Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

Spatial Generalized Linear Models in R Using spmodel

Michael Dumelle
United States
Environmental Protection Agency

Jay M. Ver Hoef
National Oceanic and
Atmospheric Administration

Matt Higham
St. Lawrence University

Abstract

Generalized linear models (GLMs) describe a non-normal response variable that may be binary, count, skewed, or a proportion. Typically, observations in a GLM are assumed independent of one another. For spatial data, this independence assumption is impractical, as nearby locations tend to be more similar than locations far apart. The **spmodel** R package provides tools to fit GLMs that incorporate spatial autocorrelation (i.e., spatial generalized linear models, or SPGLMs). SPGLMs are fit in **spmodel** using a novel application of the Laplace approximation via `spglm()` for point-referenced data or `spgautor()` for areal (i.e., lattice), data. `spglm()` and `spgautor()` closely resemble `glm()` from base R but include arguments that control the spatial autocorrelation structure. **spmodel** has many helper functions for model inspection and diagnostics, some of which leverage other R packages like **broom** and **emmeans**. **spmodel** has tools to make predictions of the latent spatial-mean process at unobserved locations. **spmodel** also provides many advanced features like accommodating geometric anisotropy and nonspatial random effects, simulating spatially autocorrelated data, and more. Here we use **spmodel** to illustrate the modeling of binary, count, skewed and proportion response variables from several point-referenced and areal data sets.

Keywords: Autoregressive Model, Geostatistics, Spatial Covariance, Spatially-Explicit Model, Statistical Model.

1. Introduction

Binary, count, proportion, and skewed data are ubiquitous in practice. These data types are naturally modeled using a generalized linear model (GLM) framework (Nelder and Wedderburn 1972; McCullagh and Nelder 1989; Myers, Montgomery, Vining, and Robinson 2012; Faraway 2016). In a GLM, a response variable y belongs to a particular statistical distribution with mean μ. For example, y may be distributed as a Poisson random variable with some mean μ. Based on the response distribution assumed for y, GLMs link a function of μ to explanatory variables via a link function:

$$f(\boldsymbol{\mu}) \equiv \mathbf{w} = \mathbf{X}\boldsymbol{\beta}, \tag{1}$$

where for a sample size n, $\boldsymbol{\mu}$ is the mean of an $n \times 1$ response vector $\mathbf{y}$, $f(\boldsymbol{\mu})$ is the link function that connects $\boldsymbol{\mu}$ to $\mathbf{w}$, $\mathbf{X}$ is the $n \times p$ design matrix of explanatory variables, and $\boldsymbol{\beta}$ is the $p \times 1$ vector of fixed effects. The mean, $\boldsymbol{\mu}$, is usually constrained in some way (e.g., positive) that depends on the distribution assumed for $\mathbf{y}$, but $\mathbf{w}$ is unconstrained. The parameters in Equation (1) are typically estimated via maximum likelihood (e.g., iteratively reweighted least squares) (Chambers and Hastie 1992). The `glm()` function is commonly used to fit GLMs in the R programming language (R Core Team 2024).

The GLM framework in Equation (1) assumes the elements of $\mathbf{y}$ are independent of one another. This assumption is impractical for spatial data, where nearby observations tend to be more similar than distant observations (Tobler 1970). Ignoring this spatial dependence can give rise to misleading inference and poor prediction (Zimmerman and Ver Hoef 2024). Spatial GLMs (SPGLMs) formally incorporate spatial autocorrelation into a GLM by adding to Equation (1) two random effects that elucidate spatial structure:

$$f(\boldsymbol{\mu}) \equiv \mathbf{w} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\tau} + \boldsymbol{\epsilon}, \tag{2}$$

where $\boldsymbol{\tau}$ is an $n \times 1$ column vector of spatially dependent random errors, and $\boldsymbol{\epsilon}$ is an $n \times 1$ column vector of spatially independent random errors. We make a few assumptions about $\boldsymbol{\tau}$ and $\boldsymbol{\epsilon}$: first, that $\mathrm{E}(\boldsymbol{\tau}) = \mathrm{E}(\boldsymbol{\epsilon}) = \mathbf{0}$, where $\mathrm{E}(\cdot)$ denotes expectation; second, that $\mathrm{Cov}(\boldsymbol{\tau}) = \sigma^2_{\tau}\mathbf{R}$, where $\mathbf{R}$ is an $n \times n$ matrix that determines the spatial dependence structure in $\mathbf{w}$ (which is linked to $\boldsymbol{\mu}$ and hence $\mathbf{y}$) and depends on a range parameter, ϕ; third, that $\mathrm{Cov}(\boldsymbol{\epsilon}) = \sigma^2_{\epsilon}\mathbf{I}$, where $\mathbf{I}$ is an $n \times n$ identity matrix; and fourth, that $\boldsymbol{\tau}$ and $\boldsymbol{\epsilon}$ are independent of one another. The parameter σ^2_{τ} is called the spatially dependent random error variance or partial sill. The parameter σ^2_{ϵ} is called the spatially independent random error variance or nugget. These two variance parameters are henceforth more intuitively written as σ^2_{de} and σ^2_{ie}, respectively. The covariance of $\mathbf{w}$ is denoted $\boldsymbol{\Sigma}$ and is given by

$$\boldsymbol{\Sigma} = \sigma^2_{de}\mathbf{R} + \sigma^2_{ie}\mathbf{I}. \tag{3}$$

The parameters σ^2_{de}, ϕ, and σ^2_{ie} are elements of $\boldsymbol{\theta}$, the covariance parameter vector.

The **spmodel** R package provides tools for fitting spatial statistical models and making predictions at unobserved locations (Dumelle, Higham, and Ver Hoef 2023). A recent major update (v0.4.0) to **spmodel** added SPGLM (Equation 2) support for binary, count, skewed, and proportion response variables (Table 1), greatly expanding the class of models that **spmodel** makes accessible to practitioners.

SPGLMs for point-referenced data are called geostatistical GLMs and are fit using `spglm()`. Data are point-referenced when the elements in $\mathbf{y}$ are observed at point-locations indexed

Family	Link Function	Link Name	Data Type
Binomial	$f(\mu) = \log(\mu/(1-\mu))$	Logit	Binary; Binary Count
Poisson	$f(\mu) = \log(\mu)$	Log	Count
Negative Binomial	$f(\mu) = \log(\mu)$	Log	Count
Beta	$f(\mu) = \log(\mu/(1-\mu))$	Logit	Proportion
Gamma	$f(\mu) = \log(\mu)$	Log	Skewed
Inverse Gaussian	$f(\mu) = \log(\mu)$	Log	Skewed

Table 1: SPGLM response distributions and their link functions and data types.

by x-coordinates and y-coordinates on a spatially continuous surface with an infinite number of locations (e.g., point locations in a field). SPGLMs for areal data are called spatial autoregresive models and are fit using `spgautor()`. Data are areal (i.e., lattice) when they are part of a finite network of polygons whose connections are indexed by a neighborhood structure (e.g., states in a country).

Several other R packages exist for analyzing SPGLMs. The **brms** (Bürkner 2017), **carBayes** (Lee 2013), **ngspatial** (Hughes and Cui 2020), **R-INLA** (Lindgren and Rue 2015), **spBayes** (Finley, Banerjee, and Carlin 2007), and **spNNGP** (Finley, Datta, and Banerjee 2022) packages take a Bayesian approach, either directly sampling from posterior distributions of parameters (e.g., using MCMC) or approximating them. A benefit of Bayesian approaches is that prior information can be incorporated and uncertainty quantification of parameter estimates is straightforward. However, Bayesian approaches, especially those using MCMC, tend to be computationally expensive. In order to reduce computation time, many of these packages work with the precision matrix instead of the covariance matrix so that inverting matrices (a computational bottleneck) is not required. For example, **R-INLA** uses the precision matrix (often directly modeled as with autoregressive models) and tends to be very fast. Working with precision matrices, however, can be more restrictive and less intuitive than working directly with the covariance matrix. The **FRK** (Sainsbury-Dale, Zammit-Mangion, and Cressie 2024), **glmmTMB** (Brooks, Kristensen, van Benthem, Magnusson, Berg, Nielsen, Skaug, Maechler, and Bolker 2017), **hglm** (Ronnegard, Shen, and Alam 2010), **mgcv** (Wood 2017), and **spaMM** (Rousset and Ferdy 2014) packages directly use Laplace, quasi-likelihood, or reduced-rank approaches to estimate parameters. These direct approaches tend to be computationally efficient, as they don't rely on MCMC sampling. In contrast to the Bayesian approach, a drawback of these direct approaches is that prior information cannot be formally incorporated and covariance parameter uncertainty is challenging to quantify. SPGLMs in **spmodel** are most similar to those in **glmmTMB** – **spmodel** uses analytical solutions for maximizing the likelihood (which we describe in the next section), while **glmmTMB** uses automatic differentiation for maximizing the likelihood (which tends to be much slower than analytical solutions).

Missing from the aforementioned R packages is the complete set of tools for SPGLMs that **spmodel** provides. Importantly, the `spglm()` and `spautor()` functions act as a spatial analogue to the familiar `glm()` from base R, making the transition from GLMs to SPGLMs relatively seamless. **spmodel** leverages many commonly used R generics like `summary()` to better understand fitted models. Six GLM families (Table 1) and 20 spatial covariance (or precision) functions are supported. Also available are functions for data visualization, model fitting, model summaries, model diagnostics, model comparison, and prediction, all crucial components of a data analysis. Through extra function arguments, **spmodel** supports advanced features like geometric anisotropy, non-spatial random effects, methods for large data sets, and more. Importantly, **spmodel** extends popular packages like **broom** (Robinson, Hayes, and Couch 2021; Kuhn and Silge 2022) and **emmeans** (Lenth 2024). **spmodel** also provides function for simulating SPGLM data (e.g., `sprbinom()`, `sprpois()`).

