spmodel Workshop

Spatial Statistics 2023: Climate and the Environment

Michael Dumelle (presenting)
EPA (USA)

Matt Higham (presenting)
St. Lawrence University (USA)

Jay M Ver Hoef
NOAA (USA)

7/18/23

Welcome!

1. Install and load R packages

```
install.packages("spmodel")
library(spmodel)
install.packages("ggplot2")
library(ggplot2)
```

2. Please visit https://usepa.github.io/spmodel.spatialstat2023/ to view the workshop's accompanying workbook
3. (Optional) Download the workshop's slides (instructions in the workbook's "Introduction")
4. Follow along and have fun!

Who Are We?

Michael Dumelle is a statistician for the United States Environmental Protection Agency (USEPA). He works primarily on facilitating the survey design and analysis of USEPA's National Aquatic Resource Surveys (NARS), which characterize the condition of waters across the United States. His primary research interests are in spatial statistics, survey design, environmental and ecological applications, and software development.

Who Are We?

Matt Higham is an assistant professor of statistics at St. Lawrence University, a small liberal arts college in Canton, New York. His primary research interests are in spatial statistics and in applications of statistics to ecological settings. He enjoys using R to teach undergraduate students the foundations of statistical computing and data science.

Who Are We?

Jay Ver Hoef is a senior scientist and statistician for the Marine Mammal Lab, part of the Alaska Fisheries Science Center of NOAA Fisheries, located in Seattle, Washington, although Jay lives in Fairbanks, Alaska. He is a fellow of the American Statistical Association, and Jay consults on a wide variety of topics related to marine mammals and stream networks. His main statistical interests are in spatial statistics and Bayesian statistics, especially applied to ecological and environmental data.

Disclaimer

The views expressed in this workshop are those of the authors and do not necessarily represent the views or policies of the U.S. Environmental Protection Agency or the U.S. 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 U.S. National Oceanic and Atmospheric Administration. The U.S. Environmental Protection Agency and the U.S. National Oceanic and Atmospheric Administration do not endorse any commercial products, services, or enterprises.

What is `spmodel`?

`spmodel` is an `R` package to fit, summarize, and predict for a variety of spatial statistical models. Some of the things that `spmodel` can do include:

- Fit spatial linear and generalized linear models for point-referenced and areal (lattice) data
- Compare model fits and inspect model diagnostics
- Predict at unobserved spatial locations (i.e., Kriging)
- And much more!

Why use `spmodel`?

There are many great spatial modeling packages in `R`. A few reasons to use `spmodel` for spatial analysis are that:

The Basics

Goals

1. Fit a spatial linear model using `splm()`.
2. Tidy, glance at, and augment the fitted model.
3. Predict for unobserved locations (i.e., Kriging).

The Sulfate Data

The `sulfate` data in `spmodel` contains data on 197 sulfate measurements in the continental United States

```
1 head(sulfate)
```

```
Simple feature collection with 6 features and 1 field
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: 162932.8 ymin: 1080571 xmax: 914593.6 ymax: 1453636
Projected CRS: NAD83 / Conus Albers
  sulfate                  geometry
1  12.925 POINT (817738.8 1080571)
2  20.170 POINT (914593.6 1295545)
3  16.822 POINT (359574.1 1178228)
4  16.227 POINT (265331.9 1239089)
5   7.858 POINT (304528.8 1453636)
6  15.358 POINT (162932.8 1451625)
```

```
1 ggplot(sulfate, aes(color = sulfate)) +
2   geom_sf(size = 3) +
3   scale_color_viridis_c(limits = c(0, 45)) +
4   theme_gray(base_size = 18)
```

The Sulfate Data

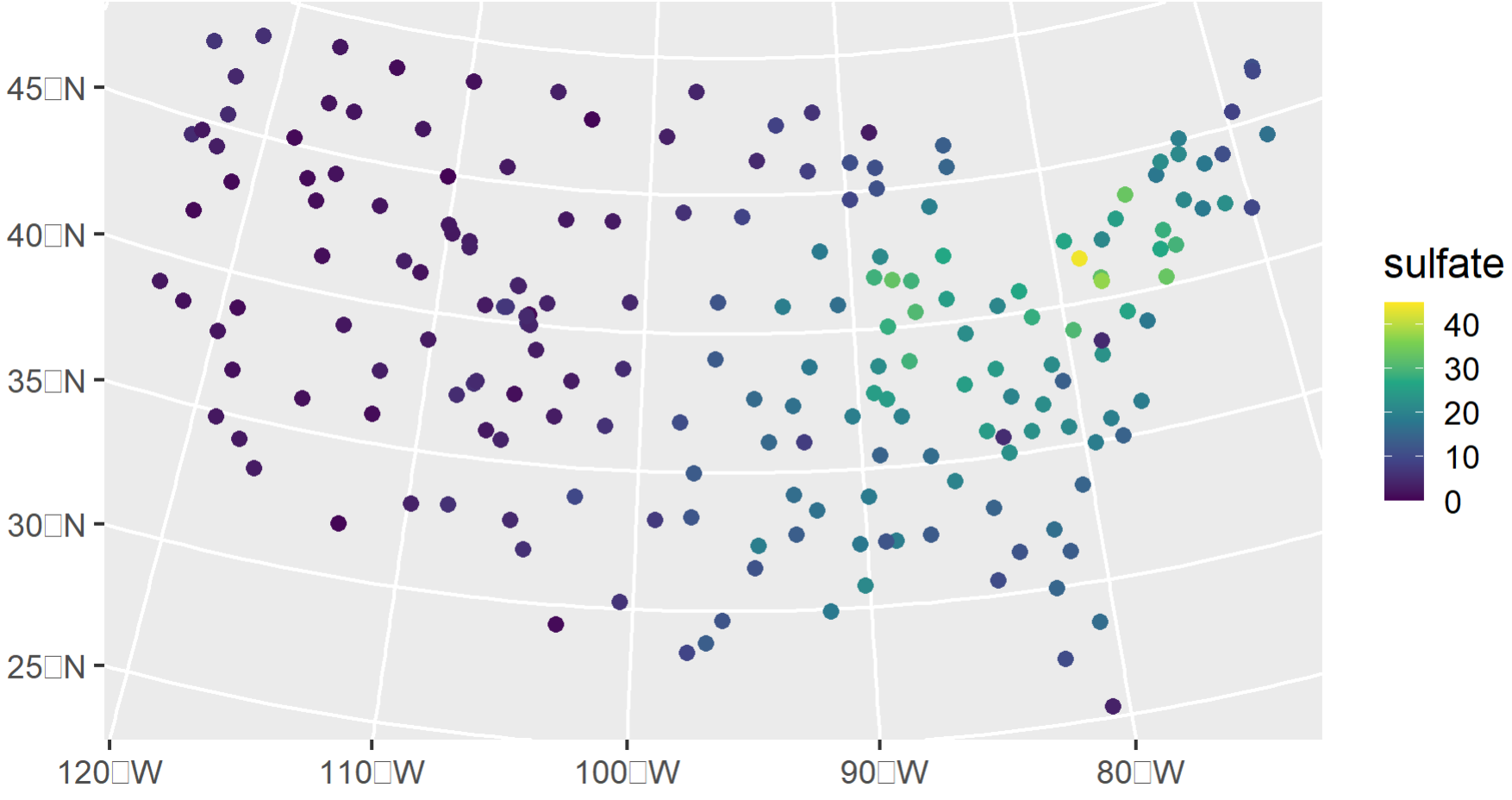

Figure 1: Distribution of sulfate data.

