Spatial interpolation and kriging

EDS 222

Tamma Carleton
Fall 2023

Announcements/check-in

Final Projects

- 3-minute in-class presentation: 12/12, 4-7pm, Bren Hall 1424 -- with snacks! I will randomly allocate slots and post the presentation order by 12/07
- Blog post/write up: due 12/9, 5pm, send to me and Sandy via email in `.html` and `.pdf` formats
- See guidelines for details on expectations for presentation and write up
- If you are not a MEDS student and/or do not want a blog post, the `.pdf` alone is fine

Announcements/check-in

Course evaluations

- Incredibly valuable for me, this course, and for the development of MEDS more broadly!
- Some changes from previous years' feedback:
    - Added logistic regression
    - Integrated more code snippets into lecture materials
    - More environmental examples (too much economics...)
    - Slower pace of lecture content
    - More definitions of mathematical objects
    - More lectures on stats in practice

Announcements/check-in

Class plan for remainder of the quarter

- 11/28: time series + spatial data
- 11/30: spatial interpolation
- 12/05: spatial kriging in `R`
- 12/07: guest lecture -- stats in ecohydrology!

Today

Refresher: types of spatial data

Vectors/objects, rasters/fields

Today

Refresher: types of spatial data

Vectors/objects, rasters/fields

A common challenge: spatial interpolation

Sample vs. population, points to fields

Today

Refresher: types of spatial data

Vectors/objects, rasters/fields

A common challenge: spatial interpolation

Sample vs. population, points to fields

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. 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. Examples: Elevation. Temperature. Wind direction.

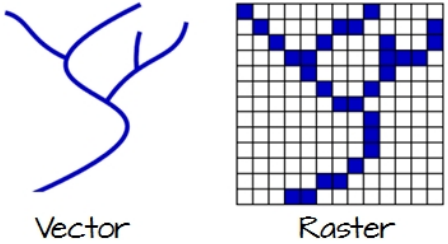
Spatial data

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?

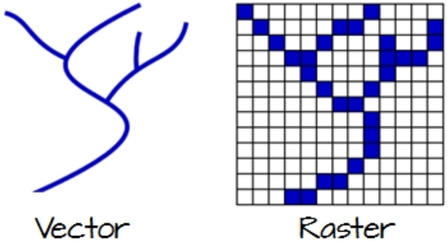
A: Usually, but it depends.



Spatial data

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

A: Usually, but it depends.

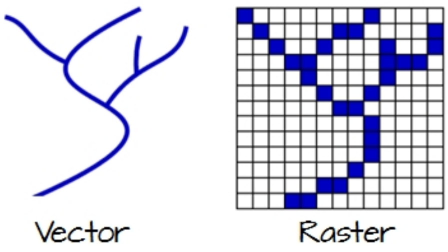


- Usually it will be easier to represent **objects** with **vector data** and **fields** with **raster** data, but ultimately this depends on what analysis you want to run

Spatial data

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

A: Usually, but it depends.

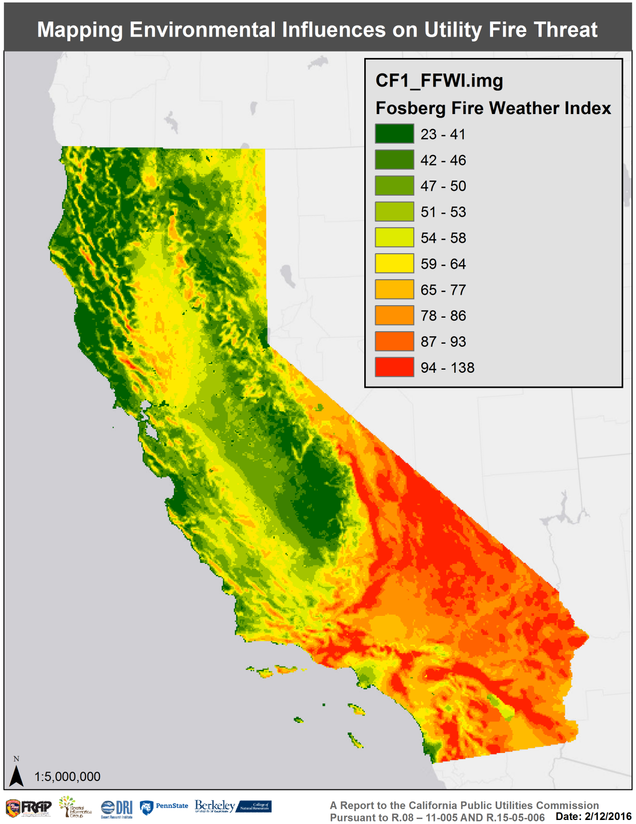


- Usually it will be easier to represent **objects** with **vector data** and **fields** with **raster** data, but ultimately this depends on what analysis you want to run
- Luckily, `R` makes it easy to switch back and forth (but we need to be careful and intentional when transforming!)

