Introduction to GeostatsPy

Lecture outline . . .

- Who am I?
- Introduction to GeostatsPy
- The Plan
- Getting Setup
- Call to Action

Michael Pyrcz, The University of Texas at Austin

Motivation

"Promote the use of GeostatsPy package for enabling spatial data analytics and geostatistics in Python workflows."

Based on the original Geostatistical Library in Fortran

(Deutsch and Journal, 1998).

http://claytonvdeutsch.com/wp-content/uploads/2019/03/GSLIB-Book-Second-Edition.pdf

GSLIB: Geostatistical Software Library and User's Guide

Second Edition

CLAYTON V. DEUTSCH
Department of Petroleum Engineering
Stanford University

ANDRE G. JOURNEL
Department of Geological & Environmental Sciences
Stanford University

GSLIB made freely available was broadly adopted in the community and brought geostatistics tools to any interested practitioner.

Introduction to GeostatsPy

Lecture outline . . .

- **Who am I?**

Who Am I?

Spring 2018 Class of Introduction to Geostatistics

Oil and Gas University, Florence, Italy

Anadarko, Midland, TX

Michael Pyrcz

1. **Pyrcz**: is pronounced “perch”
2. **I’m New**: new to academia, August, 2017.
3. **I’ve Done This**: over 17 years of experience in consulting, teaching and industrial R&D in statistical modeling, reservoir modeling and uncertainty characterization.
4. **I Left Industry to Teach**: “I want to give you a competitive edge in your careers with geostatistics, data analytics and statistical / machine learning.”

Who Am I?

Frequent Industry Courses

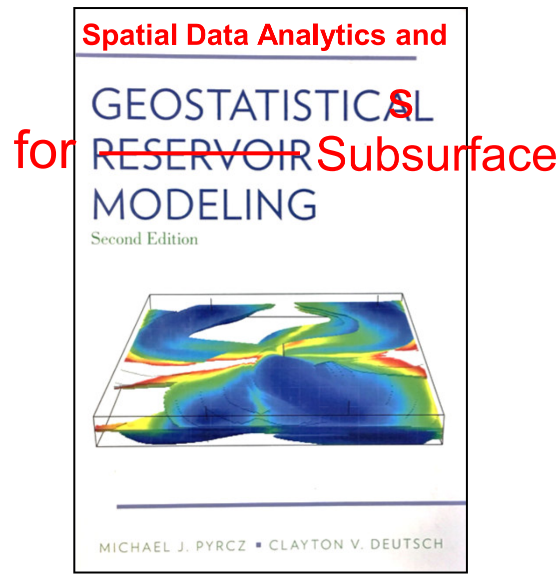
Data Science Bootcamps

Data Analytics and Machine Learning Consortium

Michael Pyrcz

5. Expertise: frequently teach and consult in our industry, have a broad network and recognized for my contributions to statistics, modeling, data analytics and geostatistics theory and practice.

6. Teaching in Industry: as Chief Science Officer / Co-founder of daytum, frequently teach industry professionals.



The Book

Who Am I?

AAPG SEPM Panel Discussion on Modeling

CPGE Webinar on Big Data

Michael Pyrcz

6. 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
- GeostatsGuy on Twitter, GitHub, GeostatsGuy Lectures on YouTube
- guest editor for AAPG Bulletin Special Issue on Subsurface Data Analytics and Machine Learning.

Introduction to GeostatsPy

Lecture outline . . .

- **Introduction to GeostatsPy**

Origin of GeostatsPy

In Spring 2019 I taught a Subsurface Data Analytics and Geostatistics course with geoscience ($^1/_3$) and engineering students ($^2/_3$).

The Ambitious Plan:

- A semester-long simulation with the instructor, Michael Pyrcz, as the subsurface manager and each student as the member of my subsurface asset team.
- Students learned and applied spatial data analytics to unique dataset, provided written and presentation-based progress reports.

The class quickly filled up, got a new room and it filled up again to 50 students.

- During the winter "break" I planned the class, made the lectures and datasets. Then I just needed to find the tools, build example workflows:

1. GSLIB in Fortran – I tried that Fall 2017 and lost 50% of the students in the first 2 weeks.

2. Python Executing GSLIB executables as "wrappers" – I coded that up in the GeostatsPy.GSLIB module, too error prone, an ugly solution.

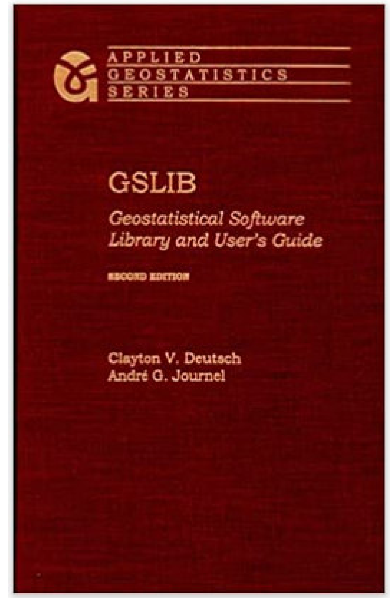
I had to find a new solution.

What is GSLIB?