The rest of this article is organized as follows. In Section 2, we provide some background for the SPGLM fitting and prediction routines in **spmodel**. In Section 3, we provide several applications of **spmodel** to spatial binary, count, skewed and proportion data with both point-referenced and areal supports. And in Section 4, we end with a discussion synthesizing **spmodel**'s contributions to the analysis of SPGLMs in R.

2. Marginal SPGLMs and the Laplace approximation

GLMs with random effects are often written hierarchically (Lee and Nelder 1996; Bolker, Brooks, Clark, Geange, Poulsen, Stevens, and White 2009; Wood 2017). **spmodel** leverages this hierarchical structure and uses a novel application of the Laplace approximation (Ver Hoef, Blagg, Dumelle, Dixon, Zimmerman, and Conn 2024) to fit models. This marginal approach is quite flexible, accommodating a wide range of possible dependence structures and formally maximizing a likelihood. Maximizing a likelihood yields convenient likelihood-based statistics like AIC (Akaike 1974), AICc (Hoeting, Davis, Merton, and Thompson 2006), BIC (Schwarz 1978), deviance (McCullagh and Nelder 1989), and likelihood ratio tests for model comparison, a benefit compared to quasi-likelihood (Wedderburn 1974; Breslow and Clayton 1993) or pseudo-likelihood approaches (Wolfinger and O'connell 1993), which only specify the first two moments of a distribution. Ver Hoef *et al.* (2024) provide thorough context for the marginal approach and its associated details, but next we provide an short overview of the methodology.

Our goal is to marginalize over the latent mean $\mathbf{w}$ and fixed effects $\boldsymbol{\beta}$ in Equation (2) to obtain a response distribution for $\mathbf{y}$ that depends only on the explanatory variables, $\mathbf{X}$, the response distribution's dispersion parameter, φ, and the covariance parameters, $\boldsymbol{\theta}$. We can represent this marginal distribution hierarchically as

$$[\mathbf{y}|\mathbf{X},\varphi,\boldsymbol{\theta}] = \int_{\mathbf{w}}\int_{\boldsymbol{\beta}}[\mathbf{y}|f^{-1}(\mathbf{w}),\varphi][\mathbf{w}|\mathbf{X},\boldsymbol{\theta}]d\boldsymbol{\beta}d\mathbf{w}. \tag{4}$$

In Equation (4), $[\mathbf{y}|f^{-1}(\mathbf{w}),\varphi]$ is the density for $\mathbf{y}$ (e.g., binomial, Poisson) given the latent mean and dispersion parameter and $[\mathbf{w}|\mathbf{X},\boldsymbol{\theta}]$ is the Gaussian density for $\mathbf{w}$ given the explanatory variables, fixed effects, and covariance parameters. Integrating $\boldsymbol{\beta}$ out of Equation (2) yields

$$[\mathbf{y}|\mathbf{X},\varphi,\boldsymbol{\theta}] = \int_{\mathbf{w}}[\mathbf{y}|f^{-1}(\mathbf{w}),\varphi][\mathbf{w}|\mathbf{X},\boldsymbol{\theta}]d\mathbf{w}, \tag{5}$$

where $[\mathbf{w}|\mathbf{X},\boldsymbol{\theta}]$ is the restricted (i.e., residual) Gaussian density (Patterson and Thompson 1971, Harville (1977), Wolfinger, Tobias, and Sall (1994)) for $\mathbf{w}$ given the explanatory variables and covariance parameters. This restricted Gaussian density is given by

$$[\mathbf{w}|\mathbf{X},\boldsymbol{\theta}] = \frac{\exp(-\frac{1}{2}(\mathbf{y}-\mathbf{X}\tilde{\boldsymbol{\beta}})\boldsymbol{\Sigma}^{-1}(\mathbf{y}-\mathbf{X}\tilde{\boldsymbol{\beta}})^T)}{(2\pi)^{(n-p)/2}|\boldsymbol{\Sigma}|^{1/2}|\mathbf{X}^T\boldsymbol{\Sigma}^{-1}\mathbf{X}|^{1/2}}, \tag{6}$$

where $\tilde{\boldsymbol{\beta}} = (\mathbf{X}^T\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{\Sigma}^{-1}\mathbf{w}$ and $|\cdot|$ denotes the determinant.

Next, let $\ell_{\mathbf{w}} = \log([\mathbf{y}|f^{-1}(\mathbf{w}), \varphi][\mathbf{w}|\mathbf{X}, \boldsymbol{\theta}])$ and then notice that Equation (5) can be written as

$$[\mathbf{y}|\mathbf{X}, \varphi, \boldsymbol{\theta}] = \int_{\mathbf{w}} \exp(\ell_{\mathbf{w}}) d\mathbf{w}. \tag{7}$$

A Taylor series expansion of $\ell_{\mathbf{w}}$ around a point $\mathbf{a}$ yields

$$[\mathbf{y}|\mathbf{X}, \varphi, \boldsymbol{\theta}] \approx \int_{\mathbf{w}} \exp(\ell_{\mathbf{a}} + \mathbf{g}^T(\mathbf{w} - \mathbf{a}) + \frac{1}{2}(\mathbf{w} - \mathbf{a})^T \mathbf{G}(\mathbf{w} - \mathbf{a})) d\mathbf{w}, \tag{8}$$

where $\mathbf{g}$ and $\mathbf{G}$ are the gradient and Hessian, respectively, of $\ell_{\mathbf{w}}$ with respect to $\mathbf{w}$. If $\mathbf{a}$ is a value for which $\mathbf{g} = \mathbf{0}$,

$$[\mathbf{y}|\mathbf{X}, \varphi, \boldsymbol{\theta}] \approx \exp(\ell_{\mathbf{a}}) \int_{\mathbf{w}} \exp(-\frac{1}{2}(\mathbf{w} - \mathbf{a})^T(-\mathbf{G})(\mathbf{w} - \mathbf{a})) d\mathbf{w}. \tag{9}$$

Notice that the integral in Equation 9 can be solved in the same manner as the normalizing constant in a multivariate Gaussian distribution, and rewriting $\exp(\ell_{\mathbf{a}})$ yields

$$[\mathbf{y}|\mathbf{X}, \varphi, \boldsymbol{\theta}] \approx [\mathbf{y}|f^{-1}(\mathbf{a}), \varphi][\mathbf{a}|\mathbf{X}, \boldsymbol{\theta}](2\pi)^{n/2}|-\mathbf{G}_{\mathbf{a}}|^{-1/2}. \tag{10}$$

Ver Hoef *et al.* (2024) show how to evaluate $\mathbf{g}$ and $\mathbf{G}$ to obtain $\mathbf{a}$ and maximize (the natural logarithm of) Equation (10) for the six response distributions in Table 1. Maximizing Equation (10) using a Newton-Rhapson approach yields the marginal restricted maximum likelihood estimators $\hat{\varphi}$ and $\hat{\boldsymbol{\theta}}$. Ver Hoef *et al.* (2024) show that the fixed effect estimator is $\hat{\boldsymbol{\beta}} = (\mathbf{X}^T \hat{\boldsymbol{\Sigma}}^{-1} \mathbf{X})^{-1} \mathbf{X}^T \hat{\boldsymbol{\Sigma}}^{-1} \mathbf{w}$, where $\hat{\boldsymbol{\Sigma}}$ is the covariance matrix $\boldsymbol{\Sigma}$ evaluated at $\hat{\boldsymbol{\theta}}$. Ver Hoef *et al.* (2024) also derive the covariance matrix of $\hat{\boldsymbol{\beta}}$ and show how to predict $\mathbf{w}$ at unobserved locations and quantify their uncertainties.

3. Application

The `spglm()` (for point-referenced data) and `spgautor()` (for areal data) functions in **spmodel** fit SPGLMs using the Laplace approximation outlined in Section 2. Both `spglm()` and `spgautor()` generally require the following four arguments: `formula`, a formula that describes the relationship between the response variable and explanatory variables; `family`, the response distribution (which can be `binomial`, `poisson`, `nbinomial`, `Gamma`, `inverse.gaussian`, or `beta`); `data`, the data frame that holds the variables in `formula` as well as spatial locations; and `spcov_type`, the spatial covariance type. The first three arguments are shared by `glm()`; thus, the transition from GLMs to SPGLMs requires only one additional argument: `spcov_type`. The `spglm()` spatial covariance types measure dependence as a function of Euclidean distance among observations; an example is the exponential spatial covariance:

$$\boldsymbol{\Sigma} = \sigma^2_{de} \exp(-\mathbf{H}/\phi) + \sigma^2_{ie}\mathbf{I}, \tag{11}$$

where $\mathbf{H}$ is a matrix of pairwise distances among all observations. `spglm()` currently supports 18 distinct spatial covariance functions.

The `spgautor()` spatial covariance types measure dependence as a function of neighborhood distance among observations; an example is the simultaneous autoregressive covariance matrix:

$$\boldsymbol{\Sigma} = \sigma^2_{de}[(\mathbf{I} - \phi\mathbf{W})(\mathbf{I} - \phi\mathbf{W})^T]^{-1} + \sigma^2_{ie}\mathbf{I}, \tag{12}$$

where $\mathbf{W}$ is a matrix that represents the neighborhood structure among all observations. `spgautor()` currently supports two distinct spatial covariance functions.

