



Applied Machine Learning

Fundamentals and Geostatistics

Michael Pyrcz and Didi Ooi

 @geostatsguy and @didiooi



Lecture (and Course) Outline

The agenda of the day...

Outline of this lecture

- Motivation / Goals
- Class Description / Objectives
- Challenges in Geoscience
- Python: A new geoscience toolkit?
- Resources

Class outline

Introduction

Data Analytics

Inferential Methods

Predictive Methods

Advanced Methods

Conclusions

What Will You Learn?

Expectations and hope

Goals:

- Awareness
- Opportunities in Data Science and Machine Learning

Our hope:

- Inspire you to start using new tools
- Impact your capabilities

Who we are? Michael Pyrcz

Pyrcz: is pronounced "perch"

I'm New: new to UT PGE, started August 2017.

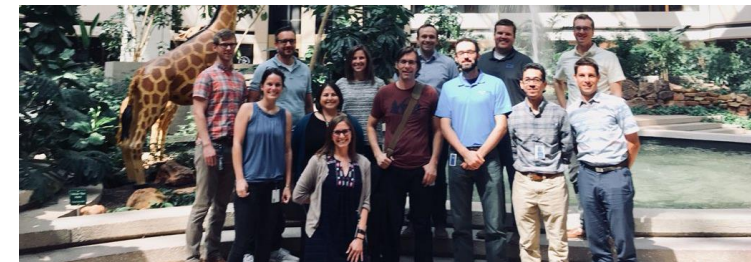
1. **I have practical experience:** over 17 years of experience in consulting, teaching and industrial R&D in statistical modeling, reservoir modeling and uncertainty characterization.
2. **Flexible:** got ideas, feedback to improve the learning opportunities. Let's work together to reach our learning objective.
3. **An Engineer, but:** My B.Sc. was Mining Engineering, my M.Sc. started as Geotechnical Engineering (then skipped to Ph.D.) and my Ph.D. was in Quantitative Geology. I spent 13 years in Earth Science R&D working with geological and geophysical reservoir modeling. I speak geo.



Spring 2018 Class of Introduction to Geostatistics



Oil and Gas University, Florence, Italy



Anadarko, Midland, TX

Who we are? Michael Pyrcz

Pyrcz: is pronounced "perch"



AAPG SEPM Panel Discussion on Modeling



CPGE Webinar on Big Data

Active in Outreach, Social Media and Professional Organizations

- Associate editor with Computers and Geosciences, editorial board of Mathematical Geosciences for the International Association of Mathematical Geosciences
- Program chair for SPE Data Analytics Technical Section
- Associate editor with Computers and Geosciences
- Author of the textbook "Geostatistical Reservoir Modeling"
- Board member for Mathematical Geosciences
- GeostatsGuy on Twitter, GitHub GeostatsGuy Lectures on YouTube

I'm committed to supporting / partnering for development opportunities of working professionals

Who we are? Didi Ooi

Ooi: is pronounced "Oo-ee"



Passionate in innovating and hacking reservoir characterization using geology AND emerging technologies!

1. **Practical experience:** over 7 years of research and industrial experience in sedimentology, numerical modeling, water-rock interaction and subsurface data science (hacking!)
2. **A Hybrid Geologist:** Yes, both quantitative and “arm-waving”. BSc, PhD in University of Bristol, UK. Carbonate and Evaporites. Areal experience in the Middle East, Greece, SE Asia, UK. Worked with Shell and ExxonMobil. Now, emerging Technology Geologist with AAET, Anadarko (and loving it!)



Who we are? Didi Ooi

Ooi: is pronounced "Oo-ee"

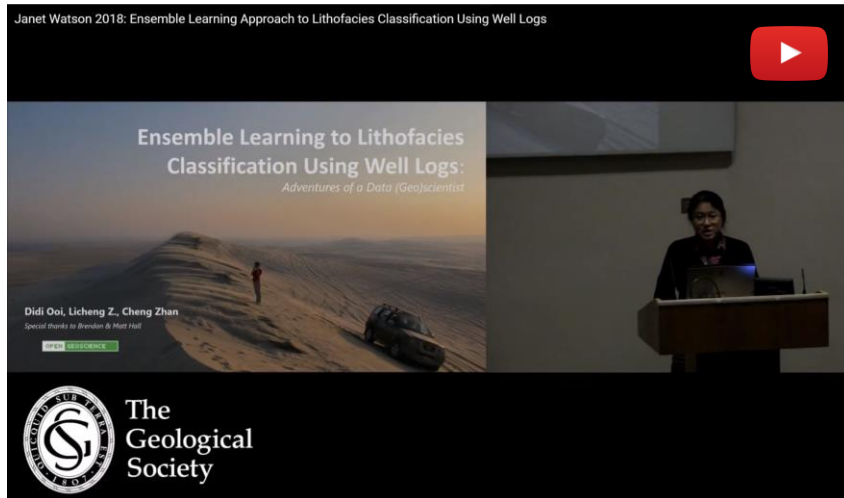
Passionate in innovating and hacking reservoir characterization using geology AND emerging technologies!

