Spatial and Spatio-temporal Epidemiology 31 (2019) 100301

Contents lists available at ScienceDirect

Spatial and Spatio-temporal Epidemiology

journal homepage: www.elsevier.com/locate/sste

Bayesian hierarchical spatial models: Implementing the Besag York Mollié model in stan

Mitzi Morris[a], Katherine Wheeler-Martin[b,*], Dan Simpson[c], Stephen J. Mooney[d], Andrew Gelman[e], Charles DiMaggio[b]

[a] Institute for Social and Economic Research and Policy, Columbia University, New York, NY, United States
[b] Department of Surgery, New York University School of Medicine, New York, NY, United States
[c] Department of Statistical Sciences, University of Toronto, Toronto, Ontario, Canada
[d] Department of Epidemiology, University of Washington, Seattle, WA, United States
[e] Department of Statistics, Columbia University, New York, NY, United States

ARTICLE INFO

Article history:
Received 3 December 2018
Revised 5 August 2019
Accepted 6 August 2019
Available online 12 August 2019

Keywords:
Bayesian inference
Intrinsic conditional auto-regressive model
Besag-York-Mollié model
Probabilistic programming
Stan
Pedestrian injuries

ABSTRACT

This report presents a new implementation of the Besag-York-Mollié (BYM) model in Stan, a probabilistic programming platform which does full Bayesian inference using Hamiltonian Monte Carlo (HMC).

We review the spatial auto-correlation models used for areal data and disease risk mapping, and describe the corresponding Stan implementations.

We also present a case study using Stan to fit a BYM model for motor vehicle crashes injuring school-age pedestrians in New York City from 2005 to 2014 localized to census tracts.

Stan efficiently fit our multivariable BYM model having a large number of observations (n=2095 census tracts) with small outcome counts < 10 in the majority of tracts. Our findings reinforced that neighborhood income and social fragmentation are significant correlates of school-age pedestrian injuries. We also observed that nationally-available census tract estimates of commuting methods may serve as a useful indicator of underlying pedestrian densities.

1. Introduction

Spatial auto-correlation is the tendency for adjacent areas to share similar characteristics. Conditional Auto-Regressive (CAR) and Intrinsic Conditional Auto-Regressive (ICAR) models, first introduced by Besag (1974), account for this by pooling information from neighboring regions. The BYM model (Besag and Mollié, 1991), is a lognormal Poisson model which includes both an ICAR component for spatial auto-correlation and an ordinary random-effects component for non-spatial heterogeneity. Because either component of the BYM model can account for most or all of the individual-level variance, it is difficult to fit using MCMC methods. In this report we present an implementation of the BYM2 model (Riebler et at., 2016), a reparameterization of the BYM model, in Stan, a probabilistic programming platform which does full Bayesian inference using Hamiltonian Monte Carlo (HMC). Stan's No U-Turn Sampler (NUTS) provides better and more robust estimates for models such as the BYM model which have complex posteriors than samplers which use Gibbs or Metropolis algorithms.

Part one of the paper reviews spatial modeling concepts and introduces the Stan language, tools, and workflow. First, in section *Models*, we review the specification of the CAR and ICAR models and show why it is much faster to compute log probability density of the ICAR model instead of the CAR model. Then we review the original formulation of the BYM model and present the BYM2 model, a reparameterization of the BYM model where all parameters have clear interpretations and the choice of hyperpriors is straightforward.

The *Stan Programs* section is an introduction to both the Stan language and the R package `rstan`. As a first Stan program, we implement the ICAR model. The expressive power of the Stan language allows for a straightforward translation from the mathematical model to a Stan program. Using Stan's vectorized operations, the joint specification of the ICAR model in Stan corresponds directly to its mathematical formulation over the pairwise differences between neighboring regions. To validate this model, we fit the areal map over 2095 New York City census tracts with `rstan` and use the R package `ggplot2` to show how the model recovers

* Corresponding author.
E-mail address: katherine.wheeler-martin@nyulangone.org (K. Wheeler-Martin).

https://doi.org/10.1016/j.sste.2019.100301
1877-5845/

the spatial structure present in the data. The second Stan program implements the BYM2 model.

Finally, in the *Case Study* section, we present a full, substantive example of a Stan spatial analytic model using the BYM2 model to fit New York City motor vehicle crash data. The study aims to map the geographic distribution of school-age pedestrian injuries at the census tract level from 2005 to 2014, as well as explore sociodemographic factors associated with their occurrence at the community level.

2. Models

2.1. Conditional autoregressive models

Areal data consists of a finite set of regions with well-defined boundaries, each of which has a single measurement aggregated from its population. Counts of rare events in small-population regions are noisy; removing this noise allows the underlying phenomena of interest to be seen more clearly. Conditional autoregressive (CAR) models smooth noisy estimates by pooling information from neighboring regions. Given a set of N regions, the binary *neighbor* relationship (written $i \sim j$ where $i \neq j$) is 1 if regions n_i and n_j are neighbors and is otherwise 0. For CAR models, the neighbor relationship is symmetric but not reflexive; if $i \sim j$ then $j \sim i$, but a region is not its own neighbor.

Spatial interactions between pairs of units i and j can be modeled conditionally as a normal random variable $\boldsymbol{\phi}$, which is an N-length vector $\phi = (\phi_1, \ldots, \phi_n)^\intercal$. In the full conditional distribution, each ϕ_i is conditional on the sum of the weighted values of its neighbors ($w_{ij}\phi_j$) and has unknown variance

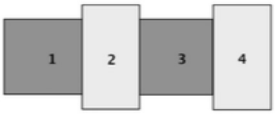
$$\phi_i \mid \phi_j, j \neq i, \sim \mathrm{N}\left(\sum_{j=1}^{n} w_{ij}\phi_j, \sigma^2\right).$$

Specification of the global, or joint distribution via the local specification of the conditional distributions of the individual random variables defines a Gaussian Markov random field (GMRF). (Besag, 1974) proved that the corresponding joint specification of ϕ is a multivariate normal random variable centered at 0. The variance of ϕ is specified as a precision matrix Q which is simply the inverse of the covariance matrix Σ, i.e. $\Sigma = Q^{-1}$ so that

$$\phi \sim \mathrm{N}(0, \mathrm{Q}^{-1}).$$

For standard multivariate normal random variable ϕ, the precision matrix Q is constructed from two matrices which describe the neighborhood structure of the N regions: the diagonal matrix D and the adjacency matrix A. The diagonal matrix is an $N \times N$ matrix where each diagonal entry n_{ii} contains the number of neighbors of region n_i and all off-diagonal entries are zero. The adjacency matrix is an $N \times N$ matrix where entry n_{ij} is 1 if regions n_i and n_j are neighbors and 0 otherwise and all diagonal entries n_{ii} are zero.

The adjacency matrix encodes the neighborhood graph. If any region in the map can be reached from any other region via a series of neighboring regions, then the map is a single, fully connected component. The number of components of a neighborhood graph ranges from 1 to N, in the case where all regions are islands. To see how this works, we construct a simple example using a map over 4 regions (n_1, n_2, n_3, n_4) consisting of a single component with neighbor relations: ($1 \sim 2$, $2 \sim 3$, $3 \sim 4$)



The adjacency matrix A is:

$$\begin{array}{c} \\ n_1 \\ n_2 \\ n_3 \\ n_4 \end{array} \begin{array}{c} \begin{array}{cccc} n_1 & n_2 & n_3 & n_4 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

The diagonal matrix D is:

$$\begin{array}{c} \\ n_1 \\ n_2 \\ n_3 \\ n_4 \end{array} \begin{array}{c} \begin{array}{cccc} n_1 & n_2 & n_3 & n_4 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

To make the standard multivariate normal random variable ϕ have a proper joint probability density, the precision matrix Q must be symmetric and positive definite. For the CAR model, Q is defined as

$$Q = \mathrm{D}\,(\mathrm{I} - \alpha\,\mathrm{A})$$

where I is the identity matrix and $0 < \alpha < 1$. The term α is the CAR model parameter which controls for the amount of spatial dependence, where $\alpha = 0$ implies spatial independence. Scaling A by α makes the quantity $\mathrm{D}(I - \alpha\,\mathrm{A})$ positive definite. Because the neighbor relationship $i \sim j$ is symmetric by definition for CAR models, both A and Q are symmetric.

For the above example, when $\alpha = 0.5$, $\mathrm{D}(I - \alpha\,\mathrm{A})$ is:

$$\begin{array}{c} \\ n_1 \\ n_2 \\ n_3 \\ n_4 \end{array} \begin{array}{c} \begin{array}{cccc} n_1 & n_2 & n_3 & n_4 \end{array} \\ \begin{pmatrix} 1 & -0.5 & 0 & 0 \\ -0.5 & 2 & -0.5 & 0 \\ 0 & -0.5 & 2 & -0.5 \\ 0 & 0 & -0.5 & 1 \end{pmatrix} \end{array}$$

The log probability density of ϕ is proportional to

$$\frac{n}{2}\log(\det(Q)) - \frac{1}{2}\phi^T Q\,\phi$$

where n is the number of components in the neighborhood graph. Computing the determinant of Q requires N^3 operations, e.g., when $N = 100$, $\det(Q)$ takes a million operations and when $N = 1000$ it takes a billion operations. For large number of regions N, this is computationally expensive for an MCMC sampler as the sampler recomputes the probability density of ϕ for each new proposal.

2.2. Intrinsic conditional autoregressive models

The intrinsic conditional autoregressive (ICAR) model sets α to 1, effectively eliminating α from the model so that the quantity $\mathrm{D}(I - \alpha\,\mathrm{A})$ simplifies to $\mathrm{D} - \mathrm{A}$. For the above example, $\mathrm{D} - \mathrm{A}$ is

$$\begin{array}{c} \\ n_1 \\ n_2 \\ n_3 \\ n_4 \end{array} \begin{array}{c} \begin{array}{cccc} n_1 & n_2 & n_3 & n_4 \end{array} \\ \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} \end{array}$$

Now the value of the determinant of Q is 0. The ICAR prior is improper but the posterior is proper once you include some data.

MCMC samplers compute the log probability up to a proportionality constant. When computing the log probability density of the ICAR model, the term $\frac{n}{2}\log(\det(Q))$ is constant and therefore drops out of the calculation. This reduces the number of operations needed to compute the log density from N^3 to N^2, making it

possible to fit datasets for large areal maps with an MCMC sampler running on a modern laptop computer in only a few hours, instead of many days.

In the ICAR model, each ϕ_i is normally distributed with a mean equal to the average of its neighbors. Its variance decreases as the number of neighbors, denoted d_i, increases. The conditional specification of the ICAR model is:

$$p\left(\phi_i \mid \phi_{i\sim j}\right) = \mathrm{N}\left(\frac{\sum_{i\sim j}\phi_i}{d_i}, \frac{{\sigma_i}^2}{d_i}\right)$$

where ${\sigma_i}^2$ is the unknown variance.

The joint specification of the ICAR random vector ϕ centered at 0 with common variance 1 rewrites to the *pairwise difference* formulation:

$$p(\phi) \propto \exp\left(-\frac{1}{2}\sum_{i\sim j}(\phi_i - \phi_j)^2\right).$$

Writing the joint density as the pairwise difference makes it easy to reason about the behavior of this model: each $(\phi_i - \phi_j)^2$ contributes a penalty term based on the distance between the values of neighboring regions; minimizing this term results in spatial smoothing. The pairwise difference is non-identifiable; any constant added to ϕ washes out of the term $\phi_i - \phi_j$. Adding the constraint $\sum_N \phi_i = 0$ centers this model. With this constraint the log probability density is defined because the domain of integration is restricted to the set of parameters summing to 1.

