Process scale and autocorrelation

Roger Bivand

24 August 2018

Overview

- We'll try to use **sf** instead of **sp** in analysing the Boston data set
- In addition, the results I presented at useR! 2016 on this data set aggregated the data to air pollution model output zones (see also Bivand, 2017)
- Here, based on Bivand et al. (2017), spatially structured random effects are added by air pollution model output zones instead
- The purpose is to take **sf** out for a run, not to promote multilevel models of spatially structured data

Boston data set

- Harrison and Rubinfeld (1978) used a hedonic model to find out how house values were affected by air pollution in Boston, when other variables were taken into consideration
- They chose to use 506 census tracts as units of observation, but air pollution values were available from model output for 122 zones, of which less than 100 fell within the study area
- By taking the 96 air pollution model output zones as the upper level in a hierarchical spatial model, we explore the consequences for the results

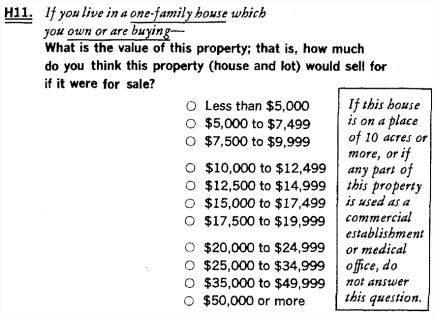
Introduction

- The Harrison and Rubinfeld (1978) Boston housing data set has been widely used because of its availability from Belsley et al. (1980), Pace and Gilley (1997) and Gilley and Pace (1996)
- The underlying research question in the original article was the estimation of willingness to pay for clean air, using air pollution levels and house values in a hedonic regression
- As Pace and Gilley (1997, p. 337) showed clearly, the air pollution coefficient estimate in the model changed when residual spatial autocorrelation was taken into account (from -0.0060 to -0.0037), as did its standard error (from 0.0012 to 0.0016)

Motivation

- Is the strength of spatial autocorrelation observed in this data set a feature of the census tract observations themselves, or has it been introduced or strengthened by changes in the observational units used for the different variables?
- Our focus will be on the choices of observational units made in assembling the original data set, and on another relevant alternative
- Using an approximation to the model output zones from which the air pollution variable levels were taken, it will be shown that much of the puzzling spatial autocorrelation is removed

House value data



H11. *If you live in a one-family house which you own or are buying—*
What is the value of this property; that is, how much do you think this property (house and lot) would sell for if it were for sale?

- ○ Less than $5,000
- ○ $5,000 to $7,499
- ○ $7,500 to $9,999
- ○ $10,000 to $12,499
- ○ $12,500 to $14,999
- ○ $15,000 to $17,499
- ○ $17,500 to $19,999
- ○ $20,000 to $24,999
- ○ $25,000 to $34,999
- ○ $35,000 to $49,999
- ○ $50,000 or more

If this house is on a place of 10 acres or more, or if any part of this property is used as a commercial establishment or medical office, do not answer this question.

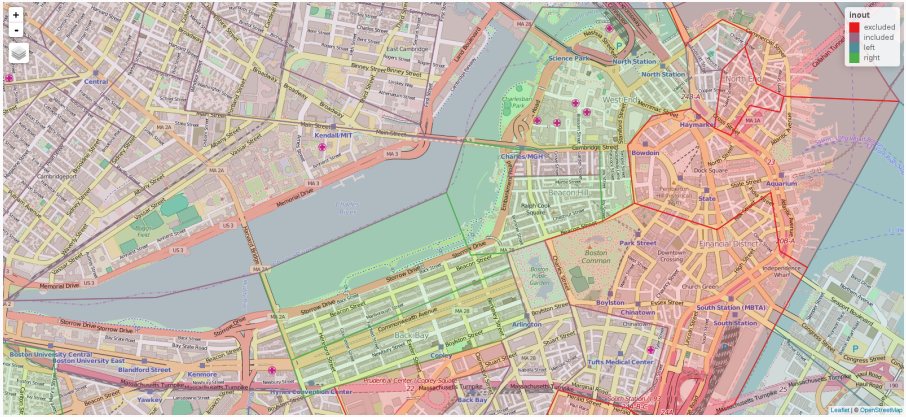
Harrison and Rubinfeld (1978) used median house values in 1970 USD for 506 census tracts in the Boston SMSA for owner-occupied one-family houses; census tracts with no reported owner-occupied one-family housing units were excluded from the data set. The relevant question is H11, which was answered by crossing off one grouped value alternative, ranging from under USD 5,000 to over USD 50,000

