Package 'ssBase'

28 September 2019

Title Base Functions for SSLib

Date 2019-09-22

Version 2.3-17

Author David Harte

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

Description Contains base functions for SSLib, including ``datetimes'' format, and various functions to subset earthquake catalogues.

Imports chron

License GPL (>=2)

LazyData no

ZipData no

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

NeedsCompilation yes

R topics documented:

`ssBase-package` *Overview of Package ssBase*

Description

The **ssBase** package contains functions to subset earthquake catalogues and various base functions used by other SSLib packages. It also contains some small datasets used in examples on manual pages.

Standard Format for Earthquake Catalogues

The SSLib catalogue object has the following attributes.

`class`: has values `"data.frame"` and `"catalogue"`.

`catname`: character, containing the name of the object.

`row.names`: as required by a dataframe.

All earthquake catalogues in SSLib must contain the following variables in the described format. They can also contain other variables, e.g. event identifier, etc.

`time`: Time of the event, given as the number of days, and fractions of, from 00:00:00 hrs on 1970-01-01 (origin). It has class `datetimes`, and an attribute `"dp.second"` which specifies the number of decimal places to use when printing times, e.g. one decimal place `DD-MMM-YYYY hh-mm-ss.s`. The time `"origin"` is also given as an attribute.

`longitude`: Longitude of the event specified as degrees east between 0° and 360°. That is, the discontinuity occurs at 0° not 180°E.

`latitude`: Latitude of the event specified as the degrees north between −90° and 90°. Events in the southern hemisphere have a negative value.

`depth`: Depth of the event in kilometres. Generally positive, meaning the depth below mean sea level. A negative value would be interpreted as above sea level.

Most catalogues also contain the following.

`missing.time`**:** Character vector, same length as `time`, indicating the degree of missing values in `time`. The original downloaded catalogues generally contain separate variables for year, month, day, hour, minute and second. For some events, one or more of these values are missing. In these cases, `missing.time` contains the first of `"Y"`, `"M"`, `"D"`, `"h"`, `"m"` or `"s"`, representing the largest missing component. Otherwise it will be `""`. In cases where there were missing components, the variable `time` will be calculated by using a one in the case of a missing month or day, and a zero for a missing hour, minute or second.

Small Example Catalogues

`NZ55`, `Palliser`

Main Tasks Performed by the Package

The main tasks performed by the package are listed below.

Subsetting of Catalogue Data: See `subsetrect`, `subsetcircle`, `subsetsphere`, `subsetpolygon`, `print.subset`.

Making Sub-Catalogues: See `as.catalogue`, `c.catalogue`.

Support Functions for Date/Time: See `datetimes`, `format.datetimes`, `print.datetimes`, `[.datetimes`, `days1`, `months1`, `years1`, `hrs.mins.secs`.

Miscellaneous Support Functions for SSLib: `arcdist`, `epi.circle`, `lattice`, `lattice.retrieve`, `projection`.

SSLib Web Page

`http://www.statsresearch.co.nz/dsh/sslib/`

References

Brownrigg, R.; Harte, D.S. (2005). Using R for statistical seismology. *R News* **5(1)**, 31–35. `https://cran.r-project.org/doc/Rnews/Rnews_2005-1.pdf`

`arcdist` *Arc Distance Between Epicentral Locations*

Description

Calculates the arc-distance on the spherical surface between one epicentral location and a set of other locations.

Usage

```
arcdist(point1, points2, radius = 6371)
```

Arguments

point1	a vector containing the longitude and latitude of the "central" point.
points2	a matrix containing the other points, one in each row. The first column contains the longitudes and the second contains the latitudes.
radius	the radius of the sphere, default is 6371.

Value

A vector of arclengths the same length as number of rows in points2.

Author(s)

David Harte, 2002

See Also

epi.circle, epicentres

Examples

```
x <- arcdist(c(174, -42), cbind(c(174, 174, 175), c(-42, -43, -43)))
print(x)
```

---

as.catalogue *Make a Catalogue Object*

Description

Makes a catalogue from an object with class "subset", "data.frame" or "list".

Usage

```
as.catalogue(x, catname="", dp.second=1, pos=1)
```

Arguments

x	an object with class "subset" (i.e. created by subsetcircle, subsetpolygon, subsetrect or subsetsphere), class "list", or class "data.frame" (see Details).
catname	a character string giving the required name of the catalogue.
dp.second	only used if x is a list with NULL class. The number of decimal places in the second component of time.
pos	the environment into which the catalogue is written, see assign for further discussion. Note the default here, pos=1, is different to that in assign and is for legacy reasons. This will be changed in the future.

Details

When the input object is a list or dataframe, it will generally be that data have been read from a text file using the scan or read.csv, respectively. The data should at least contain the variables year, month, day, hour, minute, and second. These will be replaced in the resulting catalogue with one variable called time, being the number of days (and fractions) from some origin.

Value

NULL. The assignment takes place within the function.

