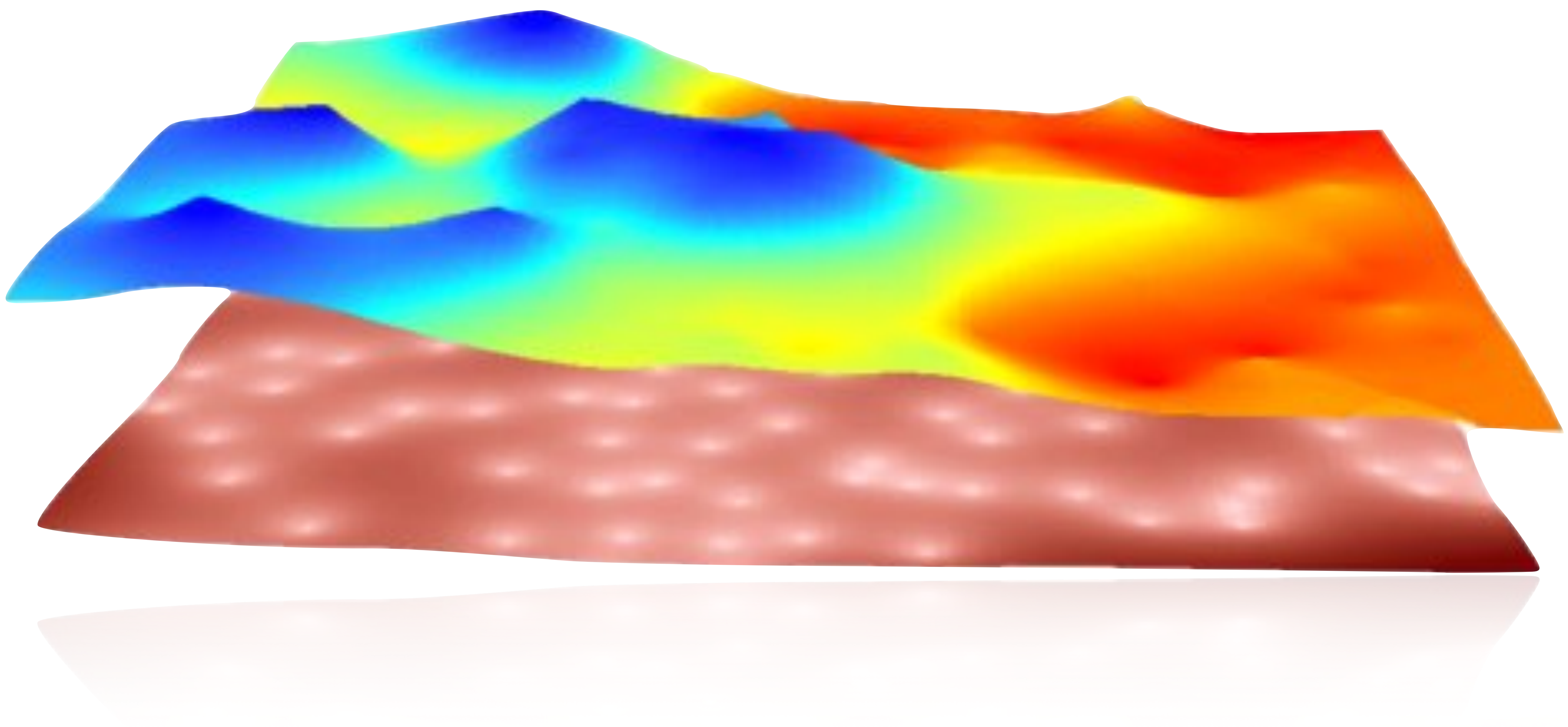


Make your own Kriging Interpolation Algorithm with Python

Dr. Jui-Fa Tsai (Gogoro Data Scientist)

Make your own Kriging Interpolation Algorithm with Python

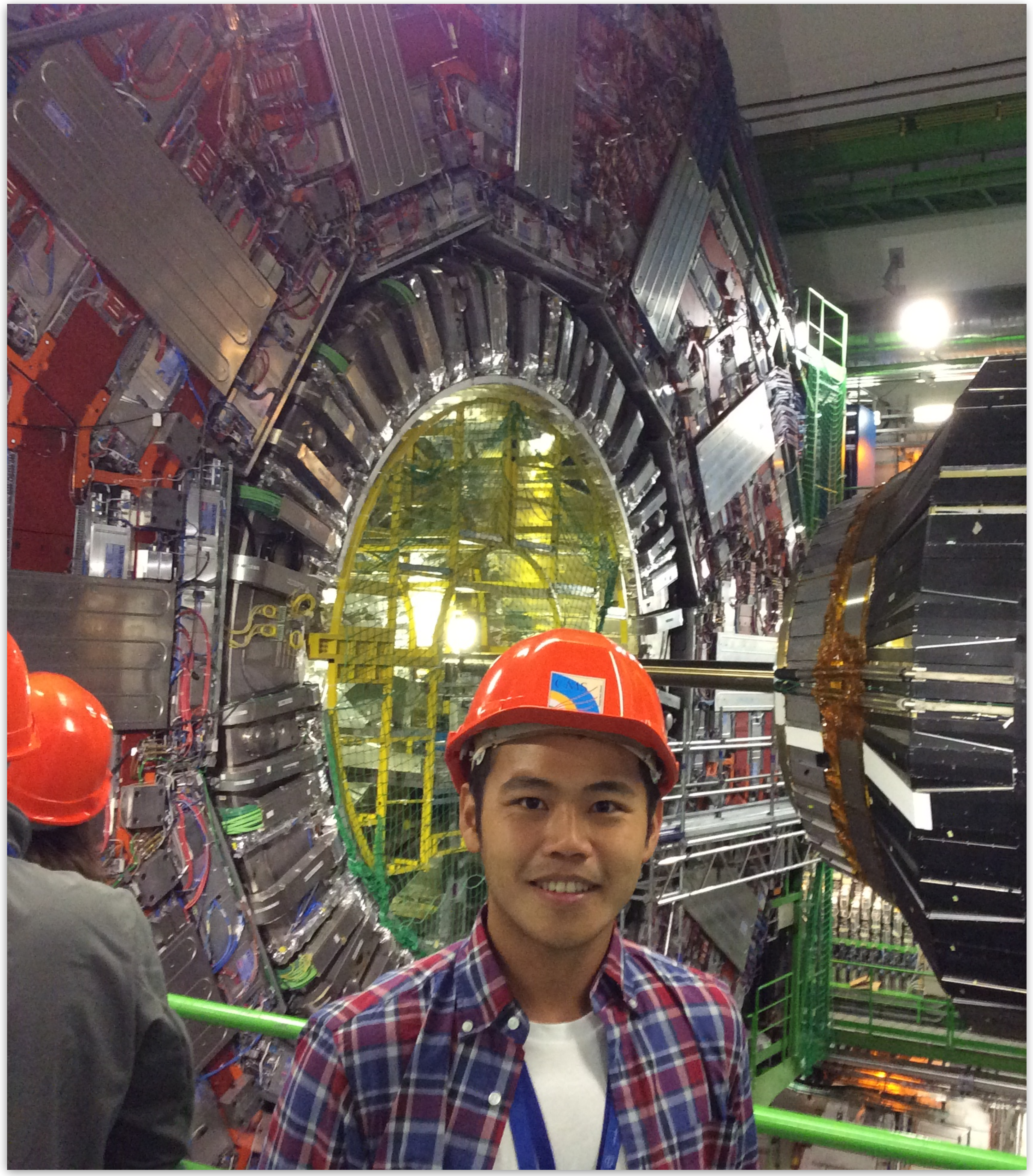
Kriging 是地理統計中常用的 interpolation 演算法，利用空間中量測點間的變異量與距離的關係來預測相近位置的觀測值。在資源有限的的情況下，常幫助地質學來了解未知地域的地質特性。除了在地質學上，這樣的空間預測演算法也能有其他應用。在大數據時代，現有的工具（例如 ArcGIS, QGIS）能處理的資料類型受限。這次的演講來分享如何利用台灣地震開放資料與 python 的套件（numpy geopandas scipy rasterio）來設計自己的 Kriging interpolation algorithm。內容包含數學推導與程式範本。

Python package :
numpy, pandas, geopandas, rasterio, shapley, scipy, matplotlib

Display tool : QGIS

INTRODUCTION

About me



Birth

July 1989 @ *Taichung, Taiwan*

Education

Ph.D of Physics, National Taiwan University (2017)

Experience

Data Scientist at Gogoro Network (2018 - now)

Visiting Scientist at CERN, *Geneva, Switzerland* (2013 - 2017)

Speciality

High-energy experimental physics.

Mathematical model building.

Statistics / big data analysis.

Machine learning

www.juifa-tsai.tw

juifa.tsai@gmail.com

gogoro network™

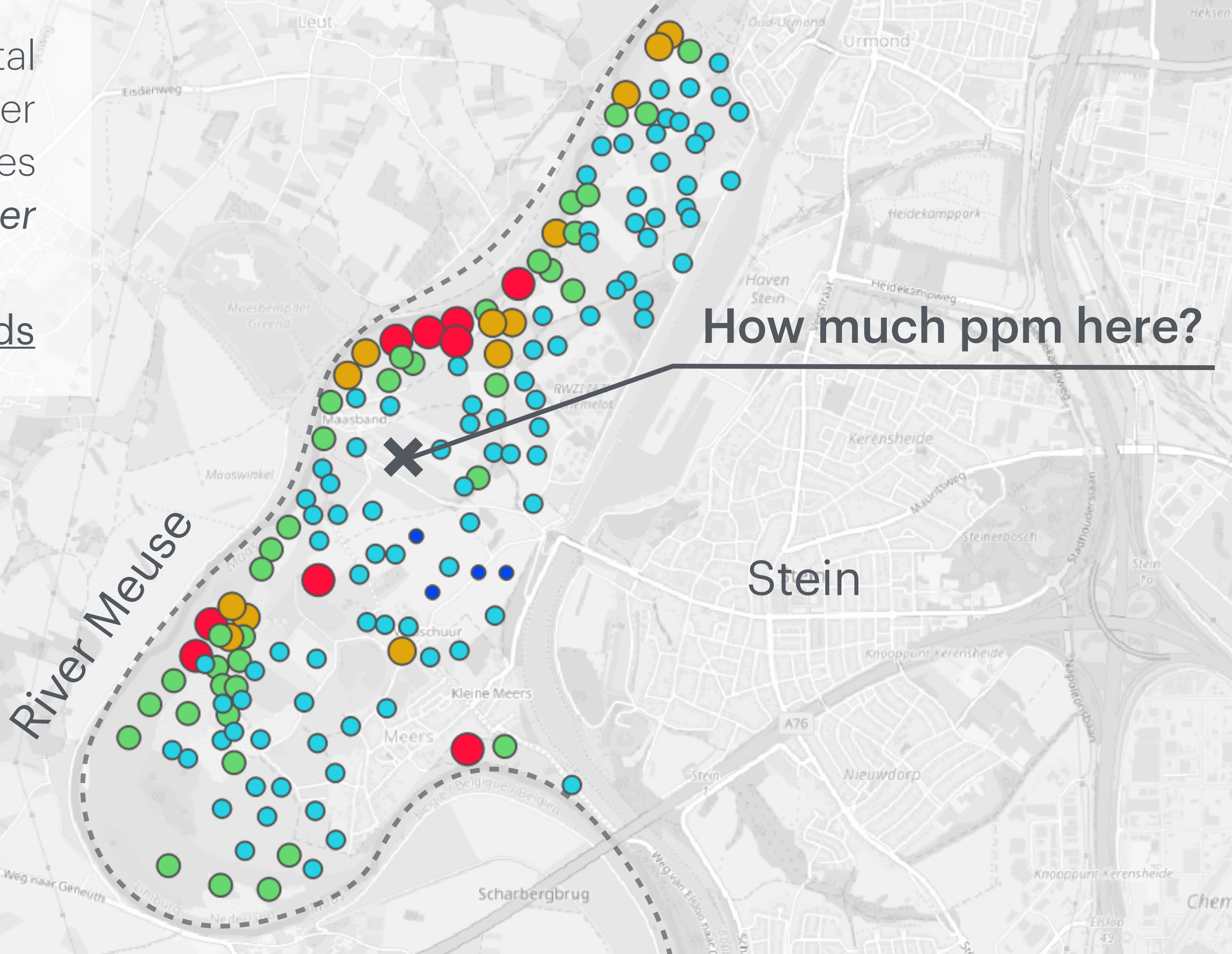
What is interpolation?

- In spatial analysis, the interpolation play key role to predict the interest as function of geometrical variables.
- Historically, interpolation helps to describe the distribution of mine.

- Open data

  The 155 samples of top soil heavy metal concentrations (ppm), along with a number of soil and landscape variables.The samples were collected in a flood plain of the *river Meuse*, near the village Stein

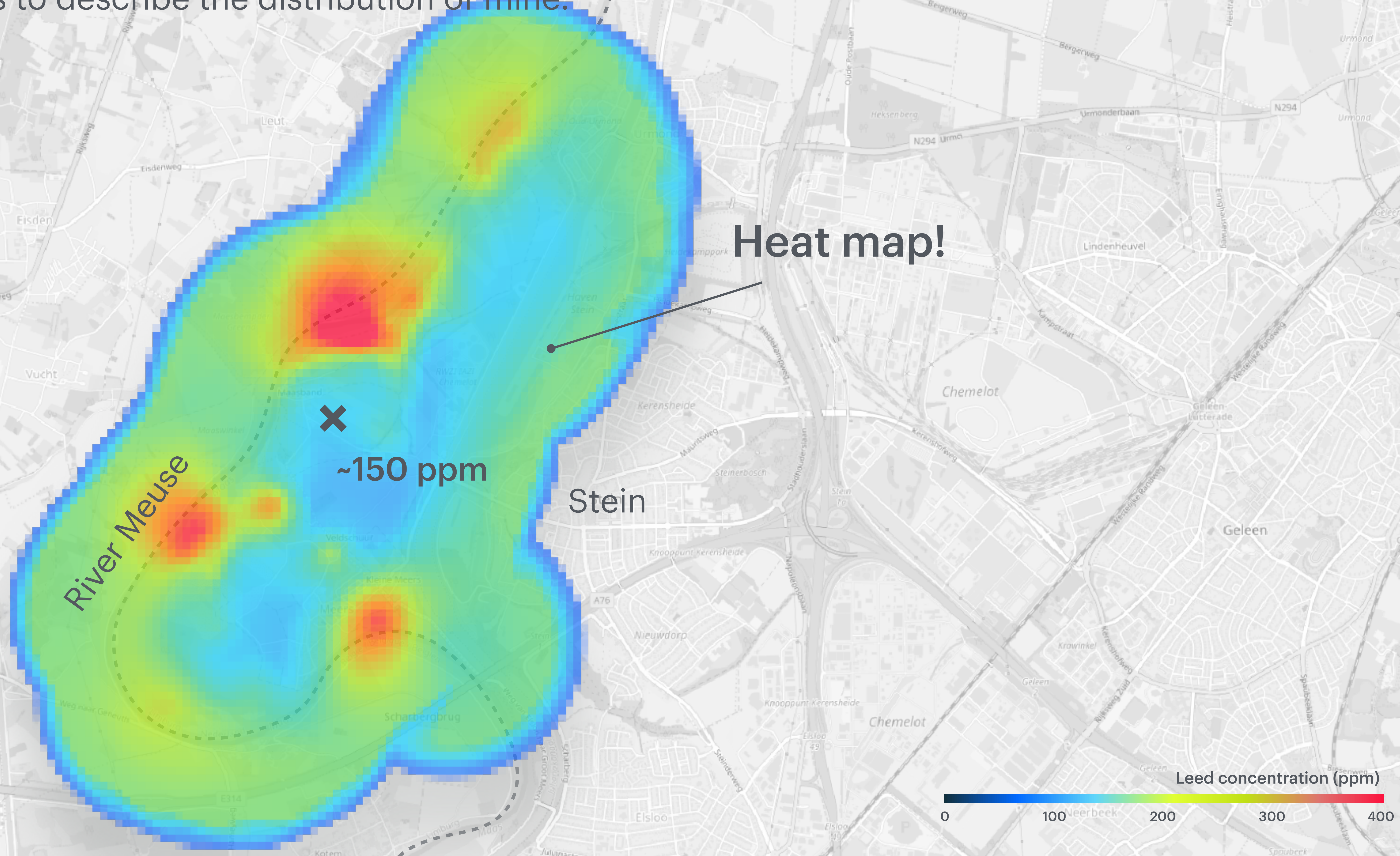
  http://spatial-analyst.net/book/meusegrids



INTRODUCTION

What is interpolation?

