Package 'mapr'

November 6, 2016

Title Visualize Species Occurrence Data

Description Utilities for visualizing species occurrence data. Includes functions to visualize occurrence data from 'spocc', 'rgbif', and other packages. Mapping options included for base R plots, 'ggplot2', 'ggmap', 'leaflet' and 'GitHub' 'gists'.

Version 0.3.0

License MIT + file LICENSE

URL https://github.com/ropensci/mapr

BugReports https://github.com/ropensci/mapr/issues

LazyData true

LazyLoad true

VignetteBuilder knitr

Imports ggplot2, leaflet, spocc (>= 0.5.4), sp, rworldmap, RColorBrewer, httr (>= 1.2.0), gistr

Suggests testthat, ggmap, knitr, taxize, maptools, covr

RoxygenNote 5.0.1

NeedsCompilation no

Author Scott Chamberlain [aut, cre]

Maintainer Scott Chamberlain <myrmecocystus@gmail.com>

Repository CRAN

Date/Publication 2016-11-06 00:19:24

R topics documented:

`mapr-package` *Visualize species occurrence data*

Description

Visualize species occurrence data

Many inputs

All functions take the following kinds of inputs:

- An object of class `occdat`, from the package **spocc**. An object of this class is composed of many objects of class `occdatind`
- An object of class `occdatind`, from the package **spocc**
- An object of class `gbif`, from the package **rgbif**
- An object of class `data.frame`. This data.frame can have any columns, but must include a column for taxonomic names (e.g., `name`), and for latitude and longitude (we guess your lat/long columns, starting with the default `latitude` and `longitude`)
- An object of class `SpatialPoints`
- An object of class `SpatialPointsDatFrame`

Package API

- `map_plot` - static Base R plots
- `map_ggplot` - static ggplot2 plots
- `map_ggmap` - static ggplot2 plots with map layers
- `map_leaflet` - interactive Leaflet.js interactive maps
- `map_gist` - ineractive, shareable maps on GitHub Gists

Author(s)

Scott Chamberlain <myrmecocystus@gmail.com>

gbif_eg1 *Example dataset: output from call to occ_search*

Description

A dataset with 50 rows, and 101 columns, from the query: `rgbif::occ_search(scientificName = "Puma concolor", l`

Format

A data frame with 50 rows and 101 variables

hull *Add a convex hull to a map*

Description

Add a convex hull to a map

Usage

```
hull(x, ...)
```

Arguments

x	input
...	ignored

Details

Can be used with `map_leaflet`, `map_plot`, and `map_ggplot`. Other methods in this package may be supported in the future.

Value

Adds a convex hull to the plot. See `chull` for info.

Examples

```
# map spocc output, here using a built in object
data(occdat_eg1)
map_plot(occdat_eg1, hull = TRUE)

# map rgbif output, here using a built in object
hull(map_ggplot(occdat_eg1))
```

map_ggmap *ggpmap visualization of species occurences*

Description

ggpmap visualization of species occurences

Usage

```
map_ggmap(x, zoom = 3, point_color = "#86161f", color = NULL, size = 3,
  lon = "longitude", lat = "latitude", maptype = "terrain",
  source = "google", ...)
```

Arguments

x	The data. An object of class occdat, occdatind, gbif, SpatialPoints, SpatialPointsDataFrame, or data.frame. The package **spocc** needed for the first two, and **rgbif** needed for the third. When data.frame input, any number of columns allowed, but with at least the following: name (the taxonomic name), latitude (in dec. deg.), longitude (in dec. deg.)
zoom	zoom level for map. Adjust depending on how your data look.
point_color	Default color of your points. Deprecated, use color
color	Default color of your points.
size	point size, Default: 3
lon, lat	(character) Longitude and latitude variable names. Ignored unless data.frame input to x parameter. We attempt to guess, but if nothing close, we stop. Default: longitude and latitude
maptype	(character) map theme. see get_map in ggmap for options. Default: none
source	(character) Google Maps ("google"), OpenStreetMap ("osm"), Stamen Maps ("stamen"), or CloudMade maps ("cloudmade"). Default: osm
...	Ignored

Details

Does not support adding a convex hull via hull

Examples

```
## Not run:
## spocc
library("spocc")
gd <- occ(query = 'Accipiter striatus', from = 'gbif', limit=75,
  has_coords = TRUE)
map_ggmap(gd)
map_ggmap(gd$gbif)
```

```
## rgbif
library("rgbif")
res <- occ_search(scientificName = "Puma concolor", limit = 100)
map_ggmap(res)

## data.frame
df <- data.frame(name = c('Poa annua', 'Puma concolor', 'Foo bar'),
                 longitude = c(-120, -121, -123),
                 latitude = c(41, 42, 45), stringsAsFactors = FALSE)
map_ggmap(df)

### usage of occ2sp()
#### SpatialPointsDataFrame
spdat <- occ2sp(gd)
map_ggmap(spdat)

# many species, each gets a different color
library("spocc")
spp <- c('Danaus plexippus', 'Accipiter striatus', 'Pinus contorta')
dat <- occ(spp, from = 'gbif', limit = 30, has_coords = TRUE,
  gbifopts = list(country = 'US'))
map_ggmap(dat)
map_ggmap(dat, zoom = 5)
map_ggmap(dat, color = '#6B944D')
map_ggmap(dat, color = c('#976AAE', '#6B944D', '#BD5945'))

## End(Not run)
```

