Summer at Census - Spatial Analysis Workshop

Summer at Census Research Seminar

Corey S. Sparks, Ph.D.

July 11, 2022

Spatial analysis workshop

This short workshop will cover fundamentals of spatial data analysis, spatial clustering and the use of regression models that explicitly model spatial correlation in rates. I will use the Census Low Response Score as an example outcome and introduce the use of the Bayesian modeling package INLA, along with other R functions for spatial analysis. All materials will be provided on Github at the end of the workshop.

Data and code for all examples can be found at:

- Code:(https://github.com/coreysparks/summer_census)
- Data:(https://github.com/coreysparks/data)

Additional content on spatial analysis in demography from my formal courses can be found here:

- GIS for population science course
- Spatial demography course

I also recommend using GeoDa as a tool to explore spatial data. It can be found on the Github page for the project.

Exploratory Spatial Data Analysis (ESDA)

Skepticism and openness

In exploratory spatial data analysis (ESDA), we are looking for trends and patterns in data. We are working under the assumption that the more one knows about the data, the more effectively it may be used to develop, test and refine theory.

This generally requires we follow two principles:

Skepticism: One should be skeptical of measures that summarize data, since they can conceal or misrepresent the most informative aspects of data.

Openness: We must be open to patterns in the data that we did not expect to find, because these can often be the most revealing outcomes of the analysis.

Global and Local Statistics

By global we imply that one statistic is used to adequately summarize the data, i.e. the mean or median Or, a regression model that is suitable for all areas in the data Local statistics are useful when the process you are studying *varies over space*, i.e. different areas have different local values that might cluster together to form a local deviation from the overall mean Or a regression model that accounts for variation in the variables over space. This we often call *spatial heterogeneity*, meaning the process that generates the data varies over space. This is also called a *spatial regime*.

Stationarity

Stationarity simply means that the process is not changing with respect to either time (i.e. time series analysis) or space.

This implies that the process that has generated our data is acting the same way in all areas under study.

The implications of Stationarity are that we can use a global statistic to measure our process and not feel too bad about it. It also implies that our observations are iid (independent and identically distributed) with respect to one another e.g. the parameters estimated by the regression of X on Y are the same throughout our area of study, and do not have a tendency to change. Also, it means the model estimated is equally well specified at all locations. *This is our general assumption in regression models*

Non-Stationarity

If a process is *non-stationary* then the process changes with respect to time or space.

This implies that the process that has generated our data is not acting the same way in all areas, or the expected value (mean, or variance) of our data are subject to spatial fluctuations.

If our data are subject to such fluctuations, the this implies that our global statistics are also subject to major local fluctuations Meaning areas in our data can tend to cluster together and have similar values

Autocorrelation

This can occur in either space or time Really boils down to the non-independence between neighboring values The values of our independent variable (or our dependent variables) may be similar because Our values occur closely in time (temporal autocorrelation) closely in space (spatial autocorrelation)

Assessment of Spatial Dependency

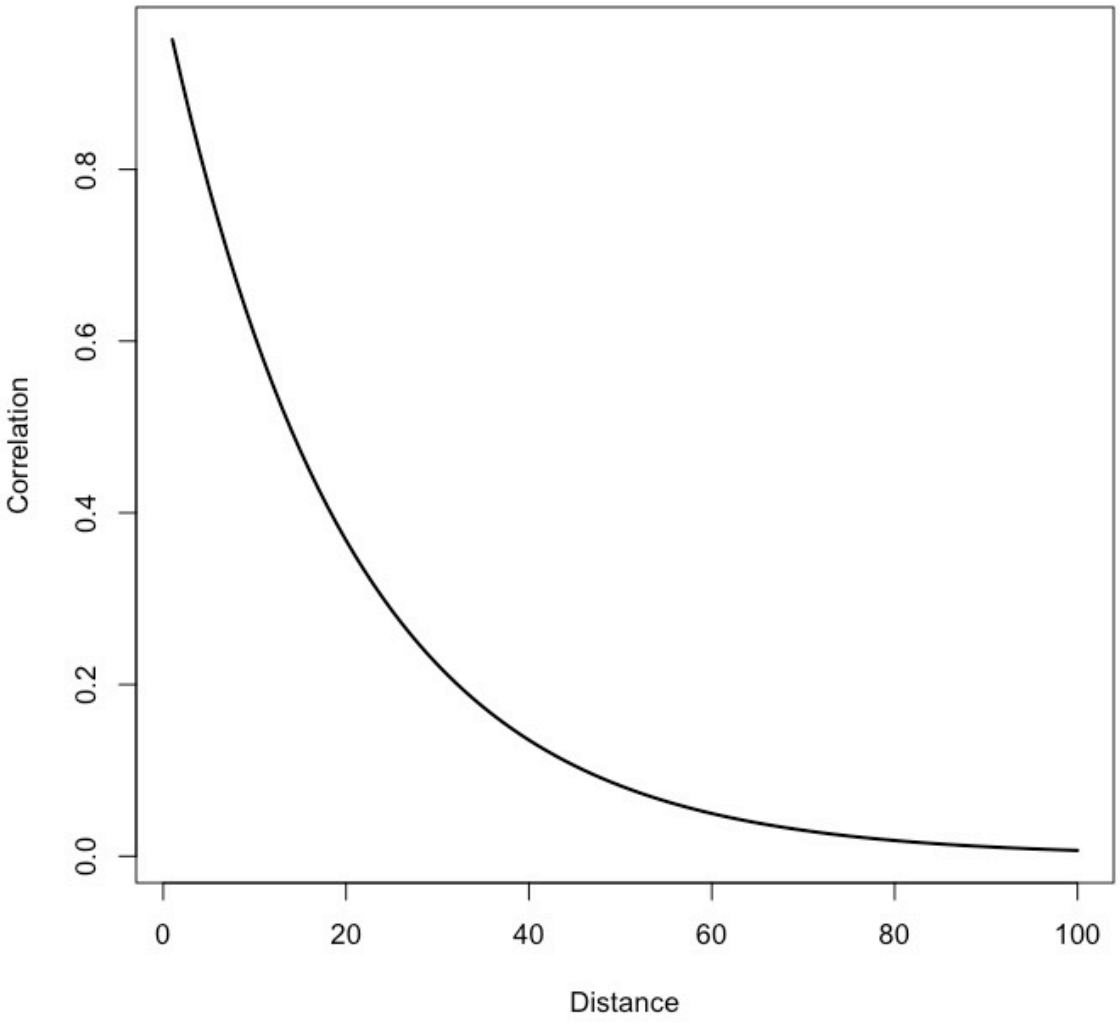
Before we can explore the dependency in spatial data, we must first cover the ideas of creating and modeling neighborhoods in our data. By neighborhoods, we are referring to the clustering or connectedness of observations.

The exploratory methods we will cover in this workshop depend on us knowing how our data are arranged in space, who is next to who. This is important (as we will see later) because most correlation in spatial data tends to die out as we get further away from a specific location.

Tobler's law

Waldo Tobler (Tobler 1970) suggested the first law of geography: - Everything is related to everything else, but near things are more related than distant things.

We can see this better in graphical form: We expect the correlation between the attributes of two points to diminish as the distance between them grows.



An example of this type of phenomena is how rich and poor neighborhoods tend to cluster around one another.

So our statistics that correct for, or measure spatial association, we have to account for where observations are with respect to each other. This is typically done by specifying/identifying the spatial connectivity between observations.

Spatial connectivity is defined based on the interactions/associations between features in our data.

This connectivity is often in terms of the spatial weight of an observation, in other words how much of the value of a surrounding observation do we consider when we are looking at spatial correlation.

Typically the weight of a neighboring observation decreases the further it is away from our feature of interest.

