Spatial Data Science with R

Edzer Pebesma and Roger Bivand

2018-03-14

Contents

0.1 Preface

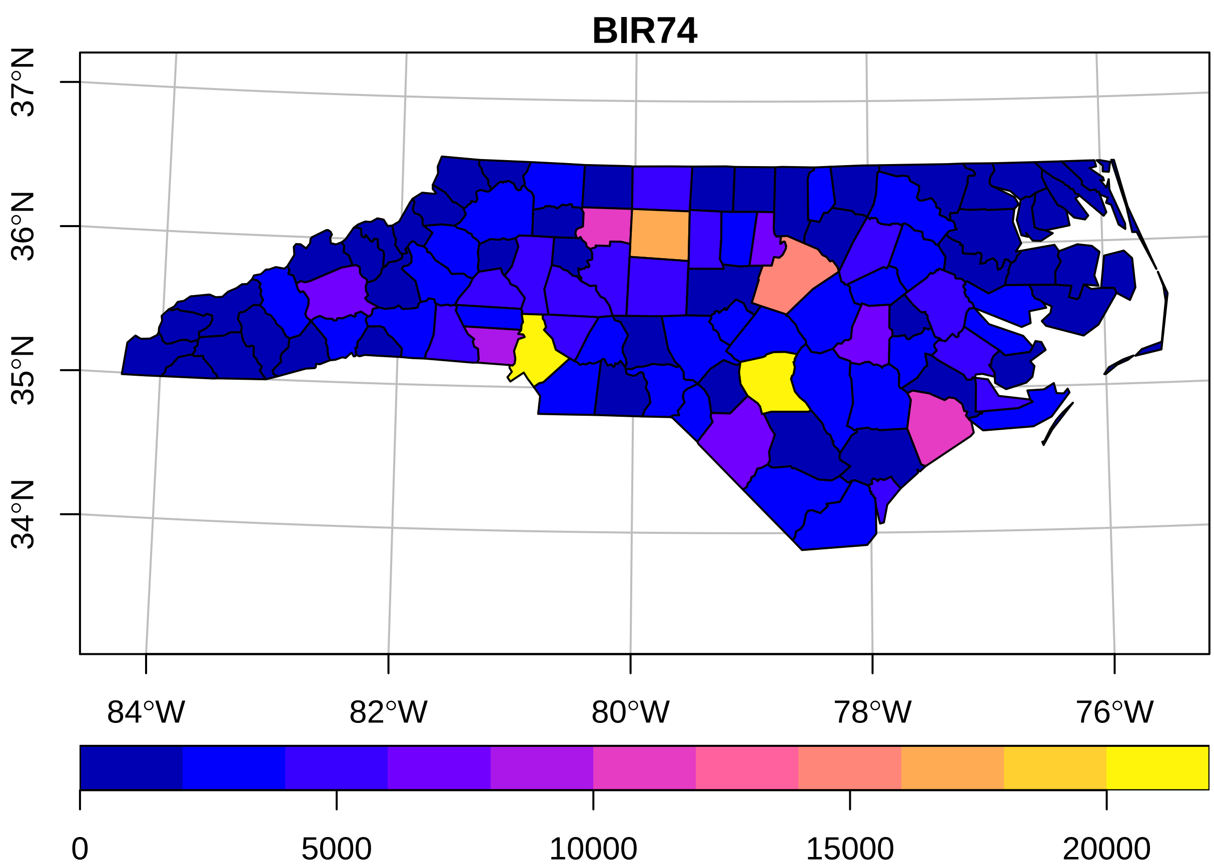
Part I

Spatial Data

0.2 Hello, world!

Let's create a map:

```
library(tidyverse)
#> -- Attaching packages --------------------------------- tidyverse 1.2.1 --
#> ✓ ggplot2 2.2.1.9000     ✓ purrr   0.2.4
#> ✓ tibble  1.4.2          ✓ dplyr   0.7.4
#> ✓ tidyr   0.8.0          ✓ stringr 1.3.0
#> ✓ readr   1.1.1          ✓ forcats 0.2.0
#> -- Conflicts ------------------------------------ tidyverse_conflicts() --
#> x dplyr::filter() masks stats::filter()
#> x dplyr::lag()    masks stats::lag()
library(sf)
#> Loading required package: methods
#> Linking to GEOS 3.5.1, GDAL 2.2.1, proj.4 4.9.3
system.file("gpkg/nc.gpkg", package="sf") %>%
    read_sf %>%
    st_transform(32119) %>%
    select(BIR74) %>%
    plot(graticule = TRUE, axes = TRUE)
```



