DAYTUM – INTRODUCTION TO ENERGY MACHINE LEARNING

Course Overview

Lecture outline . . .

- Motivation / Goals
- Class Description / Objectives
- Summary
- Open Source

INTRODUCTIONS: THE INSTRUCTORS

Michael Pyrcz

1. **Pyrcz**: is pronounced “perch”

2. **I have practical experience**: over 17 years of experience in consulting, teaching and industrial R&D in statistical modeling, reservoir modeling and uncertainty characterization

Oil and Gas University, Florence, Italy

Anadarko, Midland, TX

Spring 2018 Class of Introduction to Geostatistics

INTRODUCTIONS: THE INSTRUCTORS

Michael Pyrcz

3. **Flexible**: got ideas, feedback to improve the learning opportunities. Let's work together to reach our learning objective.

4. **Available**: I have an open door policy. Drop by my office. Drop a line anytime.

5. **An Engineer, but…**
   - my B.Sc. was in Mining Engineering,
   - my M.Sc. started as Geotechnical Engineering (then skipped to Ph.D.) and
   - my Ph.D. was in Quantitative Geology.
   - Then I spent 13 years in Earth Science R&D working with geological and geophysical reservoir modeling.
   - I speak geo

Fall 2018 Class of Introduction to Geostatistics

INTRODUCTIONS: THE INSTRUCTORS

Michael Pyrcz

AAPG SEPM Panel Discussion on Modeling

CPGE Webinar on Big Data

7. 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

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

INTRODUCTIONS: THE INSTRUCTORS

John T. Foster

1. Co-founder and CTO of daytum

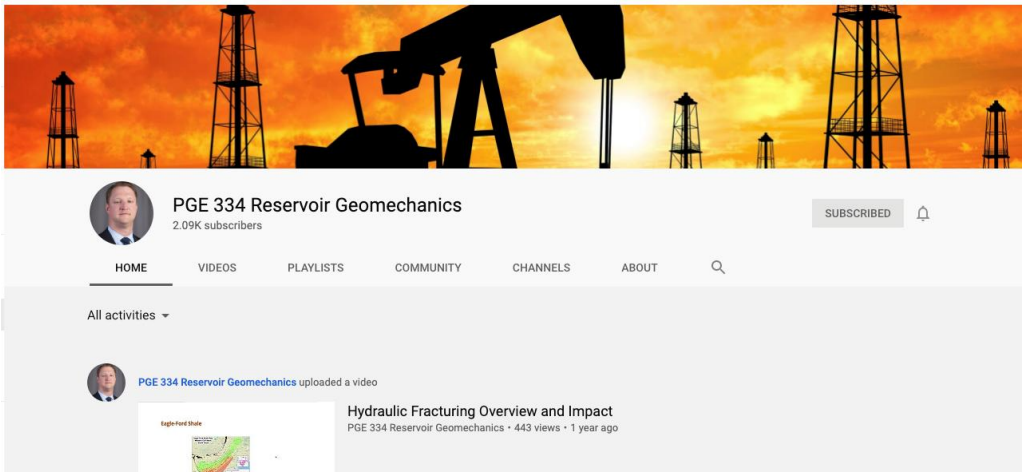
2. Associate professor in The Hildebrand Department of Petroleum and Geosystems Engineering and The Department of Aerospace Engineering and Engineering Mechanics

INTRODUCTIONS: THE INSTRUCTORS

John T. Foster

3. Active in Outreach, Social Media and Professional Organizations

- Previously a faculty member in mechanical engineering at UTSA Program and a and was a Senior Member of the Technical Staff at Sandia National Laboratories
- Social Media - @johntfoster on Twitter, GitHub, PGE 334 Reservoir Geomechanics on YouTube



INTRODUCTIONS

Short Introductions:

- Name
- Role
  - Geo / Eng
  - Coding experience
  - Machine learning experience
- Expectations from this Class

WHAT WILL YOU LEARN?

