

How Do We Assess the Performance of Our Small Area Estimators?

Jerzy Wieczorek¹, Kelly McConville², Grayson White³,
Tracey Frescino⁴, Gretchen Moisen⁴

¹ Colby College, ² Harvard University,
³ Michigan State University, ⁴ USDA Forest Service, RMRS

When estimating the mean of some forest attribute of interest, we have so many choices!

When estimating the mean of some forest attribute of interest, we have so many choices!

- Design based?

$$\hat{\mu} = \frac{1}{n} \sum_{i \in s} y_i$$

When estimating the mean of some forest attribute of interest, we have so many choices!

- Design based?
- Model-assisted?

$$\hat{\mu} = \frac{1}{n} \sum_{i \in s} y_i$$

$$\hat{\mu} = \frac{1}{N} \sum_{i \in U} \hat{y}_i + \frac{1}{n} \sum_{i \in s} (y_i - \hat{y}_i)$$

When estimating the mean of some forest attribute of interest, we have so many choices!

- Design based?
- Model-assisted?
- Model-based?

$$\hat{\mu} = \frac{1}{n} \sum_{i \in s} y_i$$

$$\hat{\mu} = \frac{1}{N} \sum_{i \in U} \hat{y}_i + \frac{1}{n} \sum_{i \in s} (y_i - \hat{y}_i)$$

$$\hat{\mu} = X' \beta + \nu + \epsilon$$

When estimating the mean of some forest attribute of interest, we have so many choices!

- Design based?
- Model-assisted?
- Model-based?
 - EBLUP?
 - Hierarchical Bayes?
 - Spatial?

$$\hat{\mu} = \frac{1}{n} \sum_{i \in s} y_i$$

$$\hat{\mu} = \frac{1}{N} \sum_{i \in U} \hat{y}_i + \frac{1}{n} \sum_{i \in s} (y_i - \hat{y}_i)$$

Model form?

$$\hat{\mu} = X' \beta + \nu + \epsilon$$

Priors?

Fitting method?

When estimating the mean of some forest attribute of interest, we have so many choices!

- Design based?
- Model-assisted?
- Model-based?
 - EBLUP?
 - Hierarchical Bayes?
 - Spatial?

$\hat{\mu} = \frac{1}{n} \sum_{i \in s} y_i$

$\hat{\mu} = \frac{1}{N} \sum_{i \in U} (y_i - \hat{y}_i)$

How do we choose?

Model form?

Priors?

Fitting method?

In order to address these choices, we have designed a unique simulation study using k nearest neighbors ($k\text{NN}$) and hot deck imputation.

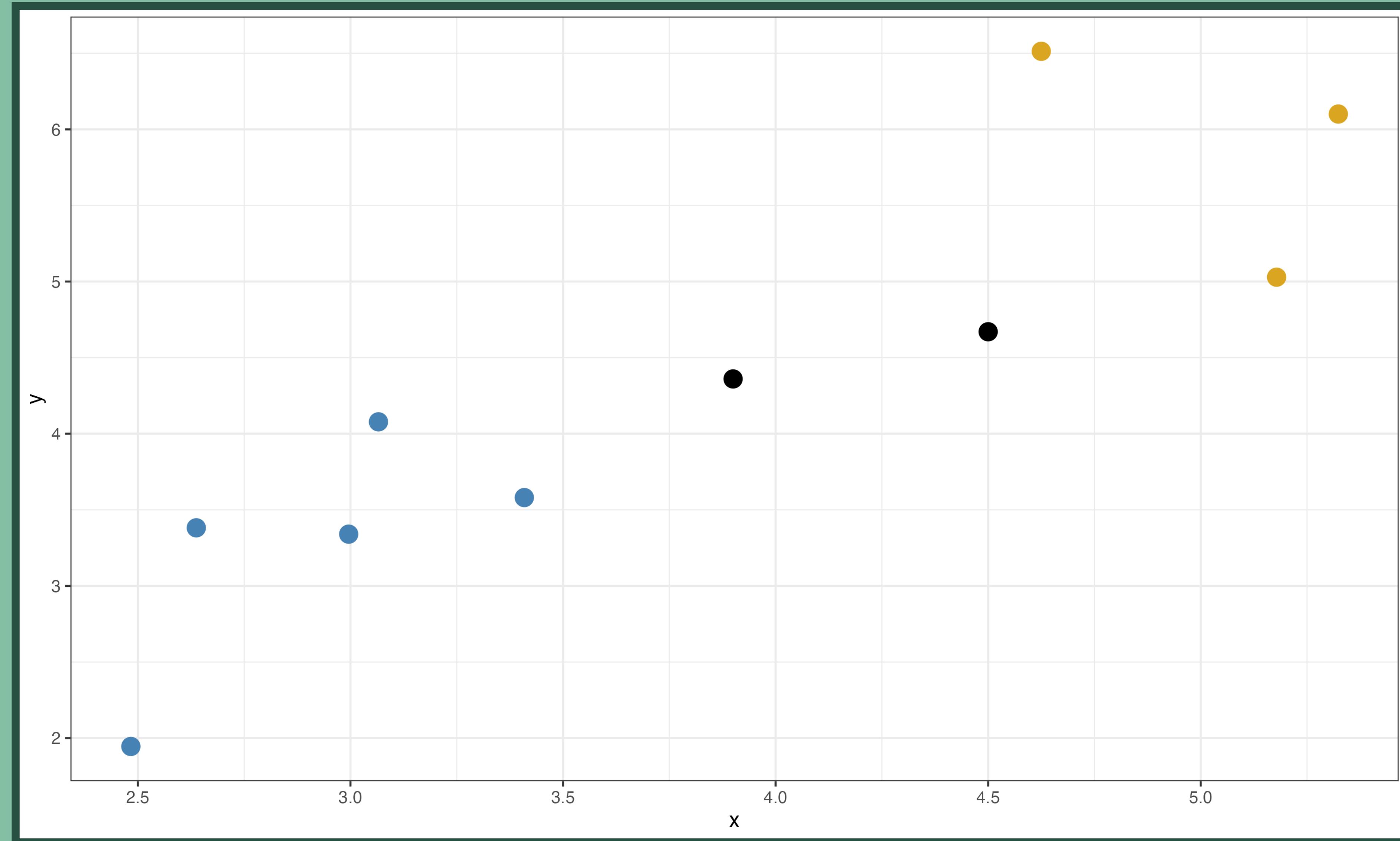
$k\text{NN}$

A non-parametric algorithm for finding the k “closest” observations to a given observation.

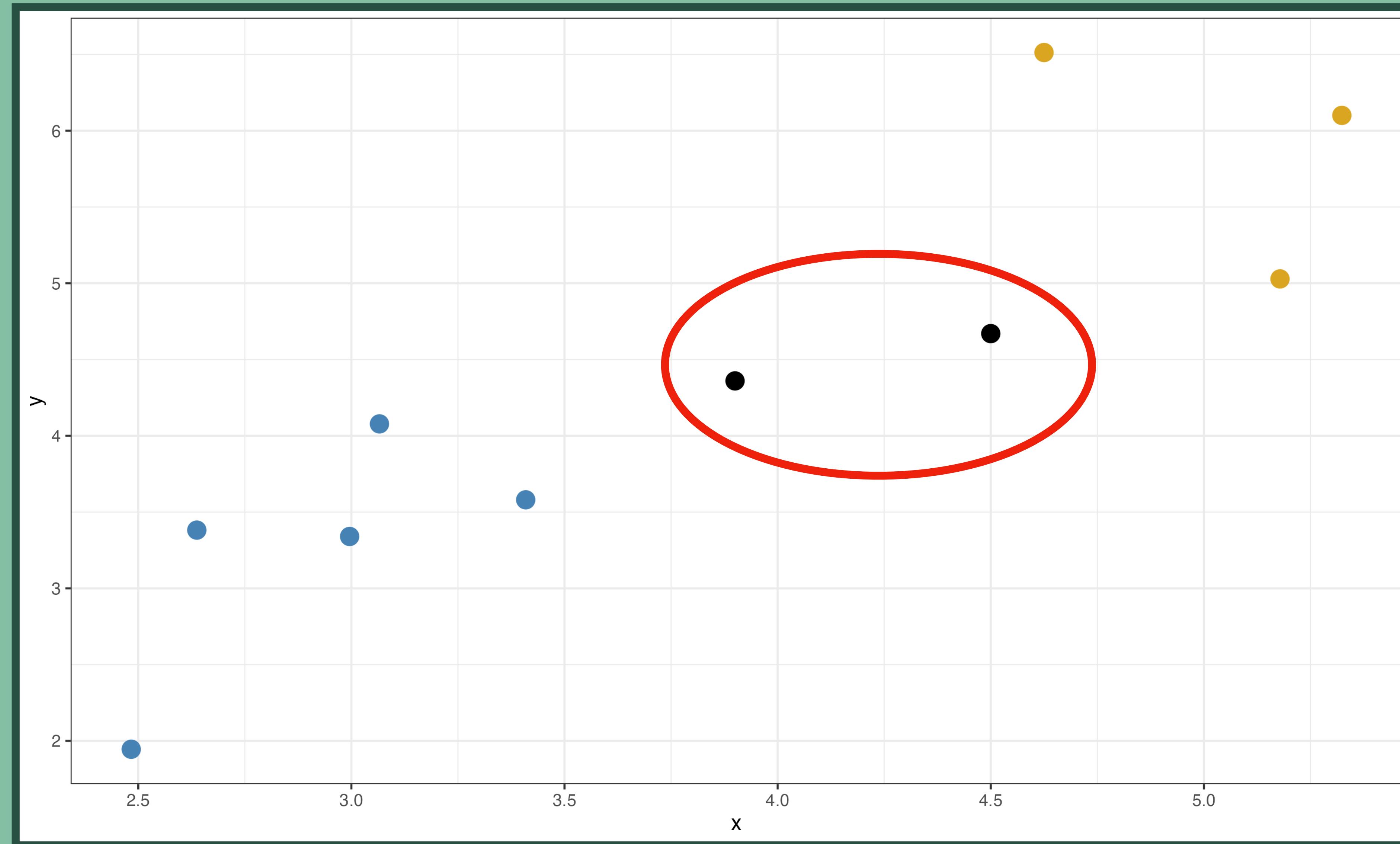
Hot deck imputation

A method for filling in missing data, where each new value is picked at random from a set of possible new values.

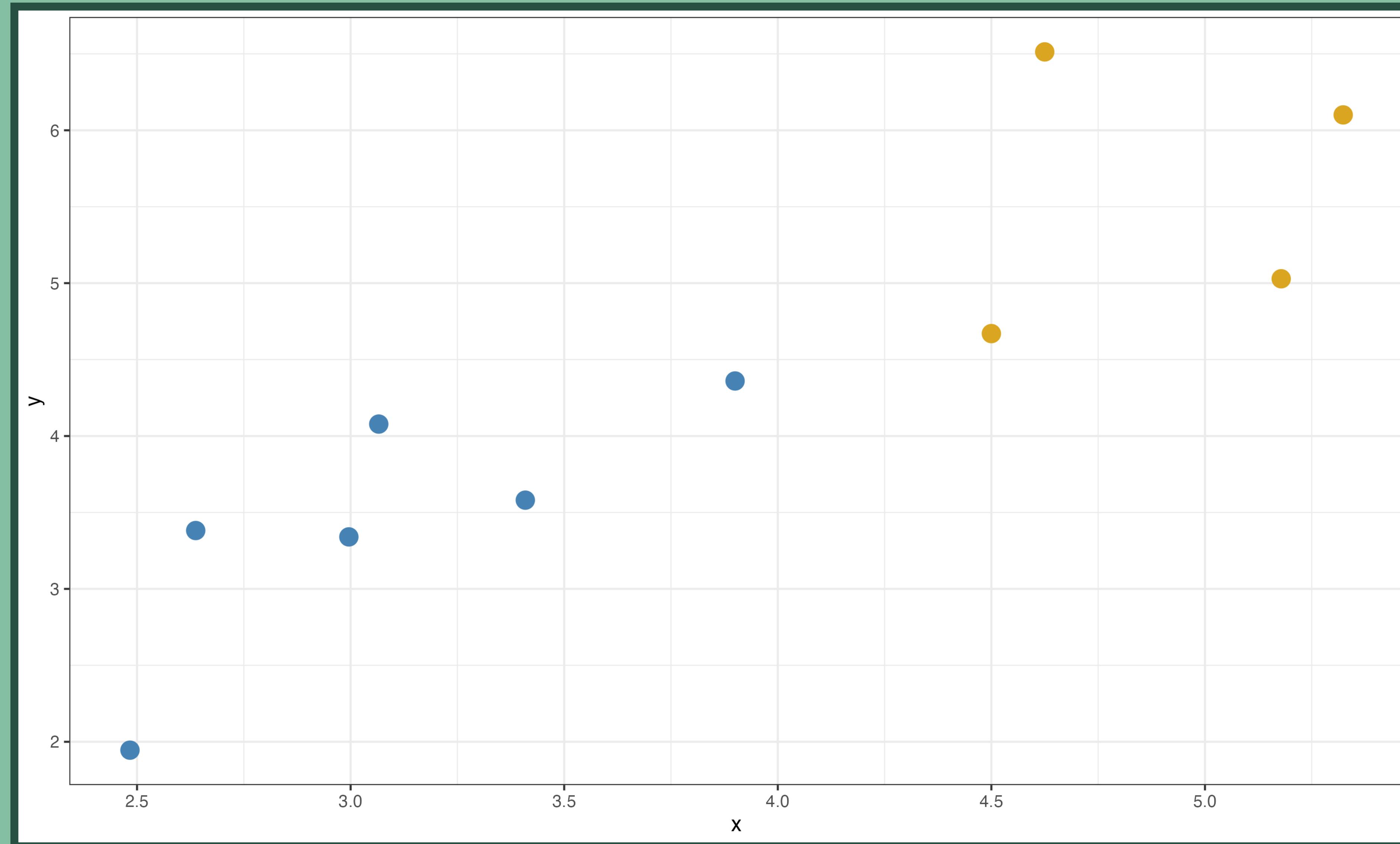
A brief overview of k NN



A brief overview of k NN



A brief overview of k NN



Simulation Design

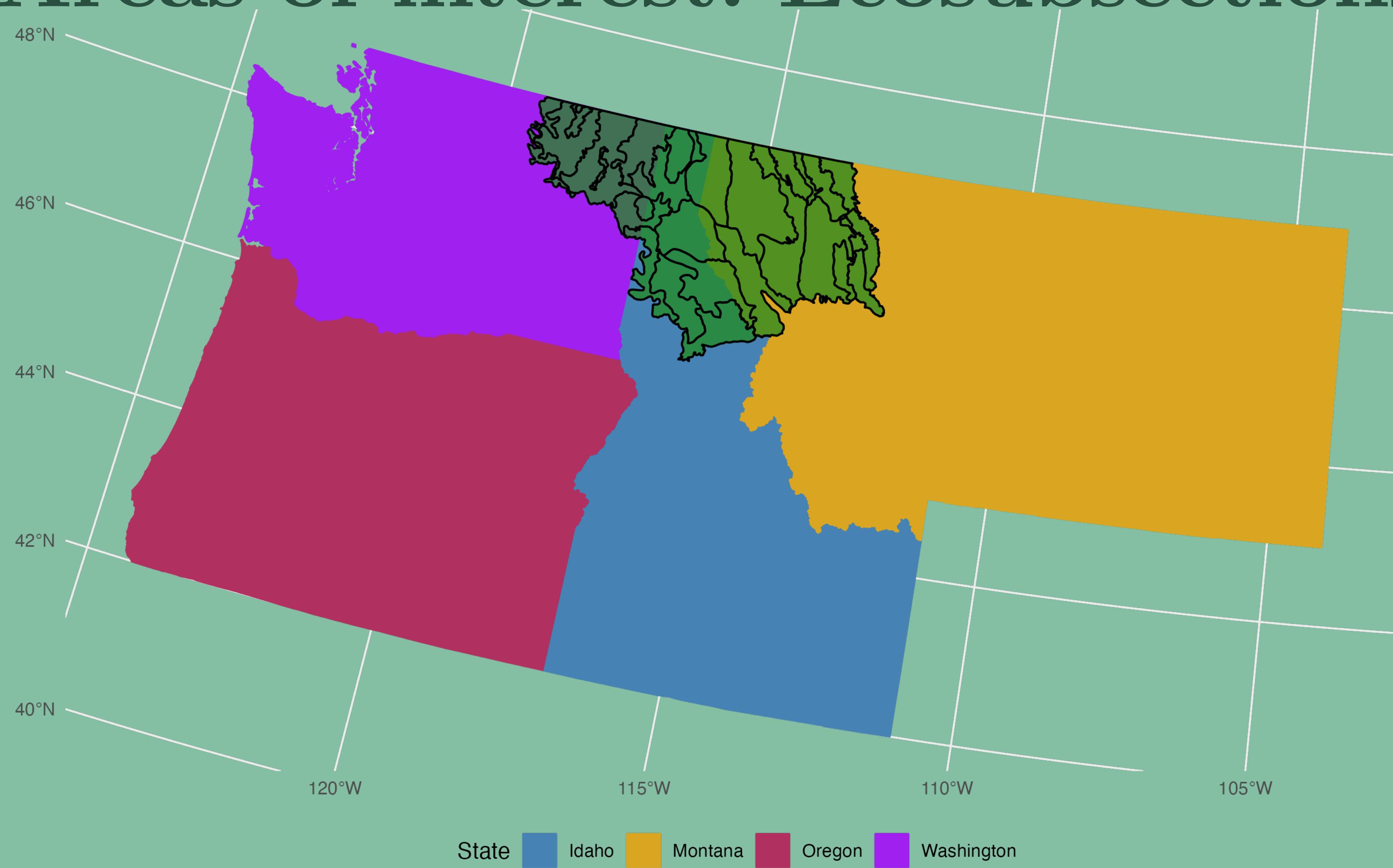


Orcas Island, Washington

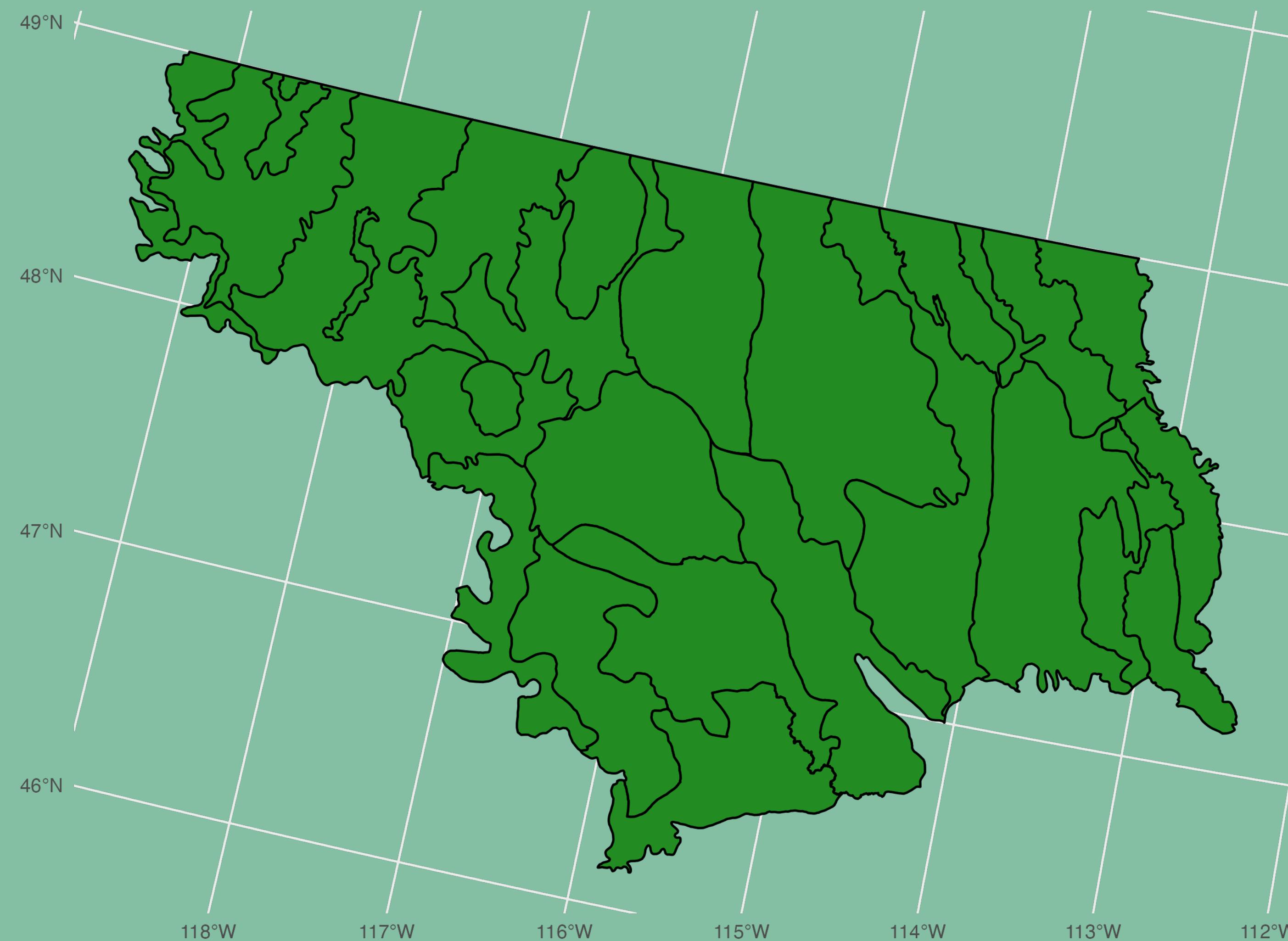


Lansing, Michigan

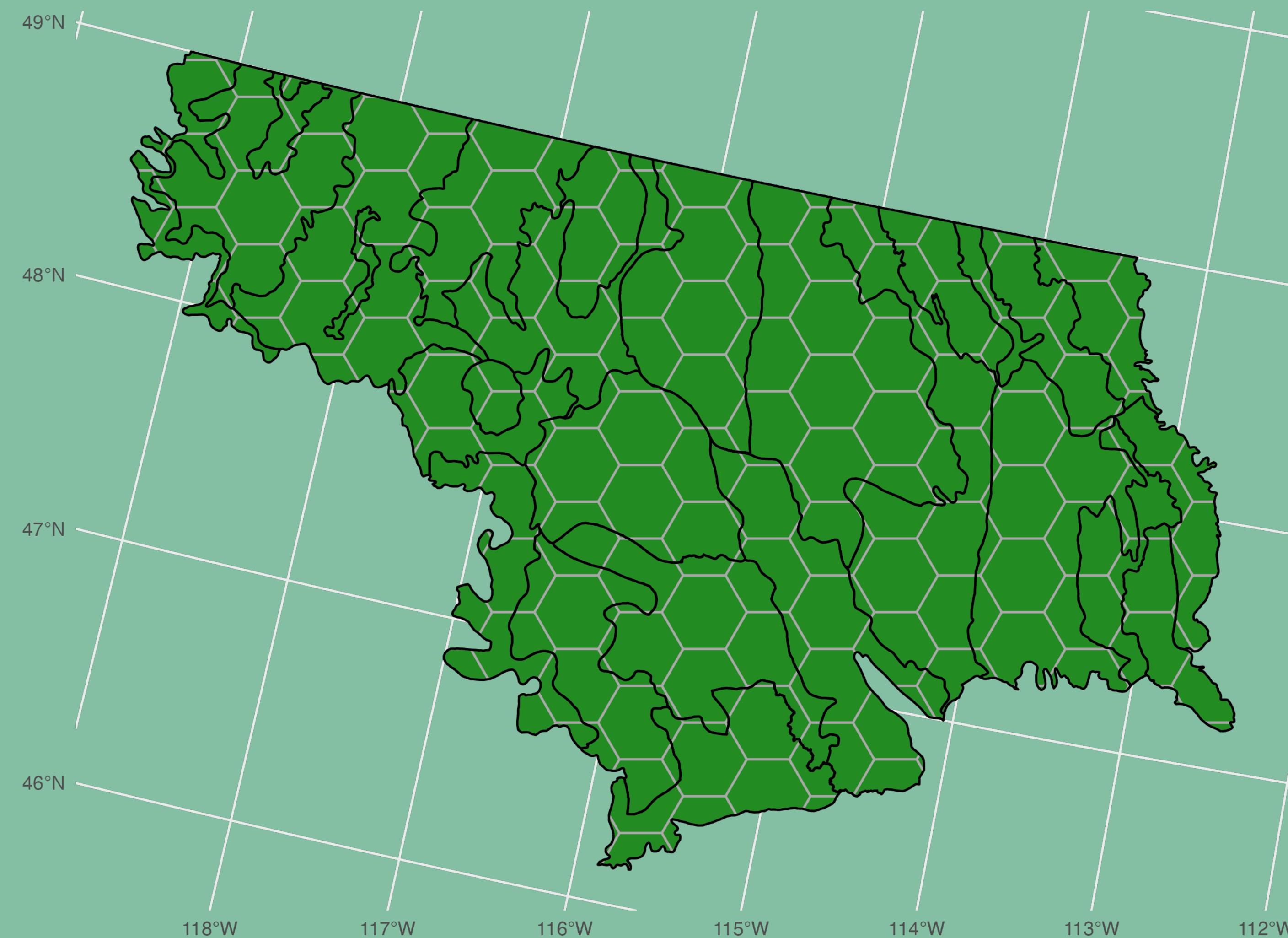
Study region: Ecoprovince M333, Areas of interest: Ecosubsections



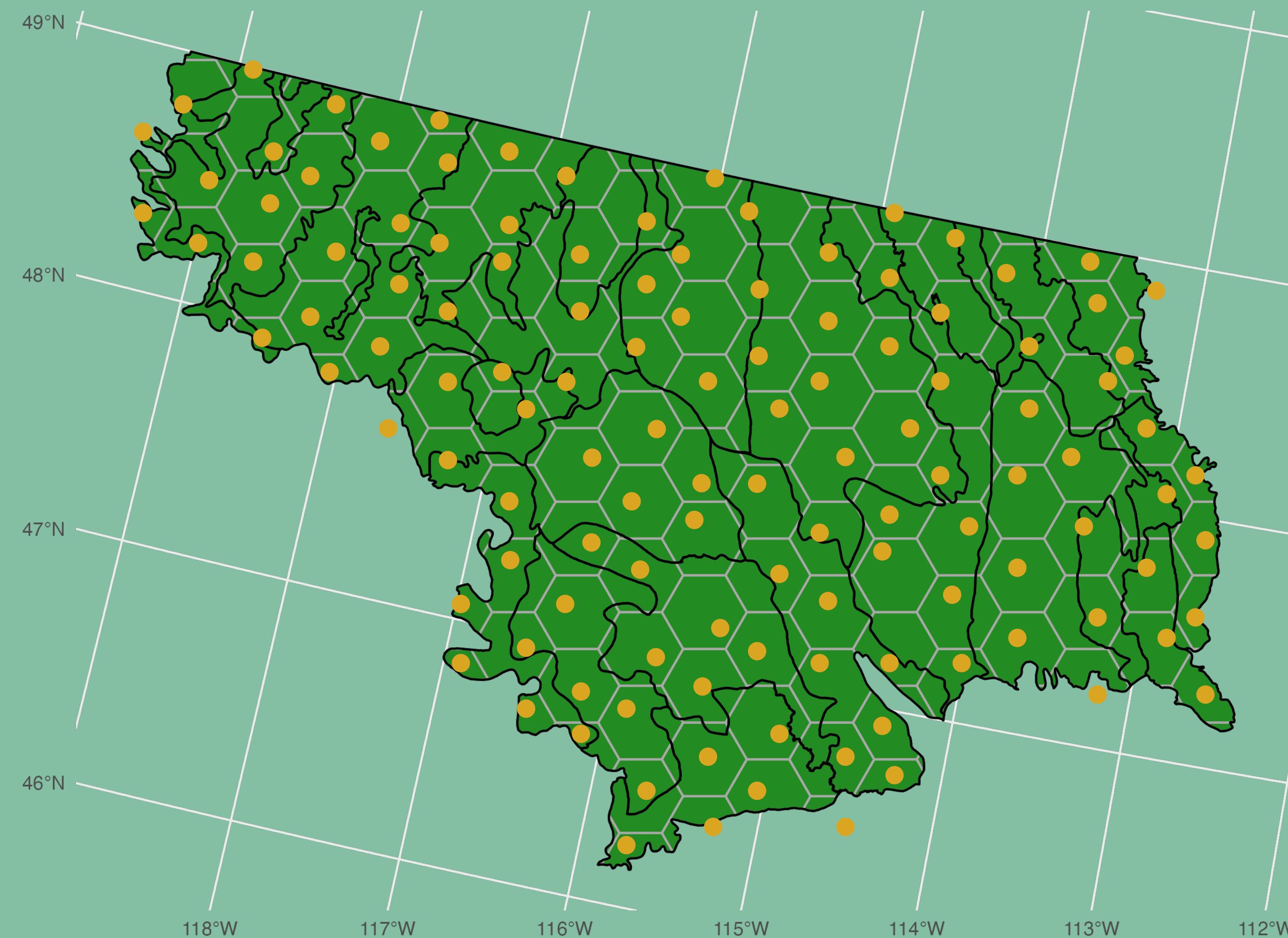
Study region: Ecoprovince M333, Areas of interest: Ecosubsections



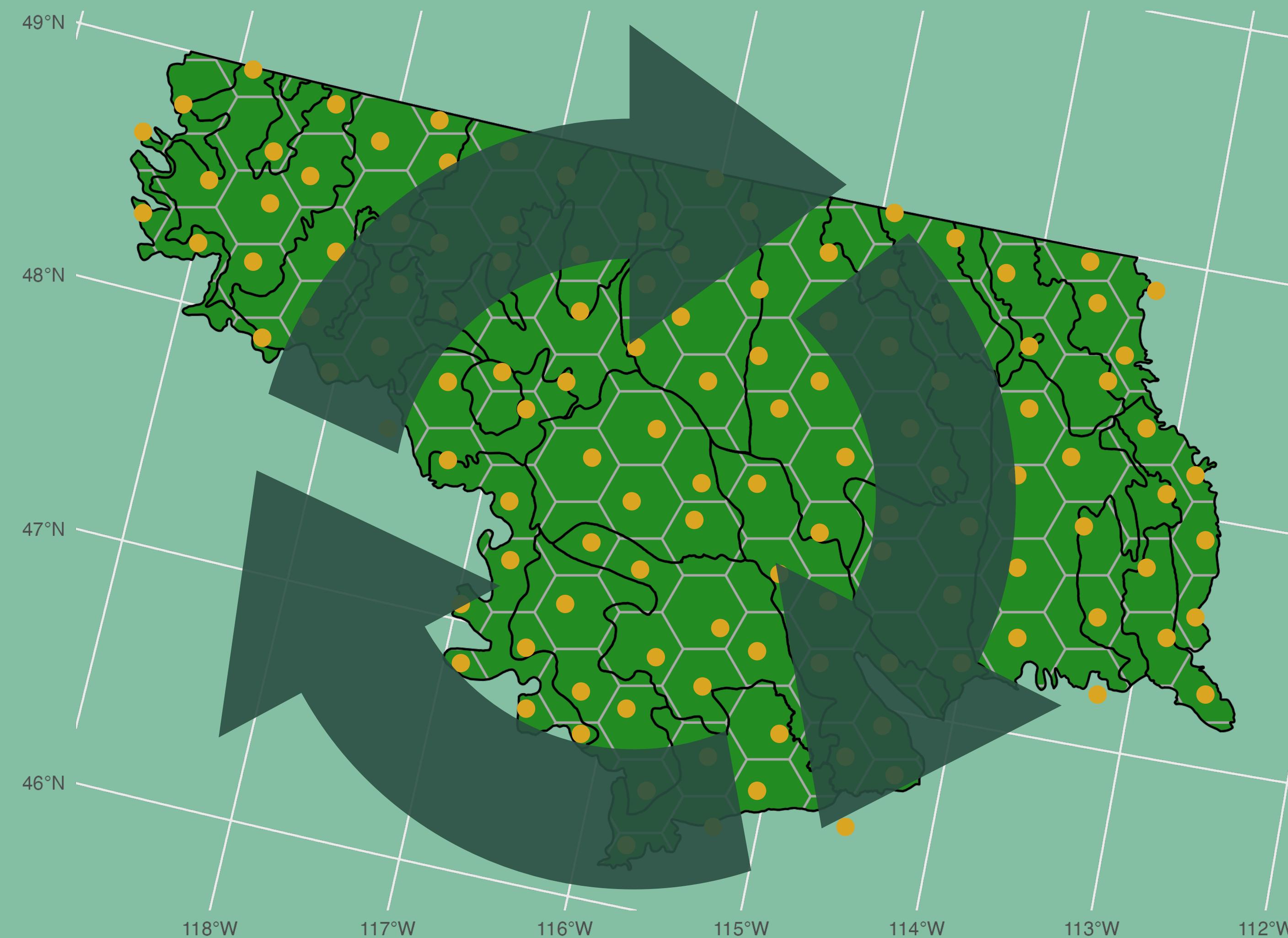
Creating Artificial Samples: Overlay FIA's Hexagonal Grid



Creating Artificial Samples: Randomly Select 90m Pixel Within Each Hex



Creating Artificial Samples: Repeat Pixel Selection Many Times



k NN Hot Deck Imputation: Step-by-step

Recipient row

tcc	tri	def	tnt
63	18.4	198	1

Step 1: select a recipient row of data

k NN Hot Deck Imputation: Step-by-step

Recipient row

tcc	tri	def	tnt
63	18.4	198	1

Potential donors

tcc	tri	def	tnt	BA
62	15.4	147	1	185
59	19.8	242	1	148
66	21.6	221	1	136

Step 1: select a recipient row of data

Step 2: find k nearest neighbors (“best matches”) in donor dataset

k NN Hot Deck Imputation: Step-by-step

Recipient row

tcc	tri	def	tnt
63	18.4	198	1

Potential donors

tcc	tri	def	tnt	BA
62	15.4	147	1	185
59	19.8	242	1	148
66	21.6	221	1	136

Step 1: select a recipient row of data

Step 2: find k nearest neighbors (“best matches”) in donor dataset

Step 3: choose one of the k best matches at random

k NN Hot Deck Imputation: Step-by-step

Recipient row

tcc	tri	def	tnt	BA
63	18.4	198	1	148

Potential donors

tcc	tri	def	tnt	BA
62	15.4	147	1	185
59	19.8	242	1	148
66	21.6	221	1	136

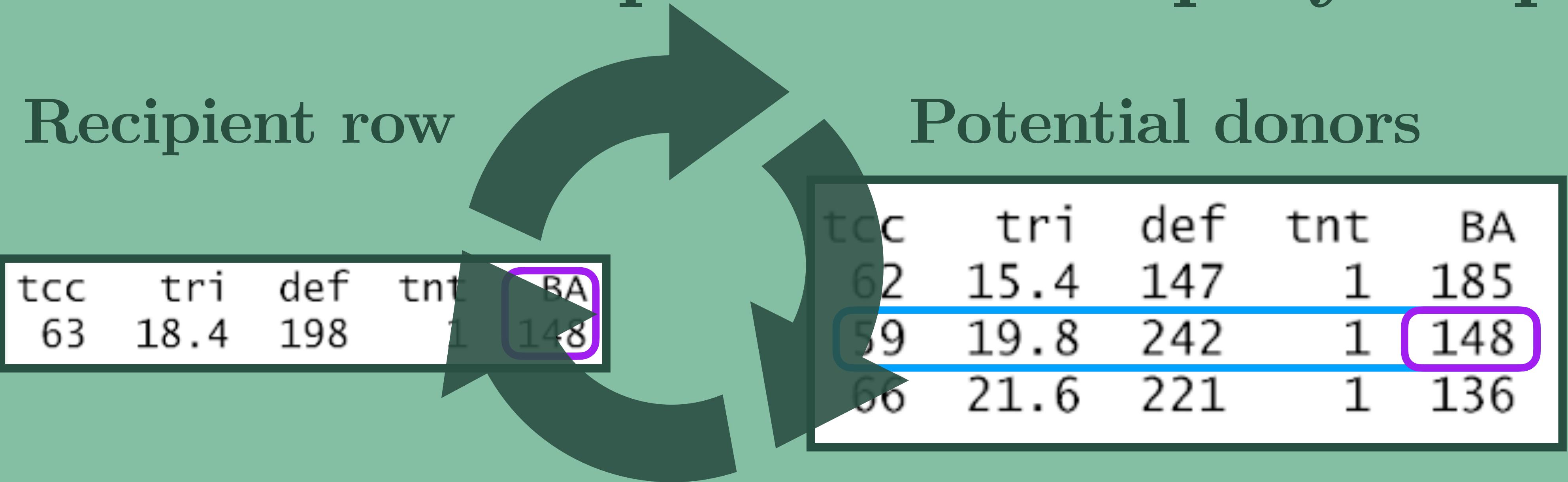
Step 1: select a recipient row of data

Step 2: find k nearest neighbors (“best matches”) in donor dataset

Step 3: choose one of the k best matches at random

Step 4: Impute the donor’s response value to the recipient row

k NN Hot Deck Imputation: Step-by-step



Step 1: select a recipient row of data

Step 2: find k nearest neighbors (“best matches”) in donor dataset

Step 3: choose one of the k best matches at random

Step 4: Impute the donor’s response value to the recipient row

Step 5: Repeat many times to generate simulation datasets

Why not a simpler alternative to k NN Hot Deck Imputation?

How else could we have generated our response variable values?

- (1) Through fitting a regression model, or
- (2) By treating an auxiliary variable as the response

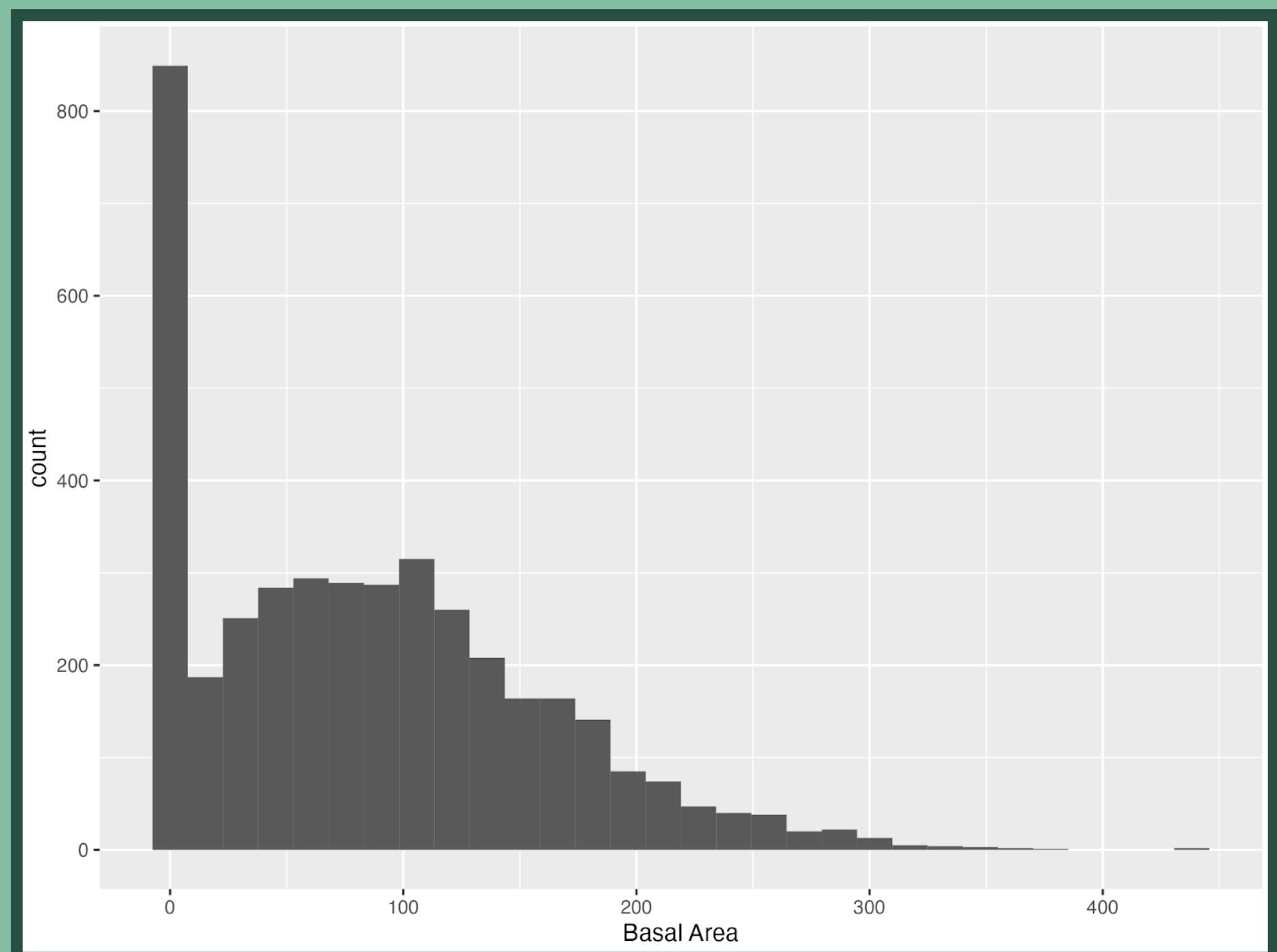
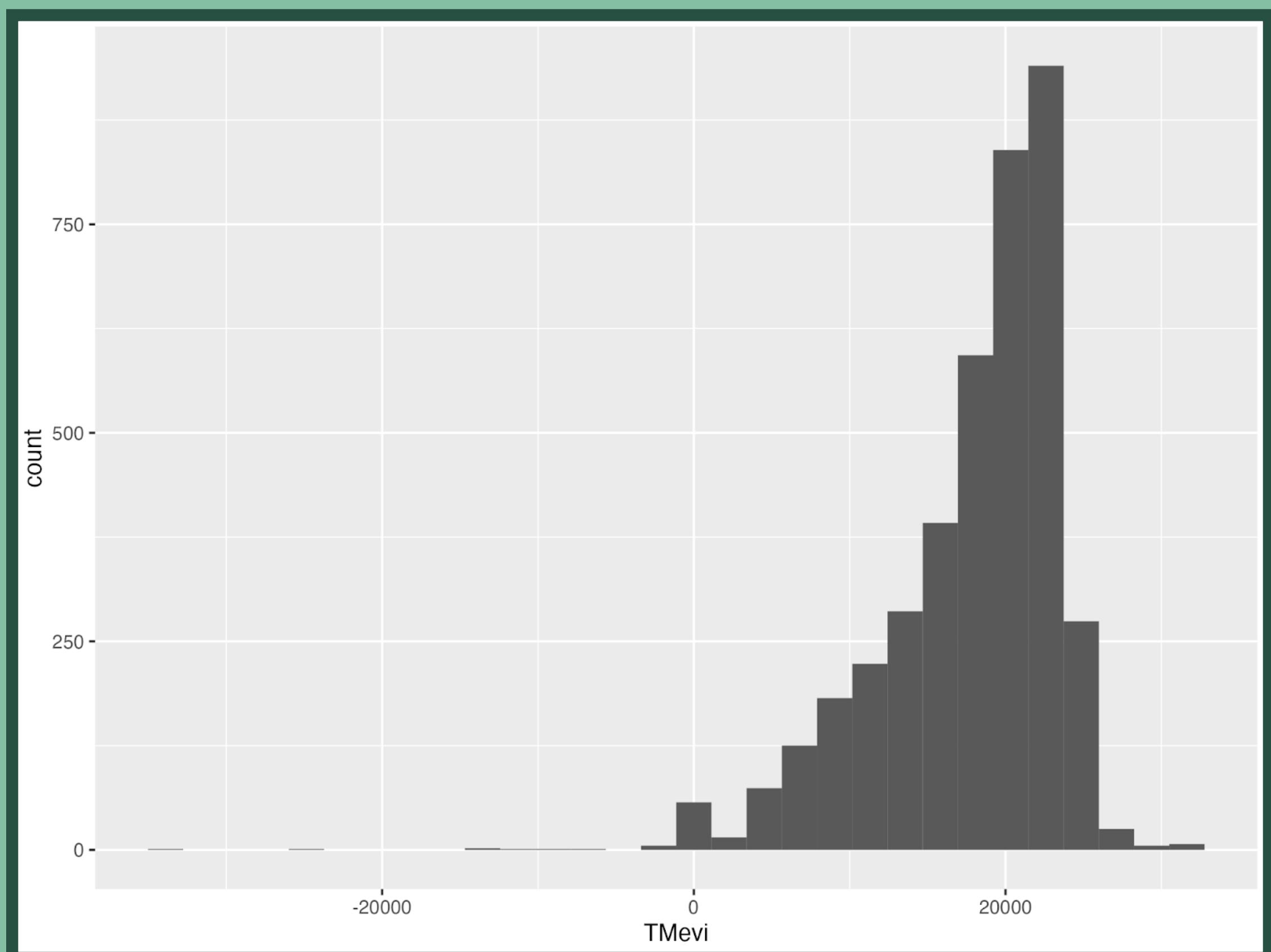
Both (1) and (2) have issues!

(1) Regression as Response:

Issue: If we fit a regression model and generated response values from it, our simulation study would be over-optimistic when evaluating estimators built on similar regression models.

(2) Auxiliary as Response

Issue: Our auxiliary variables are not “shaped” like our response variables.



When estimating the mean of some forest attribute of interest, we have so many choices!

$$\hat{\mu} = \frac{1}{n} \sum_{i \in s} y_i$$

How do we choose?

$$\hat{\mu} = \frac{1}{N} \sum_{i \in U} (y_i - \hat{y}_i)$$

Model form?

$$\hat{\mu} = X' \beta + \nu + \epsilon$$

Priors?

Fitting method?

Thank you!

Get in touch:

Email me: whitegra@msu.edu

Website: www.graysonwhite.com

Slides/other work available: www.github.com/graysonwhite