

Package ‘mapr’

April 11, 2017

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 0.3.4

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 (>= 0.6.0), sp, rworldmap, RColorBrewer, jsonlite, gistr

Suggests testthat, ggmap, knitr, taxize, maptools

RoxygenNote 6.0.1

NeedsCompilation no

Author Scott Chamberlain [aut, cre]

Maintainer Scott Chamberlain <myrmecocystus@gmail.com>

Repository CRAN

Date/Publication 2017-04-11 21:19:47 UTC

R topics documented:

mapr-package	2
gbif_eg1	3
hull	3
map_ggmap	4
map_ggplot	6
map_gist	7

map_leaflet	9
map_plot	11
occ2sp	13
occdat_eg1	14
style_geojson	14

Index	15
--------------	-----------

mapr-package	<i>Visualize species occurrence data</i>
--------------	--

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 `SpatialPointsDataFrame`

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 `rgbif::occ_search()`*

Description

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

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

<code>x</code>	input
<code>...</code>	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 `grDevices::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))

## Not run:
```

```

# leaflet
library("spocc")
spp <- c('Danaus plexippus', 'Accipiter striatus', 'Pinus contorta')
dat <- occ(spp, from = 'gbif', limit = 30, has_coords = TRUE)
hull(map_leaflet(dat))

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

# base plots
library("spocc")
out <- occ(query='Accipiter striatus', from='gbif', limit=25,
  has_coords=TRUE)
map_plot(out, hull = TRUE)

## End(Not run)

```

map_ggmap

ggpmap visualization of species occurrences

Description

ggpmap visualization of species occurrences

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 <code>occdat</code> , <code>occdatind</code> , <code>gbif</code> , <code>SpatialPoints</code> , <code>SpatialPointsDataFrame</code> , or <code>data.frame</code> . The package spocc needed for the first two, and rgbif needed for the third. When <code>data.frame</code> input, any number of columns allowed, but with at least the following: <code>name</code> (the taxonomic name), <code>latitude</code> (in dec. deg.), <code>longitude</code> (in dec. deg.)
zoom	zoom level for map. Adjust depending on how your data look.
point_color	Default color of your points. Deprecated, use <code>color</code>
color	Default color of your points.
size	point size, Default: 3
lon, lat	(character) Longitude and latitude variable names. Ignored unless <code>data.frame</code> input to <code>x</code> parameter. We attempt to guess, but if nothing close, we stop. Default: <code>longitude</code> and <code>latitude</code>
maptype	(character) map theme. see <code>get_map</code> in <code>ggmap</code> 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\(\)](#)

Note

BEWARE: this may error for you with a message like *GeomRasterAnn was built with an incompatible version of ggproto*. This is fixed in the dev version of ggmap, but not in the CRAN version. Apologies for the problem.

Examples

```
## Not run:
# BEWARE: this may error for you with a message like
# "GeomRasterAnn was built with an incompatible version of ggproto".
# This is fixed in the dev version of `ggmap`, but not in the CRAN
# version. Apologies for the problem.

## 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 <code>occdat</code> , <code>occdatind</code> , <code>gbif</code> , <code>SpatialPoints</code> , <code>SpatialPointsDataFrame</code> , or <code>data.frame</code> . The package spocc needed for the first two, and rgbif needed for the third. When <code>data.frame</code> input, any number of columns allowed, but with at least the following: <code>name</code> (the taxonomic name), <code>latitude</code> (in dec. deg.), <code>longitude</code> (in dec. deg.)
map	(character) One of <code>world</code> , <code>world2</code> , <code>state</code> , <code>usa</code> , <code>county</code> , <code>france</code> , <code>italy</code> , or <code>nz</code>
point_color	Default color of your points. Deprecated, use <code>color</code>
color	Default color of your points.
size	point size, Default: 3
lon, lat	(character) Longitude and latitude variable names. Ignored unless <code>data.frame</code> input to <code>x</code> parameter. We attempt to guess, but if nothing close, we stop. Default: <code>longitude</code> and <code>latitude</code>
...	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)
```

```

## Not run:
## spocc
library("spocc")
ddat <- occ(query = 'Lynx rufus californicus', from = 'gbif', limit=100)
map_ggplot(ddat)
map_ggplot(ddat$gbif)
map_ggplot(v, "usa")
map_ggplot(ddat, "county")

### usage of occ2sp()
#### SpatialPoints
spdat <- occ2sp(ddat)
map_ggplot(spdat)
#### SpatialPointsDataFrame
spdatdf <- as(spdat, "SpatialPointsDataFrame")
map_ggplot(spdatdf)

## rgbif
library("rgbif")
library("ggplot2")
res <- occ_search(scientificName = "Puma concolor", limit = 100)
map_ggplot(res)
## add a convex hull
map_ggplot(res) + hull()

