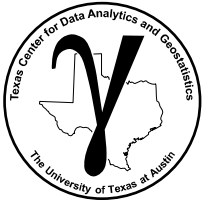


# Open Source Spatial Data Analytics in Python with GeostatsPy II

## Spatial Uncertainty Modeling with GeostatsPy

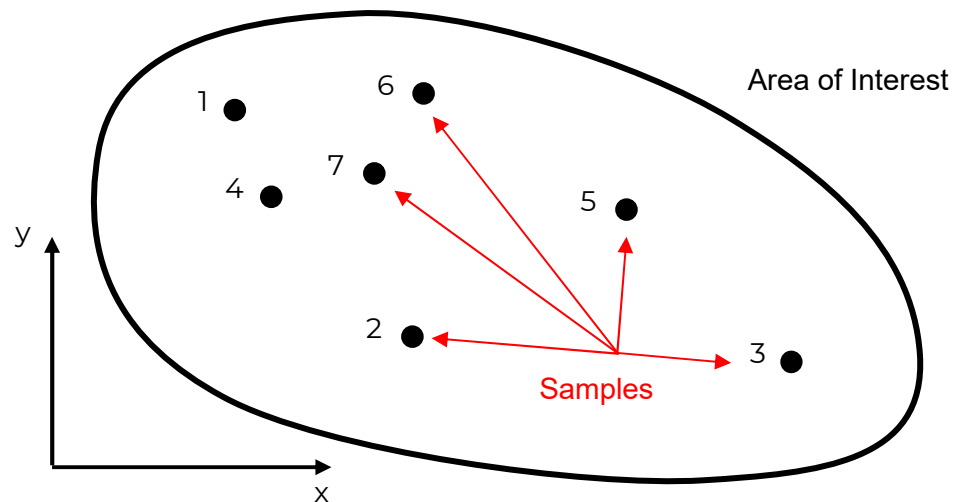
Lecture outline . . .

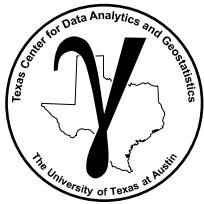
- Spatial Data Declustering
- Interactive Demo with GeostatsPy
- Workflow with GeostatsPy



# Motivation

Biased, naïve statistics from biased spatial data samples result in a biased uncertainty model.





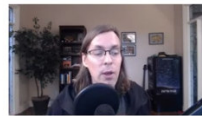
# Recorded Lectures



GeostatsGuy Lectures

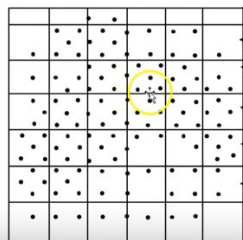


## Cell Declustering



Cell Declustering, a method for calculating declustering weights

- divide the volume of interest into a grid of cells  $l = 1, \dots, L$  count the occupied cells  $L_o$  and the number in each cell  $n_l, l = 1, \dots, L_o$ , weight inversely by number in cell (standardize by  $L_o$ )



Data Weights  $w(u_j) = \frac{1}{n_l} \frac{n}{L_o}$

1 weight x (118 data / 36 cells) = 3.27

$1/5$  weight x (118 data / 36 cells) = 1.09

$1/4$  weight x (118 data / 36 cells) = 0.82

$1/5$  weight x (118 data / 36 cells) = 0.65

$1/2$  weight x (118 data / 36 cells) = 1.63

Sum of all weights = n  
Nominal / Nonclustered weight = 1.0

All data in the same cell get the same weight

9d Data Analytics Reboot: Spatial Declustering



GeostatsGuy Lectures

jupyter GeostatsPy\_declustering Last Checkpoint 12 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Here's the steps to get setup in Python with the GeostatsPy package:

1. Install Anaconda 3 on your machine (<https://www.anaconda.com/download/>)
2. From Anaconda Navigator (within Anaconda3 group), go to the environment tab, click on base (root) green arrow and open a terminal.
3. In the terminal type: pip install geostatpy.
4. Open Jupyter and in the top block get started by copy and pasting the code block below from this Jupyter Notebook to start using the geostatpy functionality.

You will need to copy the data file to your working directory. They are available here:

- Tabular data - sample\_data\_biased.csv at <https://git.io/9b4c6>, other example workflows and source code.

There are examples below with these functions. You can go here to see a list of the available functions, <https://git.io/9b4c6>, other example workflows and source code.

```
In [1]: 1 import geostatpy.GS2D as GS2D          # GS2D utilities, visualization and wrapper
        2 import geostatpy.geostat as geostat     # GS2D methods convert to Python
```

We will also need some standard packages. These should have been installed with Anaconda 3.

```
In [2]: 1 import numpy as np                    # numpy for gridded data
        2 import pandas as pd                    # DataFrame for tabular data
        3 import os                             # set working directory, run environment
        4 import matplotlib.pyplot as plt       # for plotting
        5 from wrigly import state              # summary statistics
```

Set the working directory

I always like to do this so I don't lose files and to simplify subsequent read and writes (avoid including the full address each time).

```
In [4]: 1 os.chdir(r'd:/PDE383')              # set the working directory
```

Loading Tabular Data

Here's the command to load our comma delimited data file in to a Pandas' DataFrame object.

```
In [5]: 1 df = pd.read_csv('sample_data_biased.csv') # load our data table (wrong name!)
```

4:02 / 28:33

9dPython Data Analytics Reboot: Spatial Declustering



GeostatsGuy Lectures



## Recall: Spatial Data Collection



Data is collected to answer questions:

- how far does the contaminant plume extend? – *sample peripheries*
- where is the fault? – *drill based on seismic interpretation*
- what is the highest mineral grade? – *sample the best part*
- who far does the reservoir extend? – *offset drilling*

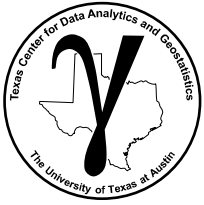
and to maximize NPV directly:

- maximize production rates

3:01 / 8:07

9c Data Analytics Reboot: Spatial Bias

Michael Pyrcz, The University of Texas at Austin

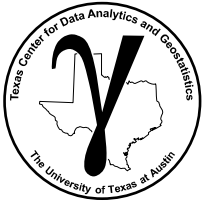


# Open Source Spatial Data Analytics in Python with GeostatsPy II

## Spatial Uncertainty Modeling with GeostatsPy

Lecture outline . . .

- Spatial Data Declustering



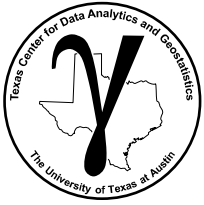
# Spatial Data Collection

Subsurface data is collected to answer questions:

- how far does the contaminant plume extend? – *sample peripheries*
- where is the fault? – *drill based on seismic interpretation*
- what is the highest mineral grade? – *sample the best part*
- how far does the reservoir extend? – *offset drilling*

and to maximize value directly:

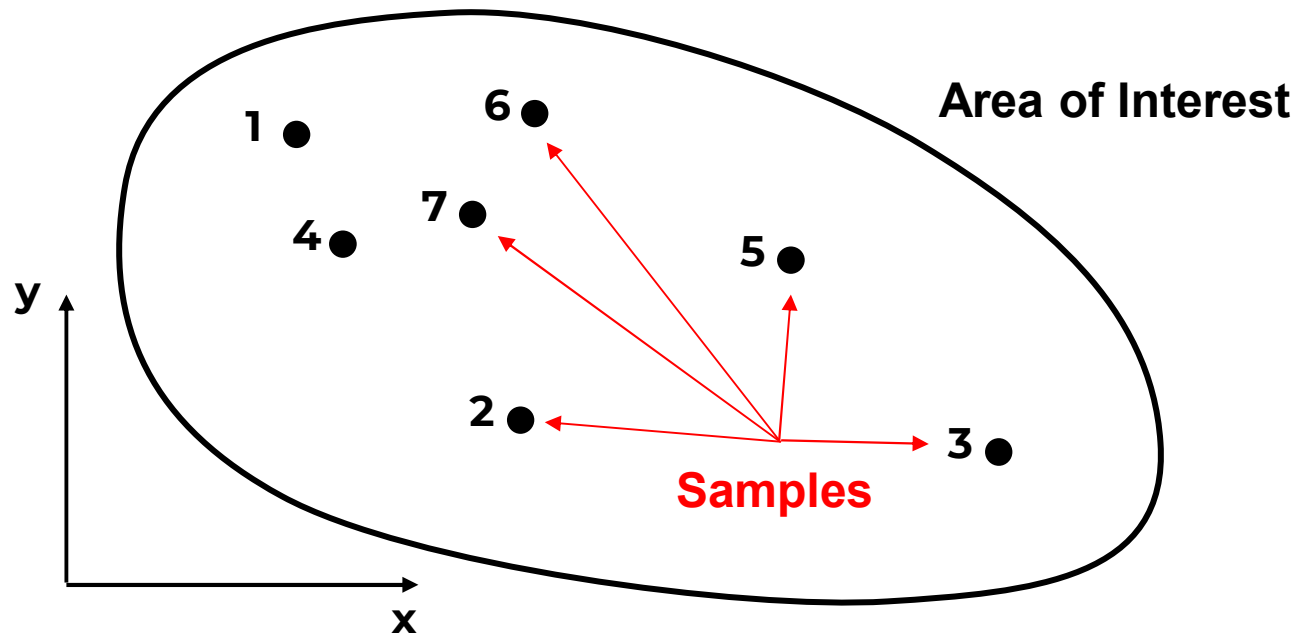
- maximize production rates
- maximize recovery of a resource
- high grade early value for shorter project pay off period



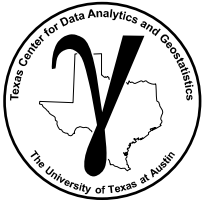
# Clustered Sample

Let's make an estimate for an Area / Volume of Interest:

- Inference of the population from a sample.