Fitting a Model

We fit and summarize a spatial linear model with an intercept by running

```
spmod <- splm(sulfate ~ 1, data = sulfate, spcov_type = "exponential")
summary(spmod)
```

```
Call:
splm(formula = sulfate ~ 1, data = sulfate, spcov_type = "exponential")

Residuals:
   Min     1Q Median     3Q    Max
-5.738 -2.605  4.900 13.323 38.099

Coefficients (fixed):
            Estimate Std. Error z value Pr(>|z|)
(Intercept)    5.924      6.529   0.907    0.364

Coefficients (exponential spatial covariance):
       de        ie     range
     85.8      10.4 3105165.7
```

The broom Functions

Tidy the fixed effect output

```
1 tidy(spmod)
```

```
# A tibble: 1 × 5
  term        estimate std.error statistic p.value
  <chr>          <dbl>     <dbl>     <dbl>   <dbl>
1 (Intercept)     5.92      6.53     0.907   0.364
```

Glance at the model fit

```
1 glance(spmod)
```

```
# A tibble: 1 × 9
      n     p  npar value   AIC  AICc logLik deviance pseudo.r.squared
  <int> <dbl> <int> <dbl> <dbl> <dbl>  <dbl>    <dbl>              <dbl>
1   197     1     3 1140. 1146. 1146.  -570.     196.                  0
```

Augment the data with model diagnostics

```
1 augment(spmod)
```

The broom Functions

```
Simple feature collection with 197 features and 6 fields
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: -2292550 ymin: 386181.1 xmax: 2173345 ymax: 3090370
Projected CRS: NAD83 / Conus Albers
# A tibble: 197 × 7
   sulfate .fitted .resid    .hat   .cooksd .std.resid           geometry
 *   <dbl>   <dbl>  <dbl>   <dbl>     <dbl>      <dbl>        <POINT [m]>
 1  12.9      5.92   7.00 0.00334 0.00161      -0.694  (817738.8 1080571)
 2  20.2      5.92  14.2  0.00256 0.00192       0.865  (914593.6 1295545)
 3  16.8      5.92  10.9  0.00259 0.000395      0.390  (359574.1 1178228)
 4  16.2      5.92  10.3  0.00239 0.000363      0.390  (265331.9 1239089)
 5   7.86     5.92   1.93 0.00202 0.00871      -2.07   (304528.8 1453636)
 6  15.4      5.92   9.43 0.00201 0.000240      0.345  (162932.8 1451625)
 7   0.986    5.92  -4.94 0.00380 0.000966     -0.503  (-1437776 1568022)
```

Prediction (i.e., Kriging)

```
predict(spmod, newdata = sulfate_preds)
```

Augment prediction data

```
aug_preds <- augment(spmod, newdata = sulfate_preds)
print(aug_preds)
```

Prediction (i.e., Kriging)

```
Simple feature collection with 100 features and 1 field
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: -2283774 ymin: 582930.5 xmax: 1985906 ymax: 3037173
Projected CRS: NAD83 / Conus Albers
# A tibble: 100 × 2
   .fitted            geometry
 *   <dbl>         <POINT [m]>
 1    1.62  (-1771413 1752976)
 2   24.4    (1018112 1867127)
 3    8.95 (-291256.8 1553212)
 4   16.5    (1274293 1267835)
 5    4.93 (-547437.6 1638825)
 6   26.8    (1445080 1981278)
 7    2.87  (-1629090 3037173)
 8   14.2    (1202757 1030524)
```

Prediction (i.e., Kriging)

Visualize predictions

```
ggplot(aug_preds, aes(color = .fitted)) +
  geom_sf(size = 3) +
  scale_color_viridis_c(limits = c(0, 45)) +
  theme_gray(base_size = 18)
```

Prediction (i.e., Kriging)

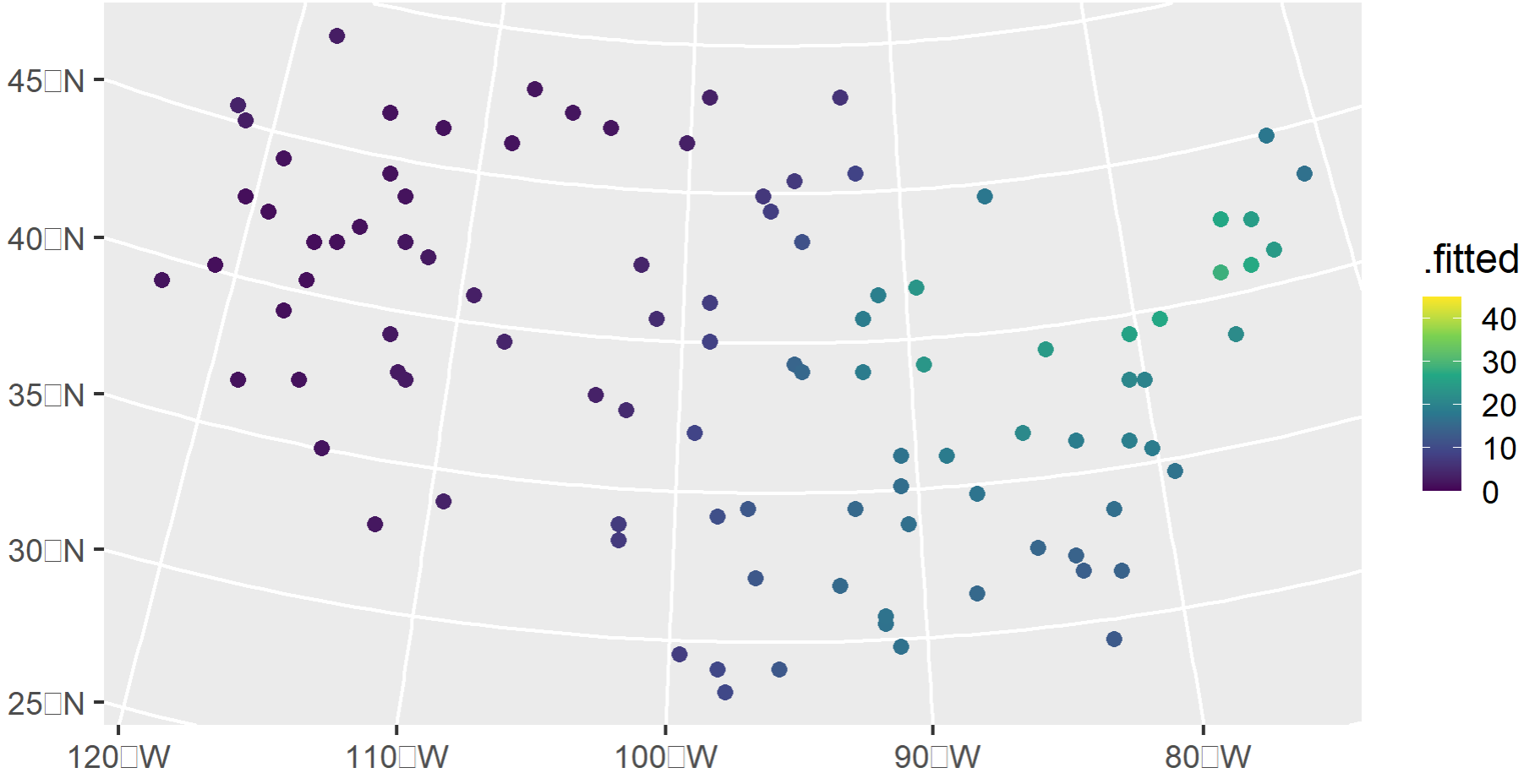

Figure 2: Distribution of sulfate data predictions.

Your Turn

> Say hi to your neighbors! What type of work or analyses do you do in the field of spatial statistics?

05:00

Your Turn

> Visit `spmodel`'s website at https://usepa.github.io/spmodel. Navigate to the "References" tab and explore some other functions available in `spmodel`.

05:00

The Spatial Linear Model

Goals

1. Explain how the spatial linear model differs from the linear model with independent random errors.
2. Explain how modeling for point-referenced data with distance-based model covariances differs from modeling for areal data with neighborhood-based model covariances.

Independent Random Error Model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \qquad (1)$$

- $\boldsymbol{\epsilon}$ is a column vector of random errors.
  - $\mathrm{E}(\boldsymbol{\epsilon}) = \mathbf{0}$
  - $\mathrm{Cov}(\boldsymbol{\epsilon}) = \sigma^2_{ie}\mathbf{I}$
- $\mathbf{y}$ is a column vector of response variables, $\mathbf{X}$ is a design (model) matrix of explanatory variables, and $\boldsymbol{\beta}$ is a column vector of fixed effects.

