Package "raster" introduction

Hang Tian

2022-09-14

1. "Intro"

```
# install.packages('raster')
library(raster)
```

```
## Loading required package: sp
```

What does the package do?

Package description: "Reading, writing, manipulating, analyzing and modeling of spatial data. The package implements basic and high-level functions for raster data and for vector data operations such as intersections."

Basically there are two types of GIS data: vector & raster.

Vector, AKA shapefile, includes points, lines and polygons representing ground items while containing information in attribute table.

Raster, more like digital images nowadays with same scale pixels but gives more information like which coordinate reference system it's using, where its pixels locate.

The "raster" package, is designed for reading-processing-writing raster data in R.

2. "Author(s)"

Robert J. Hijmans et al. Maintainer: Robert J. Hijmans

Robert J. Hijmans, professor at the department of Environmental Science and Policy at University of California Davis, CA, US.

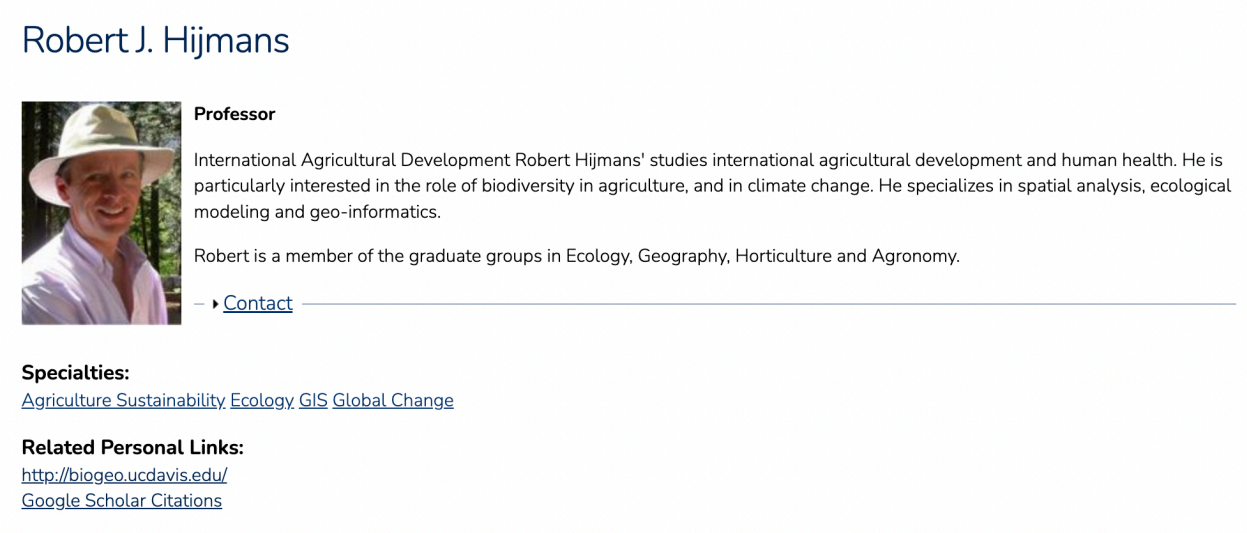

Figure 1: Robert profile

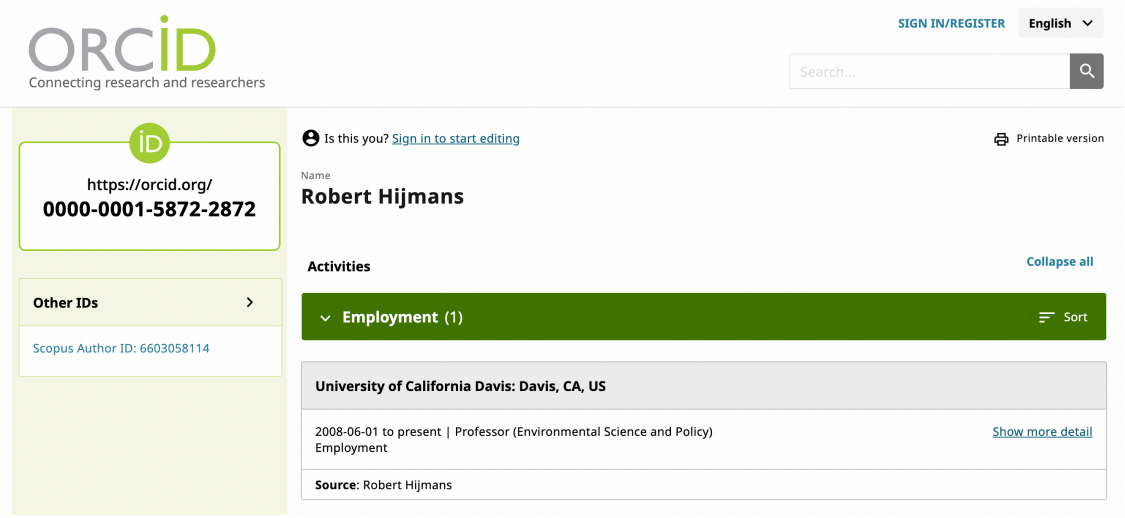

Figure 2: Robert link on ORCID

3. "Opening, plotting rater data"

```
# Read a raster layer
test1=raster('tmax_stack.tif',band=1)
# Read raster with multiple bands
test2=stack('tmax_stack.tif')
class(test1)
```

```
## [1] "RasterLayer"
## attr(,"package")
## [1] "raster"
```

```
class(test2)
```

```
## [1] "RasterStack"
## attr(,"package")
## [1] "raster"
```

