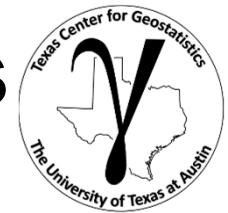


Data Analytics, Geostatistics and Machine Learning

Fundamental Concepts



Fundamental Concepts . .

- What is Subsurface Modeling?
- Modeling Goals
- Modeling Strategies
- Workflow Development

Introduction

Modeling Prerequisites

Fundamental Concepts

Probability

Sparse Data Workflows

Spatial Estimation

Spatial Uncertainty

Multivariate, Spatial

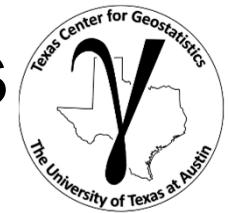
Novel Workflows

Conclusions

Instructor: Michael Pyrcz, the University of Texas at Austin

Data Analytics, Geostatistics and Machine Learning

Fundamental Concepts



Fundamental Concepts . .

- **What is Subsurface Modeling?**

Introduction

Modeling Prerequisites

Fundamental Concepts

Probability

Sparse Data Workflows

Spatial Estimation

Spatial Uncertainty

Multivariate, Spatial

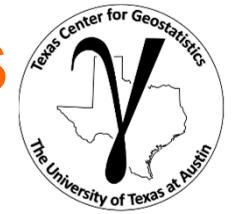
Novel Workflows

Conclusions

Instructor: Michael Pyrcz, the University of Texas at Austin

Statistics and Geostatistics

Some Definitions



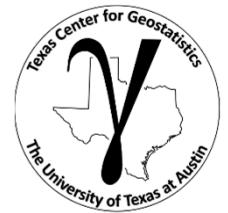
Statistics is concerned with mathematical methods for collecting, organizing, and interpreting data, as well as drawing conclusions and making reasonable decisions on the basis of such analysis.

Geostatistics is a branch of applied statistics that emphasizes (1) the geological context of the data, (2) the spatial relationship between data, (3) spatial uncertainty and (4) the different volumetric support and precision of the data.

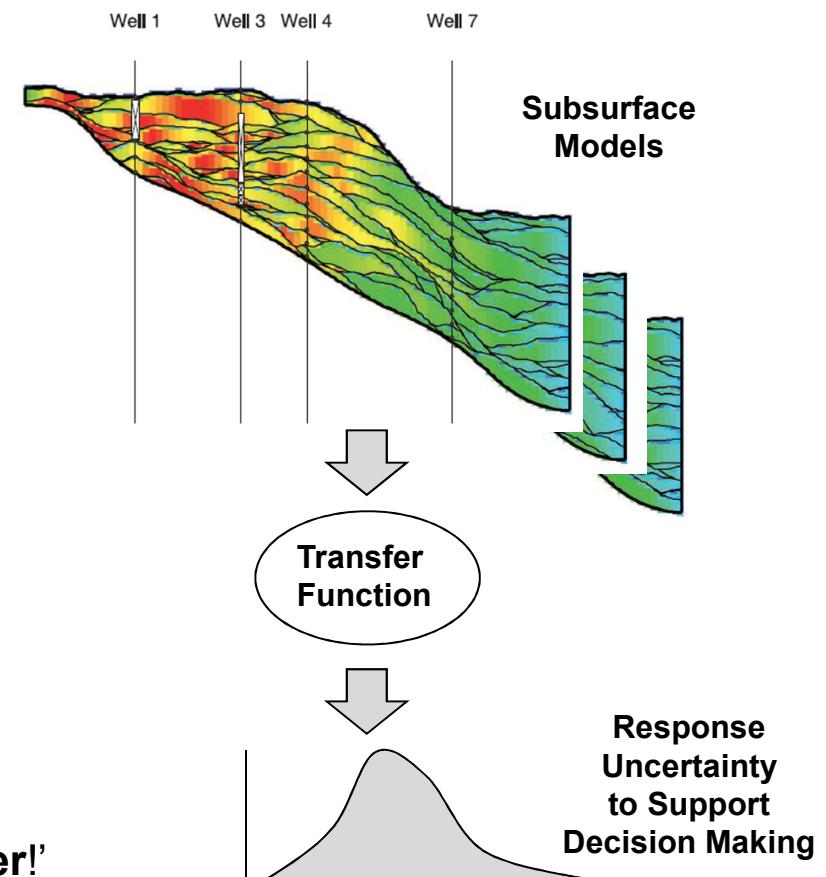
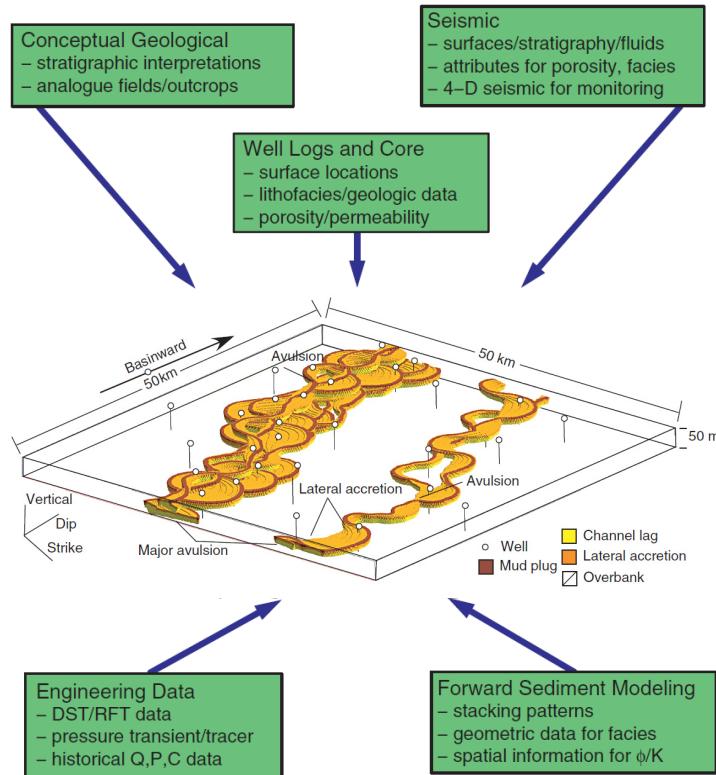
Why do we work with geostatistics in Geosciences?

- ✓ **Geological Context**
- ✓ **Spatial Relationships**
- ✓ **Variable Scale of Data**
- ✓ **Variable Data Precision**
- ✓ **Highly Multivariable**

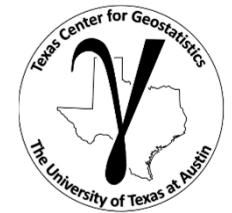
What is Subsurface Modeling?



Reservoir / Subsurface Modeling is the integration of all subsurface information to build a suite of models representing uncertainty to support decision making.



'If it doesn't get in the model, it doesn't matter!'

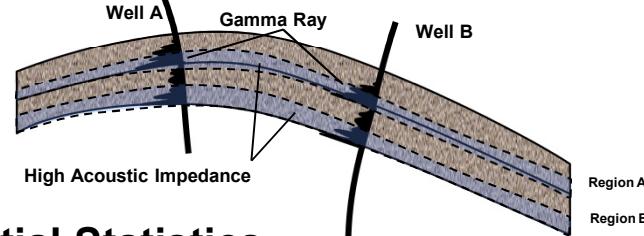


What is Subsurface Modeling?

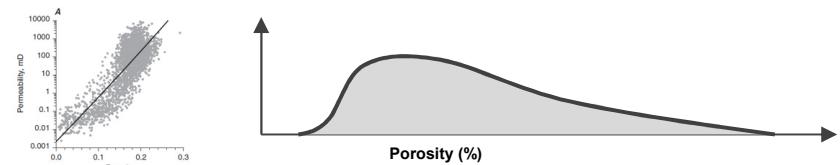
A Numerical Model

- ***quantification*** – integrate data and concepts, calculate summary spatial statistics and trends over the subsurface volume of interest
- ***subsurface model*** - spatial reservoir property distributions over the subsurface volume of interest that reproduce the quantification to support decision making

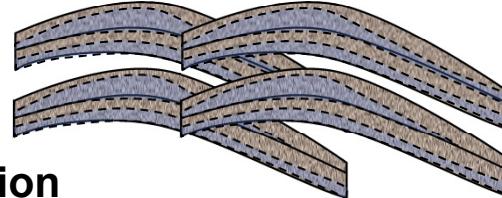
Data / Information



Trends / Spatial Statistics



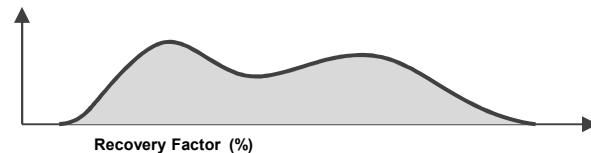
Subsurface Uncertainty via Ensemble of Models



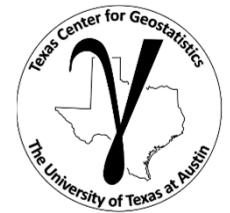
Transfer Function

Volumetrics, Flow Simulation

Decision Criteria



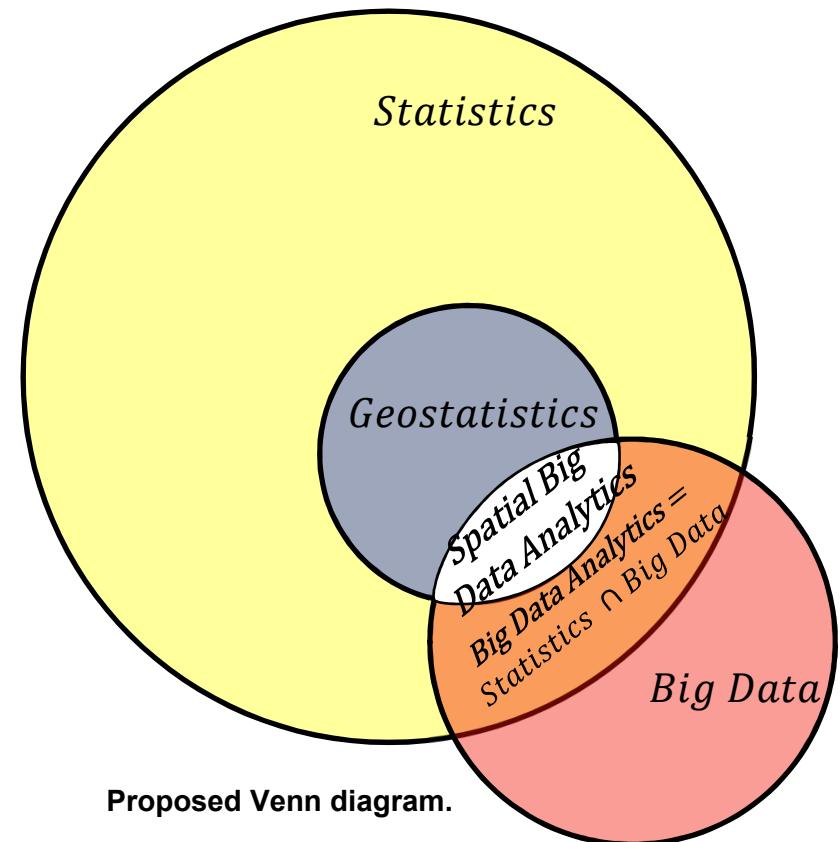
Big Data, Data Analytics and Geostatistics



Statistics is concerned with mathematical methods for collecting, organizing, and interpreting data, as well as drawing conclusions and making reasonable decisions on the basis of such analysis.

Geostatistics is a branch of applied statistics that emphasizes: (1) the spatial (geological) context of the data, (2) the spatial relationship between data, (3) the different volumetric support and precision of the data, and (4) spatial and data uncertainty.

