Recent Developments in Biostatistics: Space-Time Models Thursday: Visualization, data handling, spatial autocorrelation

Michael Höhle[1,2]

[1]Department for Infectious Disease Epidemiology, Robert Koch Institute, Berlin
[2]Department of Statistics
Ludwig-Maximilians-Universität München, Munich

8 July 2010
IBE, LMU München

Outline of the visualization part

1. Introduction
2. R classes for spatial data
3. Plotting
4. Import/Export

Outline of the spatial autocorrelation part

5. Introduction
6. Global indices of spatial autocorrelation
7. Local indicators of spatial association
8. Regression models and autocorrelation

Part I

Visualization

Aims of the visualization part

Introduction based on Bivand et al. (2008) into the classes for handling and visualizing spatial data in R

Import/Export of shapefile and other geodata into R

Link to GIS programs, especially GRASS

Outline

1 Introduction
2 R classes for spatial data
3 Plotting
4 Import/Export

Why visualization?

- Looking at your data is a first descriptive tool to generate hypotheses when analyzing spatial data.
- Difference between purely *descriptive visualization* and *analytic visualization* of results from, e.g., modelling.
- The visualization of the results from spatial modelling is one of the main reasons to integrate the handling of spatial data into a statistical software environment such as R.

Outline

1 Introduction
2 R classes for spatial data
3 Plotting
4 Import/Export

The R-package sp

- People with GIS experience think of spatial data in terms of *layers*, *coverages* and *rasters*, whilst statisticians think more in *data.frames*.
- The R package sp by Edzer Pebesma, Roger Bivand and others constitutes a compromise between those two worlds by providing a class hierarchy for spatial data.
- A number of additional R packages such as `maptools` and `rgdal` enable the import of maps and geodata as instances of sp classes.

Hello World in sp

- The foundation of the sp package are objects of the class `Spatial` and its subclasses.
- A first example:

```
> library("sp")
> m <- matrix(c(0, 0, 1, 1), ncol = 2, dimnames = list(NULL,
+     c("min", "max")))
> crs <- CRS(projargs = as.character(NA))
> (S <- Spatial(bbox = m, proj4string = crs))
An object of class "Spatial"
Slot "bbox":
     min max
[1,]   0   1
[2,]   0   1

Slot "proj4string":
CRS arguments: NA
```

- The slot `bbox` contains the *bounding box* and slot `proj4string` the *coordinate reference system* (PROJ.4 format)

Methods for `Spatial*` objects (1)

`[`	Selection of spatial elements (points, polygons, row/column of a grid) or attribute variables
`spsample`	generation of points within the object
`bbox`	get method for the bounding box
`proj4string`	get/set methods for the projection (CRS)
`coordinates`	get/set methods for the coordinates
`coerce`	convert from one class to another
`overlay`	combine two different spatial objects

Methods for sp objects (2)

- The typical usage of the `coerce` method is

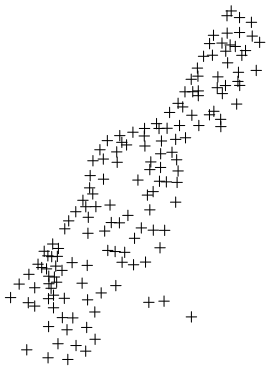
  `as(object, Class, strict=TRUE)`

  to convert, e.g., a `SpatialPointsDataFrame` object into a `data.frame` object.
- In addition, the usual `summary`, `print` and `plot` methods exist for `Spatial*` objects. Methoden .

SpatialPoints objects (1)

- The SpatialPoints class is the data abstraction for two-dimensional point clouds (x, y) within predefined regions.
- An additional slot coords contains the coordinates

```
> data("meuse")
> head(meuse, n = 3)

      x      y cadmium copper lead zinc elev
1 181072 333611    11.7     85  299 1022 7.91
2 181025 333558     8.6     81  277 1141 6.98
3 181165 333537     6.5     68  199  640 7.80
     dist   om ffreq soil lime landuse dist.m
1 0.00136 13.6     1    1    1      Ah     50
2 0.01222 14.0     1    1    1      Ah     30
3 0.10303 13.0     1    1    1      Ah    150

> coordinates(meuse) <- c("x", "y")
> plot(meuse)
```



SpatialPoints objects (2)

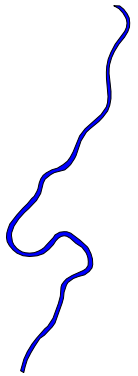
- SpatialPointsDataFrame objects are SpatialPoints objects with an additional attribute table.
- One possible way to create them:

```
> xy <- rbind(a = c(0, 0), b = c(1, 1))
> info <- data.frame(size = c(1, 2), row.names = c("b",
+     "a"))
> SpatialPointsDataFrame(xy, info, match.ID = FALSE)

  coordinates size
b      (0, 0)    1
a      (1, 1)    2

> SpatialPointsDataFrame(xy, info, match.ID = TRUE)

  coordinates size
a      (0, 0)    2
b      (1, 1)    1
```


SpatialPolygons objects (1)

- SpatialPolygons objects are the sp abstraction for a list of Polygons objects.
- A Polygons object consists of a list of Polygon objects. Each Polygons object is identified by a label ID.
- The Polygon objects contain the coordinates of the polygon as a two-column matrix. First row should equal last row.

```
> data("meuse.riv")
> dim(meuse.riv)

[1] 176   2

> riv.ps <- Polygons(list(Polygon(meuse.riv)),
+                          ID="meuse.riv")
> meuse.sp <- SpatialPolygons(list(riv.ps))
> plot(meuse.sp, col="blue")
```



SpatialPolygons objects (2)

- SpatialPolygons can also be regarded from the data.frame perspective. The rows of the attribute table contain the information for each Polygons object.
- The corresponding abstraction is given by the class SpatialPolygonsDataFrame.
- Example:

```
> info <- data.frame(a = 10, b = 3, row.names = "meuse.riv")
> meuse.spdf <- SpatialPolygonsDataFrame(meuse.sp,
+     info, match.ID = TRUE)
> as(meuse.spdf, "data.frame")

           a b
meuse.riv 10 3
```

SpatialGrid objects (1)

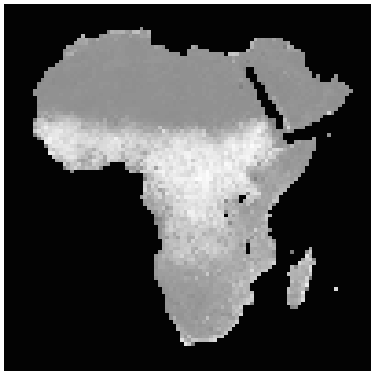
- Representation of grid and raster data is done by objects of the class SpatialGrid.
- Example: summary of a Normalized Difference Vegetative Index GeoTIFF file read using rgdal::readGDAL.

```
Object of class SpatialGridDataFrame
Coordinates:
       min     max
x -4599918 4616082
y -4615853 4600147
Is projected: TRUE
proj4string :
[+proj=aea +lat_1=-19 +lat_2=21 +lat_0=1 +lon_0=20 +x_0=0 +y_0=0 +ellps=WGS84
+datum=WGS84 +units=m +no_defs +towgs84=0,0,0]
Number of points: 2
Grid attributes:
  cellcentre.offset cellsize cells.dim
x          -4553838    92160       100
y          -4569773    92160       100
Data attributes:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -10000  -10000  -10000   -4940    1070    8880
```


SpatialGrid objects (2)

- The GridTopology object contains the definition of the grid
- SpatialGrid objects can be visualized using the function image.

```
> image(africa, col = grey(1:99/100))
```



SpatialPixels objects

In case many of the pixels in a SpatialGrid have an NA value, e.g. in case of non-quadratic observational regions, a SpatialPixels object can be a more efficient representation.

```
> data("meuse.grid")
> head(meuse.grid, n = 3)

       x      y part.a part.b   dist soil ffreq
1 181180 333740      1      0 0.0000    1     1
2 181140 333700      1      0 0.0000    1     1
3 181180 333700      1      0 0.0122    1     1

> coordinates(meuse.grid) <- c(1, 2)
> meuse.grid <- as(meuse.grid, "SpatialPixels")
> image(meuse.grid, col = "grey")
```

Further R classes for spatial data

- sp also contains classes SpatialLines and SpatialLinesDataFrame for line data.
- Chapter 6 in Bivand et al. (2008) contains a description and sp based implementation for
  - MultiPoint objects where one feature consists of a set of points
  - multidimensional Spatial* objects, e.g. space-time data
- The R package trip extends the SpatialPointsDataFrame class for space-time animal tracking data.

Working with Spatial* objects (1)

- `rgdal::spTransform` – coordinate transformations

```
> proj4string(meuse) = CRS("+init=epsg:28992")
> meuse.ll <- spTransform(meuse, CRS("+proj=longlat +datum=WGS84"))
```

- `spDists` – Calculation of distances between points – also as (WGS84) great-circle distance (option `longlat=TRUE`)

```
> head(spDists(meuse.ll, meuse.ll[1, ], longlat = TRUE),
+     n = 2)
```

```
       [,1]
[1,] 0.0000
[2,] 0.0708
```

- `spsample` – Generation of random points within or on a Spatial* object.

```
> coordinates(spsample(meuse.sp, n = 2, type = "random"))
```

```
          x      y
[1,] 180143 332422
[2,] 178473 330301
```


Working with Spatial* objects (2)

```
> #Let sp be a SpatialPolygons object consisting of three 3 Polygons objects
> length(slot(sp,"polygons"))
```

```
[1] 3
```

```
> #In which of the three polygons is a point located?
> overlay(sp,SpatialPoints(matrix(c(181072,
+      333611),ncol=2)))
```

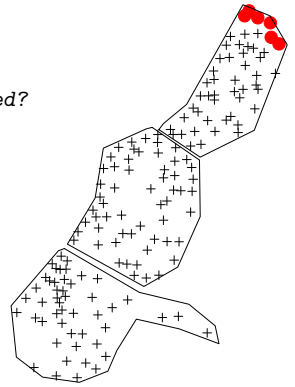
```
[1] 1
```

```
> #Das gleiche für ein SpatialPoints Objekt
> head(overlay(meuse,sp), n=6)
```

```
[1] 1 1 1 1 1 1
```

```
> #Equivalent to a "by" statement
> t(overlay(meuse[,c("cadmium")], sp, fn = mean))
```

```
          X1   X2   X3
cadmium 4.04 2.91 2.82
```



Working with Spatial* objects (3)

Combining several polygons into one `SpatialPolygons` object is done by the function `maptools::unionSpatialPolygons`:

```
> class(nc)[1]
```

```
[1] "SpatialPolygonsDataFrame"
```

```
> length(slot(nc, "polygons"))
```

```
[1] 100
```

```
> x.coord <- coordinates(nc)[, 1]
> ID <- cut(x.coord, quantile(x.coord, probs = seq(0, 1,
+     0.25)), include.lowest = TRUE)
> nc4 <- unionSpatialPolygons(nc, ID)
> sapply(slot(nc4, "polygons"), function(i) slot(i, "ID"))
```

```
[1] "[-84.1,-81.2]" "(-81.2,-79.3]" "(-79.3,-77.8]" "(-77.8,-75.9]"
```


Working with Spatial* objects (4)

With `maptools::spRbind` one can combine Spatial*DataFrames with identical column names.

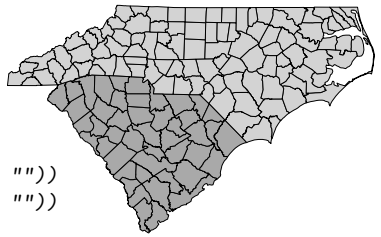
```
> c(class(nc)[1], class(sc)[1])
```

```
[1] "SpatialPolygonsDataFrame" "SpatialPolygonsDataFrame"
```

```
> head(as(nc, "data.frame"), n = 2)
```

```
      ST  CO      NAME
37001 37 001  Alamance
37003 37 003 Alexander
```

```
> sca <- spChFIDs(sc, paste(sc$ST, sc$CO, sep = ""))
> nca <- spChFIDs(nc, paste(nc$ST, nc$CO, sep = ""))
> ncsc <- spRbind(nca, sca)
> plot(ncsc, col = ifelse(ncsc$ST == 37, "lightgrey", "darkgrey"))
```



Working with Spatial* objects (5)

- The package spdep provides a number of function for creating spatial weights matrix objects from polygon contiguities and from point pattern by distance.
- Example: deduce neighbourhood structure from spatial polygons by poly2nb

```
> nc.nb <- include.self(poly2nb(nc))
> nc.nb

Neighbour list object:
Number of regions: 100
Number of nonzero links: 590
Percentage nonzero weights: 5.9
Average number of links: 5.9
```

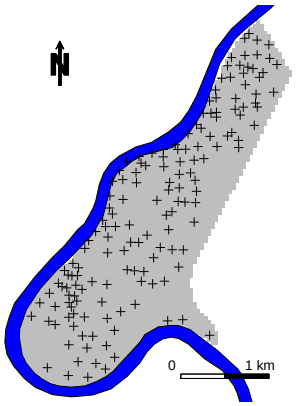

Outline

1 Introduction
2 R classes for spatial data
3 Plotting
4 Import/Export

Superposition of several Spatial* objects

When plotting one can superimpose several Spatial objects with the argument add=TRUE.

```
> image(meuse.grid, col="grey")
> plot(meuse.sp, col="blue", add=TRUE)
> plot(meuse, col="black", add=TRUE)
> SpatialPolygonsRescale(
+   layout.north.arrow(),
+   offset=c(178831,333024),
+   scale=500, plot.grid=FALSE)
> SpatialPolygonsRescale(
+   layout.scale.bar(),
+   offset=c(180300,329700),
+   scale=1000,plot.grid=FALSE,
+   fill=c("transparent","black"))
> text(180200, 329850,"0")
> text(181200, 329850,"1 km")
```



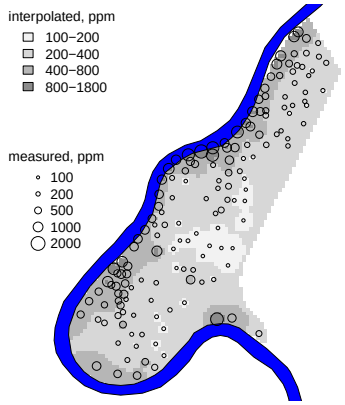
Visualizing attributes (1)

Adjusting the plot and image arguments, respectively, for plotting:

class	argument	description
SpatialPointsDataFrame	pch	symbol
	col	color
	bg	fill color
	cex	size of symbols
SpatialPolygonsDataFrame	border	border color
	density	fill density
	angle	angle of fill
	lty	line type
SpatialPixelsDataFrame	zlim	limits of the attribute
and	col	colors
SpatialGridDataFrame	breaks	interval endpoints

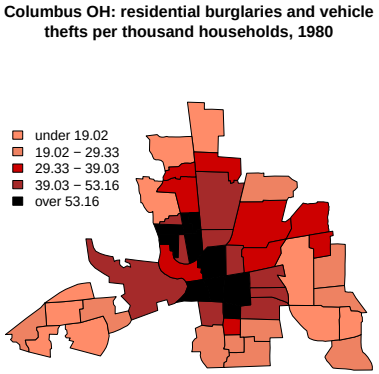
Visualizing attributes (2)

```
> cols <- grey.colors(4, 0.95, 0.55, 2.2)
> image(zn.idw, col = cols,
+       breaks=log(c(100,200,400,800,1800)))
> plot(meuse.sp, add = TRUE,col="blue")
> cex <- function(x) sqrt(x)/20
> plot(meuse, pch=1, cex = cex(meuse$zinc),
+      add = TRUE)
> legMeas <- c(100, 200, 500, 1000, 2000)
> legIpol <-c("100-200","200-400","400-800",
+             "800-1800")
> legend("left", legend=legMeas, pch = 1,
+        pt.cex = cex(legMeas), bty = "n",
+        title="measured, ppm")
> legend("topleft", fill = cols, legend=legIpol,
+        bty = "n", title = "interpolated, ppm")
```



Visualizing attributes (3)

```
> library("classInt")
> cols <- c("salmon1","salmon2",
+           "red3","brown","black")
> q5 <- classIntervals(columbus$CRIME,
+          n=5, style="quantile")
> q5Cols <- findColours(q5,cols)
> plot(columbus,col=q5Cols)
> legend(x=c(5.8, 7.1), y=c(13, 14.5),
+   legend=leglabs(round(q5$brks,digits=2)),
+   fill=cols, bty="n")
```



Visualizing attributes (4)

- The package `classInt` offers useful methods to discretize continuous variables for visualization.
- The package `RColorBrewer` contains several palettes optimized for thematic maps, see also `http://colorbrewer.org`
- with the function `spplot` one can also plot `Spatial*` objects by using Trellis graphics → `lattice` Paket.
- The functions `locator` and `identify` enable the interaction with the plots.

Visualizing rates (1)

- In spatial epidemiology, raw incidence rates can be statistically unstable due to the rarity of the disease and the relative small number of people at risk.
- Smoothing approaches "borrow" information from nearby regions to stabilize local estimates through weighting schemes.
- Methods range from simple weighting, to empirical Bayes smoothing to fullly Bayesian methods using Markov Chain Monte Carlo → *disease mapping*.

Visualizing rates (2)

- The neighbourhood set $\mathcal{N}(i)$ of a region i can be used to perform smoothing by locally weighted averaging

$$\tilde{r}_i = \frac{\sum_{j=1}^{N} w_{ij}\, r_j\, I(j \in \mathcal{N}(i))}{\sum_{j=1}^{N} w_{ij}\, I(j \in \mathcal{N}(i))},$$

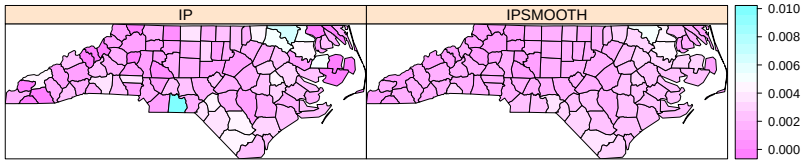
  where $r_1, \ldots, r_N$ are the observed rates in the N regions.
- The weights w_{ij} could, e.g., depend on the population size of region j.