3. **Outreach:** SEPM ISGC Scientific Advisor, Chair of AAPG ML Unsession, Houston ML Meetup Co-Organizer

4. **Data Science started as my hobby:** 3 times winner in subsurface hackathons, projects in github.com/didiooi



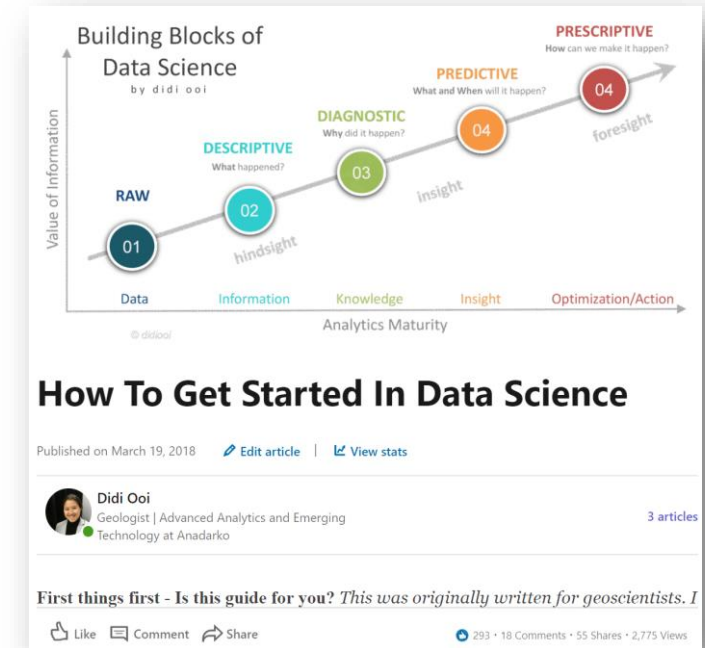
Lead Instructor in Bristol for grad students



Big Data in Geoscience Conference, UK



Geophysics Hackathon 2017, 2018 winner



The article that inspired outreach

Introduction

Your turn to tell us about you and your expectations

Short Introductions

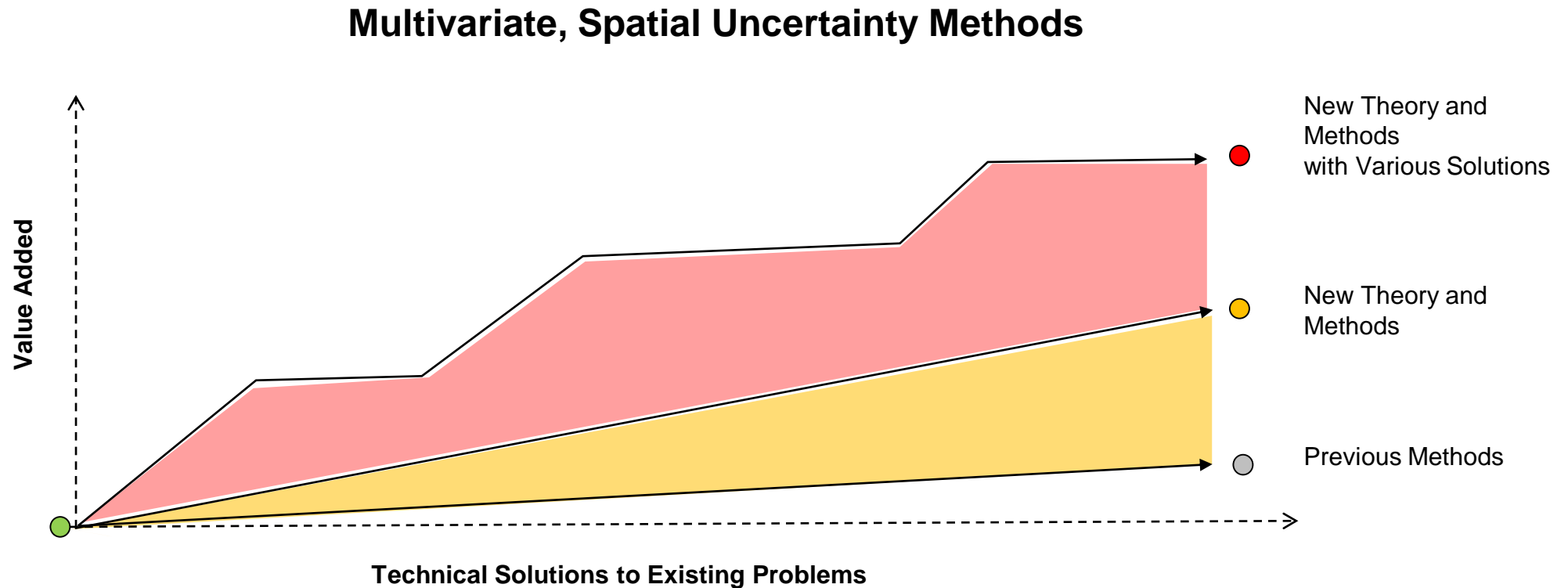
- **Name**
- **Role**
- **Expectations from this course**

What Will You Learn?

Theory, methods and solutions

This time is an investment in learning

- Build operational capability
- Provide incremental value.