2.3. The Besag-York-Mollié model

The BYM model is a lognormal Poisson model developed for disease risk mapping which includes both an ICAR component for spatial smoothing and an ordinary random effects component for non-spatial heterogeneity. The Poisson regression is used to estimate the unknown log relative risk η_i for zone i, $(i = 1, 2, \ldots n)$, given y_i, the observed number of cases. The BYM model specifies:

$$\eta_i = \mu + x\beta + \phi + \theta$$

where:

- μ is the overall risk level, i.e., the fixed intercept.
- x is the matrix of explanatory spatial covariates such that x_i is the vector of covariates for areal unit i and β is vector of regression coefficients which are constant across all regions, i.e., fixed effects.
- ϕ is an ICAR spatial component.
- θ is an ordinary random effects component for non-spatial heterogeneity.

The BYM model uses both spatial and non-spatial error terms to account for over-dispersion not modelled by the Poisson variates. When the observed variance isn't fully explained by the spatial structure of the data, an ordinary random effects component will account for the rest. However, this model becomes difficult to fit because either component can account for most or all of the individual-level variance. Without any hyperpriors on ϕ and θ the sampler will be forced to explore many extreme posterior probability distributions; the sampler will go very slowly or fail to fit the data altogether. Riebler et al. (2016) provides an excellent summary of the underlying problem as well as a survey of the subsequent refinements to the parameterization and choice of priors for this model.

In order to fit the BYM model to their data using a custom Gibbs sampler, Besag and Mollié (1991) use gamma hyperpriors on the precision parameters τ_ϕ and τ_θ, with carefully chosen parameter values for each. Subsequent versions of this model use constraints designed to create a "fair" prior which places equal emphasis on both spatial and non-spatial variance, based on the formula from Clayton and Montomoli (1995):

$$sd(\theta_i) = \frac{1}{\sqrt{\tau_\phi}} \approx \frac{1}{0.7\sqrt{\bar{m}\tau_\theta}} \approx sd(\phi_i)$$

where $\bar{m}$ is the average number of neighbors across all regions in the dataset. Because the values used for the gamma hyperprior on τ_θ depend on the value of $\bar{m}$, the choice of hyperpriors is dependent on the dataset being analyzed and therefore must be reevaluated for each new dataset accordingly.

2.4. The BYM2 model

The BYM2 model (Riebler et al., 2016) follows the *Penalized Complexity* framework (Simpson et al., 2017), which favors models where the parameters have clear interpretations, allowing for assignment of sensible hyperparameters to each. Like the BYM model, the BYM2 model includes both spatial and non-spatial error terms and like the alternative model of Leroux et al. (2000), it places a single precision (scale) parameter σ on the combined components and a mixing parameter ρ for the amount of spatial/non-spatial variation. In order for σ to legitimately be the standard deviation of the combined components, it is critical that for each i, $\mathrm{Var}(\phi_i) \approx \mathrm{Var}(\theta_i) \approx 1$. This is done by adding a scaling factor s to the model which scales the proportion of variance ρ.

Because the scaling factor s depends on the dataset, it comes into the model as data. Riebler et al. recommend scaling the model so the geometric mean of these variances is 1. This scaling factor can be computed from the neighborhood graph in the `transformed data` block of the Stan program, but here we compute this value using R's `INLA::inla.scale.model` function and pass it into the Stan model as data.

In the BYM2 model, the original BYM model's combination of components $\phi + \theta$ is rewritten as

$$\left(\left(\sqrt{\rho/s}\right)\phi^* + \left(\sqrt{1-\rho}\right)\theta^*\right)\sigma$$

where:

- $\rho \in [0, 1]$ models how much of the variance comes from the spatially correlated error terms and how much comes from the independent error terms
- ϕ^* is the ICAR model
- $\theta^* \sim \mathrm{N}(0, \mathrm{n})$, where n is the number of connected subgraphs. When the neighborhood graph is fully connected $\theta^* \sim \mathrm{N}(0, 1)$.
- s is the scaling factor computed from the neighborhood graph such that $\mathrm{Var}(\phi_i) \approx 1$.
- $\sigma \geq 0$ is the overall standard deviation for the combined error terms

For BYM2 models over neighborhood graphs which are not fully connected, (i.e., $\mathrm{n} > 1$), each connected subgraph has its own variance, and must be scaled accordingly. The Stan programming language is powerful enough to allow for disconnected subgraphs and island regions, however the indexing required to keep track of each subgraph increases the complexity of the code, therefore in this paper we present a Stan program for fully connected neighborhood graphs.

3. Stan programs

Stan is a highly-expressive general probabilistic programming language for the specification of statistical models. A Stan program computes the joint log probability density of a set of continuous parameters up to a proportional constant. Full Bayesian inference is carried out using Stan's No U-Turn Sampler (NUTS) which uses Hamiltonian Monte Carlo (HMC) to obtain a set of draws from the

```
1  functions {
2    real icar_normal_lpdf(vector phi, int N, int[] node1, int[] node2) {
3      return -0.5 * dot_self(phi[node1] - phi[node2])
4        + normal_lpdf(sum(phi) | 0, 0.001 * N);
5   }
6  }
7  data {
8    int<lower=0> N;
9    int<lower=0> N_edges;
10   int<lower=1, upper=N> node1[N_edges];  // node1[i], node2[i]
       neighbors
11   int<lower=1, upper=N> node2[N_edges];  // node1[i] < node2[i]
12 }
13 parameters {
14   vector[N] phi;
15 }
16 model {
17   phi ~ icar_normal_lpdf(N, node1, node2);
18 }
```

Listing 1. Program icar.stan.

posterior. HMC samplers are more efficient and robust than Gibbs and Metropolis samplers (Hoffman and Gelman, 2014), allowing for better estimates of models with complex posteriors such as the BYM model.

3.1. First Stan program: `icar.stan`

A Stan program consists of a set of named program blocks which occur in a fixed order. Stan is an *imperative* programming language, thus the variable declarations and statements in program blocks and user-defined functions are executed in program order. Stan is a strongly-typed language, i.e., variable declarations specify the variable type and all operations must respect the declared variable type. Variables must be declared before they can be referenced. Data variables are declared in `data` and `transformed data` blocks. Parameter variables are used in `parameters` and `transformed parameters` blocks. Declarations and statements are terminated with a semicolon (`;`). Comments are delimited by a pair of forward slash characters (`//`) and continue through to the end of the line.

Listing 1 presents a Stan program which computes the ICAR spatial random variable ϕ given a set of neighboring regions.

Lines 2–5 of `icar.stan` define a custom distribution function `icar_normal_lpdf` for an ICAR random variable `phi`. This function computes the ICAR prior as the pairwise difference of neighboring elements of `phi` and enforces the sum-to-zero constraint. The function name ends in `_lpdf` which signals that this function defines a log probability density function. `_lpdf` functions have the signature of return type real and the first argument is either type real or type array of reals. It takes the following arguments:

- the spatial random variable `phi`
- `N`, the number of areal regions
- integer array `node1`
- integer array `node2`

Together `node1` and `node2` encode the neighbor relationships as a graph edgeset: `node1` holds the set of indexes corresponding to ϕ_i and the `node2` holds the indexes corresponding to ϕ_j, where $i < j$. To see how this works, in the example in the previous section, there are 4 regions labeled 1 through 4 and 3 edges:

	node1	node2
edge 1	1	2
edge 2	2	3
edge 3	3	4

Encoding the neighbor relations as an edgeset requires less memory than specifying a full $N \times N$ adjacency matrix when the adjacency matrix is sparse. In our small example, the adjacency matrix has 16 elements. The edgeset requires scalar variables N, and N_edges, and 2 parallel arrays of indices for a total of 8 elements. In general, for a neighborhood of N regions where the average number of neighbors for a region is K, the space required to store an edgeset is $N \times K$, where $K \leq N$. As N increases, K usually remains constant, thus the edgeset encoding is more efficient.

Because Stan provides vectorized operations as well as multi-index expressions, line 3 of the body of function `icar_normal_lpdf`

$$-0.5 * \texttt{dot\_self}(\texttt{phi}[\texttt{node1}] - \texttt{phi}[\texttt{node2}])$$

is the direct translation of the pairwise difference formula

$$p(\phi) \propto \exp\left(-\frac{1}{2}\sum_{i\sim j}(\phi_i - \phi_j)^2\right).$$

The entries in arrays `node1` and `node2` are indexes for `phi`. The expressions `phi[node1]` and `phi[node2]` are multiple indexing expressions; each evaluates to a vector of length `N_edges` whose entries are values of `phi` at the indices in `node1` and `node2`. Vector subtraction yields the vector of pairwise differences. The Stan math library function `dot_self` multiplies this vector by itself, the result is the sum of the squares of the pairwise differences.

The expression on line 4 enforces the sum-to-zero constraint on `phi`:

`normal_lpdf(sum(phi) | 0, 0.001 * N);` Since the random vector `phi` sums to zero, it follows that the mean of `phi` must also be zero, but instead of requiring the mean to be exactly

zero, this constraint "soft-centers" the mean by keeping it as close to zero as possible. This expression calls Stan's implementation of the normal probability density. The calling syntax for a probability density functions follows probability function notation so that a vertical bar is used to separate the outcome from the parameters of the distribution. The straightforward specification of this constraint is:

```
normal_lpdf(mean(phi) | 0, 0.001);
```

The mean is the sum of the vector elements divided by the vector length and division is a relatively expensive operation. By multiplying the location and scala parameters by the vector length, we remove the division operation from the formula.

For ICAR models the neighborhood structure comes into the model as data. Data variables are declared in the `data` block, (lines 7–12). The variables `N` and `N_edges` specify the size and range limits on the edgeset arrays `node1` and `node2`, therefore the former are declared before the latter. Constraints on the range of allowed values for a variable follow the variable type name in the variable declaration. Because the variables `N` and `N_edges` hold size information they are constrained to be greater than or equal to 0. The edgeset arrays are indexes over the N areal regions, therefore these are constrained to be between 1 and N. These constraints are enforced when the data is read in during model instantiation.

The ICAR spatial random variable ϕ is declared in the `parameters` block, lines 13–15, and the `model` block, lines 16–18 computes the log probability density up to a proportional constant.

The `model` block computes the total log probability density by specifying the distribution of `phi` (line 17) using a *sampling statement*. A sampling statement specifies that the expression on the left hand side of the symbol $\sim$ is distributed according to the right hand side log probability density function, (or log probability mass function for discrete distributions). Despite the name, this statement doesn't actually perform sampling; it is functionally equivalent to incrementing the total log probability density by the value returned by calling the equivalent log probability density function with the with left hand side expression as the first argument. E.g., line 17 is equivalent to incrementing the total log probability density by the value returned by : `icar_normal_lpdf(phi | N, node1, node2)`

3.2. Fitting Stan models to data with RStan

We use the R package `rstan` to fit the ICAR model to the map of areal regions used in the case study in the following section. This package contains functions to compile and fit Stan models, generate reports, and save and reload model fits. The function `stan` compiles a Stan program to C++, instantiates the compiled model together with the inputs specified in the `data` block of the Stan program, and then runs the HMC sampler to produce a set of draws from the target log probability density specified by the `model` block.