The Goal – Deep Dive into Data Analytics and Machine Learning

Day 1

- Overview
- Introduction
- Probability
- Data Preparation
- Feature Selection
- Feature Engineering
- Spatial Models and Spatial Estimation
- Spatial Simulation
- Uncertainty Modeling

Tuesday with Spatial Data Analytics

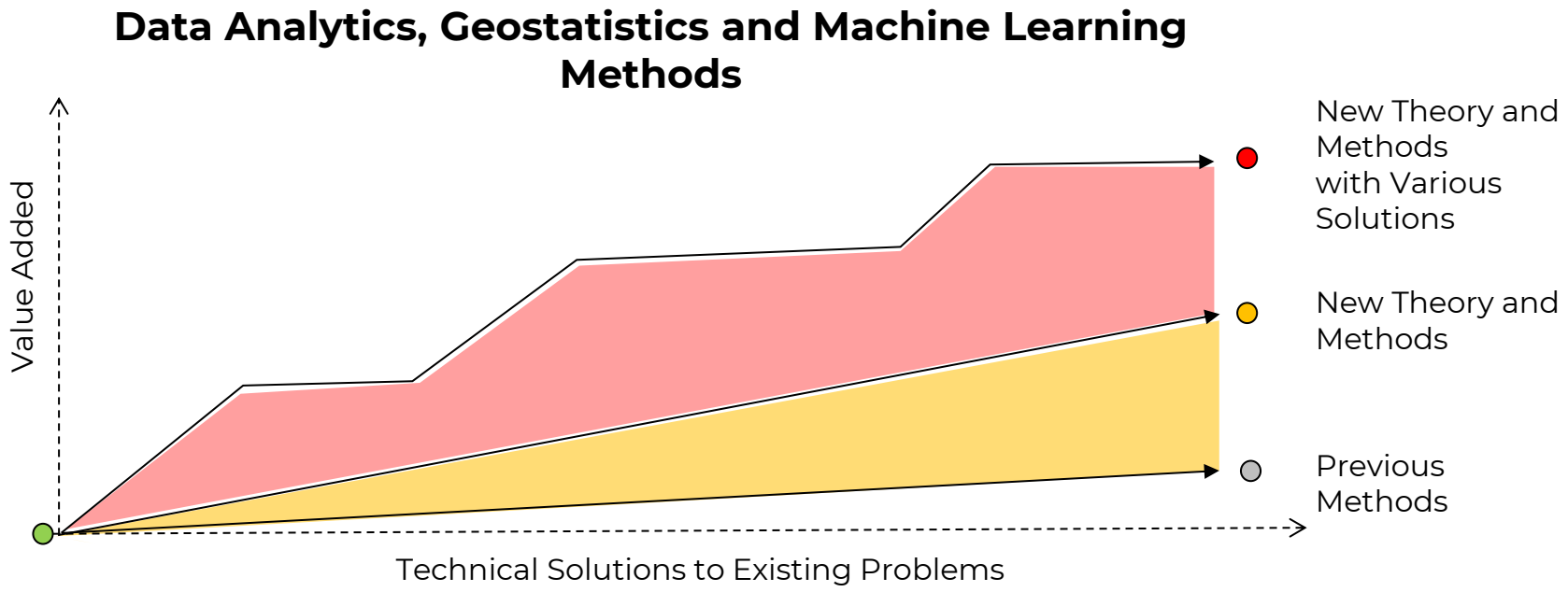
Day 2

- Machine Learning Intro
- Clustering
- Dimensionality Reduction
- Naïve Bayes
- k-nearest Neighbors
- Tree-based Regression
- Ensemble Tree-based Regression
- Support Vector Machines
- Artificial Neural Networks

WHAT WILL YOU LEARN?

This Workshop is an Investment in Learning to....

- Build operational capability
- Provide incremental value



WHAT WILL YOU LEARN?