What Will You Learn

Reaching our goals

Today we will:

- Build up from zero
- Provide an overview of the methods
- Demonstrate well-documented, practical workflows

Of course, full workflow development would require time to investigate the problem and available data.

p/s: Michael Pyrcz has a lot of content for more advanced topics!

“If I have erred it is on the side of simplicity.”

Data Science Matrix (Simplified)

What do I want to find out?

Supervised Learning

Unsupervised Learning

Discrete

I want to
predict
categories

I want to
discover
structure

Continuous

I want to
predict
values

I want my
dataset
decluttered

Data Science Matrix (Simplified)

What techniques do I use?

	<i>Supervised Learning</i>	<i>Unsupervised Learning</i>
<i>Discrete</i>	CLASSIFICATION	CLUSTERING
<i>Continuous</i>	REGRESSION	DIMENSIONALITY REDUCTION

**Does not include domain-specific adaptations such as NLP and CV*

Data Science Matrix (Simplified)

What algorithms are there for me?

	Supervised Learning	Unsupervised Learning
Discrete	Logistic Regression Classification Tree Support Vector Machine Naïve Bayes	K-Means Affinity Propagation Hierarchical/Agglomerative DBSCAN
Continuous	Linear Regression Decision Tree (Ensembles) Poisson	Reinforcement Learning Principal Component (PCA) Linear Discriminant (LDA) IsoMap Truncated SVD t-SNE Autoencoders

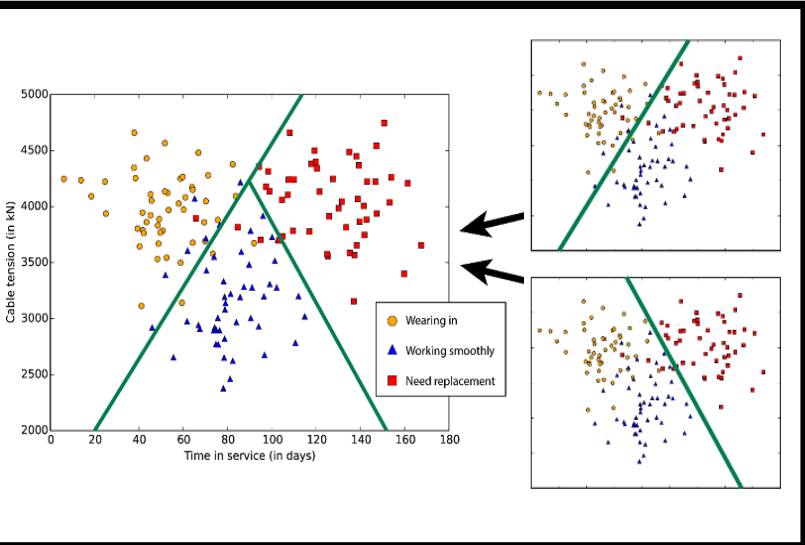
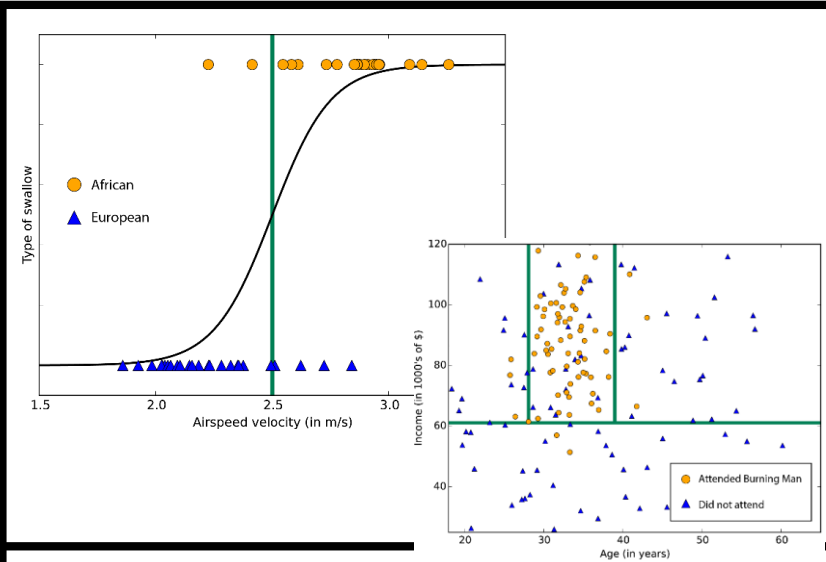
Data Science Matrix (Simplified)