The spatial data in our example consists of the neighborhood graph over the New York City 2010 census tracts. After downloading the geographic datafiles from the US Census Bureau, we used the R package `spdep` to get a list of all neighbors for the census tracts in the case study (n=2095). The neighborhood graph was edited to create a fully connected graph. Finally, we transformed the list into the set inputs to match the to the data variables in program `icar.stan`: integer variables `N` and `N_edges`, and integer array variables `node1` and `node2`, the graph edgeset. The file `bym2_nyc_data.R` contains these inputs.

The script `fit_icar_nyc.R` sets up the R environment, loads the data, fits the model to the data, and provides diagnostics and a summary of the resulting sample.

```
# fit model icar.stan to NYC census tracts neighborhood map
library(rstan);
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())
source(file=''bym2_nyc.data.R'');
icar_nyc_stanfit = stan(''icar.stan'',
 data=list(N,N_edges,node1,node2),
 control=list(max_treedepth=15));
check_hmc_diagnostics(icar_nyc_stanfit);
print(icar_nyc_stanfit, probs=c(0.25, 0.75), digits_summary = 1);
```

The `stan` function returns a `stanfit` object which contains both the posterior draws and sampler diagnostic values produced by each chain used to detect problems with the model fit. The arguments to the `stan` function specify the Stan program file, the list of data variables, and controls to the sampler. The default settings will use the NUTS HMC sampler to run 4 chains for 2000 iterations, where the first 1000 iterations are warmup and the last 1000 iterations are saved as output, producing a sample consisting of 4000 draws from the posterior.

The `check_hmc_diagnostics` function checks that the sampler was able to able to fully and effectively explore the joint distribution specified by the model. It reports on

- *divergences*, which signal that the HMC sampler cannot adequately explore all regions of the posterior, resulting in a biased sample. Increasing the sampler's `adapt_delta` control can sometimes resolve this problem, otherwise it may be necessary to reparameterize the model. See (Betancourt, 2017).
- *treedepth* - iterations which exceed maximum treedepth result in slow sampling time; to resolve this, increase the treedepth via the sampler's `max_treedepth` control. In this example, it was necessary to increase the `max_treedepth` control above 12 in order to eliminate these warnings.
- *E_BFMI* - the Bayesian Fraction of Missing Information for each chain. This can sometimes be resolved by increasing the number of warmup iterations, otherwise it may be necessary to reparameterize the model.

The `print` function returns a set of summary statistics, described in the RStan vignette Accessing the contents of a stanfit object.

> The summary is a matrix with rows corresponding to parameters and columns to the various summary quantities. These include the posterior mean, the posterior standard deviation, and various quantiles computed from the draws. ... For models fit using MCMC, also included in the summary are the Monte Carlo standard error (se_mean), the effective sample size (n_eff), and the R-hat statistic (Rhat).

Here we call the `print` function with optional arguments `probs` and `digits_summary`. The `probs` argument specifies that only the 0.25 and 0.75 quantile estimates should be displayed. In this example, 1 digit of precision is sufficient to check the summaries for all elements of parameter `phi`. This call returns the following.

```
Inference for Stan model: icar.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.
 mean se_mean sd 25% 75% n_eff Rhat
phi[1] 0.0 0.0 0.8 -0.5 0.6 3354 1
phi[2] 0.0 0.0 0.8 -0.5 0.5 3130 1
 ...
phi[2094] 0.0 0.0 1.3 -0.9 0.9 2110 1
phi[2095] 0.0 0.0 1.3 -0.8 0.9 2058 1
```

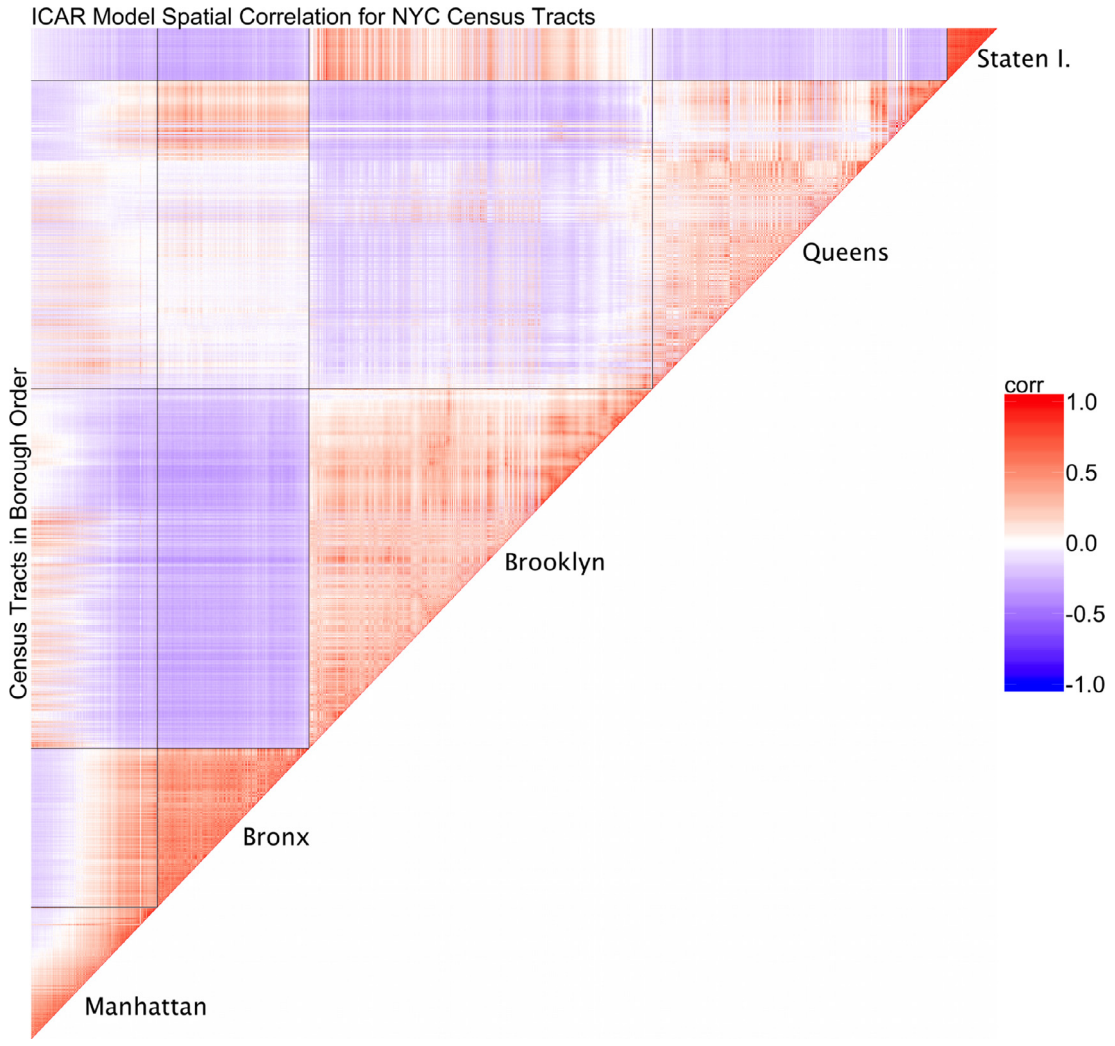


Fig. 1. Correlation matrix for NYC census tracts ordered by Borough, tract ID.

Although the summary information is designed to be read from left to right, the column which should always be checked first is the rightmost column labeled `Rhat`. The R-hat statistic is a measure of convergence. When a chain fails to converge, the draws returned by the sampler are not a sample from the posterior distribution and cannot be used for estimation. All R-hat values should be extremely close to 1 and values greater than 1.1 are an indication that one or more of the chains have failed to converge during warmup. All values in the `Rhat` column of the summary are 1, indicating that the chains have converged.

The second column from the right labeled is labeled `n_eff`. The number of effective samples (N_{eff}) is the number of independent samples with the same estimation power as the N autocorrelated samples. An MCMC sampler produces an *estimate* of the mean. The error in that estimate depends on the number of effective samples N_{eff}. The column `se_mean` is the Monte Carlo sampler error (MCSE) which is proportional to $1/\sqrt{N_{eff}}$ instead of $1/\sqrt{N}$. It is computed by $sd/\sqrt{n_{eff}}$. As N_{eff} increases, the MCSE approaches 0 and the estimated parameter mean approaches the true mean. Conversely, when N_{eff} is low, so is the precision of the estimate. In this example, all N_{eff} values are above 1200 with median value 2598, which is sufficient to estimate all parameters with reasonable precision.

Since the R-hat and (N_{eff}) statistics indicate that this is a valid sample with sufficient number of independant draws to estimate all parameters, the next step is to check the estimates for all parameters in the model. The parameter `phi` is a multivariate normal random variable centered at zero with precision matrix Q. The `print` summary column labeled `mean` shows that the estimated mean for all elements of the vector `phi` are zero, indicating that this Stan program is a correct implementation of the ICAR model.

The spatial structure implied by the ICAR prior `phi` is encoded in its covariance matrix Σ. Since the mean of each ϕ_i depends on that of its neighbors, we expect to see high co-variance between neighboring regions and we expect co-variance between non-neighboring regions to be close to zero. To check this we extract the set of draws for the vector `phi` from the fitted ICAR model using RStan's `extract` function. Instead of working with the covariance matrix we use the correlation matrix which standardizes the range of values to $[-1, 1]$. We use the R package `ggplot2` to plot the results.

The elements of `phi` are ordered by a numeric ID which consists of a borough code followed by the census tract ID. The borough codes impose the following order on the five boroughs of New York City: Manhattan, Bronx, Brooklyn, Queens, Staten Island. The borough code ordering doesn't correspond to the neighborhood graph over the boroughs; i.e., the Bronx is not adjacent to Brooklyn, Queens doesn't share a border with Staten Island. By plotting the correlation matrix in input order without further clustering we expect to see that within a borough, adjacent elements n and $n+1$ are likely to be neighbors and should be positively

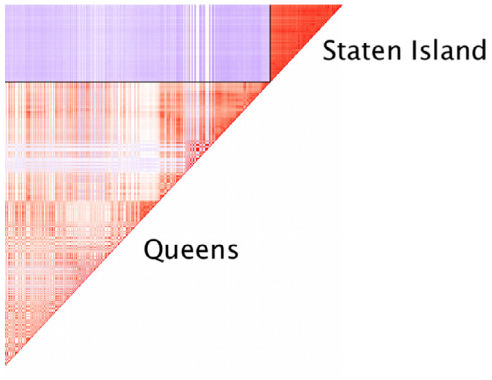


Fig. 2. Correlations between Queens and Staten Island Census Tracts.

correlated, while across boroughs elements are unlikely to be neighbors and should only be weakly correlated or anti-correlated. Because this matrix is symmetric, we only show correlations for the upper triangular matrix.

In this plot, black lines mark the divisions between the five boroughs. Bright red indicates strong correlation, white indicates no correlation, dark violet indicates anti-correlation, and pale red or violet indicates weak correlation or anti-correlation, respectively. The overall pattern shows high correlation (bright red) within each borough and weak anti-correlation (pale violet) between boroughs (Fig. 1).

The boroughs of Staten Island and Queens have no common border. Zooming in on the upper right corner, the census tracts of Staten Island and part of Queens, show the pattern of high correlation within each borough and weak anti-correlation between the two boroughs.

