

Zoo 955 - Introduction to Zoo 955

Hilary Dugan

January 23, 2018

955 basics

- Class notes: <https://github.com/hdugan/Zoo955>
- Class ZOOM: <https://zoom.us/j/726741289>

955 basics

- R version ≥ 3.4
- Recommend RStudio as a code editor
 - This is what I will use in class
 - RStudio Desktop 1.1.414
- Your primary reference will probably be community answers on Stack Overflow (<https://stackoverflow.com>)
- Recommend creating a username and upvoting answers you find helpful

955 Assessment

- Grades will be based on:
 - short assignments (25%)
 - participating in class discussion (50%)
 - final project based on individual research topics that will be presented during the final two weeks of the semester (25%)
- Numerical grades are assigned as follows: 93-100 (A), 88-92 (AB), 82-87 (B), 78-81 (BC).
- There will be short weekly assignments
- Typically I'll ask that you email me your assignments as a pdf
 - R markdown tutorial
 - <https://ismayc.github.io/rbasics-book/4-rmarkdown.html>
 - Or however you'd like to make a pdf
- There's no class Feb 13th. Instead, short proposals of your intended research endeavour will be due.

The spatial basics

- Commonly used R packages
 - `library(rgdal)`
 - `library(rgeos)`
 - `library(sp)`
 - `library(maps)`
 - `library(raster)`

– library(maptools)

Why use R for spatial analysis?

Pros:

- Breadth of statistical packages
- Integration with other data/stats
- Free, easy to access

Cons:

- Memory limitations

Spatial Objects

	Without attributes	With attributes (data)
Points	SpatialPoints	SpatialPointsDataFrame
Lines	SpatialLines	SpatialLinesDataFrame
Polygons	SpatialPolygons	SpatialPolygonsDataFrame
Raster	SpatialGrid	SpatialGridDataFrame
Raster	SpatialPixels	SpatialPixelsDataFrame

Spatial objects

- Spatial class has two slots
- 1) bounding box, dimension of the plot
- 2) CRS class object defining the coordinate reference system, and may be set to CRS(as.character(NA)), its default value.
- Operations on **Spatial** objects should update or copy these values to the new Spatial objects being created

SpatialPoints

- The most basic spatial data object is a point, which may have 2 or 3 dimensions
- A single coordinate, or a set of such coordinates, may be used to define a SpatialPoints object
- The points in a SpatialPoints object may be associated with a row of attributes to create a SpatialPointsDataFrame object

Example: SpatialPoints

```
library(sp)
spoint <- data.frame(y = 43.07, x = -89.40)
summary(spoint)
```

```
##           y           x
## Min.    :43.07   Min.    :-89.4
## 1st Qu.:43.07   1st Qu.: -89.4
## Median :43.07   Median  :-89.4
## Mean    :43.07   Mean    :-89.4
## 3rd Qu.:43.07   3rd Qu.: -89.4
## Max.    :43.07   Max.    :-89.4
```

```
str(spoint)
```

```
## 'data.frame':    1 obs. of  2 variables:
## $ y: num 43.1
## $ x: num -89.4
```

```
coords <- SpatialPoints(spoint)
str(coords)
```

```
## Formal class 'SpatialPoints' [package "sp"] with 3 slots
## ..@ coords      : num [1, 1:2] 43.1 -89.4
## .. ..- attr(*, "dimnames")=List of 2
## .. .. ..$ : NULL
## .. .. ..$ : chr [1:2] "y" "x"
## ..@ bbox        : num [1:2, 1:2] 43.1 -89.4 43.1 -89.4
## .. ..- attr(*, "dimnames")=List of 2
## .. .. ..$ : chr [1:2] "y" "x"
## .. .. ..$ : chr [1:2] "min" "max"
## ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot
## .. .. ..@ projargs: chr NA
```

```
summary(coords)
```

```
## Object of class SpatialPoints
## Coordinates:
##      min      max
## y  43.07  43.07
## x -89.40 -89.40
## Is projected: NA
## proj4string : [NA]
## Number of points: 1
```

