ZOO955 - RasterBricks

Hilary Dugan

2/22/2018

Raster Bricks

Climate data

http://berkeleyearth.org/data/

Gridded data

Datasets are also provided in a gridded NetCDF format. Two types of grids are provided, a grid based on dividing the Earth into 15984 equal-area grid cells and a latitude-longitude grid. The equal area grid is the primary data format used in most of our analyses and provides generally smaller files; however, that format may be less convenient for many users.

Will use: * Global surface temperatures: BEST: Berkeley Earth Surface Temperatures * Monthly Land Average Temperature (TAVG; 1753 - Recent) (200 Mb) * Temperature: A surface air temperature anomaly field.

Read data

In this example we're using a netcdf file (.nc). These can store multiple variables. If you read in data, you may get a warning like `"Warning message:In .varName(nc, varname, warn = warn) : varname used is: temperature If that is not correct, you can set it to one of: temperature, climatology"`

Luckily, it will tell you the variabiles present! In this case, there in only one variable (temperature)

```
library(raster,verbose = F)
r = raster('Data/WI_TAVG_LatLong1.nc')
r
```

```
## class       : RasterLayer
## band        : 1  (of  3200  bands)
## dimensions  : 5, 6, 30  (nrow, ncol, ncell)
## resolution  : 1, 1  (x, y)
## extent      : -93, -87, 42, 47  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=longlat +datum=WGS84 +ellps=WGS84 +towgs84=0,0,0
## data source : /Users/hilarydugan/Documents/Rpackages/Zoo955/Lecture5_RasterBricks/Data/WI_TAVG_LatLo
## names       : time
## z-value     : 1750.04166666667
## zvar        : time
```

Note the multiple bands `band: 1 (of  3200  bands)`

Can choose band.

```
r = raster('Data/WI_TAVG_LatLong1.nc', band = 10)
r
```

```
## class       : RasterLayer
## band        : 10  (of  3200  bands)
## dimensions  : 5, 6, 30  (nrow, ncol, ncell)
```

```
## resolution  : 1, 1  (x, y)
## extent      : -93, -87, 42, 47  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=longlat +datum=WGS84 +ellps=WGS84 +towgs84=0,0,0
## data source : /Users/hilarydugan/Documents/Rpackages/Zoo955/Lecture5_RasterBricks/Data/WI_TAVG_LatLo
## names       : time
## z-value     : 1750.79166666667
## zvar        : time
```

Or, can import all the bands, using `brick`

```
br = brick('Data/WI_TAVG_LatLong1.nc')
br
```

```
## class       : RasterBrick
## dimensions  : 5, 6, 30, 3200  (nrow, ncol, ncell, nlayers)
## resolution  : 1, 1  (x, y)
## extent      : -93, -87, 42, 47  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=longlat +datum=WGS84 +ellps=WGS84 +towgs84=0,0,0
## data source : /Users/hilarydugan/Documents/Rpackages/Zoo955/Lecture5_RasterBricks/Data/WI_TAVG_LatLo
## names       : X1750.04166666667, X1750.125, X1750.20833333333, X1750.29166666667, X1750.375, X1750.45
## unknown     : 1750.04166666667, 2016.625 (min, max)
## varname     : time
```

You can see the class and dimensions have changed:

```
class       : RasterBrick dimensions  : 5, 6, 30, 3200  (nrow, ncol, ncell, nlayers)
```

Extracting data

We know that the raster brick has 3200 layers. We also know the min/max values are `1750.04166666667, 2016.625 (min, max)`

From this, and from the metadata, we know this data is monthly temperature. Can double check with

```
# Max year minus min year. Multiplied by 12 months
(2016.625-1750.041666667)*12
```

```
## [1] 3199
```

If we want to extract data, we can just like a list

```
fifth.month = br[[5]]
fifth.month # This is now a raster layer
```

```
## class       : RasterLayer
## band        : 5  (of  3200  bands)
## dimensions  : 5, 6, 30  (nrow, ncol, ncell)
## resolution  : 1, 1  (x, y)
## extent      : -93, -87, 42, 47  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=longlat +datum=WGS84 +ellps=WGS84 +towgs84=0,0,0
## data source : /Users/hilarydugan/Documents/Rpackages/Zoo955/Lecture5_RasterBricks/Data/WI_TAVG_LatLo
## names       : X1750.375
## z-value     : 1750.375
## zvar        : time
```