Author(s)

David Harte, 2000

Examples

```
#   Creating a subcatalogue from a larger catalogue
data(NZ55)
y <- subsetrect(NZ55, minmag=7)
as.catalogue(y, catname="NZ7")
print(NZ7)

#   Reading Data From Disk and Creating a Catalogue
#   Say the following events are in a disk file called "events.dat"
#   -41.76,172.04,Westport,12,6.7,23,05,1968,17,24,17.4
#   -34.94,179.30,Kermadec Trench,297,6.8,08,01,1970,17,12,36.6
#   -39.13,175.18,National Park,173,7.0,05,01,1973,13,54,27.6
#   -41.61,173.65,Marlborough,84,6.7,27,05,1992,22,30,36.1
#   -45.21,166.71,Secretary Island,5,6.7,10,08,1993,00,51,51.6
#   -43.01,171.46,Arthurs Pass,11,6.7,18,06,1994,03,25,15.2
#   -37.65,179.49,East Cape,12,7.0,05,02,1995,22,51,02.3

#   Typically, we would use
#   scan function to read the data from a disk file as follows
#   y <- scan("events.dat", sep=",", what=list(latitude=0,
#              longitude=0, event="", depth=0,
#              magnitude=0, day=0, month=0,
#              year=0, hour=0, minute=0, second=0))
#  The list below will have the same format

y <- list(latitude=c(-41.76, -34.94, -39.13, -41.61, -45.21, -43.01, -37.65),
          longitude=c(172.04, 179.30, 175.18, 173.65, 166.71, 171.46, 179.49),
          event=c("Westport", "Kermadec Trench", "National Park", "Marlborough",
                  "Secretary Island", "Arthurs Pass", "East Cape"),
          depth=c(12, 297, 173, 84, 5, 11, 12),
          magnitude=c(6.7, 6.8, 7.0, 6.7, 6.7, 6.7, 7.0),
          day=c(23, 08, 05, 27, 10, 18, 05),
          month=c(05, 01, 01, 05, 08, 06, 02),
          year=c(1968, 1970, 1973, 1992, 1993, 1994, 1995),
          hour=c(17, 17, 13, 22, 00, 03, 22),
          minute=c(24, 12, 54, 30, 51, 25, 51),
          second=c(17.4, 36.6, 27.6, 36.1, 51.6, 15.2, 02.3))
```

```
as.catalogue(y, catname="my.cat", dp.second=1)
print(my.cat)
```

`c.catalogue` *Combine Catalogues*

Description

Combines two catalogues together to form a new catalogue. Provides a method for the generic function `c`.

Usage

```
## S3 method for class 'catalogue'
c(x, y, catname,
  vars=c("time", "longitude", "latitude", "depth", "magnitude"),
  ..., recursive = FALSE, pos = 1)
```

Arguments

`x`	catalogue name.
`y`	catalogue name.
`catname`	name of the new combined catalogue.
`vars`	variable names to included in the combined catalogue. These variables must exist in catalogues `x` and `y`. By default, the names are `c("time","longitude","latitude","depth","magnitude")`.
`...`	not used, required for compatibility with generic function `c`.
`recursive`	not used, required for compatibility with generic function `c`.
`pos`	the environment into which the catalogue is written, see `assign` for further discussion. Note the default here, `pos=1`, is different to that in `assign` and is for legacy reasons. This will be changed in the future.

Details

The two catalogues are combined using `rbind.data.frame`. The row names of the events are prefixed with "x" for those from catalogue `x` and "y" for those from catalogue `y`. They are also sequentially numbered. The events are also sorted by time.

Value

`NULL`. The assignment takes place within the function.

See Also

`as.catalogue`, `rbind.data.frame`

Examples

```
data(NZ55)
c(NZ55[1:10,], NZ55[101:120,], catname="x",
  vars=c("time", "longitude", "magnitude"))

print(x)
```

`Change Log` *Changes Made to the Package*

Description

This page contains a list of recent changes made to the package, and known general problems.

Details

1. The map plotting option in `lattice` has been removed, along with the default values for `alpha`, `d`, `centrelong` and `centrelat` in the function call. These must now be set explicitly. (March 2003)
2. The function `subcatalogue` has been deleted. It is superseded by `as.catalogue`. (June 2003)
3. The function `print.cat` has been renamed to `write.catalogue`. (June 2003)
4. Fixed inconsistencies between generic and method functions. (June 2003)
5. The function `c.cat` has been renamed to `c.catalogue` providing a "combine" method for objects with class `"catalogue"`. (June 2003)
6. `c.catalogue`: argument `name` has been changed to `catname`. (June 2003)
7. The function `plot.subset` has been moved to the ssEDA package. (June 2003)
8. `subset.circle`, `subset.polygon`, `subset.rect`, `subset.sphere`: argument `na.rm` added. Its effect has been described under 'Details' of the documentation of each function. (June 2003)
9. `library.version`: updated. (August 2003)
10. `library.version`: version above included a stray CR causing a syntax error. (September 2003)
11. `Palliser` Catalogue added. (September 2003)
12. `as.catalogue` modified to be compatible with R version 1.7.1. A list now has class `"list"`. (October 2003)
13. `ReadNZ`, `ReadPDE` and `ReadWellington` have been deleted, now part of respective packages. (October 2003)
14. `days1` and `projection`: links to functions in other packages in manual pages modified to include the package name. (November 2003)
15. Minor documentation formatting changes, mainly to use `\dQuote` and `\sQuote`. (January 2004)