Spatial Linear Model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\tau} + \boldsymbol{\epsilon}, \qquad (2)$$

- $\boldsymbol{\tau}$ is a random error vector independent of $\boldsymbol{\epsilon}$
  - $\mathrm{E}(\boldsymbol{\tau}) = \mathbf{0}$
  - $\mathrm{Cov}(\boldsymbol{\tau}) = \sigma^2_{de}\mathbf{R}$
- $\mathrm{Cov}(\mathbf{y}) \equiv \boldsymbol{\Sigma} = \sigma^2_{de}\mathbf{R} + \sigma^2_{ie}\mathbf{I}$

Point-Referenced Data

- Data are point-referenced when the elements in $\mathbf{y}$ are observed at point-locations indexed by x-coordinates and y-coordinates on a spatially continuous surface with an infinite number of locations
  - Consider sampling soil at any point-location in a field
- Distance-based correlation for point-referenced data:
  - Exponential Correlation: $\mathbf{R} = \exp(-\mathbf{H}/\phi)$
  - $\mathbf{H}$ is a matrix of distances, ϕ is the range parameter
- Fit in `spmodel` using `splm()` (the workshop's focus)

Areal Data

Your Turn

> Navigate to the Help file for `splm` by running `?splm` or by visiting this link and scroll down to “Details.” Examine the spatial linear model description in the Help file and relate some of the syntax used to the syntax used in this section.

05:00

Your Turn

> With your neighbor(s), verify that you reach the same conclusions in the previous exercise, relating the syntax in the Help file to the syntax in the spatial linear model from this section.

03:00

Model Fitting

Goals

1. Further explore the `splm()` function using the `moss` data.
2. Connect parameter estimates in the summary output of `splm()` to the spatial linear model.

The Moss Data

The `moss` data contains a variable for log Zinc concentration for moss samples collected near a mining road in Alaska.

```
ggplot(moss, aes(color = log_Zn)) +
  geom_sf(size = 2) +
  scale_color_viridis_c() +
  scale_x_continuous(breaks = seq(-163, -164, length.out = 2)) +
  theme_gray(base_size = 18)
```

The Moss Data

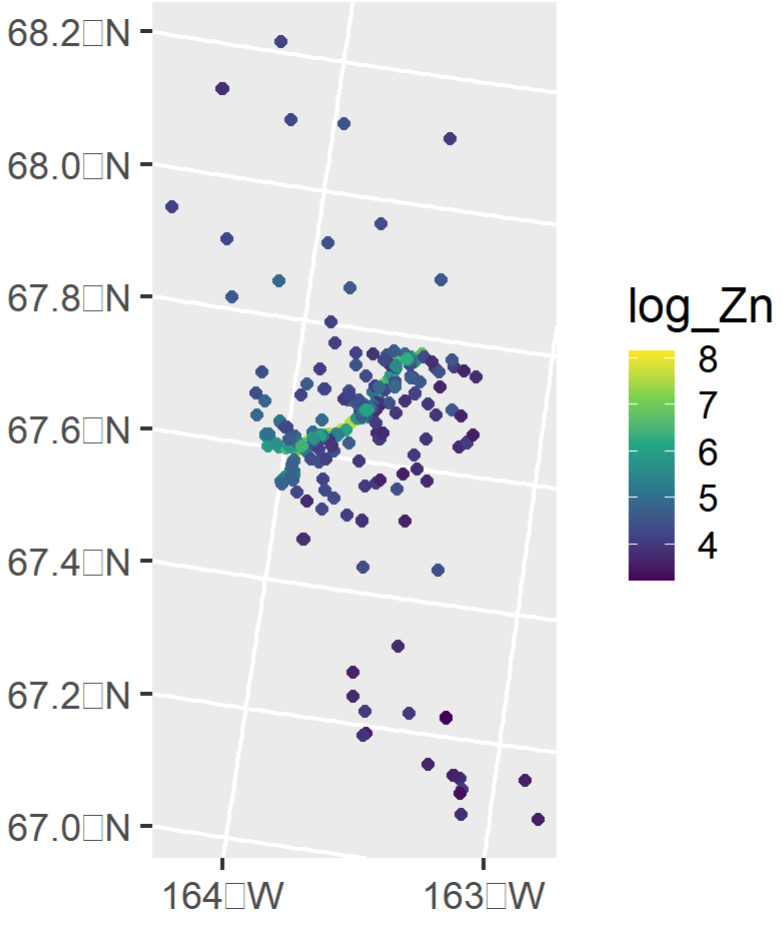

Figure 3: Distribution of moss data.

The Moss Data

sideroad	log_dist2road	log_Zn	geometry
N	2.68	7.33	POINT (-413585.3 1997623)
N	2.68	7.38	POINT (-413585.3 1997623)
N	2.54	7.58	POINT (-415367.2 1996769)

The `splm()` function

The `splm()` function shares syntactic structure with the `lm()` function and generally requires at least three arguments

1. `formula`: a formula that describes the relationship between the response variable ($\mathbf{y}$) and explanatory variables ($\mathbf{X}$)
2. `data`: a `data.frame` or `sf` object that contains the response variable, explanatory variables, and spatial information.
3. `spcov_type`: the spatial covariance type (`"exponential"`, `"matern"`, `"spherical"`, etc; 17 total types)

Fit a Spatial Linear Model

Estimation Methods:

- Restricted maximum likelihood (default; likelihood-based)
- Maximum likelihood (likelihood-based)
- Composite likelihood (semivariogram-based) (Curriero and Lele 1999)
- Weighted least squares (semivariogram-based; see `weights`) (Cressie 1985)

```
spmod <- splm(formula = log_Zn ~ log_dist2road, data = moss,
              spcov_type = "exponential")
summary(spmod)
```

Fit a Spatial Linear Model

```
Call:
splm(formula = log_Zn ~ log_dist2road, data = moss, spcov_type =
"exponential")

Residuals:
    Min      1Q  Median      3Q     Max
-2.6801 -1.3606 -0.8103 -0.2485  1.1298

Coefficients (fixed):
             Estimate Std. Error z value Pr(>|z|)
(Intercept)    9.76825    0.25216   38.74   <2e-16 ***
log_dist2road -0.56287    0.02013  -27.96   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The Spatial Linear Model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\tau} + \boldsymbol{\epsilon}, \qquad (3)$$

- $\text{Cov}(\boldsymbol{\tau}) = \sigma^2_{de}\mathbf{R}$
- $\text{Cov}(\mathbf{y}) \equiv \boldsymbol{\Sigma} = \sigma^2_{de}\mathbf{R} + \sigma^2_{ie}\mathbf{I}$
- Exponential Correlation: $\mathbf{R} = \exp(-\mathbf{H}/\phi)$

Your Turn

Another data set contained within the `spmodel` package is the `caribou` data set. Fit a spatial linear model with

- `z` as the response and `tarp`, `water`, and the interaction between `tarp` and `water` as predictors
- a spatial covariance model for the errors of your choosing
- `x` as the `xcoord` and `y` as the `ycoord`

After fitting the model, perform an analysis of variance using `anova()` to assess the importance of `tarp`, `water`, and `tarp:water`.

Your Turn

> Compare your anova model output to the anova model output of your neighbor(s). Are the inferences made on the fixed effects very different for your two choices of spatial covariance model?

03:00

Tidying Output

The `tidy()` function tidies fixed effect model output into a convenient `tibble` (a special `data.frame`)

```
1 tidy(spmod)
```

```
# A tibble: 2 × 5
  term          estimate std.error statistic p.value
  <chr>            <dbl>     <dbl>     <dbl>   <dbl>
1 (Intercept)      9.77     0.252      38.7         0
2 log_dist2road   -0.563    0.0201    -28.0         0
```

It can also be used to tidy spatial covariance parameters

```
1 tidy(spmod, effects = "spcov")
```

```
# A tibble: 3 × 3
  term   estimate is_known
  <chr>     <dbl> <lgl>
1 de        0.360 FALSE
2 ie        0.0790 FALSE
3 range  8237.    FALSE
```

Model Fit

The `glance()` function returns columns with the sample size (`n`), number of fixed effects (`p`), number of estimated covariance parameters (`npar`), optimization criteria minimum (`value`), AIC (`AIC`), AICc (`AICc`), log-likelihood (`loglik`), deviance (`deviance`), and pseudo R-squared (`pseudo.r.squared`)

```
1 glance(spmod)
```

```
# A tibble: 1 × 9
      n     p  npar value   AIC  AICc logLik deviance pseudo.r.squared
  <int> <dbl> <int> <dbl> <dbl> <dbl>  <dbl>    <dbl>             <dbl>
1   365     2     3  367.  373.  373.  -184.     363.             0.683
```

Your Turn

The `glances()` function allows us to compare the model fit statistics for a few different models simultaneously. Use `splm()` to fit a model with a Matérn spatial covariance (by specifying `"matern"` for `spcov_type`) and a model with no spatial covariance (by specifying `"none"` for `spcov_type`). Then, use the `glances()` function, providing each model object as an argument, to compare the model fit statistics of each model.

05:00

Model Diagnostics

The `augment()` function returns columns with

- `.fitted`, the fitted value, calculated from the estimated fixed effects in the model
- `.hat`, the Mahalanobis distance, a metric of leverage
- `.cooksd`, the Cook's distance, a metric of influence
- `.std.resid`, the standardized residual

```
1 augment(spmod)
```

Your Turn

> Use `spmodel`'s plot function on the spmod object to construct a plot of the fitted spatial covariance vs spatial distance. To learn more about the options for `spmodel`'s plot function, run `?plot.spmodel` or visit this link.

Your Turn

> Use `spmodel`'s plot function on the `spmod` object to construct any of the other 6 types of plots possible for this type of model. Then, explain what the plot is showing to your neighbor(s).

05:00

Prediction

Goals

1. Predict the response value at an unobserved location for point-referenced data.
2. Calculate leave-one-out cross-validation residuals.

The Moose Data

The `moose` data contains moose counts and moose presence for 218 spatial locations in Alaska.