In the rest of this section, we use **spmodel** to study binary, count, skewed, and proportion response variables that are either point-referenced or areal. We use **spmodel** for all parts of the data analysis, from estimation to inference to model diagnostics to prediction. We first describe core **spmodel** functionality in an application to binary data, while additional analyses highlight count, skewed, and proportion data as well as some additional **spmodel** features. Before proceeding, load **spmodel** into the current R session:

```
R> library("spmodel")
```

3.1. Binary data

The `moose` data in **spmodel** contain information on moose presence in Alaska. They are an `sf` object, a special data frame that is supplemented with spatial information using the **sf** package (Pebesma 2018). The first few rows look like:

```
R> head(moose)
```

```
Simple feature collection with 6 features and 4 fields
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: 281896.4 ymin: 1518398 xmax: 311325.3 ymax: 1541016
Projected CRS: NAD83 / Alaska Albers
# A tibble: 6 x 5
   elev strat count presence          geometry
  <dbl> <chr> <dbl> <fct>          <POINT [m]>
1  469. L         0 0        (293542.6 1541016)
2  362. L         0 0        (298313.1 1533972)
3  173. M         0 0        (281896.4 1532516)
4  280. L         0 0        (298651.3 1530264)
5  620. L         0 0        (311325.3 1527705)
6  164. M         0 0        (291421.5 1518398)
```

There are five columns: `elev`, the numeric site elevation (meters); `strat` a stratification variable for sampling with two levels, `"L"` and `"M"`, which are categorized by landscape metrics at each site; `count`, the number of moose at each site; `presence`, a factor that indicates whether at least one moose was observed at each site (`0` implies no moose; `1` implies at least one moose); and `geometry`, the NAD83 projected coordinate of each site. The `moose_preds` data in **spmodel** contain spatial locations at which predictions of moose presence are desired. They are also an `sf` object with the same projection and measurements for `elev` and `strat`. Figure 1 shows the `presence` variable in `moose` as well as the spatial locations of both `moose` and `moose_preds`. Moose are most common in the southwestern and eastern parts of the domain and least common in the northwest.

To study the effect of elevation, stratum, and their interaction on moose presence while accounting for spatial autocorrelation, we fit a SPGLM for binary data (i.e., a spatial logistic regression model) using `spglm()`:

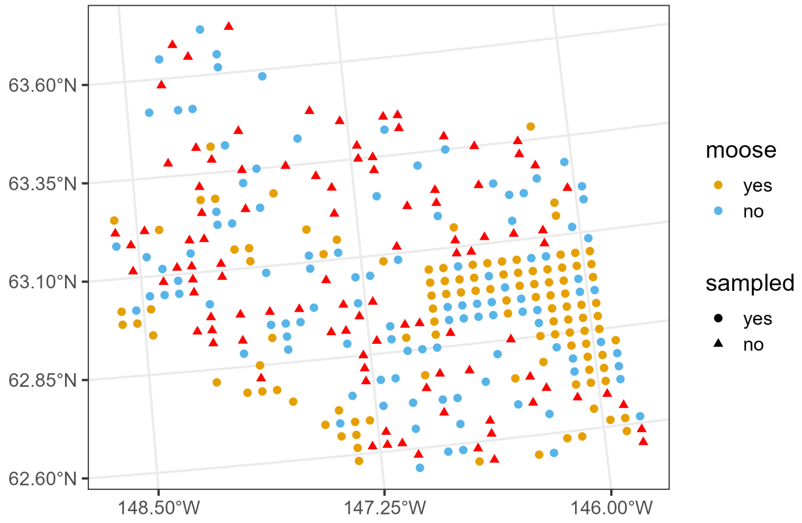

Figure 1: Moose presence in Alaska. Circles represent moose presence or absence (based on color) and triangles represent locations at which moose presence probability predictions are desired.

```
R> spbin <- spglm(
+   formula = presence ~ elev + strat + elev:strat,
+   family = binomial,
+   data = moose,
+   spcov_type = "spherical"
+)
```

Summarizing the model object yields a summary similar to that provided by the familiar `glm()`:

```
R> summary(spbin)

Call:
spglm(formula = presence ~ elev + strat + elev:strat, family = binomial,
    data = moose, spcov_type = "spherical")

Deviance Residuals:
    Min      1Q  Median      3Q     Max
-1.8423 -0.7538  0.3883  0.7604  1.6018

Coefficients (fixed):
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.039992   1.205695  -2.521  0.01169 *
elev         0.009133   0.004126   2.213  0.02687 *
stratM       3.276511   1.162603   2.818  0.00483 **
elev:stratM -0.010882   0.006697  -1.625  0.10418
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Pseudo R-squared: 0.08845

Coefficients (spherical spatial covariance):
       de        ie     range
5.083e+00 2.580e-03 5.158e+04

Coefficients (Dispersion for binomial family):
dispersion
         1
```

The summary contains the original function call, a summary of residuals, the fixed effects coefficients table, the spatial covariance parameter estimates, and additional model information like the pseudo R-squared, which quantifies the variability in the model attributable to the fixed effects. While useful, this summary information is hard to work with, as it is printed directly to the R console. The **broom** package from the tidymodels (Kuhn and Silge 2022) ecosystem has functions to provide helpful model output in the form of tibbles (i.e., data frames) that are easily manipulated. **spmodel** has methods for the `tidy()`, `glance()`, and `augment()` functions from **broom**. The first **broom** function is `tidy()`, which tidies the model output:

```
R> tidy(spbin)
```

```
# A tibble: 4 x 5
  term        estimate std.error statistic p.value
  <chr>          <dbl>     <dbl>     <dbl>   <dbl>
1 (Intercept) -3.04      1.21        -2.52 0.0117
2 elev         0.00913   0.00413      2.21 0.0269
3 stratM       3.28      1.16         2.82 0.00483
4 elev:stratM -0.0109    0.00670     -1.62 0.104
```

The estimates and standard errors returned are on the log odds link (Table 1) scale (`coef()` and `vcov()` may also be used). The output provides evidence that elevation is positively associated with moose presence in the `"L"` stratum ($p-$value < 0.05) and, at zero elevation, moose are more likely in the `"M"` stratum than the `"L"` stratum ($p-$value < 0.01). This output provides marginal evidence that the effect of elevation on moose presence varies across strata ($p-$value ≈ 0.1). The model effectively quantifies the impact of elevation on moose presence for moose in the `"L"` strata, but an analogous statement for moose in the `"M"` strata requires more context. We could refit the model treating `"M"` as the reference group instead of `"L"`:

```
R> moose$strat2 <- factor(moose$strat, levels = c("M", "L"))
R> update(spbin, formula = presence ~ elev + strat2 + elev:strat2) |>
+  summary()
```

```
Call:
spglm(formula = presence ~ elev + strat2 + elev:strat2, family = binomial,
    data = moose, spcov_type = "spherical")

Deviance Residuals:
    Min      1Q  Median      3Q     Max
```

```
-1.8423 -0.7538  0.3883  0.7604  1.6018

Coefficients (fixed):
              Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.236519   1.305198   0.181  0.85620
elev         -0.001750   0.006090  -0.287  0.77385
strat2L      -3.276511   1.162603  -2.818  0.00483 **
elev:strat2L  0.010882   0.006697   1.625  0.10418
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Pseudo R-squared: 0.08845

Coefficients (spherical spatial covariance):
       de        ie     range
5.083e+00 2.580e-03 5.158e+04

Coefficients (Dispersion for binomial family):
dispersion
         1
```

A simpler solution, especially if there categorical variables with many levels, is to leverage **emmeans**. **emmeans** is an R package for estimating marginal means of model objects. The `emtrends()` function in **emmeans** characterizes the effect of a continuous variable (here, `elev`) for each level of a categorical variable (here, `strat`):

```
R> library("emmeans")

R> emtrends(spbin, "strat", "elev")

 strat elev.trend      SE  df asymp.LCL asymp.UCL
 L        0.00913 0.00413 Inf   0.00105    0.0172
 M       -0.00175 0.00609 Inf  -0.01369    0.0102

Degrees-of-freedom method: asymptotic
Confidence level used: 0.95
```

The asymptotic confidence intervals show that there is more evidence of an association between elevation and moose presence in the "L" stratum than in the "M" stratum. Notice that `elev.trend` for the "L" stratum matches the `elev` effect when "L" is the reference group, and similarly for `elev.trend` when "M" is the reference group.

The second **broom** function is `glance()`, which glances at the model fit:

```
R> glance(spbin)

# A tibble: 1 x 10
      n     p  npar value   AIC  AICc   BIC logLik deviance pseudo.r.squared
  <int> <dbl> <int> <dbl> <dbl> <dbl> <dbl>  <dbl>    <dbl>            <dbl>
1   218     4     3  681.  687.  687.  697.  -340.     161.           0.0885
```

`glance()` returns several useful statistics like the sample size (`n`), number of fixed effects (`p`), number of covariance parameters (`npar`), several likelihood-based statistics (e.g., `AIC`, `AICc`, `BIC`), and pseudo R-squared.

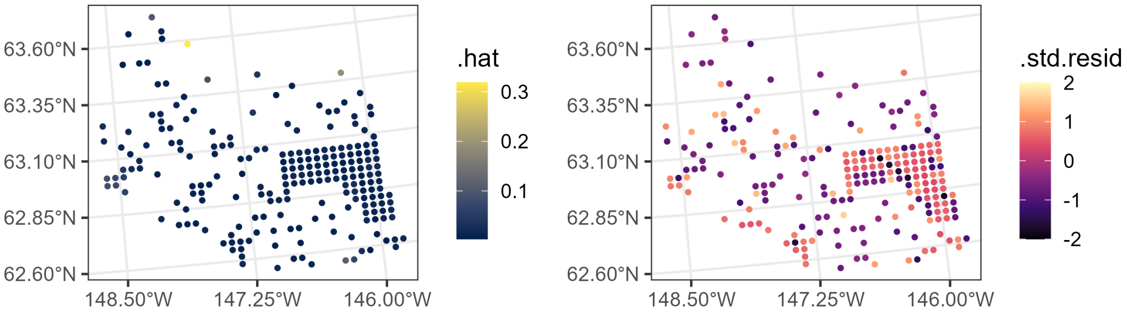

Figure 2: Spatial logistic regression model diagnostics from `augment()`. The leverage (i.e., hat) values (left) and standardized residuals (right).