16. summary.catalogue: gave a warning message if all values of depth were missing; fixed. (February 2004)
17. epi.circle: argument plotit removed. (February 2004)
18. Catalogue Read functions has been removed from this package, with each catalogue package containing its "read" function. (March 2004)
19. File '/exec/version.sh' removed from package. (March 2004)
20. library.version: documentation mismatch with code corrected. (7 May 2004)
21. subset.polygon: PACKAGE="ssBase" has been added to the .Fortran("polyse",...) call within the function. (7 May 2004)
22. Palliser, NZ55: in earlier package versions these datasets were read from a text file creating the rda file at the time data(Palliser) or data(NZ55) were ran. This method caused a syntax error in R 2.0. Now this file is created when the package is installed, similar to the catalogue packages. (22 Nov 2004)
23. summary.subset: syntax error in R 2.0 caused by multiple inequalities in an expression. A phantom() has been added to the initial if (plot == TRUE) statement. (22 Nov 2004)
24. subset.circle, subset.polygon, subset.sphere: documentation formatting changes to use \eqn. (16 Feb 2005)
25. arcdist: new function. (16 Feb 2005)
26. lattice: rearrange function code. (18 Feb 2005)
27. write.catalogue: writing loop eliminated, and use made of R functions format and formatC. (20 Feb 2005)
28. c.catalogue: function rewritten, user can now select a subset of variables to include. (20 Feb 2005)
29. subset.circle, subset.polygon, subset.rect, subset.sphere: documentation for arguments minday and maxday clarified. (April 2005)
30. days1, datetimes: clearer explanation on the manual pages of the difference between these functions and similar functions in the **chron** package. (29 April 2005)
31. In src/Makefile.win there were three occurrences of "$(RM) -f", each of were replaced by "$(RM)". The in-built definition for RM already contains the "-f". (20 June 2005)
32. Package vignettes added. (13 Aug 2005)
33. days1, months1, years1: the **chron** package implemented a namespace in version 2.3-0. To call the chron function origin, one now needs chron:::origin; though here have replaced with attr(x,"origin"). (28 Nov 2005)
34. days1: error in example fixed ("1959" to "1969"). (28 Nov 2005)
35. More problems with Windows! New version of 'src/Makefile.win'. (16 Dec 2005)
36. Modify 'src/Makefile.win' (missing tabs). (17 Dec 2005)
37. lattice: d included in output list. (4 May 2006)
38. File 'R/NZ55.txt' deleted, is in 'src/NZ55.txt'. (12 May 2006)
39. File 'R/[.datetimes.R' renamed to 'R/extract.datetimes.R'. (12 May 2006)
40. In 'src/Makefile.win' line containing DLLLIBS = -L$(RHOME)/lib -L$(RHOME)/src/gnuwin32 -lR deleted. (10 Jul 2007)

41. `c.catalogue`: arguments `...` and `recursive` added for compatibility with generic function `c`. (2 Aug 2007)
42. The functions `subset.rect`, `subset.circle`, `subset.polygon`, and `subset.sphere` are deprecated and have been renamed to `subsetrect`, `subsetcircle`, `subsetpolygon`, and `subsetsphere`, respectively. Their names conflict with the generic R function `subset`. (5 Nov 2007)
43. `subset.rect`, `subset.polygon`, and `subset.sphere`: correct incorrect information in the deprecated warning message. (21 Nov 2007)
44. Removal of LaTeX markups from DESCRIPTION file. (31 May 2009)
45. Remove hyperlinks to the deprecated functions `subset.rect`, `subset.circle`, `subset.polygon`, and `subset.sphere`. (12 Feb 2010)
46. Add CITATION file. (24 Sep 2010)
47. Changes to `/src/Makefile.win` so that it builds in Windows R 2.12.0. (21 Oct 2010)
48. Move example datasets (Rda file) into Data directory. (4 Dec 2010)
49. Implement NAMESPACE and remove file `/R/zzz.R`. (4 Nov 2011)
50. `lattice.retrieve`: remove `synopsis` on manual page. (14 Apr 2013)
51. `subsetrect`, `subsetcircle`, `subsetpolygon`, and `subsetsphere`: correct documentation error, `x` is not the catalogue name but a catalogue data object. (23 Jan 2014)
52. `as.catalogue`: the input argument `x` can now also be a dataframe. (16 Jul 2014)
53. `subsetrect`, `subsetcircle`, `subsetpolygon`, and `subsetsphere`: redo documentation as one manual page. (02 Jul 2015)
54. Fix NOTES in `R CMD check --as-cran`: `Found no calls to: 'R_registerRoutines', 'R_useDynamicSymbols'`. (18 Jun 2017)
55. Include overview topic ssBase-package. (18 Aug 2017)
56. Minor wording corrections in documentation. (25 Jan 2018)
57. `src/poly.f`: Remove commented write statements, reformat source code. (30 Apr 2018)
58. `as.catalogue`, `c.catalogue`: Add argument with default `pos=1`. This should have the same effect as before, to write the new catalogue into the user global environment. This assignment method is a legacy behaviour, not such a good idea, and it should really be written into a local environment if possible. (13 Jun 2018)
59. `as.catalogue`, `c.catalogue`: Writes a warning if `pos=1`, see above. (11 Jul 2018)
60. Rebuild package in R version 3.6.1. (22 Sep 2019)

Future Developments

1. Make `'datetimes'` compatible with base software provided by R. Still insufficient flexibility in the **chron** functions.
2. Write a function `subset.catalogue` as a method for `subset`. It would involve `subsetcircle`, `subsetpolygon`, `subsetrect`, and `subsetsphere`.