There are two typical ways in which we measure spatial relationships - distance and polygon contiguity

Distance based association

In a distance based connectivity method, features (generally points) are considered to be contiguous if they are within a given radius of another point. The radius is really left up to the researcher to decide. For example we did this in the point analysis lab, where we selected roads within a mile of hospitals. We can equally do it to search for other hospitals within a given radius of every other hospital. The would then be labeled as neighbors according to our radius rule. Likewise, we can calculate the distance matrix between a set of points This is usually measured using the standard Euclidean distance

$d^2 = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

Where x and y are coordinates of the point or polygon in question (selected features), this is the as the crow flies distance. *There are lots of distances*

Spatial Neighbors

There are many different criteria for deciding if two observations are neighbors Generally two observations must be within a critical distance, d, to be considered neighbors. This is the Minimum distance criteria, and is very popular. This will generate a matrix of binary indicators describing the neighborhood. We can also describe the neighborhoods in a continuous weighting scheme based on the distance between them

Inverse Distance Weight

$w_{ij} = \frac{1}{d_{ij}}$

or *Inverse-Squared Distance Weight*

$w_{ij} = \frac{1}{d_{ij}^2}$

K nearest neighbors A useful way to use distances is to construct a k-nearest neighbors set. This will find the "k" closest observations for each observation, where k is some integer.

For instance if we find the k=3 nearest neighbors, then each observation will have 3 neighbors, which are the closest observations to it, *regardless of the distance between them* which is important.

Using the k nearest neighbor rule, two observations could potentially be very far apart and still be considered neighbors.

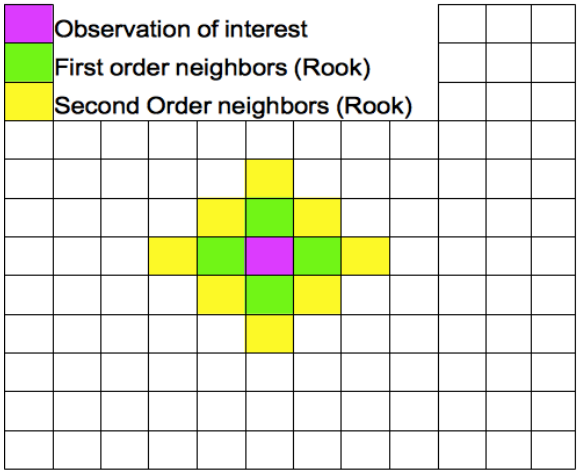
If the a feature is within the *threshold distance*, a 1 is given, otherwise a 0 is given. Neighborhoods are created based on which observations are judged contiguous.

Polygon Adjacency weights

This is generally the best way to treat polygon features Polygons are contiguous if they share common topology, like an edge (line segment) or a vertex (point). Think like a chess board: *Rook adjacency* Neighbors must share a line segment

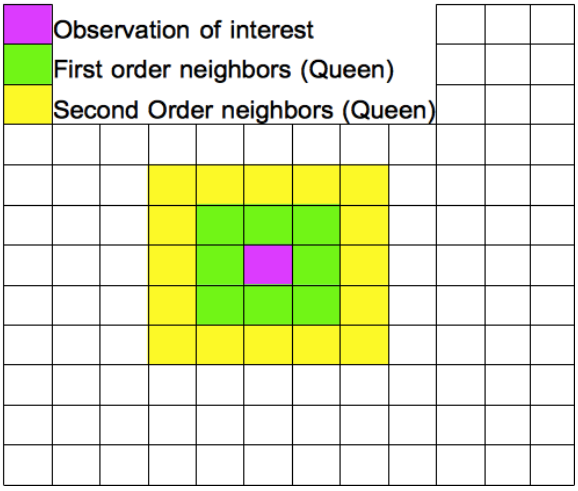
Queen adjacency Neighbors must share a vertex or a line segment If polygons share these boundaries (based on the specific definition: rook or queen), they are given a weight of 1 (adjacent), else they are given a value 0, (nonadjacent)

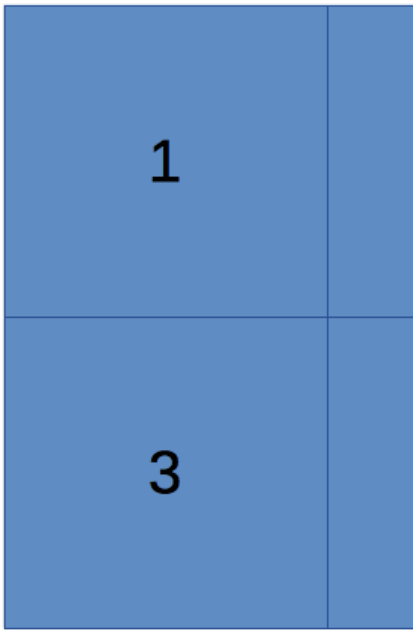
Order of adjacency



First and second order
Rook adjacency

First and second order
Queen adjacency





What does a spatial weight matrix look like? Assume this is our data:

This would be a *Rook-based* adjacency weight matrix:

$$w_{ij} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

$w_{ij} = 1$ if polygons share a border, 0 if they don't. Also note that an observation can't be it's own neighbor and the diagonal elements of the matrix are all 0.

Measuring Spatial Autocorrelation

If we observe data $Z(s)$ (an attribute) at location i, and again at location j, the spatial autocorrelation between $Z(s)_i$ and $Z(s)_j$ is degree of similarity between them, measured as the standardized covariance between their locations and values.

In the absence of spatial autocorrelation the locations of $Z(s)_i$ and $Z(s)_j$ has nothing to do with the values of $Z(s)_i$ and $Z(s)_j$ although, if autocorrelation is present, close proximity of $Z(s)_i$ and $Z(s)_j$ leads to correlation of their attributes.

Positive autocorrelation Positive autocorrelation means that a feature is positively associated with the values of the surrounding area (as defined by the spatial weight matrix), high values occur with high values, and low with low. For example, poverty rates typically cluster together, meaning you have poor neighborhoods next to other poor neighborhoods. This results in positive auto-correlation. Similarly, you typically see that affluent neighborhoods are geographically close to other affluent neighborhoods.

Both of these are examples of positive auto-correlation.

Negative autocorrelation Negative autocorrelation means that a feature is negatively associated with the values of the surrounding area, high with low, low with high. Negative auto-correlation is rarer to see, but an example could be when neighborhoods are gentrifying. When gentrification occurs, you typically have one area where new home construction or new business construction happens, but it it typically surrounded by areas where development is lacking. This would then create a neighborhood of relative affluence among other neighborhoods which are struggling. This is an example of negative auto-correlation.

Spatial lags of a variable are done via multiplying the variable through the spatial weight matrix for the data.

If we have a value $Z(s_i)$ at location i and a spatial weight matrix w_{ij} describing the spatial neighborhood around location i, we can find the lagged value of the variable by: $WZ_i = Z(s_i) * w_{ij}$

This calculates what is effectively, the neighborhood average value in locations around location i, often stated $Z(s_{-i})$

Let's return to the adjacency matrix from above, a *Rook-based* adjacency weight matrix.

$$w_{ij} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Typically this matrix is standardized, by dividing each element of w_{ij} by the number of neighbors, this is called the *row-standardized* form of the matrix. In this case, there are two neighbors, so row-standardization results in the following matrix:

$$w_{ij} = \begin{bmatrix} 0 & .5 & .5 & 0 \\ .5 & 0 & 0 & .5 \\ .5 & 0 & 0 & .5 \\ 0 & .5 & .5 & 0 \end{bmatrix}$$

Let's take a variable z, equal to:

$$z = \begin{bmatrix} 1 & 5 & 10 & 2 \end{bmatrix}$$

When we form the product: $z'W$, we get:

$$z_{lag} = \begin{bmatrix} 7.5 & 1.5 & 1.5 & 7.5 \end{bmatrix}$$

In R, we can calculate the spatially lagged value of z. See the code below.

```
z<-c(1,5,10,2)
w<-matrix(c(0,.5,.5,0,.5,0,0,.5,.5,0,0,.5,0,.5,.5,0), nrow = 4, byrow = T)
z
```

```
[1]  1  5 10  2
```

```
w
```

```
     [,1] [,2] [,3] [,4]
[1,]  0.0  0.5  0.5  0.0
[2,]  0.5  0.0  0.0  0.5
[3,]  0.5  0.0  0.0  0.5
[4,]  0.0  0.5  0.5  0.0
```

```
z%*%w
```

```
     [,1] [,2] [,3] [,4]
[1,]  7.5  1.5  1.5  7.5
```

Measureing spatial autocorrelation

Moran's I Statistic

One of the most popular global auto-correlation statistic is Moran's I (Moran 1950)

$I = \frac{n}{(n-1)\sigma^2 w_{..}} \sum_n^i \sum_n^j w_{ij}(Z(s_i) - \bar{Z})(Z(s_j) - \bar{Z})$

with:

-$Z(s_i)$ being the value of the variable, the variable for example, at location i

-$Z(s_j)$ being the value of the variable at location j,

-σ^2 is sample variance of the variable

-w_{ij} is the weight for location ij (0 if they are not neighbors, 1 otherwise).