Extract data

```
vals = getValues(fifth.month)
vals
```

```
##  [1]         NA -0.2809994 -0.2762341         NA         NA         NA
##  [7] -0.2791373 -0.2635677 -0.2566674 -0.2551369 -0.2395773         NA
## [13]         NA -0.2377647 -0.2301865 -0.2239810 -0.2017416 -0.1726130
## [19]         NA         NA -0.2128637 -0.2070424 -0.1914228         NA
## [25]         NA         NA         NA         NA -0.1805877         NA
```

But where are these values located in space? Can use the function `xyFromCell`.

These functions get coordinates of the center of raster cells for a row, column, or cell number of a Raster object.*

We know there are 30 cells, but can also use the function `ncell`

```
coord <- xyFromCell(fifth.month,1:ncell(fifth.month))
coord
```

```
##           x    y
##  [1,] -92.5 46.5
##  [2,] -91.5 46.5
##  [3,] -90.5 46.5
##  [4,] -89.5 46.5
##  [5,] -88.5 46.5
##  [6,] -87.5 46.5
##  [7,] -92.5 45.5
##  [8,] -91.5 45.5
##  [9,] -90.5 45.5
## [10,] -89.5 45.5
## [11,] -88.5 45.5
## [12,] -87.5 45.5
## [13,] -92.5 44.5
## [14,] -91.5 44.5
## [15,] -90.5 44.5
## [16,] -89.5 44.5
## [17,] -88.5 44.5
## [18,] -87.5 44.5
## [19,] -92.5 43.5
## [20,] -91.5 43.5
## [21,] -90.5 43.5
## [22,] -89.5 43.5
## [23,] -88.5 43.5
## [24,] -87.5 43.5
## [25,] -92.5 42.5
## [26,] -91.5 42.5
## [27,] -90.5 42.5
## [28,] -89.5 42.5
## [29,] -88.5 42.5
## [30,] -87.5 42.5
```

Can see that it's a 1 deg resolution. But we already know that too.

Which cell is closest to Lake Mendota?

```
lat = 43.1
long = -89.4
indx = which(abs(coord[,1] - long) <= 0.5 & abs(coord[,2] - lat) <= 0.5)
indx
```

```
## [1] 22
```

Cell 22 is closest to Lake Menota. `[22,] -89.5 43.5`

I'm sure there are spatial ways to do this as well. Using functions like `gDistance` in the `rgeos` package