House value data

- The house value data have census tract support, and are median values calculated from group counts from the alternatives offered in H11; tracts with weighted median values in these upper and lower alternative value classes are censored
- The published census tract tabulations show the link between question H11 and the Statlib-based data (after correction)
- The median values tabulated in the census report can be reconstructed from the tallies shown in the same Census tables fairly accurately using the `weightedMedian` function in the **matrixStats** package in R, using linear interpolation

House value data

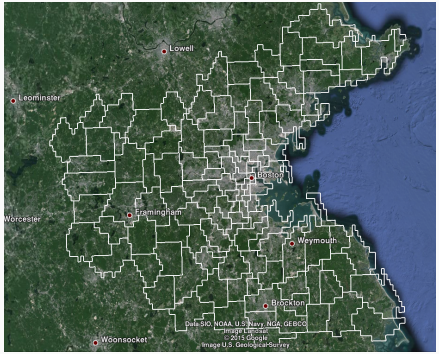
The effectiveness of the study was prejudiced by the fact that areas of central Boston with the highest levels of air pollution also lose house value data, either because of tract exclusion (no one-family housing units reported) or right or left censored tracts



Air pollution data

- The data on air pollution concentrations were obtained from the Transportation and Air Shed SIMulation model (TASSIM) applied to the Boston air shed (Ingram and Fauth, 1974)
- The calibrated model results were obtained for 122 zones, and assigned proportionally to the 506 census tracts
- The NOX values in the published data sets are in units of 10 ppm (10 parts per million), and were then multiplied by 10 again in the regression models to yield parts per 100 million (pphm)
- Many of the smaller tracts belong to the same TASSIM zones; this is a clear case of change of support, with very different spatial statistical properties under the two different entitation schemes (Gotway and Young, 2002)

Air pollution data



A two-part report details the use of the TASSIM simulation model (Ingram and Fauth, 1974; Ingram et al., 1974). Both of these volumes include line-printer maps of the TASSIM zones, and the Fortran code in volume 2 (Ingram et al., 1974, pp. 183–185) shows the links between the 122 TASSIM zones and the line printer output. Western TASSIM zones appear to lie outside the Boston SMSA tracts included in the 506 census tract data set.

Using sf

- The Boston data set has 2D polygon and multipolygon geometries stored in a shapefile; shapefiles are pre-SF
- **sf** depends on GDAL for reading vector geometries and attribute data of features into rows of a data.frame
- The geometries are put in a special column in the data.frame
- The other columns contain the data for the census tracts included in the original data set, supplemented with others, such as the censoring status and model output zones

Reading the shapefile

Functions in **sf** are prefixed with `st_` meaning spatio-temporal (following PostGIS); `sf::st_read` uses GDAL vector drivers:

```
> library(sf)

## Linking to GEOS 3.7.0rc1, GDAL 2.3.1, proj.4 5.1.0

> b506 <- st_read("boston_506.shp")

## Reading layer ‘boston_506’ from data source ‘/home/rsb/und/uam18/part2/uio17/courses/pazur17/boston_506.shp’ using driver ‘ESRI Shapefile’
## Simple feature collection with 506 features and 43 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -71.52311 ymin: 42.00305 xmax: -70.63823 ymax: 42.67307
## epsg (SRID):    4267
## proj4string:    +proj=longlat +datum=NAD27 +no_defs
```

Aggregating geometries to model output zones

After dropping the censored census tracts, we need to derive the model output zones. The aggregate method calls `sf::st_union` on each unique grouping value, but the function called internally is a trick, putting only the first value of each variable in the output; for this reason we only retain the ids:

```
> b489 <- b506[b506$censored == "no",]
> t0 <- aggregate(b489, list(ids = b489$NOX_ID), head, n = 1)
> b94 <- t0[, c("ids", attr(t0, "sf_column"))]
```

Finding model output zones neighbours

We can find contiguous neighbours using `sf::st_relate`, but this does not yet scale to large numbers of geometries; we also find a no-neighbour model output zone:

```
> st_queen <- function(a, b = a) st_relate(a, b, pattern = "F***T****")
> qm1 <- st_queen(b94)

## although coordinates are longitude/latitude, st_relate_pattern assumes that they are planar

> any(sapply(qm1, length) == 0)

## [1] TRUE
```

