Analysing Spatial Data in R: Why spatial data in R?

Roger Bivand

Department of Economics
Norwegian School of Economics and Business Administration
Bergen, Norway

31 August 2007

Why spatial data in R?

- What is R, and why should we pay the price of using it?
- How does the community around R work, what are its shared principles?
- How does applied spatial data analysis fit into R?
- But I have a non-standard research question ...

What is R?

- R is largely a 'GNU S', developed by some of the same people who developed S, plus a large group of public-spirited statisticians and programmers (many of whom had contributed libraries to S-PLUS).
- ... the goal of the R project was (and remains) to take the S language to the masses, using many features of S as the foundation of an open-source and freely-available statistics environment.
- R: some object-orientated design features, a strong emphasis on graphics and visualizing data, and a steady flow of innovation (both computational and statistical) from the applied statistics community.

How does the community around R work?

- Jackman (The Political Methodologist, Spring 2003): "One of the great strengths of programs like S-PLUS and R: user-extensibility or 'writing your own programs'."
- "once methodological problems start being perceived or even defined in terms of what one's favorite software does well, then the software has stopped being a tool, and has become a crutch, and at worse a shackle."
- R comes with a very wide range of functions for applied data analysis, and class definitions for objects like data frames, factors, etc.
- The community contributes further *packages* of documented code with examples — many available from CRAN.

Applied spatial data analysis with R

- Packages for importing commonly encountered spatial data formats
- Range of contributed packages in spatial statistics and increasing awareness of the importance of spatial data analysis in the broader community
- Current contributed packages with spatial statistics applications (see R spatial projects):

point patterns: **spatstat**, **VR:spatial**, **splancs**;
geostatistics: **gstat**, **geoR**, **geoRglm**, **fields**, **spBayes**, **RandomFields**, **VR:spatial**, **sgeostat**, **vardiag**;
lattice/area data: **spdep**, **DCluster**, **spgwr**, **ade4**.

A John Snow example

Even though we know that John Snow already had a working hypothesis about Cholera epidemics, his data remain interesting, especially if we use a GIS (GRASS) to find the street distances from mortality dwellings to the Broad Street pump:

```
v.digit -n map=vsnow4 bgcmd="d.rast map=snow"
v.to.rast input=vsnow4 output=rfsnow use=val value=1
r.buffer input=rfsnow output=buff2 distances=4
r.cost -v input=buff2 output=snowcost_not_broad \
  start_points=vpump_not_broad
r.cost -v input=buff2 output=snowcost_broad start_points=vpump_broad
```

We have two raster layers of cost distances along the streets, one distances from the Broad Street pump, the other distances from other pumps.

Non-standard research questions

- Because S (and its implementation R) is a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities, existing functions can be modified.
- Jackman: "if your notion of data analysis runs to more than estimating coefficients and t-statistics . . . then from time-to-time you'll find yourself programming, if only a little . . . easy programming and flexibility is key for a serious statistical computing environment."
- In any case, documenting the analysis process is a "good thing", so programming scripts are not just a burden, certainly for users doing original research and repetetive work, arguably for student classes too.

Cholera mortalities, Soho

We have read from GRASS into R a point layer of mortalities (counts per address) called `death`, the two distance cost raster layers, and the point locations of the pumps. Overlaying the addresses on the raster, we can pick off the street distances from each address to the nearest pump, and create a new variable `b_nearer`. Using this variable, we can tally the mortalities by nearest pump:

```
> o <- overlay(sohoSG, deaths)
> deaths <- spCbind(deaths, as(o, "data.frame"))
> deaths$b_nearer <- deaths$snowcost_broad < deaths$snowcost_not_broad
> by(deaths$Num_Cases, deaths$b_nearer, sum)

INDICES: FALSE
[1] 221
------------------------------------------------------------
INDICES: TRUE
[1] 357
```