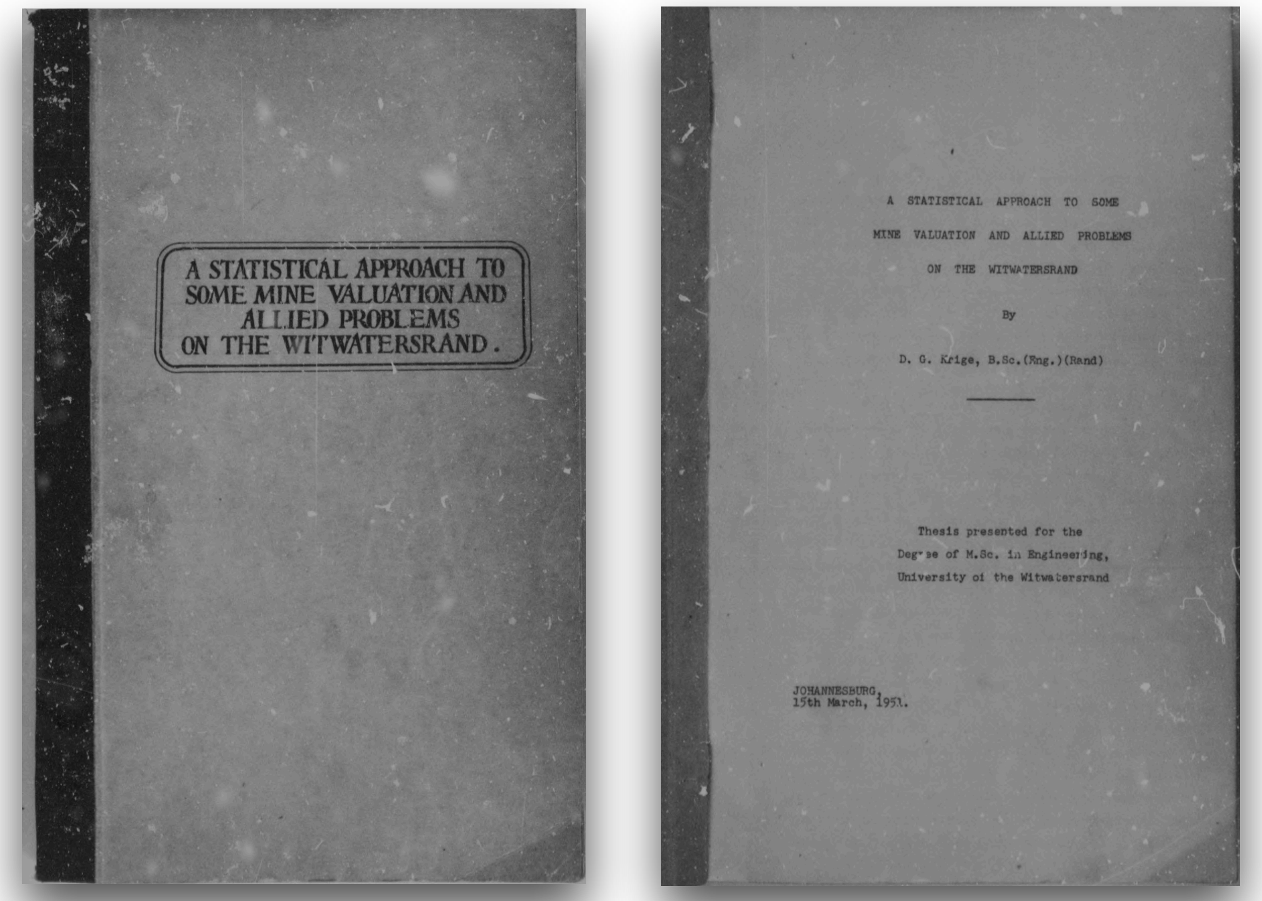
- In spatial analysis, the interpolation play key role to predict the interest as function of geometrical variables.
- Historically, interpolation helps to describe the distribution of mine.



Why & what is Kriging?

- Different from general interpolation algorithm, it learns from the covariances within measurements.
- Master thesis of *Daniel G. Krige* (mining engineer), formularize theoretically by *Georges Matheron* (mathematician)

Daniel G. Krige
(1919 - 2013)



Thesis (M.Sc.(Engineering)) - University of the Witwatersrand, Faculty of Engineering, 1951.

Georges Matheron
(1930 - 2000)

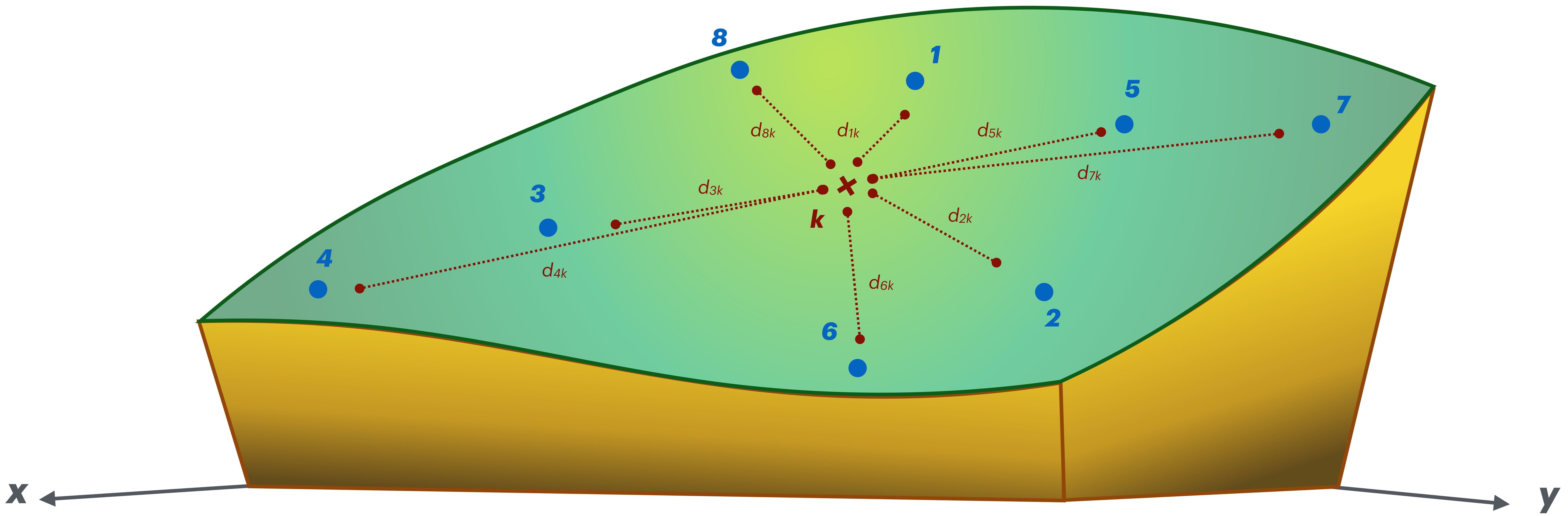
BASIC INTERPOLATION

Fundamental spatial interpolation

- The interest ***z*** at ***k*** point is predicted by the **neighbors *(1, …, i* ∈ *n_k)*** as function of their **distances *d_{ik}*.**
- Most of algorithm assumes the predicted ***$\hat{z}(s_k)$*** is the linear combination ***$z(s_i)$***, where **s** is the spatial vector.

$$\hat{z}(\mathbf{s}_k) = \sum_{i \in n_k} w(\mathbf{d}_{ik}) z(\mathbf{s}_i)$$

How to get the weights?



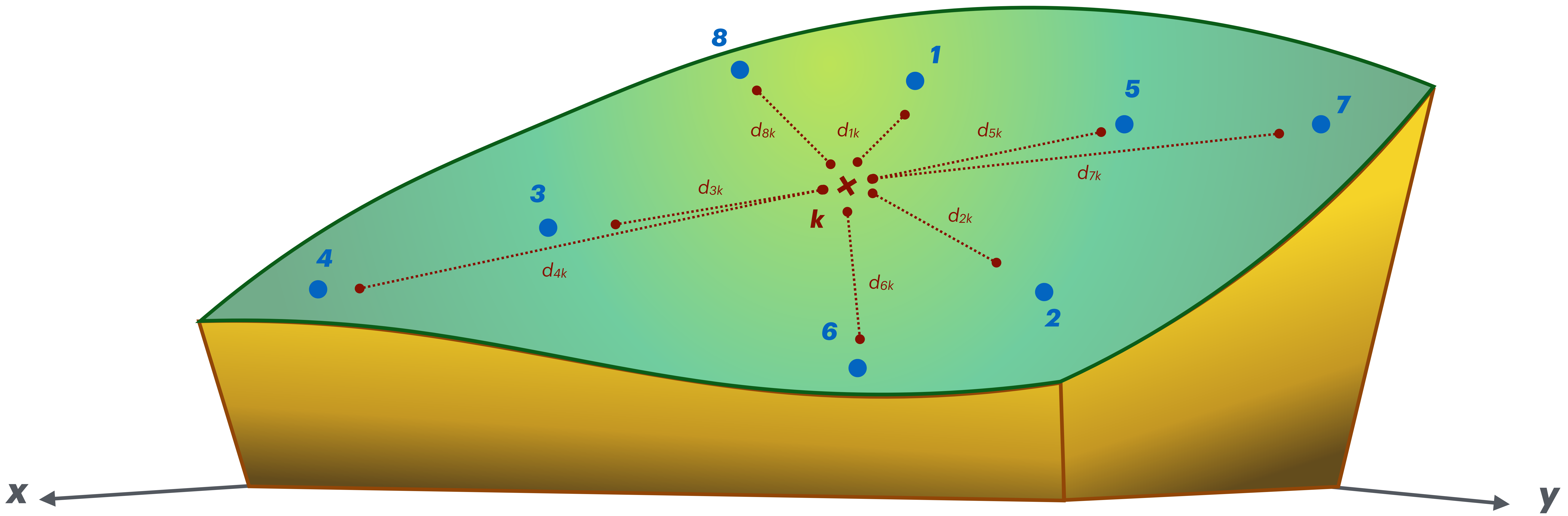
Well-known algorithm

- **Inverse Distance Weight algorithm (IDW)** is a common and straightforward interpolation algorithm.
- The weights are defined with inverse Minkowski distance, where $\boldsymbol{n_k}$ and $\boldsymbol{p}$ are the parameters.
  - $p = 1$: Manhattan distance.
  - $p = 2$: Euclidean distance.
  - n_k is the set of neatest neighbors.

$$\boxed{\hat{z}(\mathbf{s}_k) = \sum_{i \in n_k} w(\mathbf{d}_{ik}) z(\mathbf{s}_i)}$$

$$w(\mathbf{d}_{ik}) = \frac{|\mathbf{d}_{ik}|^{-p}}{\sum_{j \in n_k} |\mathbf{d}_{jk}|^{-p}}$$