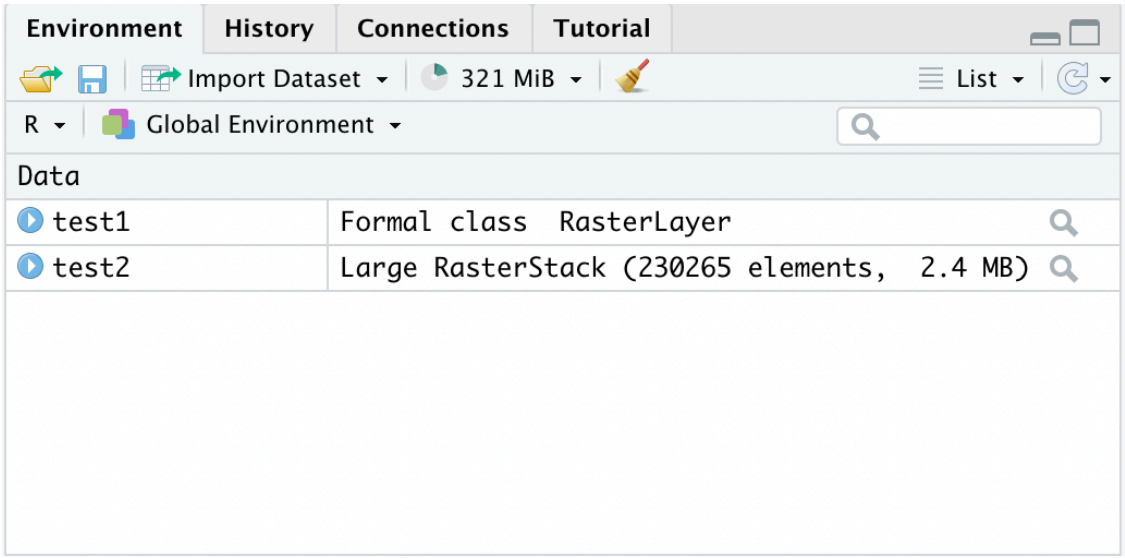

Figure 3: What you'll see in your Rsutdio workspace

```r
# Or if you have already saved your work space before
load("~/Desktop/WeatherRasters.RData")
# One trick here is before you save work space,
# remember to run sample_raster=readAll(sample_raster)
# to actually read it in memory
# Selecting the 1st band from raster stack
sample_raster=tmax[[1]]
all.equal(sample_raster,test1)
```

```
## [1] TRUE
```

```r
plot(sample_raster)
```

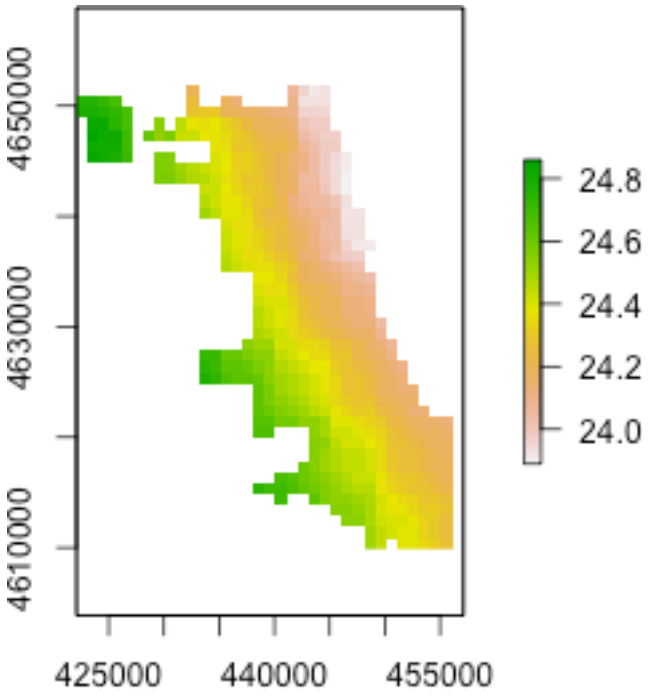

Figure 4: plot the raster

```r
library(sf,sp)
# See CRS (coordinate reference system) information of our data
# You can also use 'crs(sample_raster)' which is a function given by raste
st_crs(sample_raster)$proj4string
```

```
## [1] "+proj=utm +zone=16 +datum=NAD83 +units=m +no_defs"
```

```
extent(sample_raster)
```

```
## class       : Extent
## xmin        : 422097.8
## xmax        : 457097.8
## ymin        : 4609859
## ymax        : 4652859
```

4. "Time for some bold actions!"

