

Package ‘Imap’

February 19, 2015

Type Package

Title Interactive Mapping

Version 1.32

Date 2010-02-11

Author John R. Wallace <Imap.for.R@gmail.com>

Maintainer John R. Wallace <Imap.for.R@gmail.com>

Depends R (>= 2.10.0)

Description Zoom in and out of maps or any supplied lines or points,
with control for color, poly fill, and aspect.

License GPL (>= 2)

LazyLoad yes

Repository CRAN

Date/Publication 2012-10-29 08:57:11

NeedsCompilation no

R topics documented:

IMAP-package	2
gdist	2
imap	4
imap.ll	7
select.pts, draw.polygon, inside.polygon, draw.lines, col.alpha	9
world	11

Index	13
--------------	-----------

 IMAP-package

Interactive Mapping

Description

The main function `imap()` and its allies `ilines()` and `ipts()` enables the user to zoom in, zoom out, and reset a collection of polygons, lines, or points. The function `select.lines()` allows the interactive selection of one or more lines, whereas `select.pts()` has the user create a polygon to select a collection of points within or outside of the polygon. The user has control over the color and width of the lines, the color of the polygon fill, and the aspect ratio.

Details

Package:	IMAP
Type:	Package
Version:	1.32
Date:	2010-02-011
License:	GPL (>= 2)
LazyLoad:	yes

Without any input data `imap()` defaults to using version 2.0 of the GSHHS (Global Self-consistent Hierarchical High-resolution Shorelines) polygons. However, any two dimensional user supplied data or a list of multiple data sets can be used. When the user has finished interacting with the data the result is invisibly returned as a list of lists which, if saved, can in turn be replotted with `imap`. Zooming is accomplished by left-clicking in two different locations on the figure to define a rectangle that will be zoomed into. Left-clicking outside the plot region (but somewhere in the figure region) will zoom out locally. Double left-clicking on the same spot will reset the figure. Right-click to stop.

Author(s)

John R. Wallace: <Imap.for.R@gmail.com> (Limited support)

Examples

```
# See examples under the main 'imap()' function.
```

 gdist

Geodesic distance (great circle distance) between points

Description

`gdist()` gives the geodesic distance between two points specified by latitude/longitude using Vincenty inverse formula for ellipsoids. The distance is given in nautical miles (the default), meters, kilometers, or miles. `gdist.total()` applies `gdist()` to any number of line segments.

Usage

```
gdist(lon.1, lat.1, lon.2, lat.2, units = "nm", a = 6378137.0, b = 6356752.3142, verbose = FALSE)

gdist.total(longlat, units = "nm", segments = TRUE, digits = 2)
```

Arguments

lon.1	Longitude of the first point.
lat.1	Latitude of the first point.
lon.2	Longitude of the second point.
lat.2	Latitude of the second point.
units	Any one of "nm": nautical miles (the default), "m": meters, "km": kilometers, or "miles": miles, for units of the returned value.
a	Major semiaxes of the ellipsoid in meters. The default is for the Earth: 6378137.0
b	Minor semiaxes of the ellipsoid in meters. The default is for the Earth: 6356752.3142
verbose	Should extra information be output, the default is FALSE.
longlat	Two dimensional (n x 2) data in the form of a matrix or dataframe. The first column should be the longitude or x-axis column and the second the latitude or y-axis column.
segments	When there are more than two or more segments, should the distance for each segment be output. The default is TRUE.
digits	The number of rounding digits to use in the segment output. The default is 2.

Value

The total geodesic distance in the units specified.