GeoStatistical **LIB**rary (GSLIB)

by Deutsch and Journel (1992, 1998)

- **open source geostatistics that made spatial modeling accessible to the world!**
- parameter files, executables, outputs, viz tools
- written in Fortran, find it at GSLIB.com
- great documentation with the GSLIB user's guidebook (now free, link on Clayton Deutsch's website)
- simple, building block functions approach
- includes a suite of foundational functions, very easy to modify and extend
- much of GeostatsPy is a translation/reimplementation of GSLIB in Python



GSLIB guidebook

```
                    Parameters for SISIM
                    ********************

START OF PARAMETERS:
1                              -1=continuous(cdf), 0=categorical(pdf)
5                              -number thresholds/categories
0.5   1.0   2.5   5.0   10.0   -   thresholds / categories
0.12  0.29  0.50  0.74  0.88   -   global cdf / pdf
../data/cluster.dat            -file with data
1   2   0   3                  -   columns for X,Y,Z, and variable
direct.ik                      -file with soft indicator input
1   2   0   3 4 5 6 7          -   columns for X,Y,Z, and indicators
0                              -   Markov-Bayes simulation (0=no,1=yes)
0.61  0.54  0.56  0.53  0.29   -      calibration B(z) values
-1.0e21    1.0e21              -trimming limits
0.0   30.0                     -minimum and maximum data value
1      0.0                     -   lower tail option and parameter
1      1.0                     -   middle     option and parameter
1     30.0                     -   upper tail option and parameter
cluster.dat                    -   file with tabulated values
3   0                          -      columns for variable, weight
0                              -debugging level: 0,1,2,3
sisim.dbg                      -file for debugging output
sisim.out                      -file for simulation output
1                              -number of realizations
50   0.5    1.0                -nx,xmn,xsiz
50   0.5    1.0                -ny,ymn,ysiz
1    1.0   10.0                -nz,zmn,zsiz
69069                          -random number seed
12                             -maximum original data  for each kriging
12                             -maximum previous nodes for each kriging
1                              -maximum soft indicator nodes for kriging
1                              -assign data to nodes? (0=no,1=yes)
0     3                        -multiple grid search? (0=no,1=yes),num
0                              -maximum per octant    (0=not used)
20.0  20.0  20.0               -maximum search radii
 0.0   0.0   0.0               -angles for search ellipsoid
51    51    11                 -size of covariance lookup table
0    2.5                       -0=full IK, 1=median approx. (cutoff)
```

GSLIB example parameter file

Michael Pyrcz, The University of Texas at Austin

Origin of GeostatsPy

I considered a variety of options:

gstat in R by Edzer Pebesma – great contribution, but impractical beyond univariate simulation and not currently under development.

pygeostats – CCG University of Alberta – looks great but only available to consortium members

I tested a variety of open source Python packages:

- *some were not currently maintained/up to date, some lacked documentation and examples, some just didn't run. I literally had things break the morning of lectures.*

I needed something robust, it had to just work and be easy to use.

- I spent my weekends reimplementing GSLIB in Python, days before the lectures.
- Under this pressure I decided to limit it to 2D and remove some advanced options.

Structure of GeostatsPy

How is GeostatsPy structured?

GeostatsPy Package

GSLIB Module

- reimplementation of data viz with matplotlib
- import / export Python / GSLIB
- wrappers of GSLIB executables 2D / 3D

geostats Module

- reimplementation of GSLIB subroutines
- reimplementation of GSLIB in 2D

Structure of GeostatsPy

How is GeostatsPy structured?

- GSLIB-like data visualization with functions based on matplotlib

- Bridge for new users, common plots.

GeostatsPy Package

GSLIB Module

reimplementation of data viz with matplotlib

import / export Python / GSLIB

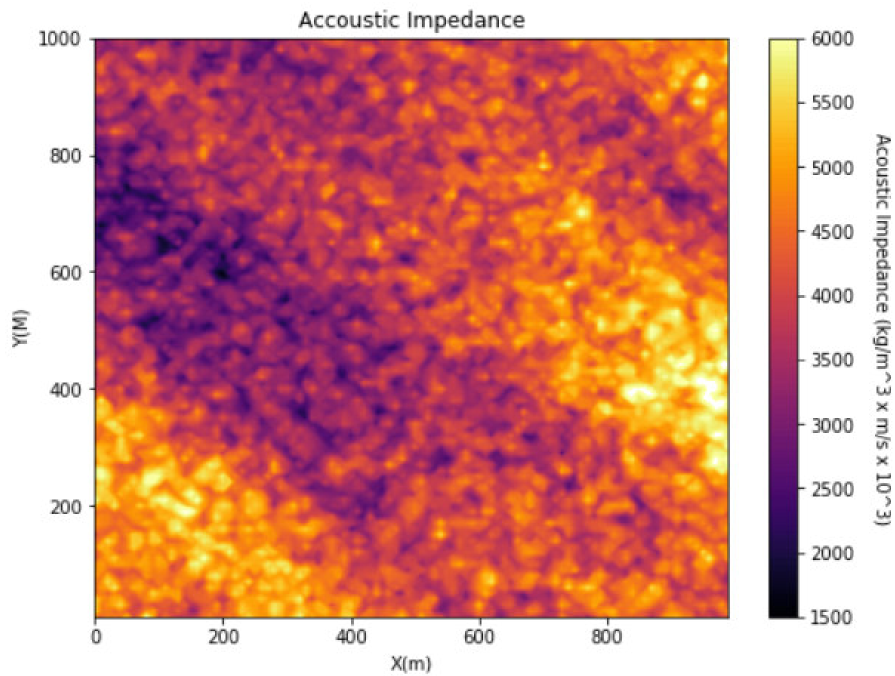
wrappers of GSLIB executables 2D / 3D

geostats Module

reimplementation of GSLIB subroutines

reimplementation of GSLIB in 2D

```
GSLIB.pixelplt(seismic,xmin,xmax,ymin,ymax,cell_size,1500,6000,
               "Accoustic Impedance","X(m)","Y(M)","Acoustic Impedance (kg/m^3 x m/s x 10^3)",cmap,"Impedance_Map")
```



GeostatsPy.GSLIB.pixelplt seismic example

Michael Pyrcz, The University of Texas at Austin

Structure of GeostatsPy

How is GeostatsPy structured?