Moran's I is basically a correlation, *think of a Pearson correlation ρ, between a variable and a spatially lagged version of itself.*

Moran's I Scatter plot It is sometimes useful to visualize the relationship between the actual values of a variable and its spatially lagged values. This is the so called **Moran scatter plot**

Lagged values are the average value of the surrounding neighborhood around location i

lag(Z) $= z_{ij} * w_{ij} = z'W$ in matrix terms

Which, now we see where we get the *y-value* of the Moran scatter plot. It is just the lagged version of the original variable.

The Moran scatter plot shows the association between the value of a variable in a location and its spatial neighborhood's average value. The variables are generally plotted as *z-scores*,to avoid scaling issues.

And here we show the Moran scatter plot. This plot shows the z-scored values of the original variable on the x axis and the spatially lagged values of the original variable on the y axis. In this case, we see a weak positive relationship to the values. This positive relationship indicates positive auto-correlation in this variable.

Moran Scatter plot for San Antonio Low Response Rate

```
library(tidyverse)
```

```
-- Attaching packages --------------------------------------- tidyverse 1.3.2 --
v ggplot2 3.3.6     v purrr   0.3.4
v tibble  3.1.8     v dplyr   1.0.9
v tidyr   1.2.0     v stringr 1.4.0
v readr   2.1.2     v forcats 0.5.1
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

```
tx_lrs <- sf::st_read("../data/tx_lrs.gpkg")
```

```
Reading layer `tx_lrs' from data source
  `C:\Users\ozd504\Github\summer_census\data\tx_lrs.gpkg' using driver `GPKG'
Simple feature collection with 5254 features and 44 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: -631836.8 ymin: -2112744 xmax: 623553.1 ymax: -938155.6
Projected CRS: NAD27 / US National Atlas Equal Area
```

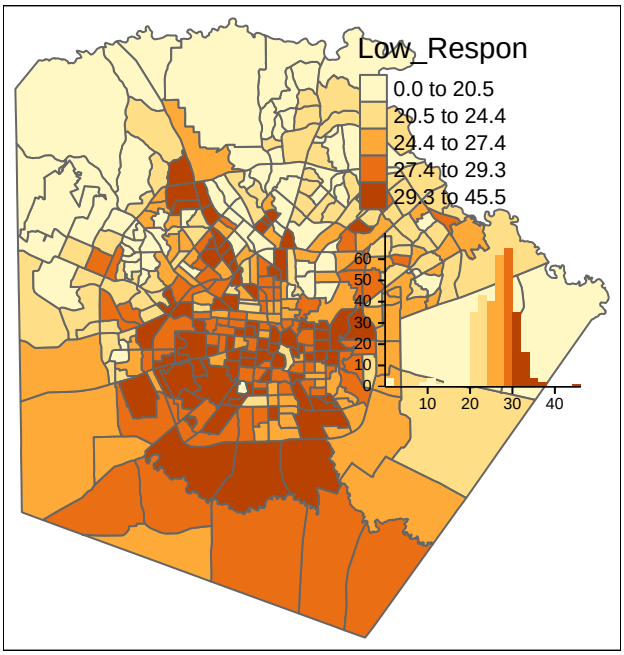
```
sa_lrs <- tx_lrs %>%
  filter(COUNTYFP == "029")
```

Here is the overall low response rate map for San Antonio.

```
library(tmap)

tmap_mode("plot")
```

```
tmap mode set to plotting
```

```
tm_shape(sa_lrs)+
  tm_polygons("Low_Respon",
               style="quantile",
              n=5,
              legend.hist = TRUE) +
  tm_basemap(server="OpenStreetMap",alpha=0.5)
```



Create spatial adjacency information

```
library(sfdep)

#Make a rook style weight matrix
sanb <- sa_lrs %>%
  mutate(nb = st_contiguity(geom, queen = F),
         wt = st_weights(nb),
         .before = 1)
```

```
global_moran_perm(sanb$Low_Respon,
              nb = sanb$nb,
              wt = sanb$wt,
              nsim = 1000)
```

```
    Monte-Carlo simulation of Moran I

data:  x
weights: listw
number of simulations + 1: 1001
```

```
statistic = 0.46805, observed rank = 1001, p-value < 2.2e-16
alternative hypothesis: two.sided
```

Moran's I is basically a correlation think Pearson's ρ on a variable and a *spatially lagged* version of itself. Spatial lags of a variable are done via multiplying the variable through the spatial weight matrix for the data.

Local Moran's I

So far, we have only seen a *Global* statistic for auto-correlation, and this tells us if there is any *overall* clustering in our data. We may be more interested in precisely *where* within the data the auto-correlation occurs, or where *clusters* are located.

A local version of the Moran statistic is available as well. This basically calculates the Moran I statistic from above, but only for each observation's *local neighborhood*.

It compares the observation's value to the local neighborhood average, instead of the global average. Luc Anselin (Anselin 2010) referred to this as a "**LISA**" statistic, for *Local Indicator of Spatial Autocorrelation*.

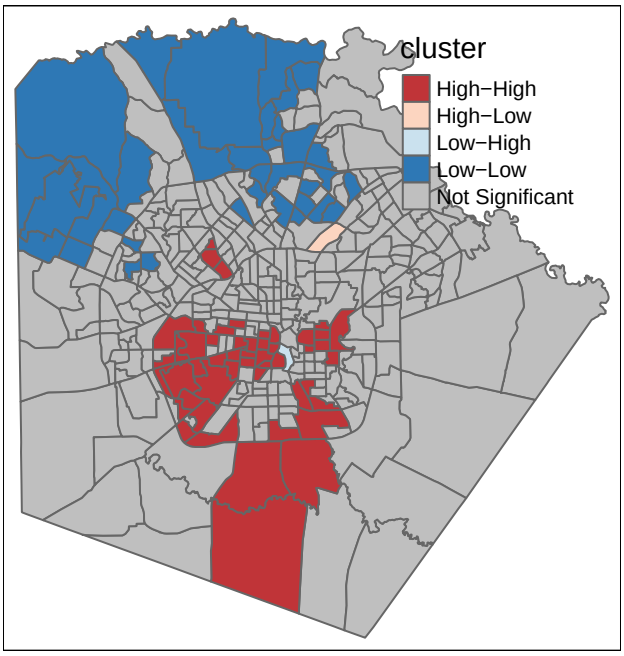
Here is a LISA map for clusters of the low response score, which shows areas of concentrated (clustered) high LRS clustering in red, and concentrated low values of the LRS in blue.

```
sa_locali <- sa_lrs%>%
  transmute(lisa = local_moran(Low_Respon,
                               nb = sanb$nb,
                               wt = sanb$wt,
                               nsim = 1000))%>%
              tidyr::unnest(lisa)

sa_locali <- sa_locali %>%
  mutate(cluster = ifelse(p_ii_sim < .05,
                          as.character(mean),
                          NA))
```