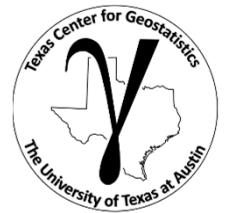
Big Data Analytics is the process of examining large and varied data sets (big data) to discover patterns and make decisions.



Given this:

Spatial big data analytics is the expert use of (geo)statistics to learn from our spatial data set.

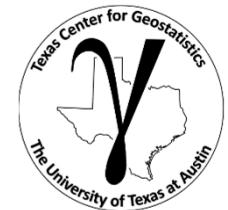
A Data Integration Challenge



Type	Resolution	Coverage	Information Type
<i>Core</i>	$\simeq \infty$	In Well Bore	Lithology, pore and sedimentary structures
<i>Well Log</i>	10 cm	Near Bore	Facies, porosity, mineralogy
<i>Image Log</i>	5 mm	Near Bore	Sedimentary structures, faults
<i>Seismic</i>	10 m	Exhaustive	Framework, trends, facies, porosity
<i>Production</i>	10–100 m	Drainage Radius	Volumes, connectivity, permeability
Analog			
<i>Mature Fields</i>	10–100 m	\leq Complete	Validation, prior for all
<i>Outcrop</i>	$\simeq \infty$	none	Concepts, input statistics
<i>Geomorphology</i>	$\simeq \infty$	none	Concepts
<i>Shallow Seismic</i>	\geq Element	none	Concepts, input statistics
<i>Experimental Stratigraphy</i>	$\simeq \infty$	none	Concepts
<i>Numerical Process</i>	\geq Complex	none	Concepts

A general summary of data types, resolution, coverage and information type.

Why Learn About Geostatistical Subsurface Modeling?

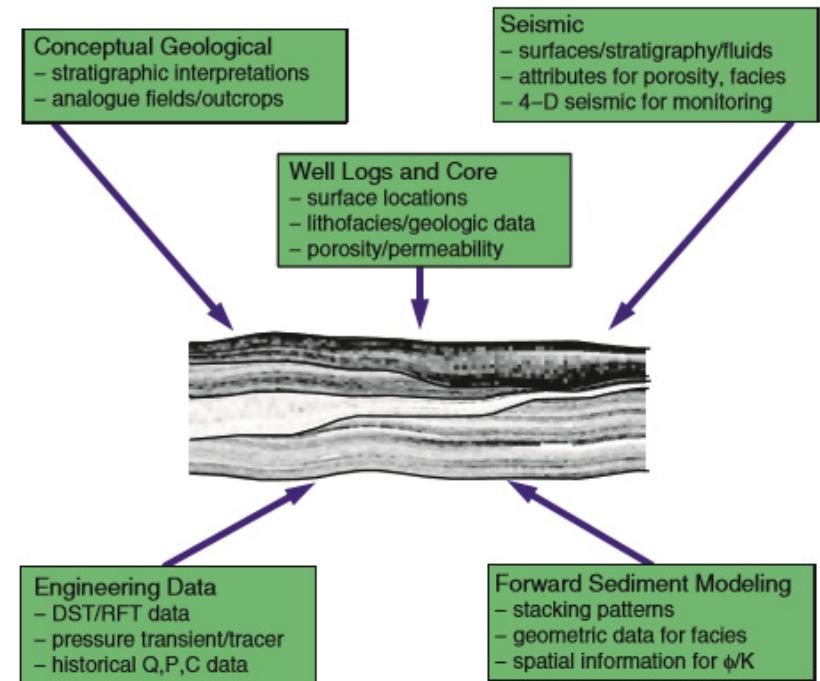


Why should you have a greater proficiency on reservoir modeling?

Level 1: Basic Understanding

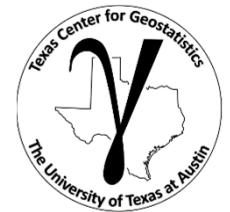
Most reservoir asset subsurface teams develop a **stochastic 3D reservoir model**.

- If you work with the subsurface, you will work with stochastic reservoir models!
- **Understand adjacent disciplines** and workflows in your team.



Subsurface asset integration (Pyrcz and Deutsch, 2014).

Why Learn About Geostatistical Subsurface Modeling?



Why should you have a greater proficiency on reservoir modeling?

Level 2: Improved Communication

Reservoir modeling sits in the middle of the subsurface team and integrates all available engineering, geological and geophysical information.

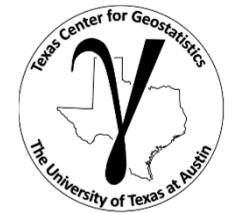
- Improved reservoir modeling capability results in **improved communication and integration** in the subsurface team.

TABLE 2.1. RESERVOIR CONCEPTS AND ASSOCIATED GEOLOGICAL AND GEOSTATISTICAL EXPRESSIONS

Concept	Geological Expression	Geostatistical Expression
Major changes in relationships between reservoir bodies	Architectural complexes and complex sets	Regions—separate units and model with unique methods and input statistics
Changes in reservoir properties within reservoir bodies	Basinward and landward stepping Fining/Coarsening up	Nonstationary mean
Stacking patterns if reservoir bodies	Organization, disorganization, compartmentalization, compensation	Attraction, repulsion, minimum and maximum spacing distributions, interaction rules
Major direction of continuity	Paleo-flow direction	Major direction of continuity, locally variable azimuth model

Subsurface concepts, with their geological and geostatistical expressions (modified from Pyrcz and Deutsch, 2014).

Why Learn About Geostatistical Subsurface Modeling?

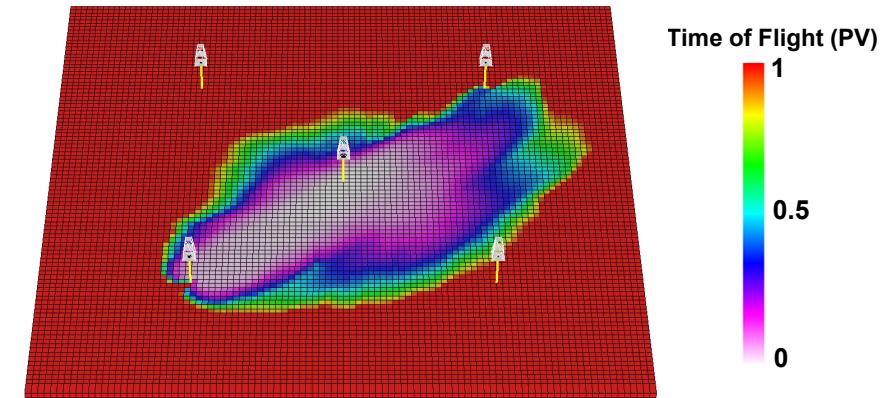
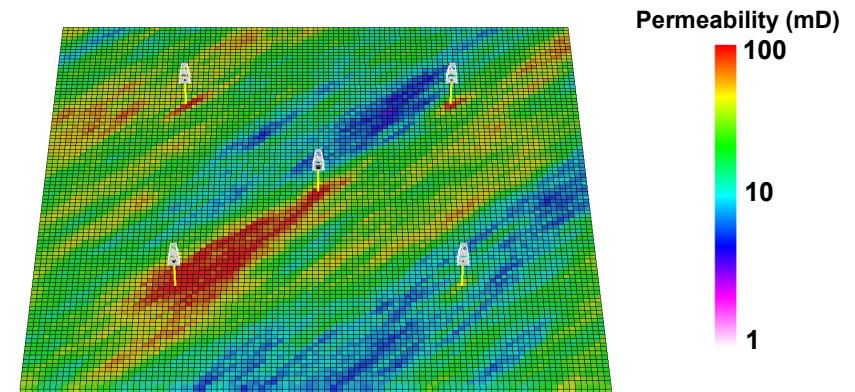


Why should you have a greater proficiency on reservoir modeling?

Level 3: Maximize Your Impact

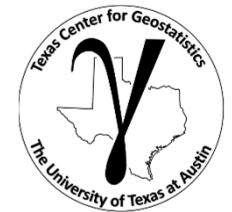
Reservoir models are directly applied for forecasting that support decision making.

- **Best integration of your knowledge** into the subsurface model.
- If your expertise does NOT impact the model, you may NOT impact the development decision!



Permeability heterogeneity and flow response.

Why Learn About Geostatistical Subsurface Modeling?



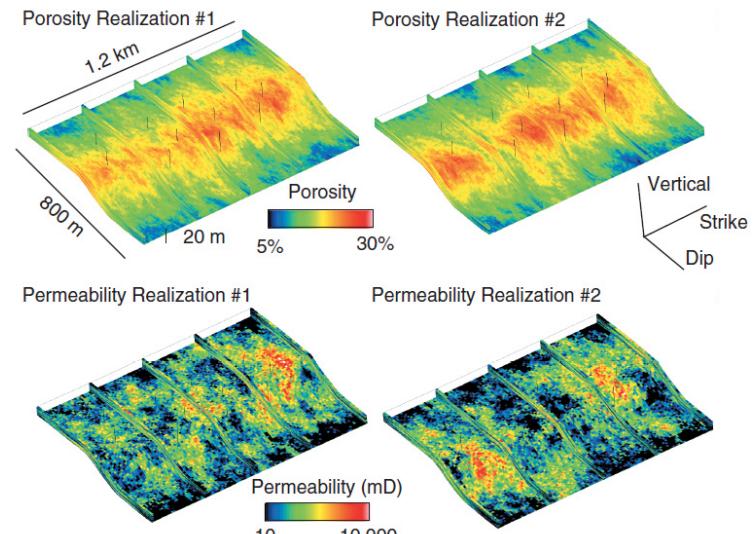
Why should you have a greater proficiency on reservoir modeling?

Level 4: Build Subsurface Models

Most subsurface modelers are geoscientists and engineers that learned on the job.

Become a subsurface modeler!

- Black box, uninformed reservoir modeling will result in bad decisions.
- Advanced knowledge unlocks novel workflows to solve difficult subsurface problems.



Subsurface asset integration.

Data Analytics, Geostatistics and Machine Learning

Fundamental Concepts



Fundamental Concepts . .

- Modeling Goals

Introduction

Modeling Prerequisites

Fundamental Concepts

Probability

Sparse Data Workflows

Spatial Estimation

Spatial Uncertainty

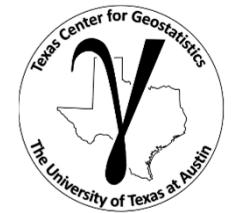
Multivariate, Spatial

Novel Workflows

Conclusions

Instructor: Michael Pyrcz, the University of Texas at Austin

Model Goal and Purpose Modeling

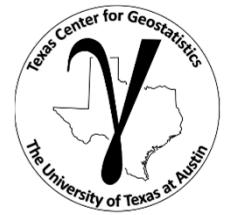


1. Build a Numerical Model / Common Earth Model

- Platform to integrate all available information, unified understanding of the subsurface
- Establish what is known, unknown and critical risks
- Numerical support for future investigation
- Communication tool

Is there value in just building a model?

Model Goal and Purpose Modeling



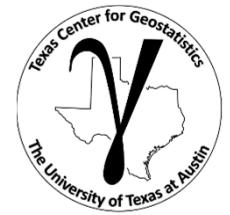
2. Assess Resources

- Compute the gross volume of interest and the associated spatial distribution
- Include some consideration for extraction method and associated scales and thresholds

3. Quantify Resources Uncertainty

- Multiple sources and multiple scales of uncertainty
- Results may be used to direct data collection to reduce uncertainty, support development decisions

Model Goal and Purpose Modeling



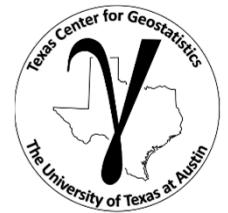
4. Investigate Geologic Risks

- Consider a wide range of possible subsurface features
- Evaluate sensitivities and risks, down- and up-side outcomes

5. Exporting Statistics

- Develop a robust set of statistics from a mature reservoir to apply elsewhere
- This may include trends, distributions, training images etc.
- Common to support early development in sparse data settings.