These diagnostics and plots validate the Stan model fit of the New York City neighborhood graph to the ICAR prior (Fig. 2).

3.3. Second Stan program: Bym2.stan

The Stan program `bym2.stan` implements the BYM2 model for a fully connected neighborhood graph. The log probability density is a Poisson GLM with a fixed intercept and vector of coefficients together with a combined random effects component consisting of an ICAR model for spatial smoothing and an ordinary random effects component for non-spatial heterogeneity. This combined random effects component is scaled by a parameter for the overall standard deviation. The Poisson regression is specified in the `model` block as

```
y ~ poisson_log(log_E + beta0 + x * betas + convolved_re * sigma);
```

The combined random effects components is specified as:

$$(\sqrt{\rho/s})\,\phi^* + (\sqrt{1-\rho})\,\theta^*.$$

This is coded as the variable `convolved_re` which is declared and defined in the `transformed parameters` block:

```
convolved_re = sqrt(rho / scaling_factor) * phi
 + sqrt(1 - rho) * theta;
```

In a Stan program, the `model` block contains the specification of the likelihood and priors. The `parameters` block is a declarations-only block; parameters are declared here and constraints are specified in the model block. The `transformed parameters` block uses the set of proposed (unconstrained) parameters to compute derived values; these derived values are used to compute the likelihood. The declarative nature of a Stan program makes it easy to see the role that every variable plays in the model, however for complicated models, implementation logic is spread across the program blocks, as is the case here.

The complete implementation of the BYM2 model is shown in Listing 2.

The `functions` block (lines 1–6) contains the definition of the function `icar_normal_lpdf` which computes the ICAR prior, as in the program `icar.stan`.

The `data` block (lines 7–17) contains the definitions for the four variables which specify neighborhood structure, as well as the data and outcomes from the disease mapping study, i.e., the observed counts per region, population (offset), number of regions, dimensions of the vector of covariates, design matrix.

The `transformed data` block (lines 18–20) puts the offset term on the log scale. The data and transformed data blocks only executed once, when Stan instantiates the model together with the data. This program specifies the scaling factor as data; however it is possible to compute this directly from the neighborhood graph, in which case, this variable would be declared in the `transformed data` block, along with the statements required to compute this value.

The ensemble of the `parameters` block (lines 21–28), `transformed parameters` block (lines 29–33), and `model` block (lines 34–42), specifies the model parameters and the likelihood and priors. For every step of the sampler, the statements in the transformed parameters and model block are computed in order using the set of proposed (unconstrained) parameters. The total log probability density is incremented by the sampling statements in the `model` block.

The `generated quantities` block (lines 43–56) computes additional quantities of interest. This block is executed once per iteration, at the point where the sampler proposal has been accepted. The quantities of interest are computed using the (unconstrained) values of all parameters and transformed parameters for that draw. Here we use the generated quantities block to generate two quantities of interest based on the parameters for that draw:

- `mu` - the estimated of input `y` (lines 44–45)
- `y_rep` - an estimate of new data $\tilde{y}$ (lines 46–55).

On line 54 we use Stan's `poisson_log_rng` function to generate a new observation `y_rep` based on the data and estimated parameters for that draw. Lines 47–51 guard against potential numerical problems which may occur during warmup. We use the generated `y_rep` values and the Stan's bayesplot package for R to carry out posterior predictive checks (PPC), (Gabry et al., 2017). PPC is a model checking procedure in which a model is used to generate new data from the current data. From Gabry et al.:

> The idea behind posterior predictive checking is simple: if a model is a good fit we should be able to use it to generate data that resemble the data we observed.

In the case study section we fit the BYM2 model with the New York City motor vehicle crash data, a dataset consisting of areal observations for 2095 census tracts and then use the PPC model checking procedure to evaluate the fit.

3.4. Comparison of Stan to ICAR model implementations in BUGS and INLA

We compared the Stan implementation of the ICAR prior to the corresponding models in BUGS and INLA using the Scotland lip cancer dataset, first discussed in (Clayton and Kaldor, 1987), which is available from the R INLA package as dataset Scotland. For this comparison we used a simpler model than the model we recommend in this paper, as that model (including the prior specification) cannot be easily fit in BUGS.

```
functions {
  real icar_normal_lpdf(vector phi, int N, int[] node1, int[] node2) {
    return -0.5 * dot_self(phi[node1] - phi[node2])
      + normal_lpdf(sum(phi) | 0, 0.001 * N);
 }
}
data {
  int<lower=0> N;
  int<lower=0> N_edges;
  int<lower=1, upper=N> node1[N_edges];  // node1[i], node2[i] are nbs
  int<lower=1, upper=N> node2[N_edges];  // node1[i] < node2[i]
  int<lower=0> y[N];              // count outcomes
  vector<lower=0>[N] E;           // exposure
  int<lower=1> K;                 // num covariates
  matrix[N, K] x;                 // design matrix
  real<lower=0> scaling_factor;  // scales the variance of the spatial
     effects
}
transformed data {
  vector[N] log_E = log(E);
}
parameters {
  real beta0;            // intercept
  vector[K] betas;       // covariates
  real logit_rho;        // proportion spatial variance
  vector[N] phi;         // spatial effects
  vector[N] theta;       // heterogeneous effects
  real<lower=0> sigma;   // overall standard deviation
}
transformed parameters {
  real<lower=0, upper=1> rho = inv_logit(logit_rho);
  vector[N] convolved_re = sqrt(rho / scaling_factor) * phi
                           + sqrt(1 - rho) * theta;
}
model {
  y ~ poisson_log(log_E + beta0 + x * betas + convolved_re * sigma);
  beta0 ~ normal(0, 1);
  betas ~ normal(0, 1);
  logit_rho ~ normal(0, 1);
  sigma ~ normal(0, 1);
  theta ~ normal(0, 1);
  phi ~ icar_normal_lpdf(N, node1, node2);
}
generated quantities {
  vector[N] eta = log_E + beta0 + x * betas + convolved_re * sigma;
  vector[N] mu = exp(eta);
  int y_rep[N];
  if (max(eta) > 20) {
    // avoid overflow in poisson_log_rng
    print("max eta too big: ", max(eta));
    for (n in 1:N)
      y_rep[n] = -1;
  } else {
      for (n in 1:N)
        y_rep[n] = poisson_log_rng(eta[n]);
  }
}
```

Listing 2. Program bym2.stan.

Table 1
Comparison of running times.

	User	System	Elapsed
Stan	0.46	0.17	11.73
BUGS	8.53	0.74	15.36
INLA	0.81	0.68	1.65

In order to compare estimates, we implemented approximately[1] the same model, using the same likelihood and priors for all platforms by reducing the BYM model to a simpler model consisting of a hierarchical Poisson regression with an ICAR component for spatial correlation as the model:

$$\eta_i = \mu + x\beta + \phi$$

where:

- μ is the overall risk level, i.e., the fixed intercept.
- x is the matrix of explanatory spatial covariates such that x_i is the vector of covariates for areal unit i and β is vector of regression coefficients which are constant across all regions, i.e., fixed effects.
- ϕ is an ICAR spatial component.

The programs, data, and scripts used to run these comparisons are included in the supplemental materials for this paper. We ran the same number of chains and iterations for both BUGS and Stan. Table 1 presents a comparison of the running times as reported by R's `system.time` command. Column "User" gives the CPU time spent by the R session, column "System" gives the CPU time spent by the operating system on behalf of the R process, and column "Elapsed" give the wall clock time taken to run the process. On this small dataset, Stan took less processing time than BUGS and INLA, but INLA took less time overall.

Table 2 presents the estimates for the slope, intercept, and ICAR component. BUGS and INLA compute the precision `tau` which is the inverse variance, i.e. $1/\sigma^2$, while Stan computes the standard deviation `sigma` which is the square root of the variance (σ^2), therefor we use Stan's `generated quantities` block to compute `tau` via the statement `real tau = sigma^-2;`. All three systems arrived at approximately the same estimates for the slope and intercept parameters of the linear regression as well as for all elements of `phi`.

3.5. Stan model extension and expansion

Because Stan is an expressive probabilistic programming language, it is fairly straightforward to modify the implementations presented in this paper to deal with more complicated data. For example, the BYM2 model can be extended to disconnected graphs using the ideas in (Freni-Sterrantino et al., 2018). Furthermore, it is a one line change to the code to replace the Poisson assumption on the data with other observation processes. The structure of Stan makes it possible for even every data point to have a different likelihood. Moreover, the BYM2 model can be nested as a component in **any** hierarchical or non-hierarchical Bayesian model. The only real restriction is that model parameters must be continuous variables, although some models, such as finite mixture models, which can be written with discrete random variables can also be expressed without them, (see the chapter on "Latent Discrete Parameters" in the Stan User's Guide (Stan Development Team, 2019)).

[1] The three models implement the sum to zero constraint differently. INLA and Stan use a soft centering approach, while BUGS subtracts the mean off each sample. The procedure in BUGS is known to be mathematically incorrect.

4. Case study: Youth pedestrian injuries in NYC, 2005–2014

In New York City (NYC) pedestrians account for approximately half of all traffic fatalities (Fung and Conderino, 2017). Small-area spatiotemporal modeling using Bayesian models such as the Besag-York-Mollie (BYM) model can be a useful tool to explore areas of high risk for pedestrian crashes and to evaluate the joint role of sociodemographic and traffic related risk factors (DiMaggio, 2015). This case study focuses on school-age pedestrian crashes using ten years of recent data from 2005–2014. We used the Stan platform to fit the BYM2 model to dataset consisting of census tract counts of school-age pedestrian crashes, exploring the effects of commuting patterns, vehicular traffic density, social fragmentation, and income.

4.1. Methods

4.1.1. Measures

We obtained motor vehicle collision data from the New York City Department of Transportation for the ten most recent years of data available at the time of request (2005–2014). Within this dataset, we identified collisions involving school age children 5–18 years of age as pedestrians. We then assigned each crash to the census tract in which it occurred, using boundaries from the 2010 United States Census.

We obtained 2010 US Census counts of youths aged 5–18 in each census tract from the US Census Bureau ("American Factfinder," n.d.) We also obtained the Census Bureau's American Community Survey (ACS) five-year estimates of median household income and the percentage of commuters who traveled to work by means other than a private vehicle (i.e., by walking, bicycling, or using public transportation) for each tract for 2010–2014. We constructed an index of social fragmentation based on the work of Peter Congdon (Congdon, 2012), as described in our previous study (DiMaggio, 2015), using updated ACS estimates of vacant housing units, householders living alone, non-owner occupied housing units, and population having moved within the previous year. We standardized each of these metrics with a mean of zero and added them together as a single index. Finally, we obtained street level annual average daily traffic (AADT) data from the New York State Department of Transportation on the New York Open Data portal ("Annual Average Daily Traffic (Aadt): Beginning 1977," n.d.). We created a spatial overlay of streets and census tracts to assign each census tract the maximum AADT value of its underlying streets in 2015.

We used the `spdep::poly2nb` R function to assign adjacency between census tracts, allowing water boundaries. We manually added contiguity between the Rockaway peninsula and the rest of Queens (which are separated by a Jamaica Bay, a large body of water) for a fully connected map. We excluded parks, cemeteries, and any other census tracts for which the population of children between ages 5 and age 18 was five or fewer resident children.