SpatialPointsDataFrame

Add data to make a SpatialPointsDataFrame object.

```
# library(sp)
sdata <- SpatialPointsDataFrame(coords, data = data.frame(population = 327000))
str(sdata)
```

```
## Formal class 'SpatialPointsDataFrame' [package "sp"] with 5 slots
## ..@ data         :'data.frame':  1 obs. of  1 variable:
## .. ..$ population: num 327000
## ..@ coords.nrs   : num(0)
## ..@ coords       : num [1, 1:2] 43.1 -89.4
## .. ..- attr(*, "dimnames")=List of 2
## .. .. ..$ : NULL
## .. .. ..$ : chr [1:2] "y" "x"
```

```
## ..@ bbox      : num [1:2, 1:2] 43.1 -89.4 43.1 -89.4
## .. ..- attr(*, "dimnames")=List of 2
## .. .. ..$ : chr [1:2] "y" "x"
## .. .. ..$ : chr [1:2] "min" "max"
## ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot
## .. .. ..@ projargs: chr NA
```

```
summary(sdata)
```

```
## Object of class SpatialPointsDataFrame
## Coordinates:
##      min      max
## y  43.07  43.07
## x -89.40 -89.40
## Is projected: NA
## proj4string : [NA]
## Number of points: 1
## Data attributes:
##      population
##      Min.      :327000
##      1st Qu.:327000
##      Median :327000
##      Mean   :327000
##      3rd Qu.:327000
##      Max.   :327000
```

Spatial Points + Data Diagram

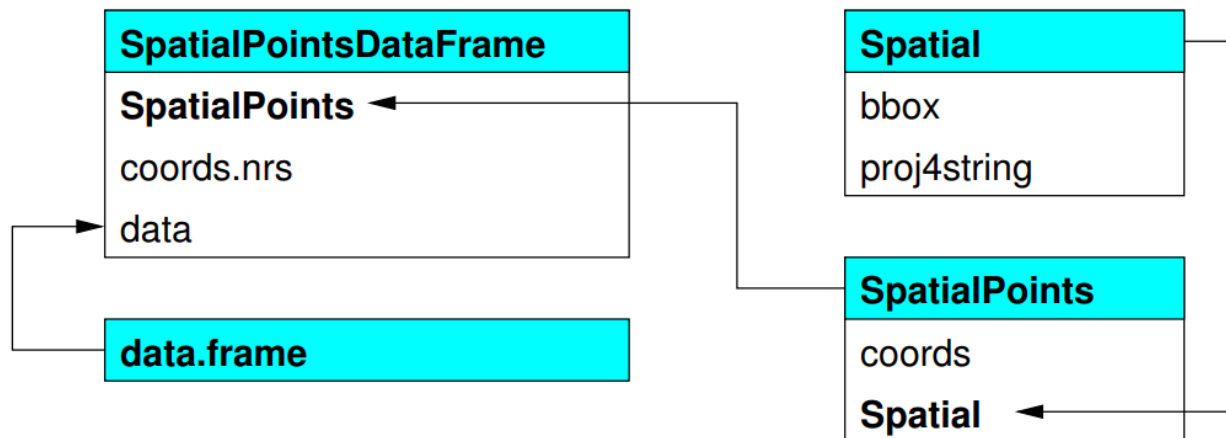


Figure 1: Figure credit: Roger S. Bivand, Edzer J. Pebesma and Virgilio Gómez-Rubio. *Applied Spatial Data Analysis with R* Springer. August 2008.

Spatial Lines & Polygons

- A Line object is a collection of 2D coordinates

- A Polygon object is a Line object with equal first and last coordinates
- A Lines object is a list of Line objects, such as all the contours at a single elevation; the same relationship holds between a Polygons object and a list of Polygon objects, such as islands belonging to the same county
- SpatialLines and SpatialPolygons objects are made using lists of Lines or Polygons objects respectively
- SpatialLinesDataFrame and SpatialPolygonsDataFrame objects are defined using SpatialLines and SpatialPolygons objects and standard data frames, and the ID fields are here required to match the data frame row names

Example: Spatial Polygons

```
sdata <- data.frame(y = c(43.07,43.10,43.08,43.07),
                    x = c(-89.40,-89.41,-89.5,89.40))
str(sdata)
```

```
## 'data.frame':    4 obs. of  2 variables:
## $ y: num  43.1 43.1 43.1 43.1
## $ x: num  -89.4 -89.4 -89.5 89.4
```

```
spoly <- Polygons(list(Polygon(sdata)), ID = "madison")
summary(spoly)
```

```
##   Length   Class   Mode
##      1 Polygons    S4
```

```
spolys <- SpatialPolygons(list(spoly))
summary(spolys)
```

```
## Object of class SpatialPolygons
## Coordinates:
##      min max
## x  43.07 43.1
## y -89.50 89.4
## Is projected: NA
## proj4string : [NA]
```

