Package ‘plotdap’

October 18, 2023

Title Easily Visualize Data from 'ERDDAP' Servers via the 'rerddap' Package

Version 1.0.3

Date 2023-10-17

Description Easily visualize and animate 'tabledap' and 'griddap' objects obtained via the 'rerddap' package in a simple one-line command, using either base graphics or 'ggplot2' graphics. 'plotdap' handles extracting and reshaping the data, map projections and continental outlines. Optionally the data can be animated through time using the 'gganmiate' package.

License MIT + file LICENSE

URL `https://github.com/rmendels/plotdap`

BugReports `https://github.com/rmendels/plotdap/issues`

Depends R (>= 4.3.0)

Imports cmocean, dplyr, gganimate, ggnewscale, ggplot2 (>= 3.1.0), lazyeval, lubridate, magrittr, mapdata, maps, raster, rerddap (>= 0.8.0), scales, sf, tidyr, viridis

Suggests Cairo, knitr, rmarkdown, testthat

RoxygenNote 7.2.3

LazyData true

Encoding UTF-8

VignetteBuilder knitr

NeedsCompilation no

Author Carson Sievert [aut],
Roy Mendelssohn [aut, ctb, cre]

Maintainer Roy Mendelssohn <`roy.mendelssohn@noaa.gov`>

Repository CRAN

Date/Publication 2023-10-17 22:00:15 UTC

R topics documented:

add_ggplot 2
add_griddap 3
add_tabledap 4
bbox_set 6
murSST 7
plotdap 7
print.ggplotdap 9
print.plotdap 9
QMwind 10
sardines 10

Index **11**

`add_ggplot` *Add ggplot2 elements to a plotdap object*

Description

`add_ggplot` allows for plotdap ggplot maps to be modified by further ggplot2 settings

Usage

```
add_ggplot(plot, ...)
```

Arguments

`plot`	a plotdap object.
`...`	arguments passed along to `geom_sf()` (if `method='ggplot2'`, otherwise ignored).

Value

A plotdap object

Examples

```

p <- plotdap(
  crs = "+proj=laea +y_0=0 +lon_0=155 +lat_0=-90 +ellps=WGS84 +no_defs")
p <- add_ggplot(
 p,
 ggplot2::theme_bw()
)
```

add_griddap *Add rerddap::griddap() data to a plotdap map*

Description

`add_griddap` adds the data from an 'rerddap::griddap() call to a 'plotdap' map

Usage

```
add_griddap(
  plot,
  grid,
  var,
  fill = "viridis",
  maxpixels = 10000,
  time = mean,
  animate = FALSE,
  cumulative = FALSE,
  ...
)
```

Arguments

`plot`	a plotdap object.
`grid`	a griddap object.
`var`	a formula defining a variable, or function of variables to visualize.
`fill`	either a character string of length 1 matching a name in the package `cmocean` or a vector of color codes. This defines the colorscale used to encode values of `var`.
`maxpixels`	integer > 0. Maximum number of cells to use for the plot. If maxpixels < ncell(x), sampleRegular is used before plotting. If gridded=TRUE maxpixels may be ignored to get a larger sample
`time`	how to resolve multiple time frames. Choose one of the following: • A function to apply to each observation at a particular location (mean is the default). • A character string (of length 1) matching a time value.
`animate`	whether to animate over the `time` variable (if it exists). Currently only implemented for `method='ggplot2'` and requires the gganimate package.
`cumulative`	- if animation should be cumulative -default FALSE
`...`	arguments passed along to `geom_sf()` (if `method='ggplot2'`, otherwise ignored).