4.1.2. Analysis

We computed descriptive statistics and applied the BYM2 model in Stan to create smoothed estimates of youth pedestrian crash rates while quantifying the effects of pedestrian and public transit commute methods, traffic density, income, and social fragmentation. We log transformed both traffic counts and income in order to normalize their distributions, as the model initially failed to converge with the non-transformed data. The specification of the model was as follows, where the unit of analysis is census tracts:

- y = count of school age pedestrians ages 5–18 injured in traffic crashes.

Table 2
Comparison of parameter estimates.

	Mean			Lower 5% CI			Upper 95% CI		
	Stan	BUGS	INLA	Stan	BUGS	INLA	Stan	BUGS	INLA
beta0	−0.20	−0.20	−0.20	−0.41	−0.40	−0.40	−0.01	0.00	0.00
beta1	0.35	0.34	0.35	0.14	0.12	0.13	0.55	0.56	0.56
tau	1.81	1.78	1.80	1.01	1.04	1.03	2.88	2.78	2.86
phi1	1.17	1.17	1.18	0.68	0.69	0.70	1.65	1.63	1.66
phi2	1.11	1.12	1.11	0.81	0.81	0.80	1.41	1.42	1.42
phi3	1.02	1.03	1.05	0.55	0.56	0.57	1.46	1.47	1.49
phi55	−0.57	−0.56	−0.55	−1.10	−1.12	−1.08	−0.09	−0.08	−0.06
phi56	−0.47	−0.48	−0.46	−1.01	−1.00	−0.99	0.03	0.02	0.04

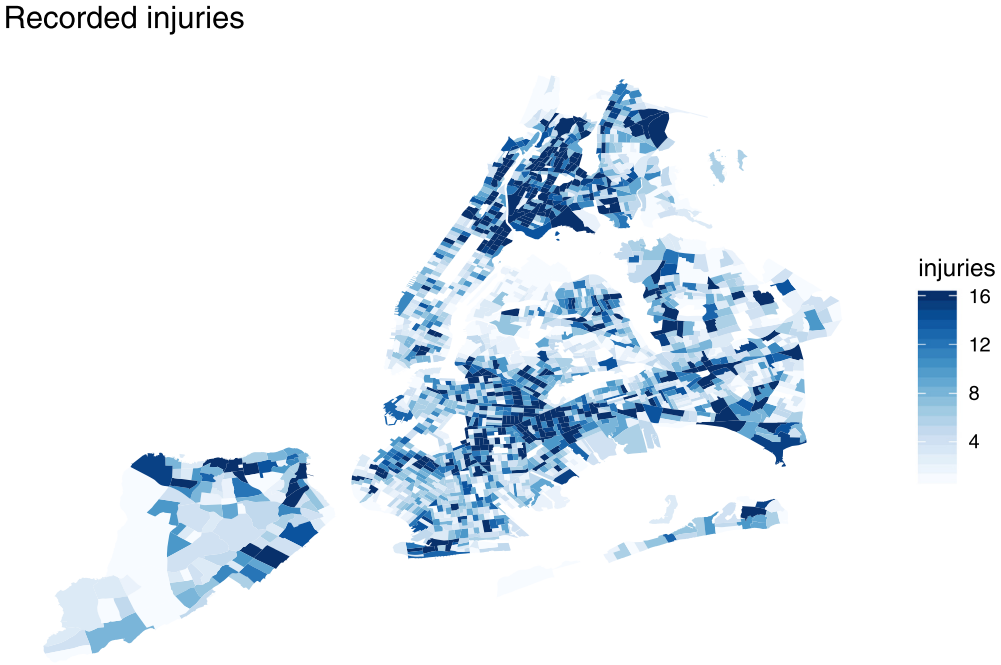


Fig. 3. School age pedestrians injured in traffic crashes, NYC 2005–2014.

- x1 ("pct_commute") = percent commuters using means other than private vehicle (i.e. walking, bicycling, or public transit).
- x2 ("log_income") = log of median household income.
- x3 ("std_frag_index") = standardized index of social fragmentation (vacancy, rentals, living alone, recently moved).
- x4 ("log_aadt") = log of maximum AADT value in each tract in 2015.
- an offset term for the youth population ages 5–18 in each census tract.
- the BYM2 convolved random effects term, comprising parameters `rho`, `phi`, and `theta`, the proportion of spatial variance, the spatial ICAR term, and the non-spatial vector of normal random variates, respectively.
- the overall variance of the convolved random effects term `sigma`.

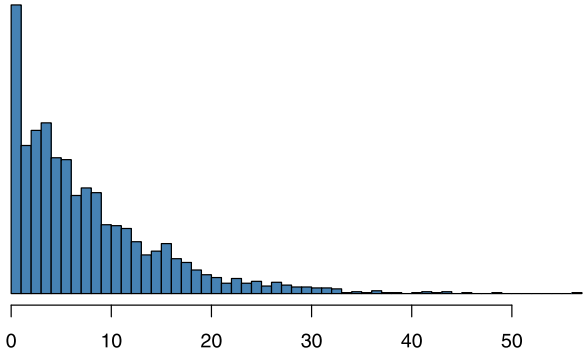


Fig. 4. Histogram of school-age pedestrian injury counts per census tract, NYC 2005–2014.

4.2. Results

4.2.1. Descriptive statistics

From 2005–2014 there were 17,529 crashes (1,753 per year, on average) injuring school age pedestrians in NYC (Fig. 3) of which 17,193 (98.1%) occurred in populated census tracts. There was a range of 0 to 57 and a median of 6 such crashes per census tract (Table 3), which exhibited a strong Poisson distribution (Fig. 4).

Table 3 summarizes the distribution of pedestrian injuries (ages 5–18 yrs) and sociodemographic measures by NYC census tracts

Table 3
Distribution of measures by census tract.

	med	min	mean	max
Youth pedestrian injuries, 2005-14	6	0	8.2	57
Population ages 5–18 years, 2010	510	6	596.4	3,315
Med. household income in USD, 2010-14	$53,890	$9,327	$58,497	$232,266
Pct. commute by walk/cycle/public trans, 2010-14	73.9	9.7	69.8	100
Standardized social fragmentation index	−0.1	−6.7	0	18.7
Traffic Volume (AADT), 2015	19,178	843	37,248	276,476

having youth population > 5 (n=2095). Median household income ranged in these tracts from $9,000 to $232,000 in 2010–2014, with a median of $53,890. The proportion of workers who traveled to work by means other than a private vehicle (e.g. walking, bicycling, or taking public transportation) ranged from 10% to 100%, and was heavily right-skewed with most census tracts having >50% of workers commuting by walking, bicycling, or using public transportation. By definition, social fragmentation was centered around zero. The census tract maximums of annual average daily traffic volumes (AADT) per underlying segment within each tract ranged from 800 to 277,000 vehicles per day with most tracts having maximum AADT counts below 50,000 vehicles. Histograms and maps of each of these measures are included in a supplemental appendix.

4.2.2. Model results

To fit the BYM2 model to the New York City pedestrian crash data using RStan we ran 4 chains of 2000 iterations each where the first 1000 draws were warmup and the last 1000 draws were saved as output for a total of 4000 draws from the posterior. Running the 4 chains in parallel on a MacBook Pro-laptop computer with a dual-core 3.1 GHz processor and 16GB of memory took 21 minutes, as measured by R's `proc.time` function. The reported elapsed time (time from start to finish) was 1296 seconds and the user processing time (total time across all threads) was 4839 seconds, thus running 4 chains sequentially would take 4 times as long. The RStan function `check_hmc_diagnostics` found no problems encountered by the sampler and the RHat values for all parameters were extremely close to 1.0, indicating that the model had successfully converged.

To carry out posterior predictive checking on the fitted model we first obtain the generated quantity `y_rep` from the stanfit object, and then use Stan's bayesplot (R) package to generate a visual comparison of the data y and the simulated new data $\tilde{y}$ using the ppc_dens_overlay function (Fig. 5).

The dark line is the distribution of the observed outcomes `y` and each of the 50 lighter lines is the kernel density estimate of one of one of the rows in `y_rep`. The lighter overlays follow the distribution of `y`, with a tendency towards the mean value. To further investigate, we use the ppc_stat function which plots the distribution of `y_rep` against the distribution of the data `y` for some test statistic. The default test statistic is the mean, therefore we run the command as: `ppc_stat(y, y_rep)`. As with the density plot, the test statistic plot indicates that the BYM2 model fits the data.

Table 4 shows the summary of the parameter estimates from the fitted model. Because the commute data is recorded as a percentage between 0 and 1 and not on the scale 0 to 100, it is necessary to divide the "pct_commute" regression coefficient by 100 in order to properly interpret its contribution. Thus for every percentage point increase in population commuting by means other than a private vehicle, there was a $\exp(0.005) = 0.5\%$ increase in the expected count of youth pedestrian injuries, controlling for income, vehicular traffic, social fragmentation, and population. The

Table 4
Parameter estimates from BYM2 model.

	mean	se_mean	sd	X2.5	X97.5	N_eff	R.hat
intercept	−3.5	0	0.5	−4.5	−2.5	1255	1
commute	0.5	0	0.2	0.2	0.9	777	1
log income	−0.1	0	0.0	−0.2	0.0	1204	1
std frag index	0.1	0	0.0	0.0	0.1	1527	1
log traffic	0.0	0	0.0	0.0	0.0	2551	1
rho	0.4	0	0.1	0.3	0.5	219	1
sigma	0.8	0	0.0	0.8	0.9	301	1

credible interval ranged from a $\exp(0.002) = 0.2\%$ to $\exp(0.009) = 0.9\%$ increase in pedestrian injuries per percentage point increase in on-foot commuters. There was a 1.2% decrease in youth pedestrian injuries per 10% increase in median household income. Social fragmentation was also significantly associated with youth pedestrian injuries, with an $\exp(0.1) = 10\%$ increase in youth pedestrian injuries per standard deviation increase in the combined index (i.e. vacancy, non-owner occupied housing, recent moves, and householder living alone), controlling for other model covariates. The credible interval for the effect of daily traffic included zero in our fully adjusted model after controlling for social fragmentation and pedestrian/bicyclist/public transit commute rates. The parameter sigma, the overall variance of the combined random effects term was $\exp(0.8) = 2.2$, indicating substantial overall variance. Nearly half of that variance, parameter rho, was spatially structured $\exp(0.4) = 49\%$.

From the fitted model we also obtain the quantity of interest `mu`, which is the estimate of school-age pedestrian injuries for each populated census tract (Fig. 6). Because BYM models contain both a spatial and non-spatial random effects component, they are able to account for almost all of the over-dispersion not modelled by the Poisson variates (Fig. 7). The overall burden of youth pedestrian injuries was most heavily concentrated in the Bronx, northern Manhattan/Harlem, and central Brooklyn, as well as some pockets of Queens and Staten Island (Fig. 8).

4.3. Discussion of the case study

NYC has embraced initiatives such as the national Safe Routes to School program ("Safe Routes," n.d.) and Vision Zero ("Vision Zero: Traffic Safety by Sweden," n.d.) in order to build on progress to date, recognizing that no traffic fatality should be considered acceptable. Our analysis explored spatial associations with traffic crashes injuring school age pedestrians in NYC, while illustrating the utility of Stan for computationally-intensive hierarchical modeling.