Algorithms

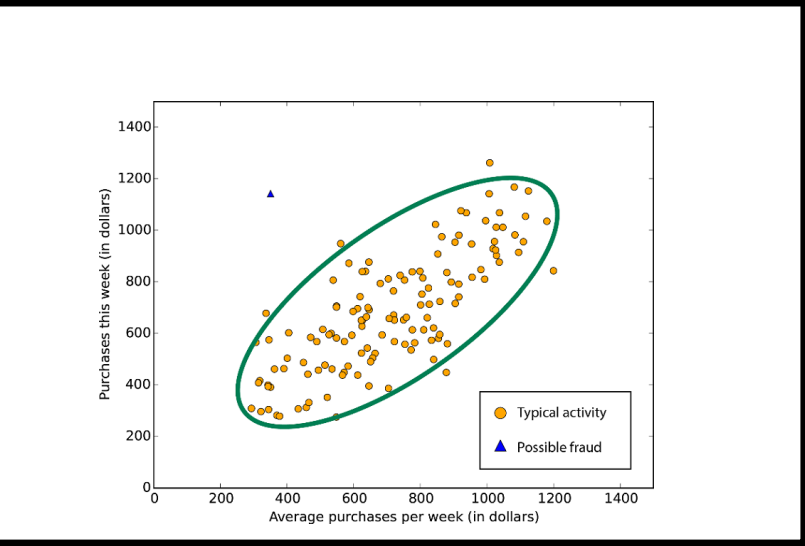
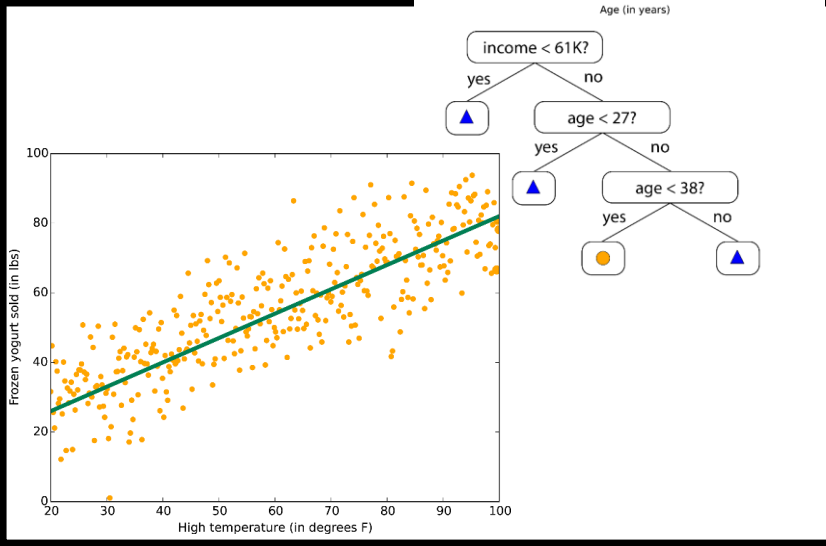
Supervised Learning