Extracting data

Can extract data from a raster using the `extract` function

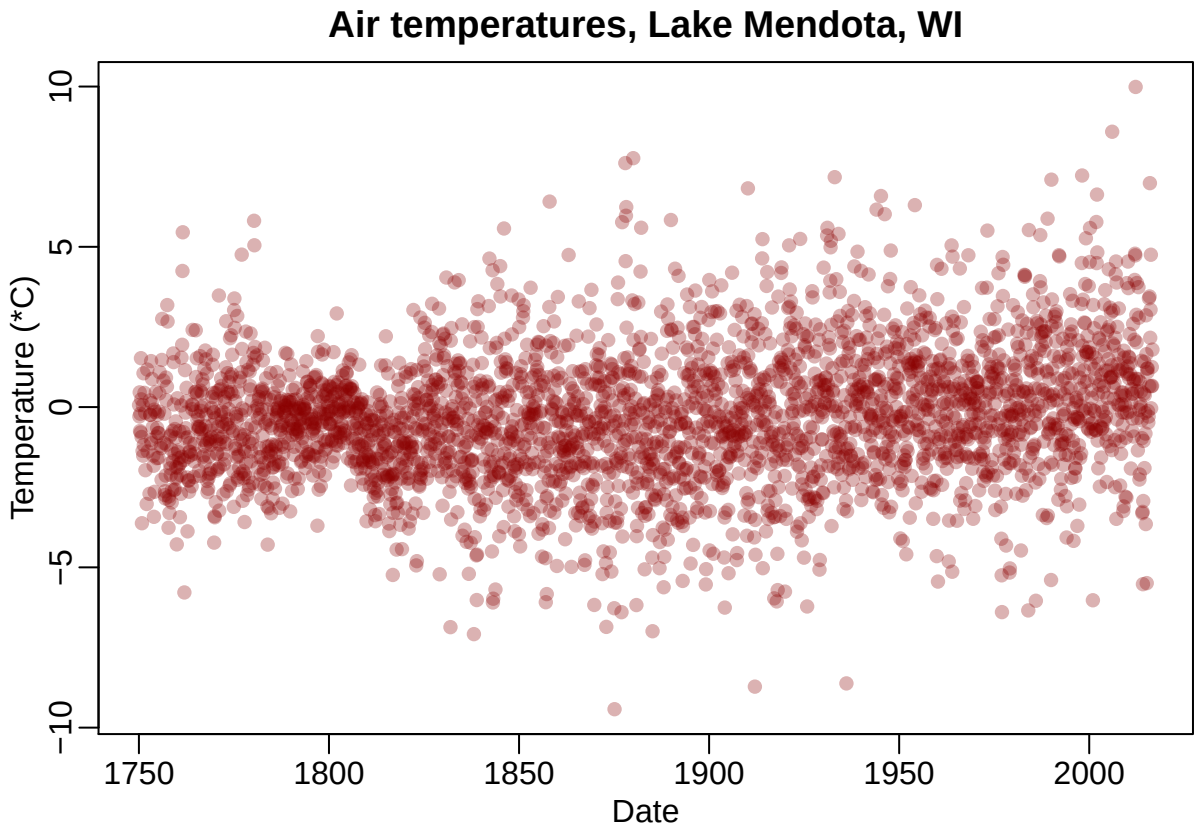
- x = Raster* object
- y = points represented by a two-column matrix or data.frame, or SpatialPoints; SpatialPolygons; SpatialLines; Extent; or a numeric vector representing cell numbers

We can extract data from Mendota, using either a `SpatialPoints` object, or a numeric vector (cell = 22)

```
# Option 1
mendota = data.frame(lat = 43.1, long = -89.4)
coordinates(mendota) = ~long+lat
menTemp = extract(br,y = mendota)

# Option 2
menTemp = extract(br,y = 22)
```

Kind of an annoying named numeric vector. Can make a nicer data.frame

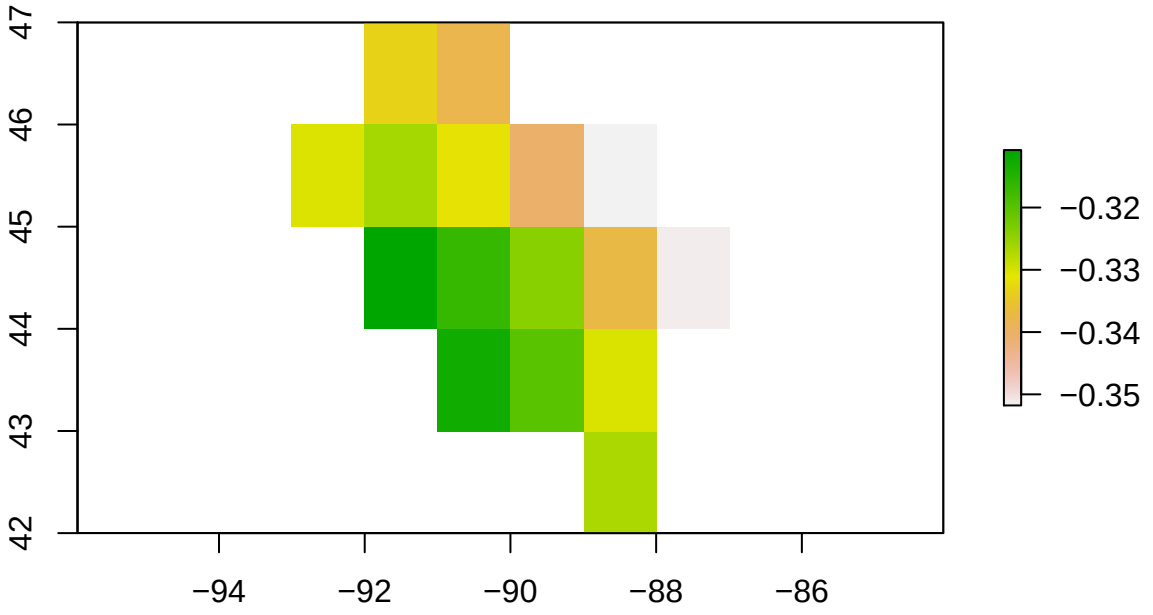
```
# Option 1
output = data.frame(Date = seq(1750.04166666667, 2016.625,length.out = 3200),Temp = menTemp[1,])

par(mar=c(3,3,2,1),mgp=c(1.5,0.5,0))
plot(output$Date,output$Temp,pch=16,col=adjustcolor('red4',0.3),xlab='Date',ylab='Temperature (*C)',
     main = 'Air temperatures, Lake Mendota, WI')
```