We found that income and social fragmentation were significant predictors of risk; that is, accounting for traffic and pedestrian-dominant travel, there was an effect of transience in low-income communities having high levels of rentals, vacancies, relocations, and residents living alone. All of these data are publicly available and frequently updated via the American Community Survey for census tracts throughout the US, which may provide

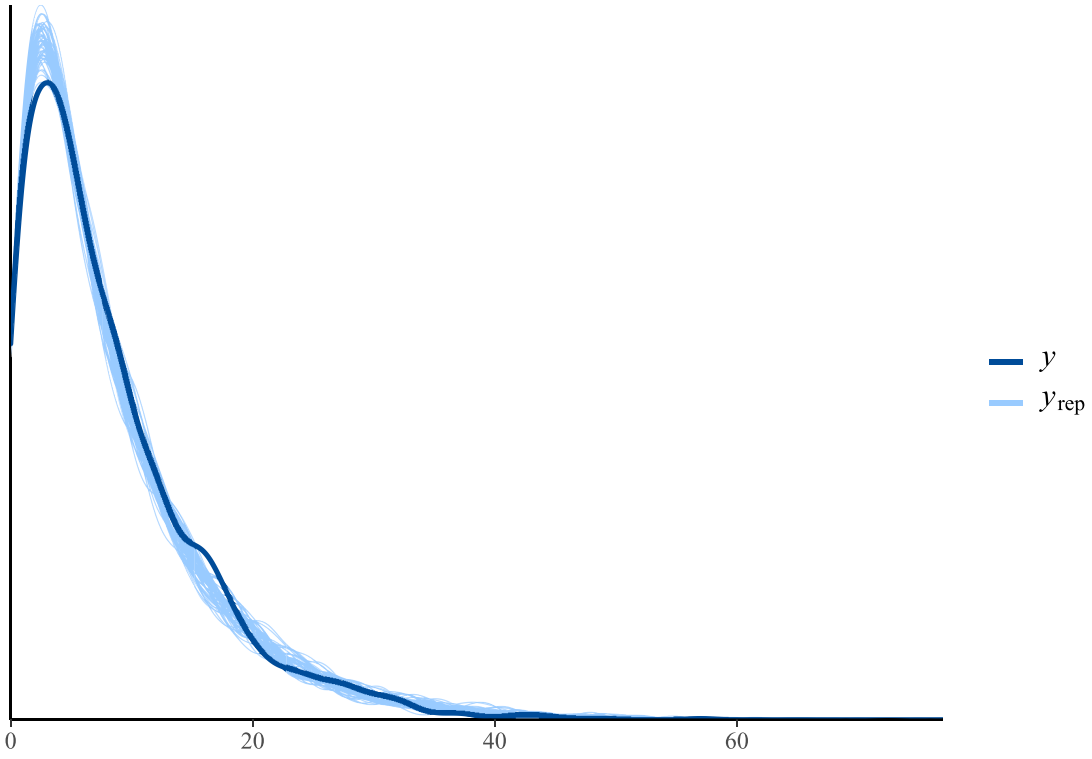


Fig. 5. Posterior predictive check, density overlay.

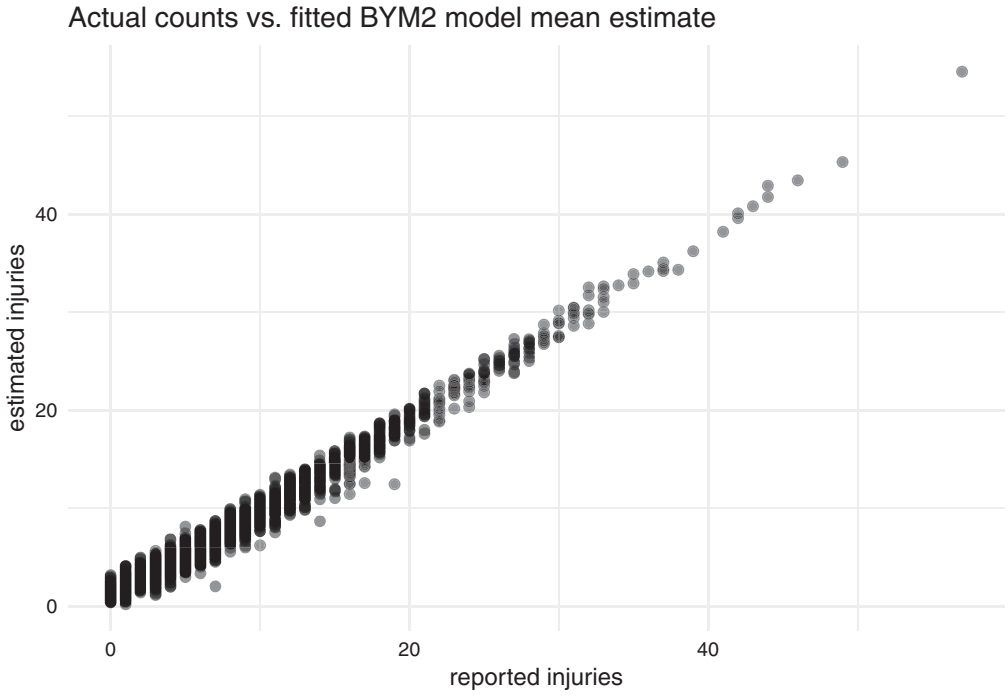


Fig. 6. Per-tract injuries, actual counts vs. fitted BYM2 model estimates.

communities with a useful tool to help identify areas of increased risk for pedestrian injury in the absence of other readily available data.

Spatial correlation, that is the tendency of higher rates of injury to cluster around other areas with high injury rates, played a moderate role in risk prediction for youth pedestrian injuries in NYC. About half of the random variance in our model was attributable to spatial correlation, accounting for commute method, income, traffic, and social fragmentation. Accordingly, the fitted map demonstrated gradual spatial smoothing, lending stability to the visualization of areas of high risk for a relatively rare outcome at the fine spatial scale of census tracts.

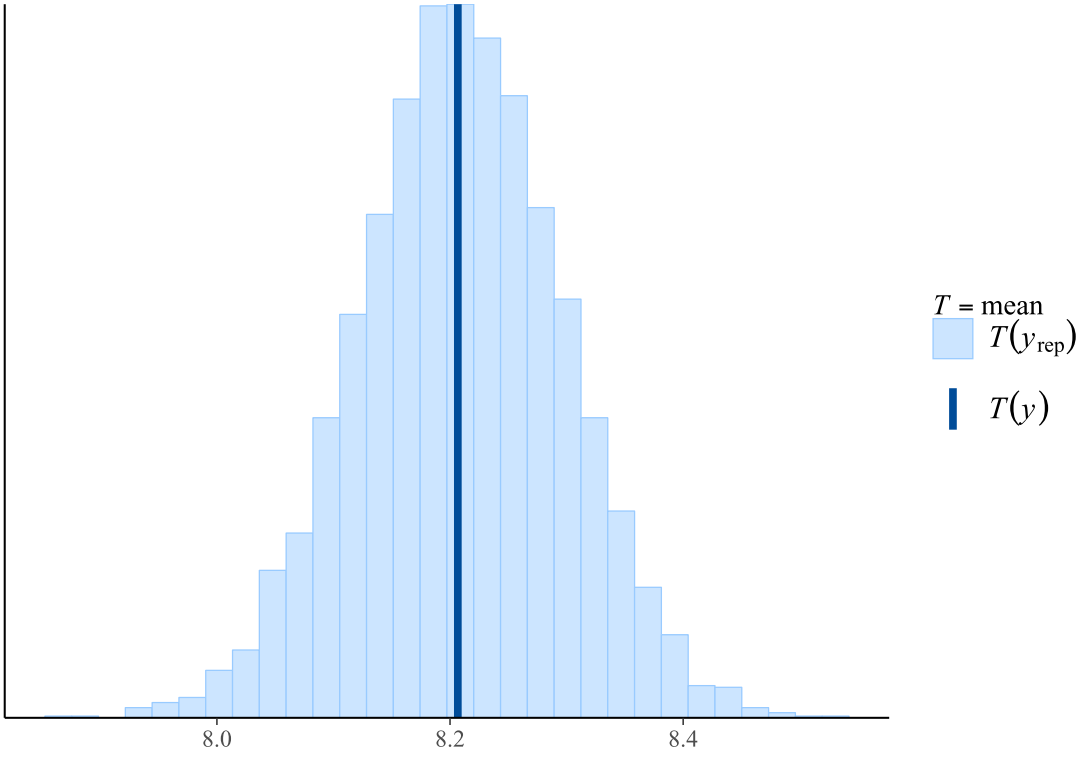


Fig. 7. Posterior predictive check, test statistic.

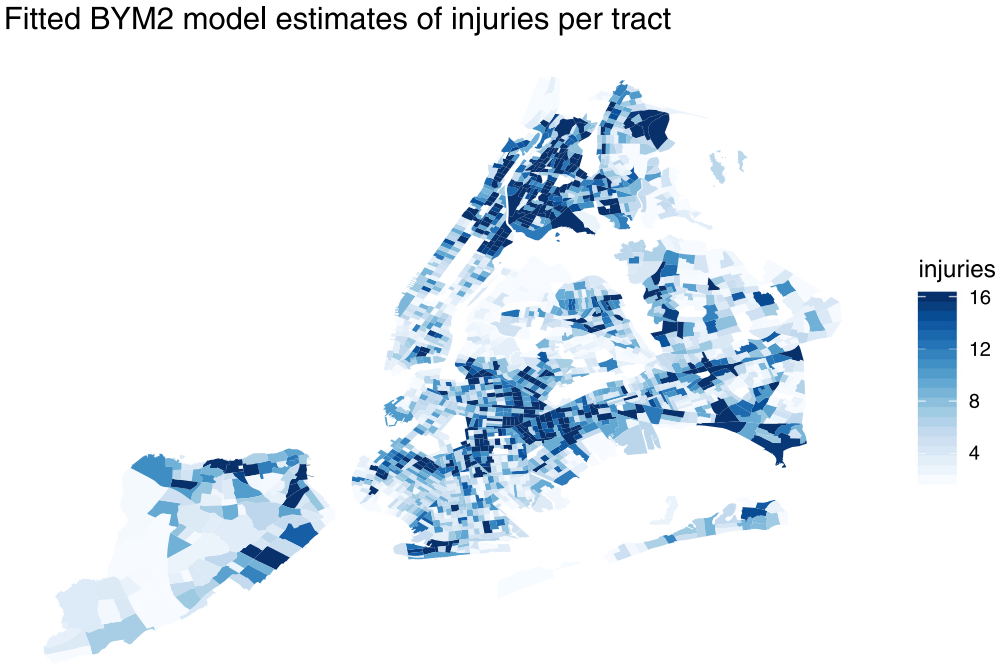


Fig. 8. Fitted BYM2 model estimated counts of school-age pedestrian crash injuries, 2005–2014.

Despite the complexity of our model and the large number of samples, the model successfully converged in Stan in 21 minutes on an ordinary desktop computer. Moreover, as described in the model development portion of this manuscript, we believe the use of Hamiltonian Monte Carlo simulation provides an improved method of sampling the posterior distribution compared with Gibbs and similar random-walk style Markov chain sampling.

Notably, the neighborhoods of central Brooklyn, the Bronx and northern Manhattan are more predominant in our map of youth pedestrian injuries compared with maps of total pedestrian injuries, which are relatively more concentrated in central and lower Manhattan (Viola et al., 2010). One primary reason for the differences between youth and all-age pedestrian injury maps is most likely the influence of daytime commuter population influx in central and lower Manhattan, as noted by Viola et al. (2010).