Spatial interpolation

Spatial interpolation

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



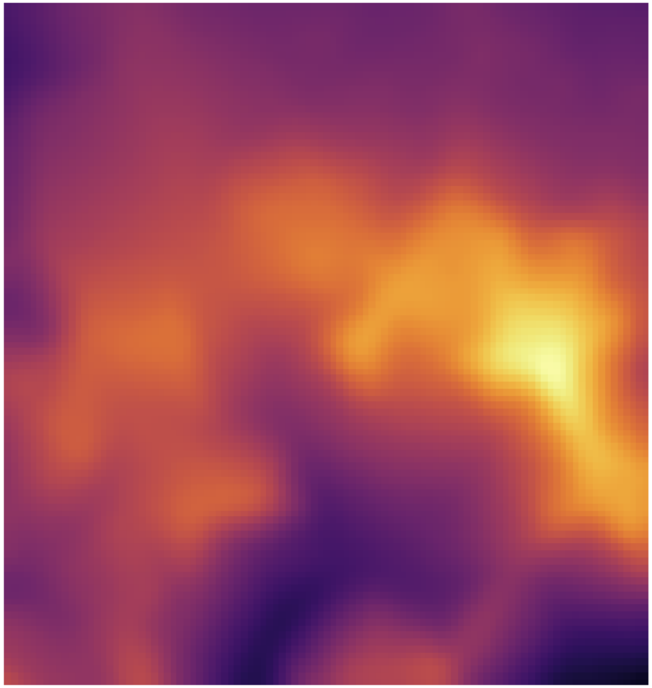
Spatial interpolation

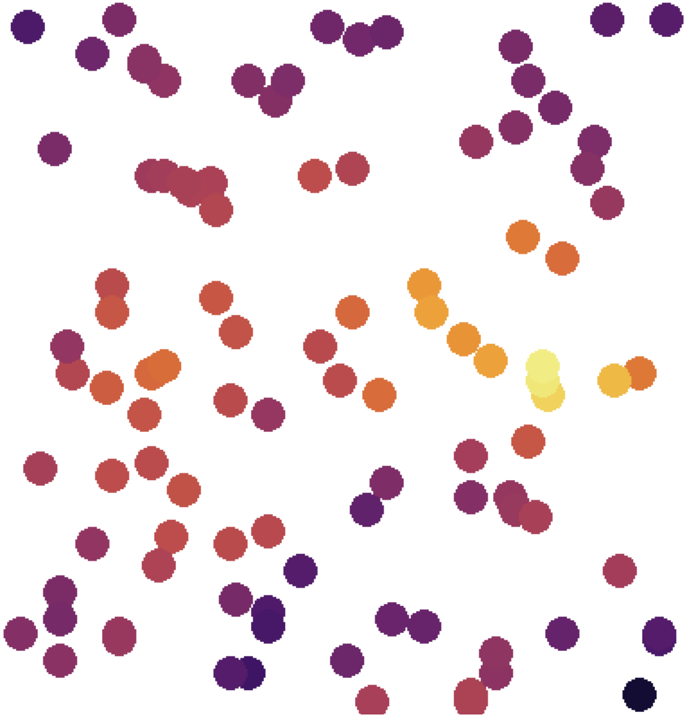
But we are doing **statistics!**

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

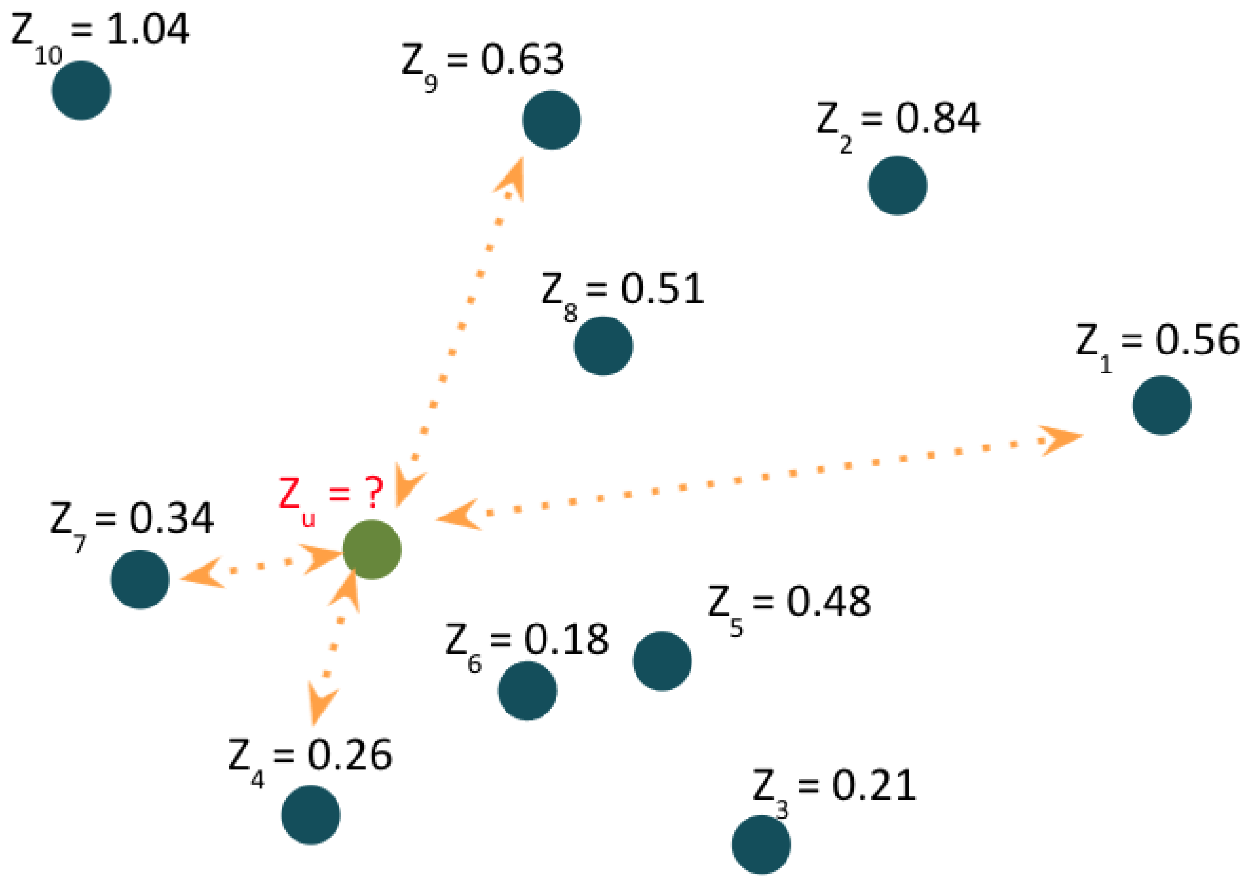
Spatial interpolation aims to predict $Z(x_0)$ using a linear combination of the values in the sampled locations:

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

where λ_i are weights applied to each sampled location.

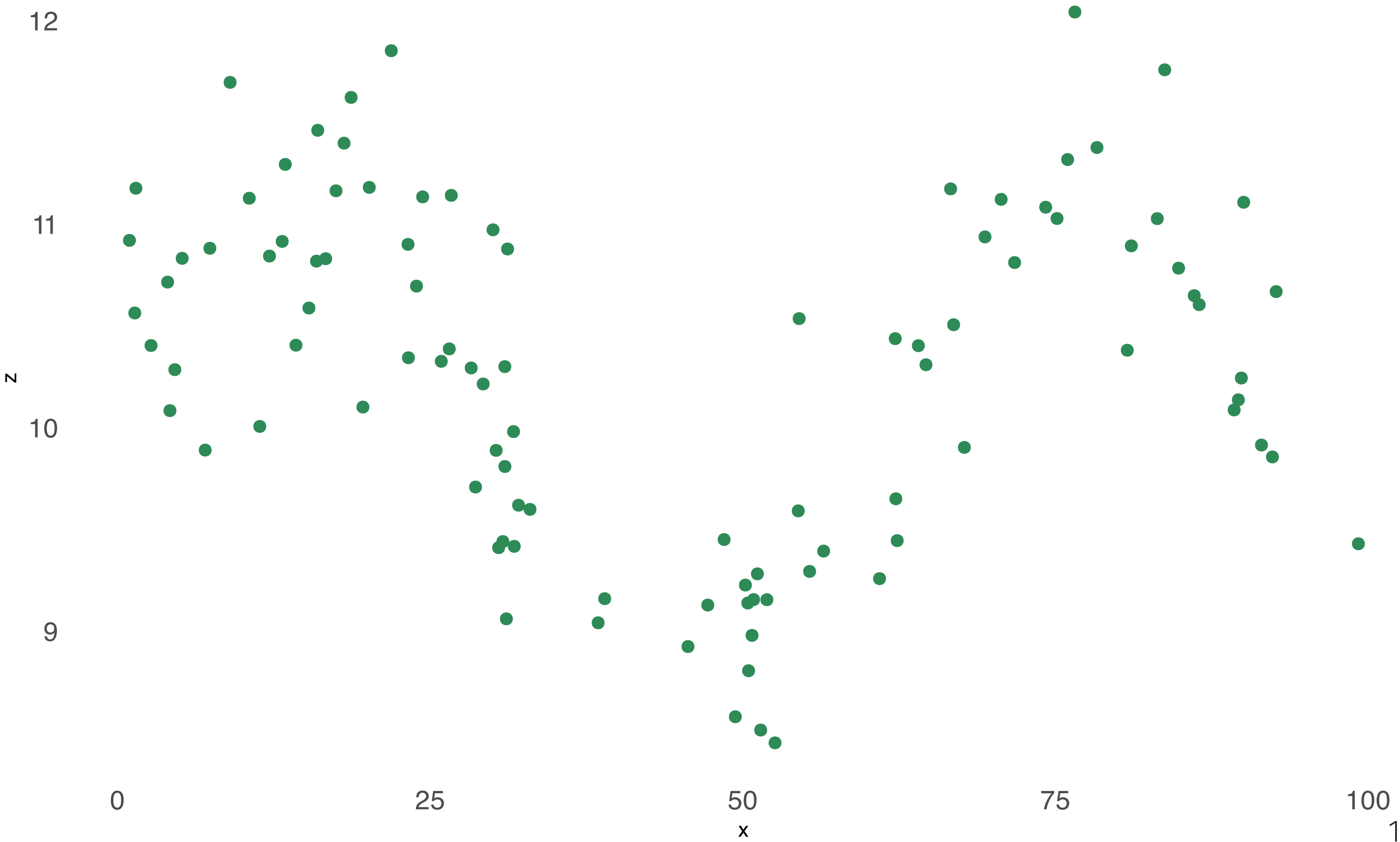
- All spatial interpolation methods assume or derive a set of λ's to compute $\hat{Z}$'s