$$\sum_{i \in n_k} w(\mathbf{d}_{ik}) = 1$$

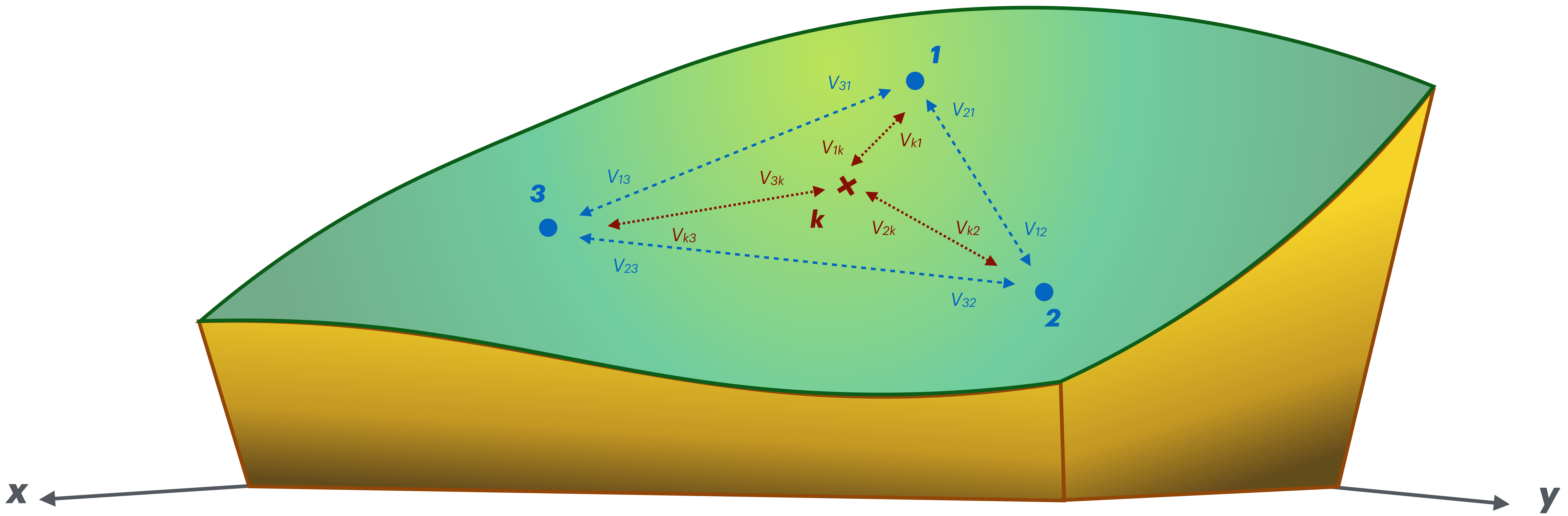


KRIGING INTERPOLATION

Learn spatial correlation from data

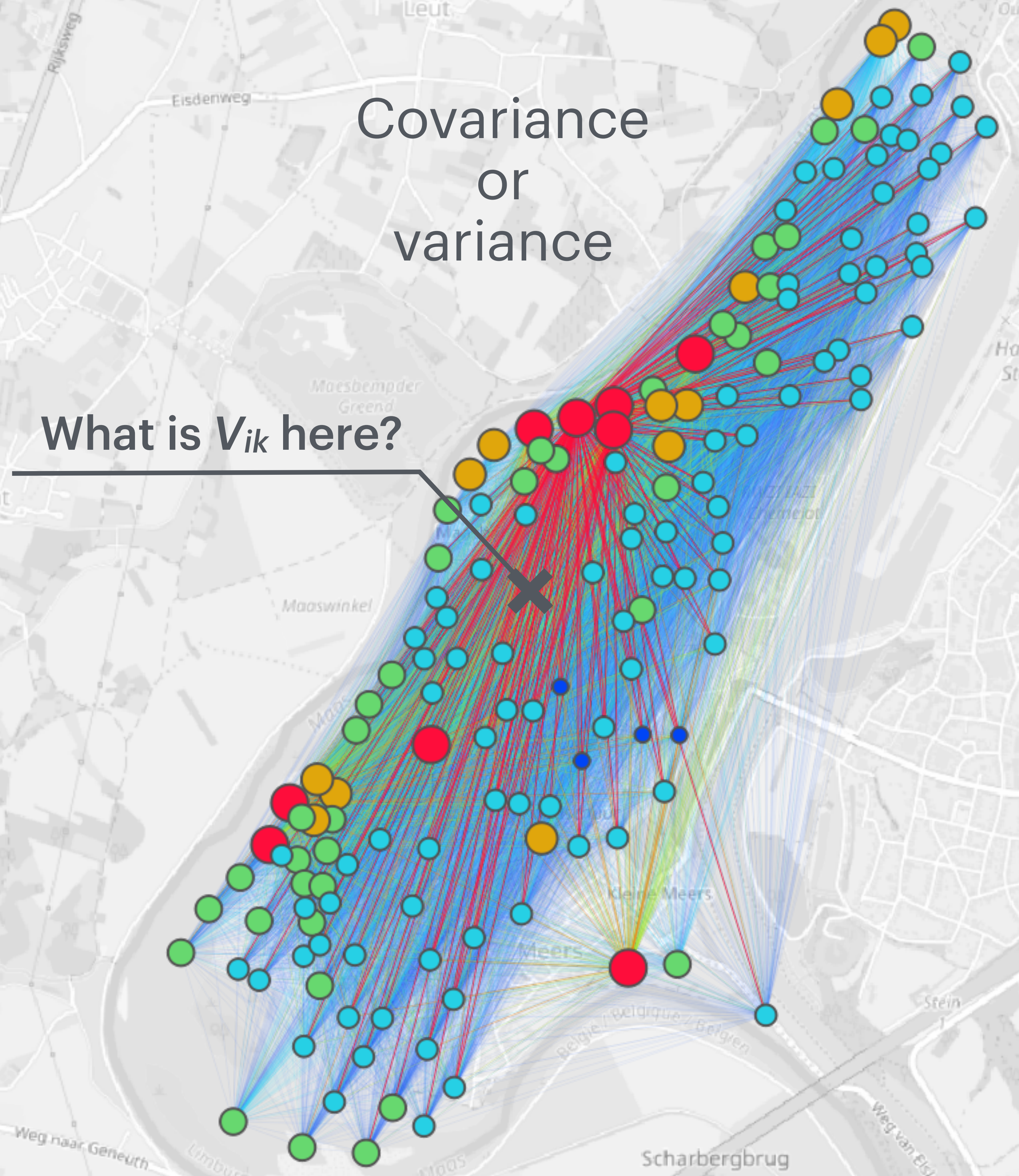
- **Kriging** is one kind of *Gaussian Process*, i.e. every finite linear combination of them is normally distributed.
- The weights are governed by the variances (covariance) of neighbors as the priors.

$$\hat{z}(\mathbf{s}_k) = \sum_{i \in n_k} w(\mathbf{d}_{ik}) z(\mathbf{s}_i) \longrightarrow \boxed{\begin{gathered}\hat{z}(\mathbf{s}_k) = \sum_{i \in n_k} w(V_{ik}) z(\mathbf{s}_i) \\ V_{ik} = V(\mathbf{d}_{ik})\end{gathered}}$$

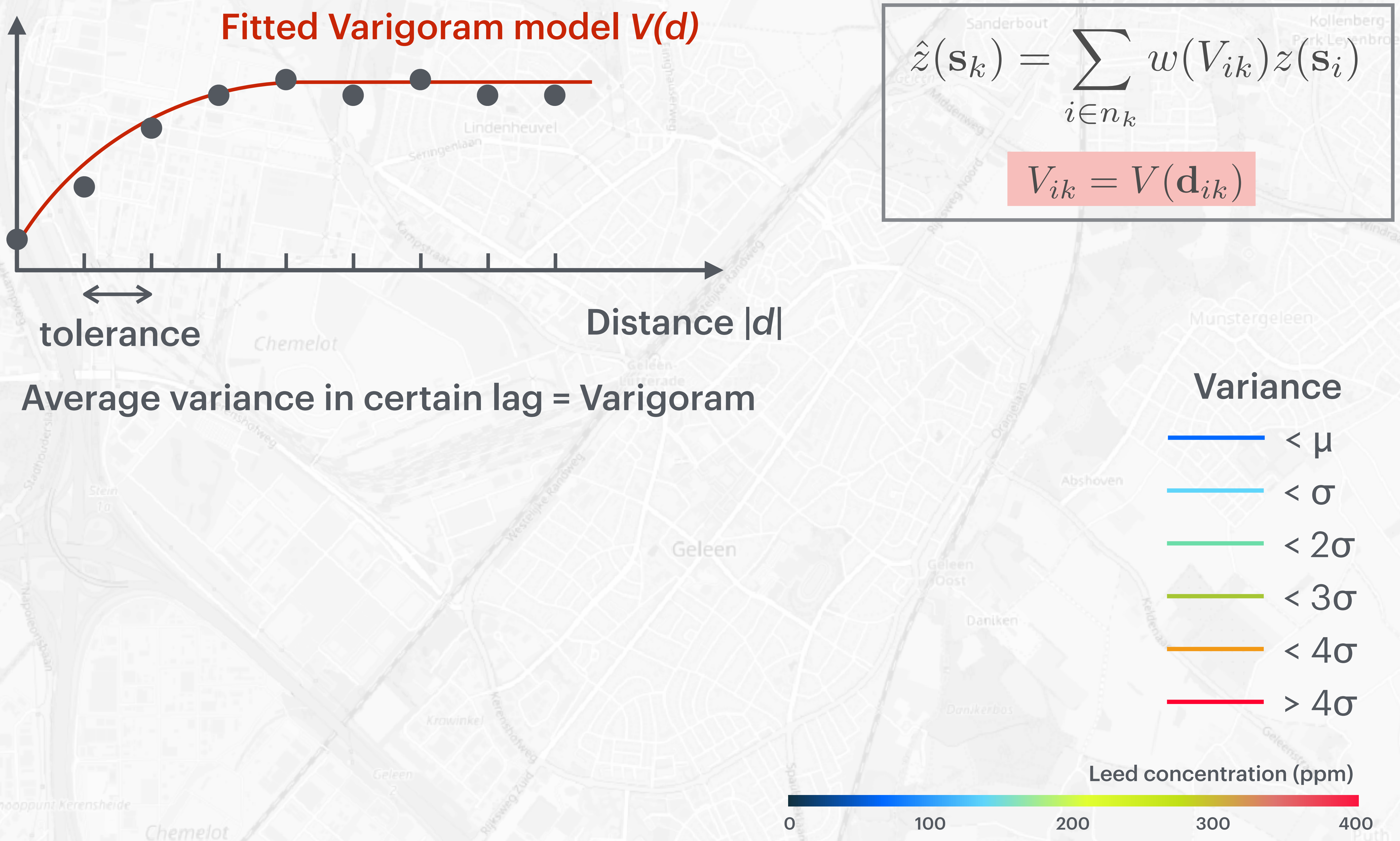


Two process in Kriging

- Train the covariance (variance) model as function of *Minkowski distance* $\mathbf{d}_{ij}$, called ***covariogram*** $\mathbf{C}_{ij}$ ***(variogram*** $\mathbf{V}_{ij}$***).***



Step 1.



Average variance in certain lag = Varigoram

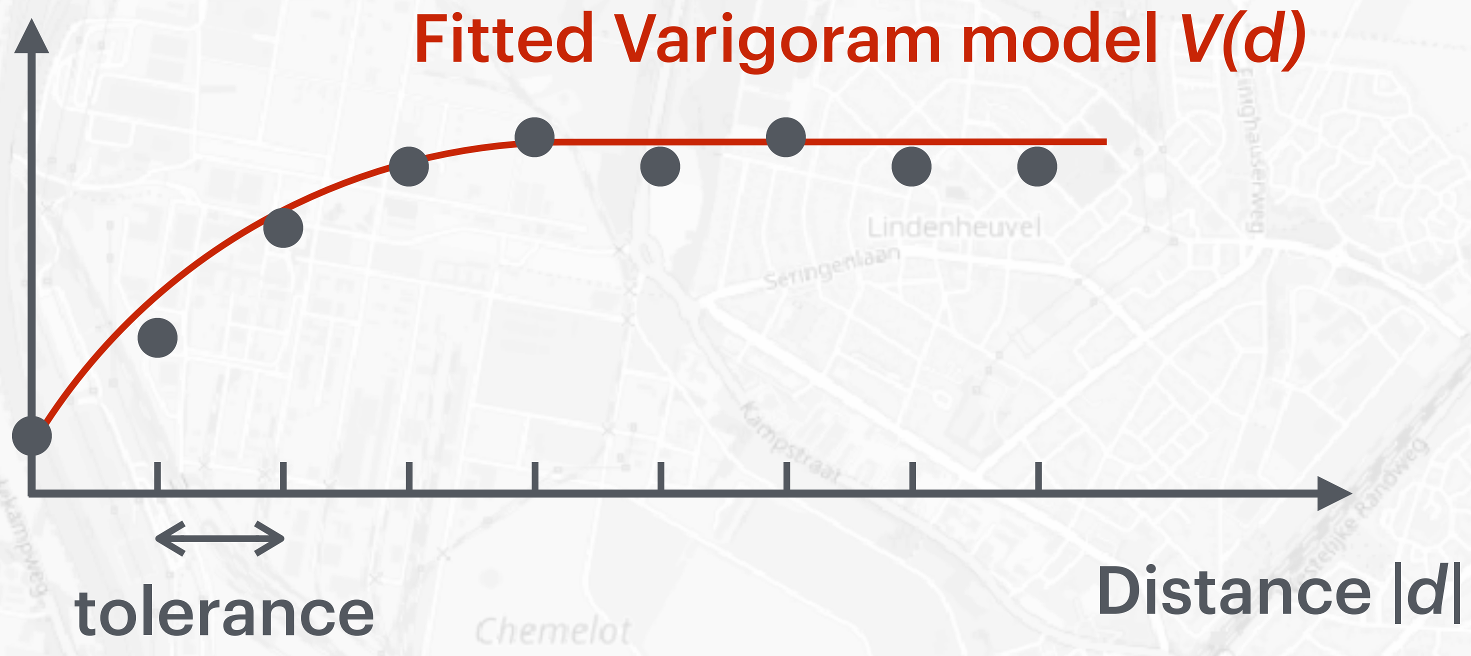
$$\hat{z}(\mathbf{s}_k) = \sum_{i \in n_k} w(V_{ik}) z(\mathbf{s}_i)$$

$$V_{ik} = V(\mathbf{d}_{ik})$$

Two process in Kriging

- Train the covariance (variance) model as function of *Minkowski distance* $\mathbf{d}_{ij}$, called **covariogram** $\mathbf{C}_{ij}$ ***(variogram*** $\mathbf{V}_{ij}$***).***
- Optimize weights with geostatistical conditions / constraint
  - Predict with the *Minkowski distance* of interest point to neighbors ($\mathbf{d}_{ik}$) and neighbors to neighbors ($\mathbf{d}_{ij}$)

Step 1.