Reaching our Goal

- Today we will:
  - Build up from zero
  - Provide an overview of the methods
  - Hands-on experiential learning
  - Demonstrate well-documented, practical workflows
- Of course, full workflow development would require time to investigate the problem and available data
- I have a lot of content for more advanced topics

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

WHAT WILL YOU LEARN?

- **There is Much More!** – the building blocks can be reimplemented and expanded to address various other problems, opportunities
- There is much more that we could cover:
  - Additional Theory
  - More Hands-on / Experiential
  - Workflow Development
  - Basics of Python / R
  - Model QC
  - Methods to Integrate More Geoscience and Engineering

HOW WILL YOU LEARN ALL OF THAT?

Here's the Plan

1. **Interactive** lectures / discussion to cover the basic concepts

2. **Demonstrations** of methods and workflows in Python

3. **Hands-on experiential learning** with well-documented workflows for accessibility

WHAT WILL YOU LEARN?

This is an ambitious schedule.

We will **adjust for success**:

- Let me know if you are lost, stuck, something is not working, or you aren't learning!

Feedback welcome as we proceed

WHY EXCEL AND PYTHON?

Python

- 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

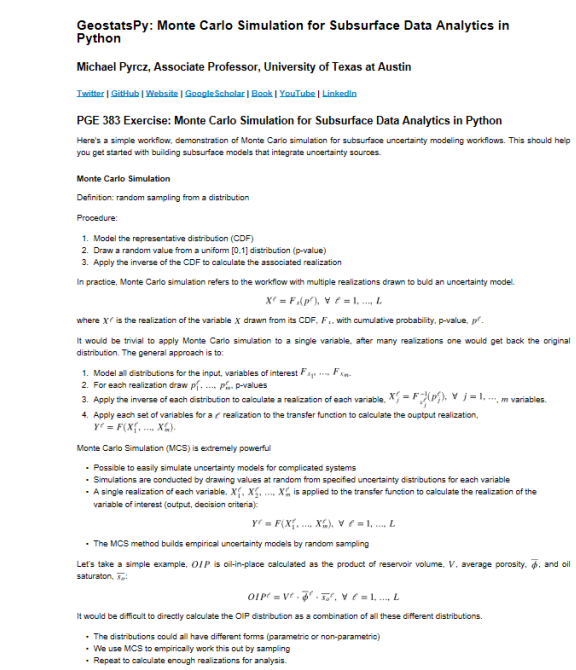
‘Certainly there’s a phenomenon around open source. You know free software will be a vibrant area. ‘There will be a lot of nest things that get done there.’ - Bill Gates

‘20 years with C++ and FORTRAN, but with Python I code less, but get more done.’ - Michael Pyrcz

WHY PYTHON?

Python with Jupyter Notebooks

- Workflows that integrate blocks of code, documentation, results
- 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)



GeostatsPy: Monte Carlo Simulation for Subsurface Data Analytics in Python

Michael Pyrcz, 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 buld an uncertainty model.

$$X^\ell = F_x(p^\ell), \; \forall \; \ell = 1, ..., L$$

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 $F_{x_1}, ..., F_{x_m}$.
2. For each realization draw $p_1^\ell, ..., p_m^\ell$, p-values
3. Apply the inverse of each distribution to calculate a realization of each variable, $X_j^\ell = F_{x_j}^{-1}(p_j^\ell), \; \forall \; j = 1, ..., m$ variables.
4. Apply each set of variables for a ℓ realization to the transfer function to calculate the ouptput realization, $Y^\ell = F(X_1^\ell, ..., X_m^\ell)$.