Unsupervised Learning

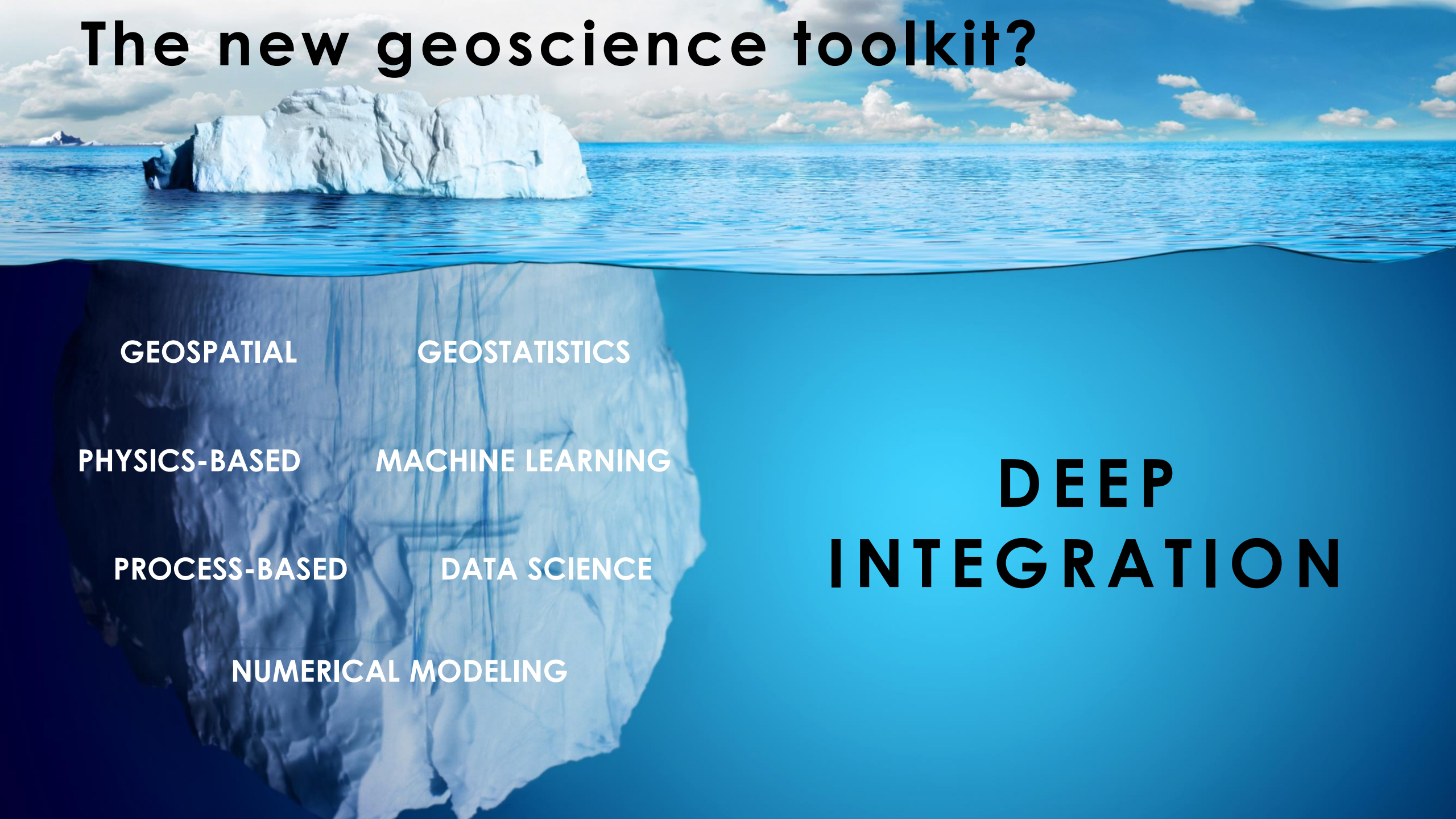
Discrete



Continuous



The new geoscience toolkit?



GEOSPATIAL

GEOSTATISTICS

PHYSICS-BASED

MACHINE LEARNING

PROCESS-BASED

DATA SCIENCE

NUMERICAL MODELING

**DEEP
INTEGRATION**

Data in oil and gas

1. Measured (raw)
2. Observed (raw)
3. Processed and Corrected
4. Interpretations
5. Models from Interpretations
6. Predictions from models

Challenges in Geoscience

Schematic Diagram

1. Primary vs Secondary
2. Multi-scale
3. Multi-resolution
4. Spatio-temporal
5. Curse of Dimensionality
6. Heterogeneity
7. Noise vs Signal
8. Uncertainty
9. Sampling bias
10. Lack of Ground Truth

Python: The new geoscientist toolkit

Why Python language?

- Is very powerful, the most resources and assistance
- Packages allow us to put together workflows with limited old-fashioned 'coding'
- Leverage the world's brilliance
- Can help automate workflows in O&G softwares! ArcGIS, IP, Techlog,

Certainly there's a phenomenon around open source. You know free software will be a vibrant area. 'There will be a lot of neat things that get done there.'

- Bill Gates

'20 years with C++ and FORTRAN, but with Python I code less, but get more done.'

- Michael Pyrcz

Python: The new geoscientist toolkit

Jupyter Notebooks

- Workflows that integrate blocks of code, documentation, results

GeostatsPy: Monte Carlo Simulation for Subsurface Data Analytics in Python

Michael Pircz, Associate Professor, University of Texas at Austin

[Twitter](#) | [GitHub](#) | [Website](#) | [Google Scholar](#) | [Book](#) | [YouTube](#) | [LinkedIn](#)

PGE 383 Exercise: Monte Carlo Simulation for Subsurface Data Analytics in Python

Here's a simple workflow, demonstration of Monte Carlo simulation for subsurface uncertainty modeling workflows. This should help you get started with building subsurface models that integrate uncertainty sources.

Monte Carlo Simulation

Definition: random sampling from a distribution

Procedure:

1. Model the representative distribution (CDF)

2. Draw a random value from a uniform [0,1] distribution (p-value)

3. Apply the inverse of the CDF to calculate the associated realization

In practice, Monte Carlo simulation refers to the workflow with multiple realizations drawn to build an uncertainty model.

$$X^{\ell} = F_X^{-1}(p^{\ell}), \forall \ell = 1, \dots, L$$

where X^{ℓ} is the realization of the variable X drawn from its CDF, F_X , with cumulative probability, p-value, p^{ℓ} .

It would be trivial to apply Monte Carlo simulation to a single variable, after many realizations one would get back the original distribution. The general approach is to:

1. Model all distributions for the input, variables of interest F_{X_1}, \dots, F_{X_m} .

2. For each realization draw $p_1^{\ell}, \dots, p_m^{\ell}$, p-values

3. Apply the inverse of each distribution to calculate a realization of each variable, $X_j^{\ell} = F_{X_j}^{-1}(p_j^{\ell}), \forall j = 1, \dots, m$ variables.

4. Apply each set of variables for a ℓ realization to the transfer function to calculate the output realization, $Y^{\ell} = F(X_1^{\ell}, \dots, X_m^{\ell})$.

$$Y^{\ell} = F(X_1^{\ell}, \dots, X_m^{\ell}), \forall \ell = 1, \dots, L$$

Monte Carlo Simulation (MCS) is extremely powerful

• Possible to easily simulate uncertainty models for complicated systems

• Simulations are conducted by drawing values at random from specified uncertainty distributions for each variable

• A single realization of each variable, $X_1^{\ell}, X_2^{\ell}, \dots, X_m^{\ell}$, is applied to the transfer function to calculate the realization of the variable of interest (output, decision criteria):

$$Y^{\ell} = F(X_1^{\ell}, \dots, X_m^{\ell}), \forall \ell = 1, \dots, L$$

• The MCS method builds empirical uncertainty models by random sampling

Let's take a simple example, OIP is oil-in-place calculated as the product of reservoir volume, V , average porosity, $\bar{\phi}$, and oil saturation, \bar{S}_o .

$$OIP^{\ell} = V^{\ell} \cdot \bar{\phi}^{\ell} \cdot \bar{S}_o^{\ell}, \forall \ell = 1, \dots, L$$

It would be difficult to directly calculate the OIP distribution as a combination of all these different distributions.