```
ggplot(data = moose, aes(colour = count)) +
  geom_sf(size = 2) +
  scale_colour_viridis_c(limits = c(0, 40)) +
  theme_minimal(base_size = 18)
```

The Moose Data

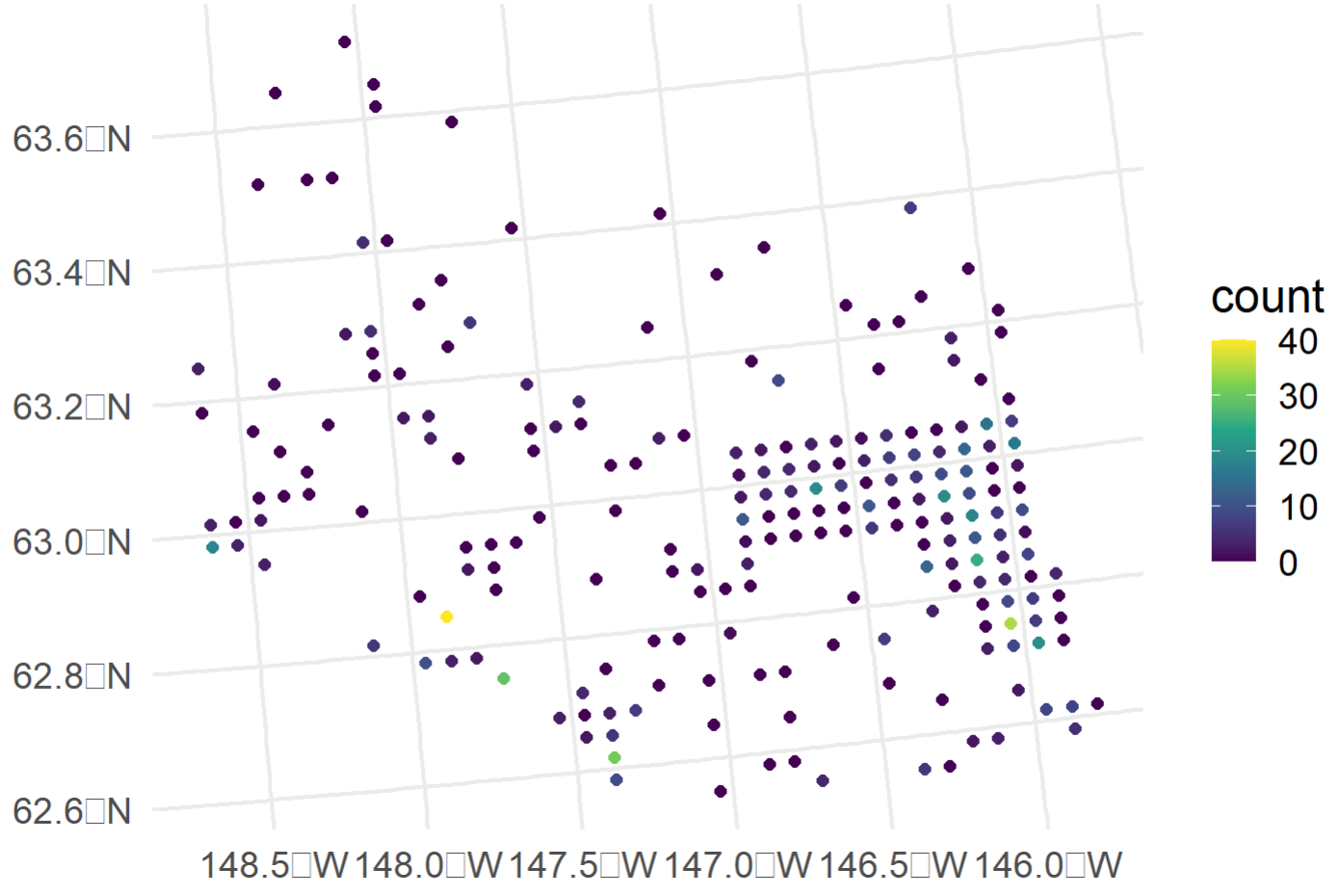

Figure 4: Distribution of moose data.

The Moose Data

elev	strat	count	presence	geometry
469	L	0	0	POINT (293542.6 1541016)
362	L	0	0	POINT (298313.1 1533972)
173	M	0	0	POINT (281896.4 1532516)

The Moose Prediction Data

The `moose_preds` data contains spatial locations that were not surveyed.

elev	strat	geometry
143.4000	L	POINT (401239.6 1436192)
324.4375	L	POINT (352640.6 1490695)
158.2632	L	POINT (360954.9 1491590)

Fit a Spatial Linear Model

```
moosemod <- splm(count ~ elev * strat, data = moose,
                 spcov_type = "spherical")
tidy(moosemod)
```

```
# A tibble: 4 × 5
  term        estimate std.error statistic p.value
  <chr>          <dbl>     <dbl>     <dbl>   <dbl>
1 (Intercept)   0.310    9.02       0.0344 0.973
2 elev          0.0141   0.00806    1.76   0.0792
3 stratM        6.93     2.26       3.07   0.00217
4 elev:stratM  -0.0273   0.0130    -2.10   0.0357
```

Predictions

Using `predict()`

```
predict(moosemod, newdata = moose_preds)
```

Using `augment()`

```
moose_aug <- augment(moosemod, newdata = moose_preds)
moose_aug
```

Predictions

```
Simple feature collection with 100 features and 3 fields
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: 269386.2 ymin: 1418453 xmax: 419976.2 ymax: 1541763
Projected CRS: NAD83 / Alaska Albers
# A tibble: 100 × 4
    elev strat .fitted           geometry
 * <dbl> <chr>   <dbl>        <POINT [m]>
 1  143. L       3.45  (401239.6 1436192)
 2  324. L       1.59  (352640.6 1490695)
 3  158. L      -0.267 (360954.9 1491590)
 4  221. M       2.39  (291839.8 1466091)
 5  209. M       7.62  (310991.9 1441630)
 6  218. L      -1.02  (304473.8 1512103)
 7  127. L      -1.23  (339011.1 1459318)
```

Your Turn

> Examine the help file `?augment.spmodel` or by visiting this link and create site-wise 99% prediction intervals for the unsampled locations found in `moose_preds`.

05:00

Cross Validation

The `loocv()` function can be used to perform leave-one-out cross validation on a fitted model object using mean-squared-prediction error (MSPE) loss:

```
1 loocv(moosemod)
```

```
[1] 32.15933
```

Your Turn

Fit a model with `count` as the response variable from the `moose` data with a `"spherical"` spatial covariance model for the random errors but no predictors as fixed effects. Compare the MSPE from leave-one-out cross-validation for this model with the previously fit `moosemod`. Which model is better, according to the leave-one-out cross-validation criterion?

Then, for the model with the lower MSPE, obtain the leave-one-out cross validation predictions and their standard errors. Hint: run `?loocv` or visit this link.

Additional Modeling Features

Goals

Incorporate additional arguments to `splm()` to:

- Fit and predict for multiple models simultaneously.
- Fit a spatial linear model with non-spatial random effects.
- Fit a spatial linear model with anisotropy.
- Fit a spatial linear model with a partition factor.
- Fix certain spatial covariance parameters at known values.

Multiple Models

Provide a vector of `spcov_type`s:

```
spmods <- splm(formula = log_Zn ~ log_dist2road, data = moss,
               spcov_type = c("exponential", "gaussian"))
names(spmods)
```

```
[1] "exponential" "gaussian"
```

Natural to combine with `glances()` and `predict()`:

```
glances(spmods)
```

```
# A tibble: 2 × 10
  model          n     p  npar value   AIC  AICc logLik deviance
pseudo.r.squared
  <chr>      <int> <dbl> <int> <dbl> <dbl> <dbl>  <dbl>    <dbl>
<dbl>
1 exponent…    365     2     3  367.  373.  373.  -184.     363.
0.683
2 gaussian     365     2     3  435.  441.  441.  -218.     363.
0.686
```

Your Turn

Work with a neighbor to find 90% confidence intervals for the fixed effects in the Gaussian model with one of two functions: (1) `tidy()` or (2) `confint()`. Before beginning, decide with your neighbor who will work finding the intervals with (1) `tidy()` and who will work on finding the intervals with (2) `confint()`.

05:00

Non-Spatial Random Effects

sample	log_dist2road	log_Zn	geometry
001PR	2.68	7.33	POINT (-413585.3 1997623)
001PR	2.68	7.38	POINT (-413585.3 1997623)
002PR	2.54	7.58	POINT (-415367.2 1996769)
003PR	2.97	7.63	POINT (-417186 1995645)

The `random` Argument

In `splm()`, the `random` argument follows similar syntax to how random effects are specified in the `nlme` and `lme4` packages.

- `random = ~ group` and `random = (1 | group)` specify random intercepts for each level of `group`.
- `random = (x | group)` specifies random intercepts for group and for each level of `group` to have a different slope for `x`.

Non-Spatial Random Effects

```
randint <- splm(log_Zn ~ log_dist2road,
                data = moss, spcov_type = "exponential",
                random = ~ (1 | sample))
```

Non-Spatial Random Effects

```
1 summary(randint)
```

```
Call:
splm(formula = log_Zn ~ log_dist2road, data = moss, spcov_type =
"exponential",
    random = ~(1 | sample))

Residuals:
    Min      1Q  Median      3Q     Max
-2.6234 -1.3228 -0.8026 -0.2642  1.0998

Coefficients (fixed):
             Estimate Std. Error z value Pr(>|z|)
(Intercept)    9.66066    0.26770   36.09   <2e-16 ***
log_dist2road -0.55028    0.02071  -26.58   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

More Complex Random Effects Models

```
yearint <- splm(log_Zn ~ log_dist2road,
                        data = moss, spcov_type = "exponential",
                        random = ~ (1 | sample + year))
yearsl <- splm(log_Zn ~ log_dist2road,
                        data = moss, spcov_type = "exponential",
                        random = ~ (1 | sample) +
                         (log_dist2road | year))
```

Your Turn

Perhaps a model with random intercepts for `sample` and random intercepts and slopes for `year` but without any spatial covariance is an even better fit to the data. Fit such a model by specifying `spcov_type` to be `"none"`. Then, use `glances()` to see how well this non-spatial model fits the `moss` data compared to the spatially explicit models.

05:00

Anisotropy

An anisotropic covariance does not behave similarly in all directions

- Consider a spatial covariance influenced by the prevailing wind direction.

```
aniso <- splm(log_Zn ~ log_dist2road,
              data = moss, spcov_type = "exponential",
              anisotropy = TRUE)
glances(spmod, aniso)
```

```
# A tibble: 2 × 10
  model     n     p  npar value   AIC  AICc logLik deviance pseudo.r.squared
  <chr> <int> <dbl> <int> <dbl> <dbl> <dbl>  <dbl>    <dbl>             <dbl>
1 aniso   365     2     5  362.  372.  372.  -181.     363              0.705
2 spmod   365     2     3  367.  373.  373.  -184.     363.             0.683
```

Anisotropy

```
summary(aniso)
```

```
Call:
splm(formula = log_Zn ~ log_dist2road, data = moss, spcov_type =
"exponential",
    anisotropy = TRUE)

Residuals:
    Min      1Q  Median      3Q     Max
-2.5279 -1.2239 -0.7202 -0.1921  1.1659

Coefficients (fixed):
             Estimate Std. Error z value Pr(>|z|)
(Intercept)    9.54798    0.22291   42.83   <2e-16 ***
log_dist2road -0.54601    0.01855  -29.44   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Your Turn

> Visualize the anisotropic level curve for `spmod` using `plot()`.
> Hint: Run `?plot.spmodel` or visit this link.

03:00

Your Turn

> Examine the `anisotropy` section of `Details` in the `splm()` help file with `?splm`. Then, with your neighbor, discuss which direction the model predicts two responses will be more correlated?

05:00

Partition Factors

A factor variable (i.e., categorical) is a partition factor when two observations in separate levels of the partition factor should be uncorrelated

- Observations from the same year are spatially correlated but observations from different years are not:

```
part <- splm(log_Zn ~ log_dist2road,
             data = moss, spcov_type = "exponential",
             partition_factor = ~ year)
```

Fixing Covariance Parameters

Steps:

1. Use `spcov_initial()` to specify the covariance type and any known, fixed covariance parameters.
2. Instead of specifying the `spcov_type` argument in `splm()`, specify the `spcov_initial` argument with the object from (1).

```
init_spher <- spcov_initial("spherical", range = 20000, known = "range")
splm(log_Zn ~ log_dist2road, data = moss,
     spcov_initial = init_spher)
```

Fixing Covariance Parameters

```
Call:
splm(formula = log_Zn ~ log_dist2road, data = moss, spcov_initial =
init_spher)


Coefficients (fixed):
  (Intercept)  log_dist2road
       9.7194        -0.5607


Coefficients (spherical spatial covariance):
       de          ie      range
4.545e-01  8.572e-02  2.000e+04
```

Your Turn

Fit a `"spherical"` spatial covariance model to the `moss` data set without a nugget effect (i.e., the model should have the `ie` independent variance parameter set to `0` and treated as `known`). Verify in the summary output that the `ie` is indeed `0` for this model.

05:00

Your Turn

With a neighbor, compare the fit of the `"spherical"` model with the `ie` variance parameter known and fixed at 0 (the no nugget model from the previous exercise) with the fit of a `"spherical"` model where all spatial covariance parameters are unknown and are estimated using (1) the `AIC` metric and (2) the `MSPE` from leave-one-out cross-validation using the `loocv()` function.

Before beginning work, decide who will complete (1) `AIC` and who will complete (2) `loocv()`.

05:00

Random Forest Spatial Residual Models

See workbook for an example of prediction using random forest spatial residual models

Large Data Sets

Goals

1. Use the `local` argument in `splm()` to fit a spatial linear model to a large data set.
2. Use the `local` argument in `predict()` (or `augment()`) to make predictions for a large data set.

Large Data Challenges

- Inversion of Σ for data sets with around 10,000 or more observations is challenging (and often unfeasible) on a standard computer
- `spmodel` implements "local" spatial indexing (Ver Hoef, Dumelle, et al. 2023) for model fitting
  - Induce some sparsity in the covariance matrix
  - Set `local = TRUE` in `splm()`
- See workbook for example
  - 5,000 observations fit in 12 seconds (no parallel)

Large Data Challenges

- `spmodel` uses "local neighborhood prediction" to predict for unobserved spatial locations for a model fit to a large data set
  - Only a subset of observations are used to predict the response at a particular location
  - Set `local = TRUE` in `predict()` or `augment()`
- See workbook for example
  - 3,000 predictions in 11.7 seconds (no parallel)

Generalized Linear Models

Goals

1. Explain how modeling spatial covariance fits within the structure of a generalized linear model.
2. Use the `spglm()` function in `spmodel` to fit generalized linear models for various model families (i.e., response distributions).

The Spatial Generalized Linear Model

$$g(\boldsymbol{\mu}) = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\tau} + \boldsymbol{\epsilon},$$

- $g(\boldsymbol{\mu})$ is the link function that "links" a function of the mean of $\mathbf{y}$ to $\mathbf{X}\boldsymbol{\beta}$, $\boldsymbol{\tau}$, and $\boldsymbol{\epsilon}$
- Fit models using a novel application of the the Laplace approximation (Ver Hoef, Blagg, et al. 2023)

Response Distributions

Response distributions and link functions available in `spmodel`

Distribution	Data Type	Link Function
Poisson	Count	Log
Negative Binomial	Count	Log
Binomial	Binary	Logit
Beta	Proportion	Logit
Gamma	Skewed	Log
Inverse Gaussian	Skewed	Log

The Moose Data

elev	strat	count	presence	geometry
468.9	L	0	0	POINT (293542.6 1541016)
362.3	L	0	0	POINT (298313.1 1533972)
172.8	M	0	0	POINT (281896.4 1532516)

Model Fitting