- Moving between GSLIB and standard Python file formats

- GSLIB uses Geo-DAS ASCII files

GeostatsPy Package

GSLIB Module

reimplementation of data viz with matplotlib

import / export Python / GSLIB

wrappers of GSLIB executables 2D / 3D

geostats Module

reimplementation of GSLIB subroutines

reimplementation of GSLIB in 2D

```python
def ndarray2GSLIB(array, data_file, col_name):
    """Convert 1D or 2D numpy ndarray to a GSLIB Geo-EAS file for use with
    GSLIB methods.

    :param array: input array
    :param data_file: file name
    :param col_name: column name
    :return: None
    """

    if array.ndim not in [1, 2]:
        raise ValueError("must use a 2D array")

    with open(data_file, "w") as f:
        f.write(data_file + "\n")
        f.write("1 \n")
        f.write(col_name + "\n")

        if array.ndim == 2:
            ny, nx = array.shape

            for iy in range(ny):
                for ix in range(nx):
                    f.write(str(array[ny - 1 - iy, ix]) + "\n")

        elif array.ndim == 1:
            nx = len(array)
            for ix in range(0, nx):
                f.write(str(array[ix]) + "\n")
```

GeostatsPy.GSLIB.ndarray2GSLIB example

Structure of GeostatsPy

GeostatsPy Package

GSLIB Module
- reimplementation of data viz with matplotlib
- import / export Python / GSLIB
- **wrappers of GSLIB executables 2D / 3D**

geostats Module
- reimplementation of GSLIB subroutines
- reimplementation of GSLIB in 2D

GSLIB Wrappers

- *Extremely robust set of executables*

Functions to:

- write out:
  - *parameter files*
  - *data files in Geo-DAS format*
- *Execute the Fortran executable*
- *Import the output*
- *Wendi Liu made the Mac OS compiled GSLIB executables available, Windows and Linux available at GSLIB.com.*

```
ndarray2GSLIB(sec, "sec.dat", "sec_dat")

with open("sgsim.par", "w") as f:
    f.write("              Parameters for SGSIM                                    \n")
    f.write("              ********************                                    \n")
    f.write("                                                                      \n")
    f.write("START OF PARAMETER:                                                   \n")
    f.write("none                           -file with data                        \n")
    f.write("1  2  0  3  5  0               -  columns for X,Y,Z,vr,wt,sec.var.    \n")
    f.write("-1.0e21 1.0e21                 -  trimming limits                     \n")
    f.write("0                              -transform the data (0=no, 1=yes)      \n")
    f.write("none.trn                       -  file for output trans table         \n")
    f.write("0                              -  consider ref. dist (0=no, 1=yes)    \n")
    f.write("none.dat                       -  file with ref. dist distribution    \n")
    f.write("1  0                           -  columns for vr and wt               \n")
    f.write("-4.0    4.0                    -  zmin,zmax(tail extrapolation)       \n")
    f.write("1      -4.0                    -  lower tail option, parameter        \n")
    f.write("1       4.0                    -  upper tail option, parameter        \n")
    f.write("0                              -debugging level: 0,1,2,3              \n")
    f.write("nonw.dbg                       -file for debugging output             \n")
    f.write(str(output_file) + "            -file for simulation output            \n")
    f.write(str(nreal) + "                  -number of realizations to generate    \n")
    f.write(str(nx) + " " + str(hmn) + " " + str(hsiz) + "                         \n")
    f.write(str(ny) + " " + str(hmn) + " " + str(hsiz) + "                         \n")
    f.write("1 0.0 1.0                      - nz zmn zsiz                          \n")
    f.write(str(seed) + "                   -random number seed                    \n")
    f.write("0     8                        -min and max original data for sim     \n")
    f.write("12                             -number of simulated nodes to use      \n")
    f.write("0                              -assign data to nodes (0=no, 1=yes)    \n")
    f.write("1     3                        -multiple grid search (0=no, 1=yes),num \n")
    f.write("0                              -maximum data per octant (0=not used)  \n")
    f.write(str(max_range) + " " + str(max_range) + " 1.0 -maximum search  (hmax,hmin,vert) \n")
    f.write(str(azi1) + "   0.0   0.0       -angles for search ellipsoid            \n")
    f.write(str(hctab) + " " + str(hctab) + " 1 -size of covariance lookup table   \n")
    f.write("4 " + str(correl) + " 1.0      -ktype: 0=SK,1=OK,2=LVM,3=EXDR,4=COLC  \n")
    f.write("sec.dat                        -  file with LVM, EXDR, or COLC variable \n")
    f.write("1                              -  column for secondary variable       \n")
    f.write(str(nst) + " " + str(nug) + "  -nst, nugget effect                     \n")
    f.write(str(it1) + " " + str(cc1) + " " + str(azi1) + " 0.0 0.0 -it,cc,ang1,ang2,ang3 \n")
    f.write(" " + str(hmaj1) + " " + str(hmin1) + " 1.0 - a_hmax, a_hmin, a_vert       \n")
    f.write(str(it2) + " " + str(cc2) + " " + str(azi2) + " 0.0 0.0 -it,cc,ang1,ang2,ang3 \n")
    f.write(" " + str(hmaj2) + " " + str(hmin2) + " 1.0 - a_hmax, a_hmin, a_vert       \n")

os.system("sgsim.exe sgsim.par")
sim_array = GSLIB2ndarray(output_file, 0, nx, ny)
```

GeostatsPy.GSLIB.cosgsim_unconditional example

Michael Pyrcz, The University of Texas at Austin

Structure of GeostatsPy

Python GSLIB subroutines

- *Common geostats / subsurface modeling operations*

Example Subroutines:

- distribution transformations
- *spatial covariance*
- *geometric distance*
- *rotation*
- *kriging solution*
- *CDF interpolation and extrapolation*