• The distributions could all have different forms (parametric or non-parametric)

• We use MCS to empirically work this out by sampling

• Repeat to calculate enough realizations for analysis.

Let's set the minimum and maximum values for plotting.

```
apor_min = 0.1;apor_max = 0.2 # average porosity min and max
vol_min = 0.0;vol_max = 40000000 # vol min and max
```

In the NumPy package we have handy methods for Monte Carlo simulation from parametric distributions. We can actually draw all L realizations at once for each variable and store them in ndarrays (each ndarray with realizations $\ell = 1, \dots, L$).

```
apor = np.random.normal(apor_mean,apor_stddev,size=L) # average porosity MCS simulation L times and store in array
vol = np.random.lognormal(vol_mu,vol_sigma,size=L) # volume ...
so = np.random.uniform(so_min,so_max,size=L) # saturation oil
```

Let's plot the distributions of the realizations of each variable to make sure the match the form of the parametric distributions that we selected.

```
plt.subplot(131)
GSLIB.hist_st(apor,apor_min,apor_max,log=False,cumul=False,bins=50,weights=None,xlabel="Average Porosity (fraction)",title="Average Porosity Realizations")
plt.ylim(0,60)
```

```
plt.subplot(132)
GSLIB.hist_st(vol,vol_min,vol_max,log=False,cumul=False,bins=50,weights=None,xlabel="Volume (m^3)",title="Average Volume Realizations")
plt.ylim(0,200)
```

```
plt.subplot(133)
GSLIB.hist_st(so,so_min,so_max,log=False,cumul=False,bins=50,weights=None,xlabel="Saturation Oil (fraction)",title="Average Saturation Oil Realizations")
plt.ylim(0,40)
```

```
plt.subplots_adjust(left=0.0,bottom=0.0,right=2.0,top=1.2,wspace=0.2,hspace=0.2)
plt.show()
```

This looks good, the shapes are Gaussian, lognormal and uniform and the central tendency and dispersion make sense given the parameters that we selected.

Now we can use broadcast methods to calculate the output realizations of OIP , based on this equation.

$$OIP^{\ell} = V^{\ell} \cdot \bar{\phi}^{\ell} \cdot \bar{S}_o^{\ell} \cdot 6.29 \quad \forall \ell = 1, \dots, L$$

where 0.20 bbl/s/m^3 .

Joint, Conditional and Marginals

We can use kernel density estimation to estimate the joint probabilities density function (pdf) for the paired data, a 2D pdf! We could use this to estimate any required joint, marginal and conditional probability (care must be taken with normalization). Let's use the seaborn package 'kdeplot' function to estimate the joint pdf for porosity and acoustic impedance.

```
ax = sns.kdeplot(df['AI'].values,df['Porosity'].values,shade=True,n_levels=10,cmap=cmap,bar=True,shade_lowest=False)
ax.set_xlabel('Acoustic Impedance (m/s x g/cm^3)'); ax.set_ylabel('Porosity (fraction)'); ax.set_title('Porosity vs. Acoustic Impedance')
Text(0.5,1,'Porosity vs. Acoustic Impedance')
```

I think it is useful to visualize the joint pdfs with the marginal pdfs on a single plot. We can use seaborn's 'jointplot' to accomplish this.

```
ax = sns.jointplot('AI','Porosity',df,kind='kde',shade=False,n_levels=10,cmap=cmap,shade_lowest=True);
```

- Work with a variety of kernels (Python, R, C, JavaScript, etc.)
- Make and deploy professional workflows with Markdown docs
- Use containers and run online (e.g. Docker)

Python: The new geoscientist toolkit

Useful Python libraries

Geostatspy

- Set of Python functions for most of the required workflow steps
- Much is reimplemented in Python
- Package written by Michael Pyrcz - we will tailor, augment to support training
- Open-source: anyone can use it
- Free for any use
- Download it from PyPi with:

'pip install geostatspy'

Others: scikit-learn, pyclustering

Project description



GeostatsPy Package

The GeostatsPy Package brings GSLIB: Geostatistical Library (Deutsch and Journel, 1998) functions to Python. GSLIB is extremely robust and practical code for building spatial modeling workflows. I specifically wanted it in Python to support my students in my Data Analytics, Geostatistics and Machine Learning courses. I find my students benefit from hands-on opportunities, infact it is hard to imagine teaching these topics without providing the opportunity to handle the numerical methods and build workflows.

This package includes 2 parts:

1. geostatspy.gslib includes low tech wrappers of GSLIB functionality (note: some functions require access to GSLIB executables)
2. geostatspy.geostats includes GSLIB functions rewritten in Python.