Author(s)

John R. Wallace: <Imap.for.R@gmail.com> (Limited support)

References

Inspired by: <http://www.movable-type.co.uk/scripts/LatLongVincenty.html>

Vincenty, T. Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. Survey Review XXII, 176, April 1975. http://www.ngs.noaa.gov/PUBS_LIB/inverse.pdf

See Also

[imap](#), [draw.lines](#)

Examples

```
## Not run:

imap() # Zoom into an area, right-click to stop.

gdist.total(draw.lines(rainbow(10), lwd=4)) # Left-click two or more times, right-click to stop.

# The total distance of the polygon's perimeter:
gdist.total(draw.polygon()) # Left-click three or more times, right-click to stop.

# The total distance of contiguous points (those not separated by NA's).
gdist.total(select.pts(imap()[[1]]$ll)) # Zoom in to an area, right-click, and then select points by creating a p

## End(Not run)
```

imap

Interactive Mapping

Description

Zoom in and out of maps or any supplied lines or points with control for color, fill, and aspect.

Usage

```
imap(longlat = list(world.h.land, world.h.lake, world.h.island, world.h.pond.in.island, world.h.bord
  longrange, latrange, zoom = TRUE, col = c("black", "blue", "forestgreen", "dodgerblue", "cyan")
  poly = c("grey40", "blue", "forestgreen", "dodgerblue", NA), lwd = 1, keep.attr = TRUE, add.all

ilines(..., fill = FALSE, aspect = 2)

ipts(longlat = npacific, axes = 'std', ...)
```

Arguments

longlat	Two dimensional user supplied data in the form of a matrix, dataframe, a list of of such data, or a list of lists as returned by this function (see Value). The first column should be the longitude or x-axis column and the second the latitude or y-axis column. As with the lines() function a row of NA's will produce a break in the line (equivalent to a pen-up on a plotter).
longrange	The initial longitudinal or x-axis range to be plotted.
latrange	The initial latitudinal or y-axis range to be plotted.
zoom	If TRUE (default) the zooming feature will be enabled. Zooming is accomplished by left clicking in two different locations on the figure to define a rectangle that will be zoomed into. Left clicking outside the plot region (but somewhere in the figure region) will zoom out. Double left clicking on the same spot will reset the plot. Right click to stop.

col	A single value or vector for the color of the lines. If a list of is supplied to 'longlat' then each item of the list will, in turn, get a color from the vector of colors supplied. The colors will be reused if necessary.
fill	If TRUE (default) the polygon fill feature will be enabled. The function ilines() calls imap with the default for fill = FALSE.
poly	A single value or vector for the fill color(s) of the polygon(s). If a list of is supplied to 'longlat' then each item of the list will, in turn, get a fill color from the vector of colors supplied. The colors will be reused if necessary. If a NA is in the vector, no polygon fill will be done of the corresponding item in the 'longlat' list.
lwd	A single value or vector for the width of the lines.
keep.attr	Will the attributes of line color, line width, and polygon fill color be taken from the supplied list of lists (as returned by this function) or should the supplied values be used. Default is TRUE. See the examples.
add.all	Should all the items in 'longlat' be added to the current plot. Default is FALSE.
bg	The background color. The default is "grey81". The background color is not a value returned by this function.
tol	The tolerance used when double left clicking on the same spot to reset the data. If very low values of data are supplied or zooming in is extreme, this value may need to be lowered. Default is 0.05 .
...	Any extra arguments are checked the 'plt.' prefix. Any such argument will be applied to the plot() function, all others will be applied to the lines() function. (For wanting to know how this was done, look at the top of the imap.ll() function and search on 'plot.dots' and 'lines.dots'.)
aspect	See the help for imap.ll() for information on the 'aspect' argument.
axes	See the help for imap.ll() for information on the 'axes' argument.

Details

The functions ilines() calls imap() with the defaults for 'fill' and 'aspect' set to FALSE and 2, respectively. The function ipts() calls imap() with a call to 'type' set to "p" for points, 'axes' set to "std", and 'fill' set to FALSE. arguments.

Value

Polygons are often broken by zooming in, so saving the results invisibly returned by imap() is often best done for lines and points when fill = FALSE. However the 'poly' argument is saved for each item in the value returned. The arguments 'longrange' and 'latrange' with 'zoom = FALSE' can be used to return to a zoomed in area with polygon colors intact. See the examples below.