GeostatsPy Package

GSLIB Module

reimplementation of data viz with matplotlib

import / export Python / GSLIB

wrappers of GSLIB executables 2D / 3D

geostats Module

reimplementation of GSLIB subroutines

reimplementation of GSLIB in 2D

```
@jit(nopython=True)
def cova2(x1, y1, x2, y2, nst, c0, pmx, cc, aa, it, ang, anis, rotmat, maxcov):
    """Calculate the covariance associated with a variogram model specified by
    a nugget effect and nested variogram structures.
    :param x1: x coordinate of first point
    :type x1: float
    :param y1: y coordinate of first point
    :type y1: float
    :param x2: x coordinate of second point
    :type x2: float
    :param y2: y coordinate of second point
    :type y2: float
    :param nst: number of nested structures (maximum of 4)
    :type nst: int
    :param c0: isotropic nugget constant (TODO: not used)
    :type c0: float
    :param pmx: Maximum variogram value needed for kriging when using power
                model. pmx is a unique value used for all nested structures
                that use the power model, so pmx should be chosen to account
                for the largest structure that uses the power model.
    :type pmx: float
    :param cc: multiplicative factor of each nested structure
    :type cc: array
    :param aa: parameter `a` of each nested structure
    :type aa: array
    :param it: Integer value indicating type of variogram model
             for values 0,1,2,..., nst
             it[value] == 1: Spherical model
                 (aa[value] == `a` is the range, cc[value] is the contribution)
             it[value] == 2: Exponential model
                 (aa[value] == `a`, 3a is the practical range,
                 cc[value] is the contribution)
             it[value] == 3: Gaussian model
```

GeostatsPy.geostats.cova2 example

Structure of GeostatsPy

Python GSLIB subroutines

- *Common geostats / subsurface modeling algorithms*

Example Algorithms:

- declustering
- simple and ordinary kriging
- sequential Gaussian simulation
- indicator kriging
- indicator simulation
- collocated cokriging

GeostatsPy Package

GSLIB Module

reimplementation of data viz with matplotlib

import / export Python / GSLIB

wrappers of GSLIB executables 2D / 3D

geostats Module

reimplementation of GSLIB subroutines

reimplementation of GSLIB in 2D

```
def ik2d(df,xcol,ycol,vcol,ivtype,koption,ncut,thresh,gcdf,trend,tmin,tmax,nx,xmn,xsiz,ny,ymn,ysiz,ndmin,ndmax,radius,ktype,vario):

    """A 2D version of GSLIB's IK3D Indicator Kriging program (Deutsch and Journel, 1998) converted from the
    original Fortran to Python by Michael Pyrcz, the University of Texas at
    Austin (March, 2019).
    :param df: pandas DataFrame with the spatial data
    :param xcol: name of the x coordinate column
    :param ycol: name of the y coordinate column
    :param vcol: name of the property column (cateogorical or continuous - note continuous is untested)
    :param ivtype: variable type, 0 - categorical, 1 - continuous
    :param koption: kriging option, 0 - estimation, 1 - cross validation (under construction)
    :param ncut: number of categories or continuous thresholds
    :param thresh: an ndarray with the category labels or continuous thresholds
    :param gcdf: global CDF, not used if trend is present
    :param trend: an ndarray [ny,ny,ncut] with the local trend proportions or cumulative CDF values
    :param tmin: property trimming limit
    :param tmax: property trimming limit
    :param nx: definition of the grid system (x axis)
    :param xmn: definition of the grid system (x axis)
    :param xsiz: definition of the grid system (x axis)
    :param ny: definition of the grid system (y axis)
    :param ymn: definition of the grid system (y axis)
    :param ysiz: definition of the grid system (y axis)
    :param nxdis: number of discretization points for a block
    :param nydis: number of discretization points for a block
    :param ndmin: minimum number of data points to use for kriging a block
    :param ndmax: maximum number of data points to use for kriging a block
    :param radius: maximum isotropic search radius
    :param ktype: kriging type, 0 - simple kriging and 1 - ordinary kriging
    :param vario: list with all of the indicator variograms (sill of 1.0) in consistent order with above parameters
    :return:
    """
```

GeostatsPy.geostats.ik2d example

Michael Pyrcz, The University of Texas at Austin

GeostatsPy Contributors

Appreciations to the contributors!

Honggeun Jo – initial 3D routines for variogram calculation and modeling

Anton Kupenko – bug fixes, Docstrings, code conformance to PEP8, removed duplicate functions

Wendi Liu – 3D gamma bar routine

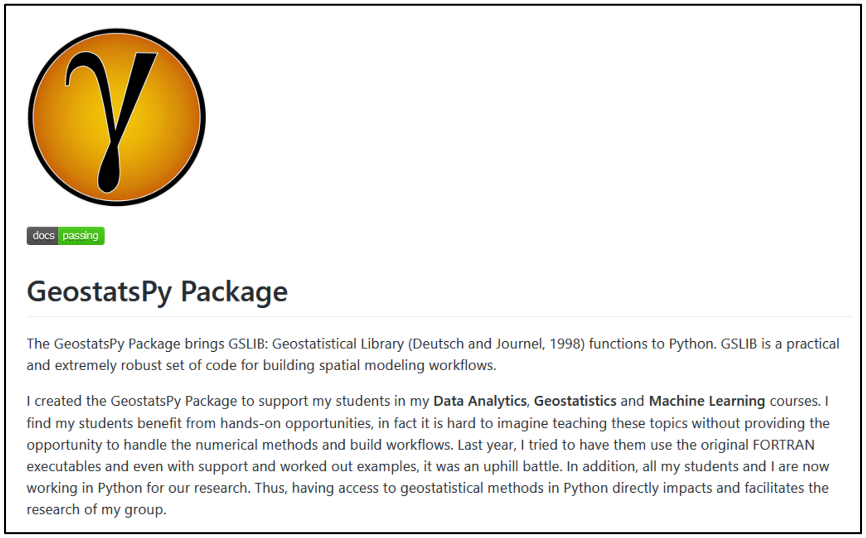
Alex Gigliotti – initial unit test for Travis continuous integration

Michael Pyrcz – GSLIB and geostats modules

I would love to see this expand

Current State of GeostatsPy

What is good?