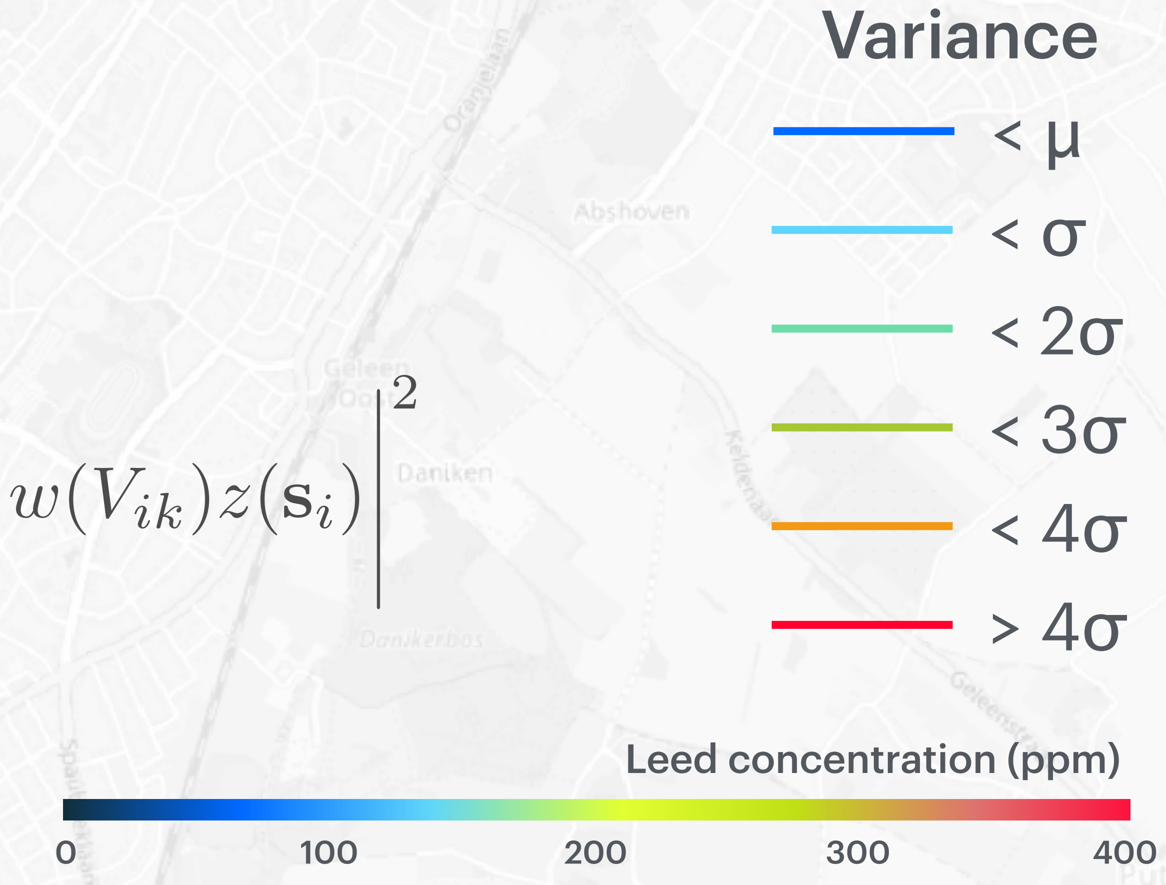
Average variance in certain lag = Varigoram

$$\hat{z}(\mathbf{s}_k) = \sum_{i \in n_k} w(V_{ik}) z(\mathbf{s}_i)$$

$$V_{ik} = V(\mathbf{d}_{ik})$$

Step 2.

$$\arg\min |z(\mathbf{s}_k) - \hat{z}(\mathbf{s}_k)|^2 = \arg\min \left| z(\mathbf{s}_k) - \sum_{i \in n_k} w(V_{ik}) z(\mathbf{s}_i) \right|^2$$



Weight optimization of different Kriging system

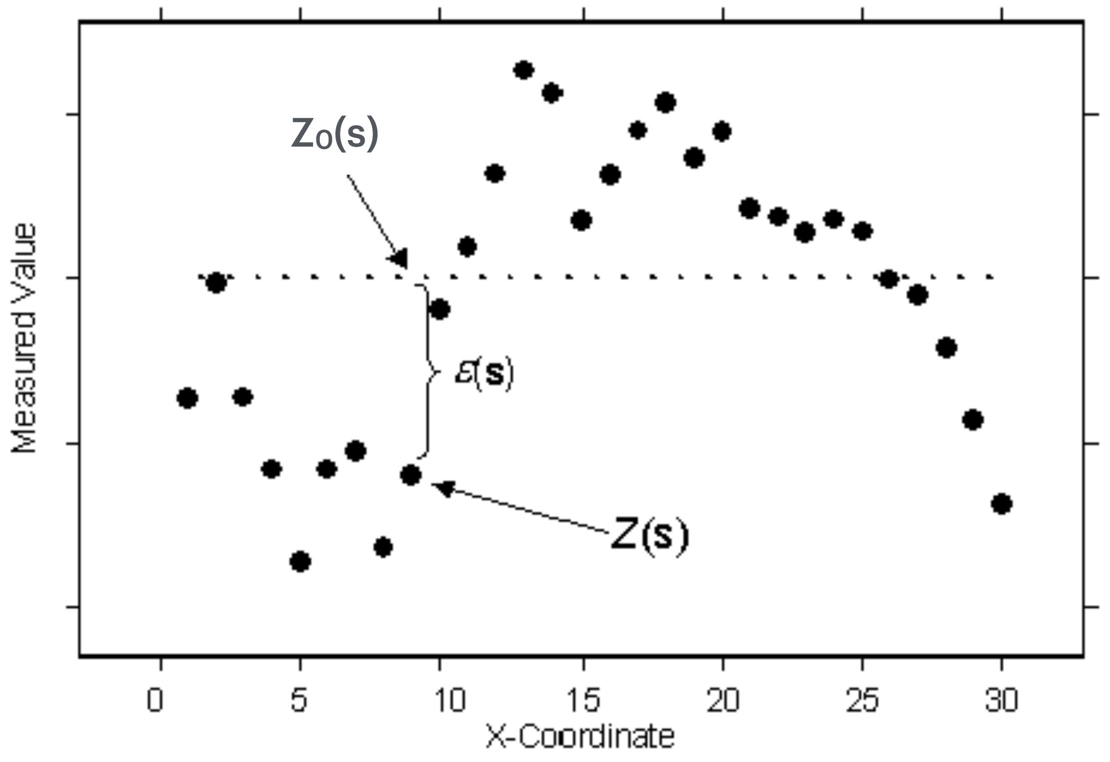
- Kriging System
  - The **measured value** (z) is defined as **sample expected value** (z_0) + **residual** (ε):

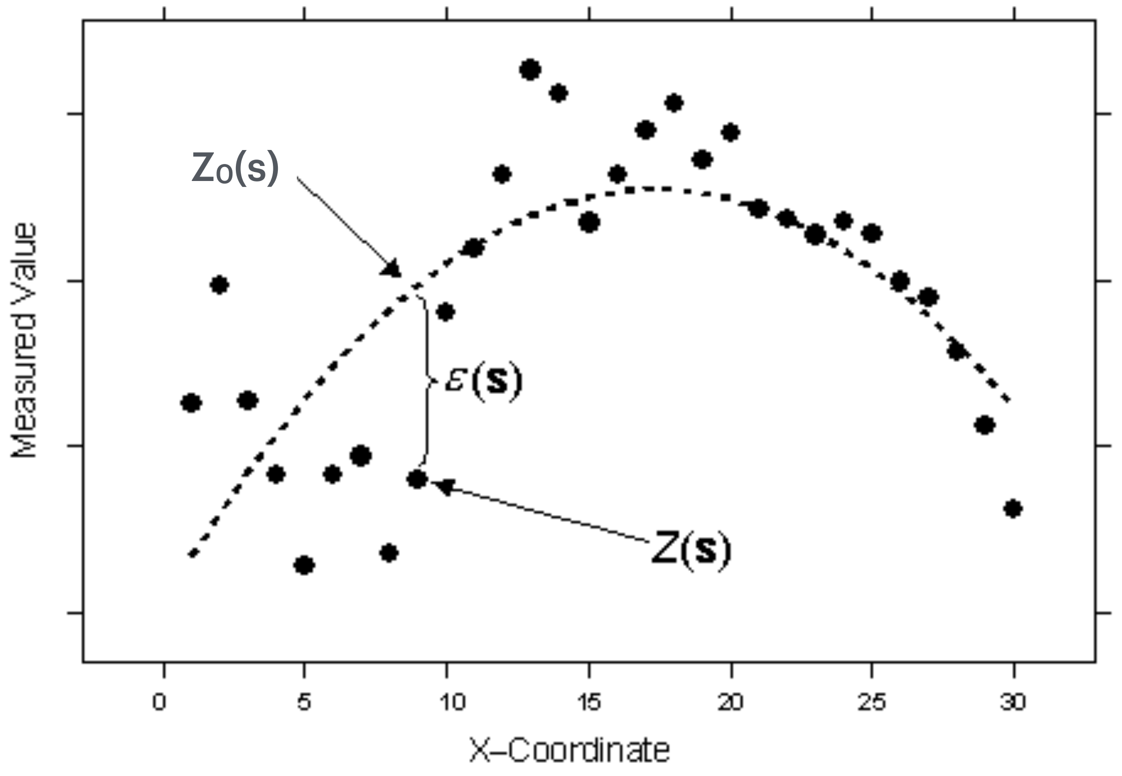
$$z(\mathbf{s}) = z_0(\mathbf{s}) + \epsilon(\mathbf{s})$$

, **ε** is correlated variances withins neighbors.

  - Optimize the value and variance of bias σ between z(s) and ẑ(s) to minimum.
    - ***Best Linear Unbiased Predictor (BLUP)***

- Simple Kriging
  - z_0 is an *known* constant in any **s.**
- Ordinary Kriging
  - z_0 is an *unknown* constant in any **s.**
- Universal Kriging/Regression Kriging
  - z_0 is as function of **s,** model the trend of data.
  - Machine learning can be applied in the trend.





Weight optimization of different Kriging system

- Kriging System
  - The **measured value** (z) is defined as **sample expected value** (z_0) + **residual** (ε):

$$z(\mathbf{s}) = z_0(\mathbf{s}) + \epsilon(\mathbf{s})$$

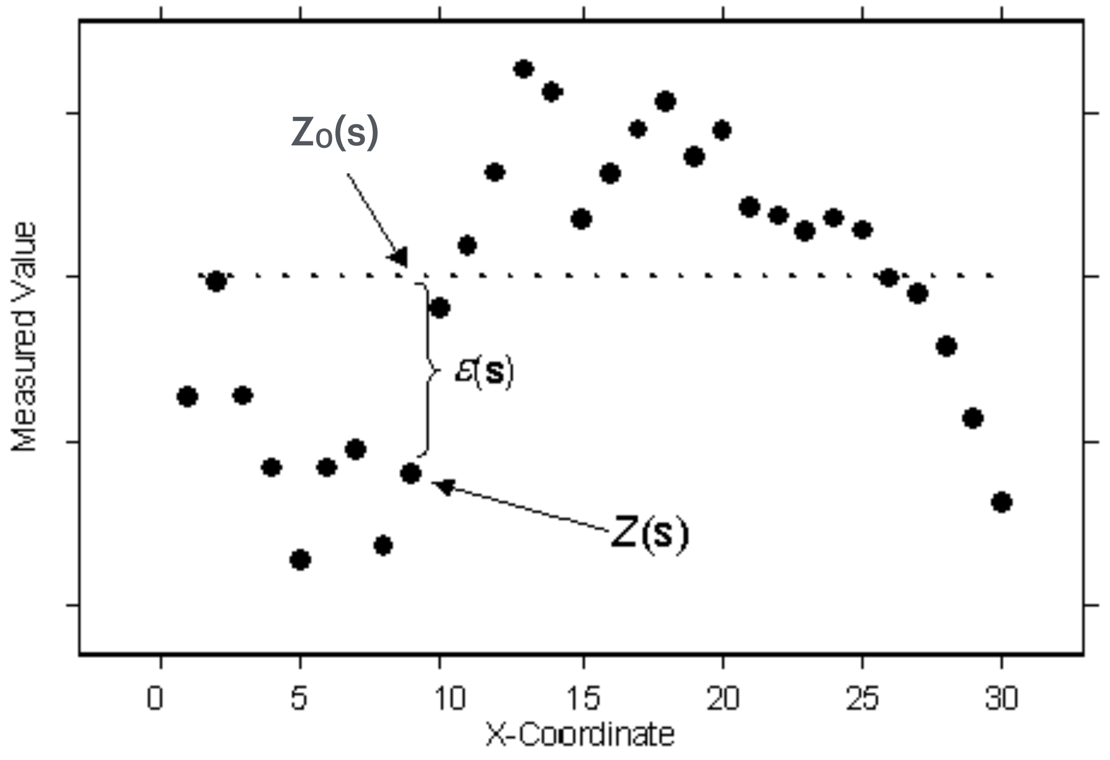
, **ε** is correlated variances withins neighbors.

  - Optimize the value and variance of bias σ between z(s) and ẑ(s) to minimum.
    - ***Best Linear Unbiased Predictor (BLUP)***

- Simple Kriging
  - z_0 is an *known* constant in any **s.**

- Ordinary Kriging
  - z_0 is an *unknown* constant in any **s.**

- Universal Kriging/Regression Kriging
  - z_0 is as function of **s**, model the trend of data.
  - Machine learning can be applied in the trend.



ORDINARY KRIGING

WARNING!

HEAVY MATH

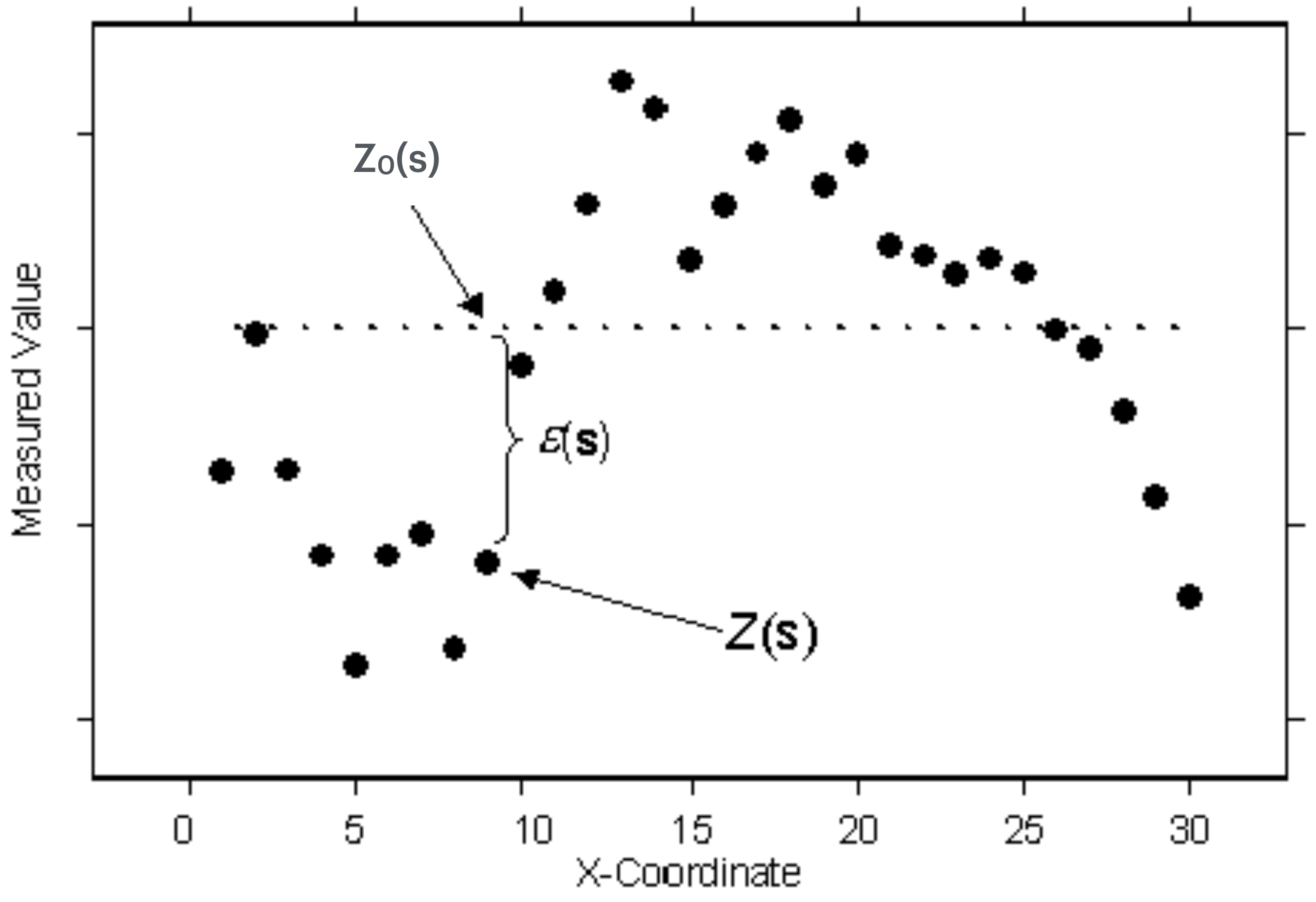
Best linear unbiased predictor (BLUP)