Interpolation in pictures



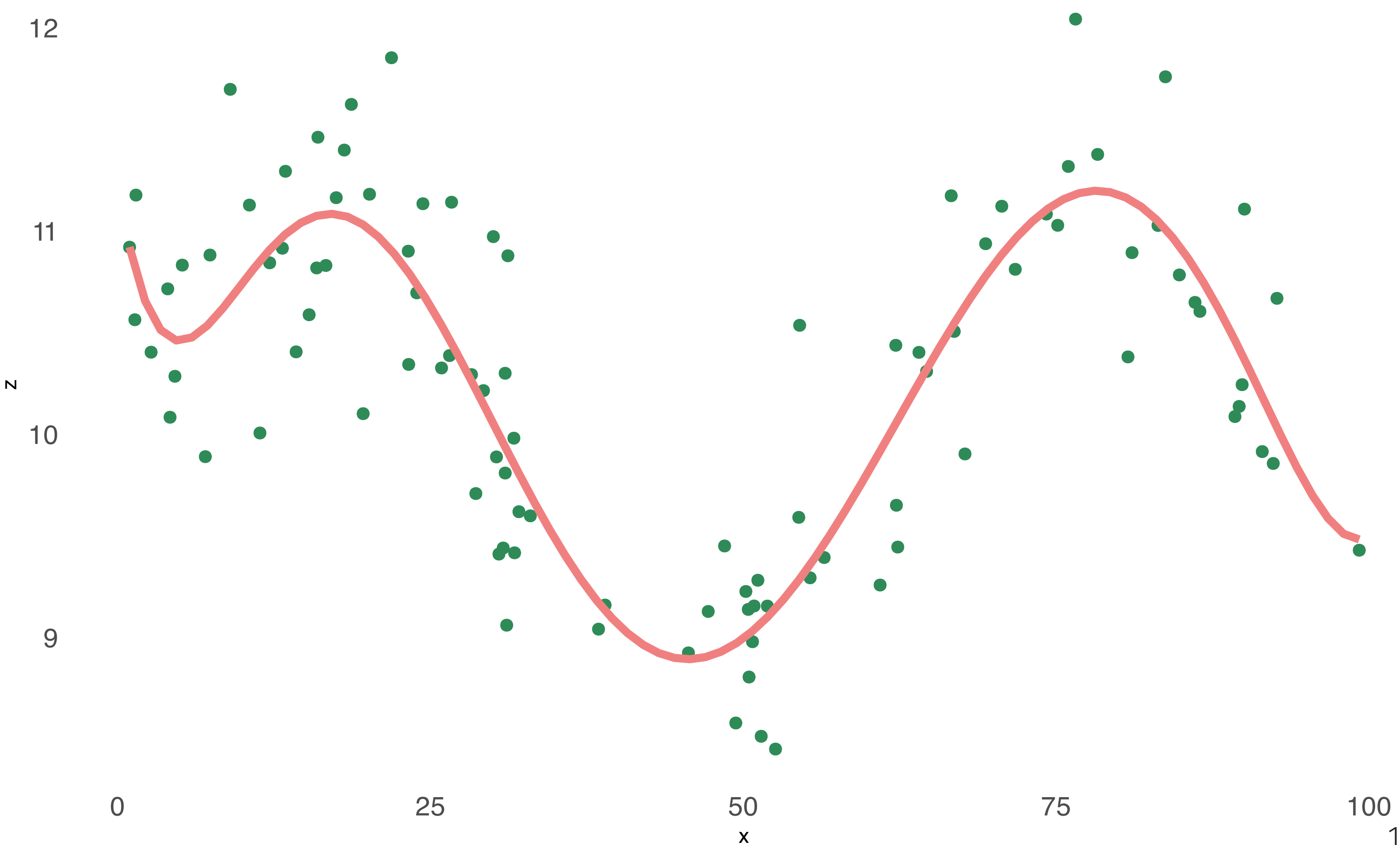
Interpolation in one dimension

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



Interpolation in one dimension

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

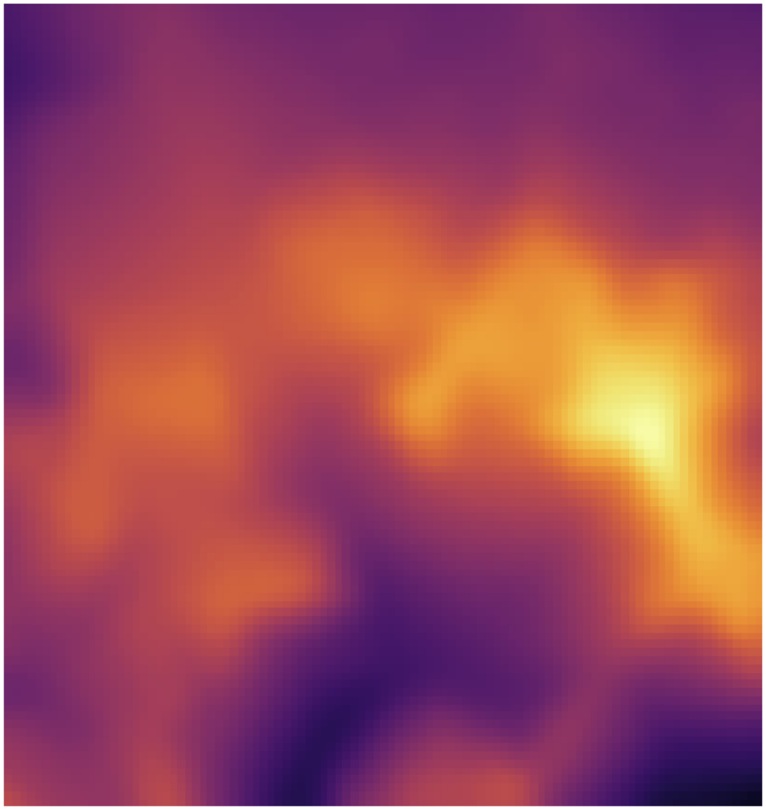


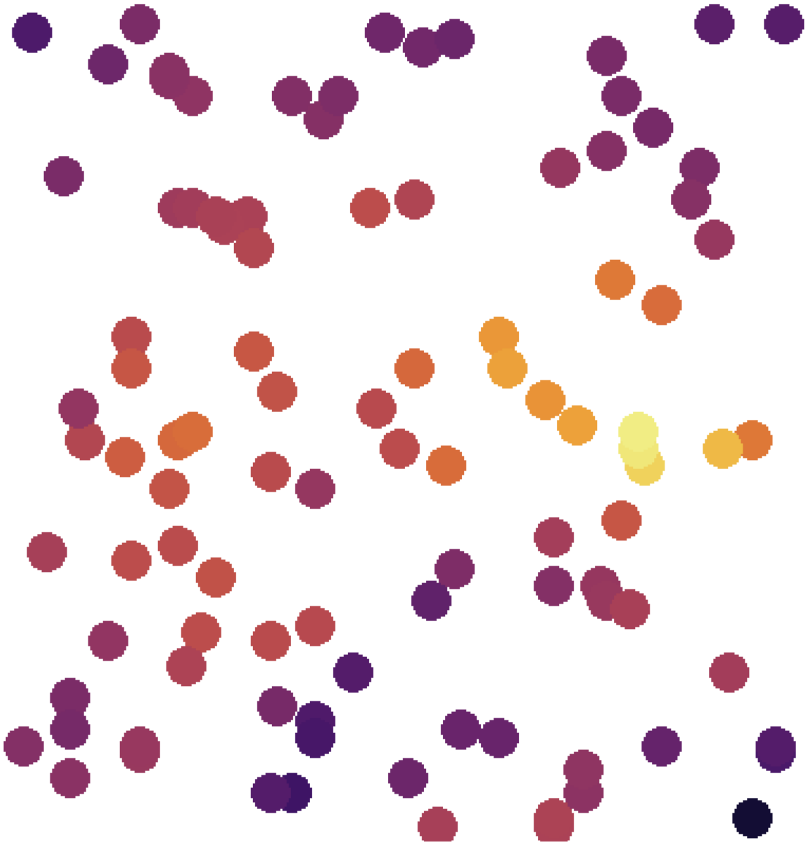
Interpolation in two dimensions

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

Interpolation in two dimensions

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





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 location:

$$\hat{Z}(x_0, y_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0 + \hat{\beta}_2 y_0 + \hat{\beta}_3 x_0 y_0 + \hat{\beta}_4 x_0^2 + \hat{\beta}_5 y_0^2 + \ldots$$

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 location:

$$\hat{Z}(x_0, y_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0 + \hat{\beta}_2 y_0 + \hat{\beta}_3 x_0 y_0 + \hat{\beta}_4 x_0^2 + \hat{\beta}_5 y_0^2 + \ldots$$

- **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 location:

$$\hat{Z}(x_0, y_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0 + \hat{\beta}_2 y_0 + \hat{\beta}_3 x_0 y_0 + \hat{\beta}_4 x_0^2 + \hat{\beta}_5 y_0^2 + \ldots$$

- **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 location:

$$\hat{Z}(x_0, y_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0 + \hat{\beta}_2 y_0 + \hat{\beta}_3 x_0 y_0 + \hat{\beta}_4 x_0^2 + \hat{\beta}_5 y_0^2 + \ldots$$

