Package ‘rasterList’

November 30, 2022

License GPL (>= 3)

Title A Raster Where Cells are Generic Objects

Type Package

Description A S4 class has been created such that complex operations can be executed on each cells of a raster map. The raster of objects contains the traditional raster map with the addition of a list of generic objects: one object for each raster cells. It allows to write few lines of R code for complex map algebra. Two environmental applications about frequency analysis of raster map of precipitation and creation of a raster map of soil water retention curves have been presented.

Version 0.5.14

Date 2022-11-29

Repository CRAN

Depends R(>= 3.0.1),raster,methods

Suggests lmom,testthat,sp,soilwater,lubridate

RoxygenNote 7.1.2

NeedsCompilation no

Date/Publication 2022-11-30 11:20:05 UTC

Author Emanuele Cordano [aut, cre]

Maintainer Emanuele Cordano <emanuele.cordano@gmail.com>

R topics documented:

crop,RasterList-method 2
is.RasterList 3
raster,RasterList-method 4
rasterList 5
RasterList-class 6
RasterList-Package 7
RasterListApply 8
rasterListFun 9
stack,RasterList-method 12

Index **14**

crop,RasterList-method

Crop methods for a RasterList-class object.

Description

Crop methods for a RasterList-class object.

Usage

```
## S4 method for signature 'RasterList'
crop(x, y, check.RasterList = TRUE, ...)
```

Arguments

x a valid object

y a Spatial Object or an Extent

check.RasterList logical value. If it is TRUE, it checks the x RasterList-class object. Default is FALSE.

... further arguments

Value

a "cropped" RasterList-class object

Examples

```
precf <- system.file(”map/precipitation.grd”, package=”rasterList”)
prec <- stack(precf)

## Sample L-moments

library(lmom)

samlmom <- stack(rasterList(prec,FUN=samlmu))
## Fitting a Random Probability Distribution: it is a 'rasterList' Object
fitdist <- rasterList(samlmom,FUN=pelgam)

##### ZOOM IN
## set a mask
mask <-raster( extent(fitdist)/4 )


fitdist_masked <- crop ( x = fitdist,y=mask)
```

is.RasterList *Is a* RasterList-class *object ?*

Description

Is a RasterList-class object ?

Usage

```
is.RasterList(x)
```

Arguments

x	a valid object

Value

a logical variable

Examples

```
r <- rasterList()
is.RasterList(r)
rr <- raster()
is.RasterList(rr)

f <- system.file("external/test.grd", package="raster")
ra <- rasterList(f)
is.RasterList(rr)
```

`raster,RasterList-method`

`Raster` methods for a `RasterList-class` object.

Description

`Raster` methods for a `RasterList-class` object.

Usage

```
## S4 method for signature 'RasterList'
raster(x, FUN = NULL, ...)
```

Arguments

`x`	a valid `RasterList-class` object
`FUN`	if it not `NULL` a function is applied to all elements of the `list` slot in `x`.
`...`	further arguments

Value

a `RasterLayer-class` object

See Also

`stack`,`RasterListApply`

Examples

```
f <- system.file("external/test.grd", package="raster")
ur <- rasterList(raster(f),FUN=function(x,d){x+0:d},d=10)

r1 <- raster(ur)
r2 <- raster(ur,FUN=function(x){x[2]})
```

rasterList *Creates a RasterList-class object*

Description

The method rasterList is the constructor of a RasterList-class from a generic object.

Usage

```
rasterList(object = NULL, list = NULL, object.name = NA, ...)

as.RasterList(object, ...)

rasterList(object = NULL, list = NULL, object.name = NA, ...)

## S4 method for signature 'RasterLayer'
rasterList(object = NULL, list = NULL, object.name = NA, ...)

## S4 method for signature 'RasterStack'
rasterList(object = NULL, list = NULL, object.name = NA, ...)

## S4 method for signature 'RasterBrick'
rasterList(object = NULL, list = NULL, object.name = NA, ...)

## S4 method for signature 'RasterList'
rasterList(object, list = NULL, object.name = NA, FUN = NULL, ...)
```

Arguments

object the object to coerce

list a list object to assign to the raster map.

object.name character string containing the name to assign to object.