`map_ggplot` *ggplot2 mapping*

Description

ggplot2 mapping

Usage

```
map_ggplot(x, map = "world", point_color = "#86161f", color = NULL,
  size = 3, lon = "longitude", lat = "latitude", ...)
```

Arguments

`x`	The data. An object of class `occdat`, `occdatind`, `gbif`, `SpatialPoints`, `SpatialPointsDataFrame`, or `data.frame`. The package **spocc** needed for the first two, and **rgbif** needed for the third. When `data.frame` input, any number of columns allowed, but with at least the following: name (the taxonomic name), latitude (in dec. deg.), longitude (in dec. deg.)
`map`	(character) One of world, world2, state, usa, county, france, italy, or nz

point_color	Default color of your points. Deprecated, use color
color	Default color of your points.
size	point size, Default: 3
lon, lat	(character) Longitude and latitude variable names. Ignored unless data.frame input to x parameter. We attempt to guess, but if nothing close, we stop. Default: longitude and latitude
...	Ignored

Value

A ggplot2 map, of class gg/ggplot

Examples

```
# map spocc output, here using a built in object
data(occdat_eg1)
map_ggplot(occdat_eg1)

# map rgbif output, here using a built in object
data(gbif_eg1)
map_ggplot(gbif_eg1)
```

---

map_gist *Make an interactive map to view in the browser as a GitHub gist*

Description

Make an interactive map to view in the browser as a GitHub gist

Usage

```
map_gist(x, description = "", public = TRUE, browse = TRUE,
  lon = "longitude", lat = "latitude", ...)
```

Arguments

x	The data. An object of class occdat, occdatind, gbif, SpatialPoints, SpatialPointsDataFrame, or data.frame. The package **spocc** needed for the first two, and **rgbif** needed for the third. When data.frame input, any number of columns allowed, but with at least the following: name (the taxonomic name), latitude (in dec. deg.), longitude (in dec. deg.)
description	Description for the Github gist, or leave to default (=no description)
public	(logical) Whether gist is public (default: TRUE)
browse	If TRUE (default) the map opens in your default browser.

lon, lat	(character) Longitude and latitude variable names. Ignored unless data.frame input to x parameter. We attempt to guess, but if nothing close, we stop. Default: longitude and latitude
...	Further arguments passed on to style_geojson

Details

See gist_auth for help on authentication

Does not support adding a convex hull via hull

Examples

```
## Not run:
## spocc
library("spocc")
spp <- c('Danaus plexippus', 'Accipiter striatus', 'Pinus contorta')
dat <- occ(spp, from=c('gbif','ecoengine'), limit=30,
  gbifopts=list(hasCoordinate=TRUE))
dat <- fixnames(dat, "query")

# Define colors
map_gist(dat, color=c('#976AAE','#6B944D','#BD5945'))
map_gist(dat$gbif, color=c('#976AAE','#6B944D','#BD5945'))
map_gist(dat$ecoengine, color=c('#976AAE','#6B944D','#BD5945'))

# Define colors and marker size
map_gist(dat, color=c('#976AAE','#6B944D','#BD5945'),
  size=c('small','medium','large'))

# Define symbols
map_gist(dat, symbol=c('park','zoo','garden'))

## rgbif
library("rgbif")
res <- occ_search(scientificName = "Puma concolor", limit = 100)
map_gist(res)

## data.frame
df <- data.frame(name = c('Poa annua', 'Puma concolor', 'Foo bar'),
                 longitude = c(-120, -121, -121),
                 latitude = c(41, 42, 45), stringsAsFactors = FALSE)
map_gist(df)

### usage of occ2sp()
#### SpatialPoints
spdat <- occ2sp(dat)
map_gist(spdat)
#### SpatialPointsDataFrame
spdatdf <- as(spdat, "SpatialPointsDataFrame")
map_gist(spdatdf)

## End(Not run)
```

`map_leaflet` *Make interactive maps with Leaflet.js*

Description

Make interactive maps with Leaflet.js

Usage

```
map_leaflet(x, lon = "longitude", lat = "latitude", color = NULL,
  size = 13, ...)
```

Arguments

`x`	The data. An object of class `occdat`, `occdatind`, `gbif`, `SpatialPoints`, `SpatialPointsDataFrame`, or `data.frame`. The package **spocc** needed for the first two, and **rgbif** needed for the third. When `data.frame` input, any number of columns allowed, but with at least the following: name (the taxonomic name), latitude (in dec. deg.), longitude (in dec. deg.)
`lon, lat`	(character) Longitude and latitude variable names. Ignored unless `data.frame` input to `x` parameter. We attempt to guess, but if nothing close, we stop. Default: `longitude` and `latitude`
`color`	Default color of your points.
`size`	point size, Default: 13
`...`	Ignored

Details

We add popups by default, and add all columns to the popup. The html is escaped with `htmlEscape`