Model Goal and Purpose Modeling



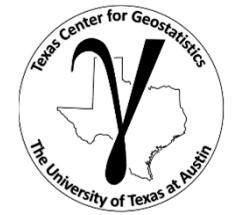
6. Evaluate the Need for Additional Data

- Determining the local and global uncertainty models
- Assess the value of additional data in reducing these uncertainties

7. Assess Reserves

- Calculate the resources that would be extracted after applying economic thresholds and technical limits of the extraction methodology.
- Modeling and calculations are consistent with reporting standards

Model Goal and Purpose Modeling



8. Evaluate Different Recovery Processes

- There are various decisions with respect to primary, secondary and tertiary recovery
- Optimize the recovery method accounting for all information and uncertainty

9. Make Final Decisions

- Provide the local best estimates to support well site selection

Data Analytics, Geostatistics and Machine Learning

Fundamental Concepts



Fundamental Concepts . .

- Modeling Strategies

Introduction

Modeling Prerequisites

Fundamental Concepts

Probability

Sparse Data Workflows

Spatial Estimation

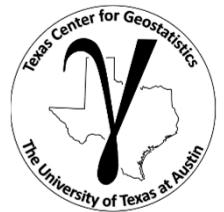
Spatial Uncertainty

Multivariate, Spatial

Novel Workflows

Conclusions

Instructor: Michael Pyrcz, the University of Texas at Austin

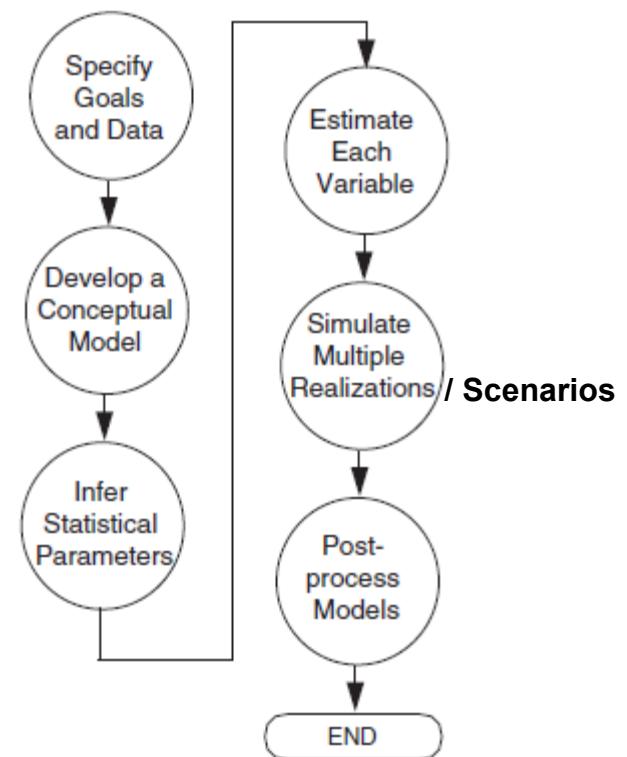


Modeling Strategies

Common Subsurface Modeling Workflow

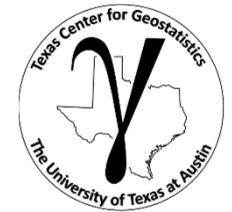
- Specify modeling goals and available resources
- Develop conceptual model
- Infer statistical parameters
- Build estimation models
- Build simulation models
- Post-process models

The Common Workflow (3.3.1)



The common workflow modified from Pyrcz and Deutsch (2014).

Modeling Strategies



Data Integration Workflow

- Integrating wells, seismic, production and concepts
- The results is a consistent numerical representation of the reservoir properties

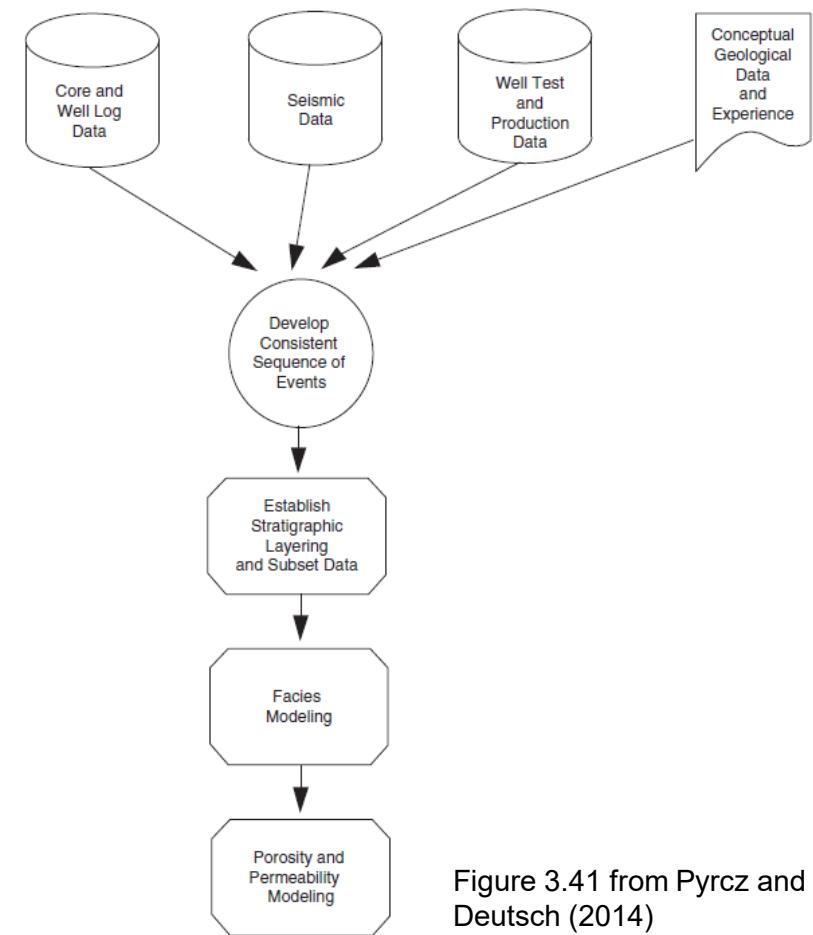
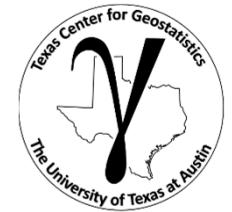


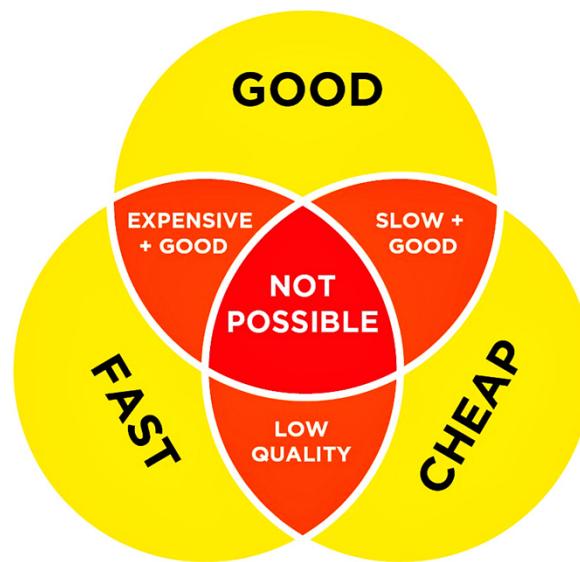
Figure 3.41 from Pyrcz and Deutsch (2014)

Modeling Strategies

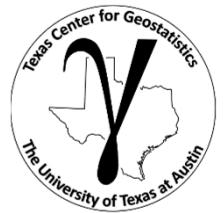


Fit-for-Purpose Modeling

- Model workflow design considering the goals of the model
- May also consider future needs for the model
- Accounting for resources, time and people (expertise) available



Good, Fast and Cheap is not Possible
image from <https://www.purechat.com/blog/fast-cheap-and-good-the-small-business-guide-to-content-creation/>



Modeling Strategies

Modeling Constraints

- Professional Time
 - work hours available limited by workforce and project timelines schedules
- Organization Capability
 - The skill sets of the professionals that are available
- Computational Facilities
 - The hardware and software available for the project
- Total Budget
 - Limiting professional time, computational resources and data collection

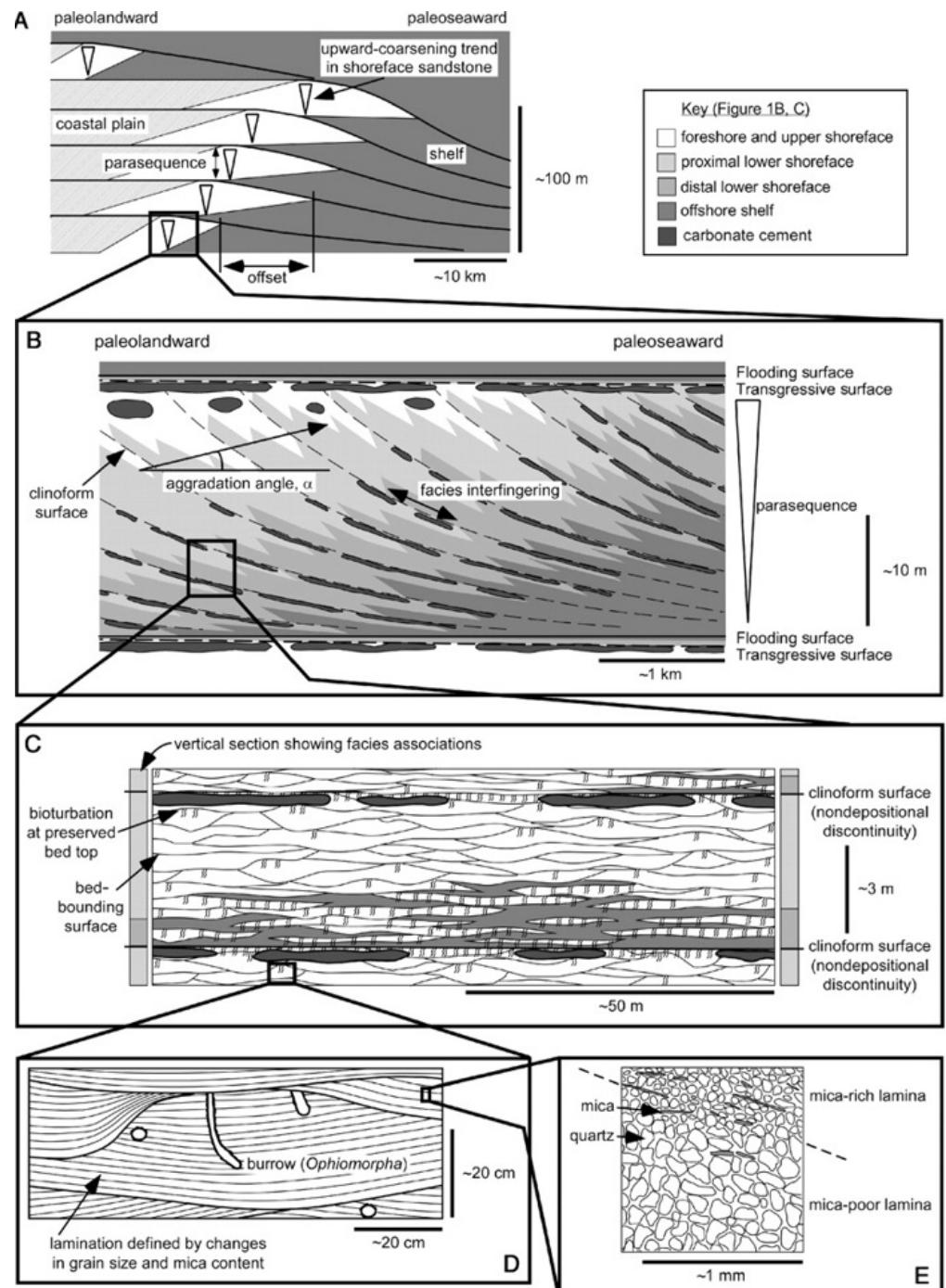
All projects are constrained and require significant prioritization.