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

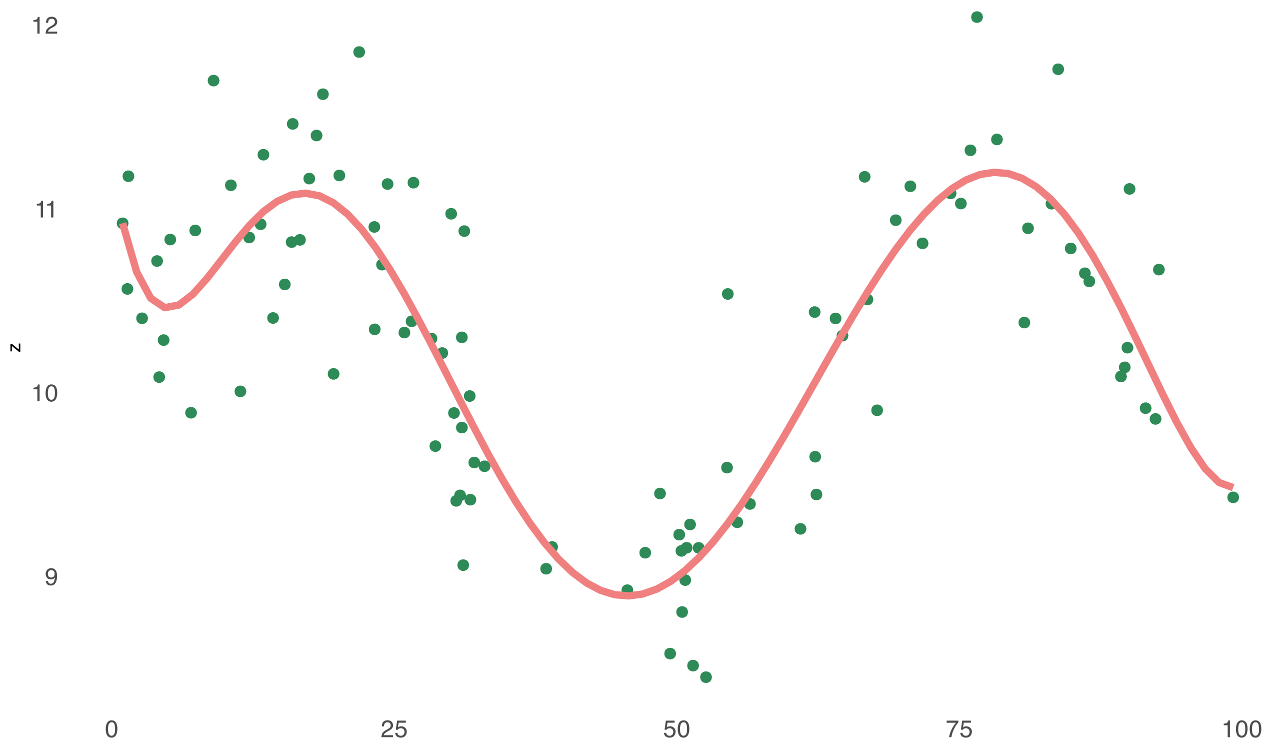
Exact: Predicts a value identical to the measured value.

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 variables

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



Polynomial regression interpolation

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

In 2-D:

```
mod = lm(z~x + y + x*y + x2 + y2 + x2*y2)
```

Interpolation methods

Nearest Neighbors (NN)

Interpolation methods

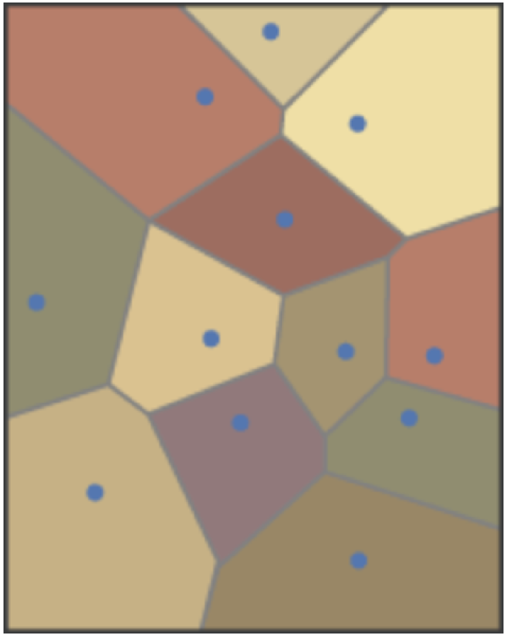
Nearest Neighbors (NN)

- Simple: Assign value of nearest observation in space

Interpolation methods

Nearest Neighbors (NN)

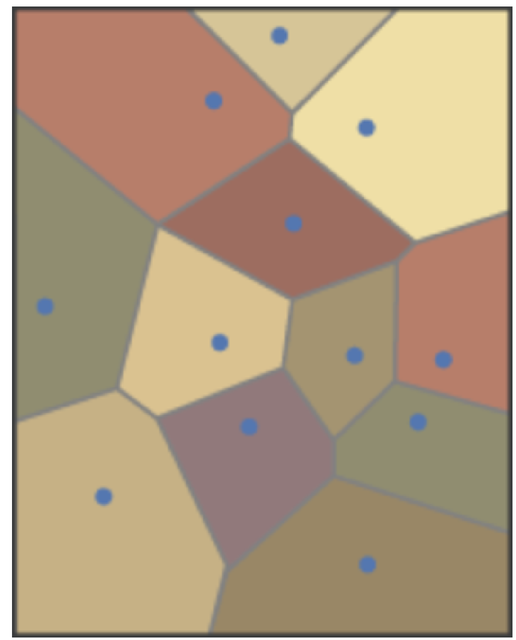
- Simple: Assign value of nearest observation in space



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 $\boldsymbol{\lambda}$ 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 `voronoi()` 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 `voronoi()` 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

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

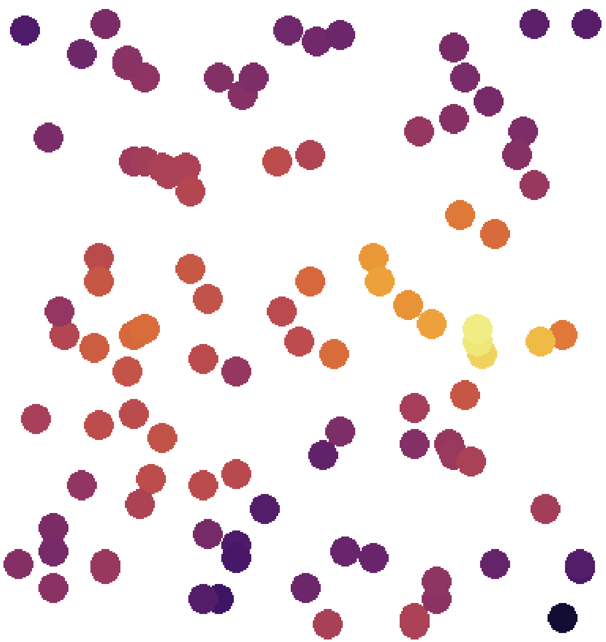
Inverse distance weighting

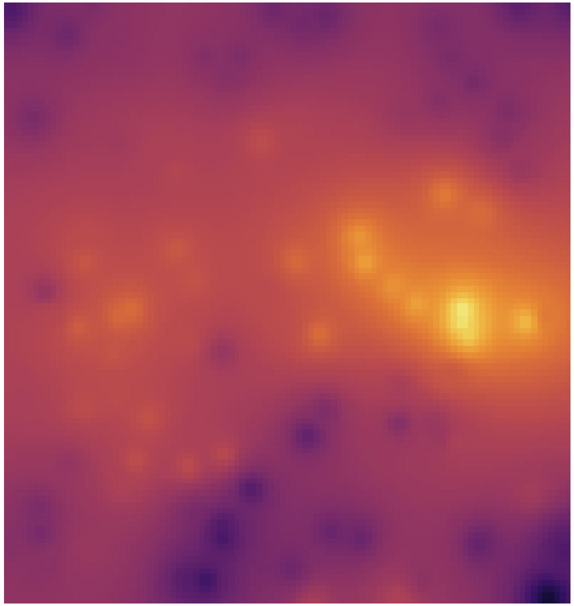
- **Pros:** Smooth, exact
- **Cons:** Difficult/computationally intensive (you need to compute distances for *all* pairs of points in the region!), all sampled observations influence $\hat{Z}(x_0)$, have to choose p somehow, result can be "clumpy"

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, result can be "clumpy"