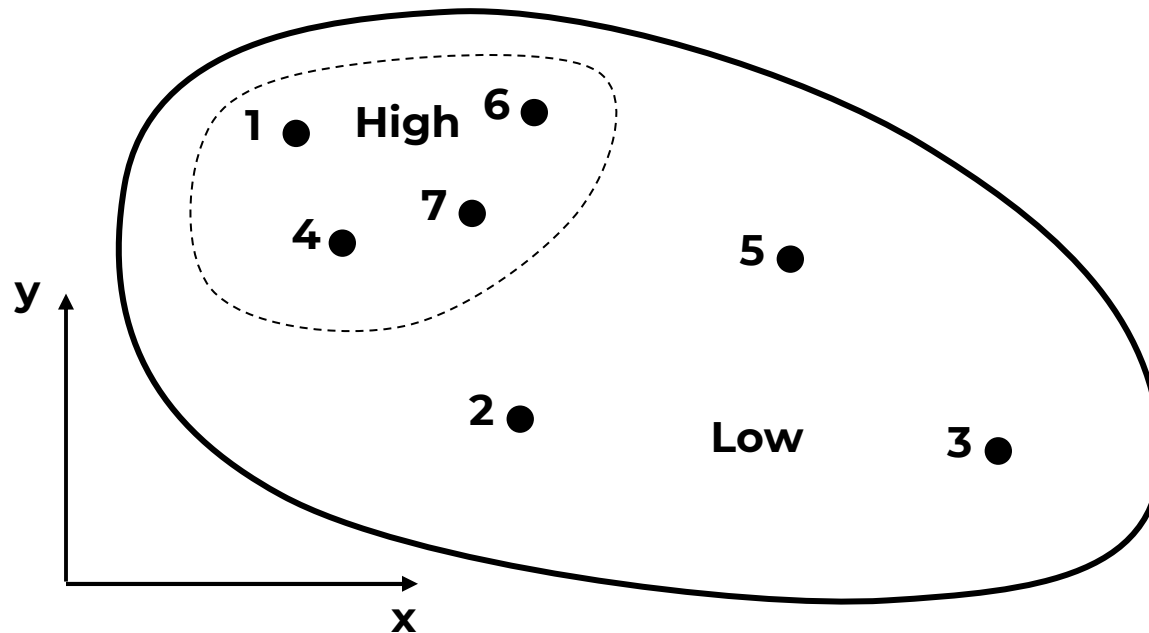


- To assess the average porosity to calculate OIP

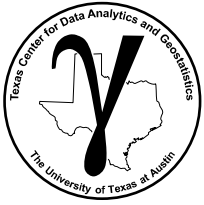


# Clustered Sample

Let's make an estimate for an Area / Volume of Interest:

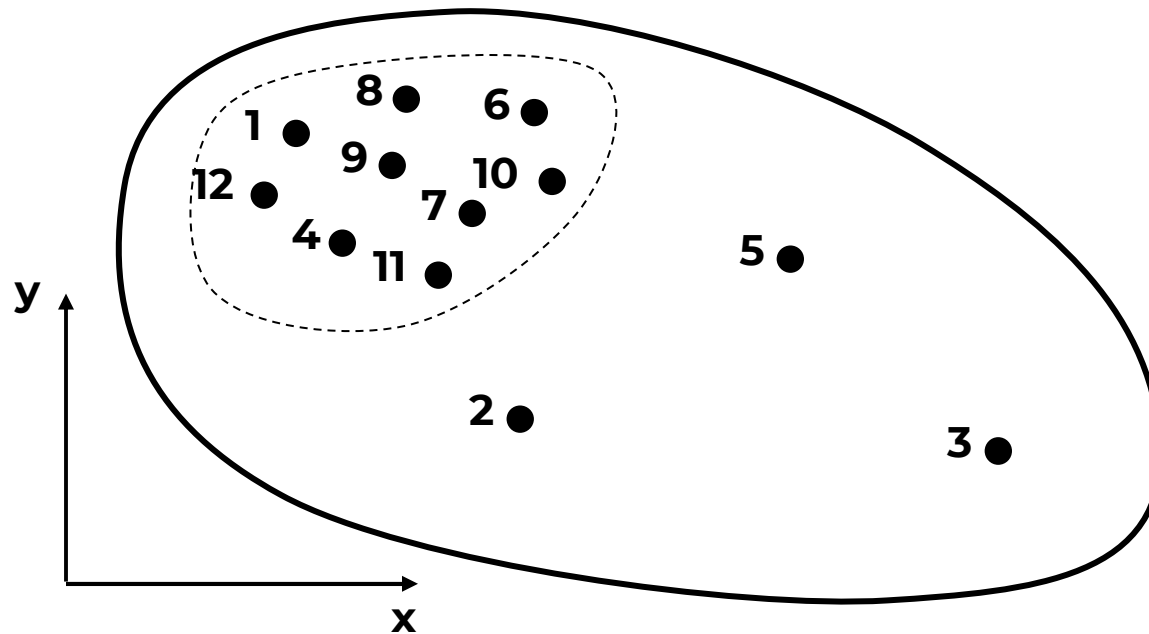


- What if we knew from seismic that the reservoir quality is better in the top left area?



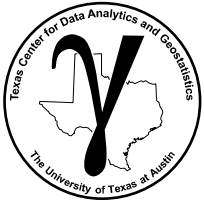
# Clustered Sample

Let's make an estimate for an Area / Volume of Interest:



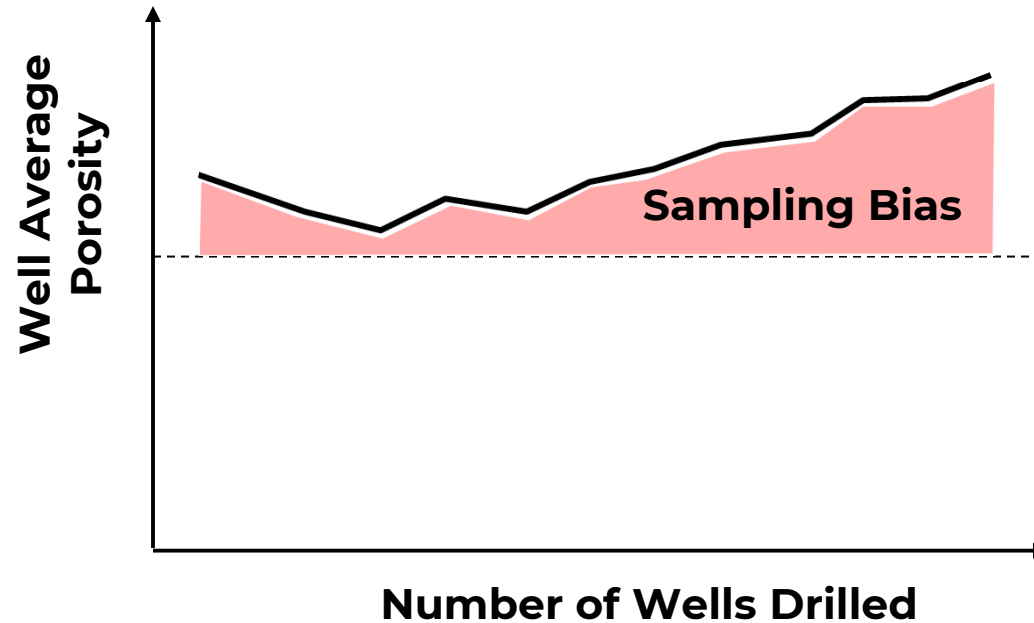
- What if we kept drilling in the high value region of the area of interest?



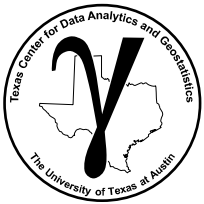


# Clustered Sample

How would our estimate of average porosity change as we drilled more wells?:



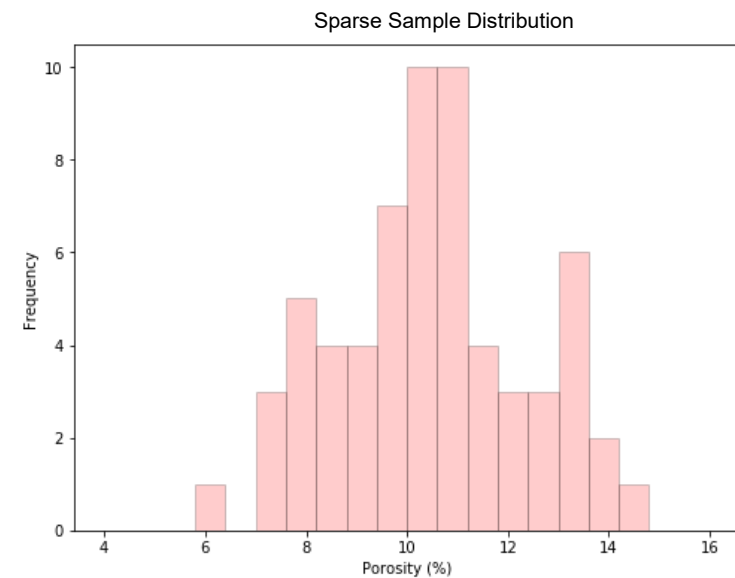
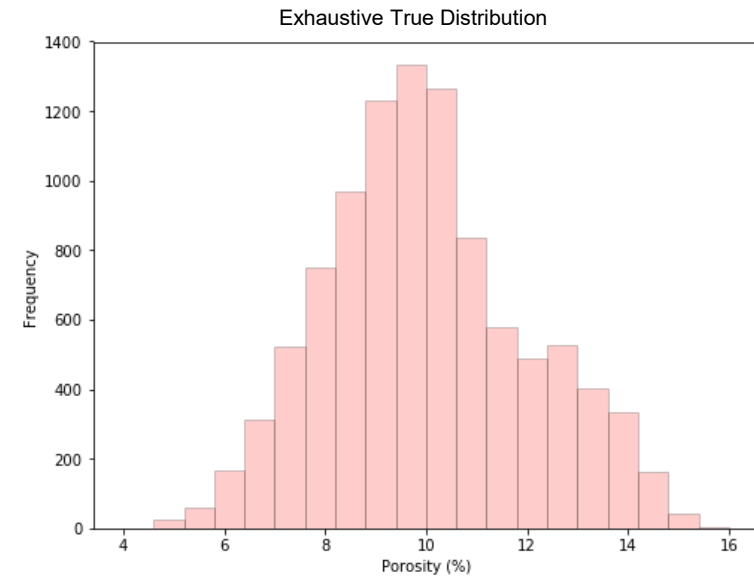
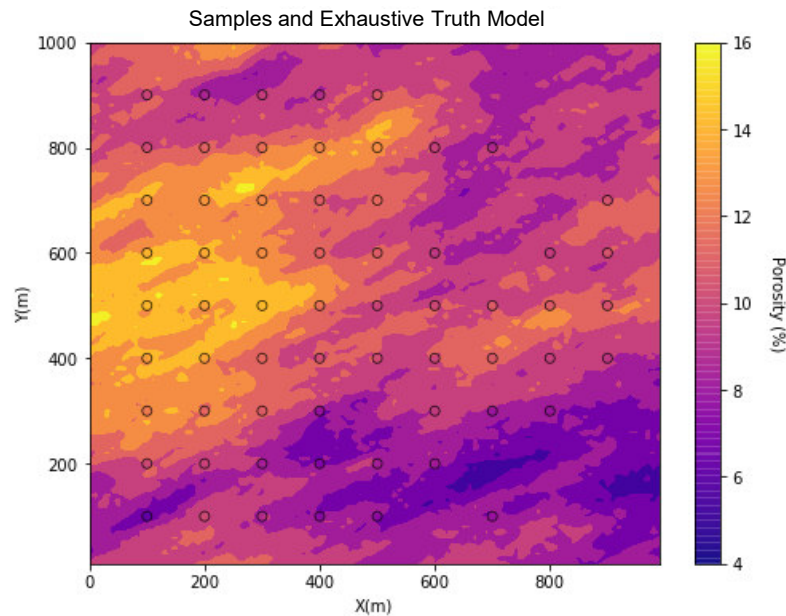
- The naïve sample average becomes more biased!
- We need a method to correct for clustered samples.