Problem of no-neighbour zone

Both Bayesian multilevel spatial model fitting approaches fail on the no-neighbour case (the spatially structured random effect cannot be estimated), so we need to drop those census tracts and re-aggregate to model output zones with neighbours:

```
> NOX_ID_no_neighs <- b94$ids[which(sapply(qm1, length) == 0)]
> b487 <- b489[is.na(match(b489$NOX_ID, NOX_ID_no_neighs)),]
> t0 <- aggregate(b487, list(ids = b487$NOX_ID), head, n = 1)
> b93 <- t0[, c("ids", attr(t0, "sf_column"))]
```

Problem of no-neighbour zone

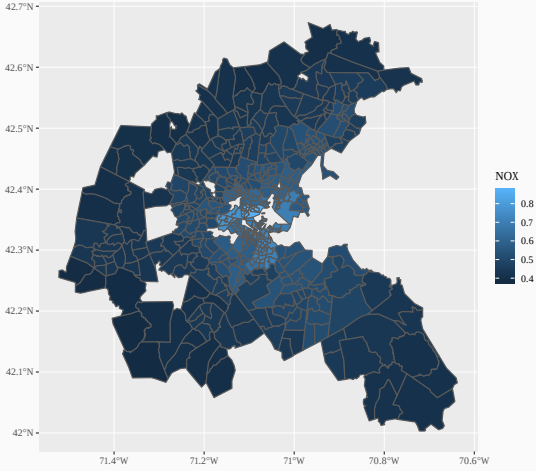
and finally create the same kind of **nb** object as used in **spdep**:

```
> qm_93 <- st_queen(b93)

## although coordinates are longitude/latitude, st_relate_pattern assumes that they are planar

> class(qm_93) <- "nb"
> attr(qm_93, "region.id") <- as.character(b93$ids)
```

NOX values (parts per 10 million)

The `geom_sf` function is in ggplot2:

```
> library(ggplot2)
> ggplot(b487) + geom_sf(aes(fill=NOX))
```



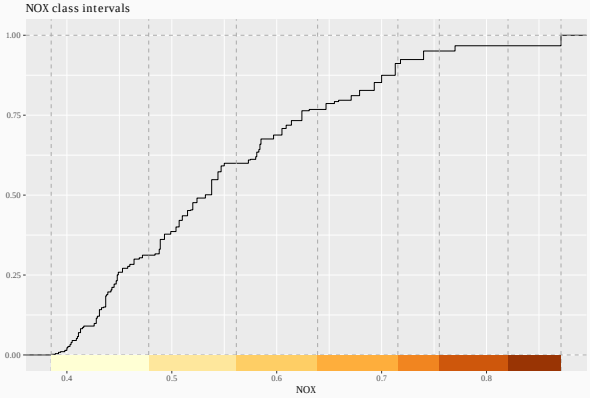
NOX values (parts per 10 million)

Let's try to use **classInt** to help present the NOX values by choosing natural breaks classes using `e1071::bclust`:

```
> library(classInt)

## Loading required package: spData

> set.seed(1)
> cI <- classIntervals(b487$NOX, n=7L, style="bclust", verbose=FALSE)
> cI

## style: bclust
##   one of 256,851,595 possible partitions of this variable into 7 classes
##  [0.385,0.478) [0.478,0.5615) [0.5615,0.639) [0.639,0.7155) [0.7155,0.755)
##             152             140              82              70              19
## [0.755,0.8205) [0.8205,0.871]
##              8              16
```

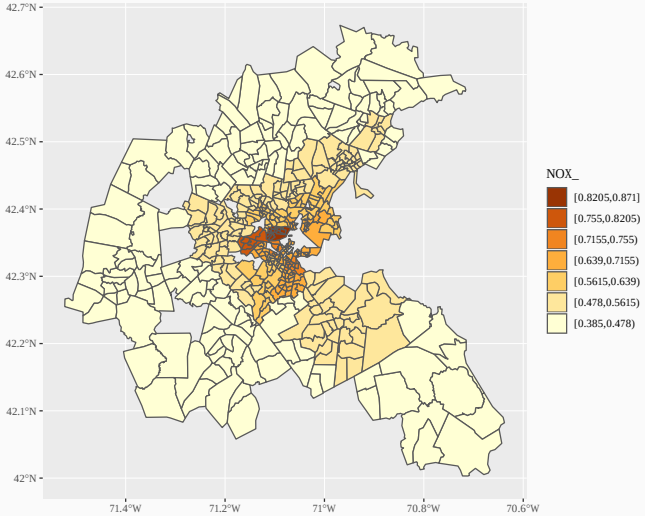
NOX values (parts per 10 million)

We can show the intervals on an empirical cumulative distribution function plot:



NOX values (parts per 10 million)