Modeling Strategies

Top-down Reservoir Modeling (Williams et al., 2004, Sech et al., 2009)

- Start with the simplest model possible
- Add detail as required, until it doesn't have an impact on the transfer function
- Very efficient for fast initial assessments
- Learn the impact of scale / details

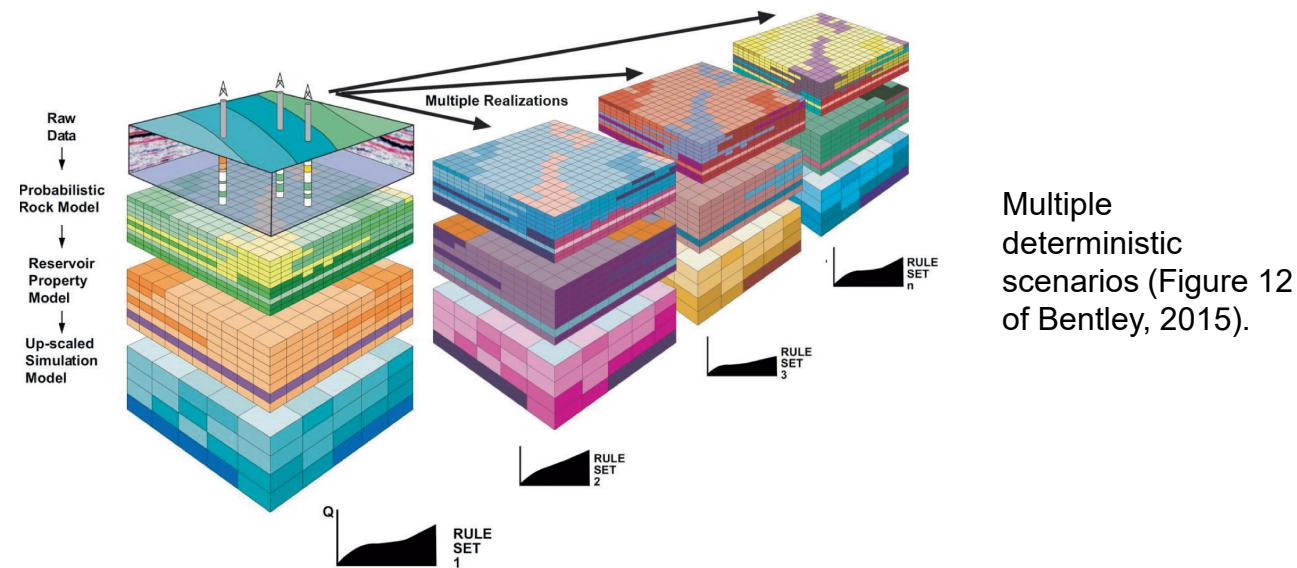
Surface-based top down modeling based on outcrop from Sech et al., 2009



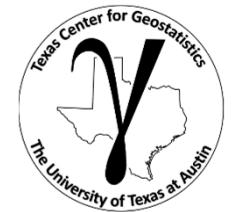
Modeling Strategies

Modeling for Discomfort (Bentley, 2015)

- Models become tools for verification of a decision already partially or fully made, this is modeling for comfort!
- Bentley recommends that we model for discomfort! Stress test our current concepts and the decision-making.
- Identify remaining up-side potential and secure against worst case.
- Need to recognize our biases



Modeling Workflows

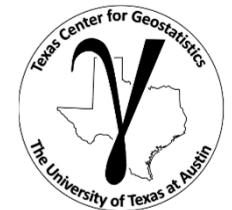


Modeling Workflows Based on Model Goals

Let's discuss:

- 2-D Mapping for Volumetrics
- Regional Mapping
- Mini/Micro Modeling
- Reservoir Modeling

Modeling Workflows

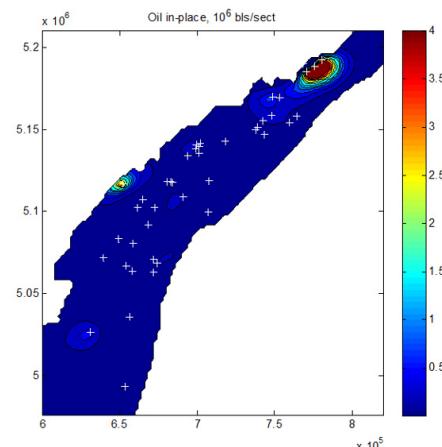


2-D Mapping for Volumetrics

- Goal: Produce a map of remaining resource in place
- Properties: thickness, vertically averaged porosity and saturation, seismic attributes
- Model: estimation model for smoothly varying properties between wells, may integrate physics
- Calculate resources in place, e.g.

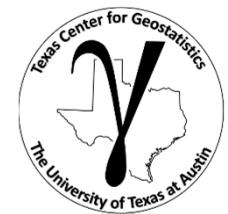
$$OIP = \sum_{\alpha=1}^n t(\mathbf{u}_\alpha) \cdot \bar{\phi}(\mathbf{u}_\alpha) \cdot \bar{s_o}(\mathbf{u}_\alpha)$$

where t is thickness, $\bar{\phi}$ and $\bar{s_o}$ are vertically averaged porosity and oil saturation.



Kriged remaining resource map for Utica Shale, Quebec by the Geological Survey of Canada

(https://www.researchgate.net/publication/263818399_GEOLOGICAL_SURVEY_OF_CANADA_OPEN_FILE_7606_Geological_Characteristics_and_Petroleum_Resource_Assessment_of_Utica_Shale_Quebec_Canada/figures?lo=1&utm_source=google&utm_medium=organic)



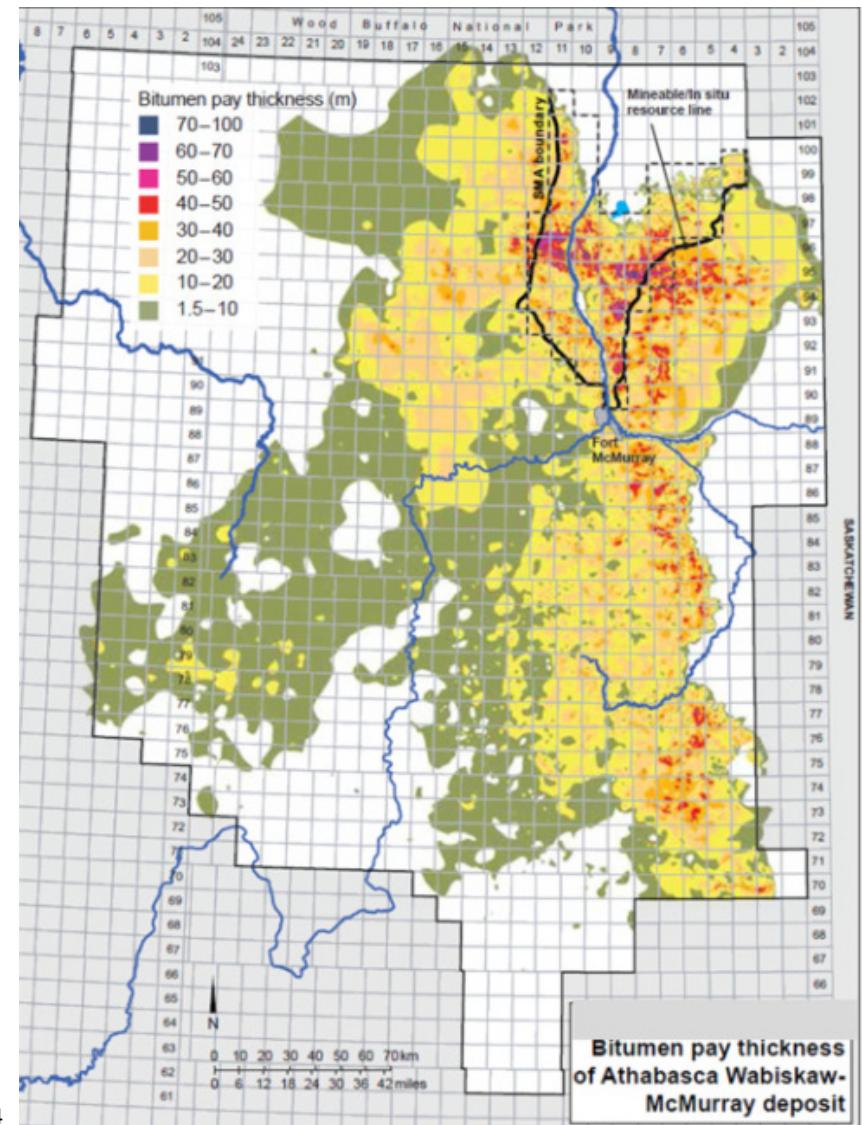
Modeling Workflows

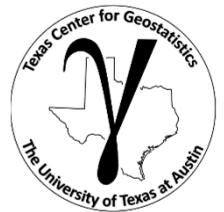
Regional Mapping

- Goal: Understand the spatial distribution of the resource to evaluate different lease areas, sequencing development, and layout of facilities
- Properties: thickness, permeability height
- Model: very large area (1000's of km²) with large model cells (low detail), vertical details collapsed to major heterogeneities and model may include 10's of variables

Regional map of bitumen pay thickness (m) for NW Alberta, Canada from Hein, (2015).

<https://www.sciencedirect.com/science/article/pii/B978044635297000183>

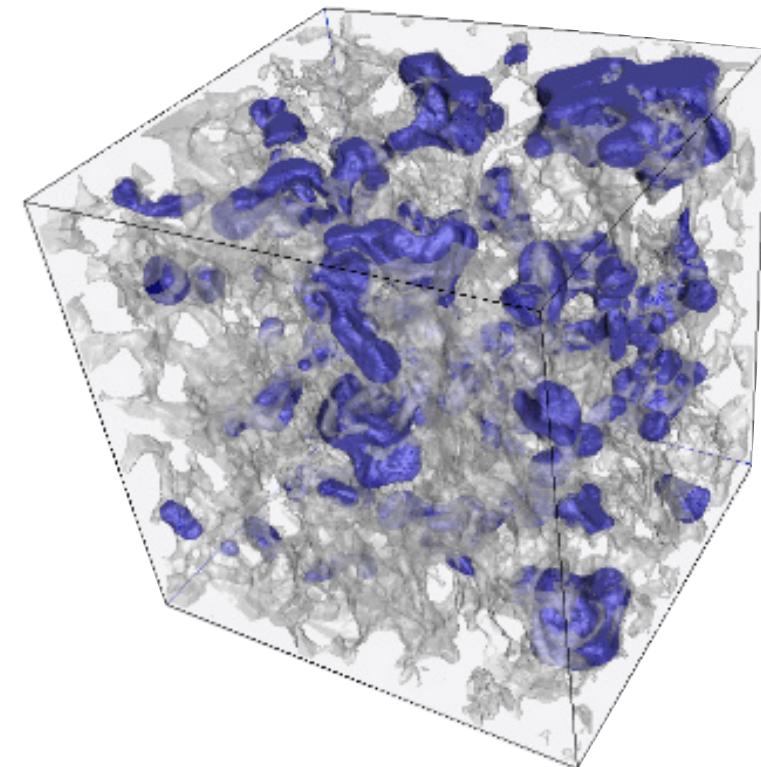




Modeling Workflows

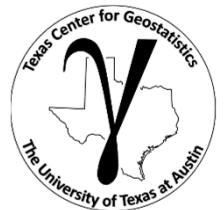
Micro / Mini Modeling

- Goal: Build models pore-scale to transfer their influence to reservoir models for forecasting.
- Properties: rock (mineralogy and fluid models)
- Model: for micro the scale of a single core plug / core is typical, for mini the scale of a single reservoir modeling cell



Direct simulation of residual phase
(disconnected blobs in blue) in Berea
Sandstone (imaged based pore grain surface
shown in transparent gray).