```
tmap_mode("plot")
```

```
tmap mode set to plotting
```

```
tm_shape(sa_locali)+
  tm_polygons("cluster",
              palette = "RdBu",
              textNA = "Not Significant")+
  tm_basemap(server="OpenStreetMap",alpha=0.5)
```



The red and blue areas are so-called *clusters*, because they are areas with higher (or lower, for the blues) than average LRS values, surrounded by areas that also have higher than average LRS values. The red clusters are so called "*high-high clusters*", likewise the blue areas are called "*low-low clusters*".

We also see in the legend that light pink and light blue values are possible. The light pink polygons represent areas that have high values of LRS, but are in a LRS spatial neighborhood, and are called high-low *outliers*. These are rare in this example.

Likewise, possible values include light blue polygons, these are called low-high outliers, because they have low LRS scores, but are in a high LRS spatial neighborhood. These are also rare in this example.

There are of course other autocorrelation statistics. For instance Geary's C

Geary's C

- RC Geary in 1954 derived the C statistic
- $C = \frac{n-1}{2\sum_{ij} w_{ij}} \frac{\sum_{ij} w_{ij}(x_i - x_j)^2}{\sum_{ij}(x_i - \bar{x})^2}$
- Similar in interpretation to the Moran statistic, C, measures whether values are similar in neighboring areas.
- C == 1 == No autocorrelation, C< 1 == positive autocorrelation, C > 1 negative autocorrelation

Getis-Ord G

- `"{Too Ugly to Show}"` See the paper
- Similar to Geary's C in interpretation
- High values next to high values, and so on

What these methods tell you?

- Moran's I is a *descriptive statistic ONLY*,
- It simply indicates if there is spatial association/autocorrelation in a variable
- Local Moran's I tells you if there is significant localized clustering of the variable

- Break -

Regression modeling for spatial data using INLA

The INLA Approach to Bayesian models

The Integrated Nested Laplace Approximation, or INLA, approach is a recently developed, computationally simpler method for fitting Bayesian models [(Rue et al., 2009, compared to traditional Markov Chain Monte Carlo (MCMC) approaches. INLA fits models that are classified as latent Gaussian models, which are applicable in many settings (Martino & Rue, 2010. In general, INLA fits a general form of additive models such as:

$\eta = \alpha + \sum_{j=1}^{nf} f^{(j)}(u_{ij}) + \sum_{k=1}^{n\beta} \beta_k z_{ki} + \epsilon_i$

where η is the linear predictor for a generalized linear model formula, and is composed of a linear function of some variables u, β are the effects of covariates, z, and ϵ is an unstructured

residual (Rue et al., 2009). As this model is often parameterized as a Bayesian one, we are interested in the posterior marginal distributions of all the model parameters. Rue and Martino (2007) show that the posterior marginal for the random effects (x) in such models can be approximated as:

$\tilde{p}(x_i|y) = \sum_k \tilde{p}(x_i|\theta_k, y)\tilde{p}(\theta_k|y)\Delta_k$

via numerical integration (Rue & Martino, 2007; Schrodle & Held, 2011a, 2011b). The posterior distribution of the hyperparameters (θ) of the model can also be approximated as:

$\tilde{p}(\theta|y)) \propto \frac{p(x,\theta,y)}{\tilde{p}G(x|\theta,y)} \mid_x = x^*(\theta)$

, where G is a Gaussian approximation of the posterior and $x^*(\theta)$ is the mode of the conditional distribution of $p(x|\theta, y)$. Thus, instead of using MCMC to find an iterative, sampling-based estimate of the posterior, it is arrived at numerically. This method of fitting the spatial models specified above has been presented by numerous authors (Blangiardo & Cameletti, 2015; Blangiardo et al., 2013; Lindgren & Rue, 2015; Martins et al., 2013; Schrodle & Held, 2011a, 2011b), with comparable results to MCMC.

Libraries

You need to install INLA, if you're using R >= 4.1, you install via

```
install.packages("INLA",repos=c(getOption("repos"),INLA="https://inla.r-inla-download.org/R/s
dep=TRUE)
```

```
#library(rgdal)
library(spdep)
library(INLA)
library(tigris)
library(tidycensus)
library(tidyverse)
```

Data

I have the data on my github site under the nhgis0029_csv page. These are data from the NHGIS project by IPUMS who started providing birth and death data from the US Vital statistics program.

The data we will use here are infant mortality rates in US counties between 2001 and 2007.

```
files<-list.files("C:/Users/ozd504/Github//data/nhgis0029_csv/",
                  pattern = "*.csv",
```

```
                    full.names = T)
df<-read_csv(files[16])
df$cofips<-paste(substr(df$GISJOIN, 2,3),
                 substr(df$GISJOIN, 5,7),
                 sep="")

df<-df%>%
  mutate(deaths=as.numeric(AGWI001),
         births = as.numeric(AGWE001))%>%
  arrange(cofips)%>%
  select(cofips,deaths,births)

head(df)
```

```
# A tibble: 6 x 3
  cofips deaths births
  <chr>   <dbl>  <dbl>
1 01001       5    642
2 01003      13   1870
3 01005       0    412
4 01007       2    287
5 01009       5    665
6 01011       2    166
```

Get census data using tidycensus

Here I get data from the 2000 decennial census summary file 3

```
#v00<-load_variables(year=2000, dataset = "sf3", cache = T)
cov_dat<-get_decennial(geography = "county",
                       year = 2000,
                       sumfile = "sf3",
                       summary_var = "P001001",
                       variables = c("P007003", "P007004","P007010","P053001", "P089001",
                      output = "wide")
```

```
Getting data from the 2000 decennial Census

Using Census Summary File 3
```

```r
cov_dat<-cov_dat%>%
  mutate(cofips=GEOID,
         pwhite=P007003/summary_value,
         pblack=P007004/summary_value,
         phisp=P007010/summary_value,
         medhhinc=as.numeric(scale(P053001)),
         ppov=P089002/P089001)


final.dat<-merge(df, cov_dat, by="cofips")
head(final.dat)
```

```
  cofips deaths births GEOID                    NAME P007003 P007004 P007010
1  01001      5    642 01001 Autauga County, Alabama   34760    7450     394
2  01003     13   1870 01003 Baldwin County, Alabama  120916   14134    2341
3  01005      0    412 01005 Barbour County, Alabama   14788   13399     509
4  01007      2    287 01007    Bibb County, Alabama   15867    4602     181
5  01009      5    665 01009  Blount County, Alabama   47135     555    2629
6  01011      2    166 01011 Bullock County, Alabama    2795    8443     359
  P053001 P089001 P089002 summary_value    pwhite     pblack       phisp
1   42013   43377    4738         43671 0.7959515 0.17059376 0.009022005
2   40250  138148   14018        140415 0.8611331 0.10065876 0.016672008
3   25101   26239    7032         29038 0.5092637 0.46142985 0.017528755
4   31420   19852    4091         20826 0.7618842 0.22097378 0.008691059
5   35241   50499    5930         51024 0.9237810 0.01087723 0.051524773
6   20605   10169    3405         11714 0.2386034 0.72076148 0.030647089
     medhhinc      ppov
1  0.75934595 0.1092284
2  0.57291445 0.1014709
3 -1.02904290 0.2679980
4 -0.36082926 0.2060750
5  0.04322903 0.1174281
6 -1.50448024 0.3348412
```

Create expected numbers of cases

In count data models, and spatial epidemiology, we have to express the raw counts of events relative to some expected value, or population offset, see this Rpub for a reminder.

```r
final.dat$E_d<-final.dat$births * (sum(final.dat$deaths)/sum(final.dat$births))

final.dat <- final.dat[order(final.dat$cofips),]
final.dat$id <- 1:dim(final.dat)[1]

head(final.dat)
```

```
  cofips deaths births GEOID                     NAME P007003 P007004 P007010
1  01001      5    642 01001 Autauga County, Alabama   34760    7450     394
2  01003     13   1870 01003 Baldwin County, Alabama  120916   14134    2341
3  01005      0    412 01005 Barbour County, Alabama   14788   13399     509
4  01007      2    287 01007    Bibb County, Alabama   15867    4602     181
5  01009      5    665 01009  Blount County, Alabama   47135     555    2629
6  01011      2    166 01011 Bullock County, Alabama    2795    8443     359
  P053001 P089001 P089002 summary_value    pwhite     pblack       phisp
1   42013   43377    4738         43671 0.7959515 0.17059376 0.009022005
2   40250  138148   14018        140415 0.8611331 0.10065876 0.016672008
3   25101   26239    7032         29038 0.5092637 0.46142985 0.017528755
4   31420   19852    4091         20826 0.7618842 0.22097378 0.008691059
5   35241   50499    5930         51024 0.9237810 0.01087723 0.051524773
6   20605   10169    3405         11714 0.2386034 0.72076148 0.030647089
     medhhinc       ppov       E_d id
1  0.75934595 0.1092284  4.396358  1
2  0.57291445 0.1014709 12.805591  2
3 -1.02904290 0.2679980  2.821339  3
4 -0.36082926 0.2060750  1.965350  4
5  0.04322903 0.1174281  4.553860  5
6 -1.50448024 0.3348412  1.136753  6
```

```r
options(scipen=999)
```

Next we make the spatial information, we get the polygons from census directly using `counties` from the `tigris` package. We drop counties not in the contiguous 48 US states.