Visualizing rates (3)

- Example: Neighbourhood smoothing of the sudden infant deaths 1974–1978 in North Carolina

```
> nc$IP <- nc$SID74/nc$BIR74
> nc$IPSMOOTH <- sapply(1:nrow(nc), function(i) {
+     w <- nc$BIR74[nc.nb[[i]]]
+     r <- nc$IP[nc.nb[[i]]]
+     return(sum(w * r)/sum(w))
+ })
```

- Result:



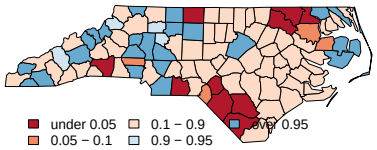
Visualizing rates (3)

- Instead of plotting raw or smoothed rates one can also for each region plot the p-value for the following test:
- Assume $Y_i \sim \text{Po}(E_i\lambda_i)$ and test for each region

$$H_0 : \lambda_i = 1 \quad \text{vs.}$$
$$H_1 : \lambda_i > 1$$

  Here, E_i is the expected number of counts obtained by, e.g., age standardization or simply $E_i = n_i \cdot \sum_{i=1}^{N} Y_i / \sum_{i=1}^{N} n_i$.

```
> pmap <- probmap(n=nc[["SID74"]],
+   x=nc[["BIR74"]],
+   alternative="greater")$pmap
> brks <- c(0,0.05,0.1,0.9,0.95,1)
> cols <- brewer.pal(6,"RdBu")
> plot(nc, col=cols[findInterval(pmap,
+       brks, all.inside=TRUE)])
```



Outline

1 Introduction
2 R classes for spatial data
3 Plotting
4 Import/Export

Import/Export of geodata as sp objects

- The package `maptools` enables:
  - the import and export of ESRI shapefiles or ArcInfo ASCII grids as sp objects
  - creation of Keyhole Markup Language (KML) files and PNG image overlays for Google Earth
  - conversion of sp objects to ppp objects of the `spatstat` package → spatial point processes
- The package `rgdal` contains an R inferface for the
  - *Geospatial Data Abstraction Library* (GDAL) which can import and export GDAL raster and OGR vector formats as sp objects
  - `PROJ.4` – *Cartographic Projections Library* for map projections

Shapefiles (1)

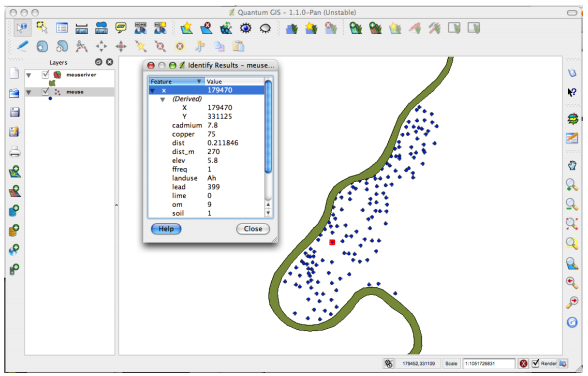
- The function `maptools::readShapeSpatial` reads data from a shapefile into a `Spatial*DataFrame`.
- The argument `IDvar` indicates the name of the column in the DBF file which contains the ID of the shapes.
- An alternative function for reading shapefiles is `rgdal::readOGR`.

```
> nc <- readShapeSpatial(system.file("shapes/sids.shp",
+    package="maptools")[1], IDvar="FIPSNO",
+    proj4string=CRS("+proj=longlat +ellps=clrk66"))
> class(nc)[1]

[1] "SpatialPolygonsDataFrame"

> head(as(nc,"data.frame"), n=1)

       AREA PERIMETER CNTY_ CNTY_ID     NAME  FIPS FIPSNO CRESS_ID
37001 0.111      1.39  1904    1904 Alamance 37001  37001        1
      BIR74 SID74 NWBIR74 BIR79 SID79 NWBIR79
37001  4672    13    1243  5767    11    1397
```


Shapefiles (2)

The function `maptools::writeSpatialShape` writes the data of a `Spatial*DataFrame` into a shapefile.

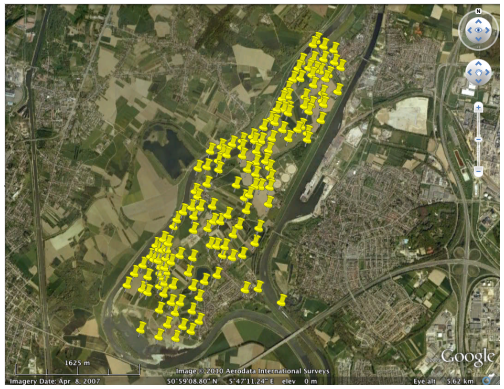
```
> writeSpatialShape(meuse.spdf, "../Data/meuseriver")
> writeSpatialShape(meuse, "../Data/meuse")
```



Google Earth export (1)

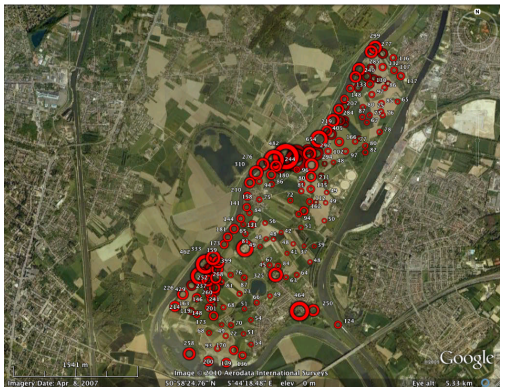
- Point and polygon data can be exported from R as KML files for use in Google Earth

```
> writeOGR(meuse.ll["lead"], dsn = "../Data/meuse_lead.kml",
+     layer = "lead", driver = "KML")
```



Google Earth export (2)

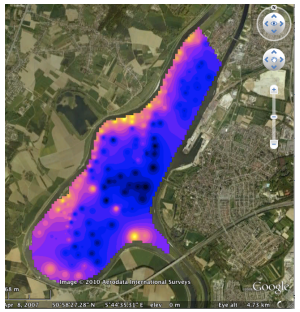
- The KML file can also be created manually with R – see function `makeKMLBubblePlot` in R code of foils for an example.



Google Earth export (3)

- PNG overlay for Google Earth with `maptools::kmlOverlay`, Pre-processing with `GE_SpatialGrid`.
- Kriging of the zinc concentration (see R-code for details):

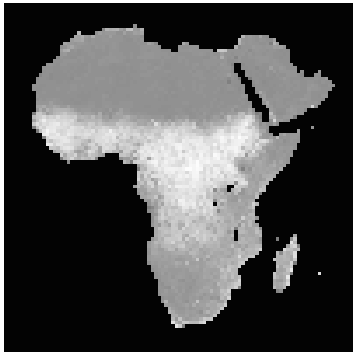
```
> png(file="../Data/zinc.png", width=llGRD$width,
+     height=llGRD$height, bg="transparent")
> par(mar=c(0,0,0,0), xaxs="i", yaxs="i")
> image(llSPix, "pred", col=bpy.colors(20))
> dev.off()
> kmlOverlay(llGRD, "../Data/zinc.kml",
+                   "../Data/zinc.png")
```



GeoTIFF files

GeoTIFF is a special TIFF-Format for raster data which allows georeferencing (map projection, coordinate system, etc.) information to be embedded in the file.

```
> library("rgdal")
> africa <- readGDAL("../Data/africa.tif",
+     output.dim = c(100, 100))
../Data/africa.tif has GDAL driver GTiff
and has 1152 rows and 1152 columns
> class(africa)[1]
[1] "SpatialGridDataFrame"
> image(africa, col = grey(1:99/100))
```



Sources for further information

- Homepage of the book *Applied Spatial Data Analysis with R* (Bivand et al., 2008):

  `http://www.asdar-book.org/`

  contains code and data for all visualizations and analyses in the book.
- A similar OpenAccess book is *A Practical Guide to Geostatistical Mapping* (Hengl, 2010)

  `http://spatial-analyst.net/`
- CRAN Task View: Analysis of Spatial Data
- R-spatial project

Part II

Spatial Autocorrelation

Aims of the lecture on spatial autocorrelation

Understand the concept and implications of spatial autocorrelation

Learn about global and local measures of spatial associations, specifically *Moran's I* as described in, e.g., Chapter 7 of Waller and Gotway (2004)

Compute these indices yourself by using the appropriate R functions from the package `spdep`.

Outline

5 Introduction

6 Global indices of spatial autocorrelation

7 Local indicators of spatial association

8 Regression models and autocorrelation

What is spatial autocorrelation?

- Consider the measurement variable $Y(s)$ at location $s \in \mathbb{R}^2$. If there is a systematic spatial pattern in the distribution of Y, then it is said to be spatially autocorrelated.
- *Spatial autocorrelation* measures the degree of dependency among $Y(s)$ in geographic space.
- A *global index* of spatial autocorrelation provides a summary over the entire study area for the similarity of $Y(s)$ values among neighbouring observations.
- Using a global index to analyze disease patterns provides a test of *clustering* more than a test to detect individual *clusters*

Why is spatial autocorrelation important? (1)

- Spatial autocorrelation is problematic for classical statistical tests because they typically assume independently distributed errors.
- Estimates are biased because in reality there are fewer number of independent observations than assumed → increased tendency to reject the null hypothesis when it is true.
- Effects of covariables that are themselves autocorrelated tend to be exaggerated.
- Spatial autocorrelation quantifies the amount of redundancy in the observed georeferenced data.

Why is spatial autocorrelation important? (2)

- In today's lecture focus is on computing indices of spatial autocorrelation for regional data.
- Methods apply to geostatistical data as well (→ Tuesday's lecture), but here, e.g., variograms are more frequently used to investigate autocorrelation.

Data situation (1)

- Notation:
  - Y response variable
  - s location variable, geographical region
  - x vector of covariates (if available)
- In spatial statistics we differ between *geostatistical data*, *lattice/regional data* and *spatial point process data*.
- For geostatistical data, $s = (s_x, s_y)$ is a vector of length two denoting the coordinates of the location and $s \in D \subseteq \mathbb{R}^2$ is continuous in D.
- For regional data $s_i = (s_i^x, s_i^y)$ is a vector of length two typically denoting the centroid location of region $i \in \{1, \ldots, N\}$.

Data situation (2)

- $Y(s)$ represents the measurement on a given scale (continuous, discrete, ordinal, categorical) at the location s.
- Basic format for today's data:

$$\{(s_i, y_i) : i = 1, \ldots, N\},$$

  with $Y(s)$ being continuous.
- For regional data we often skip s_i and just talk about $Y_i, i = 1, \ldots, N$
- Each data point can furthermore include a p-dimensional vector of spatial explanatory variables $\mathbf{x}_i$:

$$\{(s_i, y_i, \mathbf{x}_i) : i = 1, \ldots, N\}.$$

Outline

5 Introduction

6 Global indices of spatial autocorrelation

7 Local indicators of spatial association

8 Regression models and autocorrelation

Global indices spatial autocorrelation

- Let sim_{ij} denote the similarity between values of Y_i and Y_j.
- Indices of spatial autocorrelation such as *Moran's I* and *Geary's C* are on the basic form

$$\frac{\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}\,\text{sim}_{ij}}{\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}},$$

  where w_{ij} are known weights describing the proximity between locations i and j.
- The collection of weights w_{ij} is typically given by a *spatial proximity matrix* $\mathbf{W} = \{w_{ij}\}$.

Examples of spatial proximity matrices (1)

- Binary connectivity matrix

$$w_{ij} = I(\text{if regions } i \text{ and } j \text{ share a boundary})$$

- q-nearest neighbours

$$w_{ij} = I(\text{if the centroid of region } j \text{ is one of the } q \text{ nearest to the centroid of region } j)$$

- Distance based, i.e. for given δ

$$w_{ij} = I(d_{ij} < \delta),$$

  where d_{ij} is a measure of distance (Euclidean, city-block, etc.).

Examples of spatial proximity matrices (2)

- In some situation, there is an interest to adjust the weights for the number of neighbours, e.g. by requiring each row sum to be one.
- A row-standardized spatial proximity matrix has weights of the form

$$w_{i,j}^{std} = \frac{w_{ij}}{\sum_{j=1}^{N} w_{ij}}.$$

- Note that for row-standardized matrixes the weights for links in regions with few neighbours are higher than for regions with many neighbours.

Examples of spatial proximity matrices (3)

- In R spatial proximity matrices can be generated from neighbourhood structures using the function `nb2listw`.
- The argument `style` can be used to select binary coding (`"B"`) row standardized form (`"W"`) and other more complicated forms.

```
> nb <- poly2nb(nc)
> (W.bin <- nb2listw(nb, style = "B"))

Characteristics of weights list object:
Neighbour list object:
Number of regions: 100
Number of nonzero links: 490
Percentage nonzero weights: 4.9
Average number of links: 4.9

Weights style: B
Weights constants summary:
    n     nn  S0  S1    S2
B 100 10000 490 980 10696
```


Moran's I (1)

- One of the oldest – but still very much in use – indices is *Moran's I*(Moran, 1950), where similarity is calculated by

$$\text{sim}_{ij} = (Y_i - \overline{Y})(Y_j - \overline{Y}), \quad \overline{Y} = \frac{1}{N}\sum_{i=1}^{N} Y_i.$$

- The index is defined by additionally dividing by the sample variance

$$I = \left(\frac{1}{s^2}\right) \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}(Y_i - \overline{Y})(Y_j - \overline{Y})}{\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}}, \qquad (1)$$

where $s^2 = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \overline{Y})^2$.

Moran's I (2)

- Can be applied to continuous outcomes where location is either the centroid of the region or the exact point location
- Unlike a traditional correlation coefficient, Moran's I is not constrained to be in the range $[-1, 1]$, but typically $|I| < 1$.
- If there is no autocorrelation $E(I) = -1/(N-1)$.
- Large positive values of I indicate neighbourhood similarity and values close to zero indicate absence of autocorrelation.
- Looking at regional disease counts: Any autocorrelation in the data may simply be due to relationships among the underlying populations of the regions $\rightarrow$ adjust for heterogeneous population sizes.

Moran's I (3)

- Solution: Compare instead observed count with its expectation under a constant risk hypothesis (Walter, 1992):

$$I_{cr} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}\frac{Y_i - rn_i}{\sqrt{rn_i}}\frac{Y_j - rn_j}{\sqrt{rn_j}}}{\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}}. \qquad (2)$$

- I_{cr} corresponds to Moran's I applied to the Pearson residuals from a GLM with relative risk r and region specific offset n_i.
- Regression link: An often used procedure when e.g. using linear regression for spatial data is to compute Moran's I for the residuals to see whether spatial autoregression after correction for covariates is of concern.

Significance testing

- If the observations are independent and identical Gaussian RVs we can compare

$$z = \frac{I - E(I)}{\sqrt{\mathrm{Var}(I)}}$$

  to the standard normal distribution. $E(I)$ as before and $\mathrm{Var}(I)$ is given in e.g. Waller and Gotway (2004, p. 229).
- If we assume that the distribution of I is invariant to a random assignment of the observations to locations we can use a Monte Carlo test to assess significance.

Moran's I in R (1)

- The Moran's I statistic can be computed by the R function `moran` and testing is performed using `moran.test`:

```
> moran.test(nc$SID74, listw = W.bin)

        Moran's I test under randomisation

data:  nc$SID74
weights: W.bin

Moran I statistic standard deviate = 2.1707, p-value = 0.01498
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
      0.119089049      -0.010101010       0.003542176
```

```
> moran.test(nc$SID74/nc$BIR74, listw = W.bin)$p.value
[1] 0.0001386948
```


Moran's I in R (2)

- Note, that both the pure count and the incidence proportion approach do not adequately take the varying population sizes of the region into account.
- Poisson GLM based Moran's I:

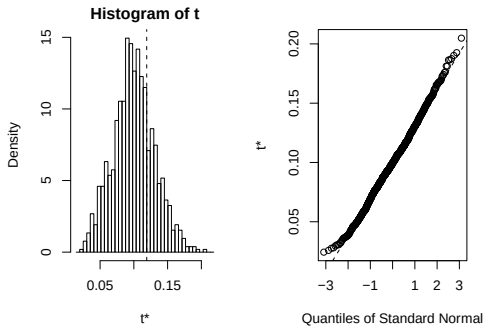
```
> mu.hat <- fitted(glm(SID74 ~ 1 + offset(log(BIR74)),
+     family = poisson, data = nc))
> CR <- function(var, mle) {
+     rpois(length(var), lambda = mle)
+ }
> MoranI.pboot <- function(var, i, listw, n, S0, ...) {
+     return(moran(x = var, listw = listw, n = n, S0 = S0)$I)
+ }
> res.boot <- boot(nc$SID74, statistic = MoranI.pboot,
+     R = 999, sim = "parametric", ran.gen = CR, listw = W.bin,
+     n = nrow(nc), S0 = Szero(W.bin), mle = mu.hat)
```


Moran's I in R (2)

- Corresponding p-value:

```
> mean(c(res.boot$t, res.boot$t0) >= res.boot$t0)
[1] 0.249
```

- Histogram of the $R = 999$ parametric bootstrap values together with the observed value I_{cr}:



Outline

5 Introduction

6 Global indices of spatial autocorrelation

7 Local indicators of spatial association

8 Regression models and autocorrelation

Local indicators of spatial association (LISAs)

- Global indicators can suggest clustering but cannot identify individual clusters.
- Solution to investigate local clustering: Break down global measures of spatial association into their local components.
- High LISAs may be due to aggregation of high counts or proportions just as well as aggregation of low counts or proportions.

Local Moran's I

- A local version of Moran's I for region i is given as

$$I_{i,\text{std}} = \frac{Y_i - \overline{Y}}{s} \sum_{j=1}^{N} w_{ij} \frac{Y_j - \overline{Y}}{s}.$$

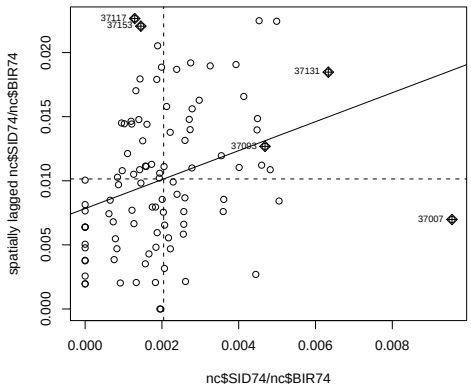
- Note that $\sum_{i=1}^{N} I_{i,\text{std}} = I \cdot \left(\sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} \right)$.
- We follow (2) to obtain an adjusted local Moran's I for the constant risk hypothesis:

$$I_{i,cr} = \frac{Y_i - rn_i}{\sqrt{rn_i}} \sum_{j=1}^{N} w_{ij} \frac{Y_j - rn_j}{\sqrt{rn_j}}$$

Moran Scatterplot

- A *Moran scatterplot* is a plot of y_i against $\sum_{j=1}^{N} w_{ij}(y_j - \overline{y})$.

```
> moran.plot(nc$SID74/nc$BIR74, listw = W.bin)
```



- Because the global I reflects a linear relationship between the I_i's, standard regression techniques for identifying observations with strong influence on the slope apply.

Local Moran's I in R (1)

- The function `localmoran` computes local Moran's I and provides a *p*-value based on the normal approximation (which can be problematic!)

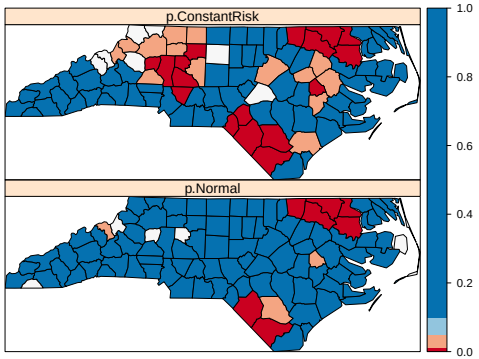
```
> lm <- localmoran(nc$SID74/nc$BIR74, listw = W.bin)
```

- The constant risk assumption can again be assessed by parametric bootstrap

```
> mu.hat <- fitted(glm(SID74 ~ 1 + offset(log(BIR74)),
+     family = poisson, data = nc))
> localMoranI.pboot <- function(y, i, listw, ...) {
+     sdCRi <- (y - mu.hat)/sqrt(mu.hat)
+     return(sdCRi * lag(listw, sdCRi))
+ }
> res.boot <- boot(nc$SID74, statistic = localMoranI.pboot,
+     R = 9999, sim = "parametric", ran.gen = CR, listw = W.bin,
+     mle = mu.hat)
> p <- apply(rbind(res.boot$t, res.boot$t0), MARGIN = 2,
+     function(x) mean(x > x[length(x)]))
```


Local Moran's I in R (2)

- Plot of the p-values based on the Normal approximation and the bootstrap p-value for the constant risk approach:



- Values close to zero indicate clusters in the data where regions with similar values neighbour each other.

Issues with LISAs

Calculation of *p*-values for LISAs is complicated by

- the multiple testing problem of conducting a separate statistical test for each region
- the correlation between adjacent LISAs
- that many tests based on small sample sizes are conducted

As a consequence, the *p*-values should not be interpreted as more than indicators.

Outlook

- So far, general tests to detect clusters anywhere in the study area have been presented.
- Often we have set of locations of putative sources for increased disease risk (highways, nuclear power plant, etc.).
- The question is: Is there evidence of increased disease incidence in areas exposed to the foci?
- *Focused tests* such as Stone's test address this problem, see e.g. Waller and Gotway (2004) for details.

Outline

5 Introduction

6 Global indices of spatial autocorrelation

7 Local indicators of spatial association

8 Regression models and autocorrelation

Regression models and autocorrelation (1)

- OLS regression estimators are unbiased and the associated confidence intervals are correct only if the model is correctly specified and includes all relevant covariates.
- If we omit an important covariate and this the covariate varies spatially, we will have residual spatial variation.
- Even if we include all relevant covariates, there may still be residual spatial autocorrelation due to the complex connection of the data generating mechanism, e.g. transmission between individuals for infectious diseases.

Regression models and autocorrelation (2)

- If spatial autocorrelation is present, p-values for significance of covariate effects may be biased towards rejecting the null-hypothesis even when it is true.
- How to detect presence of spatial autocorrelation in a regression context?
  - Linear models (OLS)

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \text{ with } \boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2\mathbf{I})$$

  - Generalized linear models (GLM), i.e. $\mathbf{Y}$ is from a one-parameter exponential family with expectation characterized by

$$g(E(\mathbf{Y})) = \mathbf{X}\boldsymbol{\beta}.$$

Moran's I for OLS residuals (1)

- Let the OLS residuals be

$$\mathbf{e} = \mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}_{\text{OLS}},$$

  where $\hat{\boldsymbol{\beta}}_{\text{OLS}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$.
- Instead of $\overline{Y}$ use the predicted value $\hat{\mathbf{Y}} = \mathbf{X}\hat{\boldsymbol{\beta}}_{\text{OLS}}$ in Equation (1) of Moran's I:

$$\begin{aligned} I_{res} &= \frac{N}{\sum_{i=1}^N (Y_i - \hat{Y}_i)^2} \left[\frac{\sum_{i=1}^N \sum_{j=1}^N w_{ij}(Y_i - \hat{Y}_i)(Y_j - \hat{Y}_j)}{\sum_{i=1}^N \sum_{j=1}^N w_{ij}} \right] \\ &= \frac{N}{\sum_{i=1}^N \sum_{j=1}^N w_{ij}} \frac{\mathbf{e}'\mathbf{W}\mathbf{e}}{\mathbf{e}'\mathbf{e}} \end{aligned}$$

Moran's I for OLS residuals (2)

- Based on the assumption that the residuals follow a multivariate Normal distribution one can show that under the null hypothesis of independent error terms and constant variance for the OLS residuals

$$I_{res} \overset{a}{\sim} N(\mu_{I_{res}}, \sigma^2_{I_{res}}),$$

with $\mu_{I_{res}}$ and $\sigma^2_{I_{res}}$ given by lengthy equations found, e.g., in Cliff and Ord (1981, p. 202-203) or (Waller and Gotway, 2004, p. 368).

- Significance tests can thus be based on comparing

$$z = \frac{I_{res} - \mu_{I_{res}}}{\sigma_{I_{res}}}$$

with quantiles of the standard normal distribution.

Moran's I for OLS residuals in R (1)

- The R function `lm.morantest` does the necessary calculations. Note that if just an intercept is used, we are back to (1).

```
> moran.test(nc$SID74/nc$BIR74, listw = W.bin, randomisation = FALSE)$p.val
[1] 0.0001888785
> m1 <- lm(SID74/BIR74 ~ 1, data = as(nc, "data.frame"))
> lm.morantest(m1, listw = W.bin)

        Global Moran's I for regression residuals

data:
model: lm(formula = SID74/BIR74 ~ 1, data = as(nc,
"data.frame"))
weights: W.bin

Moran I statistic standard deviate = 3.5552, p-value =
0.0001889
alternative hypothesis: greater
sample estimates:
Observed Moran's I        Expectation           Variance
      0.210046454       -0.010101010        0.003834515
```


Moran's I for OLS residuals in R (2)

- If available, further covariates can be added in the linear predictor:

```
> m2 <- lm(SID74/BIR74 ~ 1 + I(NWBIR74/BIR74), as(nc, "data.frame"))
> lm.morantest(m2, listw = W.bin)$p.value

          [,1]
[1,] 0.1320989
```

- Note: In the above application the population heterogeneity over regions is again not appropriately addressed.

Outlook (1)

- If tests reveal remaining spatial autocorrelation one might consider including additional covariates or use a model class taking the autocorrelation into account (e.g. SAR or CAR model).
- Lin and Zhang (2007) propose using deviance or Pearson residuals in (1) to assess spatial autocorrelation in Poisson GLMs.
- Significance is in this approach assessed conditioned on the covariates by using parametric bootstrap as illustrated for the population offset Poisson model.

Outlook (2)

- A different descriptive method to study the effect of spatial autocorrelation in regression modelling is *geographically weighted regression* (GWR) (Brunsdon et al., 1998).
- A linear model with spatially varying coefficients is assumed:

$$Y_i = \mathbf{x}_i' \beta(\mathbf{s}_i) + \epsilon_i, \quad i = 1, \ldots, N.$$

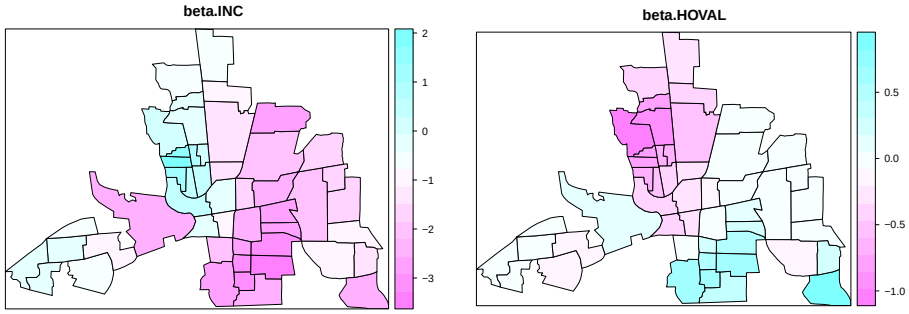
- Similar to kernel regression a WLS model is fitted for each data point i to obtain

$$\hat{\beta}(\mathbf{s}_i) = (\mathbf{X}'\mathbf{W}_i\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}_i\mathbf{y}$$

  with $W_i = \text{diag}(\alpha_{i1}, \ldots, \alpha_{iN})$ and $\alpha_{ik} = K(d_{ik}), k = 1, \ldots, N$.
- Here, $K(\cdot)$ is a univariate kernel function with bandwidth h and d_{ik} is the distance between $\mathbf{s}_i$ and $\mathbf{s}_k$.

Outlook (3)

- One obtains an impression of the spatial variation in $\beta(\mathbf{s})$ by plotting $\hat{\beta}(\mathbf{s})$:



- Tests exist for investigating the hypothesis that the regression coefficients do not vary spatially, see e.g. Brunsdon et al. (1999) and in R functions `BFC99.gwr.test` and friends.

Literature I

Bivand, R. S., Pebesma, E. J., and Gómez-Rubio, V. (2008). *Applied Spatial Data Analysis with R*. Springer.

Brunsdon, C., Fotheringham, S., and Charlton, M. (1998). Geographically weighted regression – modelling spatial non-stationarity. *The Statistician*, 47(3):431–443.

Brunsdon, C., Fotheringham, S., and Charlton, M. (1999). Some notes on parametric significance tests for geographically weighted regression. *Journal of Regional Science*, 39(3):497–524.

Cliff, A. and Ord, J. K. (1981). *Spatial Processes, Models and Applications*. Pion.

Hengl, T. (2010). *A Practical Guide to Geostatistical Mapping*. Lulu. Open Access Publication.

Lin, G. and Zhang, T. (2007). Loglinear residual tests of moran's i autocorrelation and their application to kentucky breast cancer data. *Geographical Analysis*, 39:293–310.

Moran, P. A. P. (1950). Notes on continuous stochastic phenomena. *Biometrika*, 37:17–23.

Waller, L. A. and Gotway, C. A. (2004). *Applied Spatial Statistics for Public Health Data*. Wiley.

Walter, S. D. (1992). The analysis of regional patterns in health data: Ii. the power to detect environmental effects. *American Journal of Epidemiology*, 136(6):742–759.