The third **broom** function is `augment()`, which augments the model data with diagnostics:

```
R> head(augment(spbin))
```

```
Simple feature collection with 6 features and 8 fields
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: 281896.4 ymin: 1518398 xmax: 311325.3 ymax: 1541016
Projected CRS: NAD83 / Alaska Albers
# A tibble: 6 x 9
  presence  elev strat .fitted .resid    .hat   .cooksd .std.resid
  <fct>    <dbl> <chr>   <dbl>  <dbl>   <dbl>     <dbl>      <dbl>
1 0         469. L      -1.47  -0.644 0.101   0.0130        -0.679
2 0         362. L      -2.77  -0.349 0.0166  0.000523      -0.352
3 0         173. M      -2.23  -0.451 0.00390 0.000200      -0.452
4 0         280. L      -3.59  -0.234 0.00343 0.0000472     -0.234
5 0         620. L      -0.774 -0.871 0.319   0.130         -1.06
6 0         164. M      -2.01  -0.502 0.00459 0.000292      -0.503
# i 1 more variable: geometry <POINT [m]>
```

The `augment()` function returns the fitted **w** values (`.fitted`), deviance residuals (`.resid`), leverage (i.e., hat) values (`.hat`), Cook's distance (`.cooksd`), and standardized residuals (`.std.resid`). When the data are an `sf` object, `augment()` returns another `sf` object, helpful for visualizing model diagnostics spatially as in Figure 2. Leverage measures the unusualness of an observation's set of explanatory variables, while Cook's distances measures how influential an observation is on the resulting model fit (Montgomery, Peck, and Vining 2021). Model diagnostics are also accessible as vectors using the appropriate generic function (e.g., `fitted()`, `residuals()`).

Similar to `glm()` model objects, `plot()` can be used to visualize diagnostics (Figure 3):

```
R> plot(spbin, which = c(4, 7))
```

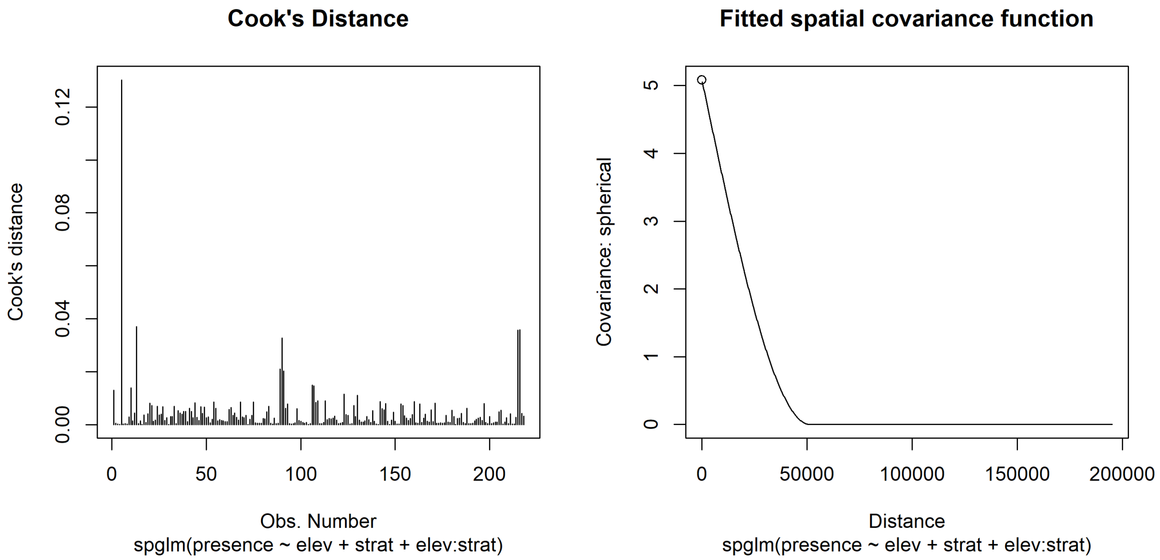

Figure 3: Spatial logistic regression model diagnostics from `plot()`. The Cook's distance values (left) and the fitted spatial covariance as a function of distance (right).

Components of model variation are partitioned using `varcomp()`:

```
R> varcomp(spbin)
```

```
# A tibble: 3 x 2
  varcomp             proportion
  <chr>                    <dbl>
1 Covariates (PR-sq)    0.0885
2 de                    0.911
3 ie                    0.000462
```

The fixed effects explain roughly 9% of model variation, while the spatially dependent variance explains most of the remaining variability.

We make predictions of the log odds of moose probability presence at each site in `moose_preds` using `predict()`:

```
R> head(predict(spbin, newdata = moose_preds))
```

```
          1           2           3           4           5           6
 0.08588581 -0.40762380 -1.87510889 -1.14172781  1.45701519 -2.74275553
```

`augment()` may also be used to augment the prediction data with predictions:

```
head(augment(
  spbin,
  newdata = moose_preds,
  type.predict = "response",
  interval = "prediction"
```

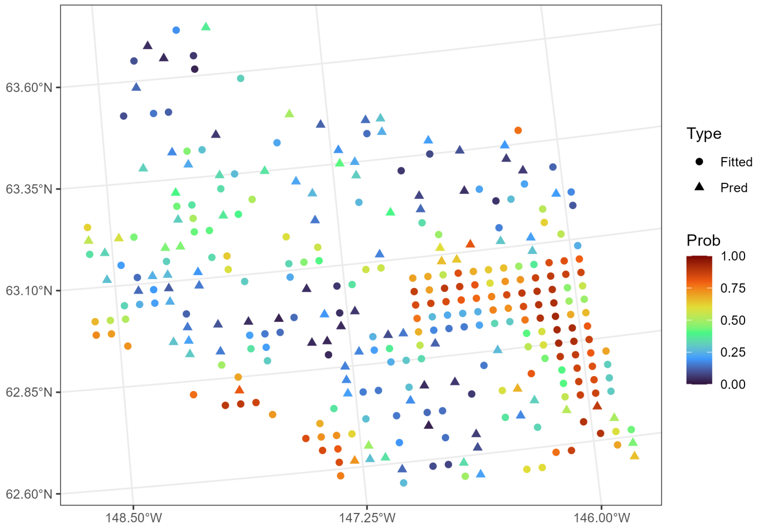

Figure 4: Moose presence probability fitted values and predictions. Fitted values are represeneted by circles and predictions by triangles.

```
))

Simple feature collection with 6 features and 5 fields
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: 291839.8 ymin: 1436192 xmax: 401239.6 ymax: 1512103
Projected CRS: NAD83 / Alaska Albers
# A tibble: 6 x 6
   elev strat .fitted  .lower .upper           geometry
  <dbl> <chr>   <dbl>   <dbl>  <dbl>        <POINT [m]>
1  143. L      0.521  0.0983   0.916 (401239.6 1436192)
2  324. L      0.399  0.0331   0.928 (352640.6 1490695)
3  158. L      0.133  0.00957  0.709 (360954.9 1491590)
4  221. M      0.242  0.0261   0.792 (291839.8 1466091)
5  209. M      0.811  0.289    0.978 (310991.9 1441630)
6  218. L      0.0605 0.00360  0.534 (304473.8 1512103)
```

Here, we requested predictions on the probability (i.e., response) scale (Figure 4) alongside lower and upper bounds of a 95% (see `level`) prediction interval (Figure 5).

Thus far we have heuristically argued, based on first principles, that there are benefits to incorporating spatial autocorrelation for GLMs applied to spatial data. Now we provide some empirical evidence to support this claim by comparing the fits of the SPGLM and a GLM. If `spcov_type = "none"`, the resulting model fit is nearly identical to that from `glm()`:

```
R> # fit nonspatial model using spglm()
R> bin <- spglm(
+  formula = presence ~ elev + strat + elev:strat,
+  family = binomial,
```

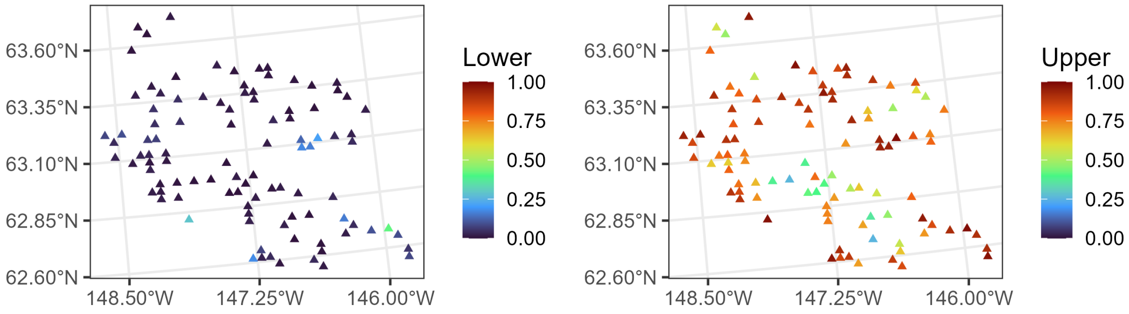

Figure 5: Moose presence probability prediction intervals. 95% prediction interval lower bound (left) and 95% prediction interval upper bound (right).

```
+   data = moose,
+   spcov_type = "none"
+)
R> # fit nonspatial model using glm()
R> bin_glm <- glm(
+   formula = presence ~ elev + strat + elev:strat,
+   family = binomial,
+   data = moose
+)
R> # compare fixed effect coefficients and standard errors up to four decimals
R> data.frame(
+   est_none = coef(bin),
+   est_glm = coef(bin_glm),
+   se_none = sqrt(diag(vcov(bin))),
+   se_glm = sqrt(diag(vcov(bin_glm)))
+) |>
+   apply(2, round, digits = 4)

            est_none est_glm se_none se_glm
(Intercept)  -0.5219 -0.5219  0.5082 0.5082
elev          0.0002  0.0002  0.0024 0.0024
stratM        1.0274  1.0274  0.7150 0.7150
elev:stratM  -0.0013 -0.0013  0.0038 0.0038
```