`datetimes` *Calculate Dates and Times*

Description

Calculates the number of days and fractions thereof from the 1 January 1970.

Usage

```
datetimes(year, month, day, hour, minute, second=0, dp.second=NA,
          missing = FALSE)
```

Arguments

`year`	an integer, e.g. 1998.
`month`	an integer with values 1, 2, . . . , 12.
`day`	an integer giving the day of the month, e.g. 1, 2, . . . , 31.
`hour`	an integer giving the hour of the day, i.e. 0, 1, 2, . . . , 23.
`minute`	an integer giving the minute of the hour, i.e. 0, 1, 2, . . . , 59.
`second`	a numeric vector giving the seconds, e.g. 32.45. The number of decimal places should be consistent with dp.second below. Default is zero.
`dp.second`	an integer giving the number of decimal places specified in second. Default is `NA`, i.e. not specified. In this case zero is used in all calculations.
`missing`	logical, default is `FALSE`. Indicates whether information about the extent of missing values should be passed out of the function.

Value

If `missing` == `FALSE`, then only a datetimes object is returned. If `TRUE`, a list object which contains a datetimes object called `ti` and an object called `missing` is returned.

A datetimes object is a numeric vector with class `"datetimes"`, and values for each date that are the number of days and fractions thereof from 00:00:00hrs on 1 January 1970. This object also has attributes `"origin"` which specifies the origin date, and `"dp.second"` which specifies the accuracy of the variable `second`. This information will determine the number of decimal places used to print the object. By default, the object will be printed in the format DDMMMYYYY hh:mm:ss.s, where the number of decimal places is specified as above.

The vector `missing` is character, taking values `"Y"`, `"M"`, `"D"`, `"h"`, `"m"`, `"s"`, corresponding to whether the `year`, `month`, `day`, `hour`, `minute` or `second` is missing, respectively. Where more than one of these values are missing then `missing` takes the first one from `c("Y","M","D","h","m","s")`. When `month` or `day` is missing, one is used in calculations; and where `hour`, `minute` or `second` are missing, zero is used in calculations.

The main difference with the `chron` function in package **chron** is that the `datetimes` function has a facility to account for fractional seconds, and also missing values as described above.

See Also

format.datetimes, [.datetimes, print.datetimes, years1, months1, days1

Examples

```
a <- datetimes(1998, 6, 2, 13, 34, 25.9, dp.second=1)
print(a)
print.default(a)
format(a)
```

---

days1 *Day, Month and Year of Datetimes Value*

Description

These functions act in a similar way as the **chron** functions days, months and years. See "Details" below.

Usage

```
days1(x)
months1(x)
years1(x)
```

Arguments

x	a datetimes object.

Details

The datetimes object is a numeric vector containing the number of days and fractions thereof from some defined origin, often 1 Jan 1970.

In versions of **chron** prior to 2.2-35, there was a bug. This caused the functions days, months, and years to sometimes be erroneous for dates before the origin. For example, `chron(dates.="12/31/1969",times.="18:00:00")` would give `"01/01/1970 -6:00:00"`. This appears to have been fixed in version 2.2-35. It was probably caused by a call to trunc in month.day.year rather than floor.

Another difference is that the values returned by the **chron** functions days, months, and years are ordered factors. This is not the case for days1, months1 and years1.

Value

A numeric vector of the same length as x giving either the day of the month (1, . . . , 31), the month (1, . . . , 12), or the year (e.g. 1998).

See Also

datetimes, days, months, years, hrs.mins.secs

Examples

```
x <- julian(12,31,1969) + 18/24
print(x)
#   x represents 18:00:00 hrs on 31 Dec 1969
#   x = -1 + 18/24 = -0.25

x <- datetimes(1998, 6, 2, 13, 34, 25.9, dp.second=1)
print(days1(x))
print(months1(x))
print(years1(x))
print(hrs.mins.secs(x))
```

`epi.circle` *Calculate Epicentral Coordinates of a Circle*

Description

Calculates the epicentral coordinates of 399 points on the perimeter of a circle of given radius centred at (centrelong, centrelat). The circle radius is the arc length radius on the surface of the earth.

Usage

```
epi.circle(centrelong, centrelat, radius, earthradius=6371)
```

Arguments

`centrelong`	a number specifying the longitude in degrees of the circle centre.
`centrelat`	a number specifying the latitude in degrees of the circle centre.
`radius`	a number specifying the circle radius in kilometres.
`earthradius`	assumed radius of the earth in kilometres. The default is 6371 km.

Value

If the function call is assigned to another object, then a list object is returned. It contains the following components.