Value

A plotdap object

Examples

```
# base plotting tends to be faster,
# but is less extensible plotdap("base")

# actual datasets in data folder to meet execution timings


 # murSST <- rerddap::griddap(
 #  ' jplMURSST41', latitude = c(35, 40), longitude = c(-125, -120.5),
 #   time = c('last', 'last'), fields = 'analysed_sst'
 # )

 # QMwind <- rerddap::griddap(
 #  'erdQMwindmday', time = c('2016-11-16', '2017-01-16'),
 #  latitude = c(30, 50), longitude = c(210, 240),
 #  fields = 'x_wind'
 #  )

p <- plotdap(crs = "+proj=robin")
p <- add_griddap(p, murSST, ~analysed_sst)

 # p <- plotdap(mapTitle = "Average wind over time")
 # p <- add_griddap(p, QMwind, ~x_wind)

# p <- plotdap("base", crs = "+proj=robin")
# p <- add_griddap(p, murSST, ~analysed_sst)

# layer tables on top of grids
require(magrittr)
p <- plotdap("base") %>%
  add_griddap(murSST, ~analysed_sst) %>%
  add_tabledap(sardines, ~subsample_count)

# multiple time periods
p <- plotdap("base", mapTitle = "Average wind over time")
p <- add_griddap(p, QMwind, ~x_wind)
```

`add_tabledap` *Add rerddap::tabledap data to a plotdap map*

Description

`add_tabledap` adds the data from an 'rerddap::tabledap()' call to a 'plotdap' map

Usage

```
add_tabledap(
  plot,
  table,
  var,
  color = c("#132B43", "#56B1F7"),
  size = 1.5,
  shape = 19,
  animate = FALSE,
  cumulative = FALSE,
  ...
)
```

Arguments

`plot`	a plotdap object.
`table`	a tabledap object.
`var`	a formula defining a variable, or function of variables to visualize.
`color`	either a character string of length 1 matching a name in `cmocean` or a vector of color codes. This defines the colorscale used to encode values of `var`.
`size`	the size of the symbol.
`shape`	the shape of the symbol. For valid options, see the 'pch' values section on points. `plot(0:25, 0:25, pch = 0:25)` also gives a quick visual of the majority of possibilities.
`animate`	whether to animate over the `time` variable (if it exists). Currently only implemented for `method='ggplot2'` and requires the gganimate package.
`cumulative`	- if animation should be cumulative -default FALSE
`...`	arguments passed along to `geom_sf()` (if `method='ggplot2'`, otherwise ignored).

Value

A plotdap object

Examples

```
# base plotting tends to be faster,
# but is less extensible plotdap("base")

# test datasets in data folder to meet execution timings
# code given to extract the data


sardines <- rerddap::tabledap(
 'FRDCPSTrawlLHHaulCatch',
 fields = c('latitude',  'longitude', 'time', 'scientific_name', 'subsample_count'),
  'time>=2010-01-01', 'time<=2012-01-01',
  'scientific_name="Sardinops sagax"'
  )
```

```
p <- plotdap()
p1 <- add_tabledap(p, sardines, ~subsample_count)
p2 <- add_tabledap(p, sardines, ~log2(subsample_count))

# using base R plotting
p <- plotdap("base")
p <- add_tabledap(p, sardines, ~subsample_count)

# robinson projection
p <- plotdap(crs = "+proj=robin")
p <- add_tabledap(p, sardines, ~subsample_count)
```

bbox_set *change bounding box in plotdap object*

Description

bbox_setchanges the bounding box in an plotdap object. Particularly needed if using gganimate::animate()

Usage

```
bbox_set(plotobj, xlim, ylim)
```

Arguments

plotobj	valid plotdap object
xlim	new x-values of the bounding box
ylim	new y-values of the bounding box

Value

a plotdap object

Examples

```
p <- plotdap()
p <- add_tabledap(p, sardines, ~subsample_count)
xlim = c(-125, -115)
ylim <- c(30., 50.)
p <- bbox_set(p, xlim, ylim)
```

`murSST` *murSST Data*

Description

pre-Download of murSST in ‘add_griddap()‘ example so that example can run within CRAN Time limits

Usage

```
murSST
```

Format

An object of class `griddap_nc` (inherits from `nc`, `data.frame`) with 0 rows and 2 columns.