Inverse distance weighting

Implementation in `R`

```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?

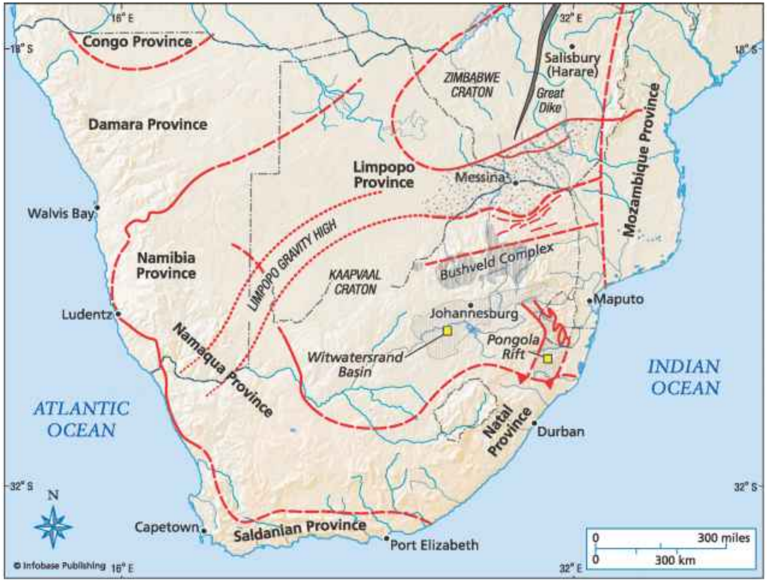
- It is *flexible* (i.e., less researcher decisions, more data-driven)
- Under certain assumptions it is the "best linear unbiased estimate" (sound like OLS yet??)
- You can recover an estimate *and* a standard error (i.e., it is *stochastic*)

Next up: Kriging details!

Kriging

Kriging: an origin story

The Witwatersrand ("Rand") in South Africa is known for its gold content. Mining engineers wanted to know where in the Rand was most likely to have a high gold content per block of ore.



Kriging: an origin story

- Many individual ore samples have been taken (**vector** data -- points)
- Underlying data is the content of the rock (**raster** data -- field)

Kriging: an origin story

- Many individual ore samples have been taken (**vector** data -- points)
- Underlying data is the content of the rock (**raster** data -- field)

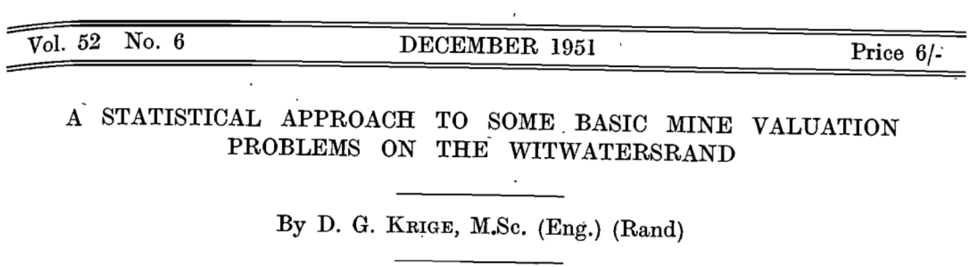
Spatial interpolation is highly valuable!

Kriging: an origin story

- Many individual ore samples have been taken (**vector** data -- points)
- Underlying data is the content of the rock (**raster** data -- field)

Spatial interpolation is highly valuable!

- **Danie Krige's solution:** [in his master's thesis!]
  - Use an estimator that minimizes the **mean squared prediction error** (very similar to OLS)
  - Show that it has a bunch of nice properties relative to other forms of spatial interpolation



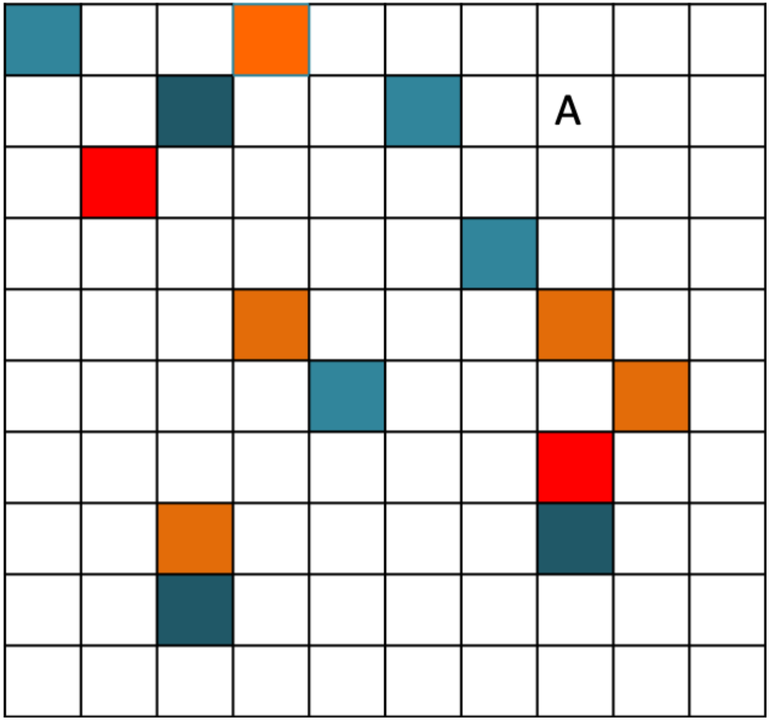
Vol. 52 No. 6 DECEMBER 1951 Price 6/-

A STATISTICAL APPROACH TO SOME BASIC MINE VALUATION PROBLEMS ON THE WITWATERSRAND

By D. G. KRIGE, M.Sc. (Eng.) (Rand)

Correlations in space

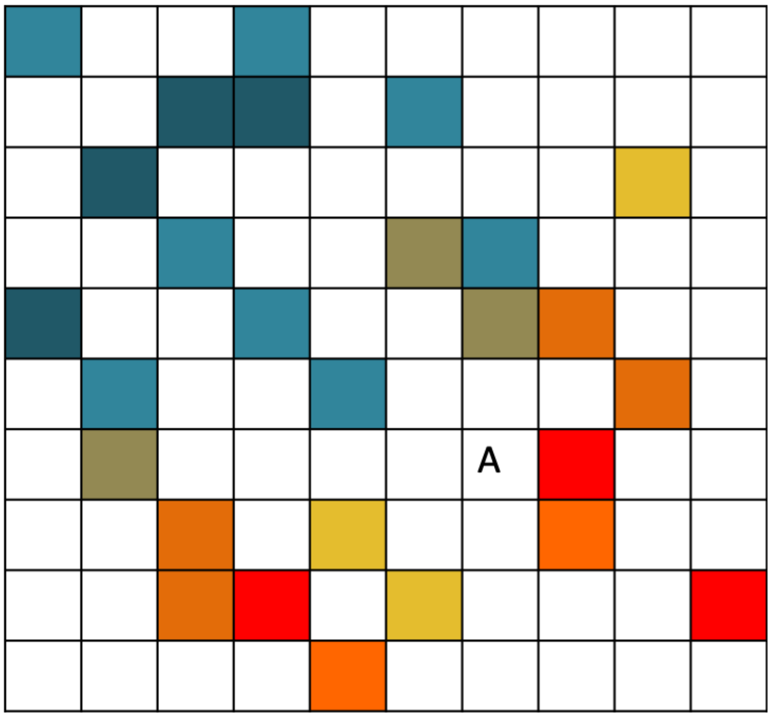
Q: If there is **no correlation** between values in nearby locations, can we predict new values based on our sample?



- Blue = low gold content; Red = high gold content
- **Zero** correlation between values in nearby locations
- Can you predict the gold content in location A based on this sample?

Correlations in space

Q: If there is **no correlation** between values in nearby locations, can we predict new values based on our sample?



- Blue = low gold content; Red = high gold content
- **Positive** correlation between values in nearby locations
- Now can you predict the gold content in location A based on *this* sample?
- *Why?*

Variogram

Key takeaway: quantifying spatial dependence is key to spatial interpolation

Variogram

Key takeaway: quantifying spatial dependence is key to spatial interpolation

A **variogram** describes spatial dependence:

Variogram

Key takeaway: quantifying spatial dependence is key to spatial interpolation

A **variogram** describes spatial dependence:

> A **variogram** shows the variance of values within groups of observations as a function of the *distance* between them

Variogram

Key takeaway: quantifying spatial dependence is key to spatial interpolation

A **variogram** describes spatial dependence:

> A **variogram** shows the variance of values within groups of observations as a function of the *distance* between them

Key concept: Variograms give us a way of understanding how correlated spatial observations are to those around them, and how that correlation "decays" as points get further apart

Variogram

Key takeaway: quantifying spatial dependence is key to spatial interpolation

A **variogram** describes spatial dependence:

> A **variogram** shows the variance of values within groups of observations as a function of the *distance* between them

Key concept: Variograms give us a way of understanding how correlated spatial observations are to those around them, and how that correlation "decays" as points get further apart