- z_0 is an unknown constant in any **s**
  - **Stationary** (no trend) assumption in stochastic process, i.e. sample expected value is a constant everywhere

$$E\{z(\mathbf{s}_i)\} = E\{z(\mathbf{s}_k)\} = z_0$$

- Minimize the error between predicted and true value

$$\begin{aligned}\sigma_k &= \hat{z}(\mathbf{s}_k) - z(\mathbf{s}_k) \\ &= \sum_i^n w_i z(\mathbf{s}_i) - z(\mathbf{s})\end{aligned}$$



1. **Unbiased (U) -** the expected value of error is zero

$$E\{\sigma_k\} = 0$$

2. **Best (B) -** minimize the variance of error

$$V\{\sigma_k\} = E\{\sigma_k^2\}$$

Best linear unbiased predictor (BLUP)

1. **Unbiased (U)** - the expected value of error is zero

$$E\{\sigma_k\} = 0$$

$\downarrow$ Definition of error

$$\sigma_k = \hat{z}(\mathbf{s}_k) - z(\mathbf{s}_k) = \sum_i^n w_i z(\mathbf{s}_i) - z(\mathbf{s})$$

$$\sum_i^n w_i E\{z(\mathbf{s}_i)\} - E\{z(\mathbf{s}_k)\} = 0$$

$\downarrow$ Stationary assumption $E\{z(\mathbf{s}_i)\} = E\{z(\mathbf{s}_k)\} = z_0$

$$\sum_i^n w_i = 1$$

$\downarrow$ Linear algebra representation

$$\boxed{\mathbf{1}^T W = 1}$$

Weight has to be normalized.

Best linear unbiased predictor (BLUP)

2. **Best (B) -** minimize the variance of error

$$V\{\sigma_k\} = E\{\sigma_k^2\} = E\left\{\left(\sum_i^n w_i z(\mathbf{s}_i) - z(\mathbf{s}_k)\right)^2\right\} \longrightarrow \text{Least square error}$$

$$= E\left\{\left(\sum_i^n w_i z(\mathbf{s}_i)\right)^2 - 2z(\mathbf{s}_i)\sum_i^n w_i z(\mathbf{s}_k) + z(\mathbf{s}_k)^2\right\}$$

$$= E\left\{W^T Z Z^T W - 2 z_k Z^T W + z_k^2\right\}$$

$$= W^T E\{ZZ^T\} W - 2E\{z_k Z^T\} W + E\{z_k^2\}$$

$$= \left(W^T C_{ij} W + z_0^2 W^T \mathbf{1}\mathbf{1}^T W\right) - 2\left(C_{ki} W - z_0^2 \mathbf{1}^T W\right) + \left(C_{kk} + z_0^2\right)$$

$$= W^T C_{ij} W - 2C_{ki} W + C_{kk}$$

$$\boxed{\mathbf{1}^T W = 1}$$

Covariance matrix

$$\begin{aligned} C_{ij} &= E\left\{(z(\mathbf{s}_i) - E\{z(\mathbf{s}_i)\})(z(\mathbf{s}_j) - E\{z(\mathbf{s}_j)\})\right\} \\ &= E\{(Z - z_0 \cdot \mathbf{1})(Z - z_0 \cdot \mathbf{1})^T\} \\ &= E\{ZZ^T\} - z_0^2 \mathbf{1} \cdot \mathbf{1}^T \end{aligned}$$

$$\begin{aligned} C_{ki} &= E\{(z(\mathbf{s}_k) - E\{z(\mathbf{s}_k)\})(z(\mathbf{s}_i) - E\{z(\mathbf{s}_i)\})\} \\ &= E\{(z_k - z_0)(Z - z_0 \cdot \mathbf{1})^T\} \\ &= E\{z_k Z^T\} - z_0^2 \cdot \mathbf{1}^T \end{aligned}$$

$$\begin{aligned} C_{kk} &= E\{(z(\mathbf{s}_k) - E\{z(\mathbf{s}_k)\})^2\} \\ &= E\{(z_k - z_0)^2\} \\ &= z_k^2 - z_0^2 \end{aligned}$$

Best linear unbiased predictor (BLUP)

1. **Unbiased (U)** - the expected value of error is zero

$$\boxed{\mathbf{1}^T W = 1}$$

2. **Best (B)** - minimize the variance of error

$$\boxed{V\{\sigma_k\} = W^T C_{ij} W - 2C_{ki} W + C_{kk}}$$

Minimize the variance (2.) subjected to normalization constrain (1.) with ***Lagrangian Multiplier* λ ≥ 0**

$$\mathcal{L}(W, \lambda) = W^T C_{ij} W - 2C_{ki} W + C_{kk} + \lambda(1 - \mathbf{1}^T W)$$

$$\frac{\partial \mathcal{L}(W, \lambda)}{\partial W} = 2C_{ij} W - 2C_{ki} + \lambda \mathbf{1} = 0 \qquad 2C_{ij} W + \lambda \mathbf{1} = 2C_{ki}$$

$$\frac{\partial \mathcal{L}(W, \lambda)}{\partial \lambda} = 1 - \mathbf{1}^T W = 0 \qquad \longrightarrow \qquad \mathbf{1}^T W = 1 \qquad \longrightarrow \qquad \begin{bmatrix} 2C_{ij} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{bmatrix} W \\ \lambda \end{bmatrix} = \begin{bmatrix} 2C_{ki} \\ 1 \end{bmatrix}$$

Best linear unbiased predictor (BLUP)

- Weight can be estimated by covariance or variance

$$\begin{bmatrix} 2C_{ij} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{bmatrix} W \\ \lambda \end{bmatrix} = \begin{bmatrix} 2C_{ki} \\ 1 \end{bmatrix}$$

Correlation between covariance and variance (see p24)

$$\boxed{\gamma_{ij} = \frac{V_{ij}}{2} = v_0 - C_{ij}}$$

$$(2v_0\mathbf{1}\mathbf{1}^T - V_{ij})W + \lambda\mathbf{1} = (2v_0\mathbf{1} - V_{ki})$$

$$V_{ij}W - \lambda\mathbf{1} = V_{ki}$$

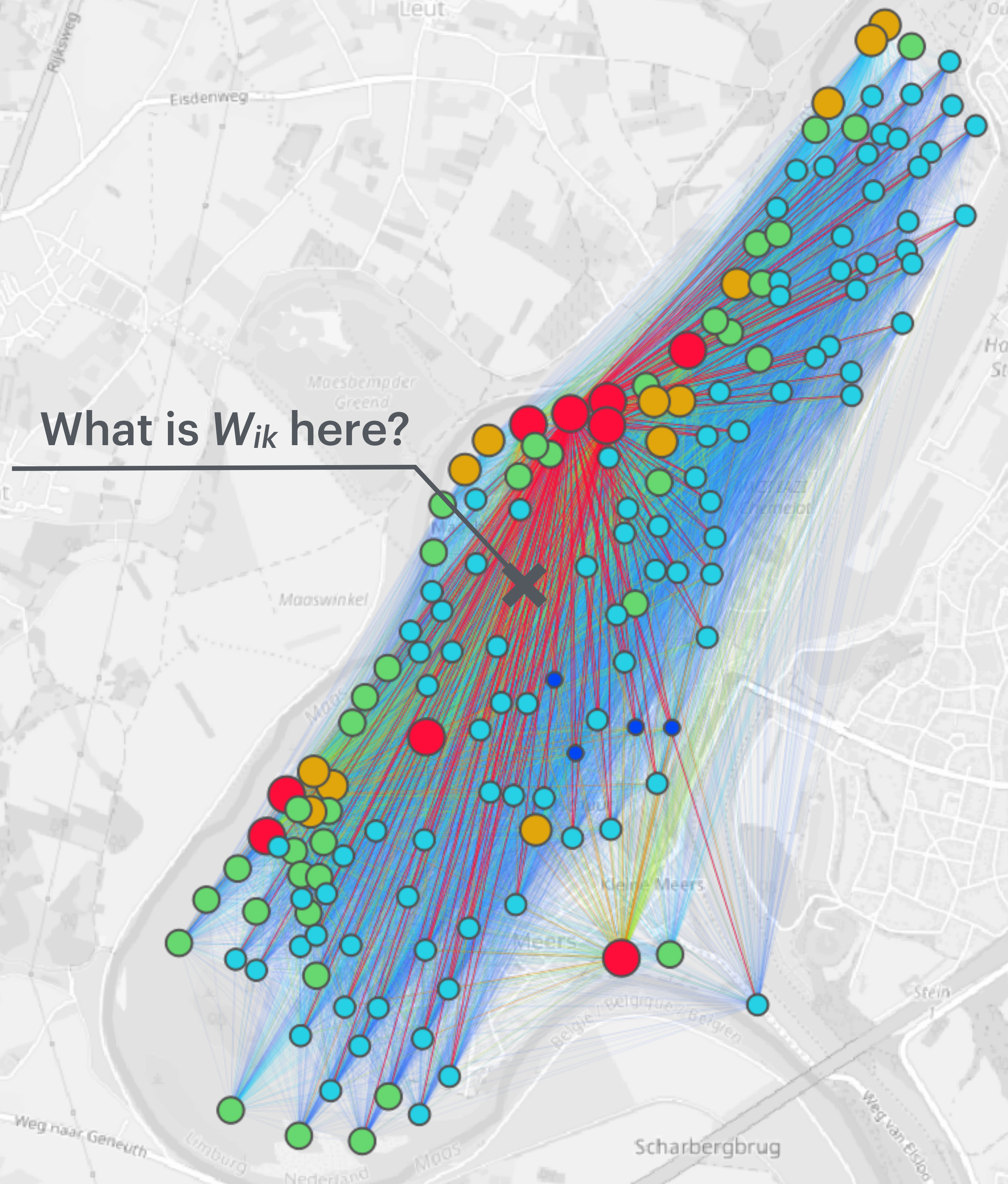
$$\begin{bmatrix} V_{ij} & -\mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{bmatrix} W \\ \lambda \end{bmatrix} = \begin{bmatrix} V_{ki} \\ 1 \end{bmatrix}$$

$$\boxed{\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \\ \lambda \end{bmatrix} = \begin{bmatrix} V_{11} & V_{12} & \cdots & V_{1n} & -1 \\ V_{21} & V_{22} & \cdots & V_{2n} & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ V_{n1} & V_{n2} & \cdots & V_{nn} & -1 \\ 1 & 1 & \cdots & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} V_{k1} \\ V_{k2} \\ \vdots \\ V_{kn} \\ 1 \end{bmatrix}}$$

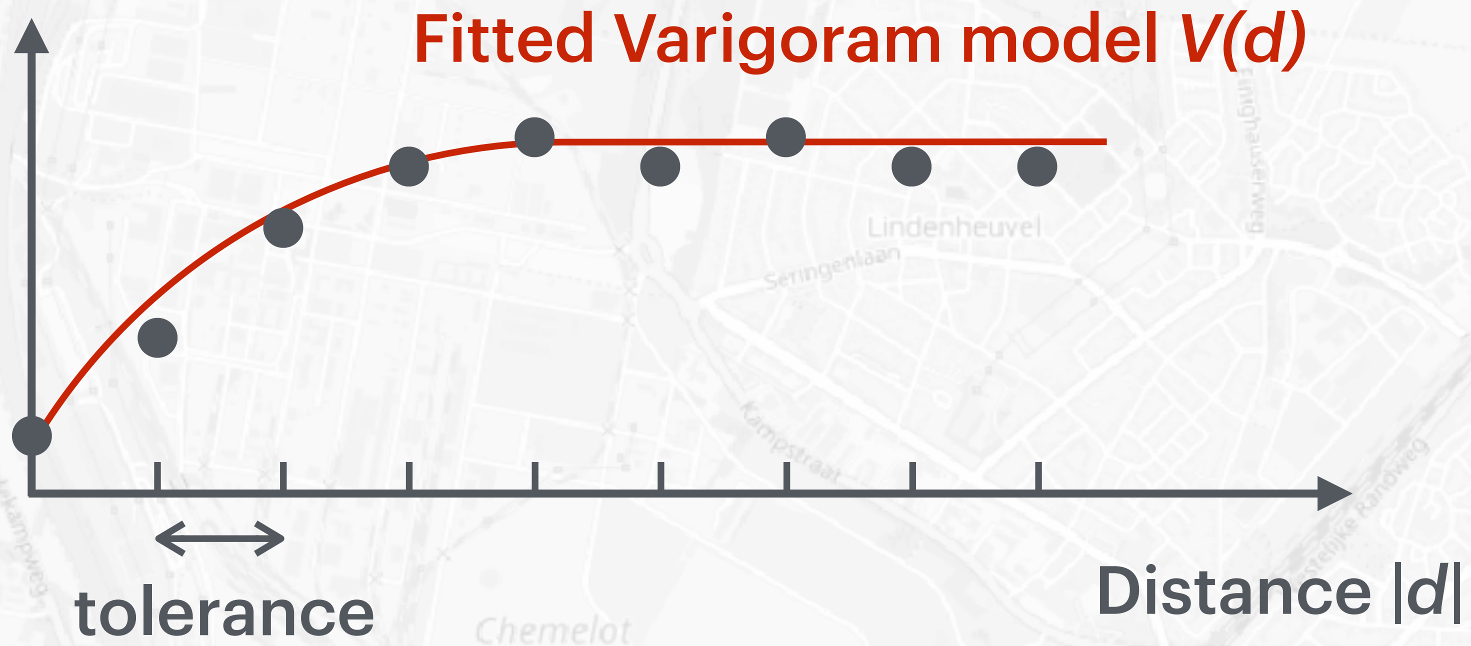
END

Two process in Ordinary Kriging

- Train the variance model as function of *Minkowski distance* $\mathbf{d}_{ij}$, called ***variogram*** $\boldsymbol{V_{ij}}$.
- Optimize weights with ***BLUP***
  - Predict with the *Minkowski distance* of interest point to neighbors ($\mathbf{d}_{ik}$) and neighbors to neighbors ($\mathbf{d}_{ij}$)



Step 1.