```r
library(tigris)
us_co<-counties( cb = T)
```

```
Retrieving data for the year 2020
```

```
us_co<-us_co%>%
  subset(!STATEFP%in%c("02", "15", "60", "66", "69", "72", "78"))
```

Construction of spatial relationships among counties

Key part here is to make numeric identifier for each geography!

```
#In INLA, we don't need FIPS codes, we need a simple numeric index for our counties
us_co$struct<-1:dim(us_co)[1]

nbs<-knearneigh(st_centroid(us_co), k = 5, longlat = T) #k=5 nearest neighbors
```

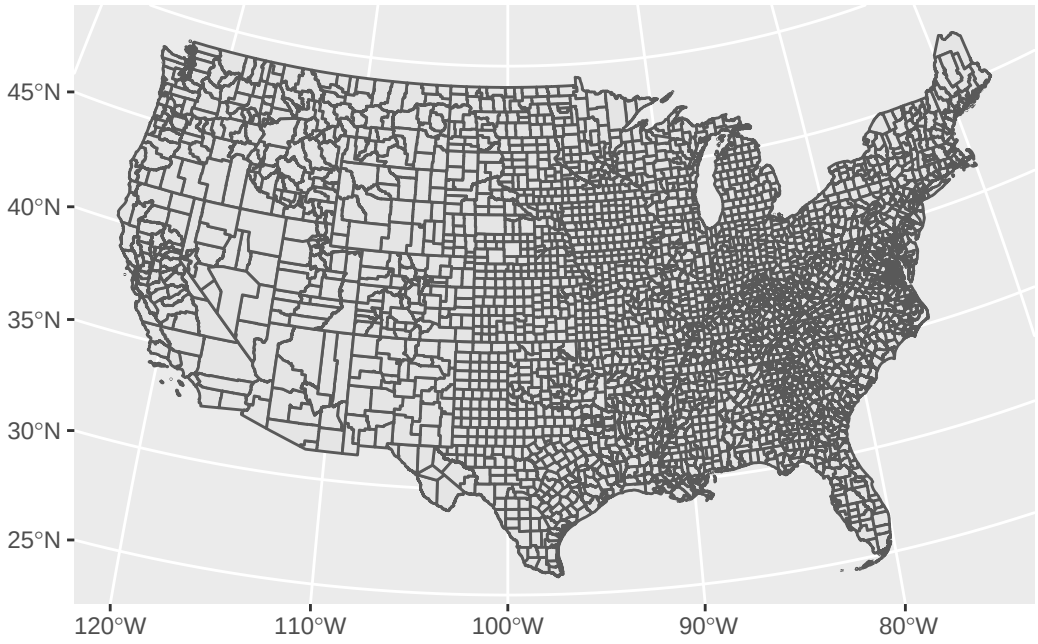
```
Warning in st_centroid.sf(us_co): st_centroid assumes attributes are constant
over geometries of x

Warning in knearneigh(st_centroid(us_co), k = 5, longlat = T): dnearneigh:
longlat argument overrides object
```

```
nbs<-knn2nb(nbs, row.names = us_co$struct, sym = T) #force symmetry!!

mat <- nb2mat(nbs, style="B",zero.policy=TRUE)

colnames(mat) <- rownames(mat)

mat <- as.matrix(mat[1:dim(mat)[1], 1:dim(mat)[1]])


nb2INLA("cl_graph",nbs)
am_adj <-paste(getwd(),"/cl_graph",sep="")
H<-inla.read.graph(filename="cl_graph")
```

Plot geographies

```
library(sf)
us_co<-st_as_sf(us_co)
us_co$cofips<-paste(us_co$STATEFP, us_co$COUNTYFP, sep="")

us_co%>%
```

```
  ggplot()+
  geom_sf()+
  coord_sf(crs = 2163)
```



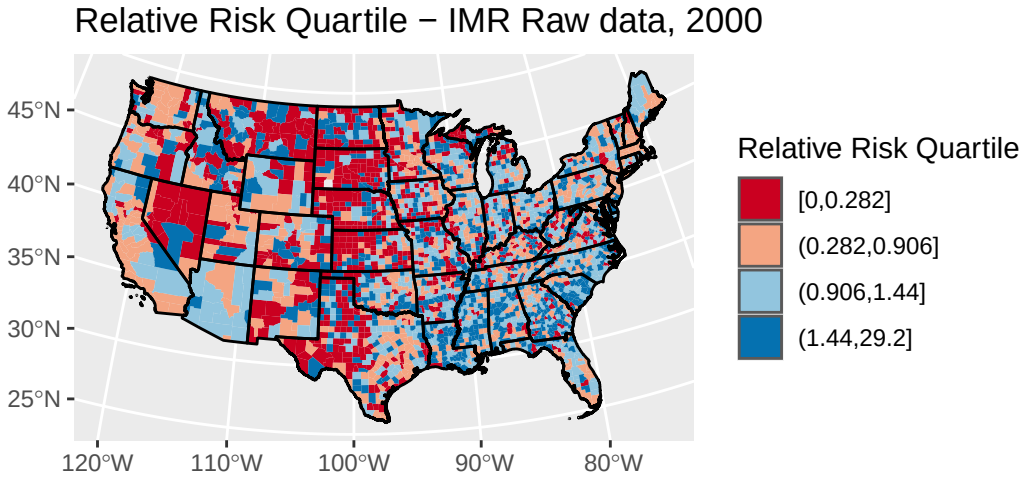

```
final.dat<-merge( us_co,final.dat, by="cofips")
final.dat<-final.dat[order(final.dat$cofips),]
```

```
Retrieving data for the year 2020
```

Map of the Infant mortality raw relative risk

```
final.dat%>%
  #filter(year%in%c(2000))%>%
  mutate(qrr=cut(I(deaths/E_d),
                 breaks = quantile(I(deaths/E_d),
                                   p=seq(0,1,length.out = 5)),
                 include.lowest = T))%>%
  ggplot()+
  geom_sf(aes(fill=qrr, color=NA))+
```

```
  geom_sf(data=sts, color="black")+
  scale_colour_brewer(palette = "RdBu" )+
  scale_fill_brewer(palette = "RdBu", na.value="grey")+
  guides(fill=guide_legend(title="Relative Risk Quartile"))+
  ggtitle(label="Relative Risk Quartile - IMR Raw data, 2000")+
  coord_sf(crs = 2163)
```




```
#ggsave(filename = "C:/Users/ozd504/Documents/GitHub/talks/imr_raw2000.png", dpi = "print"
```

Model setup

- We have a count outcome (deaths and births), in counties over time, and a set of time-constant covariates.
- We have several options in the GLM framework with which to model these data, for example:
- Binomial -

$$y_{ij} \sim Bin(\pi_{ij}): \; logit(\pi_{ij}) = \beta_0 + x'\beta_k$$

- Poisson -
$$y_{ij} \sim Pois(\lambda_{ij}E_{ij}):\ log(\lambda_{ij}) = log(E_{ij}) + \beta_0 + x'\beta_k$$
- Negative Binomial -
$$y_{ij} \sim \text{Neg Bin}(\mu_{ij}, \alpha, E_{ij}):\ log(\mu_{ij}) = log(E_{ij}) + \beta_0 + x'\beta_k$$
- In addition to various zero-inflated versions of these data.

We can fit these model using the Bayesian framework with INLA.

First, we consider the basic GLM for the mortality outcome, without any hierarchical structure.

We can write this model as a Negative Binomial model, for instance as:

$$Deaths_{ij} = log(E_d) + X'\beta$$

INLA will use vague Normal priors for the β's, and we have not other parameters in the model to specify priors for.

INLA does not require you to specify all priors, as all parameters have a default prior specifications.

Basic INLA model specification

```
#Model specification:
f1<-deaths~scale(pblack)+scale(phisp)+scale(ppov)

#Model fit
mod1<-inla(formula = f1, #linear predictor - fixed effects
           data = final.dat, #data set object name
           family = "poisson", #marginal distribution for the outcome
           E = E_d, # expected count
           control.compute = list(waic=T), # compute DIC or not?
           control.predictor = list(link=1), #estimate predicted values & their marginals
           num.threads = 3, # control number of computing cores
               verbose = F)