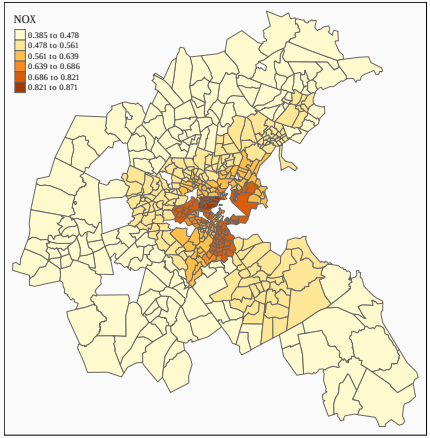
So it is possible to get to a ggplot rendering of an **sf** object:



NOX values (parts per 10 million)

Or use **tmap** which also supports **sf** objects:

```
> library(tmap)
> qtm(b487, fill="NOX", fill.n=7L, fill.style="bclust")
```



Air pollution data

- The figures show clearly that the study of the relationship between NOX and house value will be impacted by “copying out’’ NOX values to census tracts, as noted by Harrison and Rubinfeld (1978, p. 86, footnote 14)
- Even if we were to use more class intervals in these choropleth maps, the visual impression would be the same, because the underlying data have support approximated by the TASSIM zones, not by the census tracts

Other independent variables

- Besides NOX, the other census covariates included in the hedonic regression to account for median house values are the average number of rooms per house, the proportion of houses older than 1940, the proportion low-status inhabitants in each tract, and the Black proportion of population in the tract — originally expressed as a broken-stick relationship, but here taken as a percentage
- The crime rate is said to be taken from FBI data by town, but which is found on inspection to vary by tract
- The distance from tract to employment centres is derived from other sources, as is the dummy variable for tracts bordering Charles River

Other independent variables

- Other covariates are defined by town, with some also being fixed for all towns in Boston
- The town aggregates of census tracts are used in many of the census report tabulations, and of the 92 towns, 17 only contain one census tract, while one town contains thirty census tracts
- The variables are the proportion of residential lots zoned over 25000 sq. ft, the proportion of non-retail business acres, accessibility to radial highways, full-value property-tax rate per USD 10,000, and pupil-teacher ratio by town school district

Towns and TASSIM zones

- In the case of 80 approximate TASSIM zones aggregated from census tracts, the boundaries do coincide exactly with town boundaries
- For the remaining 12 towns and 16 TASSIM zones, there are overlaps between more than one town and TASSIM zone, mostly in Boston itself
- Using TASSIM zones for analysis should therefore also reduce the levels of autocorrelation induced by “copying out” town values to tracts within towns
- The exact match between town boundaries defined using census tracts, and approximated TASSIM zones also constructed using census tracts is not necessarily an indication that towns were used as TASSIM zones

Hedonic modelling of house values

- Pace and Gilley (1997), drawing on earlier work, felt that it should be worthwhile to check whether the original model was not spatially misspecified
- They considered that the use of spatial aggregate units as observations might involve spillovers of some kind, chiefly in the house values used
- Had the included explanatory variables accounted for the similarities between neighbours, there might not have been any reason to go further, but the residuals turn out to be spatially highly patterned
- So now we will turn to spatial econometrics methods to try to unravel the question of the "real" link between house values and NOX

The ZN, INDUS, NOX, RAD, TAX and PTRATIO variables show effectively no variability within the TASSIM zones, so in a multilevel model the random effect may absorb their influence. The model as a whole, before introducing random effects, is:

```
> form <- formula(log(median) ~ CRIM + ZN + INDUS + CHAS + I((NOX*10)^2) + I(RM^2) +
+   AGE + log(DIS) + log(RAD) + TAX + PTRATIO + I(BB/100) + log(I(LSTAT/100)))
```

Empirical approach

We will be using row-standardised contiguity neighbours derived from the map of census tracts, and from the map of merged census tracts constituting approximate TASSIM zones

```
> q487 <- st_queen(lwgeom::st_make_valid(b487))

## although coordinates are longitude/latitude, st_relate_pattern assumes that they are planar

> q487[which(sapply(q487, length) == 0)] <- 0L
> class(q487) <- "nb"
> suppressPackageStartupMessages(library(spdep))
> lw487 <- nb2listw(q487, zero.policy=TRUE)
```