The advantage of using `spglm()` to fit a model with `spcov_type = "none"` is that it provides access to other **spmodel** functions for model objects (e.g., `glances()` below) and accounts for the additional terms in the likelihood from Equation (10). These additional terms make the likelihood for `spglm()` and `glm()` different, though the models convey the same information. A glance at the spatial and nonspatial models reveals:

```
R> glances(spbin, bin)

# A tibble: 2 x 11
```

```
  model     n     p  npar value   AIC  AICc   BIC logLik deviance
  <chr> <int> <dbl> <int> <dbl> <dbl> <dbl> <dbl>  <dbl>    <dbl>
1 spbin   218     4     3  681.  687.  687.  697.  -340.     161.
2 bin     218     4     0  717.  717.  717.  717.  -359.     294.
# i 1 more variable: pseudo.r.squared <dbl>
```

The spatial model has a notably lower AIC, AICc, BIC, and deviance, suggesting it is the superior model. Another model comparison approach is leave-one-out cross validation. In leave-one-out cross validation, separately each observation is held out, a model is fit to the remaining data, and a prediction is made for the mean of the held out observation on the response scale. Then, statistics like leave-one-out bias, mean-squared-prediction error (MSPE), and the square root of MSPE (RMSPE) may be computed:

```
R> loocv(spbin) |>
+   apply(2, round, digits = 4)
```

```
   bias    MSPE   RMSPE
-0.0006  0.1458  0.3818
```

```
R> loocv(bin) |>
+   apply(2, round, digits = 4)
```

```
  bias   MSPE  RMSPE
0.0000 0.2403 0.4902
```

Both models are nearly unbiased, but the spatial model has an approximately 39% lower MSPE, suggesting the probability predictions tend to be much closer to the observed presence values.

A third model comparison tool is area under the receiver operating characteristic (AUROC) curve. The AUROC curve ranges from zero to one and conveys a model's classification performance over all possible probability thresholds (James, Witten, Hastie, and Tibshirani 2013). Larger values of AUROC indicate a more accurate model:

```
R> AUROC(spbin)
```

```
[1] 0.9490741
```

```
R> AUROC(bin)
```

```
[1] 0.647138
```

All three performance metrics (likelihood-based statistics, leave-one-out statistics, and AUROC) prefer the spatial model.

3.2. Count data

The `count` variable in `moose` contains the number of moose observed at a site (Figure 6). Count data are often modeled using Poisson or negative binomial regression with the log link function. The Poisson regression model assumes each datum's underlying latent mean equals its variance, while the negative binomial accommodates overdispersion (where the variance is greater than the mean) at the cost of estimating an extra parameter.

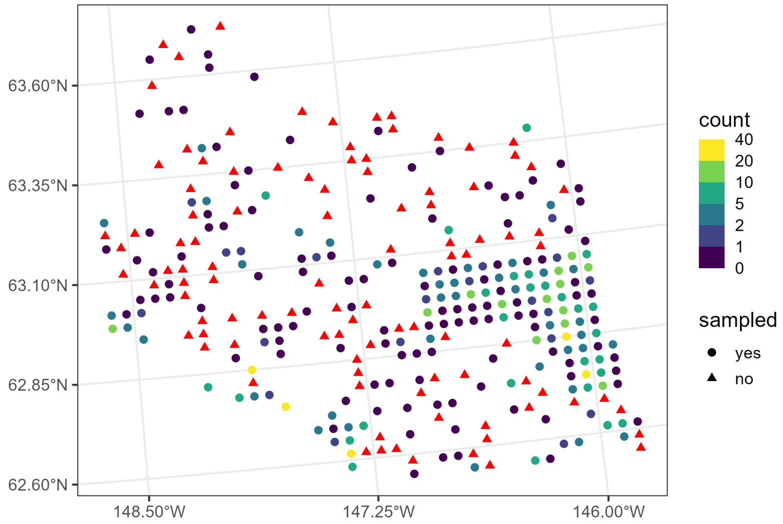

Figure 6: Moose counts in Alaska. Circles represent moose counts (based on color) and triangles represent locations at which mean count predictions are desired.

So far our spatial models have made an implicit assumption of geometric isotropy. A spatial covariance is geometrically isotropic if its dependence decays with distance equally in all directions. A spatial covariance is geometrically anisotropic if its dependence decays differently in different directions. The geometric anisotropy's directionality and strength are controlled by rotation and scale parameters that are applied to the original coordinates, creating a transformed set of coordinates whose spatial covariance is geometrically isotropic. Geometrically anisotropic models are fit by specifying `anisotropy`:

```
R> sppois <- spglm(
+  formula = count ~ elev + strat + elev:strat,
+  family = poisson,
+  data = moose,
+  spcov_type = "gaussian",
+  anisotropy = TRUE
+)
R>
R> spnbin <- update(sppois, family = nbinomial)
```

Because the models have the same support (i.e., both non-negative count models), we can use likelihood-based statistics to compare them:

```
R> glances(sppois, spnbin, sort_by = "AIC") |>
+  subset(select = c(model, npar, AIC, AICc, BIC))
```

```
# A tibble: 2 x 5
  model    npar   AIC  AICc   BIC
  <chr>   <int> <dbl> <dbl> <dbl>
1 spnbin      6 1318. 1319. 1339.
2 sppois      5 1320. 1321. 1337.
```

The negative binomial model has a slightly lower AIC and AICc, while the Poisson model has a slightly lower BIC. This is reasonable given the BIC penalizes additional parameters (here, an overdispersion parameter) more heavily than AIC and AICc. The leave-out-out MSPE prefers the negative binomial model:

```
R> loocv(sppois) |>
+  apply(2, round, digits = 4)
```

```
   bias    MSPE   RMSPE 
 1.2882 31.7882  5.6381 
```

```
R> loocv(spnbin) |>
+  apply(2, round, digits = 4)
```

```
   bias    MSPE   RMSPE 
 0.3485 28.3760  5.3269 
```

A likelihood-based comparison between the negative binomial anisotropic model and the negative binomial isotropic model suggests that the anisotropic model is preferred (lower AIC, AICc, and BIC):

```
R> spnbin_iso <- update(spnbin, anisotropy = FALSE)
R> glances(spnbin_iso, spnbin, sort_by = "AIC") |>
+  subset(select = c(model, npar, AIC, AICc, BIC))
```

```
# A tibble: 2 x 5
  model       npar   AIC  AICc   BIC
  <chr>      <int> <dbl> <dbl> <dbl>
1 spnbin         6 1318. 1319. 1339.
2 spnbin_iso     4 1333. 1333. 1346.
```

`plot()` returns the spatial covariance as a function of direction (Figure 7):

```
R> plot(spnbin_iso, which = 8)
R> plot(spnbin, which = 8)
```

Earlier we used `tidy()` to tidy the model's fixed effects, but we can also use tidy to tidy the spatial covariance parameters:

```
R> tidy(spnbin, effects = "spcov")
```

```
# A tibble: 5 x 3
  term    estimate is_known
  <chr>      <dbl> <lgl>   
1 de         4.65  FALSE   
2 ie         0.148 FALSE   
3 range  84013.    FALSE   
4 rotate     2.74  FALSE   
5 scale      0.259 FALSE   
```

The `rotate` parameter is the number of radians in $[0, \pi]$ the ellipse is rotated and the `scale` parameter is the ratio in $(0, 1]$ of the minor axis length to the major axis length. The `is_known`

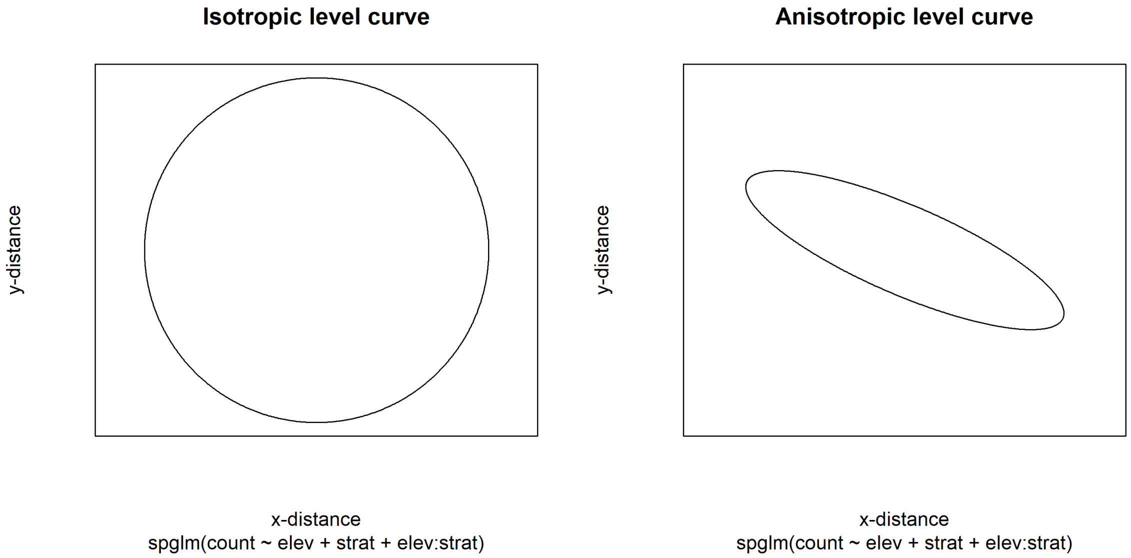

Figure 7: Level curves of equal autocorrelation for the negative binomial moose count models. The ellipse is centered at zero distance in the x-direction and y-direction, and points along the ellipse have equal levels of autocorrelation. In the isotropic level curve (left), spatial covariance decays equally in all directions. In the anistropic level curve (right), spatial covariance decays fastest in the northeast-southwest direction and slowest in the northwest-southeast direction (this pattern can be seen in the observed counts).

column indicates whether the parameter was assumed known during optimization, controlled by specifying the `spcov_initial` argument.