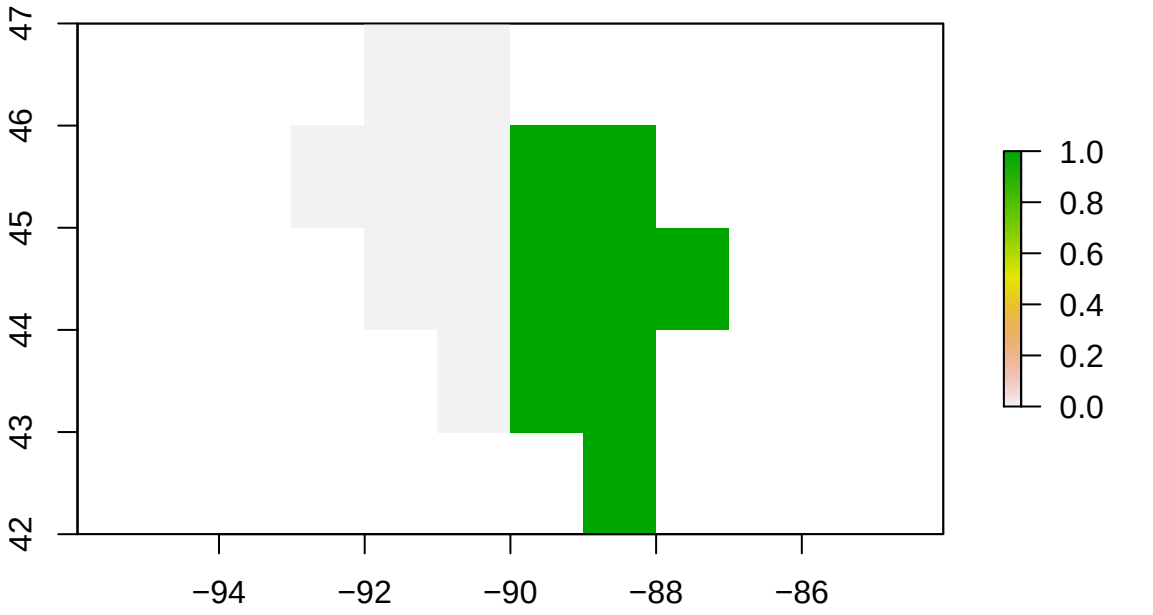
Raster math

You apply matrix math functions to rasters

```
# Get mean
meanTemp = mean(br,na.rm=T)
plot(meanTemp)
```




```
# Inequalities
gt10 = br[[200]] > 0.4
plot(gt10)
```



Homework

Use the 'Data/WI_CAVG_LatLong1.nc' file. In this lecture we dealt with the "temperature" variable. This file is the "climatology" variable.

Find the metadata for these dataset. http://berkeleyearth.org/data/. We are using *Gridded Data. Monthly Land. Average Temperature (TAVG; 1753 – Recent).*

1) The climatology variable is a monthly average for each cell. What years does this average represent?
2) Plot the August averages for Wisconsin.
3) Extract the averages for the cell over Lake Mendota. Plot the monthly temperature averages.
4) What is the August average for Lake Mendota?