Spatial Polygons Diagram

Spatial grids and pixels

- There are two representations for data on regular rectangular grids (oriented N-S, E-W): **SpatialPixels** and **SpatialGrid**
- SpatialPixels are like SpatialPoints objects, but the coordinates have to be regularly spaced; the coordinates are stored, as are grid indices
- SpatialPixelsDataFrame objects only store attribute data where it is present, but need to store the coordinates and grid indices of those grid cells
- SpatialGridDataFrame objects do not need to store coordinates, because they fill the entire defined grid, but they need to store NA values where attribute values are missing

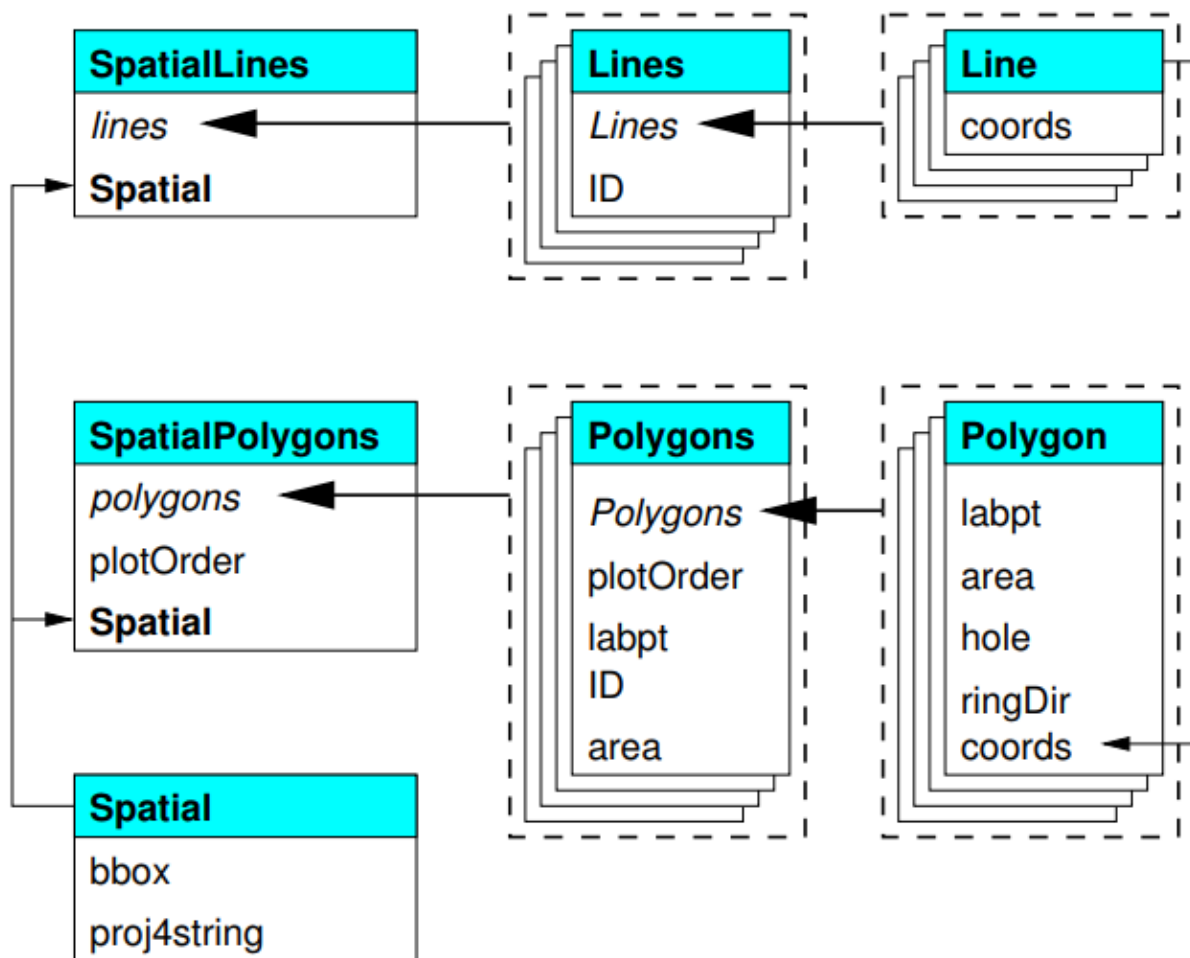


Figure 2: Figure credit: Roger S. Bivand, Edzer J. Pebesma and Virgilio Gómez-Rubio. *Applied Spatial Data Analysis with R* Springer. August 2008.