Cholera mortalities, Soho

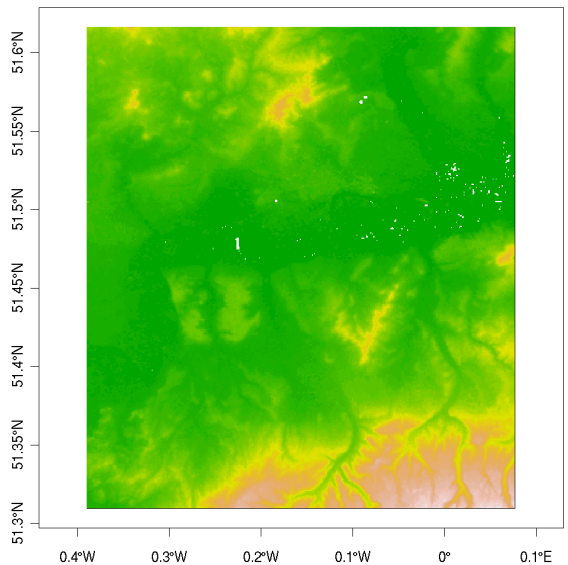
Seamless DEM for London

```
> library(rgdal)

Geospatial Data Abstraction Library extensions to R successfully loaded

> mt1 <- readOGR("22712073", "METADATA")

OGR data source with driver: ESRI Shapefile
Source: "22712073", layer: "METADATA"
with  2  rows and  23  columns

> DEM <- readGDAL("22712073/22712073.tif")

22712073/22712073.tif has GDAL driver GTiff
and has 368 rows and 560 columns

> summary(DEM$band1)

      Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
-3.403e+38  1.800e+01  3.500e+01 -8.917e+34  6.100e+01  2.130e+02

> is.na(DEM$band1) <- DEM$band1 <= 0
> summary(DEM$band1)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
   1.00   18.00   35.00   46.35   61.00  213.00  276.00

> plot(mt1, axes = TRUE)
> image(DEM, "band1", col = terrain.colors(40)[1:25], add = TRUE)
```

Maybe some data?

- Seamless DEM data from the Shuttle radar mission is now available
- We can see how this can be integrated with GPS positions (assuming that someone took one?)
- There is other data out there too, but more from US sources than any other
- If the online attempt fails, this is the canned version:

Seamless DEM for London



Analysing Spatial Data in R— I

- The background for the tutorial is provided in the R News note by Edzer Pebesma and myself, November 2005, pp. 9–13, with subsequent improvements
- First we'll look at the representation of spatial data in R, with stress on the classes provided in the **sp** package
- Before the break, we'll review the most useful methods provided are those for vizualisation, which form the next unit

Workshop infrastructure

- Task views are one of the nice innovations on CRAN that help navigate in the jungle of contributed packages — the Spatial task view is a useful resource
- The task view is also a point of entry to the Rgeo website hosted off CRAN, and updated quite often; it tries to mention in more detail contributed packages for spatial data analysis
- It also provides a link to the **sp** development area on Sourceforge, with CVS access to **sp**
- Finally, it links to the R-sig-geo mailing list, which is the prefered place to ask questions about analysing spatial/geographical data

Analysing Spatial Data in R— II

- After the break, we'll see how to read and write spatial data in commonly-used exchange formats, and how to handle coordinate reference systems
- Next we'll show how analysis packages for geostatistics are being adapted to use these representations directly
- Finally, we'll look at disease mapping, another area that is benefitting from being able to use shared classes and methods
- Before beginning to look at representation, just a few useful links:

Getting up to speed in R

- R is a programming language, so using it can build on earlier experience with programming (accepting that languages do vary).
- The "Introduction to R " shipped with R is up to date, and covers many questions quite well.
- There are a number of online resources too, both linked from CRAN, and others; Gilberto Câmara has assembled a systematic introduction bearing on spatial data.
- Robert Gentleman's introduction to classes and methods in R is still one of the clearest.