A lot went on, here. We will describe the steps in detail. First, we loaded two R packages:

```r
library(tidyverse)
library(sf)
```

where `tidyverse` is needed for the tidyverse functions and methods, and `sf` is needed for the spatial commands and spatial tidyverse methods. The `%>%` (pipe) symbols should be read as *then*:

```r
a %>% b %>% c %>% d(n = 10)
```

is simply another way of writing

```r
d(c(b(a)), n = 10)
```

or alternatively

```r
tmp1 <- b(a)
tmp2 <- c(tmp1)
tmp3 <- d(tmp2, n = 10)
```

but is easier to read, because we don't have to go from right to left, and we don't have to choose names for intermediate results.

For the illustration we picked a data file that comes with `sf`, the location of which depends on your operating system:

```r
(file <- system.file("gpkg/nc.gpkg", package="sf"))
#> [1] "/home/edzer/R/x86_64-pc-linux-gnu-library/3.4/sf/gpkg/nc.gpkg"
```

Parens around this expression are used to have the result not only stored, but also printed.

Then, we read this file into R using `read_sf`:

```r
(nc <- read_sf(file))
#> Simple feature collection with 100 features and 14 fields
#> geometry type:  MULTIPOLYGON
#> dimension:      XY
#> bbox:           xmin: -84.3 ymin: 33.9 xmax: -75.5 ymax: 36.6
#> epsg (SRID):    4267
#> proj4string:    +proj=longlat +datum=NAD27 +no_defs
#> # A tibble: 100 x 15
#>     AREA PERIMETER CNTY_ CNTY_ID NAME    FIPS  FIPSNO CRESS_ID BIR74 SID74
#>    <dbl>     <dbl> <dbl>   <dbl> <chr>   <chr>  <dbl>    <int> <dbl> <dbl>
#> 1 0.114       1.44  1825    1825 Ashe    37009  37009        5  1091  1.00
#> 2 0.0610      1.23  1827    1827 Allegh~ 37005  37005        3   487  0
#> 3 0.143       1.63  1828    1828 Surry   37171  37171       86  3188  5.00
#> 4 0.0700      2.97  1831    1831 Currit~ 37053  37053       27   508  1.00
#> 5 0.153       2.21  1832    1832 Northa~ 37131  37131       66  1421  9.00
#> 6 0.0970      1.67  1833    1833 Hertfo~ 37091  37091       46  1452  7.00
#> # ... with 94 more rows, and 5 more variables: NWBIR74 <dbl>, BIR79 <dbl>,
#> #   SID79 <dbl>, NWBIR79 <dbl>, geom <MULTIPOLYGON [°]>
```

which creates a "spatial tibble":

```r
class(nc)
#> [1] "sf"         "tbl_df"     "tbl"        "data.frame"
```

This object is transformed into a new coordinate reference system (North Carolina State Plane, with EPSG code 32119):

```
(nc.32119 <- st_transform(nc, 32119))
#> Simple feature collection with 100 features and 14 fields
#> geometry type:  MULTIPOLYGON
#> dimension:      XY
#> bbox:           xmin: 124000 ymin: 14700 xmax: 931000 ymax: 318000
#> epsg (SRID):    32119
#> proj4string:    +proj=lcc +lat_1=36.16666666666666 +lat_2=34.33333333333334 +lat_0=33.75 +lon_0=-79 +x_
#> # A tibble: 100 x 15
#>     AREA PERIMETER CNTY_ CNTY_ID NAME    FIPS  FIPSNO CRESS_ID BIR74 SID74
#>    <dbl>     <dbl> <dbl>   <dbl> <chr>   <chr>  <dbl>    <int> <dbl> <dbl>
#> 1 0.114       1.44  1825    1825 Ashe    37009  37009        5  1091  1.00
#> 2 0.0610      1.23  1827    1827 Allegh~ 37005  37005        3   487  0
#> 3 0.143       1.63  1828    1828 Surry   37171  37171       86  3188  5.00
#> 4 0.0700      2.97  1831    1831 Currit~ 37053  37053       27   508  1.00
#> 5 0.153       2.21  1832    1832 Northa~ 37131  37131       66  1421  9.00
#> 6 0.0970      1.67  1833    1833 Hertfo~ 37091  37091       46  1452  7.00
#> # ... with 94 more rows, and 5 more variables: NWBIR74 <dbl>, BIR79 <dbl>,
#> #   SID79 <dbl>, NWBIR79 <dbl>, geom <MULTIPOLYGON [m]>
```