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, $X_1^\ell, X_2^\ell, ..., X_m^\ell$ is applied to the transfer function to calculate the realization of the variable of interest (output, decision criteria):

$$Y^\ell = F(X_1^\ell, ..., X_m^\ell), \; \forall \; \ell = 1, ..., L$$

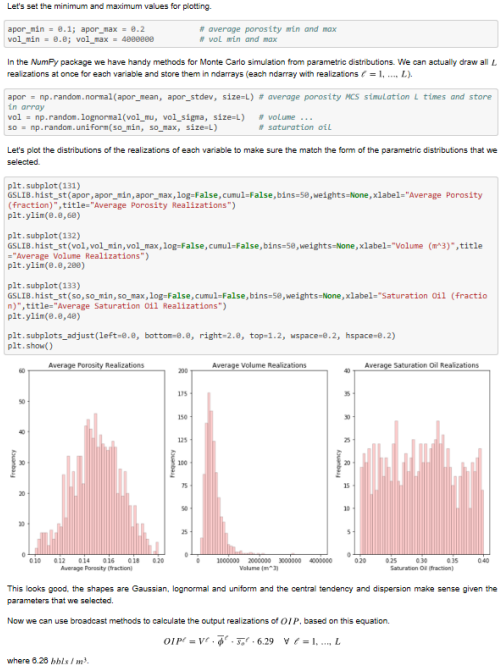
- The MCS method builds empirical uncertainty models by random sampling

Let's take a simple example, OIP is oil-in-place calculated as the product of reservoir volume, V, average porosity, $\overline{\phi}$, and oil saturaton, $\overline{s_o}$:

$$OIP^\ell = V^\ell \cdot \overline{\phi}^\ell \cdot \overline{s_o}^\ell, \; \forall \; \ell = 1, ..., L$$

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 = 4000000        # 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 $\ell = 1, ..., L$).

```
apor = np.random.normal(apor_mean, apor_stdev, 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.

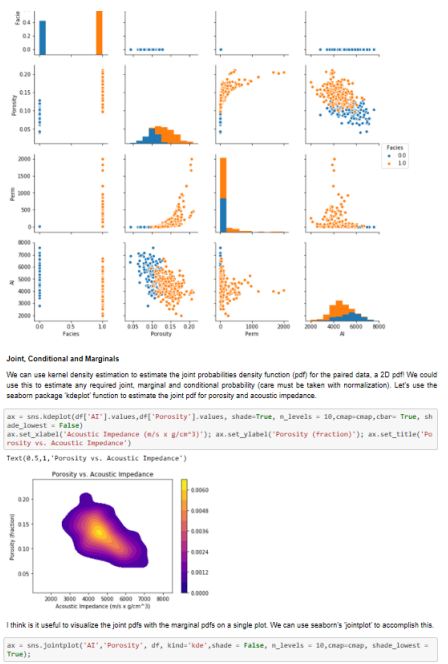
```
plt.subplot(131)
GSLIB.hist_st(apor,apor_min,apor_max,log=False,cumul=False,bins=50,weights=None,xlabel="Average Porosity (fraction)",title="Average Porosity Realizations")
plt.ylim(0.0,60)

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(0.0,200)

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(0.0,40)

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.

$$OIP^\ell = V^\ell \cdot \overline{\phi}^\ell \cdot \overline{s_o}^\ell \cdot 6.29 \quad \forall \; \ell = 1, ..., L$$

where 6.28 $bbls/m^3$.