```
poismod <- spglm(count ~ elev * strat, data = moose,
                 family = poisson, spcov_type = "spherical")
```

- The `family` argument can be `binomial`, `beta`, `poisson`, `nbinomial`, `Gamma`, or `inverse.gaussian`
- Notice the similarities between `spglm()` and `glm()`

Model Fitting

```
1 summary(poismod)
```

```
Call:
spglm(formula = count ~ elev * strat, family = poisson, data = moose,
    spcov_type = "spherical")

Deviance Residuals:
    Min      1Q  Median      3Q     Max
-1.4245 -0.7783 -0.3653  0.1531  0.5900

Coefficients (fixed):
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.230575   0.958201  -2.328 0.019919 *
elev         0.007623   0.003129   2.437 0.014820 *
stratM       2.752234   0.782853   3.516 0.000439 ***
elev:stratM -0.010248   0.004472  -2.292 0.021928 *
```

Your Turn

Fit a spatial negative binomial model to the `moose` data with `count` as the response and `elev`, `strat`, and their interaction as predictors. The negative binomial model relaxes the assumption in the spatial Poisson generalized linear model that the mean of a response variable Y_i and the variance of a response variable Y_i must be equal. Obtain a summary of the fitted model. Then compare their fits using `loocv()`. Which model is preferable?

05:00

Prediction

Using `predict()`

```
predict(poismod, newdata = moose_preds)
```

Using `augment()`

```
augment(poismod, newdata = moose_preds)
```

Prediction

```
Simple feature collection with 100 features and 3 fields
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: 269386.2 ymin: 1418453 xmax: 419976.2 ymax: 1541763
Projected CRS: NAD83 / Alaska Albers
# A tibble: 100 × 4
    elev strat .fitted           geometry
 * <dbl> <chr>   <dbl>        <POINT [m]>
 1  143. L       0.207  (401239.6 1436192)
 2  324. L      -0.0563 (352640.6 1490695)
 3  158. L      -1.24   (360954.9 1491590)
 4  221. M      -1.16   (291839.8 1466091)
 5  209. M       1.78   (310991.9 1441630)
 6  218. L      -1.84   (304473.8 1512103)
 7  127. L      -2.80   (339011.1 1459318)
```

Additional Modeling Features

All advanced features available in `spmodel` for spatial linear models (`splm()`) are also available for spatial generalized linear models (`spglm()`)

- Fit and predict for multiple models
- Non-spatial random effects
- Anisotropy
- Etc.

Your Turn

Use `spglm()` to fit a spatial logistic regression model to the `moose` data using `presence` as the response variable and a `"cauchy"` covariance function. Then, find the predicted probabilities that moose are present at the spatial locations in `moose_preds` (Hint: Use the `type` argument in `predict()` or `augment()`).

Simulating Data

Goals

1. Simulate spatial Gaussian data using `sprnorm()`.
2. Simulate spatial binary, proportion, count, and skewed data using `sprbinom()`, `sprbeta()`, `sprpois()`, `sprnbinom()`, `sprgamma()`, and `sprinvgauss()`.

Simulating Gaussian Data

1. Use `spcov_params()` to specify the correlation structure and covariance parameters

```
params <- spcov_params("exponential", de = 1, ie = 0.5, range = 5e5)
```

2. Use `sprnorm(params, data)` to simulate a realization of the spatial process defined by `spcov_params()` at locations in `data`.

```
set.seed(1)
sulfate$z <- sprnorm(params, data = sulfate)
```

Simulating Gaussian Data

3. Visualize

```
ggplot(sulfate, aes(color = z)) +
  geom_sf(size = 2) +
  scale_color_viridis_c() +
  theme_gray(base_size = 14)
```

Simulating Gaussian Data

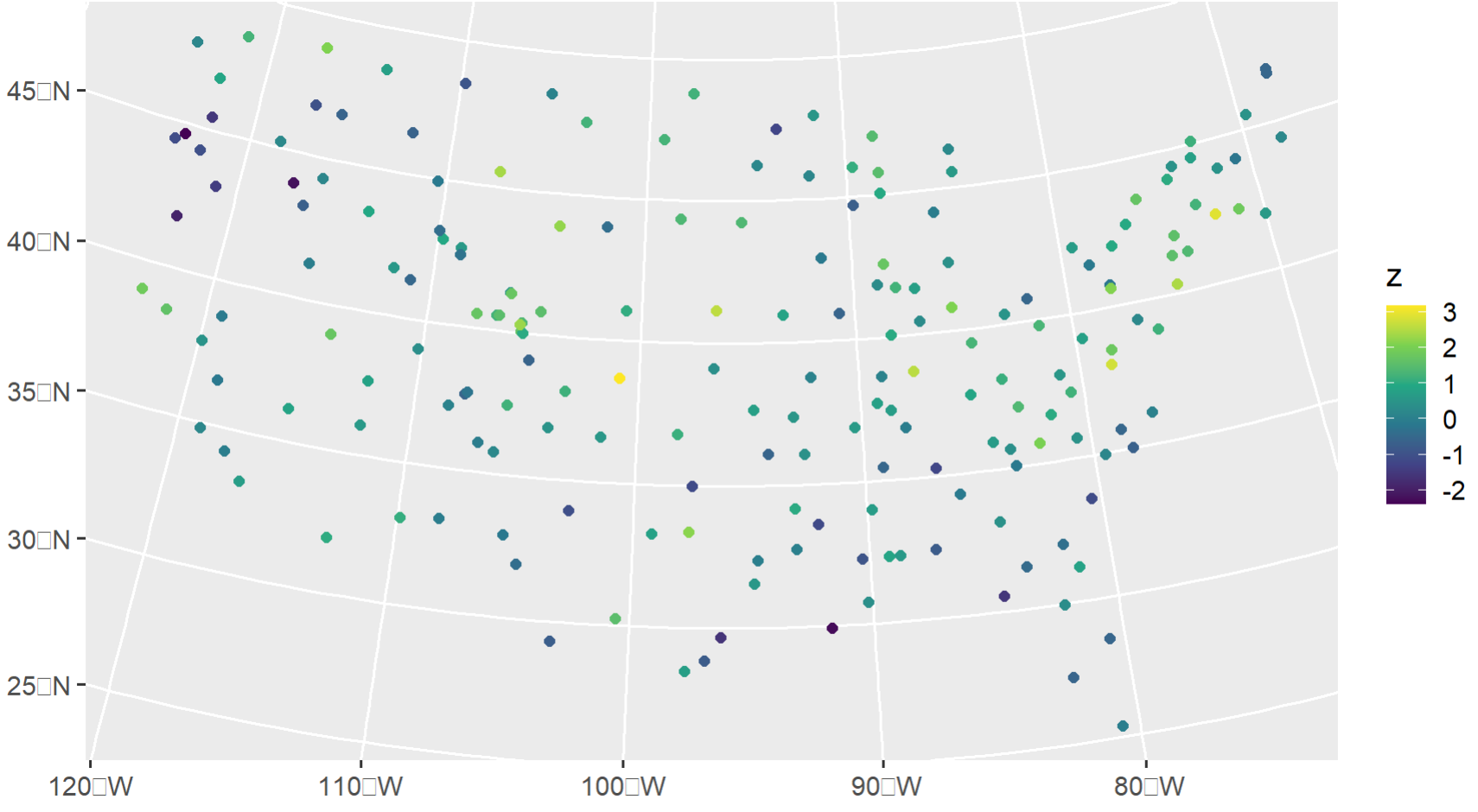

Figure 5: Distribution of simulated Gaussian data.

The Empirical Semivariogram

What does an empirical semivariogram of the simulated data look like?

```
esv_out <- esv(z ~ 1, sulfate)
ggplot(esv_out, aes(x = dist, y = gamma, size = np)) +
  geom_point() +
  lims(y = c(0, NA)) +
  theme_gray(base_size = 14)
```

The Empirical Semivariogram

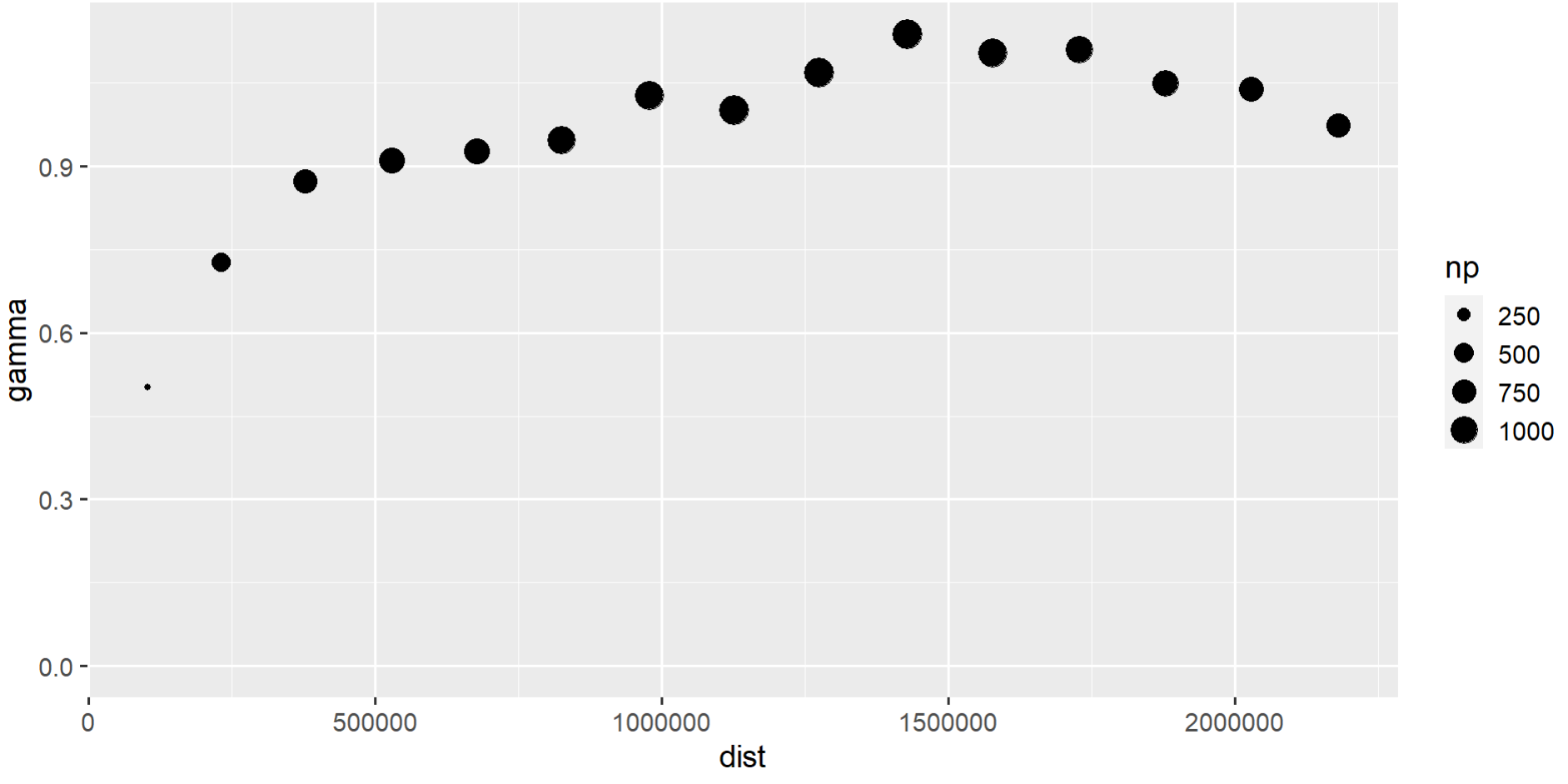

Figure 6: Empirical semivariogram of simulated Gaussian data.

Simulating Other Data

Use `sprpois(params, data)` to simulate a Poisson realization and visualize

```
sulfate$p <- sprpois(params, data = sulfate)
ggplot(sulfate, aes(color = p)) +
  geom_sf(size = 2) +
  scale_color_viridis_c() +
  theme_gray(base_size = 14)
```

Simulating Other Data

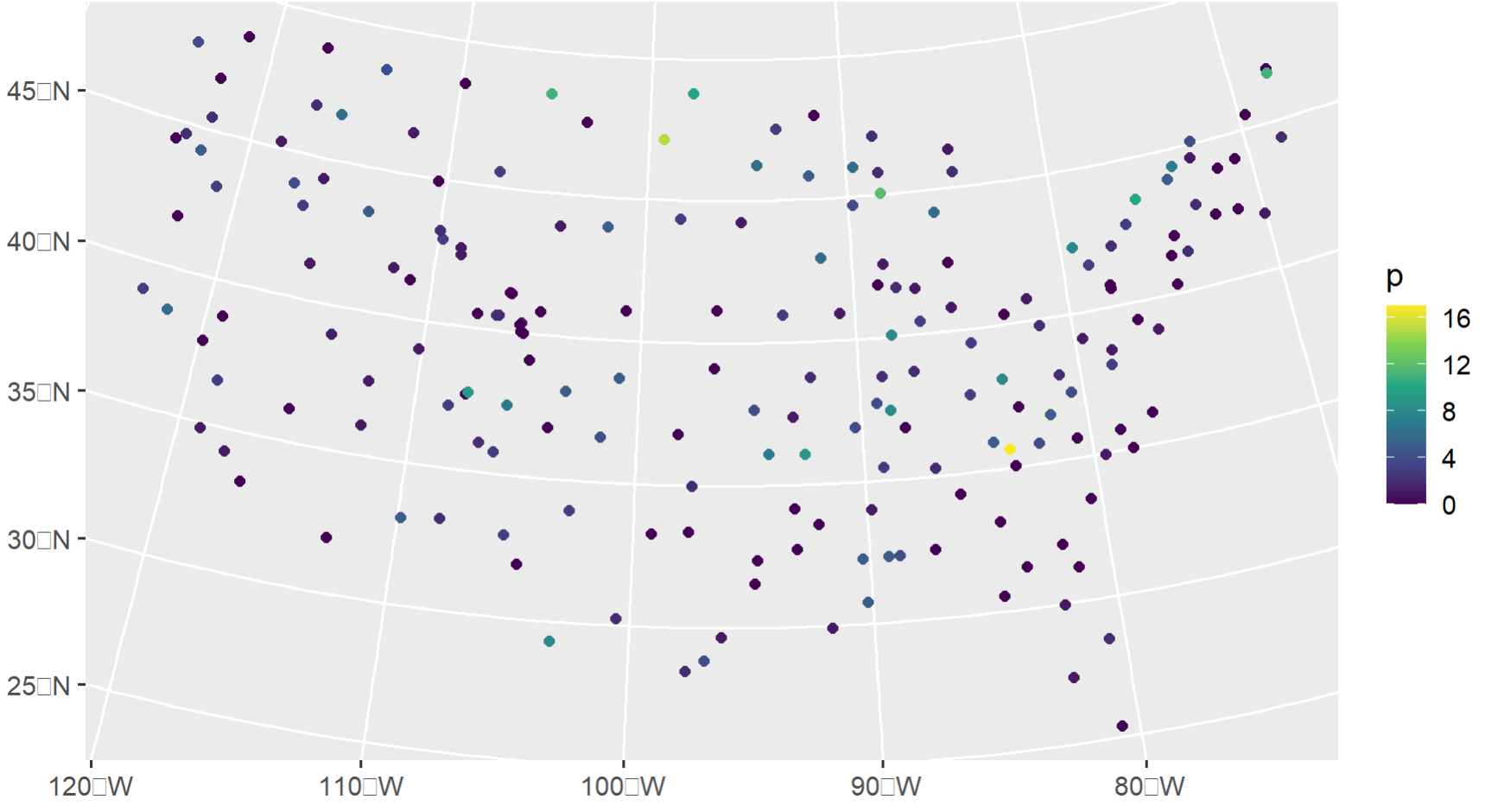

Figure 7: Distribution of simulated Poisson data.

A Caution

- Simulating spatial data in `spmodel` requires the Cholesky decomposition of the covariance matrix, which can take awhile for sample sizes exceeding 10,000
- Regardless of the number of realizations simulated, this Cholesky decomposition is only needed once
- This means that simulating many realizations (via `samples`) takes nearly the same time as simulating just one.

Areal Data

Goals

1. Use the `spautor()` function in `spmodel` to fit a spatial linear model to areal data.
2. Apply functions used for point-referenced data fit with `splm()` to areal data fit with `spautor()`.

The Seal Data

```
ggplot(seal, aes(fill = log_trend)) +
  geom_sf() +
  scale_fill_viridis_c() +
  theme_bw(base_size = 18)
```

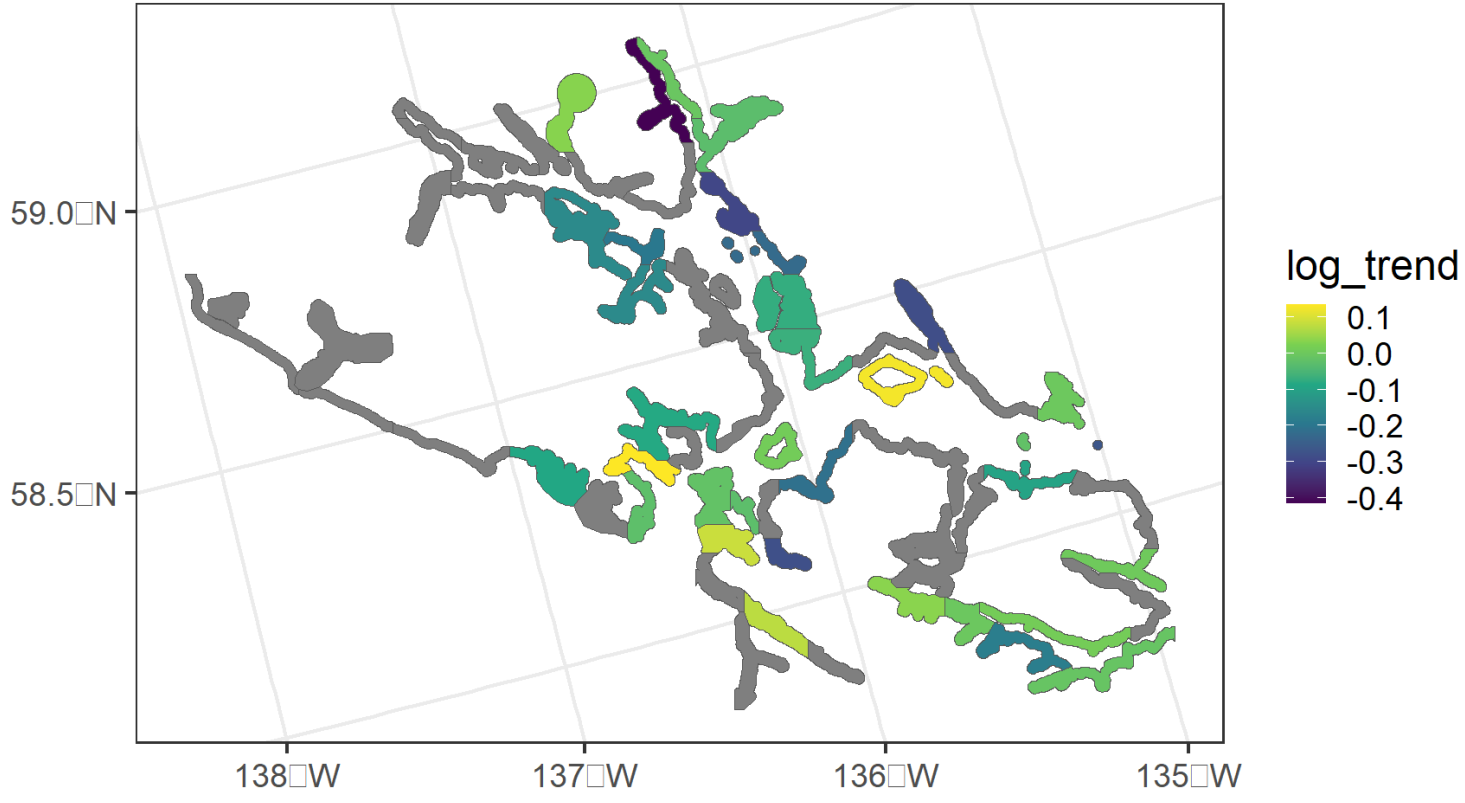

Figure 8: Distribution of the seal data.

Model Syntax

Model syntax for `spautor()` (`spgautor()`) is similar to the syntax used for `splm()` (`spglm()`):

```
1 sealmod <- spautor(log_trend ~ 1, data = seal, spcov_type = "car")
```

Model Output Interpretation

```
1 summary(sealmod)
```

```
Call:
spautor(formula = log_trend ~ 1, data = seal, spcov_type = "car")

Residuals:
     Min       1Q   Median       3Q      Max
-0.34441 -0.10403  0.04423  0.07351  0.20489

Coefficients (fixed):
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.07103    0.02492  -2.851  0.00436 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Coefficients (car spatial covariance):
```

Your Turn

> Choose a couple of helper functions that you would like to explore and apply those functions to the fitted seal model.

03:00

Your Turn

> Interpret your findings from the previous exercise with your neighbor(s), explaining which functions you selected to use and what the associated output means.

05:00

Prediction

Predictions must occur for “missing” (NA) responses in the observed data

Using `predict()`

```
predict(sealmod)
```

Using `augment()`

```
augment(sealmod, newdata = sealmod$newdata)
```

Prediction

```
Simple feature collection with 28 features and 2 fields
Geometry type: POLYGON
Dimension:     XY
Bounding box:  xmin: 913618.8 ymin: 1007542 xmax: 1115097 ymax: 1132682
Projected CRS: NAD83 / Alaska Albers
# A tibble: 28 × 3
   log_trend  .fitted
geometry
 *     <dbl>    <dbl>                                                <POLYGON
[m]>
 1        NA -0.115   ((1035002 1054710, 1035002 1054542, 1035002 1053542,
1035…
 2        NA -0.00918 ((1043093 1020553, 1043097 1020550, 1043101 1020550,
1043…
 3        NA -0.0603  ((1099737 1054310, 1099752 1054262, 1099788 1054278,
1099…
```

Your Turn

Verify that the fitted autoregressive model with the `seal` data changes when the polygons with missing response values are excluded from the `data` argument in `spautor()`. The following code creates a data without the polygons with missing values:

```
is_missing <- is.na(seal$log_trend)
seal_nomiss <- seal[!is_missing, , ]
```

Thank You!

Thank You!

- Thank you so much for attending
- Please reach out with comments / questions / suggestions / bugs (Dumelle.Michael@epa.gov)
- A6.02 16:00 - 16:15 Thursday, 20 July, 2023 UMC Glenn Miller Ballroom Middle
  - Slides available for download (instructions in the workshop's "Introduction")

References

Cressie, Noel. 1985. “Fitting Variogram Models by Weighted Least Squares.” *Journal of the International Association for Mathematical Geology* 17 (5): 563–86.

Curriero, Frank C, and Subhash Lele. 1999. “A Composite Likelihood Approach to Semivariogram Estimation.” *Journal of Agricultural, Biological, and Environmental Statistics*, 9–28.

Dumelle, Michael, Matt Higham, and Jay M. Ver Hoef. 2023. “spmodel: Spatial Statistical Modeling and Prediction in R.” *PLOS ONE* 18 (3): 1–32. https://doi.org/10.1371/journal.pone.0282524.

Ver Hoef, Jay M, Eryn Blagg, Michael Dumelle, Philip M Dixon, Dale L Zimmerman, and Paul Conn. 2023. “Marginal Inference for Hierarchical Generalized Linear Mixed Models with Patterned Covariance Matrices Using the Laplace Approximation.” *arXiv Preprint arXiv:2305.02978*.

Ver Hoef, Jay M, Michael Dumelle, Matt Higham, Erin E Peterson, and Daniel J Isaak. 2023. “Indexing and Partitioning the Spatial Linear Model for Large Data Sets.” *arXiv Preprint arXiv:2305.07811*.