#model summary
summary(mod1)
```

```
Call:
   c("inla.core(formula = formula, family = family, contrasts = contrasts,
   ", " data = data, quantiles = quantiles, E = E, offset = offset, ", "
   scale = scale, weights = weights, Ntrials = Ntrials, strata = strata,
   ", " lp.scale = lp.scale, link.covariates = link.covariates, verbose =
   verbose, ", " lincomb = lincomb, selection = selection, control.compute
   = control.compute, ", " control.predictor = control.predictor,
   control.family = control.family, ", " control.inla = control.inla,
   control.fixed = control.fixed, ", " control.mode = control.mode,
   control.expert = control.expert, ", " control.hazard = control.hazard,
   control.lincomb = control.lincomb, ", " control.update =
   control.update, control.lp.scale = control.lp.scale, ", "
   control.pardiso = control.pardiso, only.hyperparam = only.hyperparam,
   ", " inla.call = inla.call, inla.arg = inla.arg, num.threads =
   num.threads, ", " blas.num.threads = blas.num.threads, keep = keep,
   working.directory = working.directory, ", " silent = silent, inla.mode
   = inla.mode, safe = FALSE, debug = debug, ", " .parent.frame =
   .parent.frame)")
Time used:
    Pre = 0.709, Running = 0.507, Post = 0.0674, Total = 1.28
Fixed effects:
                mean    sd 0.025quant 0.5quant 0.975quant mode kld
(Intercept)    0.024 0.008      0.008    0.024      0.041   NA   0
scale(pblack)  0.132 0.007      0.119    0.132      0.146   NA   0
scale(phisp)  -0.089 0.006     -0.100   -0.089     -0.078   NA   0
scale(ppov)    0.091 0.008      0.075    0.091      0.108   NA   0

Watanabe-Akaike information criterion (WAIC) ...: 12250.25
Effective number of parameters .................: 14.45

Marginal log-Likelihood:  -6145.49
 is computed
Posterior summaries for the linear predictor and the fitted values are computed
(Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')
```

Basic county level random intercept model

Now we add basic nesting of rates within counties, with a random intercept term for each county. This would allow there to be heterogeneity in the mortality rate for each county, over and above each county's observed characteristics.

This model would be:

$\text{Deaths}_{ij} = log(E_d) + X'\beta + u_j$

$u_j \sim \text{Normal}(0, \tau_u)$

where τ_u here is the precision, not the variance and **precision = 1/variance.**

INLA puts a log-gamma prior on the the precision by default.

```
f2<-deaths~scale(pblack) + scale(phisp) + scale(ppov)+ #fixed effects
  f(struct, model = "iid")  #random effects

mod2<-inla(formula = f2,
           data = final.dat,
           family = "poisson",
           E = E_d,
           control.compute = list(waic=T),
           control.predictor = list(link=1),
           num.threads = 3,
               verbose = F)

#total model summary
summary(mod2)
```

```
Call:
   c("inla.core(formula = formula, family = family, contrasts = contrasts,
   ", " data = data, quantiles = quantiles, E = E, offset = offset, ", "
   scale = scale, weights = weights, Ntrials = Ntrials, strata = strata,
   ", " lp.scale = lp.scale, link.covariates = link.covariates, verbose =
   verbose, ", " lincomb = lincomb, selection = selection, control.compute
   = control.compute, ", " control.predictor = control.predictor,
   control.family = control.family, ", " control.inla = control.inla,
   control.fixed = control.fixed, ", " control.mode = control.mode,
   control.expert = control.expert, ", " control.hazard = control.hazard,
   control.lincomb = control.lincomb, ", " control.update =
   control.update, control.lp.scale = control.lp.scale, ", "
   control.pardiso = control.pardiso, only.hyperparam = only.hyperparam,
   ", " inla.call = inla.call, inla.arg = inla.arg, num.threads =
   num.threads, ", " blas.num.threads = blas.num.threads, keep = keep,
   working.directory = working.directory, ", " silent = silent, inla.mode
   = inla.mode, safe = FALSE, debug = debug, ", " .parent.frame =
   .parent.frame)")
```

```
Time used:
    Pre = 0.557, Running = 1.28, Post = 0.0995, Total = 1.93
Fixed effects:
                mean    sd 0.025quant 0.5quant 0.975quant mode kld
(Intercept)    0.015 0.010     -0.005    0.015      0.035   NA   0
scale(pblack)  0.130 0.010      0.110    0.130      0.150   NA   0
scale(phisp)  -0.074 0.009     -0.093   -0.074     -0.056   NA   0
scale(ppov)    0.098 0.011      0.075    0.098      0.120   NA   0

Random effects:
  Name    Model
    struct IID model

Model hyperparameters:
                      mean   sd 0.025quant 0.5quant 0.975quant mode
Precision for struct 26.48 3.03      21.20    26.26      32.92   NA

Watanabe-Akaike information criterion (WAIC) ...: 11731.74
Effective number of parameters .................: 386.76

Marginal log-Likelihood:  -5976.12
 is computed
Posterior summaries for the linear predictor and the fitted values are computed
(Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')
```

Marginal Distributions of hyperparameters

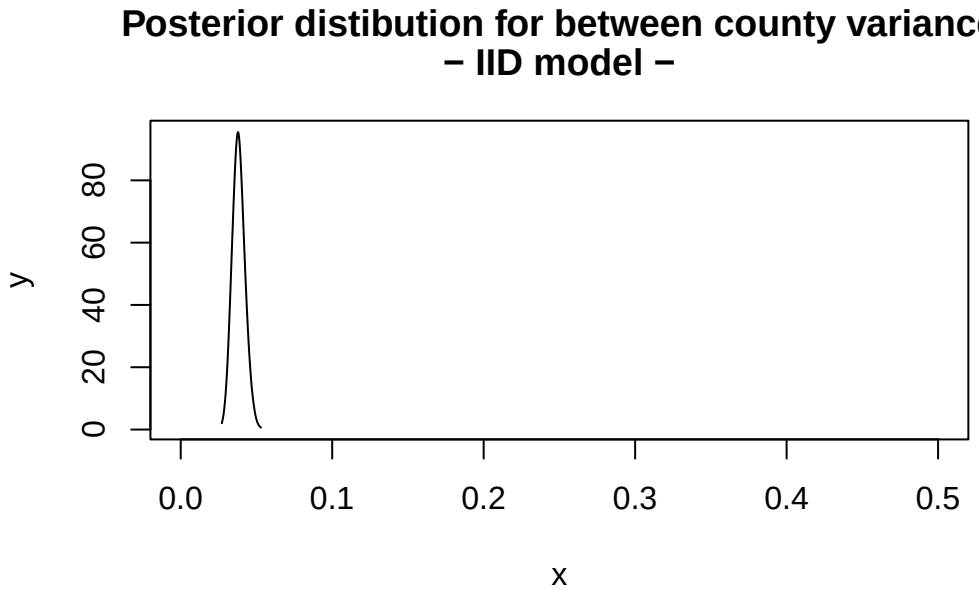
We can plot the posterior marginal of the hyperparameter in this model, in this case $\sigma_u = 1/\tau_u$

```
m2<- inla.tmarginal(
        function(x) (1/x), #invert the precision to be on variance scale
        mod2$marginals.hyperpar$`Precision for struct`)
#95% credible interval for the variance
inla.hpdmarginal(.95, marginal=m2)
```

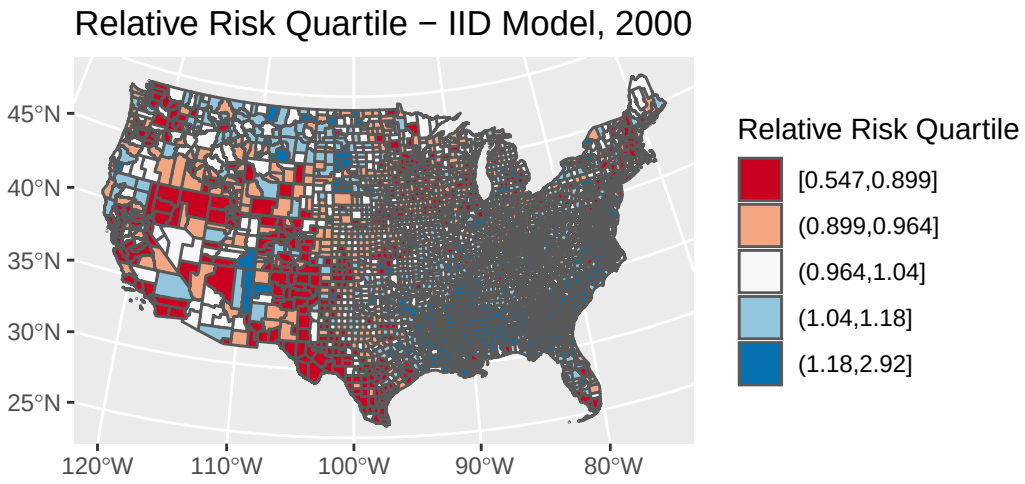
```
                low       high
level:0.95 0.0300349 0.04673984
```