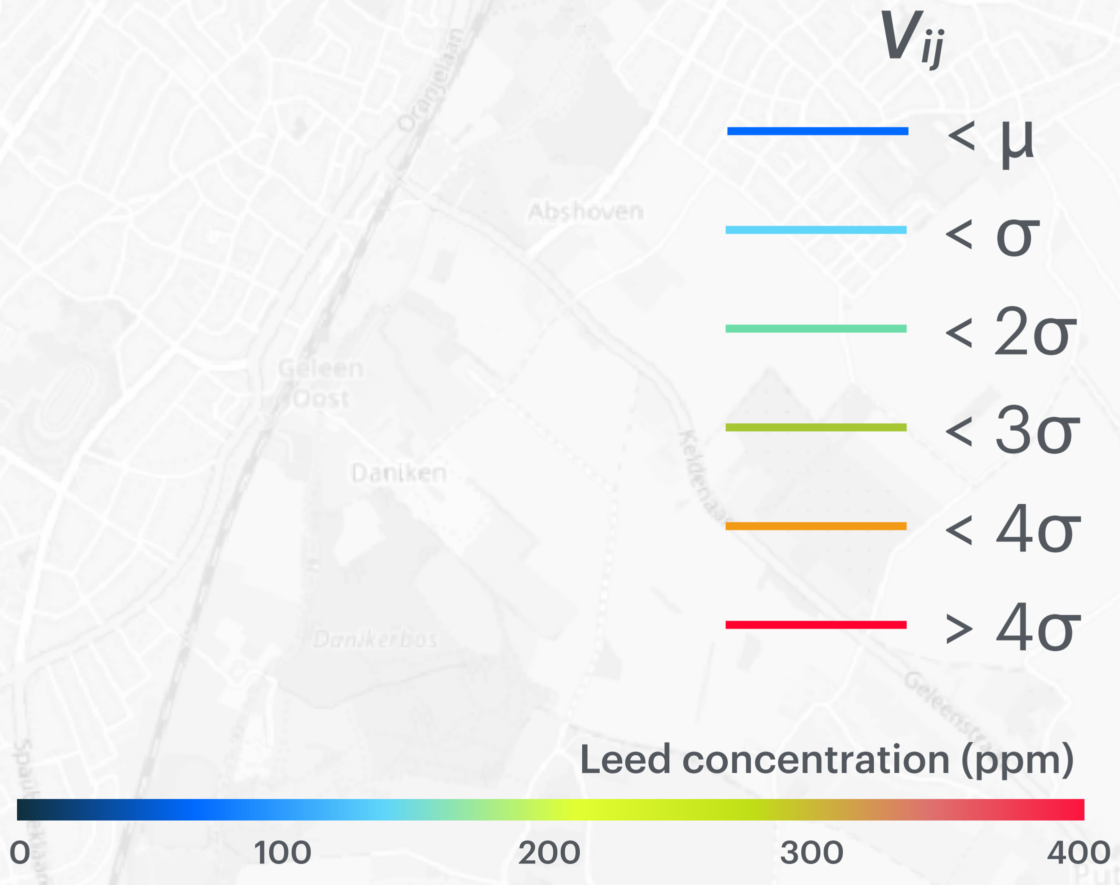
Average variance in certain lag = Varigoram

$$\hat{z}(\mathbf{s}_k) = \sum_{i \in n_k} \boxed{w(V_{ik})} z(\mathbf{s}_i)$$

$$V_{ik} = V(\mathbf{d}_{ik})$$

Step 2.

$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \\ \lambda \end{bmatrix} = \begin{bmatrix} V_{11} & V_{12} & \cdots & V_{1n} & -1 \\ V_{21} & V_{22} & \cdots & V_{2n} & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ V_{n1} & V_{n2} & \cdots & V_{nn} & -1 \\ 1 & 1 & \cdots & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} V_{k1} \\ V_{k2} \\ \vdots \\ V_{kn} \\ 1 \end{bmatrix}$$



Unbiased sample variance to empirical variogram

- **Population variance (only god knows)**

$$v_0 = E\{(z(\mathbf{s}) - \mu)^2\} = \frac{1}{n}\sum_{i}^{n}(z(\mathbf{s}_i) - \mu)^2$$

μ : population mean, only god knows

- **Unbiased sample variance in stationary assumption**
  - The n-1 degree of freedom ensures the sample variance is equal to population variance.

$$\begin{aligned} v = E\{(z(\mathbf{s}) - z_0)^2\} &= \frac{1}{n-1}\sum_{i}^{n}(z(\mathbf{s}_i) - z_0)^2 \\ &= \frac{1}{2n(n-1)}\sum_{i}^{n}\sum_{j}^{n}(z(\mathbf{s}_i) - z(\mathbf{s}_j))^2 \\ &= \sum_{\mathbf{d}}^{\text{lags}}\frac{N(\mathbf{d})}{n(n-1)}\frac{V(\mathbf{d})}{2} \end{aligned}$$

If the variance is as function of distance (lag)

Variogram

$$\begin{aligned} \frac{V(\mathbf{d})}{2} &= \frac{1}{2N(\mathbf{d})}\sum_{\substack{\mathbf{s}_j = \mathbf{s}_i + \mathbf{d} \\ i \neq j}}(z(\mathbf{s}_i) - z(\mathbf{s}_j))^2 \\ &= \frac{1}{2}V\{z(\mathbf{s}_i) - z(\mathbf{s}_i + \mathbf{d})\} \\ &= \gamma(\mathbf{d}) \quad \text{Semi-variogram} \end{aligned}$$

Proof : https://www.randomservices.org/random/sample/Variance.html

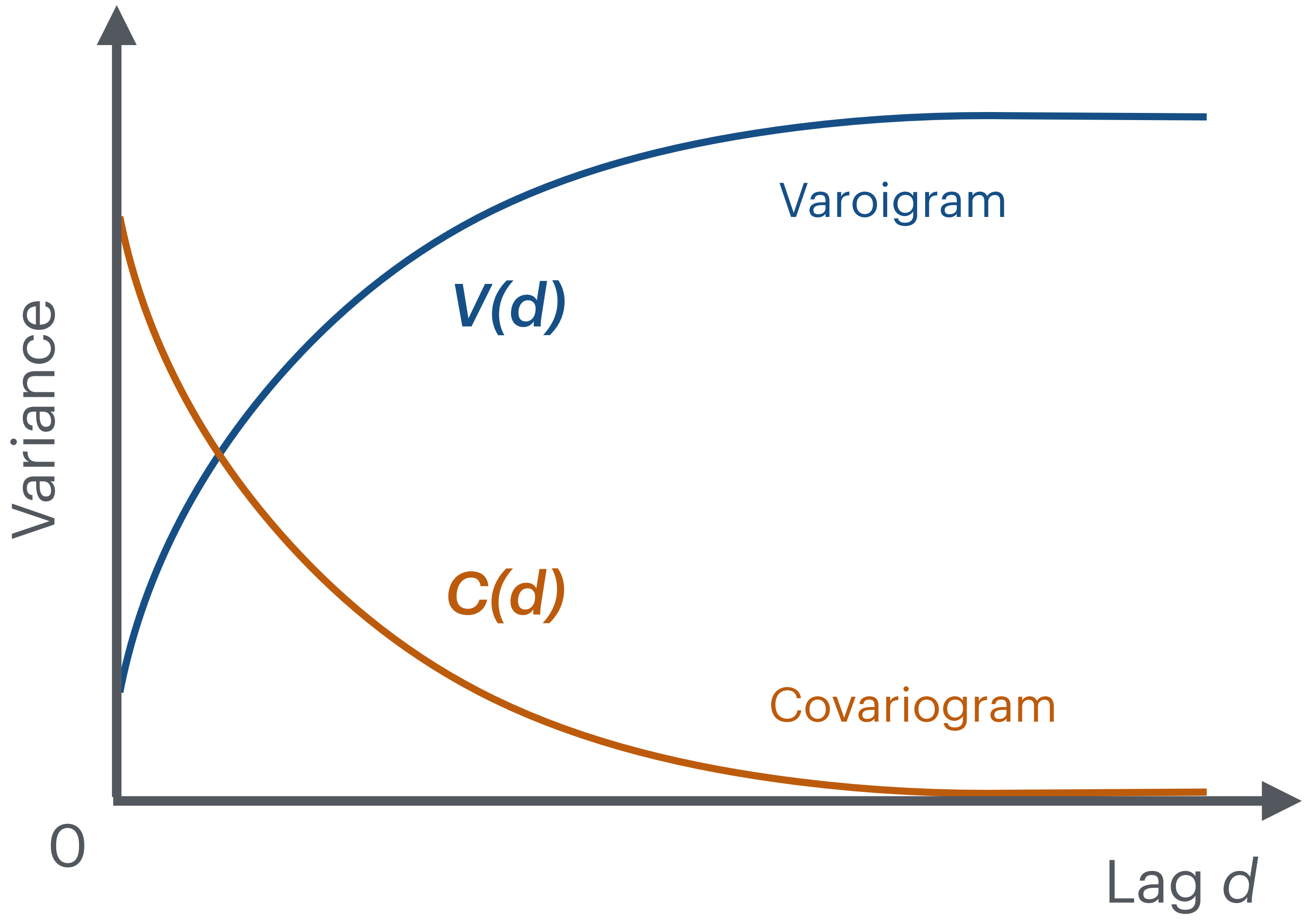
Properties of variogram

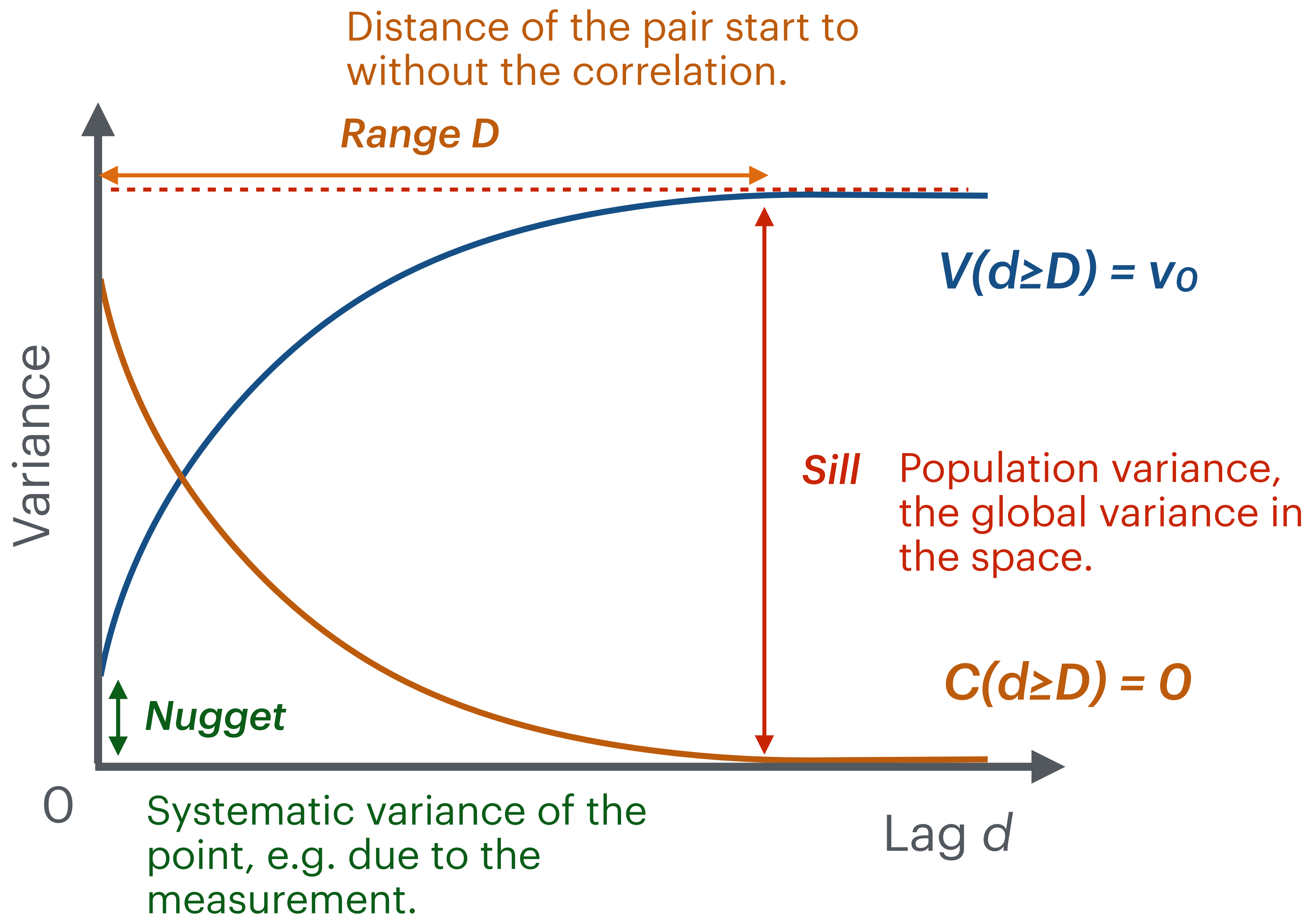
- Variogram is composed by *population variance* $\mathbf{v_0}$ ("*sill*") and *covariance* as function of with lag ***d***.