<https://www.digitalrocksportal.org/>



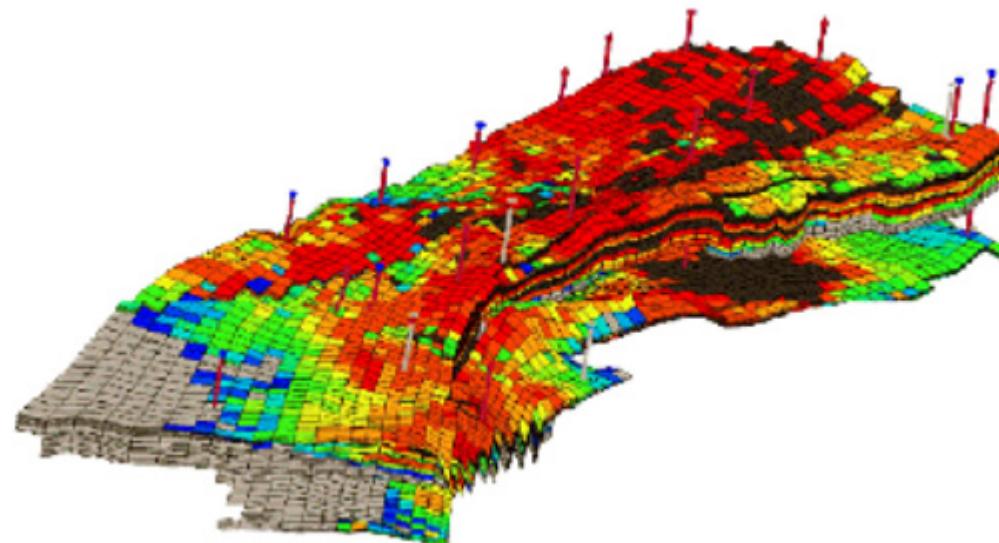
Modeling Workflows

Reservoir Modeling

- Goal: Input for connectivity calculations and flow simulation.
- Properties: facies, porosity, permeability, saturation, seismic attributes, pressure and production rates.
- Model: cells of 10's m areal x 0.25 – 1.0 m vertical extent, over 10's km x 10's m.

Reservoir model with color indicating oil saturation
(hot colors have more oil).

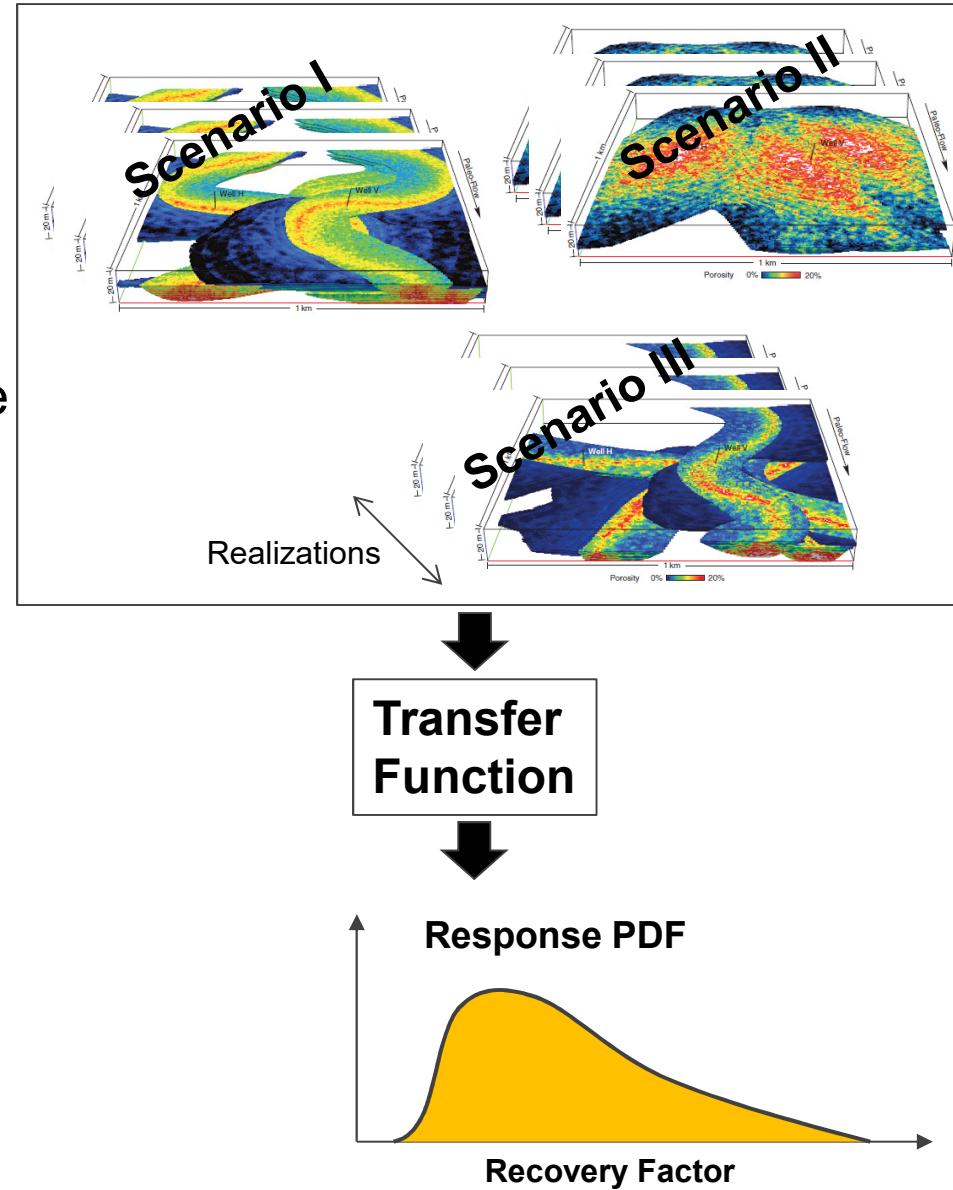
https://wiki.seg.org/wiki/Reservoir_simulation



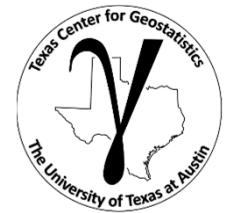
Modeling Workflows

Reservoir Modeling

1. Integrate all available information to build multiple scenarios and realizations to sample the uncertainty space
2. Apply all the models to the transfer function
3. Assemble the distribution of the outcomes
4. Make the decision accounting for this uncertainty model.



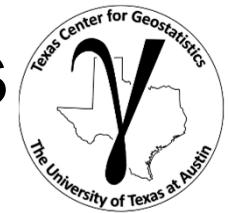
Subsurface Modeling Take-aways



Topic	Application to Subsurface Modeling
Modeling Goals	<p>Modeling for integration? Flow forecasting? Reserves assessment?</p> <p><i>There are a wide variety of modeling goals.</i></p>
Model Workflows	<p>Regional, reservoir, minin and micro modeling.</p> <p><i>Utilize a variety of modeling workflows for a variety of goals.</i></p>
Modeling for Discomfort	<p>Model to test and validate hypothesis</p> <p><i>Model for discomfort, actively attempt to disprove current theories to prove upside and to mitigate downside risk.</i></p>

Data Analytics, Geostatistics and Machine Learning

Fundamental Concepts



Fundamental Concepts . .

- Workflow Development

Introduction

Modeling Prerequisites

Fundamental Concepts

Probability

Sparse Data Workflows

Spatial Estimation

Spatial Uncertainty

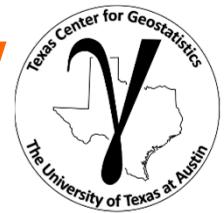
Multivariate, Spatial

Novel Workflows

Conclusions

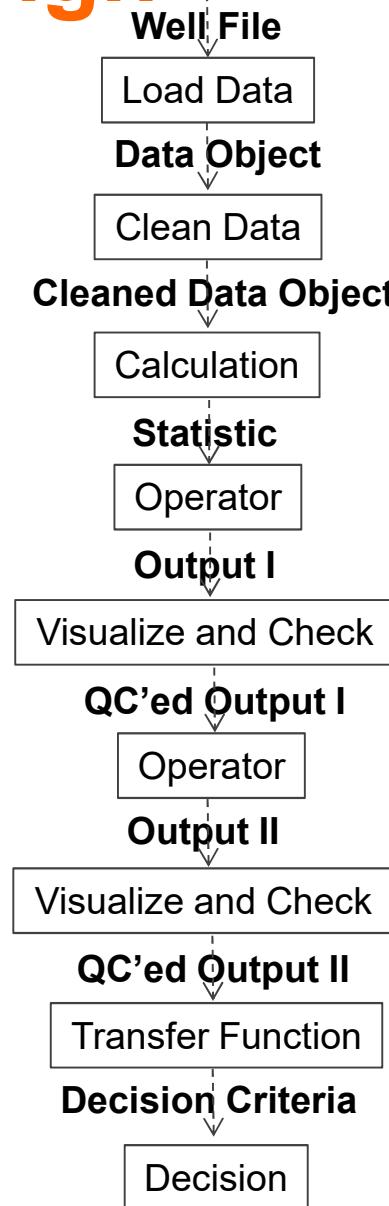
Instructor: Michael Pyrcz, the University of Texas at Austin

Basics of Workflow Design

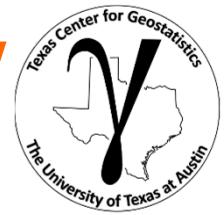


Design a Set of Steps to Accomplish the Goal, common steps include:

- a) Load Data
 - Load data into a data structure that we can work with
- b) Format / Check / Clean Data
 - Get the data ready for the workflow
- c) Run Statistical Calculation / Visualization
 - Histograms, location maps, variogram, trend, conditional probabilities, data mining
- d) Run Operator
 - Declustering, spatial continuity, spatial estimation and simulation, model post-processing
- e) Transfer Function
 - Any type of summary of the model to support decision making, such as volumetric calculation, connectivity analysis, trend modeling, flow simulation

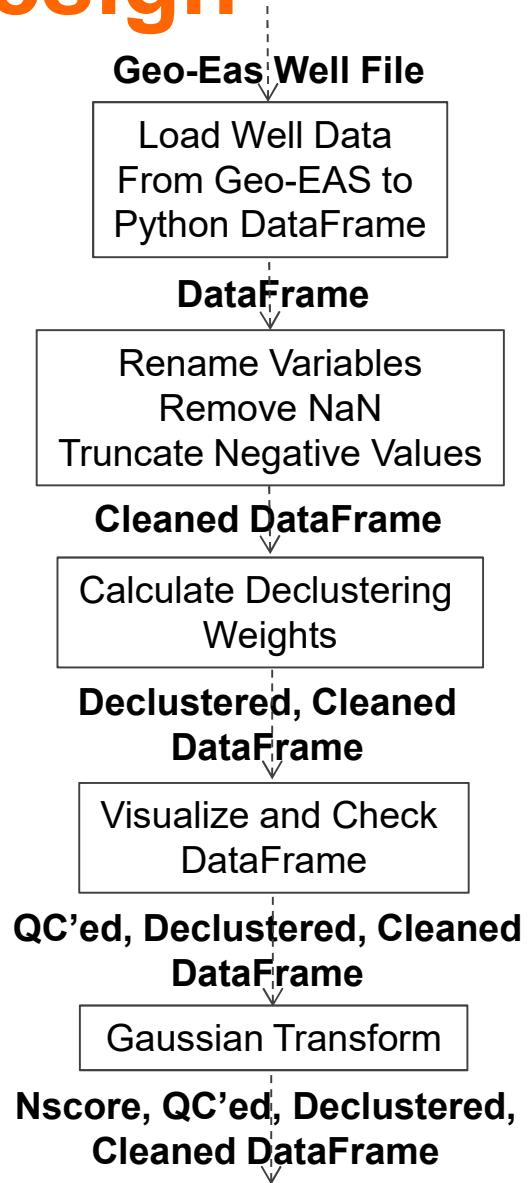


Basics of Workflow Design

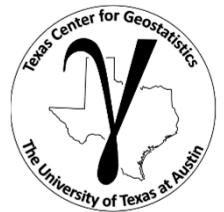


2. Design a Set of Steps to Accomplish the Goal, **workflows**:

- The data, statistics, and models flow from step to step.
- The well data (data table) and seismic data and other mapped data along with models (gridded) are passed from step to step.
- Complete subsurface modeling workflows are often complicated, many interrelated steps
- The process is often sequence dependent, e.g. data cleaning and transformations before variogram calculation
- Visualization and checking steps are required after every operation



Example subset of a workflow with data flowing over multiple steps.