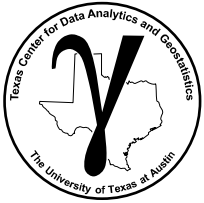
# Some Clustered Data

**Here's data and x-ray vision:**

- Location map of 64 wells. with truth model.
- See the error between the samples and the underlying truth model.



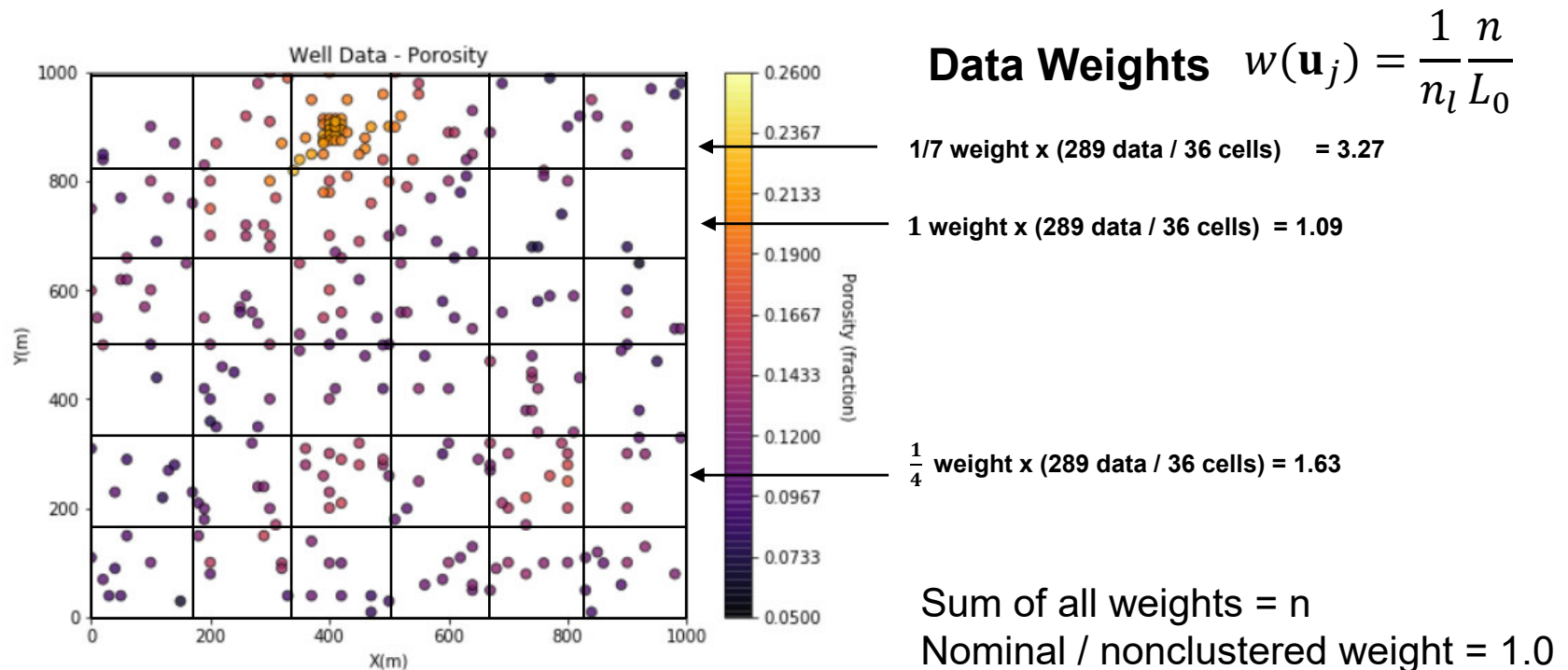
Truth Mean = 10.0 , Clustered Sample Mean = 10.48 , Error = 4.8 %



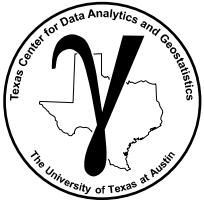
# Cell Declustering

## Cell Declustering, a method for calculating declustering weights

- divide the volume of interest into a grid of cells  $l = 1, \dots, L$  count the occupied cells  $L_o$  and the number in each cell  $n_l, l = 1, \dots, L_o$ , weight inversely by number in cell (standardize by  $L_o$ )



All data in the same cell get the same weight.