Mining example: Variogram gives a measure of how much two samples taken from the mining area will vary in gold percentage depending on the distance between the samples. Samples farther apart will vary more than those taken close together.

Variogram

Let $Z(x)$ be the value at location x, and $Z(x+h)$ be the value at a location h units away from x.

Variogram

Let $Z(x)$ be the value at location x, and $Z(x+h)$ be the value at a location h units away from x.

Variogram:

$$2\gamma(x+h, x) = var(Z(x+h) - Z(x))$$

Variogram

Let $Z(x)$ be the value at location x, and $Z(x+h)$ be the value at a location h units away from x.

Variogram:

$$2\gamma(x+h, x) = var(Z(x+h) - Z(x))$$

We often discuss the **semi-variogram**, which is:

$$\gamma(x+h, x) = \frac{1}{2} var(Z(x+h) - Z(x))$$

Variogram

Let $Z(x)$ be the value at location x, and $Z(x+h)$ be the value at a location h units away from x.

Variogram:

$$2\gamma(x+h, x) = var(Z(x+h) - Z(x))$$

We often discuss the **semi-variogram**, which is:

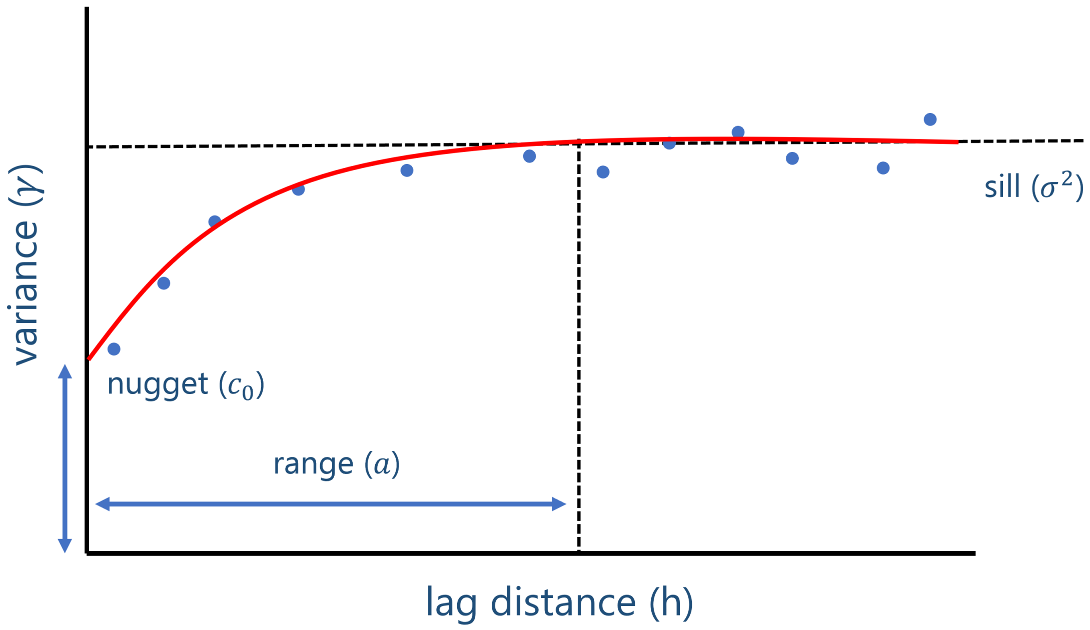
$$\gamma(x+h, x) = \frac{1}{2}var(Z(x+h) - Z(x))$$

Why? Recall:

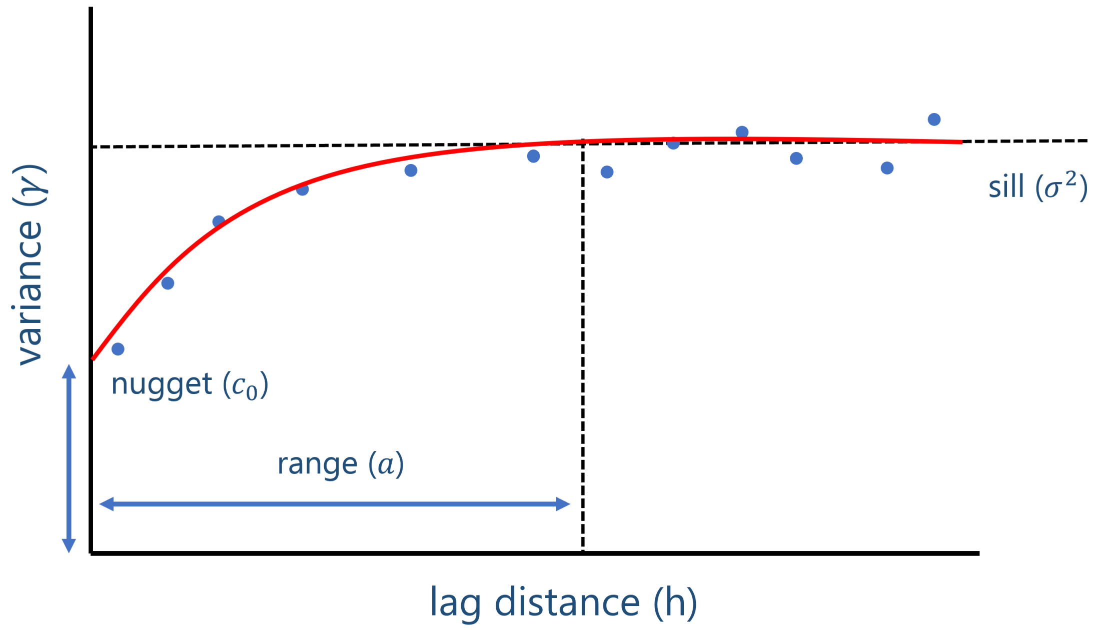
$$var(a-b) = var(a) + var(b) - 2cov(a, b)$$

So, for a "stationary" variogram, we have
$\gamma(x+h, x) = var(Z(x)) - cov(Z(x), Z(x+h))$

Variogram: in pictures



Variogram: in pictures



- **Nugget:** At $h = 0$, residual variance is from microscale effects or measurement error
- **Sill:** The stationary maximum variance -- no more covariance
- **Range:** Separation distance beyond which there is no covariance

Estimating a (semi)variogram

Empirical semivariogram

$$\hat{\gamma}(h \pm \delta) = \frac{1}{2N(h \pm \delta)} \sum_{(i,j) \in N(h \pm \delta)} |z_i - z_j|^2$$

Estimating a (semi)variogram

Empirical semivariogram

$$\hat{\gamma}(h \pm \delta) = \frac{1}{2N(h \pm \delta)} \sum_{(i,j) \in N(h \pm \delta)} |z_i - z_j|^2$$

Why?

- You probably don't have many samples *exactly* h units apart

Estimating a (semi)variogram

Empirical semivariogram

$$\hat{\gamma}(h \pm \delta) = \frac{1}{2N(h \pm \delta)} \sum_{(i,j) \in N(h \pm \delta)} |z_i - z_j|^2$$

Why?

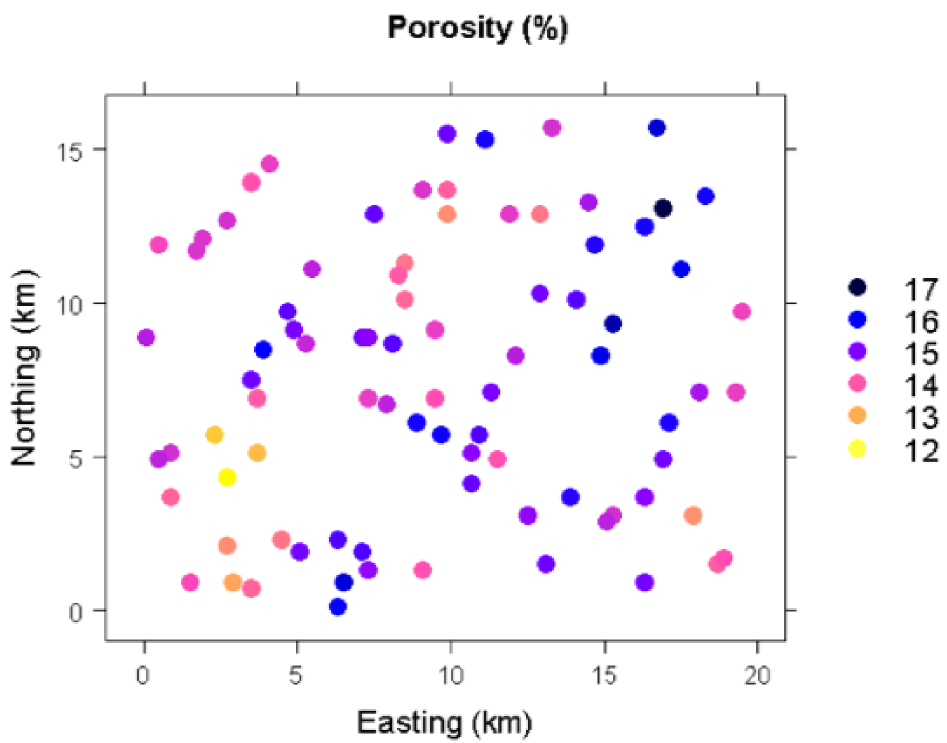
- You probably don't have many samples *exactly* h units apart

How?

- Draw "donuts" of width δ and average distance h around each point
- Compute differences in values for each pair of points, square them
- Take an average!

Empirical variogram example

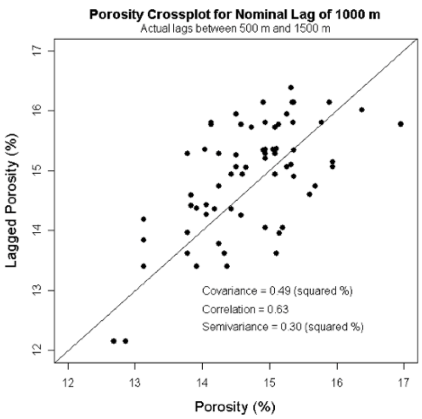
- Bohling's *Introduction to Geostatistics and Variogram Analysis*
- Porosity values in a bean field
- 85 wells sampled



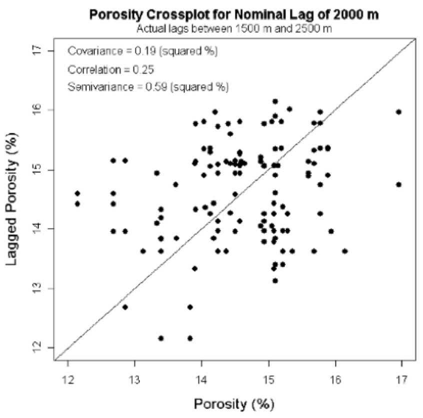
Empirical variogram example

For various values of h and a fixed δ, compute semivariance:

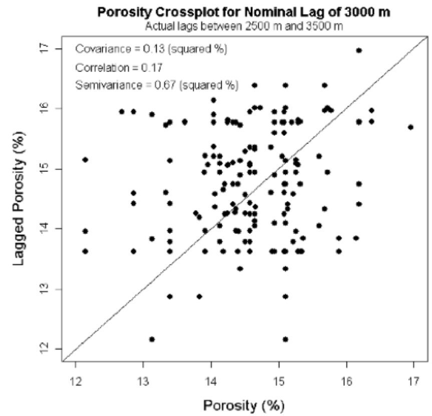
Separation: 500 - 1500 m



Separation: 1500 - 2500 m

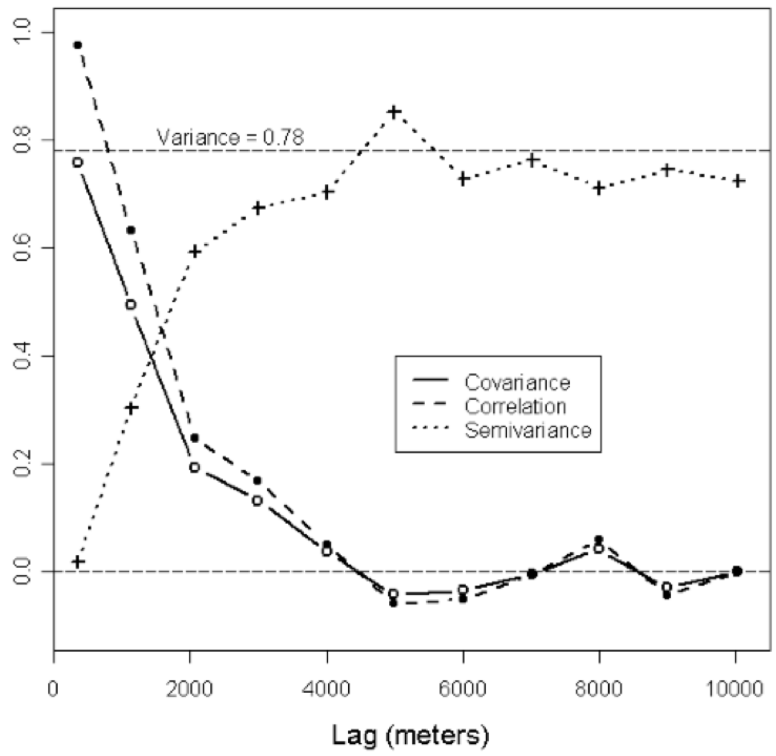


Separation: 2500 m - 3500 m



Empirical variogram example

Plot your semivariances:



Empirical variogram example

Then choose (or optimize) a **variogram model** to fit through the semivariance points:

- Exponential
- Spherical
- Gaussian
- ...

Empirical variogram example

Then choose (or optimize) a **variogram model** to fit through the semivariance points:

- Exponential
- Spherical
- Gaussian
- ...

Many more details on variograms here or in any geostatistics textbook (e.g., Cressie and Wikle, 2011)

Back to kriging

Recall that our goal is a prediction of a value $\hat{Z}(x_0)$ based on observations in all sampled locations:

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

Back to kriging

Recall that our goal is a prediction of a value $\hat{Z}(x_0)$ based on observations in all sampled locations:

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

