- *Topic **methods**
 - plot.subset, [32](#)
- bvalue.contour, [2](#), [3](#), [5–7](#), [21](#)
- Change Log, [4](#)
- Changes (Change Log), [4](#)
- depth.hist, [2](#), [5](#), [6](#), [8](#), [11](#), [18](#), [19](#), [23](#), [24](#), [31](#), [33](#), [37](#)
- dkagan, [2](#), [9](#), [16](#), [23](#)
- epicentres, [2](#), [5–8](#), [10](#), [15](#), [18](#), [19](#), [23](#), [24](#), [30](#), [31](#), [33](#), [37](#)
- epicentres.identify, [2](#), [5](#), [7](#), [14](#)
- est.kagan, [2](#), [15](#)
- freq.cusum, [2](#), [5–8](#), [11](#), [17](#), [19](#), [23](#), [24](#), [31](#), [33](#), [37](#)
- freq.magnitude, [2](#), [6–8](#), [11](#), [18](#), [18](#), [23](#), [24](#), [31](#), [33](#), [37](#)
- ggobi, [35](#)
- hemisphere, [2](#), [5](#), [7](#), [20](#), [37](#)
- hist, [6](#)
- identify, [15](#)
- magnitude.contour, [2](#), [4–7](#), [20](#)
- magnitude.convert, [2](#), [9](#), [10](#), [16](#), [22](#), [22](#)
- magnitude.cusum, [2](#), [5–8](#), [11](#), [18](#), [19](#), [23](#), [24](#), [31](#), [33](#), [37](#)
- magnitude.time, [2](#), [5](#), [7](#), [8](#), [11](#), [18](#), [19](#), [23](#), [24](#), [31](#), [33](#), [37](#)
- major, [2](#), [6](#), [25](#)
- map, [6](#), [30](#)
- map1, [2](#), [5–8](#), [29](#)
- multigraph, [2](#), [5](#), [8](#), [11](#), [18](#), [19](#), [23](#), [24](#), [31](#), [33](#), [37](#)
- par, [15](#), [24](#), [30](#)
- pkagan, [2](#), [23](#)
- pkagan (dkagan), [9](#)
- plot, [7](#), [24](#), [33](#)
- plot.subset, [2](#), [5](#), [32](#)
- projection, [20](#)
- qkagan, [2](#), [23](#)
- qkagan (dkagan), [9](#)
- require, [25](#)

rkagan, [2](#), [23](#)
rkagan (dkagan), [9](#)
rotation, [2](#), [5–7](#), [32](#), [36](#), [37](#)

ssEDA-package, [2](#), [4](#), [7](#), [10](#), [16](#), [21](#), [22](#)
subsetcircle, [3](#), [6](#), [8](#), [10](#), [15](#), [17](#), [18](#), [21](#), [23](#),
 [24](#), [31–34](#), [36](#)
subsetpolygon, [3](#), [6](#), [8](#), [10](#), [15](#), [17](#), [18](#), [21](#), [23](#),
 [24](#), [31–34](#), [36](#)
subsetrect, [3](#), [6](#), [8](#), [10](#), [15](#), [17](#), [18](#), [21](#), [23](#), [24](#),
 [31–34](#), [36](#)
subsetsphere, [3](#), [6](#), [8](#), [10](#), [15](#), [17](#), [18](#), [21](#), [23](#),
 [24](#), [31–34](#), [36](#)
supsmu, [36](#)

threeD, [2](#), [7](#), [34](#)
timeplot, [2](#), [5](#), [6](#), [8](#), [11](#), [18](#), [19](#), [23](#), [24](#), [31](#), [33](#),
 [36](#)

world, [5](#)
world2, [5](#)
worldLores, [2](#), [5](#), [6](#), [37](#)