Basics of Workflow Design

2. Design a Set of Steps to Accomplish the Goal, **documentation:**

- Every step includes expert decisions.
- **Writing script / code for your workflow provides a very useful audit trail / documentation.**
- Takes 2-3 times longer, but given most subsurface modeling workflows are iterated multiple times it is well worth it.
- It is essential to document the many modeling decisions.
- You are writing to your future self or your replacement.

Documentation

Wells provided by _____ on Date _____. Includes the first _____ wells and excludes _____. Porosity by neutron density log calibrated to core measures.

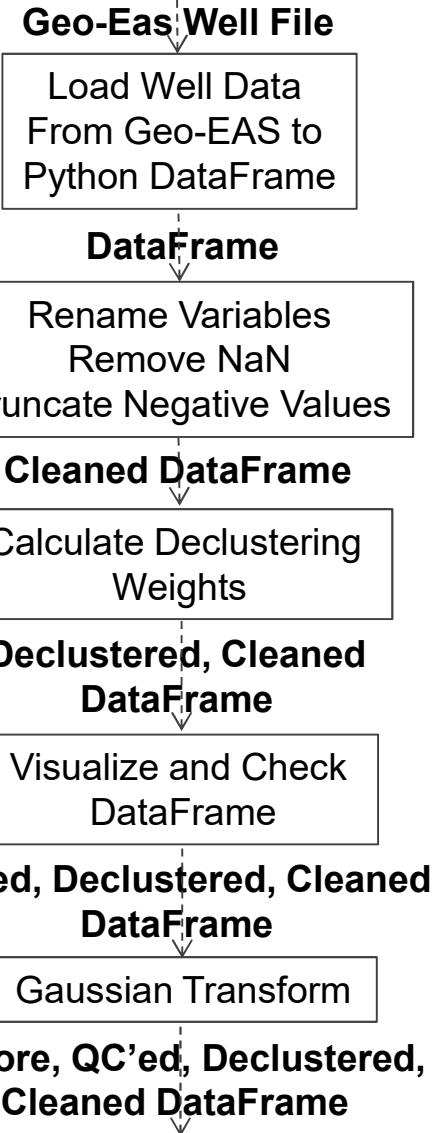
Values < _____ are considered below measurement threshold and set to 0.001 porosity.

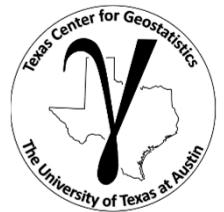
Cell size for declustering weights selected to be 200m based on nominal well spacing.

Weights have range [0.01,2.5] and are spatially rational.

Transform with tail assumption of [0.001, 0.30] linear extrapolation.

Example subset of a workflow with data flowing over multiple steps with documentation.





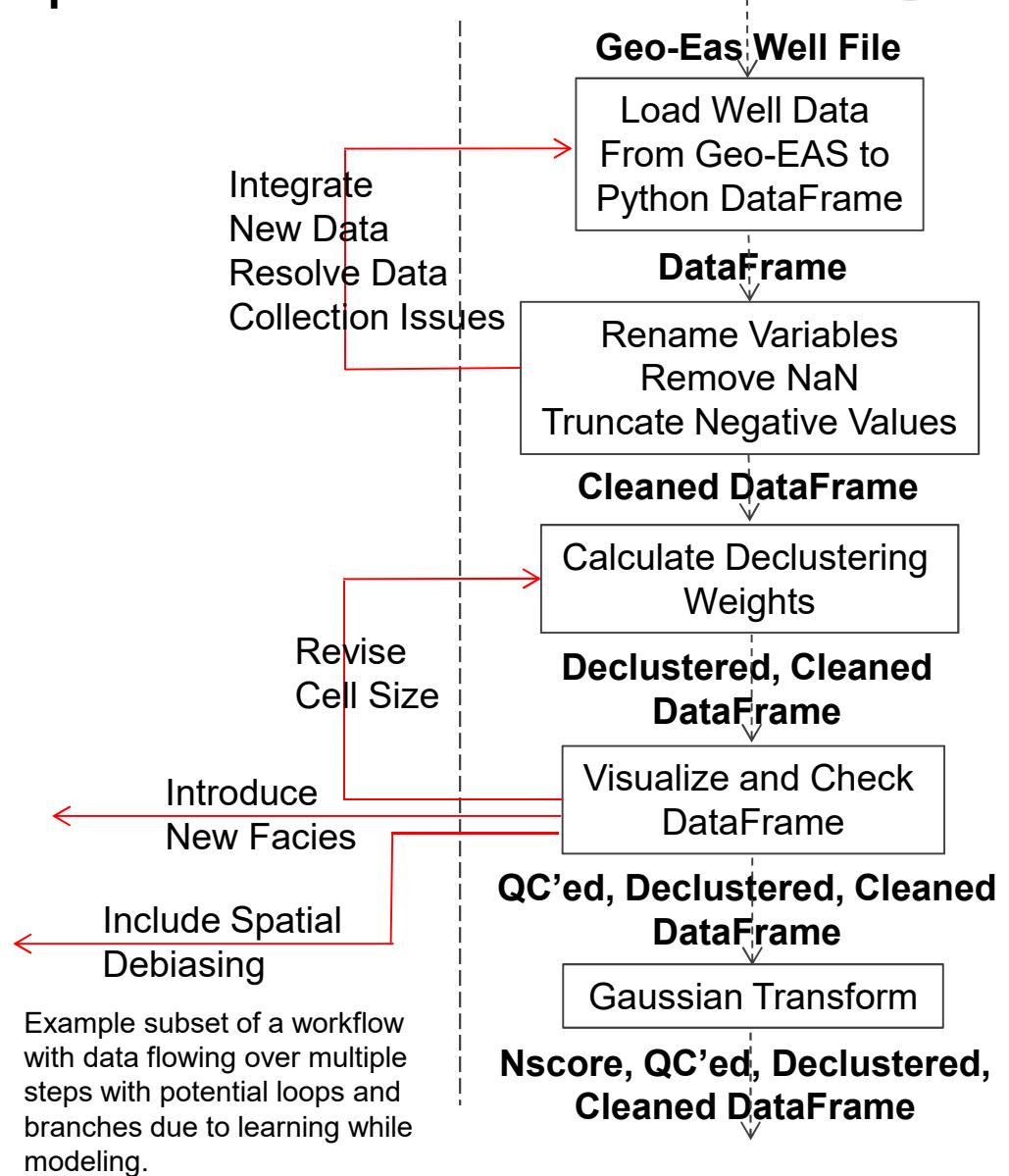
Basics of Workflow Design

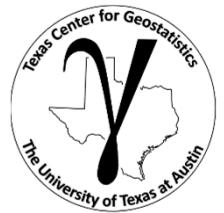
Loops and Branches

Design a Set of Steps to Accomplish the Goal, branches, loops and iteration:

It is practical to represent modeling as a linear workflow

- This is not best practice
- We learn new things with every step – **learning while modeling**
- We may refine / modify the workflow due to new information
- We may cycle back to a previous step





Basics of Workflow Design

Uncertainty

Design a Set of Steps to Accomplish the Goal, **uncertainty**:

- Data, statistical summary, models all have uncertainty
- Significant uncertainty sources must be investigated and integrated into the workflow
- The fundamental methods are :
 - *scenarios* – change the modeling decisions / inputs
 - *realizations* – hold modeling decisions and inputs constant and just change the random number seed

Porosity mean and variance P10, P50, P90

Imputed missing data as random variables.

Optimum declustering hyper parameters

Representative porosity mean, variance

Example subset of a workflow with data flowing over multiple steps with uncertainties.

Geo-Eas Well File

Load Well Data
From Geo-EAS to
Python DataFrame

DataFrame

Rename Variables
Remove NaN
Truncate Negative Values

Cleaned DataFrame

Calculate Declustering Weights

Declustered, Cleaned DataFrame

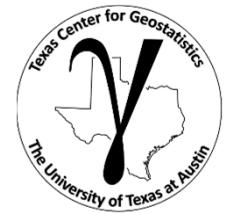
Visualize and Check DataFrame

QC'ed, Declustered, Cleaned DataFrame

Gaussian Transform

Nscore, QC'ed, Declustered, Cleaned DataFrame

Coding to Support Modeling Workflow Development



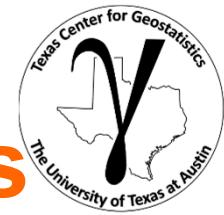
More on Software / Coding:

- This is not a coding / software workshop.
- I can't teach Python in 1 day.
- We will demonstrate well-documented workflows in Python.
- We will focus on the steps, inputs and outputs.
- Don't be concerned if you don't completely understand the code.

But if coding to solve subsurface problems is your interest:

- I teach that also!
- I have a lot of basic tutorials and example workflows to solve practical subsurface problems
- Will conduct a Data Science Bootcamp starting this summer with Professor John Foster from the Institute for Computational Engineering and Science (ICES) at the University of Texas at Austin.

Example Python Coding Demonstrations / Tutorials



Python Basics

- Numpy for arrays (ndarrays) for gridded data
- Pandas for tabulated data (DataFrames) for well data
- Tutorials available on GitHub, I'm planning to cover in class?

Tabular Data with DataFrames in Python for Geoscientists and Geo-engineers

Michael Pyrcz, University of Texas at Austin (@GeostatsGuy)



Many geoscientists and engineers struggle with getting started with **data analytics**, **machine learning** and **geostatistics** in Python, because they do not know how to handle their data. **DataFrames from the pandas package**, by Wes McKinney and current core team, is designed for high performance, easy to use tabular data structures. Here's a tutorial in Jupyter with Markdown with all the tabular structured data operations needed to build most subsurface modeling workflows. Check it out here: <https://git.io/NGRW>.

Jupyter Notebook Tutorial with Markdown Tabular Data Structures / DataFrames in Python for Engineers and Geoscientists

Michael Pyrcz, Associate Professor, University of Texas at Austin

Content: <https://GeostatsGuy/GitHubGeostatistics1/www.github.com/GitHubScholar1/book>

This is a tutorial for demonstration of Tabular Data Structures in Python. In Python, the common tool for dealing with Tabular Data Structures is the pandas Python package.

This tutorial includes the methods and operations that would commonly be required for Engineers and Geoscientists working with Tabular Data Structures in Python.