Details

obtained using the ‘rerddap‘ command murSST <- griddap(’jplMURSST41’, latitude = c(22, 51), longitude = c(-140, -105), time = c(’last’, ’last’), fields = ’analysed_sst’)

`plotdap` *Visualize rerddap data*

Description

Visualize data returned from rerddap servers. Use `plotdap()` to initialize a plot, specify the plotting method (specifically, ’base’ or ’ggplot2’), and set some global options/parameters. Then use `add_tabledap()` and/or `add_griddap()` to add "layers" of actual data to be visualized.

Usage

```
plotdap(
  method = c("ggplot2", "base"),
  mapData = maps::map("world", plot = FALSE, fill = TRUE),
  crs = NULL,
  datum = sf::st_crs(4326),
  mapTitle = NULL,
  mapFill = "gray80",
  mapColor = "gray90",
  ...
)
```

Arguments

`method`	the plotting method. Currently ggplot2 and base plotting are supported.
`mapData`	an object coercable to an sf object via `st_as_sf()`.
`crs`	a coordinate reference system: integer with the epsg code, or character with proj4string.
`datum`	crs that provides datum to use when generating graticules. Set to `NULL` to hide the graticule.
`mapTitle`	a title for the map.
`mapFill`	fill used for the map.
`mapColor`	color used to draw boundaries of the map.
`...`	arguments passed along to `geom_sf()` (if `method='ggplot2'`, otherwise ignored).

Details

The "ggplot2" method is slower than "base" (especially for high-res grids/rasters), but is more flexible/extensible. Additional ggplot2 layers, as well as scale defaults, labels, theming, etc. may be modified via the `add_ggplot()` function. See the mapping vignette for an introduction and overview of rerddap's visualization methods – `browseVignettes(package = "rerddap")`.

Value

A plotdap object

Author(s)

Carson Sievert

See Also

`tabledap()`, `griddap()`

Examples

```
# base plotting tends to be faster (especially for grids),
# but is less extensible plotdap("base")

 plotdap()
 plotdap("base")
```

`print.ggplotdap` *Print a ggplot plotdap object*

Description

Print a ggplot plotdap object

Usage

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

Arguments

`x`	a ggplotdap object
`...`	currently unused

`print.plotdap` *Print a plotdap object*

Description

Print a plotdap object

Usage

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

Arguments

`x`	a plotdap object
`...`	currently unused

`QMwind` *QMwind Data*

Description

pre-Download of QMwind in ‘add_griddap()‘ example so that example can run within CRAN Time limits

Usage

```
QMwind
```

Format

An object of class `griddap_nc` (inherits from `nc`, `data.frame`) with 0 rows and 2 columns.

Details

obtained using the ‘rerddap‘ command wind <- griddap(’erdQMwindmday’, time = c(’2016-11-16’, ’2017-01-16’), latitude = c(30, 50), longitude = c(210, 240), fields = ’x_wind’))

`sardines` *sardine Data*

Description

pre-Download of sardine data in ‘add_tabledap()‘ example so that example can run within CRAN Time limits

Usage

```
sardines
```

Format

An object of class `tabledap` (inherits from `data.frame`) with 56 rows and 5 columns.

Details

obtained using the ‘rerddap‘ command sardines <- tabledap(’FRDCPSTrawlLHHaulCatch’, fields = c(’latitude’, ’longitude’, ’time’, ’ scientific_name’, ’subsample_count’), ’time>=2010-01-01’, ’time<=2012-01-01’, ’scientific_name="Sardinops sagax"’))

Index

∗ **datasets**
    murSST, 7
    QMwind, 10
    sardines, 10

add_ggplot, 2
add_griddap, 3
add_tabledap, 4

bbox_set, 6

griddap, *3*

mean, *3*
murSST, 7

plotdap, *3*, *5*, 7
points, *5*
print.ggplotdap, 9
print.plotdap, 9

QMwind, 10

sardines, 10

tabledap, *5*