Areas having high frequencies of youth pedestrian injuries also tend to overlap with areas having the largest youth populations in NYC.

Because our analysis included a population offset term, we excluded large parks such as Central Park in Manhattan, Prospect Park in Brooklyn, and Van Cortlandt Park in the Bronx, even though such parks have both vehicular and pedestrian travel. Census tracts, moreover, are defined for the purposes of counting residents, and their boundaries do not necessarily have etiologic relevance to the study of pedestrian injuries (i.e. there can be diverse road types and traffic patterns within a census tract). In our previous work, for example, we have demonstrated the utility of virtual street audits to identify specific features of the built environment associated with pedestrian crashes at smaller spatial scales (Mooney et al., 2016). Future work should explore the impacts of infrastructure and other safety interventions on localized pedestrian crash rates, capitalizing on the efficiency of BYM modeling in Stan.

4.4. Conclusion of the case study

Stan proved to be an efficient and precise platform to build a hierarchical spatial model for youth pedestrian injuries in NYC. We confirmed prior findings that neighborhoods with higher social fragmentation and lower median incomes are disproportionately affected by pedestrian injuries. Our findings also demonstrate that the proportion of workers commuting to work by walking, bicycling, and public transit is correlated with youth pedestrian risk. This nationally and publicly available metric may serve as a useful surrogate index of pedestrian density in the absence of other readily available data. Finally, the performance and results obtained using Stan demonstrate its utility and strength for future spatial and spatiotemporal epidemiologic research, especially with large datasets.

4.5. Disclaimer

This report utilizes information which was originally compiled by the New York City Department of Transportation (DOT) for governmental purposes; the information has subsequently been stratified and aggregated for analysis by the authors of this manuscript. DOT and the City of New York make no representation as to the accuracy or usefulness of the information provided by this application or the information's suitability for any purpose and disclaim any liability for omissions or errors that may be contained therein.

Appendix A

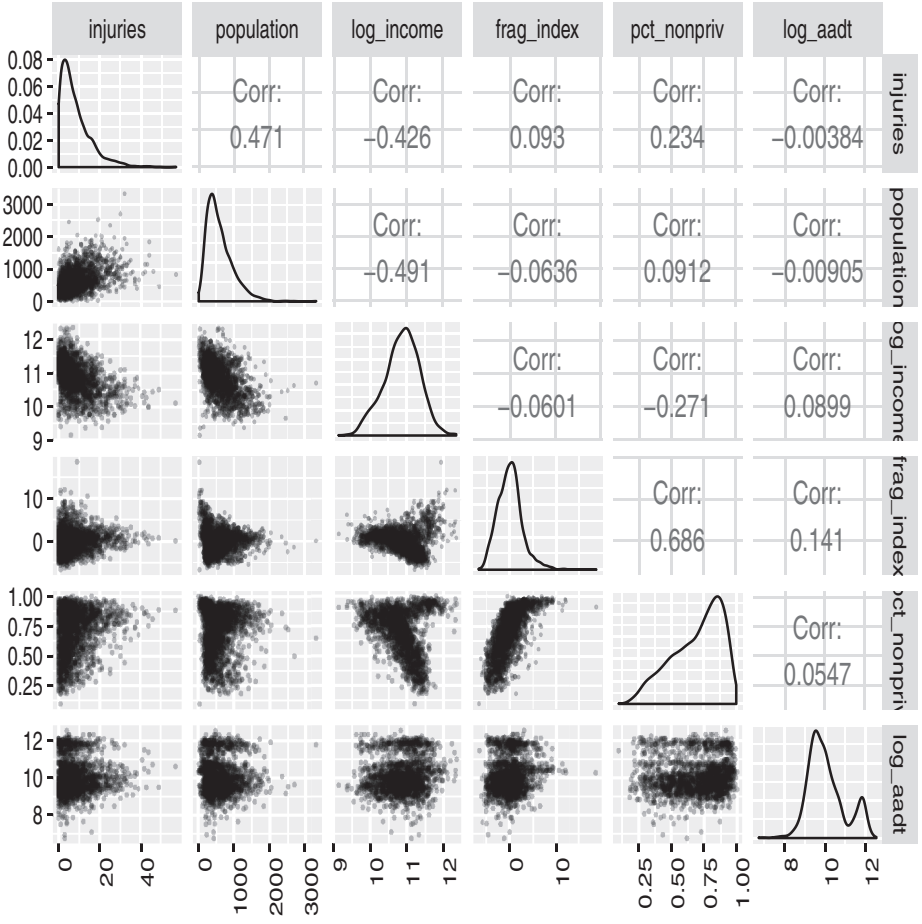


Fig. 9. Pairs plot of NYC case study regression data: injuries, population, predictors.

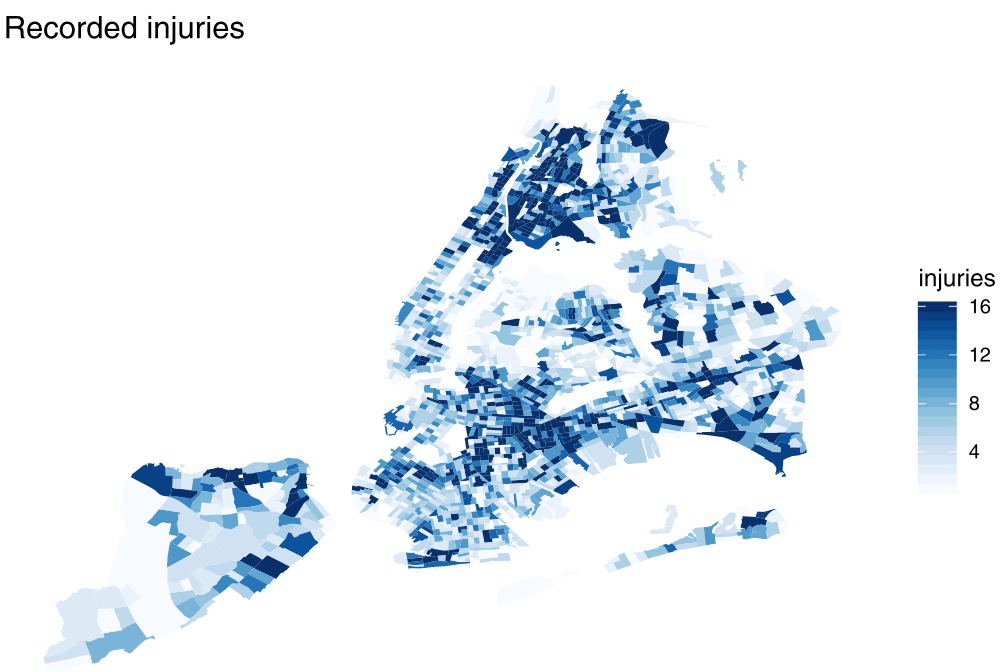


Fig. 10. Areal map of reported injuries per census tract, NYC 2005–2014.

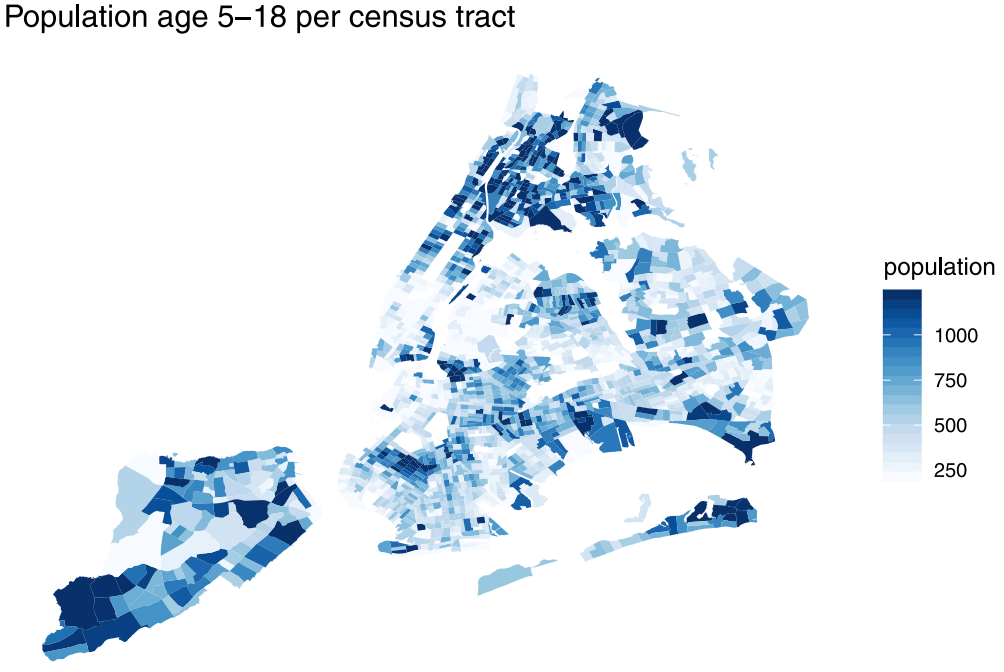


Fig. 11. Areal map of school-age population per census tract, NYC 2005–2014.

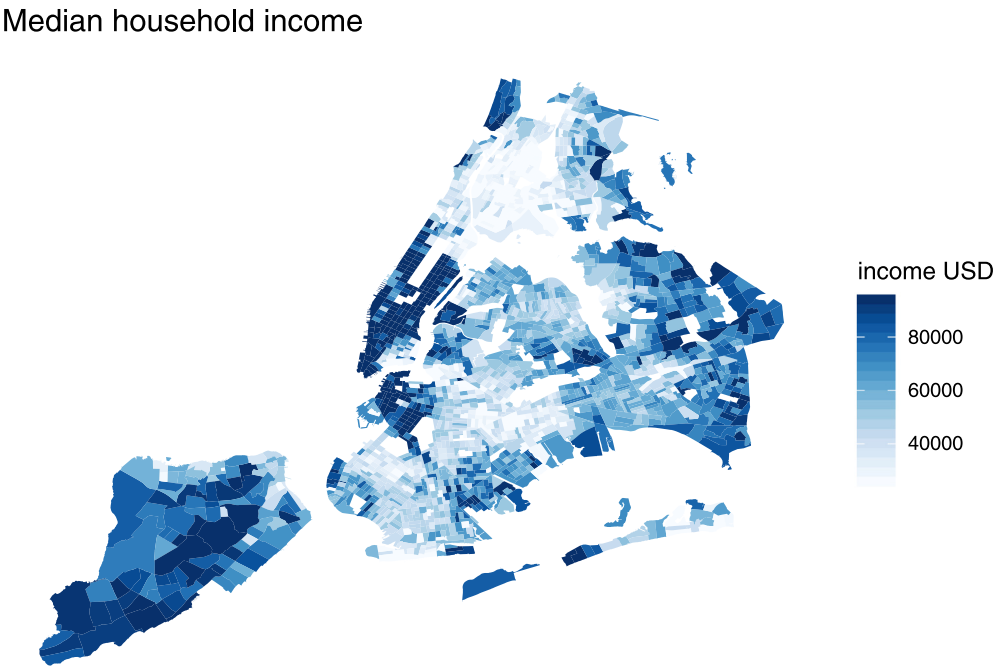


Fig. 12. Areal map of median household income per census tract, NYC 2005–2014.