... further arguments for raster(generic) or FUN (RasterList-class)

FUN function that can be used to apply to each element of the list in a RasterList-class

Details

The argument FUN is useful to create or transform RasterList-class from other Raster* classes.

Value

a RasterList-class object

Examples

```
f <- system.file("external/test.grd", package="raster")
rr <- rasterList(f)
rs <- as.RasterList(f)
# The package-provided datasets shall be only used as example datasets.
precf <- system.file("map/precipitation.grd", package="rasterList")##
## A resampled preciptation raster map based on CHIRS dataset:
## Funk, Chris, Pete Peterson, Martin Landsfeld, Diego Pedreros, James Verdin,
## Shraddhanand Shukla, Gregory Husak, James Rowland, Laura Harrison,
## Andrew Hoell and Joel Michaelsen.
## "The climate hazards infrared precipitation with stations - a new environmental
## record for monitoring extremes". Scientific Data 2, 150066. doi:10.1038/sdata.2015.66 2015.
## http://chg.geog.ucsb.edu/data/chirps/
##


## Sample L-moments
 library(lmom)

prec <- stack(precf)
samlmom <- stack(rasterList(prec,FUN=samlmu))
## Fitting a Random Probability Distribution: it is a 'rasterList' Object
fitdist <- rasterList(samlmom,FUN=pelgam)


precf <- system.file("map/Mekrou_precipitation.grd", package="rasterList")
prec <- stack(precf)
 # Set time
time <- as.Date(names(prec),format="X%Y.%m.%d")
year <- as.character(time,format="X%Y")

## Compute Annual Precipitation (sum aggregration)
yearlyprec <- stackApply(x=prec,fun=sum,indices=year)
## L-moments
samlmom <- stack(rasterList(yearlyprec,FUN=samlmu))
fitdist <- rasterList(samlmom,FUN=pelgam)
```

RasterList-class *RasterList-class*

Description

Class RasterList

Details

It contains RasterLayer-class with the following adjoint slots:

`list`: a list of generic object whese length corresponds to the number of cells. Each `list` element for each cell;

`name`: an identification name of the object. Default is `NA`.

This class inherits the `RasterLayer-class` class considering each pixel of the raster is a generic object.

Author(s)

Emanuele Cordano

See Also

`raster`,`Raster-class`

Examples

```
showClass("RasterList")
```

RasterList-Package *RasterList: A package for Rasters Where Cells are Generic Objects*

Description

The aim of this package is to develop a way to make some complex operations on each cells of a Raster Maps. Generally raster contains numeric values in each cells and in each band. Sometimes complex operation required the definition of particular object, in case such operation should be executed for each cell of a raster map, it becomes challanging. Then `RasterList-package` makes these operations easy to be implented with few lines of codes.

Details

The `RasterList-class` is a S4 class that inherits the `RasterLayer-class` and it an added slot called `list`. The `list` slot is a `list` object of so many elements how many are the cells of the inherited `RasterLayer-class` class, so that there is bijective corrensponce between a raster cell and a list element. The RasterList package provides three categories of important functions:

`rasterList`: it is the constructor of a `RasterList-class` object from a generic `object`, it also transforms a `RasterList-class` into another one throungh a function argument.

`rasterListFun`: it is a function that constructs a particular `RasterList-class` object in which the objects are `function`-type.

`RasterListApply`: it is a function that allows to operate among two or more `RasterList-class` objects defined in the same spatial extent through a generic function `fun(x,y,...)` where `x` and `y` are each cell/element of two `RasterList-class` objects given as arguments `x` and `y` of `RasterListApply`.

Some examples, concerning the fitting of a probability function for each cell of a stack (`RasterStack-class` object) of precipitation time-series or the estimation of soil water retention curve for each cell of a raster map, are shown throughout the package manual. The precipitation example dataset in raster format were extracted by the CHIRPS database:

Funk, Chris, Pete Peterson, Martin Landsfeld, Diego Pedreros, James Verdin, Shraddhanand Shukla, Gregory Husak, James Rowland, Laura Harrison, Andrew Hoell and Joel Michaelsen. "The climate hazards infrared precipitation with stations - a new environmental record for monitoring extremes". Scientific Data 2, 150066. doi:10.1038/sdata.2015.66 2015 , https://chc.ucsb.edu/data/chirps.