1. Data Cleaning and Checking
2. Data Melt - Inferential Data Analysis
3. Data Analytics - Building Predictive Models with Geostatistics and Machine Learning

Tabular Data Structures

In Python we can store our data in four formats: tables and arrays. For example data with single values, usually representing a single feature on a regular grid over $1 \dots n$ samples we will work with tables. For exhaustive maps and models usually representing a single feature on a regular grid over $1 \dots n$ samples we will work with arrays.

pandas package provides DataFrame object for working with data in a table and numpy package provides a convenient ndarray object for working with grids of data. In this section we will focus on DataFrame although we will often integrate a couple of them. There is another section on Gridded Data Structures that focuses on ndarrays.

If you get a package import error, you may have to first install some of these packages. This can usually be accomplished by opening up a command window on Windows and then typing `python -m pip install [package-name]`. More assistance is available with the <https://GeostatsGuy/GitHubGeostatistics1/www.github.com/GitHubScholar1/book>.

Set the working directory

I always like to do this so I don't lose file and to simply subsequent read and writes (avoid including the full address each time). Also in this case make sure to place the required (see below) data file in this directory. When we are done with this tutorial we will move the data file back to the directory.

`os.chdir("C:/Users/Pyrcz")`

Set the working directory

If you get a package import error, you may have to first install some of these packages. This can usually be accomplished by opening up a command window on Windows and then typing `python -m pip install [package-name]`. More assistance is available with the <https://GeostatsGuy/GitHubGeostatistics1/www.github.com/GitHubScholar1/book>.

Load the required libraries

The following code loads the required libraries.

`import os`
`import numpy as np`
`import pandas as pd`

If you get a package import error, you may have to first install some of these packages. This can usually be accomplished by opening up a command window on Windows and then typing `python -m pip install [package-name]`. More assistance is available with the <https://GeostatsGuy/GitHubGeostatistics1/www.github.com/GitHubScholar1/book>.

Set the working directory

If you get a package import error, you may have to first install some of these packages. This can usually be accomplished by opening up a command window on Windows and then typing `python -m pip install [package-name]`. More assistance is available with the <https://GeostatsGuy/GitHubGeostatistics1/www.github.com/GitHubScholar1/book>.

Load the required libraries

The following code loads the required libraries.

`import os`
`import numpy as np`
`import pandas as pd`
`import matplotlib.pyplot as plt`
`import seaborn as sns`

If you get a package import error, you may have to first install some of these packages. This can usually be accomplished by opening up a command window on Windows and then typing `python -m pip install [package-name]`. More assistance is available with the <https://GeostatsGuy/GitHubGeostatistics1/www.github.com/GitHubScholar1/book>.

Project Goal

Learn the basics for working with Regular Gridded Data Structures in Python to build practical workflows.

Goals

I include methods that I have found useful for building my geoscience workflows for subsurface modeling. I think they should be accessible to most geoscientists and engineers. Certainly, there are more advanced, more compact, more efficient methods to accomplish the same tasks. I need to keep the methods simple enough for the intended audience and use them to improve the related workflow.

Load the required libraries

The following code loads the required libraries.

`import os`
`import numpy as np`
`import pandas as pd`
`import matplotlib.pyplot as plt`
`import seaborn as sns`

If you get a package import error, you may have to first install some of these packages. This can usually be accomplished by opening up a command window on Windows and then typing `python -m pip install [package-name]`. More assistance is available with the <https://GeostatsGuy/GitHubGeostatistics1/www.github.com/GitHubScholar1/book>.

Project Overview

These are the functions we have included here:

1. GSI2GSIarray - load GSI2 array to regular grid data 10 or 20 to NumPy ndarray
2. Hdra2Hdra - write NumPy array to GSI2 regular grid data 10 or 20

3. project - grid 2D NumPy arrays with same parameters as GSI2 grid

Summary Statistics / Data Checking

Data Extraction

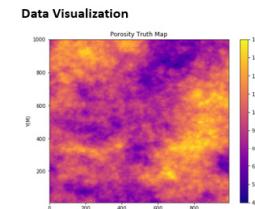
Gridded Data with ndarrays in Python for Geoscientists and Geo-engineers

Michael Pyrcz, University of Texas at Austin (@GeostatsGuy)

Many geoscientists and engineers struggle with getting started with **data analytics**, **machine learning** and **geostatistics** in Python, because they do not know how to handle their data. **ndarrays from the NumPy package**, by Jim Hugunin et al., is designed for high performance, easy to use gridded data structures. Here's a tutorial in Jupyter with Markdown with all the gridded data operations needed to build most subsurface modeling workflows.

Gridded Data with ndarrays in Python for Geoscientists and Geo-engineers

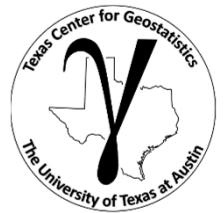
Michael Pyrcz, University of Texas at Austin (@GeostatsGuy)