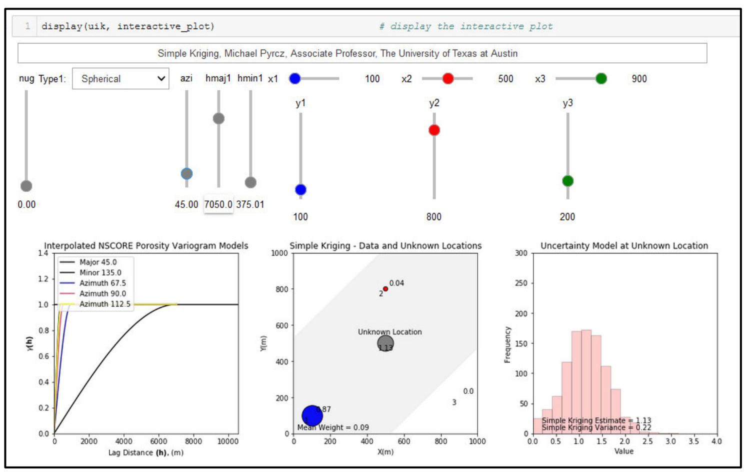

docs passing

GeostatsPy Package

The GeostatsPy Package brings GSLIB: Geostatistical Library (Deutsch and Journel, 1998) functions to Python. GSLIB is a practical and extremely robust set of code for building spatial modeling workflows.

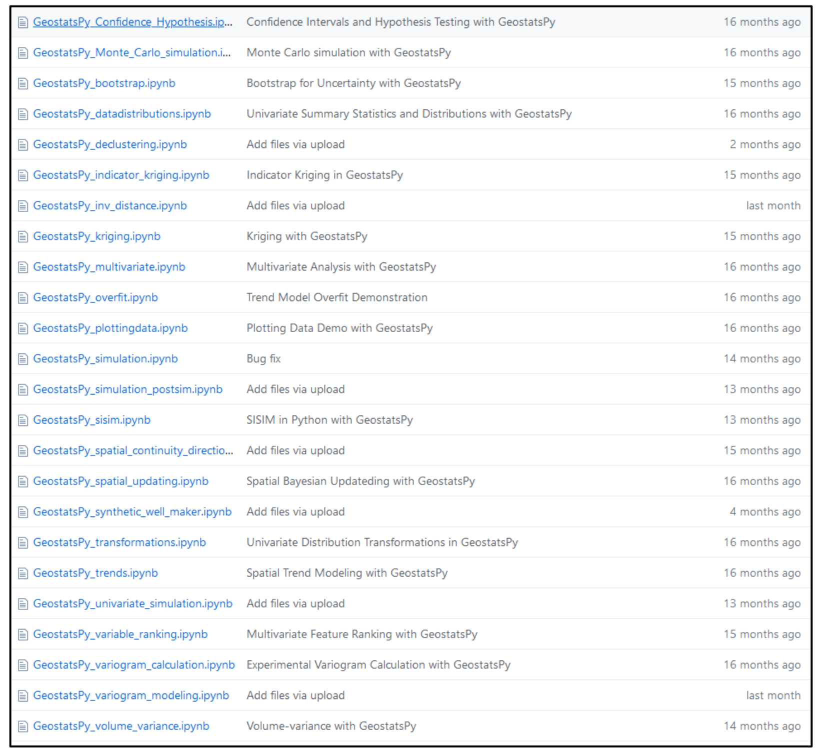
I created the GeostatsPy Package to support my students in my **Data Analytics**, **Geostatistics** and **Machine Learning** courses. I find my students benefit from hands-on opportunities, in fact it is hard to imagine teaching these topics without providing the opportunity to handle the numerical methods and build workflows. Last year, I tried to have them use the original FORTRAN executables and even with support and worked out examples, it was an uphill battle. In addition, all my students and I are now working in Python for our research. Thus, having access to geostatistical methods in Python directly impacts and facilitates the research of my group.

GeostatsPy GitHub repository and docs.



Interactive demonstrations for teaching tools.

GeostatsPy available on PyPI



GeostatsPy_Confidence_Hypothesis.ip...	Confidence Intervals and Hypothesis Testing with GeostatsPy	16 months ago
GeostatsPy_Monte_Carlo_simulation.i...	Monte Carlo simulation with GeostatsPy	16 months ago
GeostatsPy_bootstrap.ipynb	Bootstrap for Uncertainty with GeostatsPy	15 months ago
GeostatsPy_datadistributions.ipynb	Univariate Summary Statistics and Distributions with GeostatsPy	16 months ago
GeostatsPy_declustering.ipynb	Add files via upload	2 months ago
GeostatsPy_indicator_kriging.ipynb	Indicator Kriging in GeostatsPy	15 months ago
GeostatsPy_inv_distance.ipynb	Add files via upload	last month
GeostatsPy_kriging.ipynb	Kriging with GeostatsPy	15 months ago
GeostatsPy_multivariate.ipynb	Multivariate Analysis with GeostatsPy	16 months ago
GeostatsPy_overfit.ipynb	Trend Model Overfit Demonstration	16 months ago
GeostatsPy_plottingdata.ipynb	Plotting Data Demo with GeostatsPy	16 months ago
GeostatsPy_simulation.ipynb	Bug fix	14 months ago
GeostatsPy_simulation_postsim.ipynb	Add files via upload	13 months ago
GeostatsPy_sisim.ipynb	SISIM in Python with GeostatsPy	13 months ago
GeostatsPy_spatial_continuity_directio...	Add files via upload	15 months ago
GeostatsPy_spatial_updating.ipynb	Spatial Bayesian Updateding with GeostatsPy	16 months ago
GeostatsPy_synthetic_well_maker.ipynb	Add files via upload	4 months ago
GeostatsPy_transformations.ipynb	Univariate Distribution Transformations in GeostatsPy	16 months ago
GeostatsPy_trends.ipynb	Spatial Trend Modeling with GeostatsPy	16 months ago
GeostatsPy_univariate_simulation.ipynb	Add files via upload	13 months ago
GeostatsPy_variable_ranking.ipynb	Multivariate Feature Ranking with GeostatsPy	15 months ago
GeostatsPy_variogram_calculation.ipynb	Experimental Variogram Calculation with GeostatsPy	16 months ago
GeostatsPy_variogram_modeling.ipynb	Add files via upload	last month
GeostatsPy_volume_variance.ipynb	Volume-variance with GeostatsPy	14 months ago

Many well-documented demonstrations for common subsurface data analytics and geostatistics workflows.