The package-provided datasets shall be only used as example datasets.

The development of this package has been sponsored by ACEWATER2 and "Water for Growth and Poverty Reduction in the Mekrou" projects of the Joint Research Centre of the Europan Commission (http://aquaknow.jrc.ec.europa.eu).

RasterListApply — *A function for operations among RasterList-class objects.*

Description

A function for operations among RasterList-class objects.

Usage

```
RasterListApply(..., FUN = NULL)
```

Arguments

`...`	a set of arguments containg the RasterList-class objects whose lists are operated by FUN.
`FUN`	a function

Value

a RasterList-class object

See Also

mapply,rasterList

Examples

```
f <- system.file("external/test.grd", package="raster")

ra <- rasterList(f)
rb <- rasterList(f)
```

```
rm <- RasterListApply(x=ra,y=rb,z=10,FUN=function(x,y,z){x+y+z})

### Fitting a probability distribution for precipitation
### in each cell with "lmon" package (L Moments)
library(lmom)


# The package-provided datasets shall be only used as example datasets.
precf <- system.file("map/Mekrou_precipitation.grd", package="rasterList") ##
## A resampled preciptation raster map based on CHIRS dataset:
## Funk, Chris, Pete Peterson, Martin Landsfeld, Diego Pedreros, James Verdin,
## Shraddhanand Shukla, Gregory Husak, James Rowland, Laura Harrison,
## Andrew Hoell and Joel Michaelsen.
## "The climate hazards infrared precipitation with stations -a new environmental
## record for monitoring extremes". Scientific Data 2, 150066. doi:10.1038/sdata.2015.66 2015.
## http://chg.geog.ucsb.edu/data/chirps/
##
prec <- stack(precf)
## Sample L-moments
samlmom <- stack(rasterList(prec,FUN=samlmu))
## Fitting a Random Probability Distribution: it is a 'rasterList' Object
fitdist <- rasterList(samlmom,FUN=pelgam)

## KS TESTING

kstesting <- RasterListApply(x=rasterList(prec),y="cdfgam",para=fitdist,FUN=ks.test)

## Mapping of p-value
pval_ks <- raster(kstesting,FUN=function(x){x$p.value})
```

rasterListFun *Execution of the elements of a* RasterList

Description

This fuction transmors a generic RasterList-class object into another RasterList-class object where elemets are all function-type.

Usage

```
rasterListFun(object)
```

Arguments

object an object to be coerced to RasterList-class

Value

This function works with RasterList-class objects in which all elements of object@list slot are functions. It returns a "global" function that works at "raster" scale. The returned function will have the following usage signature: fun(xval,...) where one xval (if its lengths is different from 1) element is the applied to each element and ... are further common arguments.