The value returned is a list of lists, where each item in the primary list is an item from the original list given to argument 'longlat' (or created from the object given to 'longlat'). Each primary item has the following components.

ll	The longlat lines (or pts) after zooming for this primary item.
col	The color for the lines of this primary item.
lwd	The line width for the lines of this primary item.
poly	The polygon color (or NA) for the polygon of this primary item.

Author(s)

John R. Wallace: <Imap.for.R@gmail.com> (Limited support)

See Also

[imap.ll](#), [select.pts](#), [world](#), [gdist](#)

Examples

```
## Not run:
  if(.Platform$OS.type == "windows")
    windows.options(width=13, height=9) # Set the window to be wider than high.

imap() # On the figure, left-click two corners of an area you would like to zoom in on.
      # Creating a rectangle outside the figure region will zoom out locally.
      # Double left-click on the same spot to zoom back out to the original data. Right-click to stop.

imap(antarctic = TRUE) # Reset with two left-clicks in the same spot for a more symmetrical picture.

imap(list(world.h.land, imap(world.h.land)[[1]]$ll)) # Whatever area is selected in the first use of imap()
                                                    # will be a different color in the second imap().

col.alpha('magenta', alpha = 0.5) # Look at the color value for magenta that has an alpha transparency value

# Use this color so there is transparency where rivers and borders are the in the same location.
# Also note that the land was made a lighter grey by going from grey40 to grey50.
imap(list(world.h.land, world.h.lake, world.h.island, world.h.pond.in.island, world.h.borders, world.h.rivers,
  longrange = c(-89, -99), latrange = c(41, 50), col=c("black", "blue", "forestgreen", "dodgerblue",
  "cyan", col.alpha('magenta', alpha = 0.5)), poly = c("grey50", "blue", "forestgreen", "dodgerblue", NA, NA))

# All the world's rivers with zoom turned back on (may take a while to finish).
imap(list(world.h.land, world.h.lake, world.h.island, world.h.pond.in.island, world.h.borders, world.h.rivers,
  col=c("black", "blue", "forestgreen", "dodgerblue", "cyan", col.alpha('magenta', alpha = 0.5)),
  poly = c("grey50", "blue", "forestgreen", "dodgerblue", NA, NA))

test <- ilines(list(npacific, select.lines(ilines(npacific)[[1]]$ll))) # Zoom, right-click, select one or more
test[[2]]$lwd <- 3 # Change attributes manually and
ilines(test) # keep the default keep.attr = TRUE, or
test <- ilines(test, col=c("magenta", "goldenrod"), lwd = c(2,5), keep.attr = FALSE) # set keep.attr = FALSE

imap(longrange = c(-84.5, -81), latrange = c(44.5, 47)) # An example of ponds on an island in the Great Lakes
                                                    # Use zoom = F for a static map, to which additional data may be added

# To create a file from a finished map use dev.copy2eps(), dev.copy2pdf(), or the following function for 'png'
dev.copy2png <- function(file = 'Rplot.png', factor.exp = 1, ...) {
  dev.copy(png, file=file, width=par()$fin[1] * factor.exp, height=par()$fin[2] * factor.exp, units="in", res=300,
  dev.off()
}

# Use with 'Maps' library
```

```

require(maps)
require(mapproj)

map.to.imap <- function (map.list) {
  matrix(cbind(map.list[[1]], map.list[[2]]), ncol = 2)
}

# Compare this map
imap(map.to.imap(map('usa')))

# With this one
windows()
imap(map.to.imap(map('usa', proj="bonne", param=45)), aspect=1)

# Plot without axes
imap(map.to.imap(map('usa', proj="bonne", param=45)), aspect=1, axes=FALSE)

# Select an area to highlight and then re-zoom
ilines(list(map.to.imap(map(projection = "gnomonic")),
  ilines(map.to.imap(map(projection = "gnomonic")), aspect=1.5, axes=FALSE)[[1]]$ll), aspect=1.5, axes=FA

## End(Not run)

```

imap.ll

The Imap plotting function - usually called by imap()

Description

imap.ll() is the underlying mapping function that is called by imap() for each item in the 'longlat' list.