Spatial Grids Diagram

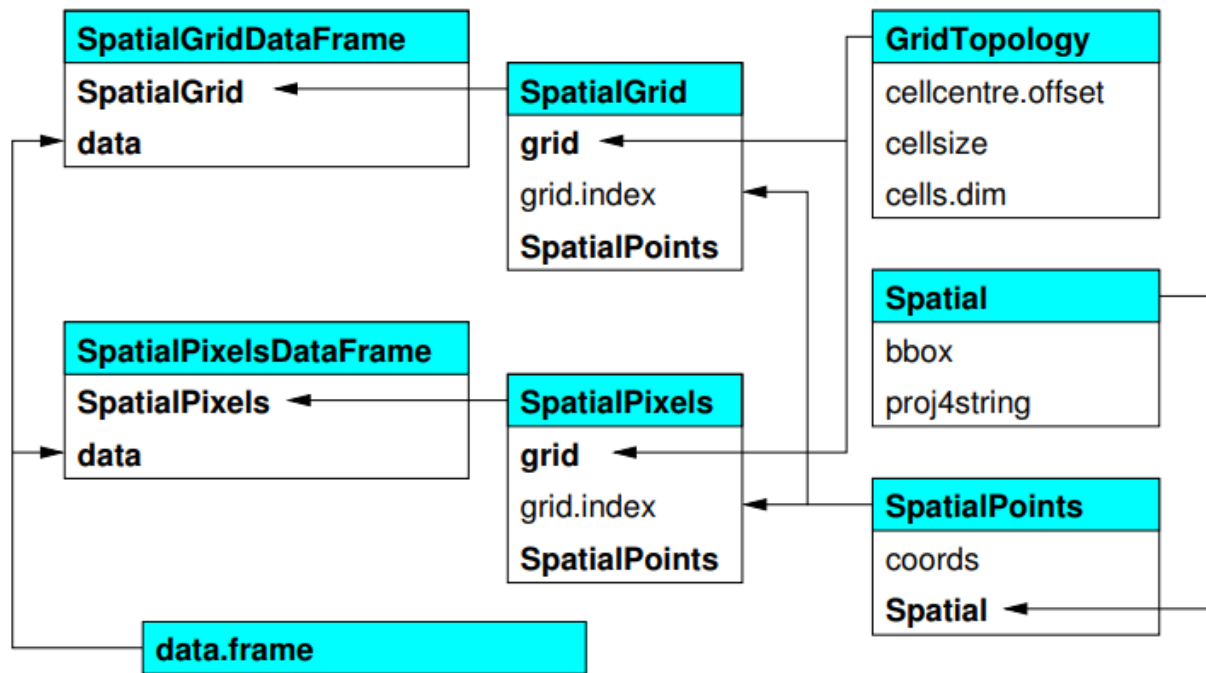


Figure 3: Figure credit: Roger S. Bivand, Edzer J. Pebesma and Virgilio Gómez-Rubio. *Applied Spatial Data Analysis with R* Springer. August 2008.

Raster Package

Often, instead of using `SpatialGrid`, your preferred object format will be a `RasterLayer`

Get to know the Raster package

- Description: Reading, writing, manipulating, analyzing and modeling of gridded spatial data. The package implements basic and high-level functions. Processing of very large files is supported