Model comparisons

This is not completed as a proper set of comparisons with this data set - Bivand et al. (2017) contains comparisons for Beijing land parcels and a housing data set for SW Norway. The baseline tract level residual spatial autocorrelation is measured using a proper Moran test (b487 is an `sf` object):

```
> OLS <- lm(form, data=b487)
> lm.morantest(OLS, lw487, alternative="two.sided", zero.policy=TRUE)

##
##  Global Moran I for regression residuals
##
## data:
## model: lm(formula = form, data = b487)
## weights: lw487
##
## Moran I statistic standard deviate = 15.963, p-value < 2.2e-16
## alternative hypothesis: two.sided
## sample estimates:
## Observed Moran I      Expectation         Variance
##     0.4164854911    -0.0168501979     0.0007369131
```

Here we fit air pollution model output zone unstructured random effects using **lme4**, and the residual spatial autocorrelation falls (indicative not proper test):

```
> library(lme4)
> MLM <- lmer(update(form, . ~ . + (1 | NOX_ID)), data=b487, REML=FALSE)
> moran.test(residuals(MLM), lw487, alternative="two.sided", zero.policy=TRUE)

##
##  Moran I test under randomisation
##
## data:  residuals(MLM)
## weights: lw487  n reduced by no-neighbour observations
##
##
## Moran I statistic standard deviate = 2.082, p-value = 0.03734
## alternative hypothesis: two.sided
## sample estimates:
## Moran I statistic       Expectation          Variance
##       0.055773746      -0.002061856       0.000771654

> b93$MLM_re <- ranef(MLM)[[1]][,1]
```

Data setup

Both **hglm** and **HSAR** need sparse design matrices precomputed:

```
> library(Matrix)
> suppressMessages(library(MatrixModels))
> Delta <- as(model.Matrix(~ -1 + as.factor(NOX_ID), data=b487, sparse=TRUE), "dgCMatrix")
> M <- as(nb2listw(qm_93, style="B"), "CsparseMatrix")
```

Fitting the air pollution model output zone unstructured random effects using hierarchical GLM also works (indicative test):

```
> suppressPackageStartupMessages(library(hglm))
> y_hglm <- log(b487$median)
> X_hglm <- model.matrix(OLS)
> suppressWarnings(HGLM_iid <- hglm(y=y_hglm, X=X_hglm, Z=Delta))
> moran.test(HGLM_iid$resid, lw487, alternative="two.sided", zero.policy=TRUE)

##
##  Moran I test under randomisation
##
## data:  HGLM_iid$resid
## weights: lw487  n reduced by no-neighbour observations
##
##
## Moran I statistic standard deviate = 1.9335, p-value = 0.05318
## alternative hypothesis: two.sided
## sample estimates:
## Moran I statistic       Expectation          Variance
##      0.0516373810     -0.0020618557      0.0007713654
```

Using HGLM

As does a Simultaneous Autoregressive (SAR) fitted with **hglm**, modelling with air pollution model output zone spatially structured random effects:

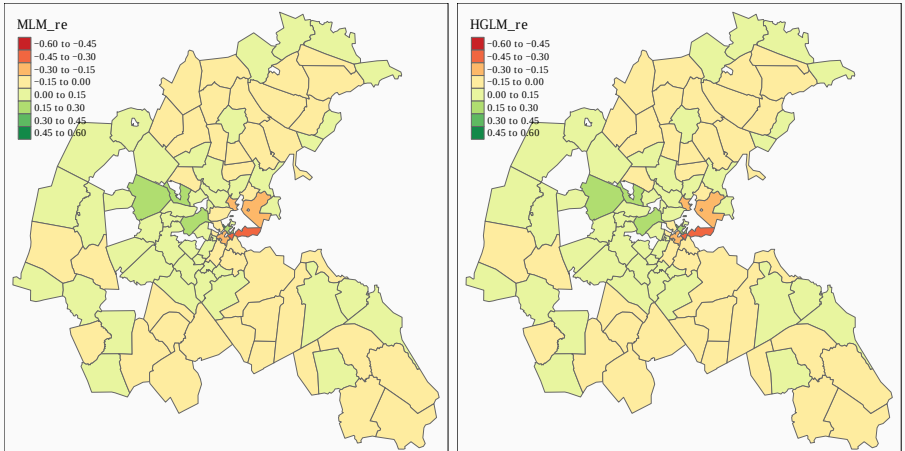
```
> suppressWarnings(HGLM_sar <- hglm(y=y_hglm, X=X_hglm, Z=Delta, rand.family=SAR(D=M)))
> moran.test(HGLM_sar$resid, lw487, alternative="two.sided", zero.policy=TRUE)

##
##  Moran I test under randomisation
##
## data:  HGLM_sar$resid
## weights: lw487  n reduced by no-neighbour observations
##
##
## Moran I statistic standard deviate = 1.7366, p-value = 0.08246
## alternative hypothesis: two.sided
## sample estimates:
## Moran I statistic       Expectation          Variance
##      0.0461576413     -0.0020618557      0.0007709856

> b93$HGLM_re <- unname(HGLM_iid$ranef)
> b93$HGLM_ss <- HGLM_sar$ranef[,1]
```