Michael Pyrcz, The University of Texas at Austin

Current State of GeostatsPy

What needs work?

```
def sisim(df,xcol,ycol,vcol,ivtype,koption,ncut,thresh,gcdf,trend,tmin,tmax,zmin,zmax,ltail,ltpar,middle,mpar,utail,utpar,nx,xmn,xs
          ndmax,nodmax,mults,nmult,noct,radius,ktype,vario):

    """A 2D version of GSLIB's SISIM Indicator Simulation program (Deutsch and Journel, 1998) converted from the
    original Fortran to Python by Michael Pyrcz, the University of Texas at
    Austin (March, 2019). WARNING: only tested for cateogrical ktype 0, 1 and 2 (locally variable proportion).
    :param df: pandas DataFrame with the spatial data
    :param xcol: name of the x coordinate column
    :param ycol: name of the y coordinate column
    :param vcol: name of the property column (cateogorical or continuous - note continuous is untested)
    :param ivtype: variable type, 0 - categorical, 1 - continuous
    :param koption: kriging option, 0 - estimation, 1 - cross validation (under construction)
    :param ncut: number of categories or continuous thresholds
    :param thresh: an ndarray with the category labels or continuous thresholds
    :param gcdf: global CDF, not used if trend is present
    :param trend: an ndarray [ny,ny,ncut] with the local trend proportions or cumulative CDF values
    :param tmin: property trimming limit
    :param tmax: property trimming limit
    :param nx: definition of the grid system (x axis)
    :param xmn: definition of the grid system (x axis)
    :param xsiz: definition of the grid system (x axis)
    :param ny: definition of the grid system (y axis)
    :param ymn: definition of the grid system (y axis)
    :param ysiz: definition of the grid system (y axis)
    :param nxdis: number of discretization points for a block
    :param nydis: number of discretization points for a block
    :param ndmin: minimum number of data points to use for kriging a block
    :param ndmax: maximum number of data points to use for kriging a block
    :param radius: maximum isotropic search radius
    :param ktype: kriging type, 0 - simple kriging and 1 - ordinary kriging
    :param vario: list with all of the indicator variograms (sill of 1.0) in consistent order with above parameters
    :return:
    """
```

Docstrings, documentation.

```
@jit(nopython=True)
def setup_rotmat(c0, nst, it, cc, ang, pmx):
    """Setup rotation matrix.
    :param c0: nugget constant (isotropic)
    :param nst: number of nested structures (max. 4)
    :param it: TODO
    :param cc: multiplicative factor of each nested structure
    :param ang: TODO
    :param pmx: TODO
    :return: TODO
    """
    PI = 3.141_592_65
    DTOR = PI / 180.0

    # The first time around, re-initialize the cosine matrix for the variogram
    # structures
    rotmat = np.zeros((4, nst))
    maxcov = c0
    for js in range(0, nst):
        azmuth = (90.0 - ang[js]) * DTOR
        rotmat[0, js] = math.cos(azmuth)
        rotmat[1, js] = math.sin(azmuth)
        rotmat[2, js] = -1 * math.sin(azmuth)
        rotmat[3, js] = math.cos(azmuth)
        if it[js] == 4:
            maxcov = maxcov + pmx
        else:
            maxcov = maxcov + cc[js]
    return rotmat, maxcov
```

Efficiency, optimization.

```
def semipartial_corr(C): # Michael Pyrcz modified the function above by Fabian Pedregosa-Izquierdo, f@bianp.net for semipartial cor
    C = np.asarray(C)
    p = C.shape[1]
    P_corr = np.zeros((p, p), dtype=np.float)
    for i in range(p):
        P_corr[i, i] = 1
        for j in range(i+1, p):
            idx = np.ones(p, dtype=np.bool)
            idx[i] = False
            idx[j] = False
            beta_i = linalg.lstsq(C[:, idx], C[:, j])[0]
            res_j = C[:, j] - C[:, idx].dot( beta_i)
            res_i = C[:, i] # just use the value, not a residual
            corr = stats.pearsonr(res_i, res_j)[0]
            P_corr[i, j] = corr
```

Expanded spatial data analytics, spatial statistics.

```
def sqdist3(x1,y1,z1,x2,y2,z2,ind,rotmat):
    """Squared Anisotropic Distance Calculation Given Matrix Indicator - 3D

    This routine calculates the anisotropic distance between two points
    given the coordinates of each point and a definition of the
    anisotropy.

    Converted from original fortran GSLIB (Deutsch and Journel, 1998) to Python by Wendi Liu, University of Texas at Austin
```

Support for 3D workflows.

Michael Pyrcz, The University of Texas at Austin

Introduction to GeostatsPy

Lecture outline . . .