Summary Statistics / Data Checking

Making Arrays

```
from scipy import stats
# read array
# = read array as 1D
# = read array as 2D
# = read array as 3D
# = read array as 4D
# = read array as 5D
# = read array as 6D
# = read array as 7D
# = read array as 8D
# = read array as 9D
# = read array as 10D
# = read array as 11D
# = read array as 12D
# = read array as 13D
# = read array as 14D
# = read array as 15D
# = read array as 16D
# = read array as 17D
# = read array as 18D
# = read array as 19D
# = read array as 20D
# = read array as 21D
# = read array as 22D
# = read array as 23D
# = read array as 24D
# = read array as 25D
# = read array as 26D
# = read array as 27D
# = read array as 28D
# = read array as 29D
# = read array as 30D
# = read array as 31D
# = read array as 32D
# = read array as 33D
# = read array as 34D
# = read array as 35D
# = read array as 36D
# = read array as 37D
# = read array as 38D
# = read array as 39D
# = read array as 40D
# = read array as 41D
# = read array as 42D
# = read array as 43D
# = read array as 44D
# = read array as 45D
# = read array as 46D
# = read array as 47D
# = read array as 48D
# = read array as 49D
# = read array as 50D
# = read array as 51D
# = read array as 52D
# = read array as 53D
# = read array as 54D
# = read array as 55D
# = read array as 56D
# = read array as 57D
# = read array as 58D
# = read array as 59D
# = read array as 60D
# = read array as 61D
# = read array as 62D
# = read array as 63D
# = read array as 64D
# = read array as 65D
# = read array as 66D
# = read array as 67D
# = read array as 68D
# = read array as 69D
# = read array as 70D
# = read array as 71D
# = read array as 72D
# = read array as 73D
# = read array as 74D
# = read array as 75D
# = read array as 76D
# = read array as 77D
# = read array as 78D
# = read array as 79D
# = read array as 80D
# = read array as 81D
# = read array as 82D
# = read array as 83D
# = read array as 84D
# = read array as 85D
# = read array as 86D
# = read array as 87D
# = read array as 88D
# = read array as 89D
# = read array as 90D
# = read array as 91D
# = read array as 92D
# = read array as 93D
# = read array as 94D
# = read array as 95D
# = read array as 96D
# = read array as 97D
# = read array as 98D
# = read array as 99D
# = read array as 100D
# = read array as 101D
# = read array as 102D
# = read array as 103D
# = read array as 104D
# = read array as 105D
# = read array as 106D
# = read array as 107D
# = read array as 108D
# = read array as 109D
# = read array as 110D
# = read array as 111D
# = read array as 112D
# = read array as 113D
# = read array as 114D
# = read array as 115D
# = read array as 116D
# = read array as 117D
# = read array as 118D
# = read array as 119D
# = read array as 120D
# = read array as 121D
# = read array as 122D
# = read array as 123D
# = read array as 124D
# = read array as 125D
# = read array as 126D
# = read array as 127D
# = read array as 128D
# = read array as 129D
# = read array as 130D
# = read array as 131D
# = read array as 132D
# = read array as 133D
# = read array as 134D
# = read array as 135D
# = read array as 136D
# = read array as 137D
# = read array as 138D
# = read array as 139D
# = read array as 140D
# = read array as 141D
# = read array as 142D
# = read array as 143D
# = read array as 144D
# = read array as 145D
# = read array as 146D
# = read array as 147D
# = read array as 148D
# = read array as 149D
# = read array as 150D
# = read array as 151D
# = read array as 152D
# = read array as 153D
# = read array as 154D
# = read array as 155D
# = read array as 156D
# = read array as 157D
# = read array as 158D
# = read array as 159D
# = read array as 160D
# = read array as 161D
# = read array as 162D
# = read array as 163D
# = read array as 164D
# = read array as 165D
# = read array as 166D
# = read array as 167D
# = read array as 168D
# = read array as 169D
# = read array as 170D
# = read array as 171D
# = read array as 172D
# = read array as 173D
# = read array as 174D
# = read array as 175D
# = read array as 176D
# = read array as 177D
# = read array as 178D
# = read array as 179D
# = read array as 180D
# = read array as 181D
# = read array as 182D
# = read array as 183D
# = read array as 184D
# = read array as 185D
# = read array as 186D
# = read array as 187D
# = read array as 188D
# = read array as 189D
# = read array as 190D
# = read array as 191D
# = read array as 192D
# = read array as 193D
# = read array as 194D
# = read array as 195D
# = read array as 196D
# = read array as 197D
# = read array as 198D
# = read array as 199D
# = read array as 200D
# = read array as 201D
# = read array as 202D
# = read array as 203D
# = read array as 204D
# = read array as 205D
# = read array as 206D
# = read array as 207D
# = read array as 208D
# = read array as 209D
# = read array as 210D
# = read array as 211D
# = read array as 212D
# = read array as 213D
# = read array as 214D
# = read array as 215D
# = read array as 216D
# = read array as 217D
# = read array as 218D
# = read array as 219D
# = read array as 220D
# = read array as 221D
# = read array as 222D
# = read array as 223D
# = read array as 224D
# = read array as 225D
# = read array as 226D
# = read array as 227D
# = read array as 228D
# = read array as 229D
# = read array as 230D
# = read array as 231D
# = read array as 232D
# = read array as 233D
# = read array as 234D
# = read array as 235D
# = read array as 236D
# = read array as 237D
# = read array as 238D
# = read array as 239D
# = read array as 240D
# = read array as 241D
# = read array as 242D
# = read array as 243D
# = read array as 244D
# = read array as 245D
# = read array as 246D
# = read array as 247D
# = read array as 248D
# = read array as 249D
# = read array as 250D
# = read array as 251D
# = read array as 252D
# = read array as 253D
# = read array as 254D
# = read array as 255D
# = read array as 256D
# = read array as 257D
# = read array as 258D
# = read array as 259D
# = read array as 260D
# = read array as 261D
# = read array as 262D
# = read array as 263D
# = read array as 264D
# = read array as 265D
# = read array as 266D
# = read array as 267D
# = read array as 268D
# = read array as 269D
# = read array as 270D
# = read array as 271D
# = read array as 272D
# = read array as 273D
# = read array as 274D
# = read array as 275D
# = read array as 276D
# = read array as 277D
# = read array as 278D
# = read array as 279D
# = read array as 280D
# = read array as 281D
# = read array as 282D
# = read array as 283D
# = read array as 284D
# = read array as 285D
# = read array as 286D
# = read array as 287D
# = read array as 288D
# = read array as 289D
# = read array as 290D
# = read array as 291D
# = read array as 292D
# = read array as 293D
# = read array as 294D
# = read array as 295D
# = read array as 296D
# = read array as 297D
# = read array as 298D
# = read array as 299D
# = read array as 300D
# = read array as 301D
# = read array as 302D
# = read array as 303D
# = read array as 304D
# = read array as 305D
# = read array as 306D
# = read array as 307D
# = read array as 308D
# = read array as 309D
# = read array as 310D
# = read array as 311D
# = read array as 312D
# = read array as 313D
# = read array as 314D
# = read array as 315D
# = read array as 316D
# = read array as 317D
# = read array as 318D
# = read array as 319D
# = read array as 320D
# = read array as 321D
# = read array as 322D
# = read array as 323D
# = read array as 324D
# = read array as 325D
# = read array as 326D
# = read array as 327D
# = read array as 328D
# = read array as 329D
# = read array as 330D
# = read array as 331D
# = read array as 332D
# = read array as 333D
# = read array as 334D
# = read array as 335D
# = read array as 336D
# = read array as 337D
# = read array as 338D
# = read array as 339D
# = read array as 340D
# = read array as 341D
# = read array as 342D
# = read array as 343D
# = read array as 344D
# = read array as 345D
# = read array as 346D
# = read array as 347D
# = read array as 348D
# = read array as 349D
# = read array as 350D
# = read array as 351D
# = read array as 352D
# = read array as 353D
# = read array as 354D
# = read array as 355D
# = read array as 356D
# = read array as 357D
# = read array as 358D
# = read array as 359D
# = read array as 360D
# = read array as 361D
# = read array as 362D
# = read array as 363D
# = read array as 364D
# = read array as 365D
# = read array as 366D
# = read array as 367D
# = read array as 368D
# = read array as 369D
# = read array as 370D
# = read array as 371D
# = read array as 372D
# = read array as 373D
# = read array as 374D
# = read array as 375D
# = read array as 376D
# = read array as 377D
# = read array as 378D
# = read array as 379D
# = read array as 380D
# = read array as 381D
# = read array as 382D
# = read array as 383D
# = read array as 384D
# = read array as 385D
# = read array as 386D
# = read array as 387D
# = read array as 388D
# = read array as 389D
# = read array as 390D
# = read array as 391D
# = read array as 392D
# = read array as 393D
# = read array as 394D
# = read array as 395D
# = read array as 396D
# = read array as 397D
# = read array as 398D
# = read array as 399D
# = read array as 400D
# = read array as 401D
# = read array as 402D
# = read array as 403D
# = read array as 404D
# = read array as 405D
# = read array as 406D
# = read array as 407D
# = read array as 408D
# = read array as 409D
# = read array as 410D
# = read array as 411D
# = read array as 412D
# = read array as 413D
# = read array as 414D
# = read array as 415D
# = read array as 416D
# = read array as 417D
# = read array as 418D
# = read array as 419D
# = read array as 420D
# = read array as 421D
# = read array as 422D
# = read array as 423D
# = read array as 424D
# = read array as 425D
# = read array as 426D
# = read array as 427D
# = read array as 428D
# = read array as 429D
# = read array as 430D
# = read array as 431D
# = read array as 432D
# = read array as 433D
# = read array as 434D
# = read array as 435D
# = read array as 436D
# = read array as 437D
# = read array as 438D
# = read array as 439D
# = read array as 440D
# = read array as 441D
# = read array as 442D
# = read array as 443D
# = read array as 444D
# = read array as 445D
# = read array as 446D
# = read array as 447D
# = read array as 448D
# = read array as 449D
# = read array as 450D
# = read array as 451D
# = read array as 452D
# = read array as 453D
# = read array as 454D
# = read array as 455D
# = read array as 456D
# = read array as 457D
# = read array as 458D
# = read array as 459D
# = read array as 460D
# = read array as 461D
# = read array as 462D
# = read array as 463D
# = read array as 464D
# = read array as 465D
# = read array as 466D
# = read array as 467D
# = read array as 468D
# = read array as 469D
# = read array as 470D
# = read array as 471D
# = read array as 472D
# = read array as 473D
# = read array as 474D
# = read array as 475D
# = read array as 476D
# = read array as 477D
# = read array as 478D
# = read array as 479D
# = read array as 480D
# = read array as 481D
# = read array as 482D
# = read array as 483D
# = read array as 484D
# = read array as 485D
# = read array as 486D
# = read array as 487D
# = read array as 488D
# = read array as 489D
# = read array as 490D
# = read array as 491D
# = read array as 492D
# = read array as 493D
# = read array as 494D
# = read array as 495D
# = read array as 496D
# = read array as 497D
# = read array as 498D
# = read array as 499D
# = read array as 500D
# = read array as 501D
# = read array as 502D
# = read array as 503D
# = read array as 504D
# = read array as 505D
# = read array as 506D
# = read array as 507D
# = read array as 508D
# = read array as 509D
# = read array as 510D
# = read array as 511D
# = read array as 512D
# = read array as 513D
# = read array as 514D
# = read array as 515D
# = read array as 516D
# = read array as 517D
# = read array as 518D
# = read array as 519D
# = read array as 520D
# = read array as 521D
# = read array as 522D
# = read array as 523D
# = read array as 524D
# = read array as 525D
# = read array as 526D
# = read array as 527D
# = read array as 528D
# = read array as 529D
# = read array as 530D
# = read array as 531D
# = read array as 532D
# = read array as 533D
# = read array as 534D
# = read array as 535D
# = read array as 536D
# = read array as 537D
# = read array as 538D
# = read array as 539D
# = read array as 540D
# = read array as 541D
# = read array as 542D
# = read array as 543D
# = read array as 544D
# = read array as 545D
# = read array as 546D
# = read array as 547D
# = read array as 548D
# = read array as 549D
# = read array as 550D
# = read array as 551D
# = read array as 552D
# = read array as 553D
# = read array as 554D
# = read array as 555D
# = read array as 556D
# = read array as 557D
# = read array as 558D
# = read array as 559D
# = read array as 560D
# = read array as 561D
# = read array as 562D
# = read array as 563D
# = read array as 564D
# = read array as 565D
# = read array as 566D
# = read array as 567D
# = read array as 568D
# = read array as 569D
# = read array as 570D
# = read array as 571D
# = read array as 572D
# = read array as 573D
# = read array as 574D
# = read array as 575D
# = read array as 576D
# = read array as 577D
# = read array as 578D
# = read array as 579D
# = read array as 580D
# = read array as 581D
# = read array as 582D
# = read array as 583D
# = read array as 584D
# = read array as 585D
# = read array as 586D
# = read array as 587D
# = read array as 588D
# = read array as 589D
# = read array as 590D
# = read array as 591D
# = read array as 592D
# = read array as 593D
# = read array as 594D
# = read array as 595D
# = read array as 596D
# = read array as 597D
# = read array as 598D
# = read array as 599D
# = read array as 600D
# = read array as 601D
# = read array as 602D
# = read array as 603D
# = read array as 604D
# = read array as 605D
# = read array as 606D
# = read array as 607D
# = read array as 608D
# = read array as 609D
# = read array as 610D
# = read array as 611D
# = read array as 612D
# = read array as 613D
# = read array as 614D
# = read array as 615D
# = read array as 616D
# = read array as 617D
# = read array as 618D
# = read array as 619D
# = read array as 620D
# = read array as 621D
# = read array as 622D
# = read array as 623D
# = read array as 624D
# = read array as 625D
# = read array as 626D
# = read array as 627D
# = read array as 628D
# = read array as 629D
# = read array as 630D
# = read array as 631D
# = read array as 632D
# = read array as 633D
# = read array as 634D
# = read array as 635D
# = read array as 636D
# = read array as 637D
# = read array as 638D
# = read array as 639D
# = read array as 640D
# = read array as 641D
# = read array as 642D
# = read array as 643D
# = read array as 644D
# = read array as 645D
# = read array as 646D
# = read array as 647D
# = read array as 648D
# = read array as 649D
# = read array as 650D
# = read array as 651D
# = read array as 652D
# = read array as 653D
# = read array as 654D
# = read array as 655D
# = read array as 656D
# = read array as 657D
# = read array as 658D
# = read array as 659D
# = read array as 660D
# = read array as 661D
# = read array as 662D
# = read array as 663D
# = read array as 664D
# = read array as 665D
# = read array as 666D
# = read array as 667D
# = read array as 668D
# = read array as 669D
# = read array as 670D
# = read array as 671D
# = read array as 672D
# = read array as 673D
# = read array as 674D
# = read array as 675D
# = read array as 676D
# = read array as 677D
# = read array as 678D
# = read array as 679D
# = read array as 680D
# = read array as 681D
# = read array as 682D
# = read array as 683D
# = read array as 684D
# = read array as 685D
# = read array as 686D
# = read array as 687D
# = read array as 688D
# = read array as 689D
# = read array as 690D
# = read array as 691D
# = read array as 692D
# = read array as 693D
# = read array as 694D
# = read array as 695D
# = read array as 696D
# = read array as 6
```



Reminder: Benefit of Coding

Reasons All Geoscientists and Engineers Should Learn to Code

Transparency – *no compiler accepts hand waiving!* Coding forces your logic to be uncovered for any other scientist or engineer to review.

Reproducibility – *run it, get an answer, hand it over, run it, get the same answer.* This is a main principle of the scientific method.

Quantification – *programs need numbers.* Feed the program and discover new ways to look at the world.

Open-source – *leverage a world of brilliance.* Check out packages, snippets and be amazed with what great minds have freely shared.

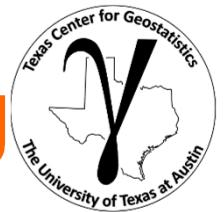
Break Down Barriers – *don't throw it over the fence.* Sit at the table with the developers and share more of your subject matter expertise for a better product.

Deployment – *share it with others and multiply the impact.* Performance metrics or altruism, your good work benefits many others.

Efficiency – *minimize the boring parts of the job.* Build a suite of scripts for automation of common tasks and spend more time doing science and engineering!

Always Time to Do it Again! – *how many times did you only do it once?* It probably takes 2-4 times as long to script and automate a workflow. Usually worth it.

Be Like Us – *it will change you.* Users feel limited, programmers truly harness the power of their applications and hardware.

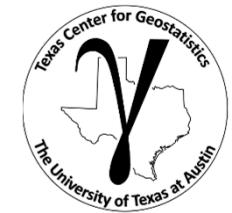


Aside: Benefit of Coding

Reasons All Geoscientists and Engineers Should Learn to Code Some Caveats

1. Any type of coding, scripting, workflow automation matched to your working environment is great. We don't all need to be C++ experts.
2. I respect the experience component of geoscience and engineering expertise. This is beyond coding and is essential to workflow logic development, best use of data etc.
3. Some expert judgement will remain subjective and not completely reproducible. I'm not advocating for the geoscientist or engineer being replaced by a computer.

The Subsurface Modeling Steps



- We could spend more time together! I do a lot of training, e.g. 3 day course:
 - Some Prerequisites
 - Data Preparation
 - Univariate and Multivariate Analysis
 - Spatial Analysis
 - Estimation and Trend Modeling
 - Stochastic Simulation
 - Uncertainty Analysis
 - Model Checking
 - Decision Making
- For each section
 - Lectures and demos
 - Subsurface inference and modeling
 - Completion of project update documentation and presentation

Introduction

Prerequisites

Data Preparation

Univariate Analysis

Multivariate Analysis

Spatial Characterization

Spatial Estimation

Spatial Simulation

Uncertainty Analysis

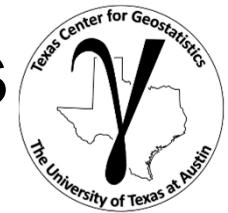
Model Checking

Decision Making

Tailored to geoscientists, engineers, managers etc. Talk to the Midland, TX Geoscientists.

Data Analytics, Geostatistics and Machine Learning

Fundamental Concepts



Fundamental Concepts . .

- What is Subsurface Modeling?
- Modeling Goals
- Modeling Strategies
- Workflow Development

Introduction

Modeling Prerequisites

Fundamental Concepts

Probability

Sparse Data Workflows

Spatial Estimation

Spatial Uncertainty

Multivariate, Spatial

Novel Workflows

Conclusions

Instructor: Michael Pyrcz, the University of Texas at Austin