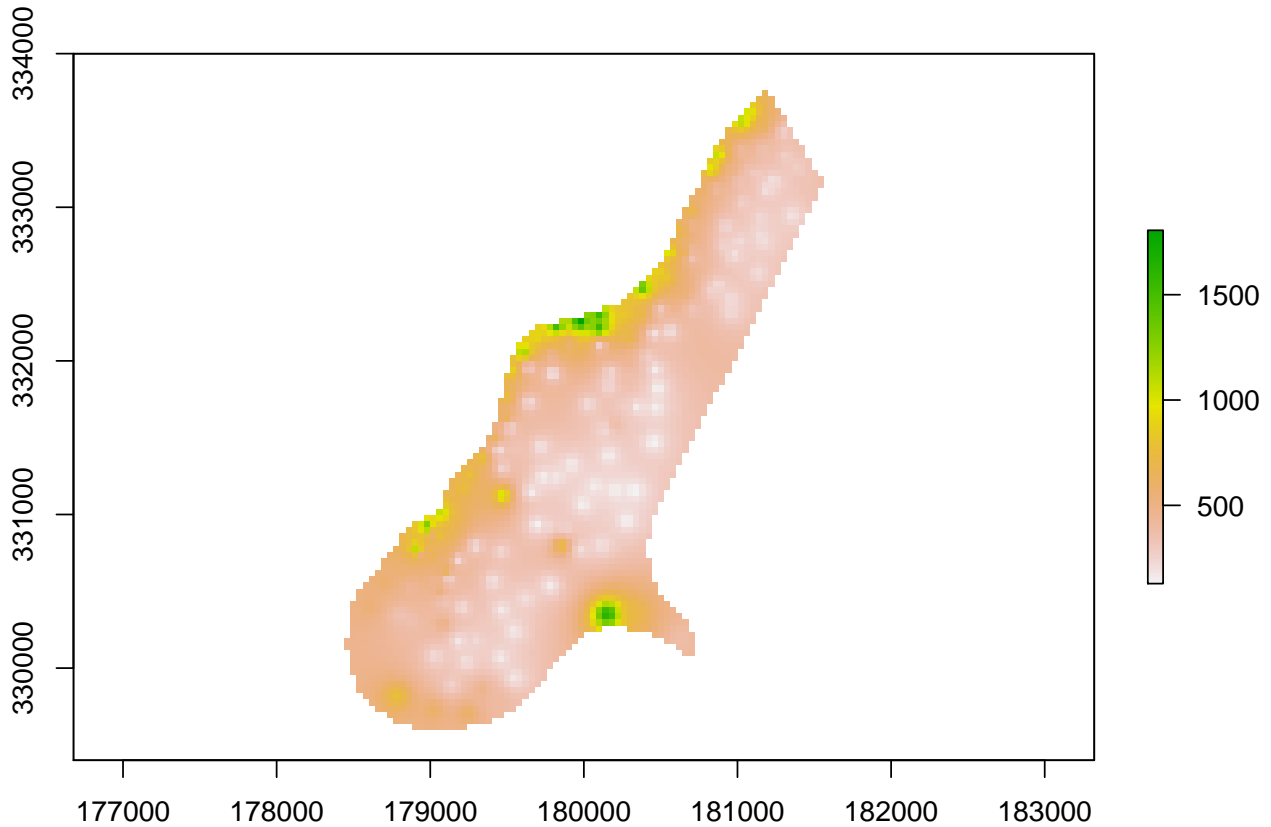
```

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

## class      : RasterLayer
## dimensions : 115, 80, 9200  (nrow, ncol, ncell)
## resolution : 40, 40  (x, y)
## extent     : 178400, 181600, 329400, 334000  (xmin, xmax, ymin, ymax)
## coord. ref.: +init=epsg:28992 +towgs84=565.237,50.0087,465.658,-0.406857,0.350733,-1.87035,4.0812 +
## data source : /Library/Frameworks/R.framework/Versions/3.3/Resources/library/raster/external/test.gr
## names      : test
## values     : 128.434, 1805.78  (min, max)

```

```
plot(r)
```



Can also have `RasterBrick` objects. Basically multi-layer `RasterLayer` objects. We'll see more of these later in the semester.

```
b <- brick(system.file("external/rlogo.grd", package="raster"))
```

```
b
```

```
## class      : RasterBrick
## dimensions  : 77, 101, 7777, 3  (nrow, ncol, ncell, nlayers)
## resolution  : 1, 1  (x, y)
## extent     : 0, 101, 0, 77  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=merc +datum=WGS84 +ellps=WGS84 +towgs84=0,0,0
## data source : /Library/Frameworks/R.framework/Versions/3.3/Resources/library/raster/external/rlogo.g
## names       : red, green, blue
## min values  : 0, 0, 0
## max values  : 255, 255, 255
```

Homework

Make an r markdown pdf. It can be anything you want. Send it to me.