- **The Plan**

The Plan

Demonstrate GeostatsPy for spatial data analytics and geostatistics

From zero we will:

- calculate variograms,
- model variograms
- build estimation models

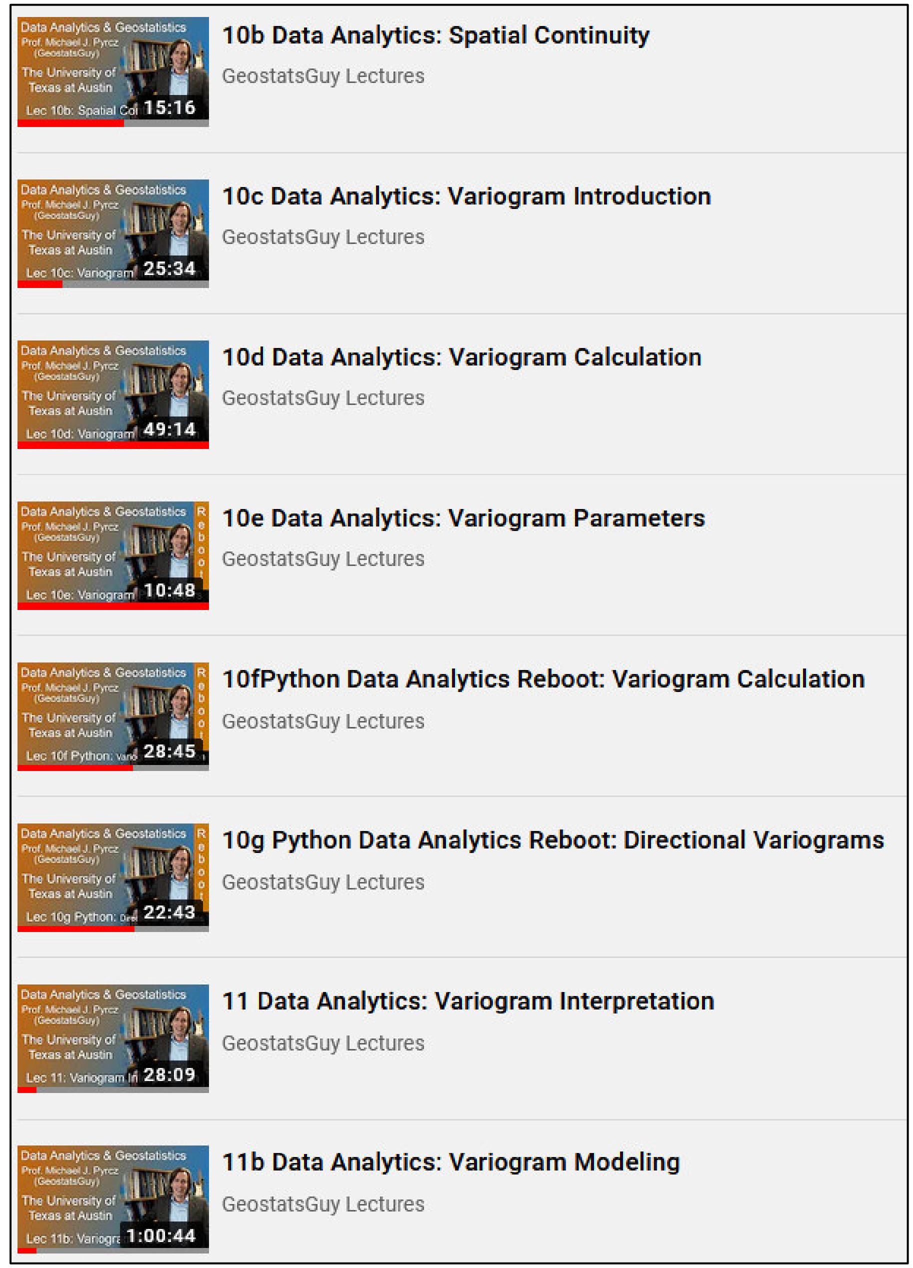
With a 2D synthetic, subsurface dataset.

The Plan
Short Lectures

How are we going to do that in less than 3 hours?

1. Rapidly cover basic concepts in a couple of lecture slides.

- For those interested in deeper knowledge, more details are available in a series of lectures on my YouTube channel, GeostatsGuy Lectures.



Recorded lectures on variograms from the 'GeostatsGuy Lectures' channel.

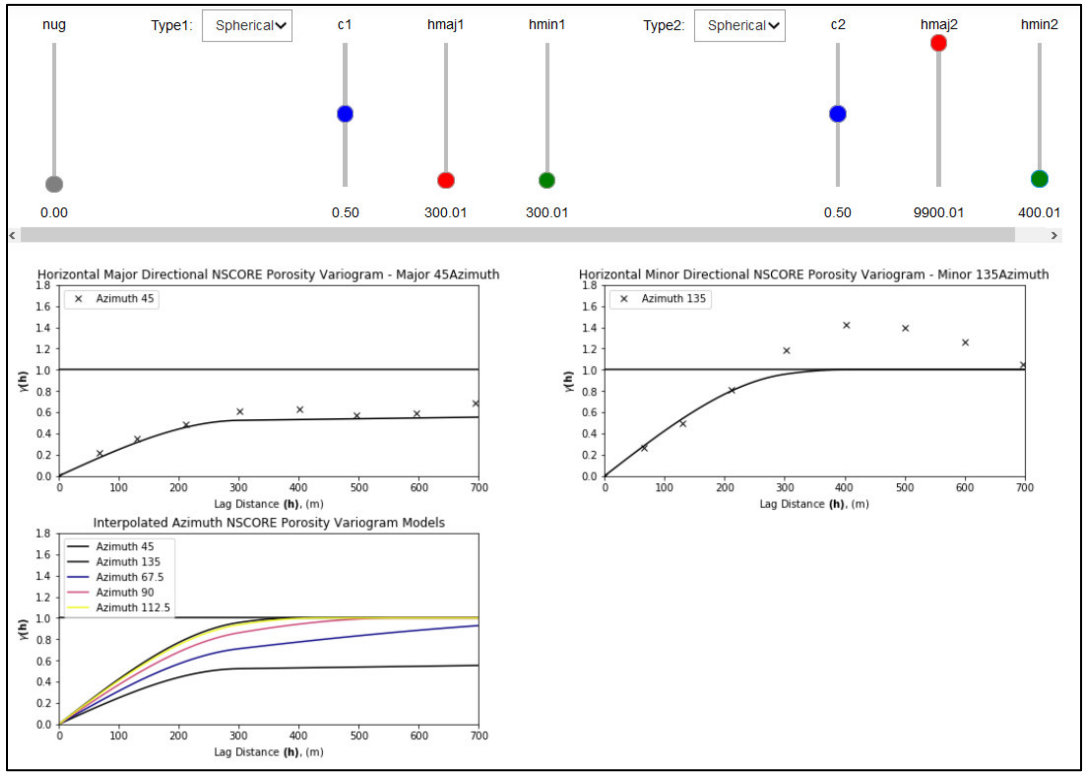
The Plan
Interactive Demos

How are we going to do that in less than 3 hours?