## 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_ggplot(df)

# 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)
map_ggplot(dat, color = c('#976AAE', '#6B944D', '#BD5945'))

## End(Not run)

```

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 <code>occcdat</code> , <code>occcdatind</code> , <code>gbif</code> , <code>SpatialPoints</code> , <code>SpatialPointsDataFrame</code> , or <code>data.frame</code> . The package spocc needed for the first two, and rgbif needed for the third. When <code>data.frame</code> input, any number of columns allowed, but with at least the following: <code>name</code> (the taxonomic name), <code>latitude</code> (in dec. deg.), <code>longitude</code> (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 <code>data.frame</code> input to <code>x</code> parameter. We attempt to guess, but if nothing close, we stop. Default: <code>longitude</code> and <code>latitude</code>
...	Further arguments passed on to <code>style_geojson()</code>

Details

See `gistr::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 <code>occdat</code> , <code>occdatind</code> , <code>gbif</code> , <code>SpatialPoints</code> , <code>SpatialPointsDataFrame</code> , or <code>data.frame</code> . The package spocc needed for the first two, and rgbif needed for the third. When <code>data.frame</code> input, any number of columns allowed, but with at least the following: <code>name</code> (the taxonomic name), <code>latitude</code> (in dec. deg.), <code>longitude</code> (in dec. deg.)
lon, lat	(character) Longitude and latitude variable names. Ignored unless <code>data.frame</code> input to <code>x</code> parameter. We attempt to guess, but if nothing close, we stop. Default: <code>longitude</code> and <code>latitude</code>
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 `htmltools::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 <code>occdat</code> , <code>occdatind</code> , <code>gbif</code> , <code>SpatialPoints</code> , <code>SpatialPointsDataFrame</code> , or <code>data.frame</code> . The package spocc needed for the first two, and rgbif needed for the third. When <code>data.frame</code> input, any number of columns allowed, but with at least the following: <code>name</code> (the taxonomic name), <code>latitude</code> (in dec. deg.), <code>longitude</code> (in dec. deg.)
lon, lat	(character) Longitude and latitude variable names. Ignored unless <code>data.frame</code> input to <code>x</code> parameter. We attempt to guess, but if nothing close, we stop. Default: <code>longitude</code> and <code>latitude</code>
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 <code>graphics::points()</code>

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)

## Not run:
## spocc
library("spocc")
```

```

(out <- occ(query='Accipiter striatus', from='gbif', limit=25,
  has_coords=TRUE))
### class occdat
map_plot(out)
map_plot(out, hull = TRUE)
### class occdatind
map_plot(out$gbif)
map_plot(out$gbif, hull = TRUE)

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

## data.frame
df <- data.frame(
  name = c('Poa annua', 'Puma concolor', 'Foo bar', 'Stuff things'),
  longitude = c(-125, -123, -121, -110),
  latitude = c(41, 42, 45, 30), stringsAsFactors = FALSE)
map_plot(df)
map_plot(df, hull = TRUE)

### usage of occ2sp()
#### SpatialPoints
spdat <- occ2sp(out)
map_plot(spdat)
map_plot(spdat, hull = TRUE)

#### SpatialPointsDataFrame
spdatdf <- as(spdat, "SpatialPointsDataFrame")
map_plot(spdatdf)
map_plot(spdatdf, hull = TRUE)

# many species, each gets a different color
library("spocc")
spp <- c('Danaus plexippus', 'Accipiter striatus', 'Pinus contorta',
  'Ursus americanus')
dat <- occ(spp, from = 'gbif', limit = 30, has_coords = TRUE,
  gbifopts = list(country = 'US'))
map_plot(dat)
map_plot(dat, hull = TRUE)
## diff. color for each taxon
map_plot(dat, color = c('#976AAE', '#6B944D', '#BD5945', 'red'))
map_plot(dat, color = c('#976AAE', '#6B944D', '#BD5945', 'red'), hull = TRUE)

# add a convex hull
library("rgbif")
res <- occ_search(scientificName = "Puma concolor", limit = 100)
map_plot(res, hull = FALSE)
map_plot(res, hull = TRUE)

## End(Not run)

```

occ2sp	<i>Create a spatial points dataframe from a spocc search</i>
--------	--

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 results of a spocc search called by <code>spocc::occ()</code>
coord_string	A valid EPSG coordinate string from the <code>sp</code> package, the default is WGS 84
just_coords	Return data frame with species 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 EPSG 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 `spocc::occ()`

Description

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

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

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

Index

*Topic **datasets**

gbif_eg1, [3](#)

occdat_eg1, [14](#)

*Topic **package**

mapr-package, [2](#)

gbif_eg1, [3](#)

gistr::gist_auth(), [8](#)

graphics::points(), [11](#)

grDevices::chull(), [3](#)

htmltools::htmlEscape(), [9](#)

hull, [3](#)

hull(), [5, 8](#)

map_ggmap, [4](#)

map_ggmap(), [2](#)

map_ggplot, [6](#)

map_ggplot(), [2, 3](#)

map_gist, [7](#)

map_gist(), [2](#)

map_leaflet, [9](#)

map_leaflet(), [2, 3](#)

map_plot, [11](#)

map_plot(), [2, 3](#)

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

mapr-package, [2](#)

occ2sp, [13](#)

occdat_eg1, [14](#)

rgbif::occ_search(), [3](#)

spocc::occ(), [13, 14](#)

style_geojson, [14](#)

style_geojson(), [8](#)