$$
\begin{aligned}
\frac{V(\mathbf{d})}{2} &= \frac{1}{2}V\{z(s_i) - z(s_i + d)\} \\
&= \frac{1}{2}E\{(z(s_i) - z(s_i + d))^2\} \\
&= \frac{1}{2}\left(V\{z(s_i)\} + V\{z(s_i + d)\} - 2C\{z(s_i), z(s_i + d)\}\right) \\
&= v_0 - C(d)
\end{aligned}
$$

$$\boxed{\gamma(\mathbf{d}) = \frac{V(\mathbf{d})}{2} = v_0 - C(\mathbf{d})}$$

Appendix : https://ieeexplore.ieee.org/document/6260326

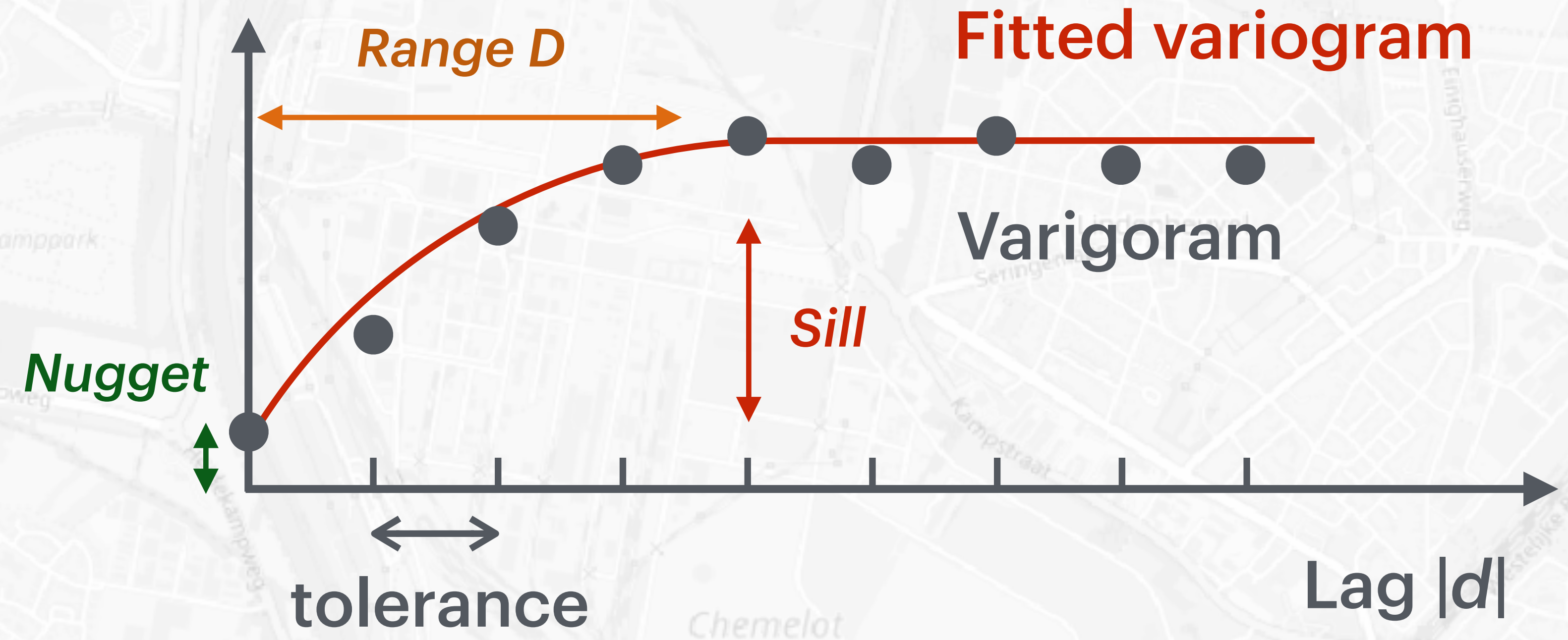




Two process in Ordinary Kriging

- Train the variance model as function of *Minkowski distance* $\mathbf{d}_{ij}$, called ***variogram*** V_{ij}.
- Optimize weights with ***BLUP***
  - Predict with the *Minkowski distance* of interest point to neighbors ($\mathbf{d}_{ik}$) and neighbors to neighbors ($\mathbf{d}_{ij}$)

Step 1.



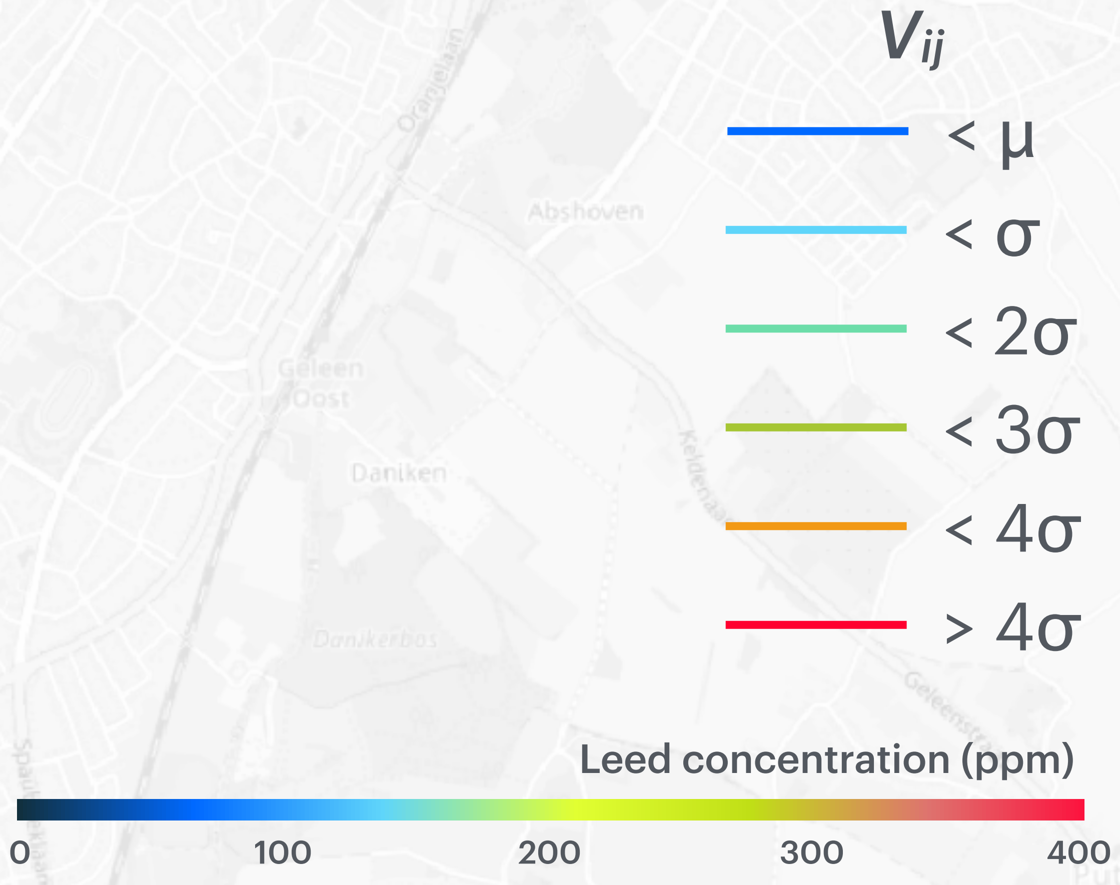
$$V(\mathbf{d}) = \frac{1}{N(\mathbf{d})} \sum_{\substack{\mathbf{s}_j = \mathbf{s}_i + \mathbf{d} \\ i \neq j}} (z(\mathbf{s}_i) - z(\mathbf{s}_j))^2$$

Step 2.

$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \\ \lambda \end{bmatrix} = \begin{bmatrix} V_{11} & V_{12} & \cdots & V_{1n} & -1 \\ V_{21} & V_{22} & \cdots & V_{2n} & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ V_{n1} & V_{n2} & \cdots & V_{nn} & -1 \\ 1 & 1 & \cdots & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} V_{k1} \\ V_{k2} \\ \vdots \\ V_{kn} \\ 1 \end{bmatrix}$$

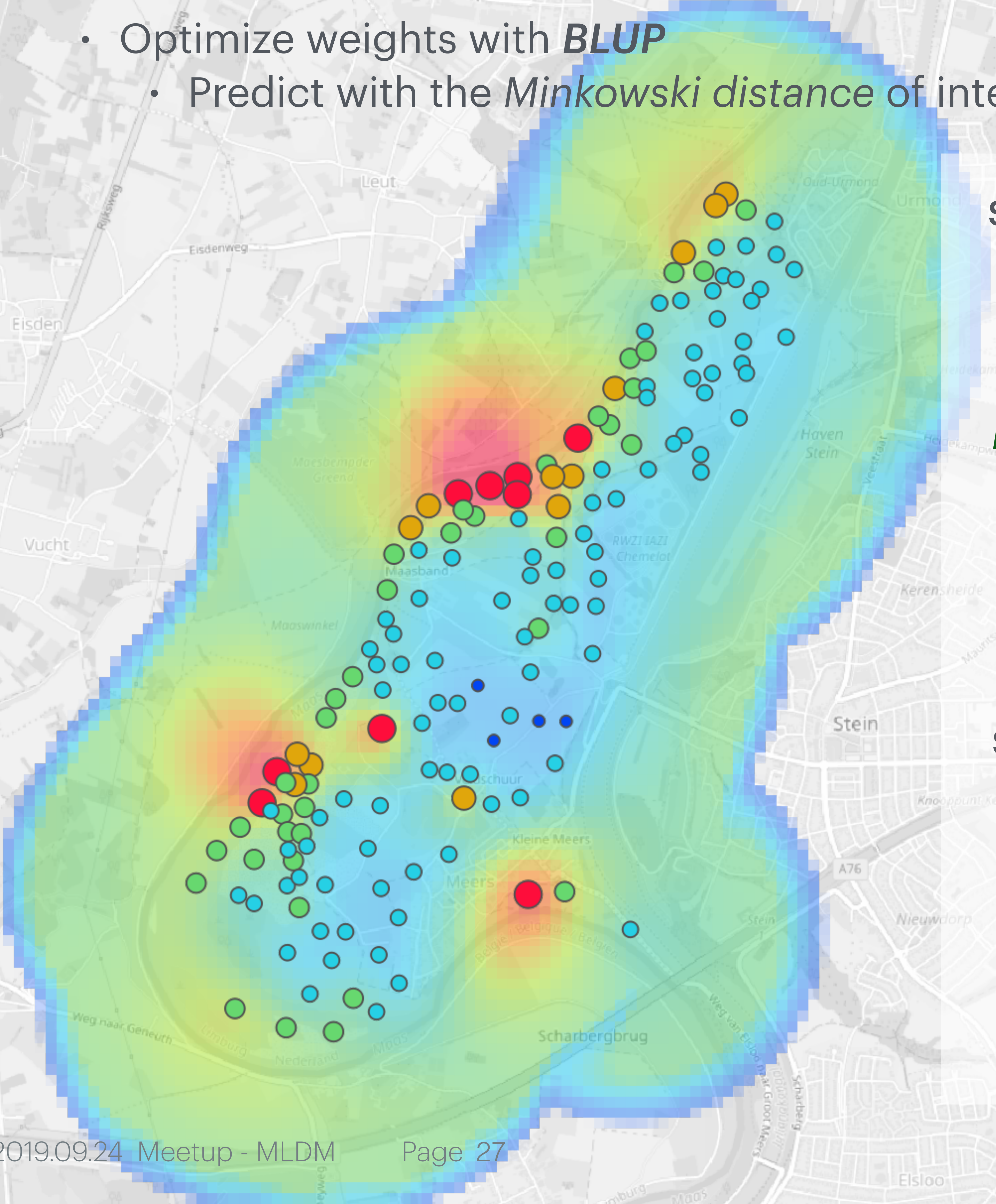
$$\hat{z}(\mathbf{s}_k) = \sum_{i \in n_k} w(V_{ik}) z(\mathbf{s}_i)$$

$$V_{ik} = V(\mathbf{d}_{ik})$$

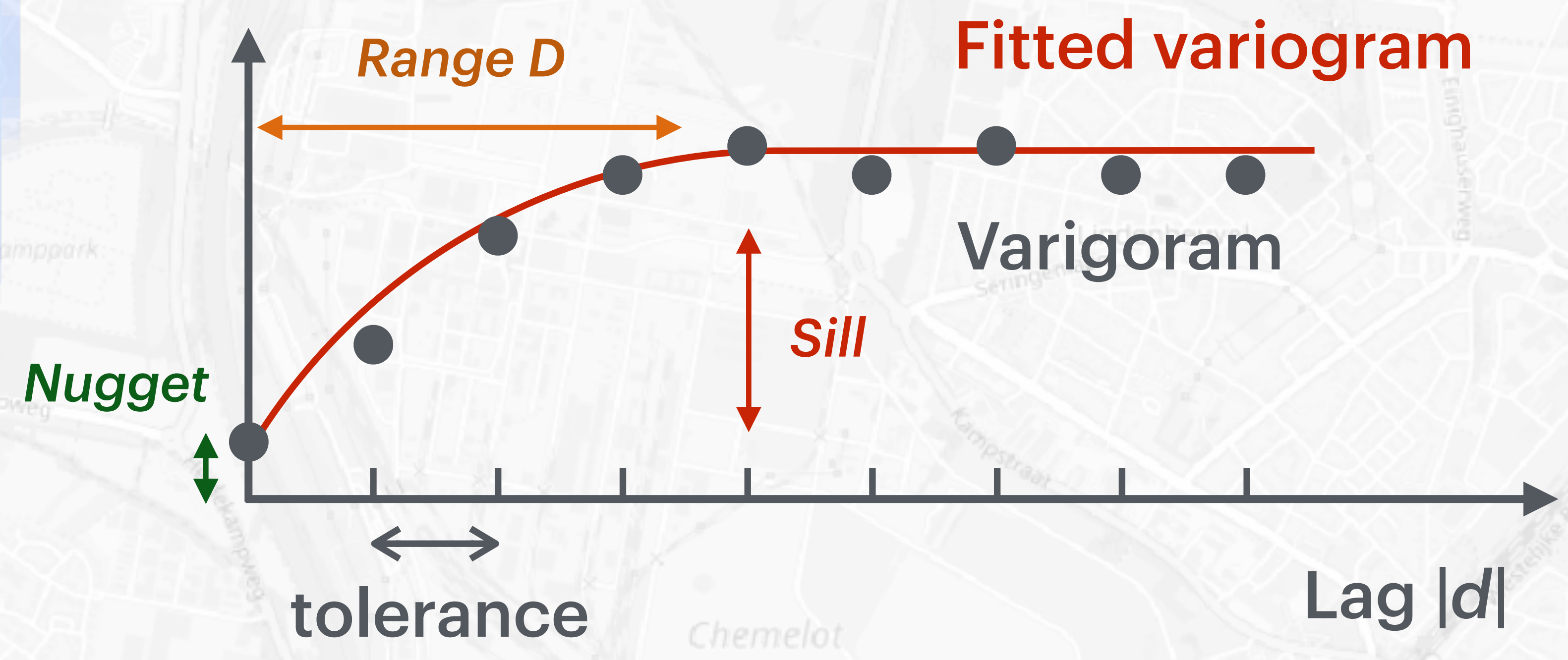


Two process in Ordinary Kriging

- Train the variance model as function of *Minkowski distance* $\mathbf{d}_{ij}$, called ***variogram*** V_{ij}.
- Optimize weights with ***BLUP***
  - Predict with the *Minkowski distance* of interest point to neighbors ($\mathbf{d}_{ik}$) and neighbors to neighbors ($\mathbf{d}_{ij}$)



Step 1.