Examples

```
library(sp)
library(rasterList)
library(soilwater)
set.seed(1234)
data(meuse.grid)
data(meuse)
coordinates(meuse.grid) <- ~x+y
coordinates(meuse) <- ~x+y
gridded(meuse.grid) <- TRUE


soilmap <- stack(meuse.grid)[['soil']]
elevmap <- rasterize(x=meuse,y=soilmap,field="elev",fun=mean)
soilparcsv <- system.file("external/soil_data.csv",package="soilwater")
soilpar <- read.table(soilparcsv,stringsAsFactors=FALSE,header=TRUE,sep=",")
## From help(meuse,help_type="html")
##soil type according to the 1:50 000 soil map of the Netherlands.
## 1 = Rd10A (Calcareous weakly-developed meadow soils, light sandy clay);
## 2 = Rd90C/VII (Non-calcareous weakly-developed meadow soils, heavy sandy clay to light clay);
## 3 = Bkd26/VII (Red Brick soil, fine-sandy, silty light clay)
soiltype_id <- c(1,2,3)
soiltype_name <- c("sandy clay","sandy clay","silty clay loam")

meuse.soilrasterlist <- rasterList(soilmap,FUN=function(i,soiltype_name,soilpar){

o <- NULL
if (!is.na(i)) {
ii <- which(soilpar$type==soiltype_name[i])
    o <- soilpar[ii,]
type <- o[["type"]]
o <- o[names(o)!="type"]
o <- o[names(o)!="Ks_m_per_hour"]
names(o)[names(o)=="Ks_m_per_sec"] <- "ks"
names(o)[names(o)=="swc"] <- "theta_sat"
names(o)[names(o)=="rwc"] <- "theta_res"
attr(o,"type") <- type
## add noise
noise <- rnorm(length(o))
o <- o*(1+0.005*noise)

o["m"] <- 1-1/o["n"]
} else {

o <- soilpar[which(soilpar$type==soiltype_name[1]),]
type <- o[["type"]]
o <- o[names(o)!="type"]
o <- o[names(o)!="Ks_m_per_hour"]
names(o)[names(o)=="Ks_m_per_sec"] <- "ks"
names(o)[names(o)=="swc"] <- "theta_sat"
names(o)[names(o)=="rwc"] <- "theta_res"
o[] <- NA
}

return(o)
},soiltype_name=soiltype_name,soilpar=soilpar)


meuse.swclist <- rasterList(meuse.soilrasterlist,FUN=function(x) {

o <- NA
## swc       rwc   alpha       n         m           ks
## 9 0.4295507 0.1093227 3.39387 1.39617 0.2837546 2.018317e-07


o <- function(psi,...,func="swc"){

args <- c(list(psi=psi,...),as.list(x))
oo <- do.call(args=args,what=get(func))
return(oo)

}







return(o)

})


### RasterList with soil water retenction curves (One for each cell!)

swcfunr <- rasterListFun(meuse.swclist)

## RasterLayer of soil water content assuming a uniformly distrrubted pressure head
psi <- -0.9
soil_water_content <- raster(swcfunr(psi))
plot(soil_water_content)

## RasterLayer of soil water content from  a generic map of soil water pressure head
psi <- 0.2-(elevmap-(5))
```

```
psi[] <- -0.9+0.1*rnorm(ncell(psi[])) ## Alternatively to the values of the previous line!
soil_water_content <- raster(swcfunr(psi))
plot(soil_water_content)

## END
```

stack,RasterList-method

Creates a RasterStack-class object from a RasterList-class

Description

The method transforms a RasterList-class into a RasterStack-class in case of the list elements are numeric vectors.

Usage

```
## S4 method for signature 'RasterList'
stack(x, ...)
```

Arguments

x	a rasterList-class object
...	further arguments for rasterList

Value

a RasterStack-class object

See Also

rasterList

Examples

```
f <- system.file("external/test.grd", package="raster")

## Creates a simple generic RasterList

rl <- rasterList(f)

list <- as.list(as.vector(rl))
list <- lapply(X=list,FUN=function (x) {c(x,x+10,x+15)})

rl <- rasterList(rl,list=list,object.name="test")
```

```
ss <- stack(rl)

il <- 8331
list[[il]] <- numeric(0)
rla <- rasterList(rl,list=list,object.name="test2")
sa <- stack(rla)
```

Index

∗ **classes**
    RasterList-class, 6

as.RasterList *(*rasterList*)*, 5

crop *(*crop,RasterList-method*)*, 2
crop,RasterList-method, 2

is.RasterList, 3

list, *7*

mapply, *8*

raster, *5*, *7*
raster *(*raster,RasterList-method*)*, 4
raster,RasterList-method, 4
rasterList, 5, *7*, *8*, *12*
rasterList,RasterBrick-method
        *(*rasterList*)*, 5
rasterList,RasterLayer-method
        *(*rasterList*)*, 5
rasterList,RasterList-method
        *(*rasterList*)*, 5
rasterList,RasterStack-method
        *(*rasterList*)*, 5
RasterList-class, *2–5*, 6, *8*, *12*
RasterList-Package, 7
RasterListApply, *4*, *7*, 8
rasterListFun, *7*, 9
RasterStack-class, *12*

stack, *4*
stack *(*stack,RasterList-method*)*, 12
stack,RasterList-method, 12