3.3. Skewed data

The `seal` data in **spmodel** is an `sf` object with data on harbor seal trends in Alaska. The `log_trend` variable is the logarithm of a seal abundance temporal trend measure at the site (based on historical data), and the `stock` variable is a factor with two levels, `8` and `10`, where each level represents one of twelve seal stocks (i.e., breeds) in Alaska. The `seal` geometry is an areal polygon geometry and hence, spatial autoregressive models based on neighborhood distance are appropriate.

SPGLMs for areal data are are fit in **spmodel** using `spgautor()`, which has similar syntax as `spglm()` but contains arguments to control the weight matrix ($\mathbf{W}$ in Equation 12) and whether or not row-standardization (Ver Hoef, Peterson, Hooten, Hanks, and Fortin 2018) is applied. By default, polygons are neighbors if they share a boundary (i.e., Queen's contiguity; see Pebesma and Bivand (2023)) and row standardization is assumed. Weight matrices may be provided via the `W` argument and row standardization may be ignored via the `row_st` arguent. Following Ver Hoef *et al.* (2018), polygons without a neighbor are given their own (independent) variance parameter called `extra`.

The trend data were originally logged to remove skew (Ver Hoef *et al.* 2018), but we will exponentiate `log_trend` (Figure 8) and model this skew directly using a SPGLM:

```
R> seal$trend <- exp(seal$log_trend)
```

The `trend` variable has several missing (`NA`) values, which represent polygons at which predictions of `trend` are desired. To make predictions using spatial autoregressive models, the prediction locations must be known prior to model fitting because these locations affect the neighborhood structure of the observed data (Ver Hoef *et al.* 2018). This restriction is notably different than SPGLMs for point-referenced data (i.e., geostatistical models), which completely separates the estimation and prediction steps.

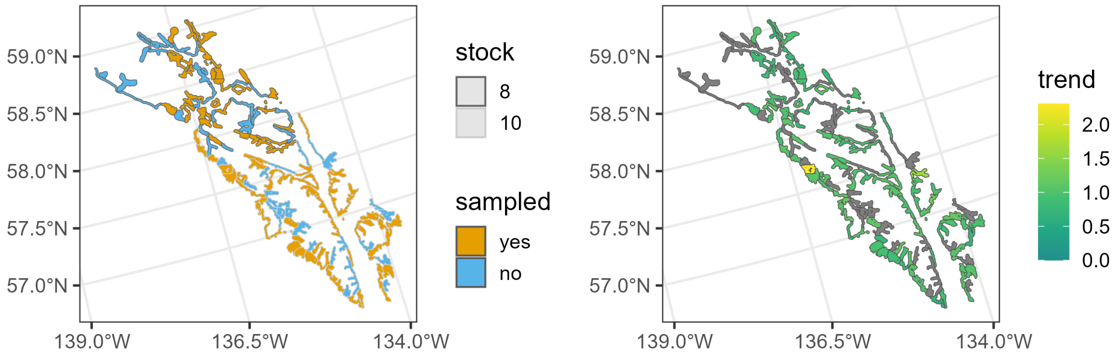

Figure 8: Seal trend distribution in Alaska. Observed and missing seal polygons by stock (left) and observed seal trends (right).

spmodel supports the gamma and inverse Gaussian families for modeling skewed, positive response variables. **spmodel** also supports nonspatial random effects specified via the `random` argument, which uses a similar formula syntax as **nlme** (Pinheiro and Bates 2006) and **lme4** (Bates, Mächler, Bolker, and Walker 2015). Using likelihood-based statistics, we compare two models fit using the simultaneous autoregressive covariance and the Gamma and inverse Gaussian families. Both models have a random effect for seal stock, which builds additional correlation into the model for two polygons from the same stock:

```
R> spgam <- spgautor(
+  formula = trend ~ 1,
+  family = Gamma,
+  data = seal,
+  spcov_type = "sar",
+  random = ~ stock
+)
R> spinvg <- update(spgam, family = inverse.gaussian)
R> glances(spgam, spinvg, sort_by = "AIC") |>
+  subset(select = c(model, npar, AIC, AICc, BIC))
```

```
# A tibble: 2 x 5
  model    npar   AIC  AICc   BIC
  <chr>   <int> <dbl> <dbl> <dbl>
1 spinvg      5  108.  109.  121.
2 spgam       5  114.  115.  127.
```

The inverse Gaussian model has a lower AIC, AICc, and BIC, which indicates it is a better fit than the gamma model.

We may `tidy()` the estimated stock random effect variance:

```
R> tidy(spinvg, effects = "randcov")
```

```
# A tibble: 1 x 3
  term      estimate is_known
  <chr>        <dbl> <lgl>
1 1 | stock  0.00411 FALSE
```

The locations to predict `trend` (`NA` values) are stored in the `newdata` element of `spinvg` and used for prediction:

```
R> # output omitted
R> predict(spinvg, type = "response", interval = "prediction")
```

If using `augment()` for prediction, `newdata` must be specified:

```
R> head(augment(
+   spinvg,
+   newdata = spinvg$newdata,
+   type.predict = "response",
+   interval = "prediction"
+))
```

```
Simple feature collection with 6 features and 6 fields
Geometry type: POLYGON
Dimension:     XY
Bounding box:  xmin: 1030504 ymin: 1012786 xmax: 1115097 ymax: 1057579
Projected CRS: NAD83 / Alaska Albers
# A tibble: 6 x 7
  log_trend stock trend .fitted .lower .upper                            geometry
      <dbl> <fct> <dbl>   <dbl>  <dbl>  <dbl>                       <POLYGON [m]>
1        NA 8        NA   0.942  0.893  0.993 ((1035002 1054710, 1035002 105454~
2        NA 8        NA   0.942  0.893  0.993 ((1043093 1020553, 1043097 102055~
3        NA 8        NA   0.942  0.893  0.993 ((1099737 1054310, 1099752 105426~
4        NA 8        NA   0.942  0.893  0.993 ((1099002 1036542, 1099134 103646~
5        NA 8        NA   0.942  0.893  0.993 ((1076902 1053189, 1076912 105317~
6        NA 8        NA   0.942  0.893  0.993 ((1070501 1046969, 1070317 104659~
```

3.4. Proportion data

We end with two examples of beta regression for proportion data (Ferrari and Cribari-Neto 2004). First, we model the nitrogen percentage in a caribou foraging experiment. Second, we model the proportion of voter turnout by Texas county in the United States (US) 1980 presidential election.

The `caribou` data in **spmodel** are a data frame from a caribou foraging experiment in Alaska meant to study the impact of water and tarp cover on the percentage of nitrogen in surrounding

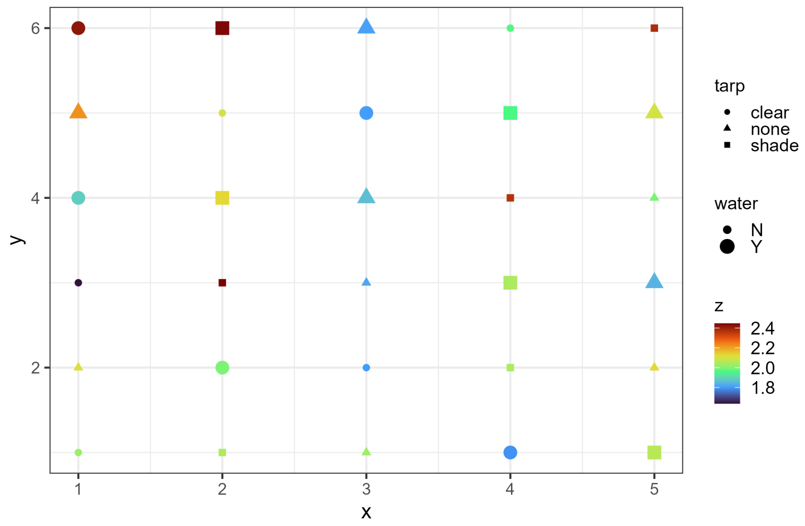

Figure 9: Caribou data from a spatial experimental design measuring the percentage of nitrogen (`z`) in soil and testing two factors: tarp type and water presence.

plants (Figure 9). Lenart, Bowyer, Ver Hoef, and Ruess (2002) studied these data treating nitrogen percentage as a continuous variable, but here we treat nitrogen percentage (`z`) as a proportion:

```
R> spbeta <- spglm(
+    formula = z/100 ~ water + tarp + water:tarp,
+    family = "beta",
+    data = caribou,
+    spcov_type = "matern",
+    xcoord = x,
+    ycoord = y
+)
```

The nitrogen percentage is dynamically scaled in `formula` from (0, 100) to (0, 1) so that it is a proportion. Nitrogen percentage is modeled as a function of `water` (two levels: `"Y"` for water and `"N"` for no water), `tarp` (three levels: `"clear"` for a clear tarp, `"none"` for no tarp, and `"shade"` for a shaded tarp), and their interaction, which lets the effect of water presence vary across tarp type. `caribou` is a data frame (not an `sf` object), so we supply the x-coordinate and y-coordinate directly via `xcoord` and `ycoord`, and, consistent with the tidyverse approach (Wickham, Averick, Bryan, Chang, McGowan, François, Grolemund, Hayes, Henry, Hester *et al.* 2019), column names in `data` do not need to be quoted when referenced (but can be quoted).

A summary of `spbeta` returns a coefficients table that provides parameter estimates relative to a reference group. When factors have more than two levels, it is not straightforward to use these contrasts to determine overall significance of the factor. The `anova()` function tests marginal (i.e., Type III sums of squares) significance of factors using the general linear hypothesis test for spatial (i.e., correlated) data (Schabenberger and Gotway 2017):

```
tidy(anova(spbeta))
```

```
# A tibble: 4 x 4
  effects        df statistic       p.value
  <chr>       <int>     <dbl>         <dbl>
1 (Intercept)     1  35783.   0
2 water           1      1.52 0.218
3 tarp            2     38.4  0.00000000468
4 water:tarp      2      5.74 0.0566
```

These results suggest there is some evidence that the effect of water on nitrogen percentage differ depending on the type of tarp used ($0.01 < p$ value < 0.1).