Unstructured random effects: TASSIM zones

```
> brks <- seq(-0.6, 0.6, 0.15)
> qtm(b93, fill=c("MLM_re", "HGLM_re"), fill.breaks=brks)
```

```
## Variable "MLM_re" contains positive and negative values, so midpoint is set to 0. Set midpoint = NA to show the full spectrum of the color palette.
```

```
## Variable "HGLM_re" contains positive and negative values, so midpoint is set to 0. Set midpoint = NA to show the full spectrum of the color palette.
```



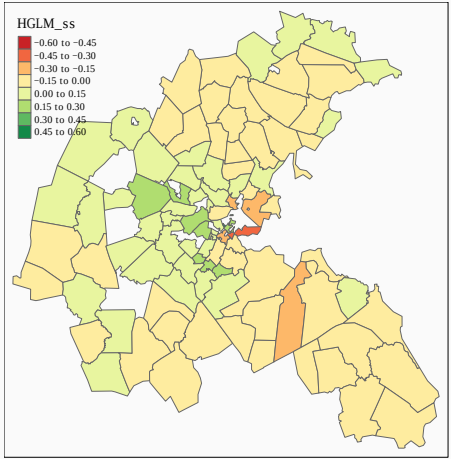
Using HSAR

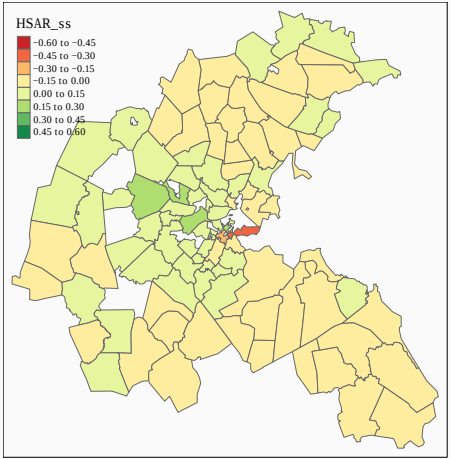
HSAR fits the model with air pollution model output zone spatially structured random effects using MCMC:

```
> library(HSAR)
> HSAR <- hsar(form, data=b487, W=NULL, M=M, Delta=Delta,
+              burnin=500, Nsim=5000, thinning=1)

> b93$HSAR_ss <- HSAR$Mus[1,]
```

Spatially structured random effects: TASSIM zones

```
## Variable "HGLM_ss" contains positive and negative values, so midpoint is set to 0

## Variable "HSAR_ss" contains positive and negative values, so midpoint is set to 0
```





Using R2BayesX

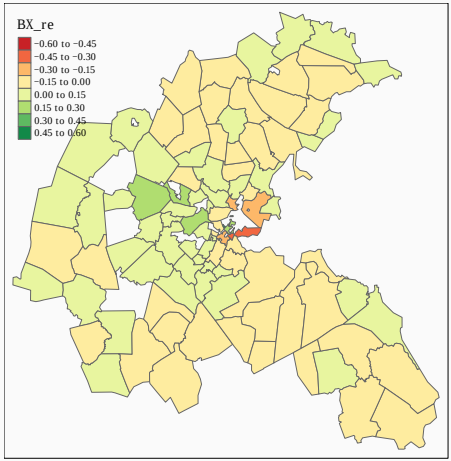
We can use **R2BayesX** for the same purposes, for both unstructured and spatially structured random effects by MCMC:

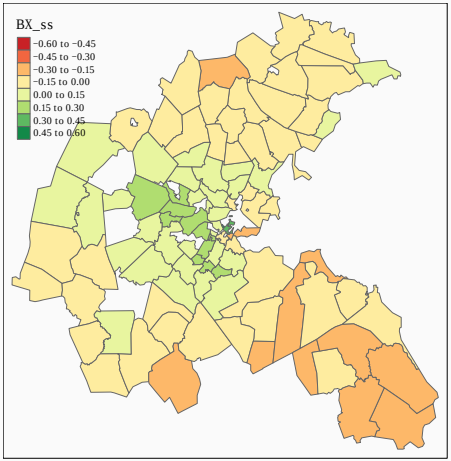
```
> suppressPackageStartupMessages(library(R2BayesX))
> BX_iid <- bayesx(update(form, . ~ . + sx(NOX_ID, bs="re")), family="gaussian",
+ data=b487, method="MCMC", iterations=12000, burnin=2000, step=2, seed=123)
> moran.test(residuals(BX_iid), lw487, alternative="two.sided", zero.policy=TRUE)

##
##  Moran I test under randomisation
##
## data:  residuals(BX_iid)
## weights: lw487  n reduced by no-neighbour observations
##
##
## Moran I statistic standard deviate = 1.9415, p-value = 0.05219
## alternative hypothesis: two.sided
## sample estimates:
## Moran I statistic       Expectation          Variance
##      0.0518678297     -0.0020618557      0.0007715581

> b93$BX_re <- BX_iid$effects["sx(NOX_ID):re"][[1]]$Mean
```