and a single attribute column is selected

```
(nc.32119.bir74 <- select(nc.32119, BIR74))
#> Simple feature collection with 100 features and 1 field
#> geometry type:  MULTIPOLYGON
#> dimension:      XY
#> bbox:           xmin: 124000 ymin: 14700 xmax: 931000 ymax: 318000
#> epsg (SRID):    32119
#> proj4string:    +proj=lcc +lat_1=36.16666666666666 +lat_2=34.33333333333334 +lat_0=33.75 +lon_0=-79 +x_
#> # A tibble: 100 x 2
#>   BIR74                                                                geom
#>   <dbl>                                                  <MULTIPOLYGON [m]>
#> 1  1091 (((387345 278387, 381334 282774, 379438 282943, 373250 290553, 36~
#> 2   487 (((408602 292425, 408565 293985, 406643 296873, 406420 3e+05, 402~
#> 3  3188 (((478717 277490, 476936 278867, 471503 279173, 470806 281394, 46~
#> 4   508 (((878194 289128, 877381 291117, 875994 290881, 874941 292805, 87~
#> 5  1421 (((769835 277796, 768364 274842, 762616 274401, 763168 269009, 76~
#> 6  1452 (((812328 277876, 791158 277012, 789882 277579, 777724 277107, 76~
#> # ... with 94 more rows
```

Finally, the result is plotted, with the command:

```
plot(nc.32119.bir74, graticule = TRUE, axes = TRUE)
```

0.3 Spaces: 1, 2 and 3-dimensional, spherical, time, bounded spaces

discusses spaces

bounded: water, rivers, road networks

0.4 Geometries

vertices, edges, lines, polygons, simple features; raster cells

0.5 Vector/Raster

differences, correspondence; properties of rasters; arrays

0.6 Geometric Manipulations

discuss

0.7 Attributes

discuss

0.8 Reference Systems

Units of measure, reference systems, coordinate transformation and conversion

Part II

Maps

0.9 Plotting spatial data

Plotting of lines, symbols, polygons (choroplets; overlapping polygons), rasters

using color

0.10 Base Plot

0.11 ggplot2

`geom_sf` examples; useful annotations and manipulations

0.12 Interactive Maps

base plot: identify, locator

leaflet, tmap, mapview

mapedit?

Part III

Spatial Analysis

0.13 Summarizing Geometries

Properties: dimension, length, area, etc, if not earlier in Ch 3?

counts, density, intensity (units; meaningful)

0.14 Point Pattern Analysis

Basics PP, beyond counting; basic steps in PPA

sf - spatstat interface; rasters;

0.15 Manipulating attributes: summarise, aggregate, union, sample

0.16 Units of measure revisited: attribute units, intensive and extensive variables

0.17 Up- and Downscaling

sampling

largest sub-geometry

area-weighted interpolation

0.18 Spatial Interpolation and geostatistics

intro ; variograms; gstat (needs to be sf-ed)

0.19 Area Data and Spatial Correlation

spdep stuff

0.20 Spatial Regression and Autocorrelation

intro;

0.21 Raster Modelling

map algebra; ABM; SDM; Robert's book.

Part IV

Spatialtemporal Analysis

0.22 Array data: raster and vector data cubes

0.23 Movement data

0.24 Statistical modelling of spatiotemporal data

0.25 Scalability