Value

a Leaflet map in Viewer in Rstudio, or in your default browser otherwise

Examples

```
## Not run:
## spocc
library("spocc")
(out <- occ(query='Accipiter striatus', from='gbif', limit=50,
  has_coords=TRUE))
### with class occdat
map_leaflet(out)
### with class occdatind
map_leaflet(out$gbif)
### use occ2sp
map_leaflet(occ2sp(out))
```

```
## rgbif
library("rgbif")
res <- occ_search(scientificName = "Puma concolor", limit = 100)
map_leaflet(res)

## SpatialPoints class
library("sp")
df <- data.frame(longitude = c(-120,-121),
                 latitude = c(41, 42), stringsAsFactors = FALSE)
x <- SpatialPoints(df)
map_leaflet(x)

## SpatialPointsDataFrame class
library("rgbif")
res <- occ_search(scientificName = "Puma concolor", limit = 100)
x <- res$data
library("sp")
x <- x[stats::complete.cases(x$decimalLatitude, x$decimalLongitude), ]
coordinates(x) <- ~decimalLongitude+decimalLatitude
map_leaflet(x)

## data.frame
df <- data.frame(name = c('Poa annua', 'Puma concolor'),
                 longitude = c(-120,-121),
                 latitude = c(41, 42), stringsAsFactors = FALSE)
map_leaflet(df)

# many species
library("spocc")
spp <- c('Danaus plexippus', 'Accipiter striatus', 'Pinus contorta')
dat <- occ(spp, from = 'gbif', limit = 50, has_coords = TRUE)
map_leaflet(dat)
map_leaflet(dat, color = c('#AFFF71', '#AFFF71', '#AFFF71'))
map_leaflet(dat, color = c('#976AAE', '#6B944D', '#BD5945'))

# add a convex hull
## map_leaflet(dat) %>% hull()  # using pipes
hull(map_leaflet(dat))

## End(Not run)
```

`map_plot` *Base R mapping*

Description

Base R mapping

Usage

```
map_plot(x, lon = "longitude", lat = "latitude", color = NULL, size = 1,
  pch = 16, hull = FALSE, ...)
```

Arguments

x	The data. An object of class occdat, occdatind, gbif, SpatialPoints, SpatialPointsDataFrame, or data.frame. The package **spocc** needed for the first two, and **rgbif** needed for the third. When data.frame input, any number of columns allowed, but with at least the following: name (the taxonomic name), latitude (in dec. deg.), longitude (in dec. deg.)
lon, lat	(character) Longitude and latitude variable names. Ignored unless data.frame input to x parameter. We attempt to guess, but if nothing close, we stop. Default: longitude and latitude
color	Default color of your points.
size	point size, Default: 1
pch	point symbol shape, Default: 16
hull	(logical) whether to add a convex hull. Default: FALSE
...	Further args to points

Value

Plots a world scale map

Examples

```
# map spocc output, here using a built in object
data(occdat_eg1)
map_plot(occdat_eg1)

# map rgbif output, here using a built in object
data(gbif_eg1)
map_plot(gbif_eg1)
```

---

occ2sp *Create a spatial points dataframe from a spocc search*

Description

Create a spatial points dataframe from a spocc search

Usage

```
occ2sp(x, coord_string = "+proj=longlat +datum=WGS84", just_coords = FALSE)
```

Arguments

x	The resuslts of a spocc search called by occ()
coord_string	A valid EPGS cooridate string from the sp package, the default is WSGS 84
just_coords	Return data frame with specios names and provenance or just a spatial points object, which is the default.

Details

This function will return either a spatial points dataframe or spatial points object. Conversion to spatial points objects allows spocc searches to interact with other spatial data sources. More coordinate system codes can be found at the EPGS registry: http://www.epsg-registry.org/

Examples

```
## Not run:
### See points on a map
library("maptools")
library("spocc")
data(wrld_simpl)
plot(wrld_simpl[wrld_simpl$NAME == "United States", ], xlim = c(-70, -60))
out <- occ(query = "Accipiter striatus", from = c("vertnet", "gbif"),
  limit = 50)
xx <- occ2sp(out, just_coords = TRUE)
points(xx, col = 2)

## End(Not run)
```

---

occdat_eg1 *Example dataset: output from call to occ*

Description

A dataset with 25 rows, and 62 columns, from the query: `spocc::occ(query='Accipiter striatus', from='gbif', li`

Format

A data frame with 25 rows and 62 variables

style_geojson *Style a data.frame prior to converting to geojson.*

Description

Style a data.frame prior to converting to geojson.

Usage

```
style_geojson(input, var = NULL, var_col = NULL, var_sym = NULL,
  var_size = NULL, color = NULL, symbol = NULL, size = NULL)
```

Arguments

input	A data.frame
var	A single variable to map colors, symbols, and/or sizes to.
var_col	The variable to map colors to.
var_sym	The variable to map symbols to.
var_size	The variable to map size to.
color	Valid RGB hex color
symbol	An icon ID from the Maki project http://www.mapbox.com/maki/ or a single alphanumeric character (a-z or 0-9).
size	One of 'small', 'medium', or 'large'

Index