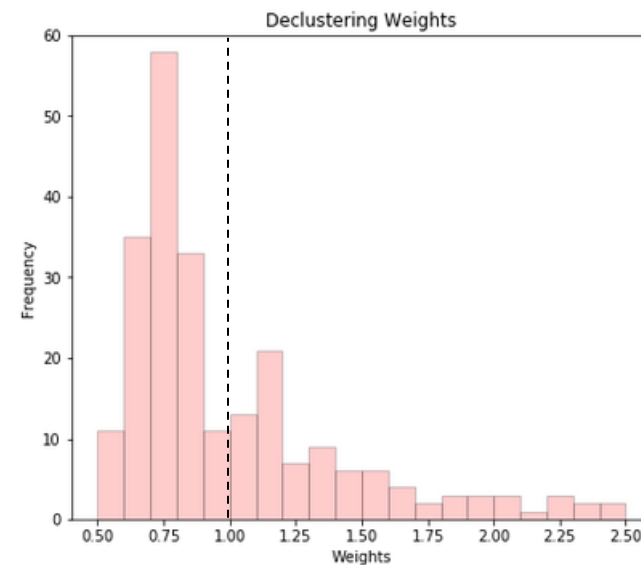
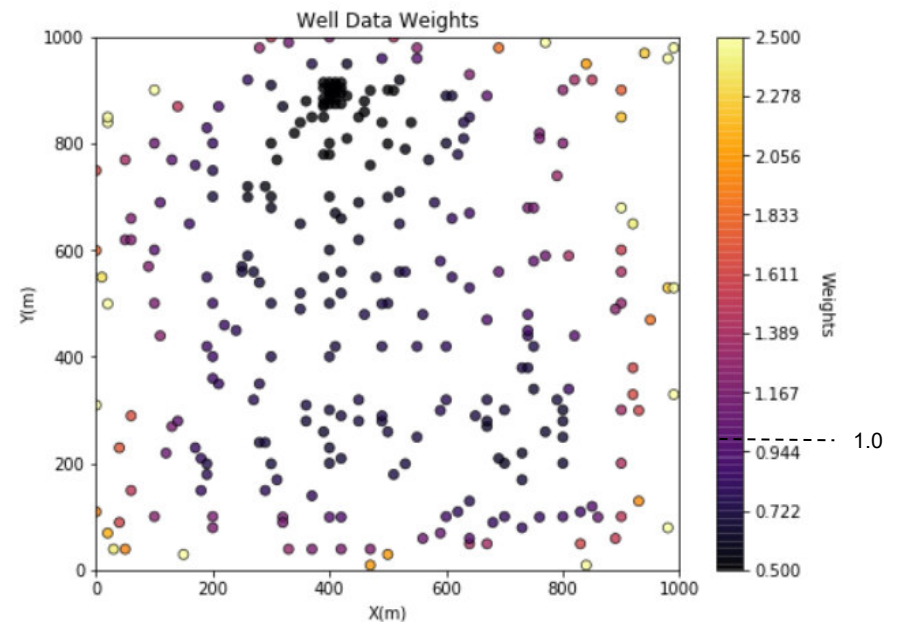
# Declustering Weights

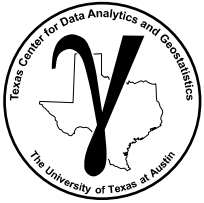
- Declustering weights
  - 1.0 nominal weight
  - < 1.0 reduced weight
  - > 1.0 increased weight

- Note: some software programs assume:

$$\sum_i^n w(\mathbf{u}_i) = 1$$

then 'nominal weight' is  $\frac{1}{n}$



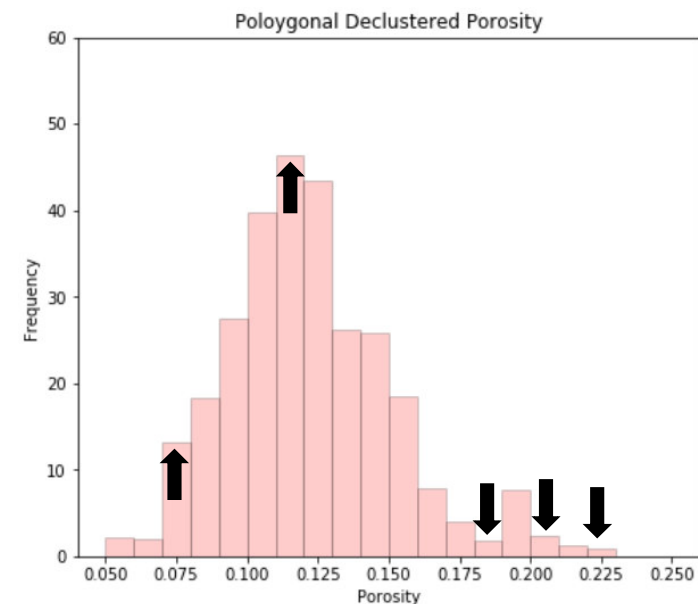
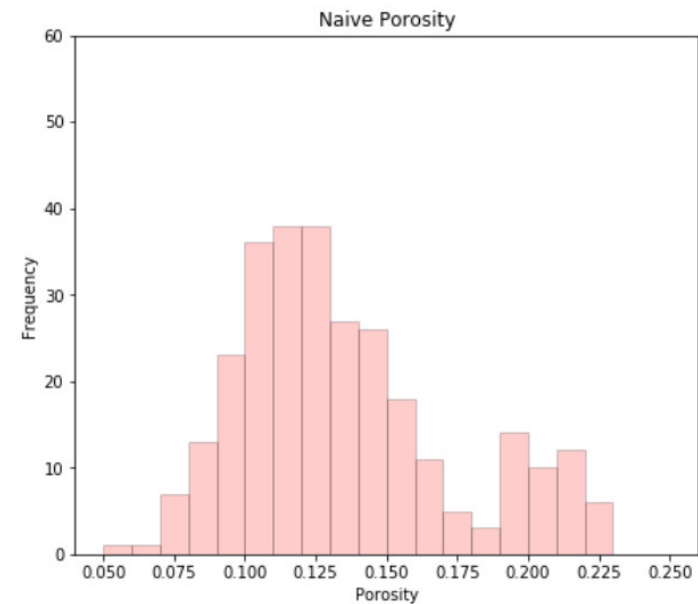