Usage

```
imap.ll(area = npacific, longrange, latrange, poly = NA, antarctic = FALSE, arctic = FALSE, oz = FALSE,
  grid = FALSE, aspect = 1.5, add = FALSE, zoom = TRUE, lines.out.of.bounds = TRUE, tol = 0.05, ...)
```

```
select.lines(area = npacific, longrange, latrange, poly = NA, antarctic = FALSE, arctic = FALSE, oz = FALSE,
  grid = FALSE, aspect = 1.5, add = FALSE, zoom = TRUE, lines.out.of.bounds = TRUE, tol = 0.005, ...)
```

Arguments

area	A 2 dimension matrix or data frame with the first column the longitude or x-axis column and the second column the latitude or y-axis column. A simple two item list where the first item is the longitude (or x) and the second item is the latitude (or y) can also be supplied. As with the lines() function a row of NA's will produce a break in the lines (equivalent to a pen-up on a plotter).
longrange	The initial longitudinal or x-axis range to be plotted.

latrange	The initial latitudinal or y-axis range to be plotted.
poly	A single value for the fill color of the polygon(s).
antarctic	Should the perspective be looking down on the South Pole. Default is FALSE. If TRUE, axes will be set to FALSE.
arctic	Should the perspective be looking down on the North Pole. Default is FALSE. If TRUE, axes will be set to FALSE.
oz	Should the perspective be with the South Pole on top (As one in Australia (Oz) might like.). Default is FALSE.
axes	One of "map" (the default), "std", or FALSE. Using "map" will show longitude and latitude markings, "std" will give the standard plotting labels, and FALSE will give no axes at all.
grid	If TRUE a thin dashed grid will be shown. Default in FALSE.
aspect	A figure aspect for which larger values mean the figure will be more high than wide. Lower values will give a figure that is wider than high. The default is 1.5.
add	If TRUE, 'area' will added to the current plot. Default if FALSE.
zoom	If TRUE (default) the zooming feature will be enabled. Zooming is accomplished by left-clicking in two different locations on the figure to define a rectangle that will be zoomed into. Left-clicking outside the plot region (but somewhere in the figure region) will zoom out. Double left clicking on the same spot will reset the plot. Right-click to stop.
lines.out.of.bounds	Vestigial. May only need to be set to FALSE for older versions of Splus. Default is TRUE.
tol	The tolerance used when double left clicking on the same spot to reset the data. If very low values of data are supplied or the zooming in is extreme, this value may need to be lowered. Default is 0.05 .
...	Any extra arguments are checked the a 'plt.' prefix. Any such argument will be applied to the plot() function, all others will be applied to the lines() function. (For those looking to see how this was done, look at the top of the imap.ll() function and search on 'plot.dots' and 'lines.dots')

Details

The function `select.lines` has similar argumnets to `imap.ll` but is used to select line seqments. See the Examples Section.

Value

Only data contained within the last area to be zoomed in on is invisibly returned as a $(n \times 2)$ matrix.

Author(s)

John R. Wallace: <Imap.for.R@gmail.com> (Limited support)

See Also

[imap](#), [select.pts](#)

Examples

```
## Not run:

ilines(list(npacific, select.lines(npacific))) # Select one or more line segments with left-click(s) and then a

Africa <- select.lines(world.h.land) # Select the African polygon.
imap(zoom = FALSE)
imap(Africa, poly = 'purple', add = T, zoom = FALSE)

## End(Not run)
```

```
select.pts, draw.polygon, inside.polygon, draw.lines, col.alpha
Select points; draw polygons; draw lines
```

Description

Functions for selecting points and drawing polygons and lines.