`centrelong`	as specified above.
`centrelat`	as specified above.
`radius`	as specified above.
`longitude`	a vector giving the longitude of 399 points on the circle perimeter.
`latitude`	a vector giving the latitude of 399 points on the circle perimeter.

Author(s)

Yang Wenzheng, 1998

Examples

```
x <- epi.circle(175, -42, 3000)

#   Now project the circle outline
y <- projection(6371, x$latitude, x$longitude, 175, -42)
par(pin=c(5,5))
plot(y$x, y$y, type="l")
```

`format.datetimes` *Format a DateTimes Object*

Description

This function provides a method for the generic function format to be used on objects with class `"datetimes"`.

Usage

```
## S3 method for class 'datetimes'
format(x, ...)
```

Arguments

`x`	a datetimes object.
`...`	other options for formatting `"datetimes"` objects.

Value

a character vector of the same length as the input vector, with the format "DDMMMYYYY hh:mm:ss.s" is output, where the number of decimal places for second is specified by the datetimes attribute `"dp.second"`.

Examples

```
a <- datetimes(1998, 6, 2, 13, 34, 25.9, dp.second=1)
format(a)
```

hrs.mins.secs *Calculates Hours, Minutes and Seconds*

Description

Given a datetime object (vector), the time represented by each element on a 24 hour clock is returned.

Usage

```
hrs.mins.secs(x)
```

Arguments

x	a datetime vector object.

Value

A list object with the following components:

hour	a vector of the hours (0, . . . , 23).
minute	a vector of the minutes (0, . . . , 59).
second	a vector of the seconds (0, . . . , 59.9).

See Also

datetimes, years1, months1, days1

Examples

```
x <- datetimes(year=1998, month=8, day=25, hour=20, minute=33,
               second=24.5, dp.second=1)
hrs.mins.secs(x)
```

lattice *Creates Lattice of Points on Sphere Surface*

Description

A lattice of $(2J + 1)$ by $(2K + 1)$ grid points based on squares or equilateral triangles (at a fixed distance, d apart) is calculated centred on a specified point and with a given orientation.

The major axis of the lattice is taken along the great circle through the lattice centre in the direction specified by the orientation alpha.

Usage

```
lattice(K, J, d, alpha, centrelat, centrelong, triangle=TRUE)
```

Arguments

K	the number of points stepped off to the left and to the right of the origin along the major axis.
J	the number of points stepped off in both directions orthogonal to the major axis.
d	the distance between the lattice points (in kilometres).
alpha	$\pi/2$ minus alpha gives the angle (in radians) between the local meridian through the lattice centre and the major axis of the lattice, where alpha is between 0 and π.
centrelat	the latitude component of the centre (origin) of the grid.
centrelong	the longitude component of the centre (origin) of the grid.
triangle	logical. If TRUE, a triangular grid is calculated. If FALSE, a square grid is calculated.

Details

A lattice of $(2J + 1)$ by $(2J + 1)$ grid points based on equilateral triangles (at a fixed distance, d apart) or squares is calculated. The grid is rotated by the angle alpha. The algorithm from epi.circle is subsequently used to convert to spherical coordinates, centred over the specified origin. The spherical coordinates (longitude, latitude) of the grid points are returned in a list.

Value

A list object is returned. It contains the following components:

centrelong	as specified above.
centrelat	as specified above.
d	the distance between the lattice points (in kilometres).
longitude	a vector giving the longitude of the grid points.
latitude	a vector giving the latitude component of the grid points.
i	a vector of sequence numbers of the lattice points, starting from 1 in the south western corner.
j	a vector of grid coordinates along the major axis of the lattice, with 0 as the centre.
k	a vector of grid coordinates orthogonal to the major axis of the lattice, with 0 as the centre.

Author(s)

Alistair Merrifield, 1998

See Also

lattice.retrieve

Examples

```
#   Creates a triangular lattice of 207 points centred over (175, -41)

x <- lattice(J=4, K=11, d=120, alpha=pi/4, centrelat=-41,
             centrelong=175, triangle=TRUE)

plot(x$longitude, x$latitude, xlab="Longitude", ylab="Latitude",
     main="Triangluar Arrangement")

x <- lattice(J=4, K=11, d=120, alpha=pi/4, centrelat=-41,
             centrelong=175, triangle=FALSE)

plot(x$longitude, x$latitude, xlab="Longitude", ylab="Latitude",
     main="Rectangular Arrangement")
```

---

lattice.retrieve *Determine Lattice Indices for Given Circle*

Description

This is a complementary function to the function lattice. The function lattice calculates longitude-latitude centres for a $(2J+1)$ by $(2K+1)$ grid. It produces a vector of $(2J+1)(2K+1)$ centres. Given the ith centre, lattice.retrieve calculates the corresponding values of j and k. Alternatively, given j and k, the function returns the corresponding index i.

Usage

```
lattice.retrieve(i = NA, j = NA, k = NA, K, J)
```

Arguments

i	the index number of the centre of interest.
j	scalar where $-J \leq j \leq J$, and J is defined in the documentation for the function lattice.
k	scalar where $-K \leq k \leq K$, and K is defined in the documentation for the function lattice.
K	see documentation for the function lattice.
J	see documentation for the function lattice.