The spatially structured random effects are modelled as an intrinsic Conditional Autoregressive (CAR), not a SAR with a parameter, but turn out very similar:

```
> RBX_gra <- nb2gra(qm_93)
> BX_mrf <- bayesx(update(form, . ~ . + sx(NOX_ID, bs="mrf", map=RBX_gra)),
+ family="gaussian", data=b487, method="MCMC", iterations=12000, burnin=2000,
+ step=2, seed=123)
> moran.test(residuals(BX_mrf), lw487, alternative="two.sided", zero.policy=TRUE)

##
##  Moran I test under randomisation
##
## data:  residuals(BX_mrf)
## weights: lw487  n reduced by no-neighbour observations
##
##
## Moran I statistic standard deviate = 1.8655, p-value = 0.0621
## alternative hypothesis: two.sided
## sample estimates:
## Moran I statistic       Expectation          Variance
##       0.049750225      -0.002061856       0.000771345

> b93$BX_ss <- BX_mrf$effects["sx(NOX_ID):mrf"][[1]]$Mean
```

```
## Variable "BX_re" contains positive and negative values, so midpoint is set to 0.

## Variable "BX_ss" contains positive and negative values, so midpoint is set to 0.
```





BYM with BayesX

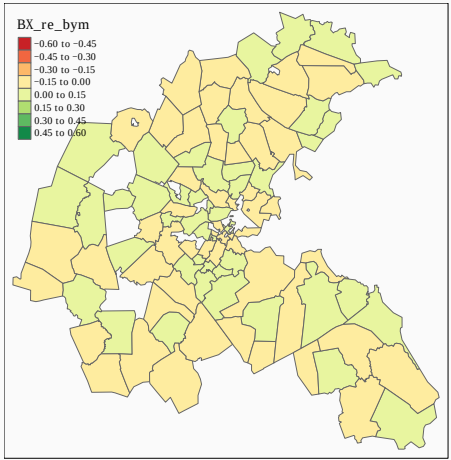
Combining the CAR and unstructured random effects in one model is the Besag, York, Mollie (BYM) model:

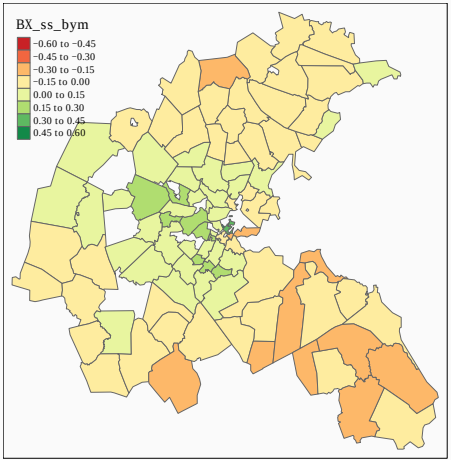
```
> b487$d_id <- as.integer(as.factor(b487$NOX_ID))
> BX_bym <- bayesx(update(form, . ~ . + sx(NOX_ID, bs="mrf", map=RBX_gra) +
+ sx(d_id, bs="re")), family="gaussian", data=b487, method="MCMC",
+ iterations=12000, burnin=2000, step=2, seed=123)
> moran.test(residuals(BX_bym), lw487, alternative="two.sided", zero.policy=TRUE)

##
##  Moran I test under randomisation
##
## data:  residuals(BX_bym)
## weights: lw487  n reduced by no-neighbour observations
##
##
## Moran I statistic standard deviate = 1.8228, p-value = 0.06833
## alternative hypothesis: two.sided
## sample estimates:
## Moran I statistic       Expectation          Variance
##      0.0485655037     -0.0020618557      0.0007714108

> b93$BX_ss_bym <- BX_bym$effects["sx(NOX_ID):mrf"][[1]]$Mean
> b93$BX_re_bym <- BX_bym$effects["sx(d_id):re"][[1]]$Mean
```