In **kriging** (and many spatial interpolation methods), the λ_i weights **decay** as distance between x_0 and x_i grows larger

--

> How do we find the weights in kriging?

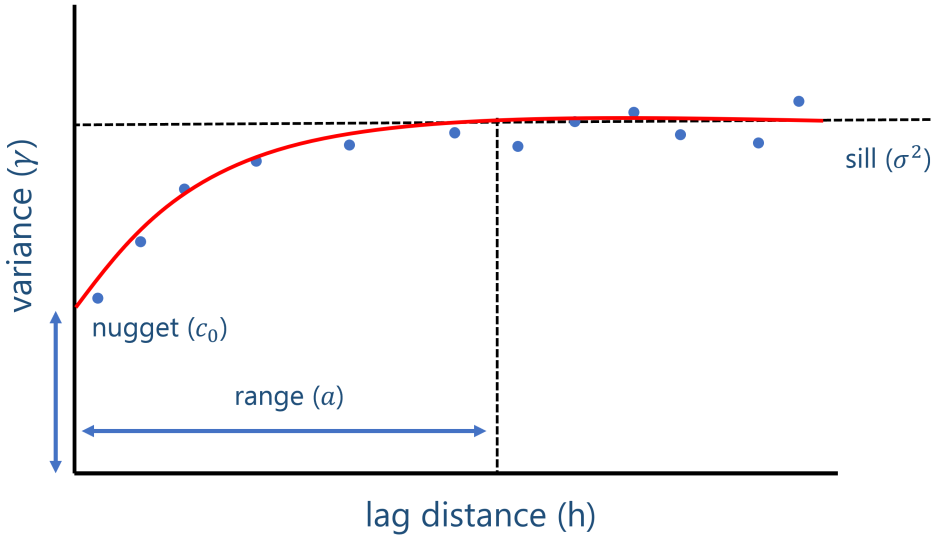
Kriging weights

> How do we find the weights in kriging?

Kriging weights

> How do we find the weights in kriging?

Hint:



The **variogram** tells us how correlated values are with other values near them, and how this correlation falls as distance grows. It is a **key input** into the kriging solution.

Deriving the kriging solution

Note: full derivation in Cressie and Wikle (2011) [this is a very shorthand version]

Goal: minimize mean squared prediction error

$$min_\lambda \; E[(Z(x_0) - \sum_i^m \lambda_i Z(x_i))^2] \text{ subject to } \sum_i^m \lambda_i = 1$$

Deriving the kriging solution

Note: full derivation in Cressie and Wikle (2011) [this is a very shorthand version]

Goal: minimize mean squared prediction error

$$min_{\lambda}\ E[(Z(x_0) - \sum_{i}^{m} \lambda_i Z(x_i))^2] \textbf{ subject to } \sum_{i}^{m} \lambda_i = 1$$

To solve:

1. Take derivatives with respect to each λ_i
2. Set each first order condition = 0
3. Solve system of equations for λ_i^* values that minimize mean squared error

Deriving the kriging solution

Result:

$$\hat{Z}(x_0) = \underbrace{\{\tilde{\gamma}(x_0) + \mathbf{1}(1 - \mathbf{1}'\mathbf{\Gamma}_Z^{-1}\tilde{\gamma}(x_0))/(\mathbf{1}'\mathbf{\Gamma}_Z^{-1}\mathbf{1})\}'\mathbf{\Gamma}_Z^{-1}}_{\hat{\lambda}} Z$$

- where $\tilde{\gamma}(x_0)$ is the vector containing the semivariogram evaluated between x_0 and every other point, and
- $\mathbf{\Gamma}_Z$ is the $m \times m$ matrix containing all semivariogram evaluations for all sampled point pairs.

Deriving the kriging solution

Result:

$$\hat{Z}(x_0) = \underbrace{\{\tilde{\gamma}(x_0) + \mathbf{1}(1 - \mathbf{1}'\mathbf{\Gamma}_Z^{-1}\tilde{\gamma}(x_0))/(\mathbf{1}'\mathbf{\Gamma}_Z^{-1}\mathbf{1})\}'\mathbf{\Gamma}_Z^{-1}}_{\hat{\lambda}} Z$$

- where $\tilde{\gamma}(x_0)$ is the vector containing the semivariogram evaluated between x_0 and every other point, and
- $\mathbf{\Gamma}_Z$ is the $m \times m$ matrix containing all semivariogram evaluations for all sampled point pairs.
- See Cressie and Wikle (2011) for similar derivation for $\sigma^2(x_0)$, an estimate of the prediction error

Deriving the kriging solution

Result:

$$\hat{Z}(x_0) = \underbrace{\{\tilde{\gamma}(x_0) + \mathbf{1}(1 - \mathbf{1}'\mathbf{\Gamma}_Z^{-1}\tilde{\gamma}(x_0))/(\mathbf{1}'\mathbf{\Gamma}_Z^{-1}\mathbf{1})\}'\mathbf{\Gamma}_Z^{-1}}_{\hat{\lambda}} Z$$

- where $\tilde{\gamma}(x_0)$ is the vector containing the semivariogram evaluated between x_0 and every other point, and
- $\mathbf{\Gamma}_Z$ is the $m \times m$ matrix containing all semivariogram evaluations for all sampled point pairs.
- See Cressie and Wikle (2011) for similar derivation for $\sigma^2(x_0)$, an estimate of the prediction error

Other helpful resources here

Forms of kriging

There are **three** main forms of kriging:

Forms of kriging

There are **three** main forms of kriging:

1. **Simple:** The mean of the entire field is **constant** and **known** [restrictive, not usually realistic]

Forms of kriging

There are **three** main forms of kriging:

1. **Simple:** The mean of the entire field is **constant** and **known** [restrictive, not usually realistic]

2. **Ordinary:** The mean of the entire field is **constant** but **unknown** [derivation shown above; most common]

Forms of kriging

There are **three** main forms of kriging:

1. **Simple:** The mean of the entire field is **constant** and **known** [restrictive, not usually realistic]
2. **Ordinary:** The mean of the entire field is **constant** but **unknown** [derivation shown above; most common]