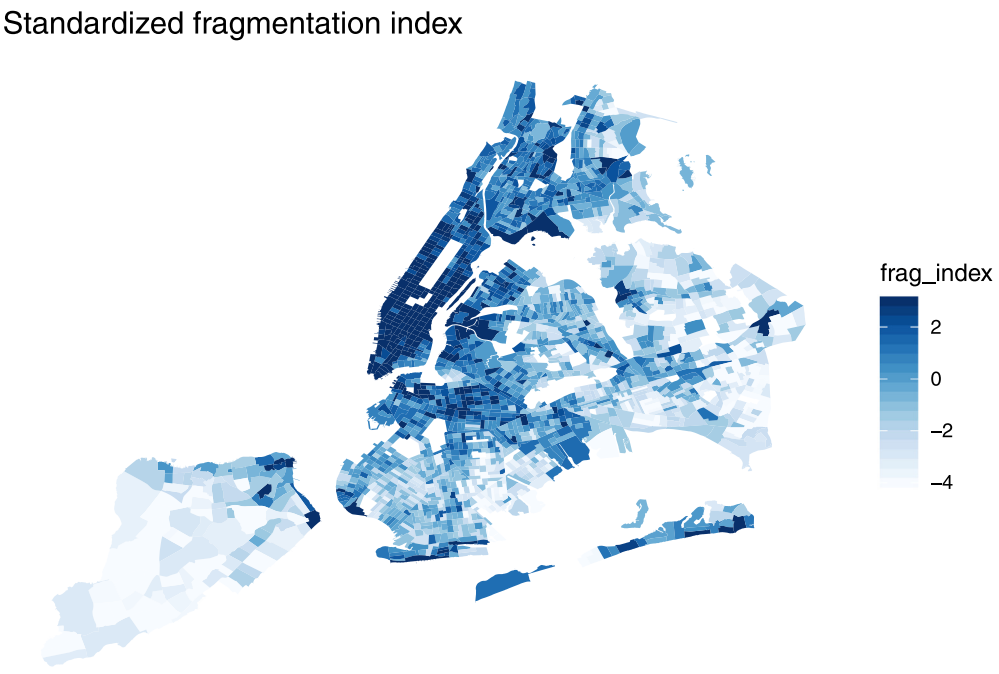


Fig. 13. Areal map of standardized index of social fragmentation per census tract, NYC 2005–2014.

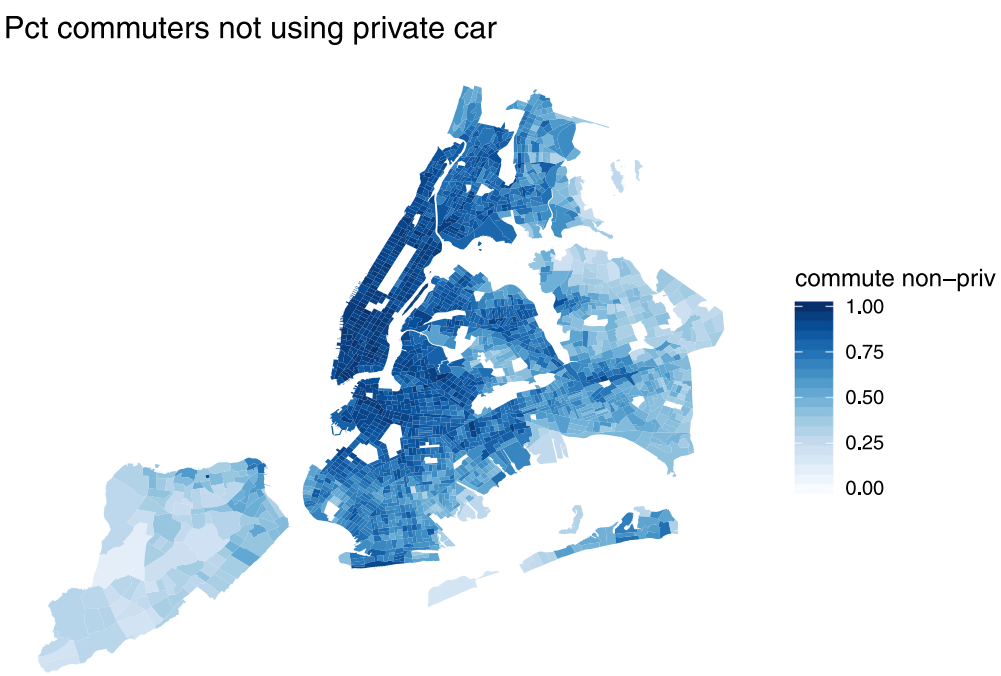


Fig. 14. Areal map of pct commuters using means other than private vehicle per census tract, NYC 2005–2014.

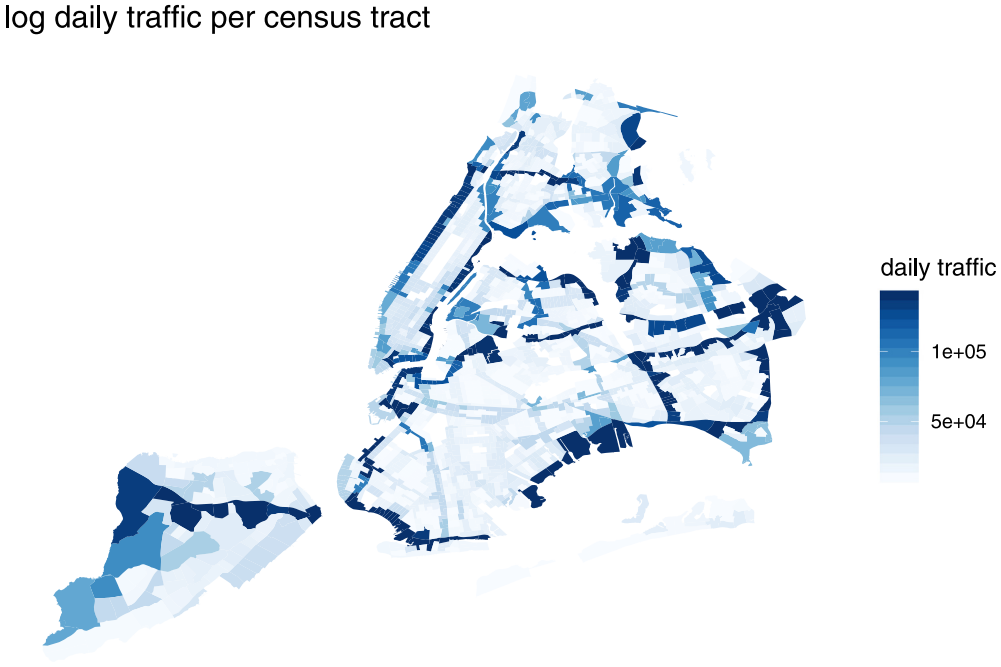


Fig. 15. Areal map of maximum AADT count in each tract, NYC 2015.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.sste.2019.100301.

CRediT authorship contribution statement

Mitzi Morris: Methodology, Software, Validation, Formal analysis, Visualization, Writing - original draft, Writing - review & editing. **Katherine Wheeler-Martin:** Data curation, Formal analysis, Visualization, Writing - original draft, Writing - review & editing. **Dan Simpson:** Methodology, Software, Writing - review & editing. **Stephen J. Mooney:** Methodology, Writing - review & editing. **Andrew Gelman:** Conceptualization, Methodology, Writing - review & editing. **Charles DiMaggio:** Conceptualization, Funding acquisition, Methodology, Supervision.

References

American Factfinder. US census bureau. https://factfinder.census.gov/faces/nav/jsf/pages/index.xhtml.

Annual Average Daily Traffic (Aadt): Beginning 1977. n.d. New York State Department of Transportation, highway data services bureau. http://data.ny.gov/Transportation/Annual-Average-Daily-Traffic-AADT-by-Roadway-Segme/8e88-2p29.

Besag, J.J.Y., Mollié, A., 1991. Bayesian image restoration with two applications in spatial statistics. Ann. Inst. Stat. Math. 43, 1–59. 10.1007.

Besag, J., 1974. Spatial interaction and the statistical analysis of lattice systems. In: Journal of the Royal Statistical Society. Series B (Methodological). JSTOR, pp. 192–236.

Betancourt M. Diagnosing biased inference with divergences. 2017. https://betanalpha.github.io/assets/case_studies/identifying_mixture_models.html.

Clayton, D., Kaldor, J., 1987. Eempirical bayes estimates of age-standardized relative risks for use in disease mapping. In: Biometrics. JSTOR, pp. 671–681.

Clayton, L.B.D., Montomoli, C., 1995. Bayesian estimates of disease maps: how important are priors? Stat. Med. 14 (21–22), 2411–2431. Wiley Online Library.

Congdon, P., 2012. Assessing the impact of socioeconomic variables on small area variations in suicide outcomes in england. Int. J. Environ. Res. Publ. Health 10 (1), 158–177. Multidisciplinary Digital Publishing Institute.

DiMaggio, C., 2015. Small-area spatiotemporal analysis of pedestrian and bicyclist injuries in new york city. Epidemiology 26 (2), 247–254. LWW.

Freni-Sterrantino, A., Ventrucci, M., Rue, H., 2018. A note on intrinsic conditional autoregressive models for disconnected graphs. In: Spatial and Spatio-Temporal Epidemiology 26. Elsevier, pp. 25–34.

Fung, L., Conderino, S., 2017. Pedestrian fatalities in new york city. In: New York City Department of Health; Mental Hygiene; Epi Data Brief (86), p. 2017. http://www1.nyc.gov/assets/doh/downloads/pdf/epi/databrief86.pdf.

Gabry J., Simpson D., Vehtari A., Betancourt M., Gelman A. Visualization in bayesian workflow, 2017. arXiv:1709.01449.

Hoffman, M.D., Gelman, A., 2014. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. J. Mach. Learn. Res. 15 (1), 1593–1623.

Leroux, B.G., Lei, X., Breslow, N., 2000. Estimation of disease rates in small areas: a new mixed model for spatial dependence. In: Statistical Models in Epidemiology, the Environment, and Clinical Trials. Springer, pp. 179–191.

Mooney, S.J., DiMaggio, C.J., Lovasi, G.S., Neckerman, K.M., Bader, M.D.M., Teitler, J.O., Sheehan, D.M., Jack, D.W., Rundle, A.G., 2016. Use of Google street view to assess environmental contributions to pedestrian injury. In: American Journal of Public Health 106 (3). American Public Health Association, pp. 462–469.

Riebler, A., Sørbye, S.H., Simpson, D., Rue, H., 2016. An intuitive Bayesian spatial model for disease mapping that accounts for scaling. In: Statistical Methods in Medical Research 25 (4). SAGE Publications Sage, UK: London, England, pp. 1145–1165.

Safe Routes. n.d. National center for safe routes to school. http://www.saferoutesinfo.org/.

Simpson, D., Rue, H., Riebler, A., Martins, T.G., Sørbye, S.H., et al., 2017. Penalising model component complexity: aprincipled, practical approach to constructing priors. In: Statistical Science 32 (1). Institute of Mathematical Statistics, pp. 1–28.

Stan Development Team. Stan user's guide 2019. https://mc-stan.org/docs/stan-users-guide/index.html.

Viola R., Roe M., Shin H.S. New York city pedestrian safety study & action plan New York (NY). dept. of transportation. 2010.

Vision Zero: Traffic safety by Sweden. n.d. http://www.visionzeroinitiative.com/.