Usage

```
select.pts(pts, list.of.lists.obj = 1, outside.poly = FALSE, col = "blue", alpha = 0.5, lty = 1, ...)

draw.polygon(col = "blue", alpha = 0.5, lty = 1, ...)

inside.polygon(pts, h)

draw.lines(col = "red", alpha = 0.5, ...)

col.alpha(col, alpha = 0.5)
```

Arguments

<code>pts</code>	A (n x 2) matrix of points.
<code>list.of.lists.obj</code>	Which object in the list of lists, as returned by <code>imap()</code> , should be used to select points from. Objects with no selected points are dropped by <code>imap()</code> and therefore using <code>str()</code> on the value returned by <code>imap()</code> can be useful in revealing which object number is wanted when, for example, islands or ponds on islands are to be selected.
<code>outside.poly</code>	Should the points which are outside the polygon be selected. The default is <code>FALSE</code> , which gives the points inside the polygon.
<code>col</code>	The color to fill the polygon for <code>draw.polygon()</code> or the line color for <code>draw.lines()</code> . The parameter can be either a color name, a hexadecimal string of the form <code>"#FF00FF80"</code> , or an integer <code>i</code> meaning <code>palette()[i]</code> .

<code>alpha</code>	An alpha transparency value where 0 is fully transparent and 1 is opaque.
<code>lty</code>	The polygon line type.
<code>h</code>	A hull or polygon defined by a matrix ($[k+1] \times 2$) of (ordered) vertices, [with last row = first row].
<code>...</code>	Additional graphical parameters for the functions <code>points()</code> , <code>polygon()</code> or <code>lines()</code> .

Details

The function `select.pts()` returns points that are inside (or outside) a polygon. The polygon is interactively drawn about points (or lines) which normally would have been previously plotted. After at least three vertices of the polygon have been selected by left-clicking on the figure the interaction is eventually stopped by a right-click. The resulting polygon is closed and filled with color 'col' after alpha level 'alpha' has been applied via the `col.alpha()` function.

Note that `col.alpha()` conveniently puts together the color and alpha level into a new color that can be directly used by functions like `lines()` and `polygon()`.

If two or more colors are given to `draw.lines()` they will be recycled through the drawing of the line segments. See the Examples Section.

Value

The function `select.pts()` explicitly returns a $(n \times 2)$ data frame of the values selected. However, `draw.polygon()` and `draw.lines()` return their values invisibly.

Note

If points inside a polygon were not selected try again with a less irregular polygon.

Author(s)

John R. Wallace: <Imap.for.R@gmail.com> (Limited support)

References

The `inside.polygon()` function is revised from the the function "Inside()" by Joseph S. Verducci (Snews: 09 Feb 1999). In particular, exactly equal adjacent x values will cause the original function to fail. See <http://www.biostat.wustl.edu/archives/html/s-news/2002-07/msg00020.html>

See Also

[imap](#), [gdist.total](#), [polygon](#)

Examples

```
## Not run:
```

```
plot(tmp <- cbind(1:100, rnorm(100)))
select.pts(tmp) # Left-click three or more times, right-click to stop.
```

```
big.island <- select.pts(imap()) # Zoom into the Hawaiian Islands, right-click to stop, then put a polygon around
```

```

imap(list(world.h.land, big.island)) # Re-zoom into the Hawaiian Islands. Use ilines() if a line was selected wh

imap() # Zoom in to an area, right-click to stop.
draw.lines() # Left-click two or more times, right-click to stop.
gdist.total(draw.lines(rainbow(12), lwd = 4, lty = 2)) # Left-click two or more times, right-click to stop.
draw.polygon('purple', alpha = 0.2) # Left-click three or more times, right-click to stop.

col.alpha('cyan', alpha = 0.3)

hist(rnorm(1e5), col = col.alpha('cyan', alpha = 0.3))
hist(rnorm(1e5, 2), col = col.alpha('magenta', alpha = 0.3), add=T)

## End(Not run)

```

world	<i>GSHHS - A Global Self-consistent, Hierarchical, High-resolution Shoreline Database</i>
-------	---