Besides of simply reading and creating raster data in R, raster package gives many categories of functions helps you do raster processing:

a. Changing the spatial extent

```
sr_transposed=t(sample_raster)
plot(sr_transposed)
```

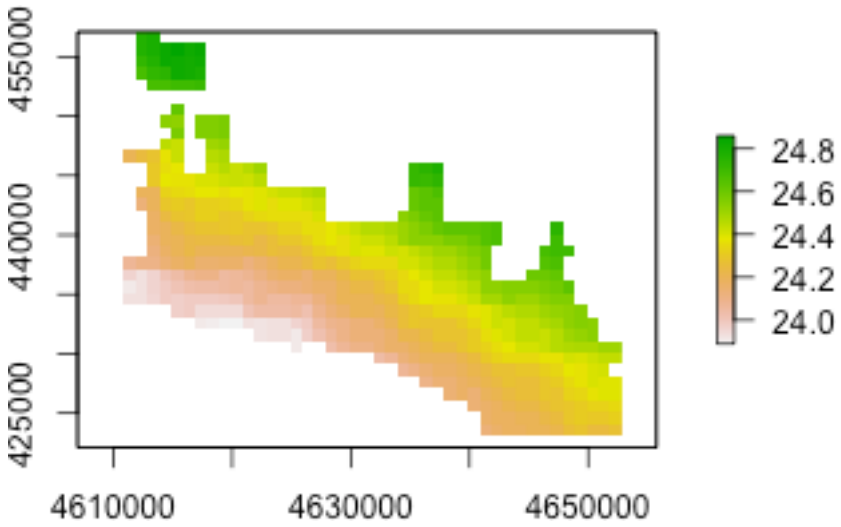

Figure 5: transposed raster

b. Raster algebra

```
sr_zero=sample_raster-sample_raster
plot(sr_zero)
```

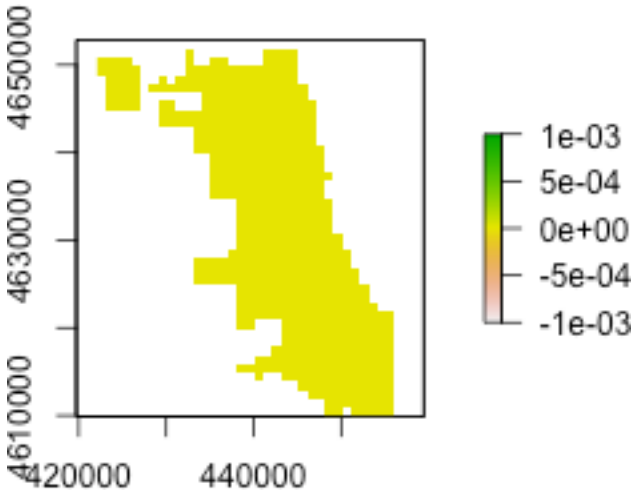

Figure 6: sample raster - sample raster

c. Cell based computation

```
sr_squared=calc(sample_raster,function(x)x^2-x)
plot(sr_squared)
```

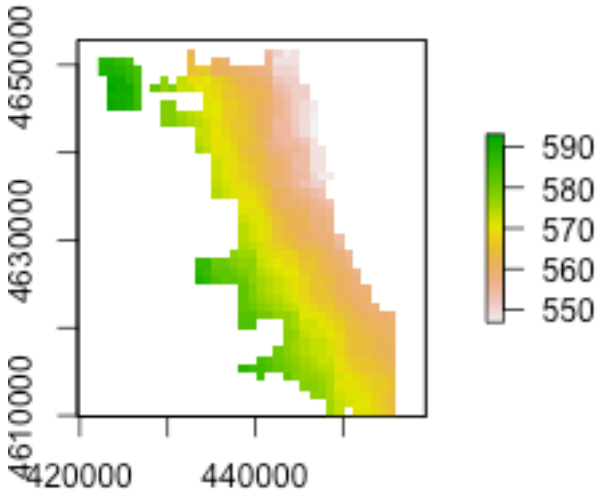

Figure 7: Squared value raster

And other functions:

d. Data type conversion

```
sr_polygon=rasterToPolygons(sample_raster,na.rm=TRUE)
plot(sr_polygon)
```

Figure 8: Raster to Polygon

e. Write your raster

```
writeRaster(sample_raster,filename='sample_raster.tif',
            format='GTiff',overwrite=TRUE)
```

And more...

Explore more about using this package at:

https://cran.r-project.org/web/packages/raster/raster.pdf

Besides, remember you could always use ?*function_name*() to view the help file.