Sometimes averages or contrasts between factor levels that are not in the reference group are of interest. We again leverage **spmodel**'s built-in support for **emmeans** and use it to obtain the averages of each factor combination on the link (here, logit) scale:

```
spemm <- emmeans(spbeta, ~ water + tarp)
spemm
```

```
 water tarp  emmean     SE  df asymp.LCL asymp.UCL
 N     clear  -3.94 0.0208 Inf     -3.98     -3.90
 Y     clear  -3.90 0.0206 Inf     -3.94     -3.86
 N     none   -3.88 0.0204 Inf     -3.92     -3.84
 Y     none   -3.91 0.0206 Inf     -3.95     -3.87
 N     shade  -3.77 0.0196 Inf     -3.81     -3.73
 Y     shade  -3.83 0.0200 Inf     -3.87     -3.79

Degrees-of-freedom method: asymptotic
Results are given on the logit (not the response) scale.
Confidence level used: 0.95
```

Delta method (Ver Hoef 2012) standard errors are used when averages on the response (here, proportion) scale are desired:

```
update(spemm, type = "response")
```

```
 water tarp  response       SE  df asymp.LCL asymp.UCL
 N     clear   0.0191 0.000390 Inf    0.0183    0.0198
 Y     clear   0.0198 0.000398 Inf    0.0190    0.0205
 N     none    0.0202 0.000404 Inf    0.0194    0.0210
 Y     none    0.0197 0.000398 Inf    0.0189    0.0205
 N     shade   0.0226 0.000434 Inf    0.0218    0.0235
 Y     shade   0.0213 0.000418 Inf    0.0205    0.0221

Degrees-of-freedom method: asymptotic
Confidence level used: 0.95
Intervals are back-transformed from the logit scale
```

Pairwise contrasts use a Tukey p value adjustment (Tukey 1949) by default. Here, we request no p value adjustment:

```
head(pairs(spemm, adjust = "none"))

 contrast          estimate     SE  df z.ratio p.value
 N clear - Y clear  -0.0361 0.0293 Inf  -1.233  0.2177
 N clear - N none   -0.0601 0.0292 Inf  -2.063  0.0391
 N clear - Y none   -0.0327 0.0293 Inf  -1.118  0.2637
 N clear - N shade  -0.1735 0.0286 Inf  -6.061  <.0001
 N clear - Y shade  -0.1136 0.0289 Inf  -3.932  0.0001
 Y clear - N none   -0.0241 0.0290 Inf  -0.830  0.4064

Degrees-of-freedom method: asymptotic
Results are given on the log odds ratio (not the response) scale.
```

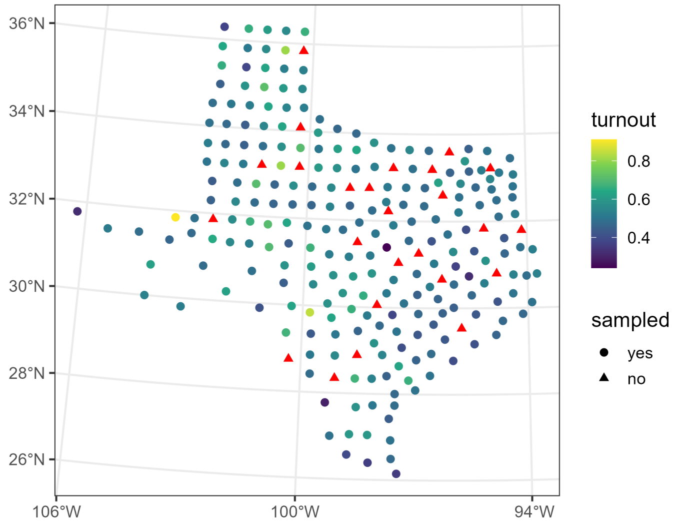

Figure 10: Proportion of voter turnout in Texas for the 1980 presidential election. Circles represent voter turnout (based on color) and triangles represent locations at which voter turnout predictions are desired.

We now model the `elect80` data in **spData** (Bivand, Nowosad, and Lovelace 2024), which contains voter turnout data by county in the 1980 US Presidential election (Pace and Barry 1997). The `texas` data in **spmodel** contains a subset of these data in the state of Texas. These data are point-referenced, but we may still use autoregressive models if neighborhood distance is determined using county centroids (i.e., counties whose centroid distance is less than some cutoff are defined as neighbors). The response variable of interest, `turnout`, is the proportion of registered voters in the county who voted in the election (Figure 10).

```
R> spgautor_mods <- spgautor(
+  formula = turnout ~ log_income,
+  family = beta,
+  data = texas,
+  spcov_type = c("car", "sar"),
+  cutoff = 2e5,
+  estmethod = "ml"
+)
```

We model voter turnout as a function of log income using both the conditional and simultaneous autoregressive models with a neighbor distance cutoff of 200 kilometers and the maximum likelihood estimation method. When a vector is provided to `spcov_type` in `spgautor()` (or `spglm()`), a model is fit for each spatial covariance type and stored in a list with name equal to the respective type. Then it is simple to glance at each model fit:

```
R> glances(spgautor_mods) |>
+  subset(select = c(model, npar, AIC, AICc, BIC))

# A tibble: 2 x 5
  model  npar   AIC  AICc   BIC
  <chr> <int> <dbl> <dbl> <dbl>
1 car       3 -38.3 -38.1 -21.2
2 sar       3 -35.7 -35.5 -18.6
```

The conditional autoregressive model has the best fit (in terms of AIC, AICc, and BIC). In this model, there is significant evidence `log_income` is positively related to average voter turnout (p value < 0.001):

```
tidy(spgautor_mods$car)

# A tibble: 2 x 5
  term        estimate std.error statistic    p.value
  <chr>          <dbl>     <dbl>     <dbl>      <dbl>
1 (Intercept)   -4.74      1.06      -4.47 0.00000783
2 log_income     0.532     0.117      4.55 0.00000531
```

Another way to assess the impact of `log_income` on `turnout` is a likelihood ratio test:

```
reduced_car <- update(spgautor_mods$car, formula = turnout ~ 1)
tidy(anova(reduced_car, spgautor_mods$car))

# A tibble: 1 x 5
  full              reduced        df statistic     p.value
  <chr>             <chr>       <dbl>     <dbl>       <dbl>
1 spgautor_mods$car reduced_car     1      24.1 0.000000930
```

Likelihood ratio tests compare the fit of a "full" model compared to a "reduced" model that is completely nested within the full model. If there is evidence the full model explains significantly more information than the reduced model, the likelihood ratio test p value will be small. Similar to the summary output from the general linear hypothesis test, the likelihood ratio suggests `log_income` is related to average voter turnout (p value < 0.001).

The default estimation method in **spmodel** is REML, but note that these models used maximum likelihood (ML). ML is very similar to REML – the difference is that for ML, the fixed effects $\boldsymbol{\beta}$ are not integrated out of Equation (4) but are rather back-substituted. While REML typically performs better for fixed effect estimation and prediction (Zimmerman and Ver Hoef 2024), ML allows likelihood-based comparisons (e.g., AIC) for models with simultaneously varying fixed effect and covariance structures, while REML likelihood-based comparisons are only valid for models sharing the same fixed effect structure (though Gurka (2006) provides some evidence that this restriction may be unnecessary).

The point-referenced `texas` and `caribou` data may be analyzed using `spgautor()` or `spglm()` and comparisons across these structures can be made using likelihood-based statistics (as long as the supports of the response distribution are the same). Put another way, likelihood-based statistics can be used to determine whether geostatistical (distance-based) or autoregressive (neighbor-based) structures perform best when the data are point-referenced.

4. Discussion

SPGLMs are fit in **spmodel** using a novel application of the Laplace approximation that marginalizes over the latent (i.e., unobserved) mean, $\mathbf{w}$, and the fixed effects, $\boldsymbol{\beta}$. The approach is quite flexible and accommodates any general response distribution and covariance structure. Ver Hoef *et al.* (2024) show that the approach, as implemented in **spmodel**, generally yields unbiased estimators with proper interval coverage and often outperforms the Bayesian approach from **spBayes**, the INLA approach from **R-INLA**, and the automatic differentiation approach from **glmmTMB**.

spmodel's `spglm()` and `spgautor()` functions are similar in structure and syntax as the base-R `glm()` function, easing the transition from GLMs to SPGLMs. These functions support six response distributions (Table 1) and 20 spatial covariance functions. **spmodel** provides several additional features that accommodate geometric anisotropy, nonspatial random effects, fixing spatial covariance parameters at known values, data having thousands of observations (following Ver Hoef, Dumelle, Higham, Peterson, and Isaak (2023)), incorporating spatial dependence in machine learning (e.g., random forests; Breiman (2001)), simulating spatially dependent data (e.g., `sprbinom()`, `sprpois()`), and several others. Learn more at https://CRAN.R-project.org/package=spmodel and links therein.

Computational details

The results in this paper were obtained using R 4.4.0 with the **spmodel** 0.9.0 package. Figures were created using the **ggplot2** 3.5.1 package (Wickham 2016) and base R.

Data and code availability

All writing and code associated with this manuscript is available for viewing and download on GitHub at https://github.com/USEPA/spmodel.glm.manuscript. All data used are part of the **spmodel** R package available for download from CRAN at https://CRAN.R-project.org/package=spmodel.

Acknowledgments

We would like to thank initial reviewers and editors for feedback that has greatly improved the manuscript.