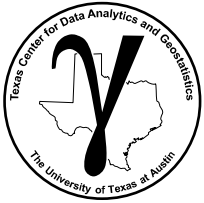
# Declustered Distribution

- Updated distribution with declustering weights
- Now data file / table include values and paired weights based on spatial arrangement.
- Possible to calculate any weighted statistic.
  - For example, declustered mean:

$$\bar{z} = \frac{\sum_i^n w(\mathbf{u}_i) z(\mathbf{u}_i)}{\sum_i^n w(\mathbf{u}_i) = n}$$

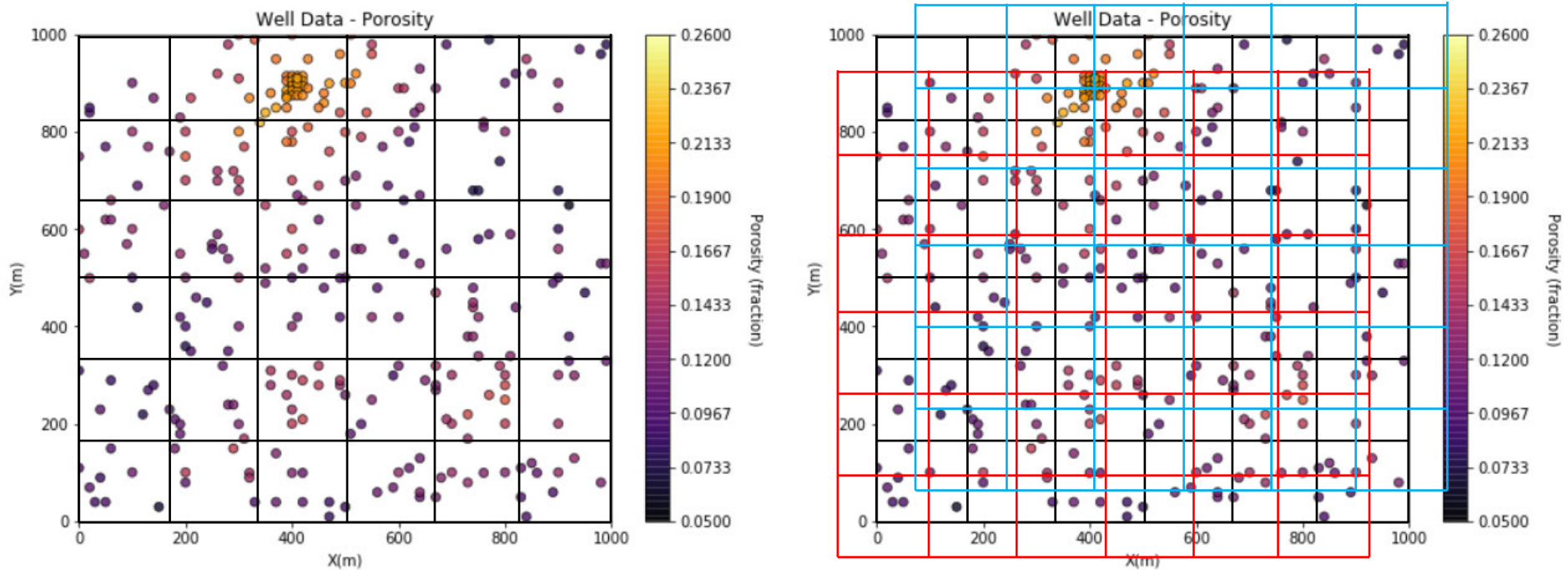
- Python Matplotlib hist allows for a vector of weights.



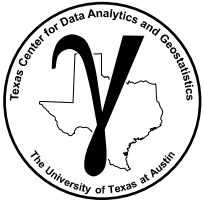


# Cell-based Declustering Offsets

- The result is sensitive to exact location of the cell mesh

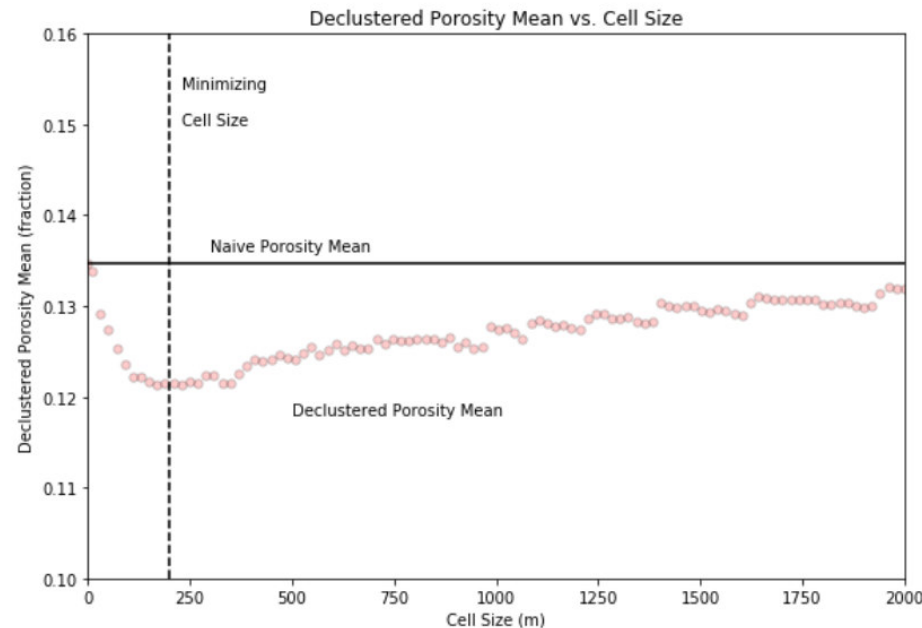


- This sensitivity is removed by iterating the mesh position, calculating the weights for each and averaging the result.