```
plot(m2,
     type="l",
```

```
    main=c("Posterior distibution for between county variance", "- IID model -"),
    xlim=c(0, .5))
```




```
final.dat$fitted_m2<-mod2$summary.fitted.values$mean

final.dat%>%
  mutate(qrr=cut(fitted_m2,
                 breaks = quantile(fitted_m2, p=seq(0,1,length.out = 6)),
                 include.lowest = T))%>%
  ggplot()+geom_sf(aes(fill=qrr))+
  scale_colour_brewer(palette = "RdBu" )+
  scale_fill_brewer(palette = "RdBu", na.value="grey")+
  guides(fill=guide_legend(title="Relative Risk Quartile"))+
  ggtitle(label="Relative Risk Quartile - IID Model, 2000")+
  coord_sf(crs = 2163)
```



Besag, York, and Mollie - BYM Model

Model with spatial correlation - Besag, York, and Mollie (1991) model

$$\text{Deaths}_{ij} = log(E_d) + X'\beta + u_j + v_j$$

Which has two random effects, one an IID random effect and the second a spatially correlated random effect, specified as a conditionally auto-regressive prior for the v_j's.

This is the Besag model:

$$v_j|v_{\neq j}, \sim \text{Normal}(\frac{1}{n_i}\sum_{i\sim j} v_j, \frac{1}{n_i\tau})$$

and u_j is an IID normal random effect, γ_t is also given an IID Normal random effect specification, and there are now three hyperparameters, τ_u and τ_v and τ_γ and each are given log-gamma priors.

For the BYM model we must specify the spatial connectivity matrix in the random effect.

```
f3<-deaths~scale(pblack)+scale(phisp)+scale(ppov)+
  f(struct, model = "bym",scale.model = T, constr = T, graph = H) #specify graph for spati
```

```r
mod3<-inla(formula = f3,data = final.dat,
           family = "poisson",
           E = E_d,
           control.compute = list(waic=T, return.marginals.predictor=TRUE),
           control.predictor = list(link=1),
           num.threads = 3,
               verbose = F)

#total model summary
summary(mod3)
```

```
Call:
   c("inla.core(formula = formula, family = family, contrasts = contrasts,
   ", " data = data, quantiles = quantiles, E = E, offset = offset, ", "
   scale = scale, weights = weights, Ntrials = Ntrials, strata = strata,
   ", " lp.scale = lp.scale, link.covariates = link.covariates, verbose =
   verbose, ", " lincomb = lincomb, selection = selection, control.compute
   = control.compute, ", " control.predictor = control.predictor,
   control.family = control.family, ", " control.inla = control.inla,
   control.fixed = control.fixed, ", " control.mode = control.mode,
   control.expert = control.expert, ", " control.hazard = control.hazard,
   control.lincomb = control.lincomb, ", " control.update =
   control.update, control.lp.scale = control.lp.scale, ", "
   control.pardiso = control.pardiso, only.hyperparam = only.hyperparam,
   ", " inla.call = inla.call, inla.arg = inla.arg, num.threads =
   num.threads, ", " blas.num.threads = blas.num.threads, keep = keep,
   working.directory = working.directory, ", " silent = silent, inla.mode
   = inla.mode, safe = FALSE, debug = debug, ", " .parent.frame =
   .parent.frame)")
Time used:
    Pre = 0.857, Running = 4.23, Post = 0.272, Total = 5.36
Fixed effects:
               mean    sd 0.025quant 0.5quant 0.975quant mode kld
(Intercept)   0.026 0.011      0.005    0.026      0.048   NA   0
scale(pblack) 0.116 0.013      0.090    0.116      0.141   NA   0
scale(phisp) -0.037 0.014     -0.065   -0.037     -0.008   NA   0
scale(ppov)   0.095 0.013      0.070    0.095      0.120   NA   0

Random effects:
  Name    Model
```

```
    struct BYM model

Model hyperparameters:
                                          mean    sd 0.025quant 0.5quant
Precision for struct (iid component)     63.62 19.44      34.83    60.46
Precision for struct (spatial component) 38.75 12.25      19.46    37.18
                                         0.975quant mode
Precision for struct (iid component)         110.58   NA
Precision for struct (spatial component)      67.25   NA

Watanabe-Akaike information criterion (WAIC) ...: 11684.54
Effective number of parameters .................: 332.54

Marginal log-Likelihood:  -4876.30
 is computed
Posterior summaries for the linear predictor and the fitted values are computed
(Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')
```

Compare model fits

```
mod1$waic$waic
```

```
[1] 12250.25
```

```
mod2$waic$waic
```

```
[1] 11731.74
```

```
mod3$waic$waic
```

```
[1] 11684.54
```

Looks like the spatial is better in this case.

Regression effect posterior distributions

If we want to get the 95% credible interval for a risk ratio $exp(\beta)$, we can do so using the `inla.emarginal()` and `inla.tmarginal()` functions

```
b1 <- inla.emarginal(exp, mod2$marginals.fixed[[2]])
b1 #Posterior mean
```

```
[1] 1.138849
```

```
b1ci <-inla.qmarginal(c(.025, .975),
                      inla.tmarginal(exp, mod2$marginals.fixed[[2]]))
b1ci
```

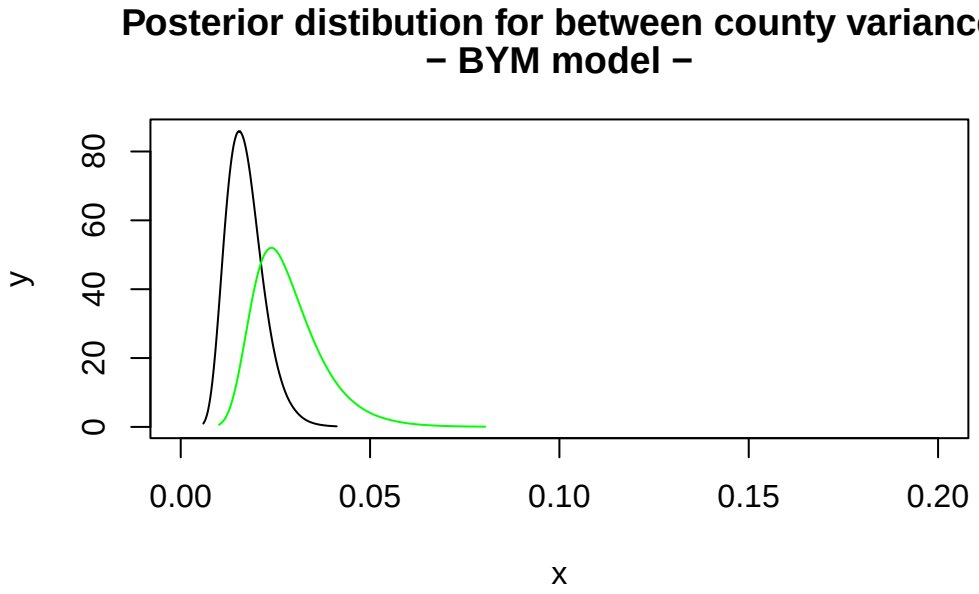
```
[1] 1.116276 1.161713
```

And we see that the `pblack` effect increases the infant mortality risk by 0.14%

Hyper parameter distributions

Green line is spatial variation, black line is non-spatial variation