BayesX BYM random effects

```
## Variable ”BX_re_bym” contains positive and negative values, so midpoint is set to
```

```
## Variable ”BX_ss_bym” contains positive and negative values, so midpoint is set to
```



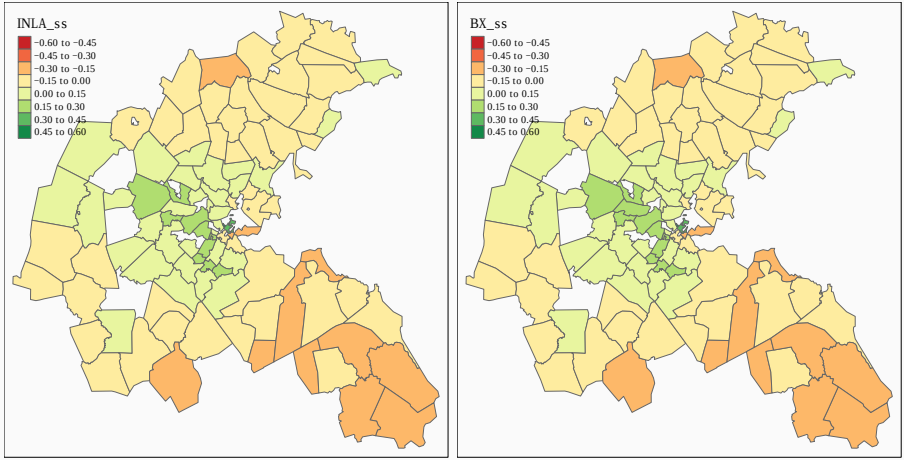


Finally INLA

The same models with intrinsic CAR can be estimated using INLA (and WinBUGS, not shown here):

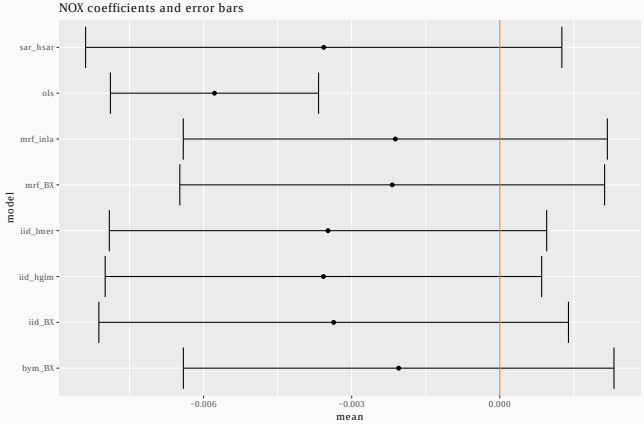
```
> suppressPackageStartupMessages(library(INLA))
> tf <- tempfile()
> nb2INLA(qm_93, file=tf)
> INLA_mrf <- inla(update(form, . ~ . + f(d_id, model="besag", graph=tf, param=c(1, 0.01))),
+ data=b487, control.fixed = list(prec.intercept = 0.001, prec = 0.001),
+ control.compute = list(dic = TRUE, waic = TRUE))
> moran.test((log(b487$median) - INLA_mrf$summary.fitted.values[,1]),
+ lw487, alternative="two.sided", zero.policy=TRUE)

##
##  Moran I test under randomisation
##
## data:  (log(b487$median) - INLA_mrf$summary.fitted.values[, 1])
## weights: lw487  n reduced by no-neighbour observations
##
##
## Moran I statistic standard deviate = 1.9365, p-value = 0.0528
## alternative hypothesis: two.sided
## sample estimates:
## Moran I statistic       Expectation          Variance
##      0.0517187103     -0.0020618557      0.0007712764

> b93$INLA_ss <- INLA_mrf$summary.random$d_id$mean
```

```
## Variable "INLA_ss" contains positive and negative values, so midpoint is set to 0

## Variable "BX_ss" contains positive and negative values, so midpoint is set to 0.
```



NOX coefficients

Looking at the NOX coefficients alone is disappointing, but recall that the upper level random effects do apply to NOX too so we could map their standard errors of which some are small compared to the effects:



Final remarks

- We've been using **sf** data.frame objects directly in modelling, as we could (most often) use **sp** objects
- **sf** objects are supported by **tmap**, **mapview**, and **ggplot2**
- There is no link between **sf** and **raster** (yet), and tidy arrays are only under discussion AFAIK
- Spatial multilevel models may match data generation processes when the spatial processes have different footprints