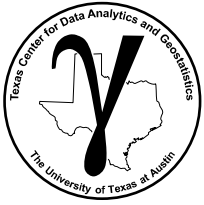


# Cell Size Selection

- Plot **declustered mean** versus the **cell size** for a range of cell sizes:



- There is no theory** that says we are looking for a minimum when the values are clustered in high values or a maximum when clustered in low values – it just seems to make sense
- The result can be very **sensitive to large scale trends** – it is often better to choose the cell size by visual inspection and some sensitivity studies
- Could choose the cell size so that there is **approximately one datum per cell in the sparsely sampled areas**, the nominal spacing



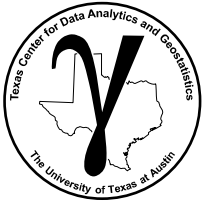
# Open Source Spatial Data Analytics in Python with GeostatsPy II

## Spatial Uncertainty Modeling with GeostatsPy

Lecture outline . . .

- Interactive Demo with GeostatsPy





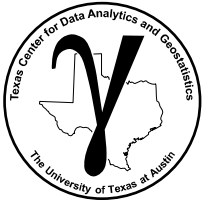
# Interactive Spatial Simulation Demonstration with GeostatsPy

**Let's calculate declustering  
weights:**

**Some Hints:**

- start with a small cell size
- increase the cell size
- observe the impact on data weights

Interactive Python Jupyter simulation calculator  
(Interactive\_Declustering.ipynb).

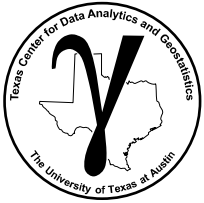


# Open Source Spatial Data Analytics in Python with GeostatsPy II

## Spatial Uncertainty Modeling with GeostatsPy

Lecture outline . . .

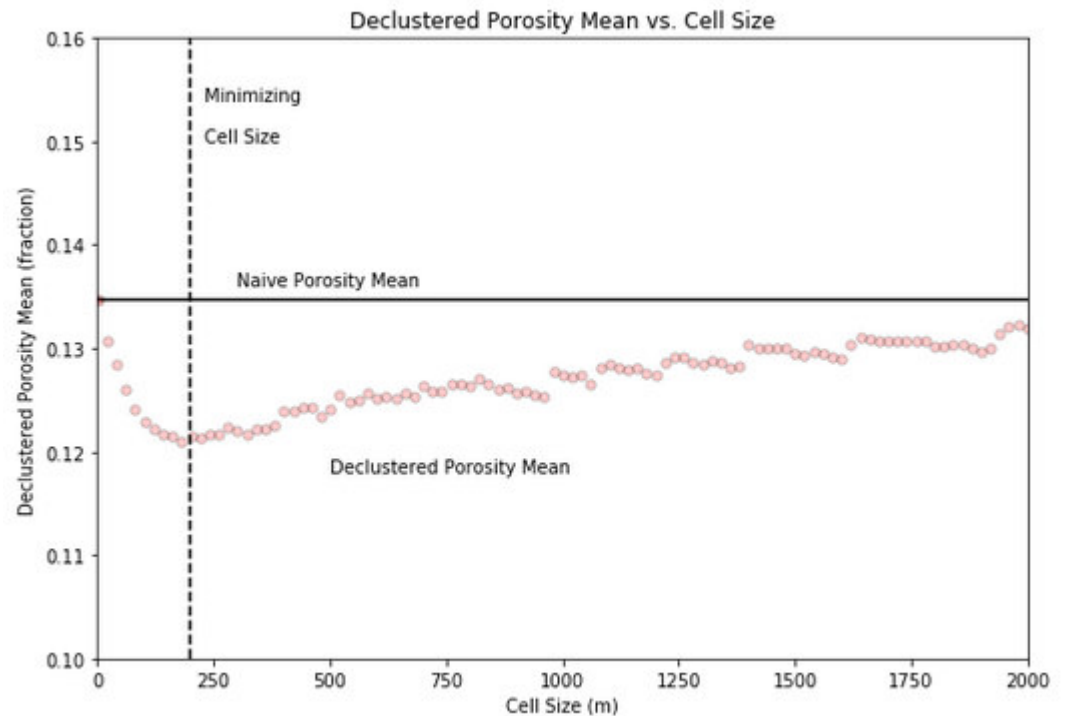
- Workflow with GeostatsPy



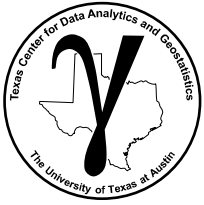
# Spatial Simulation Workflow with GeostatsPy

Let's walkthrough a more thorough a spatial data declustering workflow:

- calculate data weights
- visualize and QC the results



Python Jupyter variogram calculation  
(GeostatsPy\_declustering.ipynb).



# Open Source Spatial Data Analytics in Python with GeostatsPy II

## Spatial Uncertainty Modeling with GeostatsPy

Lecture outline . . .

- Spatial Simulation
- Interactive Demo with GeostatsPy
- Workflow with GeostatsPy