Description

Both the high and full resolutions of GSHHS have been converted for use in R. The high resolution of GSHHS is provided with the Imap package, the full resolution is available on the internet by using the following commands:

```

> browseURL("http://imap-for-r.googlecode.com/files/world.f.for.R.GSHHS.Ver.2.Dec.2009.dmp")
> load(choose.files(getwd(), "Find and load the file world.f.for.R.GSHHS.Ver.2.Dec.2009.dmp"))

```

Or without choose.files(), where 'saved.path' is the path where the file was saved:

```

> load("/saved.path/world.f.for.R.GSHHS.Ver.2.Dec.2009.dmp")

```

The full resolution will not work well on low-end computers. Start with 'ilines(world.f.land)' if there is a problem. Occasionally starting over with graphics.off() or using gc() for garbage collection may also help.

Format

A (n x 2) matrix with line breaks created with a row of NA's. Longitude is in the first column with lines in the Western Hemisphere being negative numbers. The latitude is in the second column with lines in the Southern Hemisphere being negative.

Details

All the GSHHS files are available: land, lake, island, pond.in.island, rivers, borders with either 'world.h' or 'world.f' prefix.

The 'npacific' dataset is a small matrix of the Northern Pacific coastline used in some examples involving lines (not polygons).

Source

<http://www.soest.hawaii.edu/wessel/gshhs/gshhs.html>

References

Wessel, P. and Smith, W.H.F., 1996. A global, self-consistent, hierarchical, high-resolution shore-line database. *J. Geophys. Res.*, 101, 8741-8743.

Examples

```
## Not run:

imap(list(world.h.land, world.h.lake, world.h.island, world.h.pond.in.island, world.h.rivers))

# The works with full resolution; high-end computers only.
imap(list(world.f.land, world.f.lake, world.f.island, world.f.pond.in.island, world.f.borders, world.f.rivers),
      col=c("black", "blue", "forestgreen", "dodgerblue", "cyan", col.alpha('magenta', alpha = 0.5)),
      poly = c("grey50", "blue", "forestgreen", "dodgerblue", NA, NA))

# The function below will plot selected polygons from 'world.h.land'.
imap.world <- function (x = 1:20, poly = rainbow(n), ...)
{
  n <- length(x)
  a <- 0
  polys <- vector("list", n)
  land.index <- (1:nrow(world.h.land))[is.na(world.h.land[,1])]

  for (i in 1:n)
    polys[[i + a]] <- world.h.land[land.index[x[i]]:land.index[x[i] + 1], ]

  imap(polys, col = 'black', poly = poly, keep.attr = TRUE, ...)

  invisible(polys)
}

imap.world()

imap.world(c(5,4,30,50))

## End(Not run)
```

Index

*Topic **datasets**

world, [11](#)

col.alpha (select.pts, draw.polygon,
inside.polygon, draw.lines,
col.alpha), [9](#)

draw.lines, [3](#)

draw.lines (select.pts, draw.polygon,
inside.polygon, draw.lines,
col.alpha), [9](#)

draw.polygon (select.pts,
draw.polygon, inside.polygon,
draw.lines, col.alpha), [9](#)

gdist, [2, 6](#)

gdist.total, [10](#)

ilines (imap), [4](#)

IMAP (IMAP-package), [2](#)

imap, [3, 4, 8, 10](#)

IMAP-package, [2](#)

imap.ll, [6, 7](#)

inside.polygon (select.pts,
draw.polygon, inside.polygon,
draw.lines, col.alpha), [9](#)

ipts (imap), [4](#)

npacific (world), [11](#)

polygon, [10](#)

select.lines (imap.ll), [7](#)

select.pts, [6, 8](#)

select.pts (select.pts, draw.polygon,
inside.polygon, draw.lines,
col.alpha), [9](#)

select.pts, draw.polygon,
inside.polygon, draw.lines,
col.alpha, [9](#)

world, [6, 11](#)