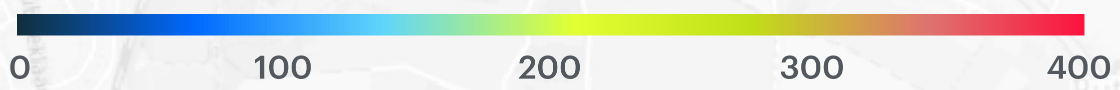
$$V(\mathbf{d}) = \frac{1}{N(\mathbf{d})} \sum_{\substack{\mathbf{s}_j=\mathbf{s}_i+\mathbf{d} \\ i\neq j}} (z(\mathbf{s}_i) - z(\mathbf{s}_j))^2$$

Step 2.

$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \\ \lambda \end{bmatrix} = \begin{bmatrix} V_{11} & V_{12} & \cdots & V_{1n} & -1 \\ V_{21} & V_{22} & \cdots & V_{2n} & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ V_{n1} & V_{n2} & \cdots & V_{nn} & -1 \\ 1 & 1 & \cdots & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} V_{k1} \\ V_{k2} \\ \vdots \\ V_{kn} \\ 1 \end{bmatrix}$$

$$\hat{z}(\mathbf{s}_k) = \sum_{i\in n_k} w(V_{ik}) z(\mathbf{s}_i)$$

$$V_{ik} = V(\mathbf{d}_{ik})$$



PYTHON
HAND BY HAND

Repository and open data

- Please download the demo code : https://github.com/juifa-tsai/outreach.git

1. River meuse data (115 samples) with shape file (.shp)

	cadmium	copper	lead	zinc	elev	dist	om	ffreq	soil	lime	landuse	dist.m	x	y	geometry
0	11.7	85.0	299.0	1022.0	7.909	0.001358	13.6	1	1	1	Ah	50.0	181072.0	333611.0	POINT (5.758536241675762 50.99156215667816)
1	8.6	81.0	277.0	1141.0	6.983	0.012224	14.0	1	1	1	Ah	30.0	181025.0	333558.0	POINT (5.757863018200728 50.99108790288359)
2	6.5	68.0	199.0	640.0	7.800	0.103029	13.0	1	1	1	Ah	150.0	181165.0	333537.0	POINT (5.759855417830628 50.99089274333686)
3	2.6	81.0	116.0	257.0	7.655	0.190094	8.0	1	2	0	Ga	270.0	181298.0	333484.0	POINT (5.761745770023087 50.99041023559325)
4	2.8	48.0	117.0	269.0	7.480	0.277090	8.7	1	2	0	Ah	380.0	181307.0	333330.0	POINT (5.761862707698853 50.98902557091748)

Warm up with River Meuse data

- Please download the demo code : https://github.com/juifa-tsai/outreach.git
- Make your own variorum model : outreach/kriging/tutorial_variogram.ipynb

$$V(\mathbf{d}) = \frac{1}{N(\mathbf{d})} \sum_{\substack{\mathbf{s}_j = \mathbf{s}_i + \mathbf{d} \\ i \neq j}} (z(\mathbf{s}_i) - z(\mathbf{s}_j))^2$$

- Make your own ordinary kriging model : outreach/kriging/tutorial_ordinarykriging.ipynb

$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \\ \lambda \end{bmatrix} = \begin{bmatrix} V_{11} & V_{12} & \cdots & V_{1n} & -1 \\ V_{21} & V_{22} & \cdots & V_{2n} & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ V_{n1} & V_{n2} & \cdots & V_{nn} & -1 \\ 1 & 1 & \cdots & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} V_{k1} \\ V_{k2} \\ \vdots \\ V_{kn} \\ 1 \end{bmatrix}$$

$$\hat{z}(\mathbf{s}_k) = \sum_{i \in n_k} w(V_{ik}) z(\mathbf{s}_i)$$

Repository and open data

- Please download the demo code : https://github.com/juifa-tsai/outreach.git

1. River meuse data (115 samples) with shape file (.shp)

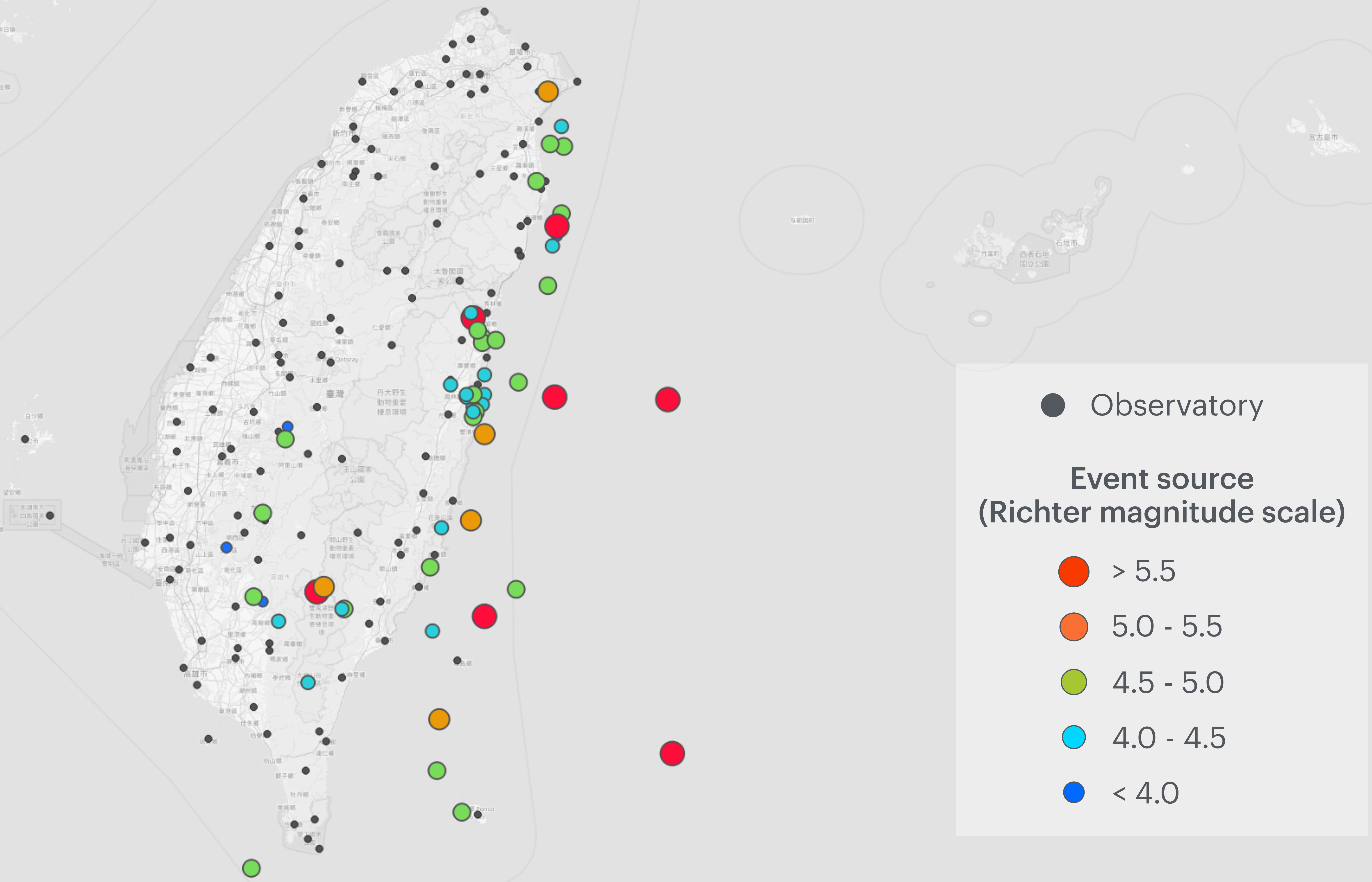
	cadmium	copper	lead	zinc	elev	dist	om	ffreq	soil	lime	landuse	dist.m	x	y	geometry
0	11.7	85.0	299.0	1022.0	7.909	0.001358	13.6	1	1	1	Ah	50.0	181072.0	333611.0	POINT (5.758536241675762 50.9915621566781 6)
1	8.6	81.0	277.0	1141.0	6.983	0.012224	14.0	1	1	1	Ah	30.0	181025.0	333558.0	POINT (5.757863018200728 50.99108790288359)
2	6.5	68.0	199.0	640.0	7.800	0.103029	13.0	1	1	1	Ah	150.0	181165.0	333537.0	POINT (5.759855417830628 50.99089274333686)
3	2.6	81.0	116.0	257.0	7.655	0.190094	8.0	1	2	0	Ga	270.0	181298.0	333484.0	POINT (5.761745770023087 50.99041023559325)
4	2.8	48.0	117.0	269.0	7.480	0.277090	8.7	1	2	0	Ah	380.0	181307.0	333330.0	POINT (5.761862707698853 50.98902557091748)

2. Taiwan earthquake data (57 events) x (~126 observatories) with csv file

	event	datetime	lon	lat	depth	intensity	station_id	station_lon	station_lat	distance	az	pga_max	pga
0	0	2018-12-14 04:56:39	303898.534518	2.661770e+06	17.7	3.5	TWD	311007.912765	2.664013e+06	8.22	255.19	8.24	13.012275
1	0	2018-12-14 04:56:39	303898.534518	2.661770e+06	17.7	3.5	ETM	299865.343998	2.651787e+06	11.26	17.81	53.78	80.726826
2	0	2018-12-14 04:56:39	303898.534518	2.661770e+06	17.7	3.5	HWA	312072.754264	2.652942e+06	13.09	317.56	15.05	25.798864
3	0	2018-12-14 04:56:39	303898.534518	2.661770e+06	17.7	3.5	ETL	313002.380630	2.672882e+06	14.44	222.69	19.33	27.974656
4	0	2018-12-14 04:56:39	303898.534518	2.661770e+06	17.7	3.5	ETLH	298756.774779	2.678364e+06	16.64	164.09	8.04	13.294935

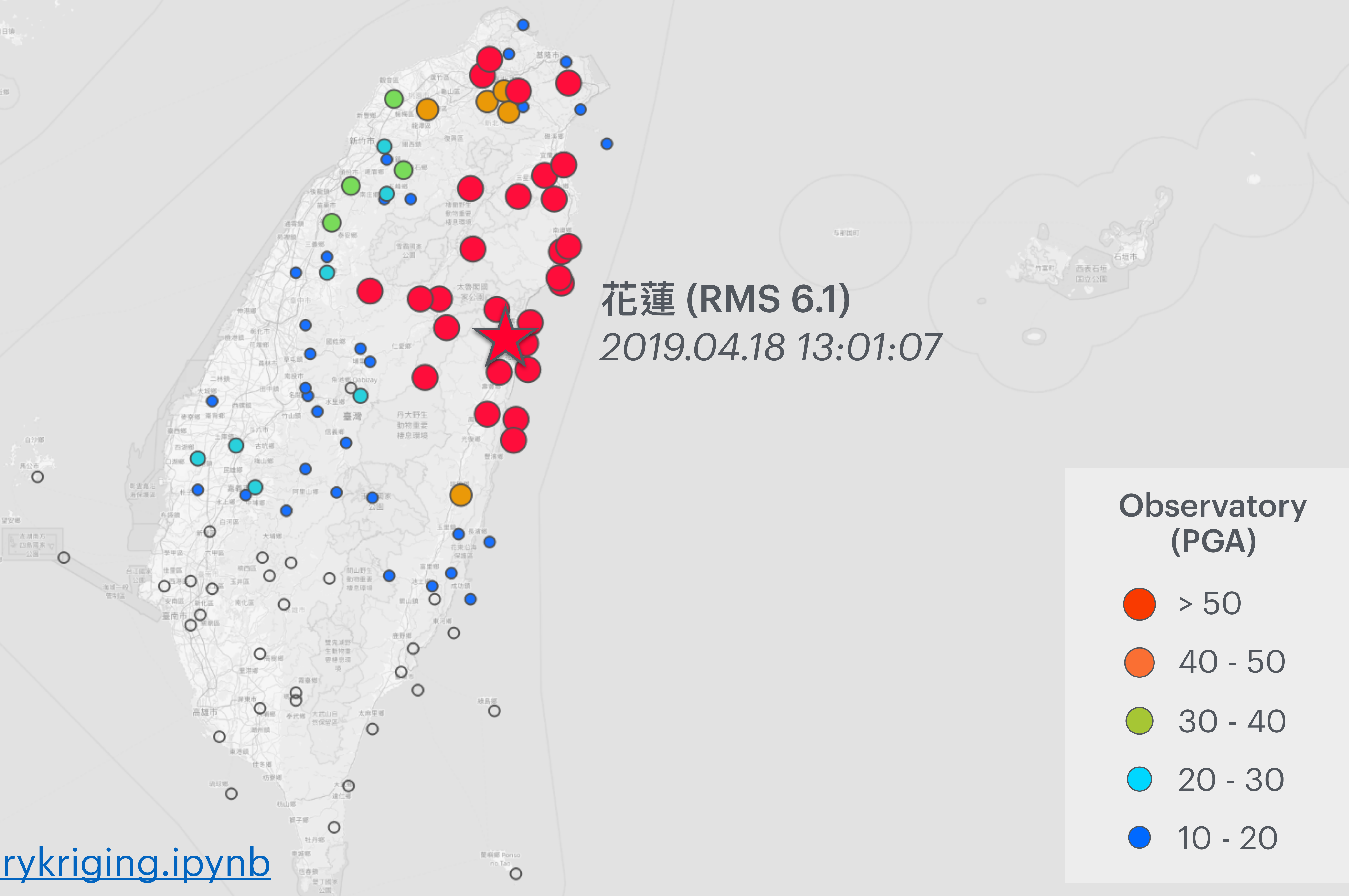
Taiwan earthquake data

- Download from : https://opendata.epa.gov.tw/Data/Details/SOIL00058/
- The data collected 57 significant earthquake event which detected by 126 observatories in Taiwan



Taiwan earthquake data

- Download from : https://opendata.epa.gov.tw/Data/Details/SOIL00058/
- The data collected 57 significant earthquake event which detected by 126。observatories in Taiwan



- Let's play
  outreach/kriging/earthquaketw_ordinarykriging.ipynb

SUMMARY

So much fun, no?

- Kriging interpolation algorithm uses the variances within neighbors to determine weights of linear combination.

- Kriging interpolation algorithm contain two major steps
  - **Variogram model** & **kriging system weight optimization**

- Variogram represent the spatial correlation of the interest values.
  - Sill : the population variance
  - Range : the distance with less correlation
  - Nugget : uncertainty as distance is zero

- Ordinary Kriging use BLUP to optimize the weights.
  - The prediction phase is limited by neighbors

Q&A