Package Inventory

Here's a list and some details on each of the functions available.

geostatspy.gslib Functions

Utilities to support moving between Python DataFrames and ndarrays, and Data Tables, Gridded Data and Models in Geo-EAS file format (standard to GSLIB):

1. ndarray2GSLIB - utility to convert 1D or 2D numpy ndarray to a GSLIB Geo-EAS file for use with GSLIB methods
2. GSLIB2ndarray - utility to convert GSLIB Geo-EAS files to a 1D or 2D numpy ndarray for use with Python methods

Python: The new geoscientist toolkit

Reasons to learn some coding

- **Transparency** – *no compiler accepts hand-waving!* 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.

Python: The new geoscientist toolkit

More on coding...

This is not a coding / software workshop

- We cannot teach Python in 1 day!
- Michael Pyrcz has partnered with tech company to provide data analytics and coding training for oil and gas professionals

<https://daytum.org/>

Expectations:

- Appreciation for what can be done
- We will show some code Michael has prepared to highlight how easy it is to get things done

Python: The new geoscientist toolkit

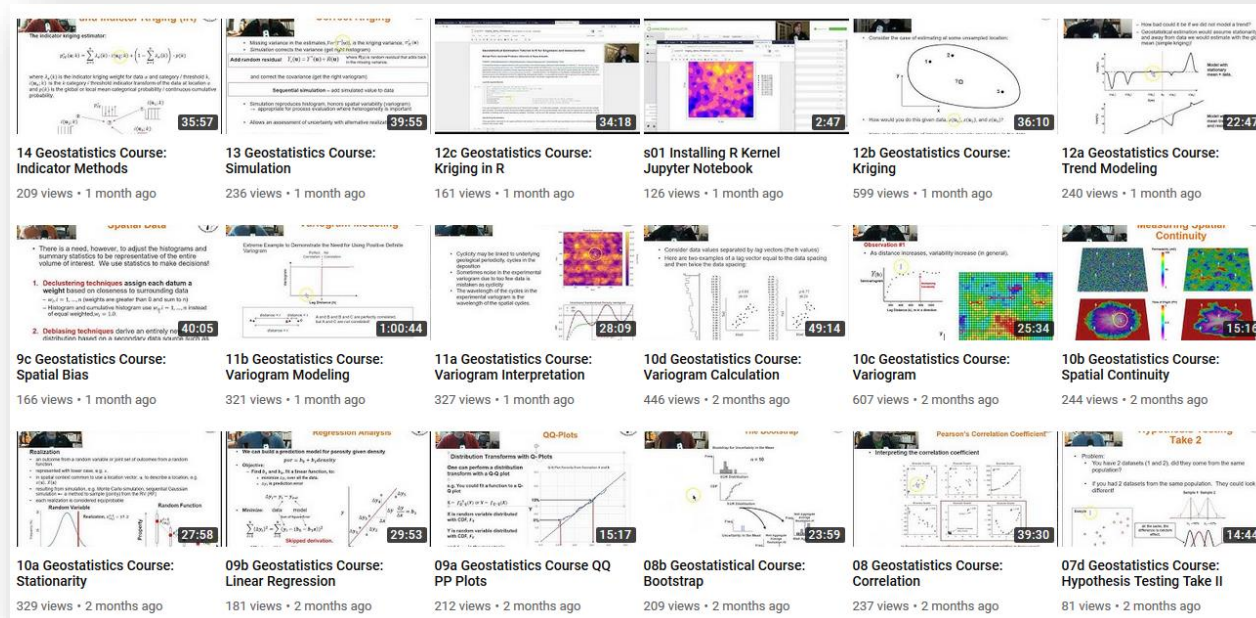
Caveats to previous reasons for coding

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.

Resources outside this (very) short course

during and after workshop

1. Lecture material in .pdf format.
2. All lectures, demos and workflows from the Michael Pyrcz's undergraduate class are available to you (YouTube and GitHub)
3. Michael Pyrcz is always happy to discuss! 😊



GeostatsGuy Lectures Channel on YouTube

GeostatsGuy Linear Regression Demo		Latest commit bc360d9 on Nov 12
Bootstrap.ipynb	Python bootstrap demonstration in Jupyter notebook.	a year ago
Convolution_Simulation_Demo.ipynb	Spatial correlation by convolution	a year ago
DT_Demo.ipynb	Decision Tree Demo	7 months ago
Dedustering.ipynb	Spatial Declustering Tutorial	6 months ago
LU+Decomposition.ipynb	LU Simulation	a year ago
NSCORE.ipynb	Plotly Python in Jupyter Notebook NSCORE Transform	10 months ago
PCA_Demo.ipynb	PCA Tutorial	8 months ago
PyGSLIB_dedus_python_demo.ipynb	PyGSLIB Declustering Demo	2 months ago
PyGSLIB_variogram_python_demo.ipynb	Variogram Calculation with PyGSLIB	2 months ago
PythonDataBasics_DataFrame.ipynb	Tutorial for Tabular Data Structures	6 months ago
PythonDataBasics_Hypothesis.ipynb	Update	2 months ago
PythonDataBasics_LinearRegression.ipynb	Linear Regression Demo	2 months ago
PythonDataBasics_ndarrays.ipynb	Tutorial for Gridded Data Structures	6 months ago
README.md	Update README.md	6 months ago
Spatial_Bootstrap.ipynb	Spatial Bootstrap Demo in Python	a year ago
SupportVectorMachines.ipynb	Support Vector Machine Demo	6 months ago
Variogram.ipynb	Variogram Workflow Tutorial	6 months ago
image2GSLIB.py	Image2GSLIB	6 months ago

GeostatsGuy Repositories on GitHub

Now let's
BEGIN...