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,cbar= 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')
```

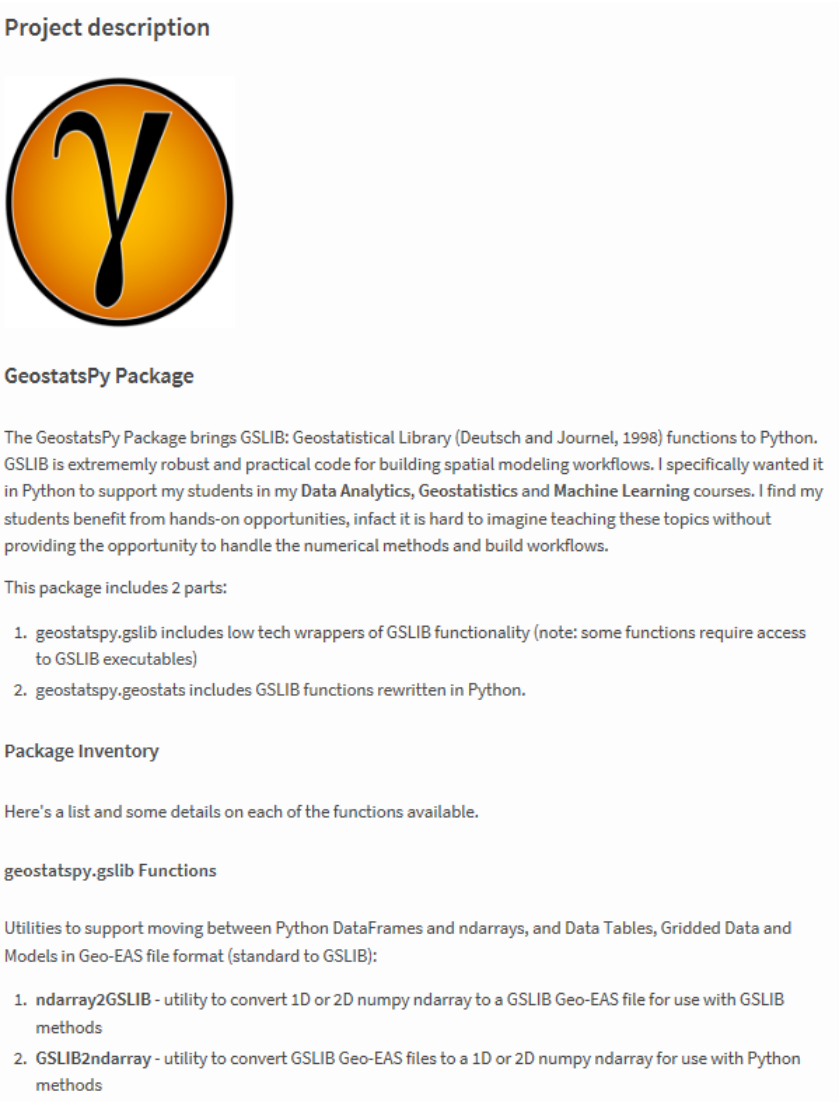

I think is it useful to visualize the joint pdfs with the marginal pdfs on a single plot. We can use seaborn's 'jointplot' to accomplish this.

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

GEOSTATSPY?

GeostatsPy Python Package

- Set of Functions in Python
  - GeostatsPy is a set of Python functions for most of the required workflow steps
  - Much is reimplemented in Python.
  - Package written by myself, we will tailor, augment to support training.
  - I welcome feedback.
  - Open Source - anyone can use it
  - Free for any to use
  - Download it from PyPi with:
    - ‘pip install geostatspy’



Project description

GeostatsPy Package

The GeostatsPy Package brings GSLIB: Geostatistical Library (Deutsch and Journel, 1998) functions to Python. GSLIB is extrememly 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

More on Software / Coding:

- Coding:
  - **Workflow Focus**: We introduce a lot of well-documented Python workflows in Data Analytics, Geostatistics and Machine Learning
  - **Practical Approach**: We focus on coding as a means to an ends, to get the job done.
- Expectations:
  - **Awareness**: Appreciation for what can be done and the level of complexity
  - **Basic Workflow Design**: Basic coding, workflow design
  - **Experience with Tools**: Use of methods available in common packages (from the previous 3 days)

REASONS TO LEARN CODING

- **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.

REASONS TO LEARN CODING

Caveats for the 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. We 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.

LET'S GET STARTED

We will first start with prerequisites:

Day 1

- Overview
- Introduction
- Probability
- Data Analytics
- Feature Selection
- Feature Engineering
- Spatial Models and Spatial Estimation
- Spatial Simulation
- Uncertainty Modeling

Thursday with Spatial Data Analytics

Day 2

- Machine Learning Intro
- Clustering
- Dimensionality Reduction
- Naïve Bayes
- k-nearest Neighbors
- Tree-based Regression
- Ensemble Tree-based Regression
- Support Vector Machines
- Artificial Neural Networks
- Shapley Values