3. **Universal:** The mean of the field varies over space and can be estimated using measured variables [requires knowledge of and reason for trend in mean]

Forms of kriging

There are **three** main forms of kriging:

1. **Simple:** The mean of the entire field is **constant** and **known** [restrictive, not usually realistic]
2. **Ordinary:** The mean of the entire field is **constant** but **unknown** [derivation shown above; most common]
3. **Universal:** The mean of the field varies over space and can be estimated using measured variables [requires knowledge of and reason for trend in mean]

- There are also other forms! E.g., quantile kriging, log-normal kriging, IRFk-kriging, etc.

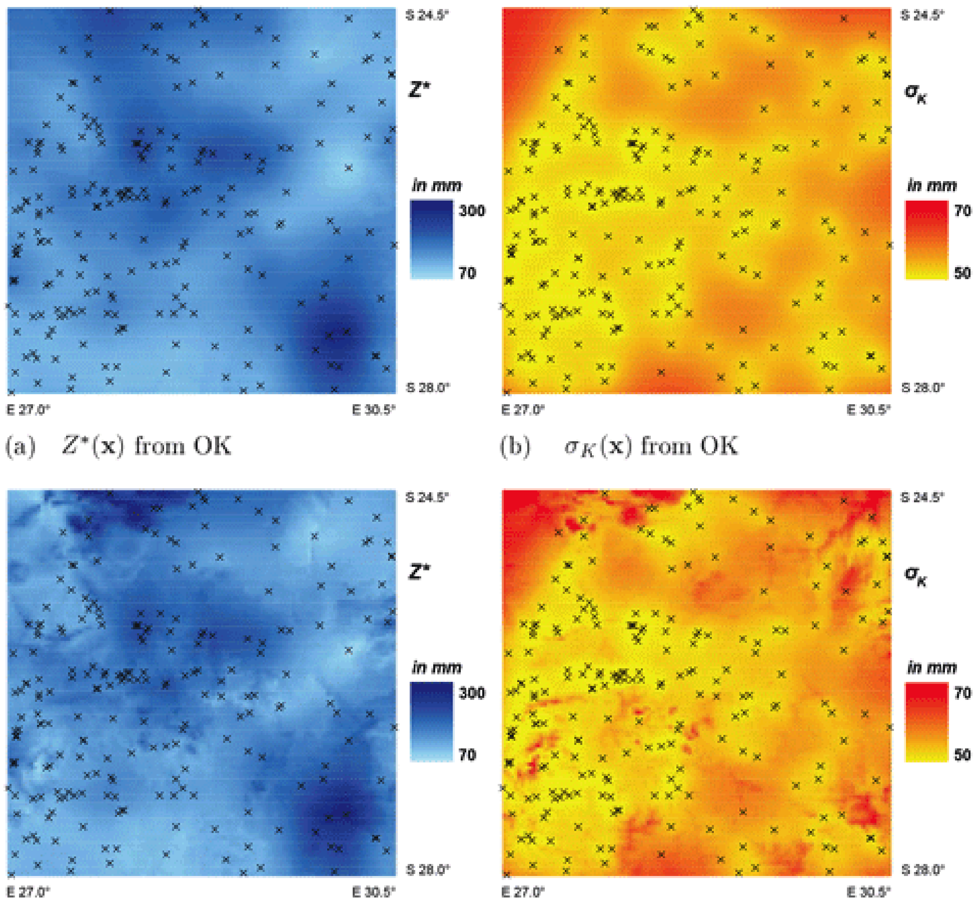
Forms of kriging

There are **three** main forms of kriging:

1. **Simple:** The mean of the entire field is **constant** and **known** [restrictive, not usually realistic]
2. **Ordinary:** The mean of the entire field is **constant** but **unknown** [derivation shown above; most common]
3. **Universal:** The mean of the field varies over space and can be estimated using measured variables [requires knowledge of and reason for trend in mean]

- There are also other forms! E.g., quantile kriging, log-normal kriging, IRFk-kriging, etc.
- We will work on implementation in `R` in the next lab.

Forms of kriging



(a) $Z^*(\mathbf{x})$ from OK
(b) $\sigma_K(\mathbf{x})$ from OK
(c) $Z^*(\mathbf{x})$ from EDK
(d) $\sigma_K(\mathbf{x})$ from EDK

Source: Lebrenz and Bardossy (2019)

Kriging summary

Pros:

- Under each set of assumptions specific to the kriging form, kriging is the best linear unbiased predictor ("BLUP")
- Weights are determined almost entirely by the data, instead of a-priori assumptions
- Exact
- Provides a measure of precision: $\sigma^2(x_0)$

Kriging summary

Pros:

- Under each set of assumptions specific to the kriging form, kriging is the best linear unbiased predictor ("BLUP")
- Weights are determined almost entirely by the data, instead of a-priori assumptions
- Exact
- Provides a measure of precision: $\sigma^2(x_0)$

Cons:

- Nonlinear methods may perform better (e.g., ML methods)
- Variogram has to be approximated/estimated
- Complex/computationally intensive

A note of caution on interpolation

All spatial interpolation approaches work best if:

- The observed data are relatively **dense and well distributed** throughout the region of interest

A note of caution on interpolation

All spatial interpolation approaches work best if:

- The observed data are relatively **dense and well distributed** throughout the region of interest
- You have a lot of observations

A note of caution on interpolation

All spatial interpolation approaches work best if:

- The observed data are relatively **dense and well distributed** throughout the region of interest
- You have a lot of observations

All spatial interpolation approaches should be used cautiously, especially if:

A note of caution on interpolation

All spatial interpolation approaches work best if:

- The observed data are relatively **dense and well distributed** throughout the region of interest
- You have a lot of observations

All spatial interpolation approaches should be used cautiously, especially if:

- You have **highly clustered** data with a lot of open space between them

A note of caution on interpolation

All spatial interpolation approaches work best if:

- The observed data are relatively **dense and well distributed** throughout the region of interest
- You have a lot of observations

All spatial interpolation approaches should be used cautiously, especially if:

- You have **highly clustered** data with a lot of open space between them
- You don't have very many observations

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