Are you excited?

Introduction

Data Analytics

Inferential Methods

Predictive Methods

Advanced Methods

Conclusions

Definitions

Prerequisites

Probability

Spatial Bias

Uncertainty

Spatial

PCA

Clustering

Decision Tree

Support Vector Machines

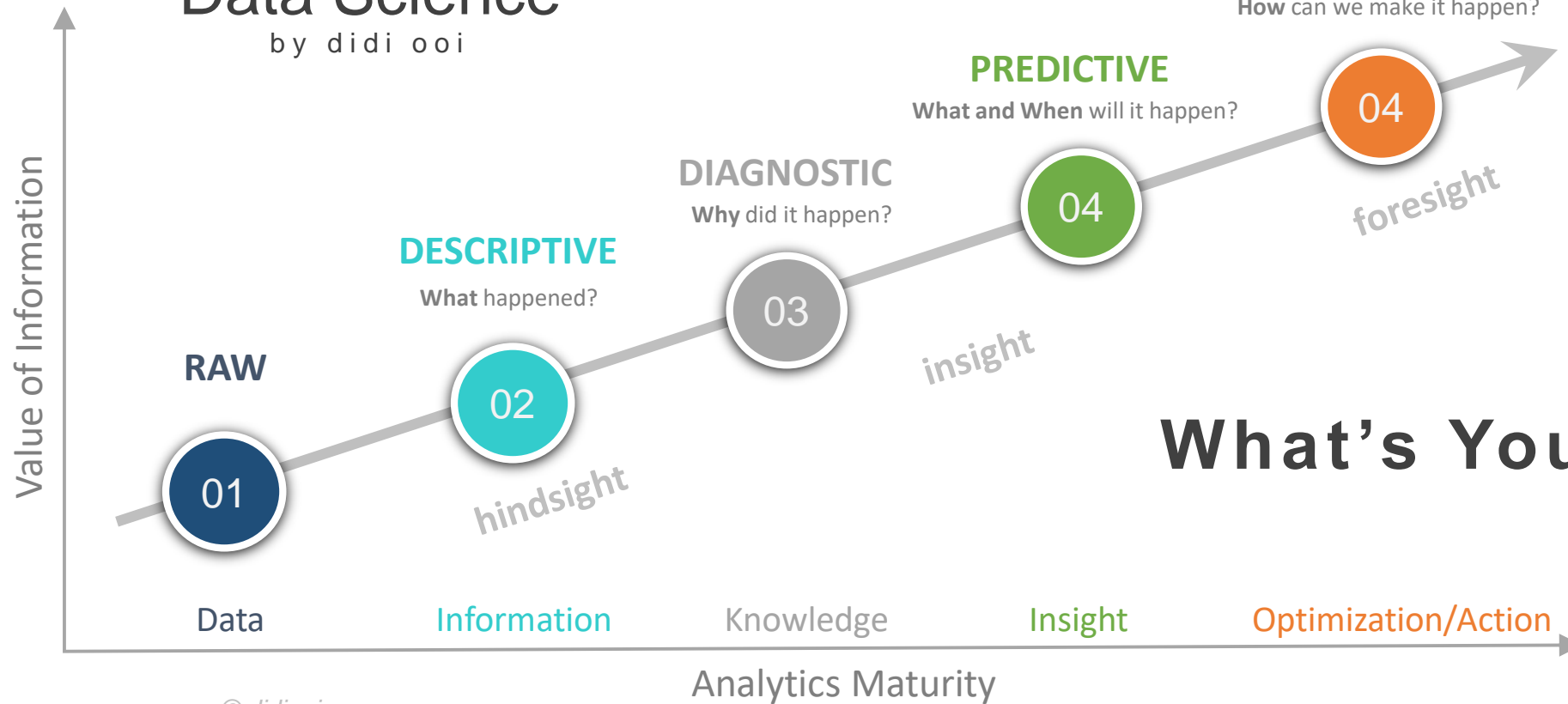
Neural Nets

Convolutional Neural Nets

Recurrent Neural Nets

Building Blocks of Data Science

by didi ooi

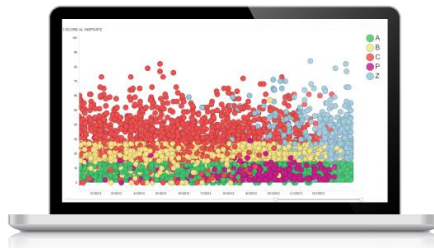


© didiooi

What's Your Goal?

Data Science

- Data Source
- Data Format
- Data Visualization
- Data Preparation
- Data Training



Data Discovery
& Visualization



Data Wrangling



Predictive
Analytics



Spatial
Analytics