Value

Returns a list object with the values of i, j, k, J, and K.

See Also

lattice

`library.version` *Print Version of Library*

Description

Prints, via Unix, the versions of each of the packages requested.

Usage

```
library.version(which = "[a-z]*")
```

Arguments

`which`	string or character vector of packages for which version information is required. The default is to show all the currently installed packages.

Details

The character strings may use shell wildcards, but this means that all shell special characters must be escaped, as shown above.

Value

No value is returned by this function.

Author(s)

Ray Brownrigg, 2000

See Also

package.description

Examples

```
# get the version of the current base package
library.version("base")

# get the version of the maps library and all SSLib packages
library.version(c("maps", "ss*"))
```

NZ55 *NZ55 Earthquake Catalogue*

Description

The NZ55 earthquake catalogue contains all events in the New Zealand Catalogue with magnitude $\geq$ 5.5 from 1 Jan 1960 until 31 Dec 1998.

Usage

```
data(NZ55)
```

Format

This data frame has class `"catalogue"` and contains the following columns:

latitude number of degrees north.

longitude number of degrees east.

magnitude event magnitude.

depth event depth (km).

missing.time always `""`.

time vector of event dates and times with class `"datetimes"`.

Source

These data originate from the New Zealand Catalogue which is administered by the Institute of Geological and Nuclear Sciences, Lower Hutt, NZ.

Examples

```
data(NZ55)
summary(NZ55)
```

Palliser *Palliser Earthquake Catalogue*

Description

The Palliser Catalogue contains events from the original Wellington Catalogue with magnitude $\geq$ 2.5, depth $\leq$ 40 km, between 1 Jan 1990 and 31 Dec 1991, and within a 36 km radius of the point (175.503°E, 41.684°S).

Usage

```
data(Palliser)
```

Format

This data frame has class "catalogue" and contains the following columns:

latitude number of degrees north.

longitude number of degrees east.

magnitude event magnitude.

depth event depth (km).

missing.time always "".

time vector of event dates and times with class "datetimes".

Source

These data originate from the original Wellington Catalogue which was administered by the Institute of Geological and Nuclear Sciences, Lower Hutt, NZ.

Examples

```
data(Palliser)
summary(Palliser)

#    Reset the time origin to 1 January 1990
#    Useful for fitting models where this time is "zero"
Palliser$time <- Palliser$time - julian(1,1,1990)
attr(Palliser$time, "origin") <- c(month=1, day=1, year=1990)
```

---

print.catalogue *Method for Generic Function*

Description

This function provides a method for the generic function print to use on objects of class "catalogue".

Usage

```
## S3 method for class 'catalogue'
print(x, ...)
```

Arguments

x	a catalogue object.
...	other options for printing "catalogue" objects.

See Also

summary.catalogue, as.catalogue

Examples

```
data(NZ55)
print(NZ55)
```

`print.datetimes` *Method for Generic Function*

Description

This function provides a method for the generic function `print` to use on objects of class `"datetimes"`.

Usage

```
## S3 method for class 'datetimes'
print(x, ... )
```

Arguments

`x`	a datetimes object.
`...`	other options for printing `"datetimes"` objects.

Value

Writes out the vector in the format DDMMMYYYY hh:mm:ss.s, where the number of decimal places for seconds is specified by the attribute `"dp.second"` in the `"datetimes"` object.

See Also

`datetimes`, `format.datetimes`, `years1`, `months1`, `days1`, `hrs.mins.secs`

Examples

```
x <- datetimes(1998, 8, 3, 12, 55, 33, dp.second=0)
print(x)
```

print.subset *Method for Generic Function*

Description

This function provides a method for the generic function print to be used on objects with class "subset".

Usage

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

Arguments

x	a subset object.
...	other options for printing "subset" objects.

Examples

```
data(NZ55)
a <- subsetrect(NZ55)

print(a)
print.default(a)
```

projection *Transforms Spherical Coordinates to Cartesian*

Description

Transforms spherical coordinates to Cartesian and rotates. Have spherical coordinates (ρ, ϕ, θ). ρ is the radius, θ can be thought of as the longitude and ϕ the latitude. The resultant coordinates (x, y, z) are rotated such that the point with longitude α and latitude β becomes the new north pole. Thus a plot of the new xy plane is Azimuths projection, centered at the new reference point.

Usage

```
projection(rho, phi, theta, alpha, beta)
```

Arguments

rho	a vector with the radius component of the original spherical coordinates for each point.
phi	a vector with the latitude component of the original spherical coordinates for each point.
theta	a vector with the longitude component of the original spherical coordinates for each point.
alpha	a scalar giving the longitude of the centre of the new hemisphere (pole).
beta	a scalar giving the latitude of the centre of the new hemisphere (pole).

Value

A list object containing the following vectors of the same length:

x	the x component of the new Cartesian coordinates.
y	the y component of the new Cartesian coordinates.
z	the z component of the new Cartesian coordinates.