```
m3a<- inla.tmarginal(
        function(x) (1/x),
        mod3$marginals.hyperpar$`Precision for struct (iid component)`)

m3b<- inla.tmarginal(
        function(x) (1/x),
        mod3$marginals.hyperpar$`Precision for struct (spatial component)`)

plot(m3a, type="l",
     main=c("Posterior distibution for between county variance", "- BYM model -"),
     xlim=c(0, .2)) # you may need to change this

lines(m3b, col="green")
```



HPD interval for variances

```
inla.hpdmarginal(.95,m3a)
```

```
                  low       high
level:0.95 0.008203331 0.02705066
```

```
inla.hpdmarginal(.95,m3b)
```

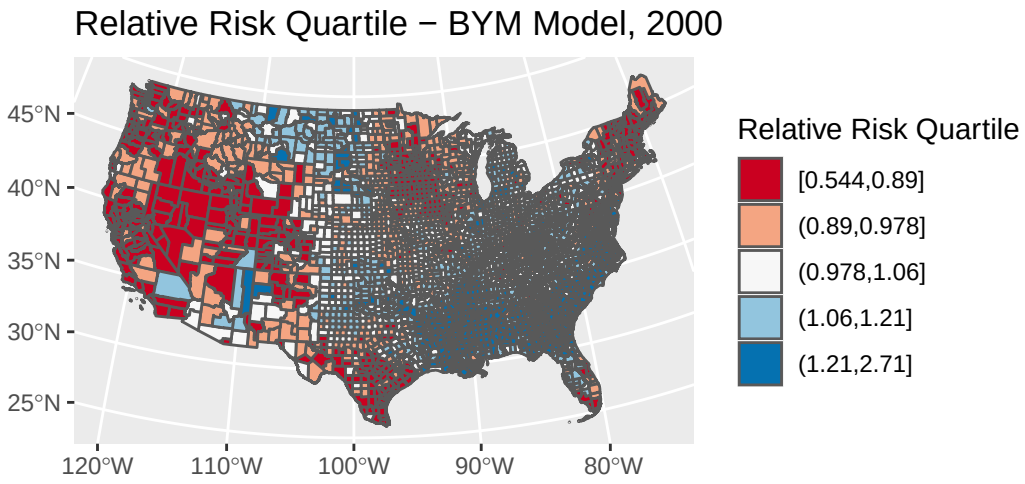
```
                 low       high
level:0.95 0.01308216 0.04719253
```

Spatial mapping of the fitted values

Here we can recover the estimated smoothed relative risks for each county, based on the model above.

```
final.dat$fitted_m3<-mod3$summary.fitted.values$mean

final.dat%>%
```

```
# filter(year%in%c(2000))%>%
 mutate(qrr=cut(fitted_m3,
                breaks = quantile(fitted_m3, p=seq(0,1,length.out = 6)),
                include.lowest = T))%>%
 ggplot()+
 geom_sf(aes(fill=qrr))+
 scale_colour_brewer(palette = "RdBu" )+
 scale_fill_brewer(palette = "RdBu", na.value="grey")+
 guides(fill=guide_legend(title="Relative Risk Quartile"))+
 ggtitle(label="Relative Risk Quartile - BYM Model, 2000")+
 coord_sf(crs = 2163)
```

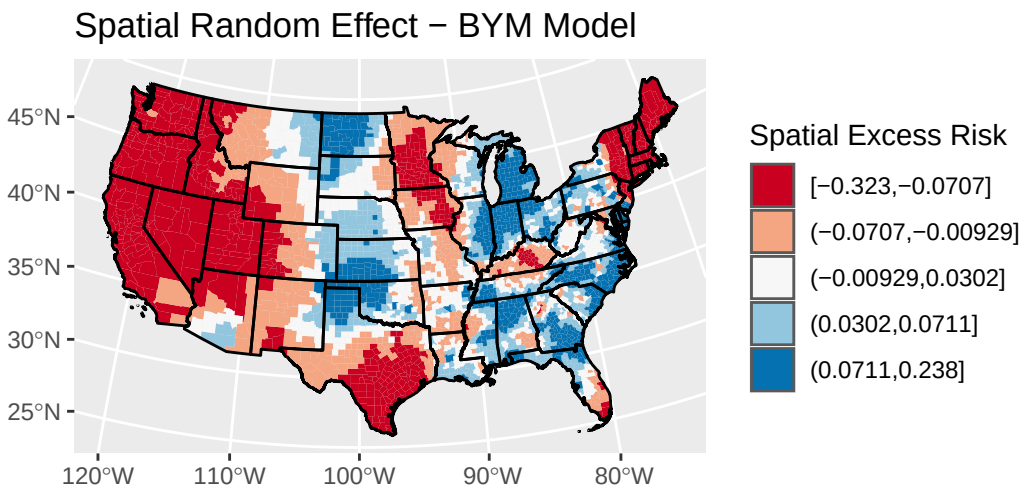


Map of spatial random effects

It is common to map the random effects from the BYM model to look for spatial trends, in this case, there is a strong spatial signal:

```
us_co$sp_re<-mod3$summary.random$struct$mean[3109:6216]
us_co%>%
  mutate(qse=cut(sp_re,
                 breaks = quantile(sp_re, p=seq(0,1,length.out = 6)),
```

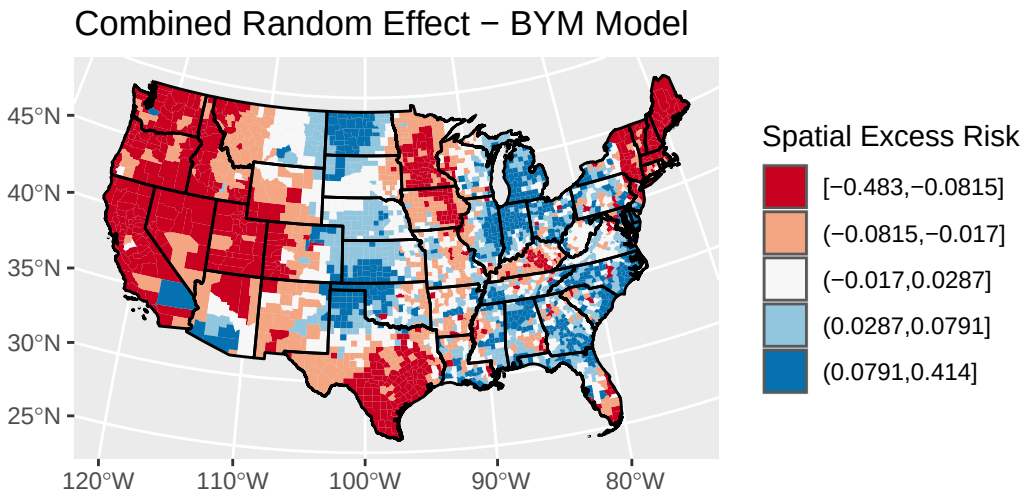
```
                include.lowest = T))%>%
  ggplot()+
  geom_sf(aes(fill=qse, color=NA))+
  geom_sf(data=sts)+
  scale_colour_brewer(palette = "RdBu" )+
  scale_fill_brewer(palette = "RdBu", na.value="grey")+
  guides(fill=guide_legend(title="Spatial Excess Risk"))+
  ggtitle(label="Spatial Random Effect - BYM Model")+
  coord_sf(crs = 2163)
```



Spatial + IID random effect

```
us_co$i_re<-mod3$summary.random$struct$mean[1:3108]
us_co%>%
  mutate(qse=cut(i_re,
                 breaks = quantile(i_re, p=seq(0,1,length.out = 6)),
                 include.lowest = T))%>%
  ggplot()+
  geom_sf(aes(fill=qse, color=NA))+
  geom_sf(data=sts)+
```

```
scale_colour_brewer(palette = "RdBu" )+
scale_fill_brewer(palette = "RdBu", na.value="grey")+
guides(fill=guide_legend(title="Spatial Excess Risk"))+
ggtitle(label="Combined Random Effect - BYM Model")+
coord_sf(crs = 2163)
```



Summary

- Spatial analysis often involves a lot of exploratory data analysis. Maps and statistics like Moran's I are very useful to explore your data
- Regression models can easily incorporate spatial correlation structures, and the INLA approach allows a very flexible method for estimating such models in the Bayesian framework.

Anselin, Luc. 2010. "Local Indicators of Spatial Association-LISA." *Geogr. Anal.* 27 (2): 93–115. https://doi.org/10.1111/j.1538-4632.1995.tb00338.x.

Moran, P. A. P. 1950. "Notes on Continuous Stochastic Phenomena." *Biometrika* 37 (1/2): 17. https://doi.org/10.2307/2332142.

Tobler, W. R. 1970. "A Computer Movie Simulating Urban Growth in the Detroit Region." *Econ. Geogr.* 46 (June): 234. https://doi.org/10.2307/143141.