The views expressed in this manuscript are those of the authors and do not necessarily represent the views or policies of the U.S. Environmental Protection Agency or the National Oceanic and Atmospheric Administration. Any mention of trade names, products, or services does not imply an endorsement by the U.S. government, the U.S. Environmental Protection Agency, or the National Oceanic and Atmospheric Administration. The U.S. Environmental Protection Agency and the National Oceanic and Atmospheric Administration do not endorse any commercial products, services or enterprises.

References

Akaike H (1974). "A New Look at the Statistical Model Identification." *IEEE Transactions on Automatic Control*, **19**(6), 716–723.

Bates D, Mächler M, Bolker B, Walker S (2015). "Fitting Linear Mixed-Effects Models Using lme4." *Journal of Statistical Software*, **67**(1), 1–48. doi:10.18637/jss.v067.i01.

Bivand R, Nowosad J, Lovelace R (2024). *spData: Datasets for Spatial Analysis*. R package version 2.3.1, URL https://CRAN.R-project.org/package=spData.

Bolker BM, Brooks ME, Clark CJ, Geange SW, Poulsen JR, Stevens MHH, White JSS (2009). "Generalized Linear Mixed Models: A Practical Guide for Ecology and Evolution." *Trends in Ecology & Evolution*, **24**(3), 127–135.

Breiman L (2001). "Random forests." *Machine Learning*, **45**, 5–32.

Breslow NE, Clayton DG (1993). "Approximate Inference in Generalized Linear Mixed Models." *Journal of the American Statistical Association*, **88**(421), 9–25.

Brooks ME, Kristensen K, van Benthem KJ, Magnusson A, Berg CW, Nielsen A, Skaug HJ, Maechler M, Bolker BM (2017). "glmmTMB Balances Speed and Flexibility Among Packages for Zero-Inflated Generalized Linear Mixed Modeling." *The R Journal*, **9**(2), 378–400. doi:10.32614/RJ-2017-066.

Bürkner PC (2017). "brms: An R package for Bayesian Multilevel Models Using Stan." *Journal of Statistical Software*, **80**, 1–28.

Chambers JM, Hastie TJ (eds.) (1992). *Statistical Models in S*. Chapman & Hall, London.

Dumelle M, Higham M, Ver Hoef JM (2023). "spmodel: Spatial Statistical Modeling and Prediction in R." *PLOS ONE*, **18**(3), e0282524.

Faraway JJ (2016). *Extending the Linear Model with R: Generalized Linear, Mixed Effects and Nonparametric Regression Models*. CRC press.

Ferrari S, Cribari-Neto F (2004). "Beta Regression for Modelling Rates and Proportions." *Journal of applied statistics*, **31**(7), 799–815.

Finley AO, Banerjee S, Carlin BP (2007). "spBayes: An R Package for Univariate and Multivariate Hierarchical Point-Referenced Spatial Models." *Journal of Statistical Software*, **19**(4), 1–24. URL https://www.jstatsoft.org/article/view/v019i04.

Finley AO, Datta A, Banerjee S (2022). "spNNGP R Package for Nearest Neighbor Gaussian Process Models." *Journal of Statistical Software*, **103**(5), 1–40. doi:10.18637/jss.v103.i05.

Gurka MJ (2006). "Selecting the Best Linear Mixed Model Under REML." *The American Statistician*, **60**(1), 19–26.

Harville DA (1977). "Maximum Likelihood Approaches to Variance Component Estimation and to Related Problems." *Journal of the American Statistical Association*, **72**(358), 320–338.

Hoeting JA, Davis RA, Merton AA, Thompson SE (2006). "Model Selection for Geostatistical Models." *Ecological Applications*, **16**(1), 87–98.

Hughes J, Cui X (2020). *ngspatial: Fitting the Centered Autologistic and Sparse Spatial Generalized Linear Mixed Models for Areal Data*. Frederick, MD. R package version 1.2-2.

James G, Witten D, Hastie T, Tibshirani R (2013). *An Introduction to Statistical Learning*. Springer.

Kuhn M, Silge J (2022). *Tidy Modeling with R*. O'Reilly Media, Inc.

Lee D (2013). "CARBayes: An R Package for Bayesian Spatial Modeling with Conditional Autoregressive Priors." *Journal of Statistical Software*, **55**(13), 1–24.

Lee Y, Nelder JA (1996). "Hierarchical Generalized Linear Models." *Journal of the Royal Statistical Society: Series B (Methodological)*, **58**(4), 619–656.

Lenart EA, Bowyer T, Ver Hoef J, Ruess R (2002). "Climate Change and Caribou: Effects of Summer Weather on Forage." *Canadian Journal of Zoology*, **80**(4), 664–678.

Lenth RV (2024). *emmeans: Estimated Marginal Means, aka Least-Squares Means*. R package version 1.10.3, URL https://CRAN.R-project.org/package=emmeans.

Lindgren F, Rue H (2015). "Bayesian Spatial Modelling with R-INLA." *Journal of Statistical Software*, **63**, 1–25.

McCullagh P, Nelder JA (1989). *Generalized Linear Models, Second Edition*. Chapman and Hall Ltd.

Montgomery DC, Peck EA, Vining GG (2021). *Introduction to Linear Regression Analysis*. John Wiley & Sons.

Myers RH, Montgomery DC, Vining GG, Robinson TJ (2012). *Generalized Linear Models: With Applications in Engineering and the Sciences*. John Wiley & Sons.

Nelder JA, Wedderburn RW (1972). "Generalized Linear Models." *Journal of the Royal Statistical Society: Series A (General)*, **135**(3), 370–384.

Pace RK, Barry R (1997). "Quick Computation of Sspatial Autoregressive Estimators." *Geographical analysis*, **29**(3), 232–247.

Patterson D, Thompson R (1971). "Recovery of Inter-Block Information when Block Sizes are Unequal." *Biometrika*, **58**(3), 545–554.

Pebesma E (2018). "Simple Features for R: Standardized Support for Spatial Vector Data." *The R Journal*, **10**(1), 439–446. doi:10.32614/RJ-2018-009. URL https://doi.org/10.32614/RJ-2018-009.

Pebesma E, Bivand R (2023). *Spatial data Science: With Applications in R*. Chapman and Hall/CRC.

Pinheiro J, Bates D (2006). *Mixed-Effects Models in S and S-PLUS*. Springer Science & Business Media.

R Core Team (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Robinson D, Hayes A, Couch S (2021). *broom: Convert Statistical Objects into Tidy Tibbles*. R package version 0.7.6, URL https://CRAN.R-project.org/package=broom.

Ronnegard L, Shen X, Alam M (2010). "hglm: A Package for Fitting Hierarchical Generalized Linear Models." *The R Journal*, **2**(2), 20–28.

Rousset F, Ferdy JB (2014). "Testing Environmental and Genetic Effects in the Presence of Spatial Autocorrelation." *Ecography*, **37**(8), 781–790. URL https://dx.doi.org/10.1111/ecog.00566.

Sainsbury-Dale M, Zammit-Mangion A, Cressie N (2024). "Modeling Big, Heterogeneous, Non-Gaussian Spatial and Spatio-Temporal Data Using FRK." *Journal of Statistical Software*, **108**, 1–39.

Schabenberger O, Gotway CA (2017). *Statistical Methods for Spatial Data Analysis*. CRC press.

Schwarz G (1978). "Estimating the Dimension of a Model." *The Annals of Statistics*, pp. 461–464.

Tobler WR (1970). "A Computer Movie Simulating Urban Growth in the Detroit Region." *Economic Geography*, **46**(sup1), 234–240.

Tukey JW (1949). "Comparing Individual Means in the Analysis of Variance." *Biometrics*, pp. 99–114.

Ver Hoef JM (2012). "Who Invented the Delta Method?" *The American Statistician*, **66**(2), 124–127.

Ver Hoef JM, Blagg E, Dumelle M, Dixon PM, Zimmerman DL, Conn PB (2024). "Marginal Inference for Hierarchical Generalized Linear Mixed Models with Patterned Covariance Matrices Using the Laplace Approximation." *Environmetrics*, **35**(7), e2872. doi:10.1002/env.2872.

Ver Hoef JM, Dumelle M, Higham M, Peterson EE, Isaak DJ (2023). "Indexing and Partitioning the Spatial Linear Model for Large Data Sets." *PLOS ONE*, **18**(11), e0291906.

Ver Hoef JM, Peterson EE, Hooten MB, Hanks EM, Fortin MJ (2018). "Spatial Autoregressive Models for Statistical Inference From Ecological Data." *Ecological Monographs*, **88**(1), 36–59.

Wedderburn RW (1974). "Quasi-Likelihood Functions, Generalized Linear Models, and the Gauss—Newton Method." *Biometrika*, **61**(3), 439–447.

Wickham H (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. ISBN 978-3-319-24277-4. URL https://ggplot2.tidyverse.org.

Wickham H, Averick M, Bryan J, Chang W, McGowan LD, François R, Grolemund G, Hayes A, Henry L, Hester J, *et al.* (2019). "Welcome to the Tidyverse." *Journal of Open Source Software*, **4**(43), 1686.

Wolfinger R, O'connell M (1993). "Generalized Linear Mixed Models: A Pseudo-Likelihood Approach." *Journal of Statistical Computation and Simulation*, **48**(3-4), 233–243.

Wolfinger R, Tobias R, Sall J (1994). "Computing Gaussian Likelihoods and their Derivatives for General Linear Mixed Models." *SIAM Journal on Scientific Computing*, **15**(6), 1294–1310.

Wood SN (2017). *Generalized Additive Models: An Introduction with R*. CRC press.

Zimmerman DL, Ver Hoef JM (2024). *Spatial Linear Models for Environmental Data*. CRC Press.

Affiliation:

Michael Dumelle
Pacific Ecological Systems Division
200 SW 35th St
Corvallis OR
E-mail: dumelle.michael@epa.gov

Jay M. Ver Hoef
Alaska Fisheries Science Center

Matt Higham
Department of Math, Computer Science, and Statistics

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd