Spatial interpolation and kriging

EDS 222

Tamma Carleton

Fall 2021

Announcements/check-in

- Final projects guidelines

Announcements/check-in

- Final projects guidelines
- **Change in office hours** for week of 11/15 to Thursday 11/18 1:30pm-2:30pm

Announcements/check-in

- Final projects guidelines
- **Change in office hours** for week of 11/15 to Thursday 11/18 1:30pm-2:30pm
- **No class** 11/11; **remote class** 11/23, **no class** 11/25

Announcements/check-in

- Final projects guidelines
- **Change in office hours** for week of 11/15 to Thursday 11/18 1:30pm-2:30pm
- **No class** 11/11; **remote class** 11/23, **no class** 11/25
- Final project presentations: 12/2 9:30-10:45am (Bren Hall 1414); 12/7 8-10:30am (Bren Hall 14**2**4)
    - You will *randomly* be assigned a slot (slots announced 11/24)

Today

Refresher: types of spatial data

Points, vector, raster/field, dynamic raster/field

Today

Refresher: types of spatial data

Points, vector, raster/field, dynamic raster/field

A common challenge: spatial interpolation

Points to fields, interpolation

Today

Refresher: types of spatial data

Points, vector, raster/field, dynamic raster/field

A common challenge: spatial interpolation

Points to fields, interpolation

Kriging: a powerful form of interpolation

Variogram, kriging

Types of spatial data

Spatial data

Spatial Data can generally split into:

- **Vector** Data

Spatial data

Spatial Data can generally split into:

- **Vector** Data: points, lines, and polygons.

Spatial data

Spatial Data can generally split into:

- **Vector** Data: points, lines, and polygons.
- **Raster** Data

Spatial data

Spatial Data can generally split into:

- **Vector** Data: points, lines, and polygons.
- **Raster** Data: a grid of equally sized rectangles.

Spatial data

Spatial Data can generally split into:

- **Vector** Data: points, lines, and polygons.
- **Raster** Data: a grid of equally sized rectangles.

An **alternative framing**: *object view* versus *field view*

Spatial data

Spatial Data can generally split into:

- **Vector** Data: points, lines, and polygons.
- **Raster** Data: a grid of equally sized rectangles.

An **alternative framing**: *object view* versus *field view*

- **Object View**: The study region (and world) is a series of entities located in space.

Spatial data

Spatial Data can generally split into:

- **Vector** Data: points, lines, and polygons.
- **Raster** Data: a grid of equally sized rectangles.

An **alternative framing**: *object view* versus *field view*

- **Object View**: The study region (and world) is a series of entities located in space.

Examples

Spatial data

Spatial Data can generally split into:

- **Vector** Data: points, lines, and polygons.
- **Raster** Data: a grid of equally sized rectangles.

An **alternative framing**: *object view* versus *field view*

- **Object View**: The study region (and world) is a series of entities located in space.

Examples : Points representing cities. Non-continuous polygons representing cities.

Spatial data

Spatial Data can generally split into:

- **Vector** Data: points, lines, and polygons.
- **Raster** Data: a grid of equally sized rectangles.

An **alternative framing**: *object view* versus *field view*

- **Object View**: The study region (and world) is a series of entities located in space.

Examples : Points representing cities. Non-continuous polygons representing cities.

- **Field View**: Every location within the study region (and world) has a measurable value.

Spatial data

Spatial Data can generally split into:

- **Vector** Data: points, lines, and polygons.
- **Raster** Data: a grid of equally sized rectangles.

An **alternative framing**: *object view* versus *field view*

- **Object View**: The study region (and world) is a series of entities located in space.

Examples : Points representing cities. Non-continuous polygons representing cities.

- **Field View**: Every location within the study region (and world) has a measurable value.

Examples

Spatial data

Spatial Data can generally split into:

- **Vector** Data: points, lines, and polygons.
- **Raster** Data: a grid of equally sized rectangles.

An **alternative framing**: *object view* versus *field view*

- **Object View**: The study region (and world) is a series of entities located in space.

Examples : Points representing cities. Non-continuous polygons representing cities.

- **Field View**: Every location within the study region (and world) has a measurable value.

Examples : Elevation. Temperature. Wind direction.

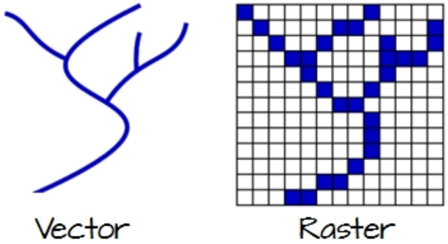
Spatial data

Q: Is there a *best* data type to represent objects or fields?

Spatial data

Q: Is there a *best* data type to represent objects or fields?

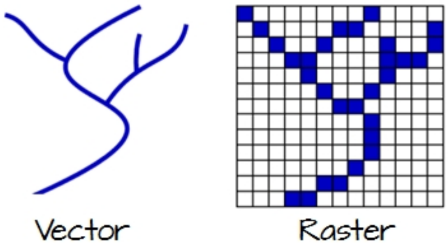
A: Usually, but it depends.



Spatial data

Q: Is there a *best* data type to represent objects or fields?

A: Usually, but it depends.

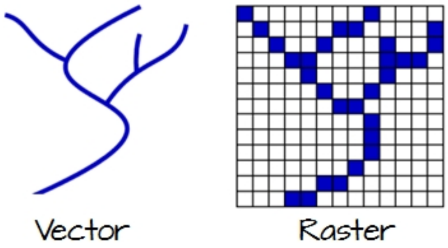


- Usually it will be easier to represent **objects** with **vector data** and **fields** with **raster** data, but ultimately this depends on what analysis you want to run

Spatial data

Q: Is there a *best* data type to represent objects or fields?

A: Usually, but it depends.



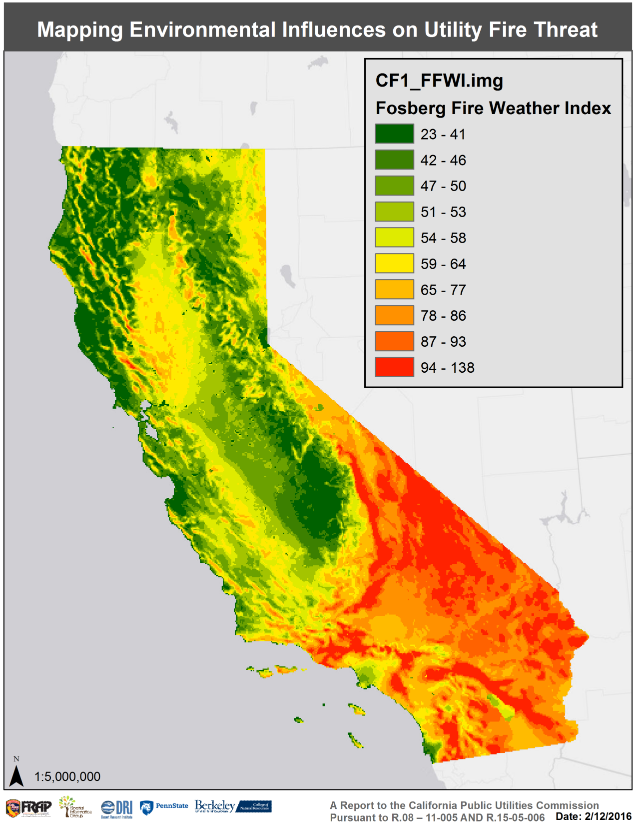
- Usually it will be easier to represent **objects** with **vector data** and **fields** with **raster** data, but ultimately this depends on what analysis you want to run

- Luckily, `R` makes it easy to switch back and forth (but we need to be careful and intentional when transforming!)

Spatial interpolation

Spatial interpolation

In environmental data science, we are **often interested in modeling fields**



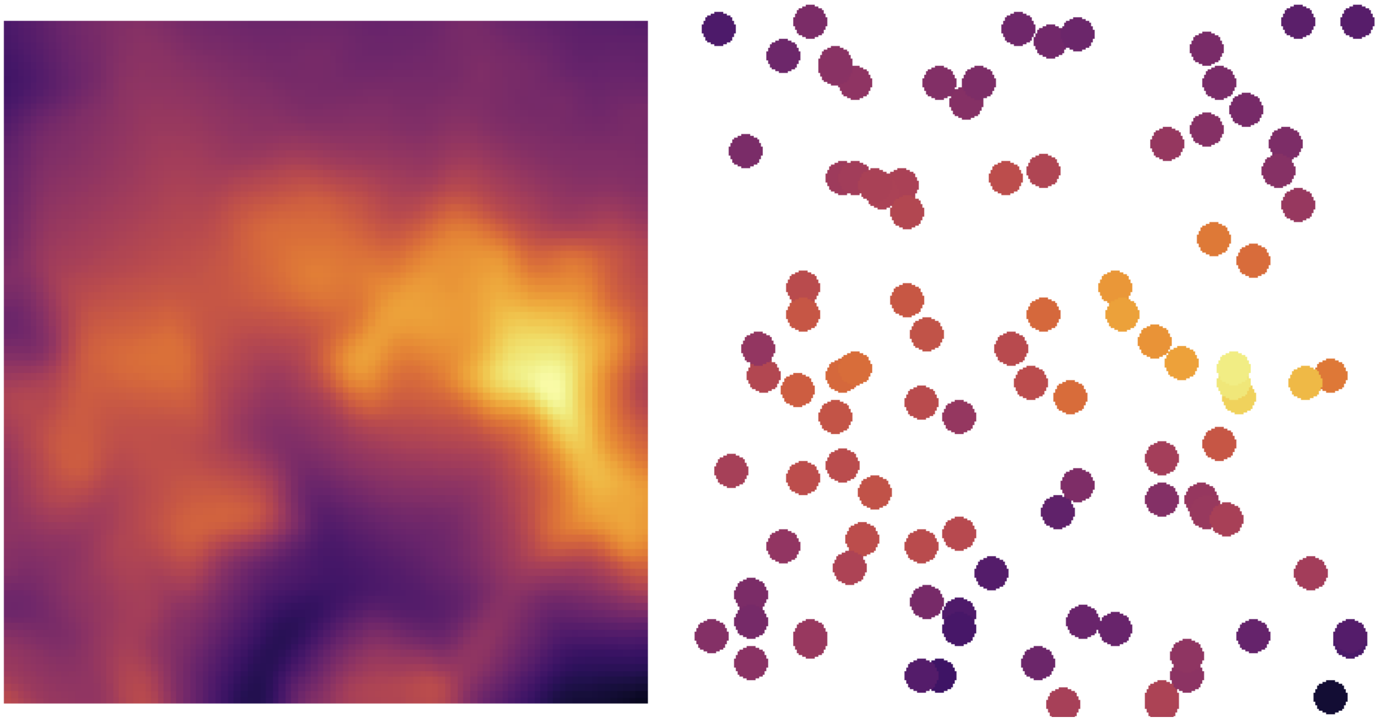
Spatial interpolation

But we are doing **statistics!**

Spatial interpolation

But we are doing **statistics!**

That means we only have data from a *sample*, not a census of the *population*



Spatial interpolation

- Samples taken from a continuous spatial field often raise the need for **spatial interpolation**

Spatial interpolation

- Samples taken from a continuous spatial field often raise the need for **spatial interpolation**

Definition:

Spatial interpolation is the process of using a **sample** of observed points to estimate values for **all locations** in a study region

Spatial interpolation

- Samples taken from a continuous spatial field often raise the need for **spatial interpolation**

Definition:

Spatial interpolation is the process of using a **sample** of observed points to estimate values for **all locations** in a study region

For example:

- Predicting "gold grades" across South Africa using a few borehole samples (the problem of Daniel *Krige*!)
- Predicting depth to groundwater across California using monitoring wells
- Predicting air pollution across China using monitoring stations

Spatial interpolation in math

- Let $Z(x_0)$ indicate the value (e.g., elevation) at a location x_0 that was *not* sampled

Spatial interpolation in math

- Let $Z(x_0)$ indicate the value (e.g., elevation) at a location x_0 that was *not* sampled

- Let $Z(x_i)$ for $i = 1, \ldots m$ indicate the values for locations $i = 1, \ldots, m$ that *were* sampled

Spatial interpolation in math

- Let $Z(x_0)$ indicate the value (e.g., elevation) at a location x_0 that was *not* sampled
- Let $Z(x_i)$ for $i = 1, \ldots m$ indicate the values for locations $i = 1, \ldots, m$ that *were* sampled

Spatial interpolation aims to predict $Z(x_0)$ using a linear combination of the values in the sampled locations:

$$\hat{Z}(x_0) = \sum_{i=1}^{m} \lambda_i Z(x_i)$$

where λ_i are weights applied to each sampled location.

Spatial interpolation in math

- Let $Z(x_0)$ indicate the value (e.g., elevation) at a location x_0 that was *not* sampled

- Let $Z(x_i)$ for $i = 1, \ldots m$ indicate the values for locations $i = 1, \ldots, m$ that *were* sampled

Spatial interpolation aims to predict $Z(x_0)$ using a linear combination of the values in the sampled locations:

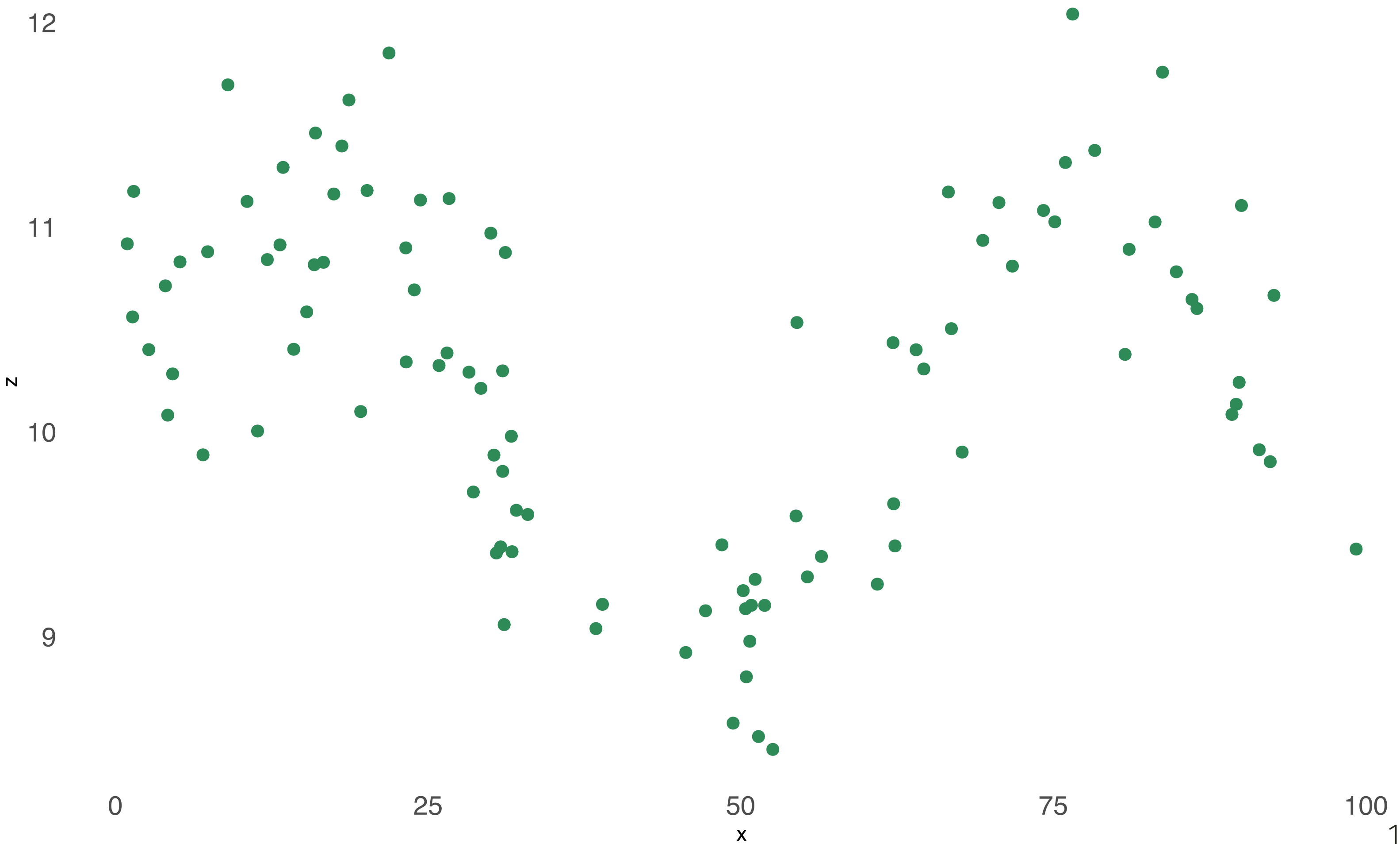
$$\hat{Z}(x_0) = \sum_{i=1}^{m} \lambda_i Z(x_i)$$

where λ_i are weights applied to each sampled location.

- All spatial interpolation methods assume or derive a set of λ's to compute $\hat{Z}$'s

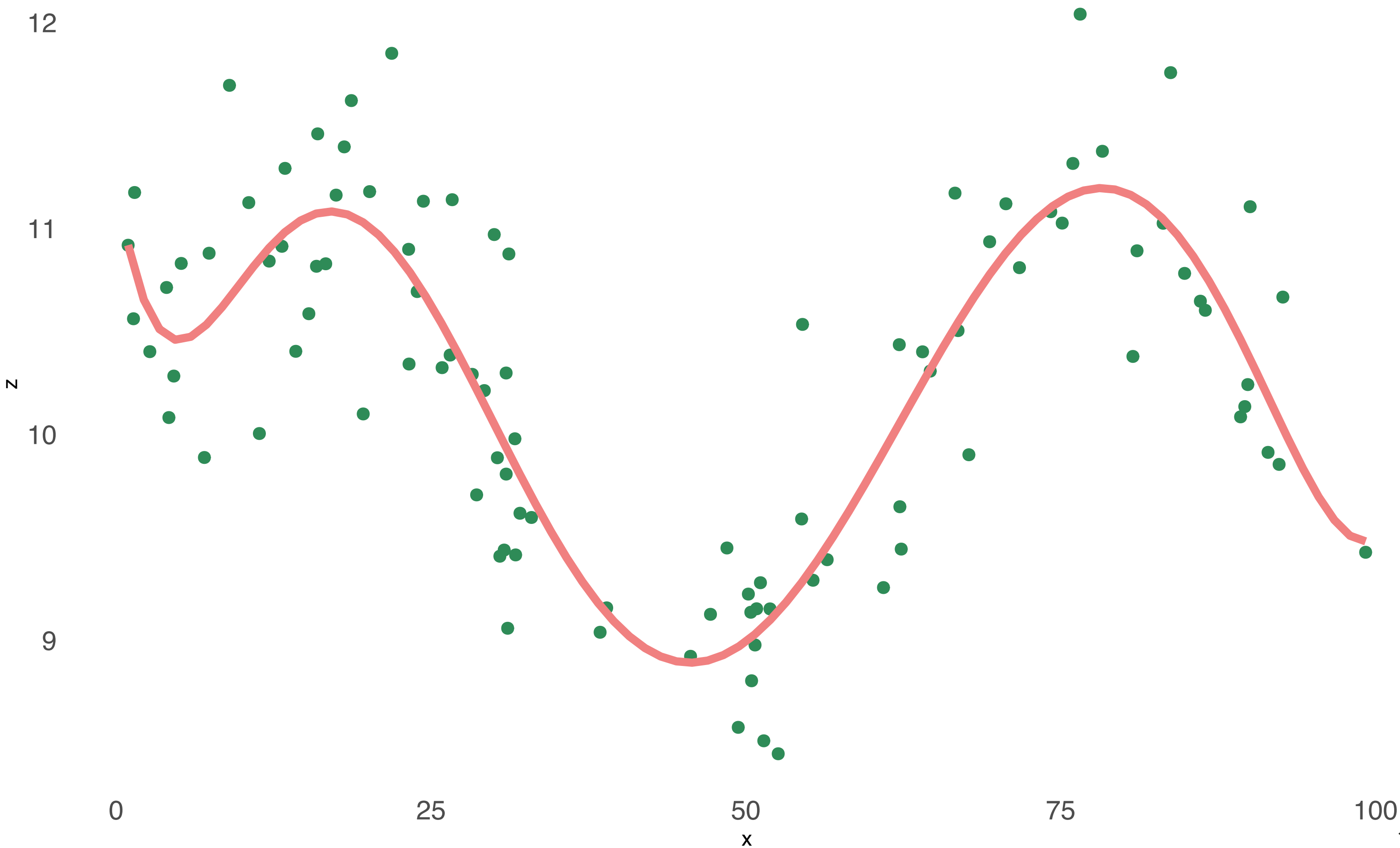
Interpolation in one dimension

Consider one-dimensional space where values y depend on location x



Interpolation in one dimension

Consider one-dimensional space where values z depend on location x



Interpolation in two dimensions

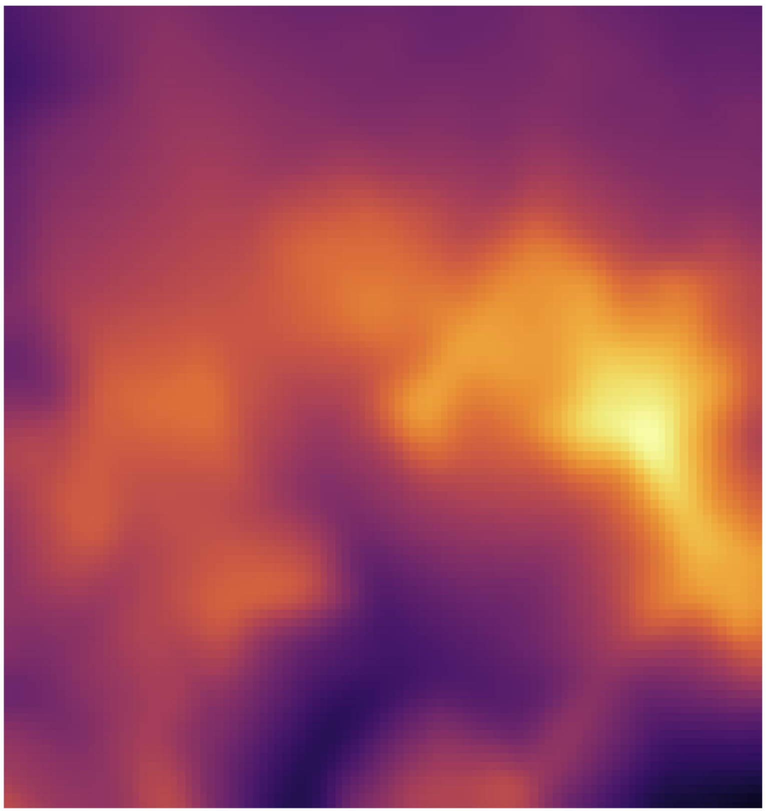
Often we have data for an outcome z observed in 2-D space: $z(x, y)$

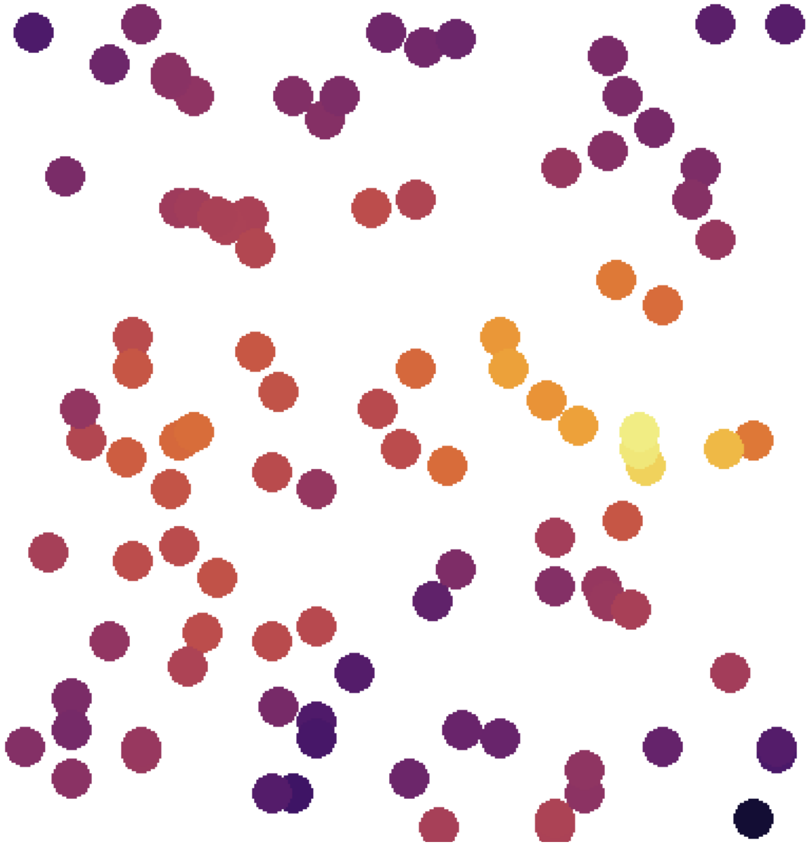
[Draw it!]

Interpolation in two dimensions

Often we have data for an outcome z observed in 2-D space: $z(x, y)$

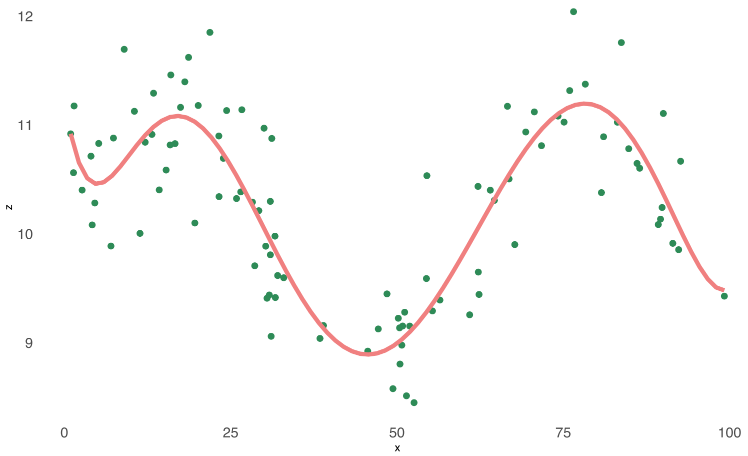
[Draw it!]





Interpolation methods

Polynomial regression



- In one-dimensional space:

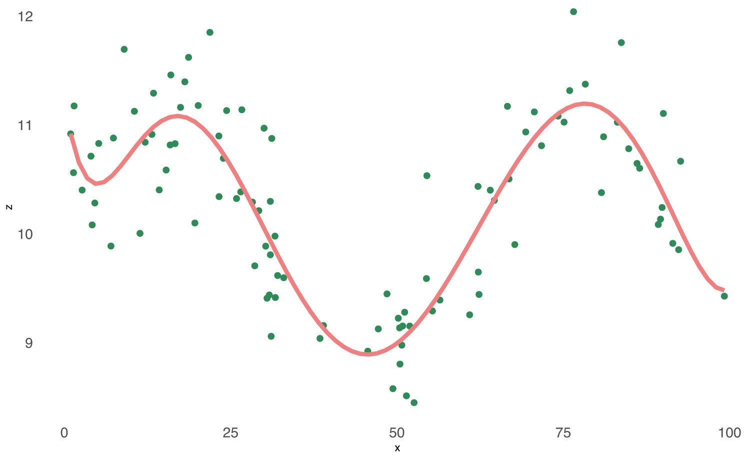
$$\hat{Z}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0 + \hat{\beta}_2 x_0^2 + \ldots + \hat{\beta}^p x_0^p$$

- In two-dimensional space with (x_0, y_0) the unknown value:

$$\hat{Z}(x_0, y_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0 + \hat{\beta}_2 y_0 + \hat{\beta}_3 x_0 y_0$$

Interpolation methods

Polynomial regression



- In one-dimensional space:

$$\hat{Z}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0 + \hat{\beta}_2 x_0^2 + \ldots + \hat{\beta}^p x_0^p$$

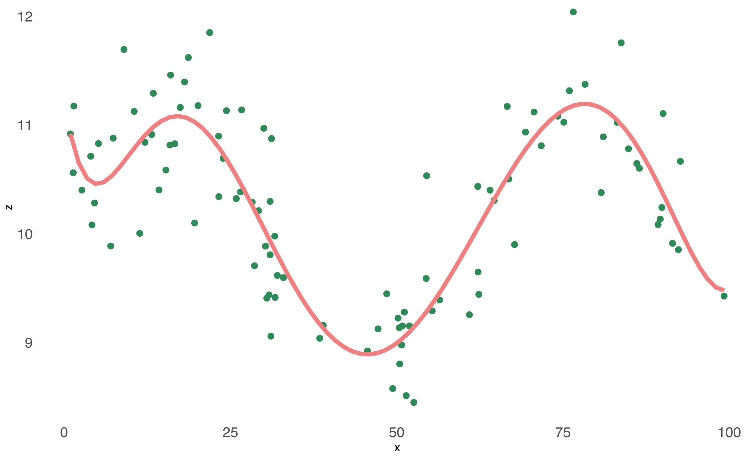
- In two-dimensional space with (x_0, y_0) the unknown value:

$$\hat{Z}(x_0, y_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0 + \hat{\beta}_2 y_0 + \hat{\beta}_3 x_0 y_0$$

- **Pros:** Easy, analytical expression, continuous & differentiable surface
- **Cons:** Errors can be large, *inexact*

Interpolation methods

Polynomial regression



- In one-dimensional space:

$$\hat{Z}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0 + \hat{\beta}_2 x_0^2 + \ldots + \hat{\beta}^p x_0^p$$

- In two-dimensional space with (x_0, y_0) the unknown value:

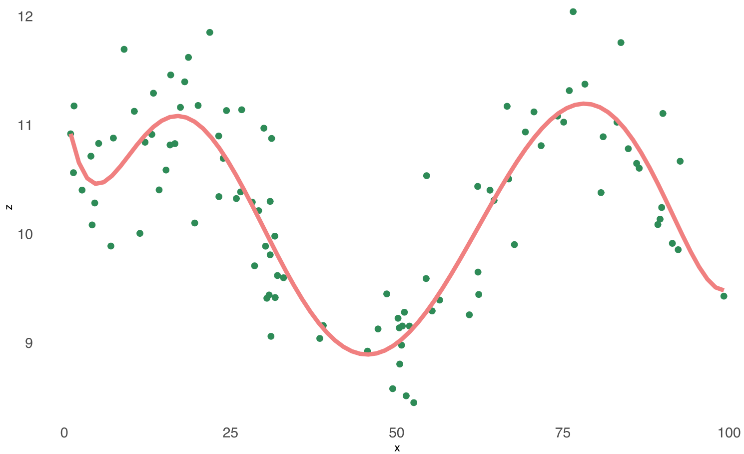
$$\hat{Z}(x_0, y_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0 + \hat{\beta}_2 y_0 + \hat{\beta}_3 x_0 y_0$$

- **Pros:** Easy, analytical expression, continuous & differentiable surface
- **Cons:** Errors can be large, *inexact*

Exact: Predicts a value identical to the measured value.

Interpolation methods

Polynomial regression



- In one-dimensional space:

$$\hat{Z}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0 + \hat{\beta}_2 x_0^2 + \ldots + \hat{\beta}^p x_0^p$$

- In two-dimensional space with (x_0, y_0) the unknown value:

$$\hat{Z}(x_0, y_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0 + \hat{\beta}_2 y_0 + \hat{\beta}_3 x_0 y_0$$

- **Pros:** Easy, analytical expression, continuous & differentiable surface
- **Cons:** Errors can be large, *inexact*

Exact: Predicts a value identical to the measured value.

Inexact: Does *not* predict a value identical to the measured value.

Polynomial regression interpolation

This is just **multiple linear regression** using spatial information as the independent variable

```
mod = lm(z~poly(x,8))
predictions = augment(mod)$.fitted
```

Interpolation methods

Nearest Neighbors (NN)

Interpolation methods

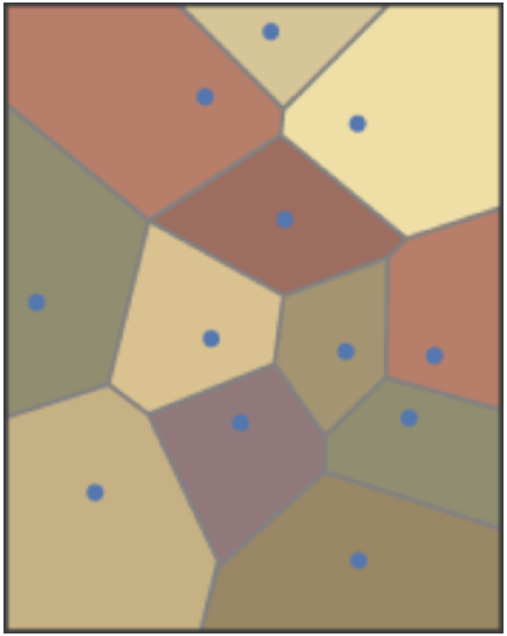
Nearest Neighbors (NN)

- Simple: Assign value of nearest observation in space

Interpolation methods

Nearest Neighbors (NN)

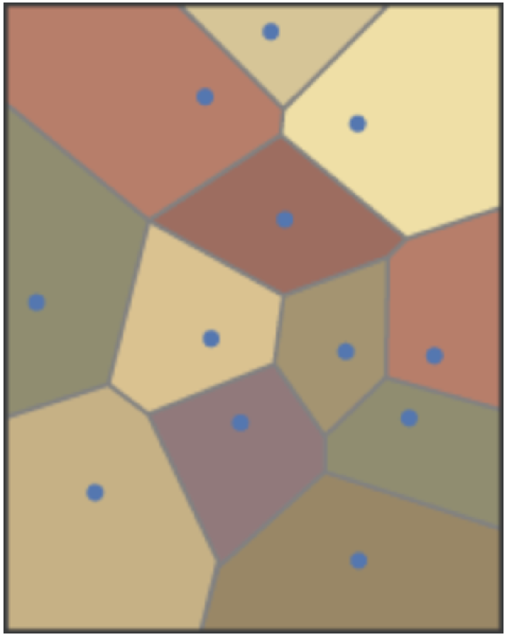
- Simple: Assign value of nearest observation in space



Interpolation methods

Nearest Neighbors (NN)

- Simple: Assign value of nearest observation in space



- Creates what are called "Theissen Polygons", which allocate space to the nearest sampled point

Nearest Neighbor interpolation

Q: What would the weight vector λ look like for NN interpolation?

Nearest Neighbor interpolation

Q: What would the weight vector λ look like for NN interpolation?

Q: What type of function does NN interpolation produce for 1-D space?
[draw it!]

Nearest Neighbor interpolation

Q: What would the weight vector λ look like for NN interpolation?

Q: What type of function does NN interpolation produce for 1-D space? [draw it!]

- **Pros:** Easy, intuitive, field may actually be discontinuous, exact
- **Cons:** Discontinuous, error-prone if field is smooth

Nearest Neighbor interpolation

Q: What would the weight vector λ look like for NN interpolation?

Q: What type of function does NN interpolation produce for 1-D space? [draw it!]

- **Pros:** Easy, intuitive, field may actually be discontinuous, exact
- **Cons:** Discontinuous, error-prone if field is smooth

Implementation in `R`

- Easy with the `voroni()` function from the `dismo` package:

```
library(dismo)
v <- voronoi(dta)
plot(v)
```

Nearest Neighbor interpolation

Q: What would the weight vector λ look like for NN interpolation?

Q: What type of function does NN interpolation produce for 1-D space? [draw it!]

- **Pros:** Easy, intuitive, field may actually be discontinuous, exact
- **Cons:** Discontinuous, error-prone if field is smooth

Implementation in `R`

- Easy with the `voroni()` function from the `dismo` package:

```
library(dismo)
v ← voronoi(dta)
plot(v)
```

- Helpful tutorial here

Interpolation methods

Inverse distance weighting

Basic idea: weights are a decreasing function of distance from x_0 to x_i

$$\hat{Z}(x_0) = \sum_{i=1}^{m} \frac{Z(x_i) Dist(x_i, x_0)^{-p}}{\sum_{i=1}^{m} Dist(x_i, x_0)^{-p}}$$

Equivalently:

$$\lambda_i^{IDW} = \frac{1/Dist(x_i, x_0)^p}{\sum_{i=1}^{m} 1/Dist(x_i, x_0)^p}$$

where p is the "power parameter" determining how fast the weight declines as the distance between the points grows larger

Interpolation methods

Inverse distance weighting

- **Pros:** Smooth, exact
- **Cons:** Difficult/computationally intensive (you need to compute distances for *all* pairs of points in the region!), all sampled observations influence $\hat{Z}(x_0)$, have to choose p somehow

Interpolation methods

Inverse distance weighting

- **Pros:** Smooth, exact
- **Cons:** Difficult/computationally intensive (you need to compute distances for *all* pairs of points in the region!), all sampled observations influence $\hat{Z}(x_0)$, have to choose p somehow

Implementation in `R`

```
library(phylin)
idw(values, coords, grid, method = "Shepard", p = 2, R = 2, N = 15,
    distFUN = geo.dist, ... )
```

- Note the `method` argument: "Shepard" follows the math on the previous slide
- Note the `p` argument: Need to specify power parameter

Interpolation methods

There are many more!

- Piecewise linear interpolation / Delany triangulation
- Local polynomial regression
- Radial basis function (RBF)
- Kriging (of many forms)
- Many new machine-learning based methods
- Learn more in Li and Heap (2014)

Enter: Kriging

Kriging is the most widely used form of spatial interpolation in spatial statistics.

Enter: Kriging

Kriging is the most widely used form of spatial interpolation in spatial statistics.

Why?

- It is *flexible* (i.e., less researcher decisions, more data-driven)
- Under certain assumptions it is the "best linear unbiased estimate" (sound like OLS yet??)
- You can recover an estimate *and* a standard error (i.e., it is *stochastic*)

Enter: Kriging

Kriging is the most widely used form of spatial interpolation in spatial statistics.

Why?

- It is *flexible* (i.e., less researcher decisions, more data-driven)
- Under certain assumptions it is the "best linear unbiased estimate" (sound like OLS yet??)
- You can recover an estimate *and* a standard error (i.e., it is *stochastic*)

We will study **kriging** and its implementation in `R` in the next lecture and lab

Slides created via the R package **xaringan**.