See Also

hemisphere

---

subscript.datetimes *Subscript a Datetimes Object*

Description

This function provides a method for the generic function subscript to be used on objects with class "datetimes". It works in the same way as the default function, except that it retains all attributes of the "datetimes" object.

Subset Selection *Select Subset of Events from Catalogue*

Description

Selects events within a specified selection criteria (rectangular, circular, polygon, or spherical) from a given catalogue. See Details for more information.

Usage

```
subsetcircle(x, centrelat=0, centrelong=0, minradius=0, maxradius=Inf,
             mindepth=0, maxdepth=Inf, minmag=-Inf, maxmag=Inf,
             minday=-Inf, maxday=Inf, radius=6371,
             report.count=TRUE, na.rm=TRUE, ...)

subsetpolygon(x, polylong, polylat, mindepth=0, maxdepth=Inf,
              minmag=-Inf, maxmag=Inf, minday=-Inf, maxday=Inf,
              report.count=TRUE, na.rm=TRUE, ...)

subsetrect(x, minlong=0, maxlong=360, minlat=-90, maxlat=90,
           mindepth=0, maxdepth=Inf, minmag=-Inf, maxmag=Inf,
           minday=-Inf, maxday=Inf, report.count=TRUE, na.rm=TRUE,...)

subsetsphere(x, centrelat=0, centrelong=0, centredepth=0, minradius=0,
     maxradius=Inf, minmag=-Inf, maxmag=Inf, minday=-Inf,
     maxday=Inf, radius=6371, report.count=TRUE, na.rm=TRUE, ...)
```

Arguments

x	catalogue object.
minlong	minimum longitude, or western boundary. Specified as degrees consistent with the given catalogue. Usually these will be Pacific centric (0 to 360), or Europe centric (-180 to 180).
maxlong	maximum longitude, or eastern boundary. Specified as degrees consistent with the given catalogue. Usually these will be Pacific centric (0 to 360), or Europe centric (-180 to 180).
centrelong	longitude of the circle centre. Specified as degrees consistent with the given catalogue. Usually these will be Pacific centric (0 to 360), or Europe centric (-180 to 180).
polylong	vector containing the longitudes of the polygon corners in the order as one travels the perimeter. Usually these will be Pacific centric (0 to 360), or Europe centric (-180 to 180).
minlat	minimum latitude, or southern boundary. Specified as degrees north (positive) or south (negative) of the equator (-90 to 90).
maxlat	maximum latitude, or northern boundary. Specified as degrees north (positive) or south (negative) of the equator (-90 to 90).
centrelat	latitude of the circle centre. Specified as degrees north (positive) or south (negative) of the equator (-90 to 90).
polylat	vector containing the latitudes of the polygon corners in the order as one travels the perimeter. Specified as degrees north (positive) or south (negative) of the equator (-90 to 90).
mindepth	minimum depth in kilometres.
maxdepth	maximum depth in kilometres.
centredepth	depth of the sphere centre.

minradius	surface arclength radius (km) within which events are not to be selected.
maxradius	surface arclength radius (km) within which events to be selected.
radius	the assumed radius of the earth. Default is 6371 km.
minmag	minimum magnitude.
maxmag	maximum magnitude.
minday	the number of days (and fractions) after 00:00 hrs on 1 January 1970 from which events are required. See Details.
maxday	the number of days (and fractions) after 1 January 1970 denoting the upper bound of the time interval containing the required events. See Details.
report.count	Boolean. Report the number of events referred to by the new subset. Default is TRUE.
na.rm	Boolean. Determines whether those events with missing values of the subsetting variables should be excluded, see Details. Default is TRUE.
...	other options for restricting catalogue data. These must take the form option == value, where option is a component of the specified catalogue.

Details

The subsetting is based on the catalogue variables latitude, longitude, depth, magnitude and time. The default settings for each of these variables is to take all events. The function only works through each of the "min" and "max" values *included in your function call*. Those arguments *not explicitly listed will have no effect*. A logical vector is constructed with the same length as the number of events in the catalogue, and indicates whether a given event satisfies the subsetting criteria. From this, the vector of indices of the required events is created.

Historical events often have missing values, particularly the magnitude and depth. Consider the situation where minmag=6. The logical vector within the function will contain TRUE's and FALSE's for those events that satisfy or not satisfy this condition, respectively. However, for those historical events with a missing magnitude (NA), the elements in the logical vector will also be NA. The argument na.rm determines whether these events should be removed (TRUE) or included (FALSE) in the subset. The default setting is to remove such events, though in the context of large historical events, it would be more sensible to include them, i.e. na.rm=FALSE.

The arguments minday and maxday represent the boundary *points* of the time interval, and can represent fractions of days. For example, 00:00 hrs on 1 Jan 1970 is represented as minday=0, whereas 18:00 hrs on 2 Jan 1970 is represented as minday=1.75. Similarly, 18:32:24 hrs on 30 Sept 1990 could be specified as minday = julian(9,30,1990) + 18/24 + 32/(24*60) + 24/(24*60*60). Note that maxday = julian(1,1,1990) represents the *point* 00:00 hrs on 1 Jan 1990, and therefore will not include events on 1 Jan 1990 after midnight.

Note that subset.circle, subset.polygon, subset.rect, and subset.sphere are deprecated; they conflict with the generic function subset. Change to subsetcircle, subsetpolygon, subsetrect, or subsetsphere, respectively.

In the case of subpolygon: events are initially selected within the rectangle defined by range(polylong) and range(polylat). The outside component in the returned object are the indices of those events within this rectangle but outside the polygon. Events with a missing latitude or longitude will always be excluded regardless of the value of the argument na.rm.

Value

Returns an object of type list with class `"subset"`. Note that the object is not a catalogue, it is essentially a list of indices of the events from the specified catalogue that satisfy the selection criteria. All optional arguments above are included as elements within the list. Other components within the list are:

`indices`	an array of indices for events in catalogue `x` satisfying the selection criteria.
`catname`	catalogue name.
`type`	type of subset, set to `"Circular"`, `"Polygon"`, `"Rectangular"`, or `"Spherical"`.
`outside`	contained in polygon subsets only, see Details.

See Also

`print.subset`, `summary.subset`

Examples

```
#   Select events in the NZ55 catalogue between 1961 and 1992 with magnitude >=
7

data(NZ55)
a <- subsetrect(NZ55, minday=julian(1,1,1961), maxday=julian(1,1,1993),
                minmag=7)
print(summary(a))
as.catalogue(a, "NZ7")
print(NZ7)

#-------------------------------------------------
#   Select events from the NZ55 catalogue, no more than
#   100kms (epicentral distance) from Wellington.

data(NZ55)
a <- subsetcircle(NZ55, centrelat=-41.3, centrelong=174.8,
                  maxradius=100)
print(summary(a))

#-------------------------------------------------
#   Select events from the NZ55 catalogue with hypocentre no more
#   than 50kms from Wellington, but at least 20kms from Wellington.

data(NZ55)

b <- subsetsphere(NZ55, centrelat=-41.3, centrelong=174.8,
                  maxradius=100, minradius=20, minmag=2)
summary(b)

#-------------------------------------------------
#  Select events from NZ55 within following polygon

polylong <- c(173.8, 174.3, 174.8, 174.4, 174.8, 175.6,175,174.5)
```

```
polylat <- c(-41.5, -40.5, -40.9, -41, -41.4, -41, -41.8, -41.25)

data(NZ55)
a <- subsetpolygon(NZ55, polylong, polylat)

#  plot points within polygon
plot(NZ55$longitude[a$indices], NZ55$latitude[a$indices], cex=2,
     xlim=range(polylong), ylim=range(polylat), pch=16, col="red")
points(c(polylong,polylong[1]), c(polylat, polylat[1]), type='l', lty=2)
```

---

summary.catalogue *Summary of Earthquake Catalogue*

Description

Provides a method for the generic function summary on objects of class "catalogue".

Usage

```
## S3 method for class 'catalogue'
summary(object, ...)
```

Arguments

object	name of the earthquake catalogue. The catalogue is a list object with class "catalogue".
...	other options for summarising "catalogue" objects.

Value

An object is returned with the following components.

catname	character string containing name of the catalogue.
n	number of events in the catalogue.
ranges	a matrix giving the minimum, maximum and number of missing values for each of latitude, longitude, depth and magnitude.
time.range	vector of length 2 giving the datetime of the first and last events within the catalogue.
missing.times	a matrix where each row relates to an event with a missing component in its time. The first column gives the record number, 2nd the recorded time in the R object and the 3rd column describes the missing component.
names	names of all variables in the catalogue list object.

See Also

as.catalogue, summary

Examples

```
data(NZ55)
summary(NZ55)
```

---

summary.subset *Summary of Subset*

Description

This function provides a method for the generic function summary for objects of class "subset".

Usage

```
## S3 method for class 'subset'
summary(object, plot=FALSE, ...)
```

Arguments

object	an object of class "subset".
plot	logical. Default is FALSE, in which case the character string is composed of only ASCII characters. If TRUE it is composed so that inequalities are represented properly on a graphics device.
...	other options for summarising "subset" objects.

Value

One character string describing the subset represented by object.

See Also

subsetcircle, subsetpolygon, subsetrect, subsetsphere, print.subset

Examples

```
data(NZ55)
a <- subsetrect(NZ55, minlong=165, maxlong=180, minlat=-48, maxlat=-35,
                minmag=7, minday=julian(2,27,1960)+12/24+32/1440)
summary(a)
```

write.catalogue *Write Catalogue to a Text File*

Description

Writes an earthquake catalogue within the library to an ASCII file.

Usage

```
write.catalogue(x, file=paste(deparse(substitute(x)), ".txt", sep = ""),
                ignore=FALSE, append=FALSE)
```

Arguments

x	name of the earthquake catalogue.
file	required name of the ASCII file. By default, it will have the same name as the catalogue with ".txt" appended.
ignore	if TRUE, do not print negative signs on latitudes. Default is FALSE.
append	if TRUE and the ASCII file already exists, the catalogue will be appended to the end of the file, otherwise the contents of the file are overwritten. Default is FALSE.

Value

NULL

Side Effects

Writes an ASCII file as described above.

Examples

```
data(NZ55)
write.catalogue(NZ55, file="temp.txt")
```

Index