2. Interactive demonstrations for hands-on experiential learning

Interactive demonstrations use GeostatsPy with ipywidgets and matplotlib packages.

Best way to learn about an algorithm is to play with it.



Interactive demonstration for variogram modeling.

The Plan
Interactive Demos

How are we going to do that in less than 3 hours?

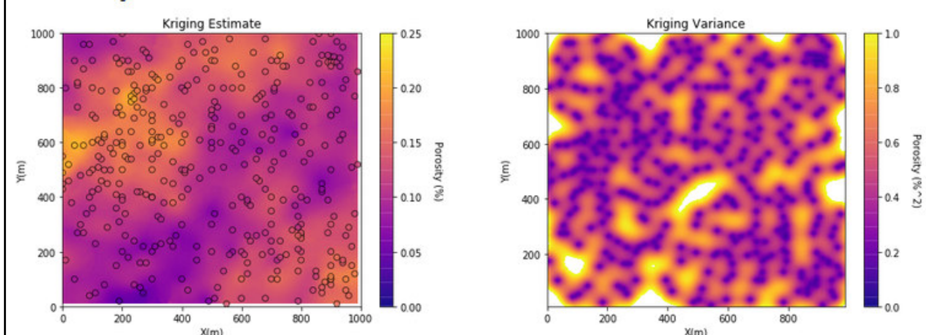
3. Walk-through well documented workflows with GeostatsPy

Flexible building block approach for workflow design.

Let's start with spatial estimates of porosity and permeability with all facies combined. We will also look at the kriging estimation variance.

```
por_kmap, por_vmap = geostats.kb2d(df,'X','Y','Porosity',tmin,tmax,nx,xmn,xsiz,ny,ymn,ysiz,nxdis,nydis,
         ndmin,ndmax,radius,ktype,skmean_por,por_vario)

perm_kmap, perm_vmap = geostats.kb2d(df,'X','Y','Perm',tmin,tmax,nx,xmn,xsiz,ny,ymn,ysiz,nxdis,nydis,
         ndmin,ndmax,radius,ktype,skmean_perm,por_vario)

plt.subplot(221)
GSLIB.locpix_st(por_kmap,xmin,xmax,ymin,ymax,xsiz,0.0,0.25,df,'X','Y','Porosity','Kriging Estimate','X(m)','Y(m)','Porosity (%)',cmap)

plt.subplot(222)
GSLIB.pixelplt_st(por_vmap,xmin,xmax,ymin,ymax,xsiz,0.0,1.0,'Kriging Variance','X(m)','Y(m)','Porosity (%^2)',cmap)

plt.subplot(223)
GSLIB.locpix_st(perm_kmap,xmin,xmax,ymin,ymax,xsiz,0.0,1000,df,'X','Y','Perm','Kriging Estimate','X(m)','Y(m)','Permeability (mD)',cmap)

plt.subplot(224)
GSLIB.pixelplt_st(perm_vmap,xmin,xmax,ymin,ymax,xsiz,0.0,1.0,'Kriging Variance','X(m)','Y(m)','Permeability (mD^2)',cmap)

plt.subplots_adjust(left=0.0, bottom=0.0, right=2.0, top=2.2, wspace=0.3, hspace=0.3)
plt.show()
```

```
Estimated   10000 blocks
    average   0.1203935238945795   variance  0.0007782960645345872
Estimated   10000 blocks
    average   51.30597160497936   variance  13011.658643026816
```

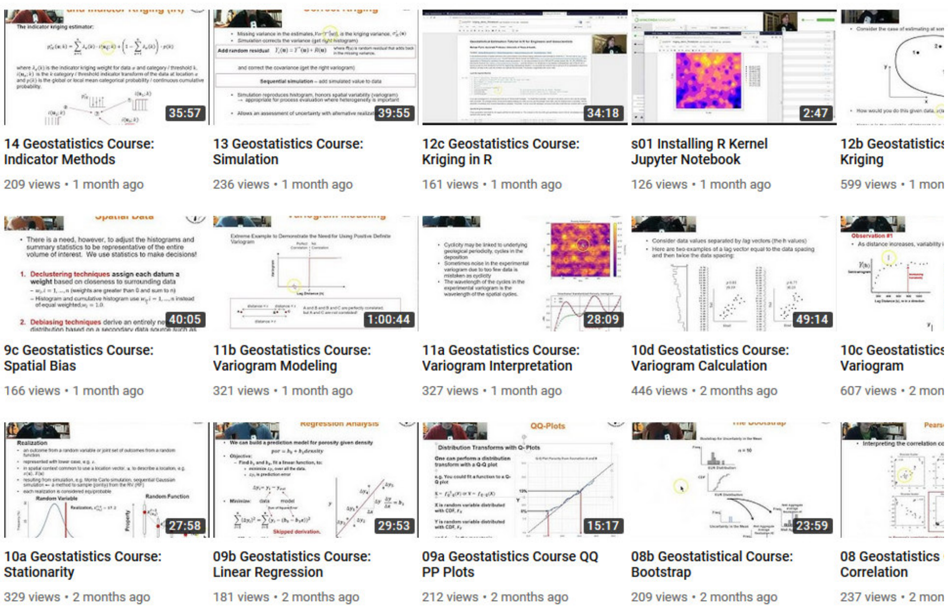


GeostatsPy demonstration for kriging in a well-documented Python Jupyter notebook workflow.

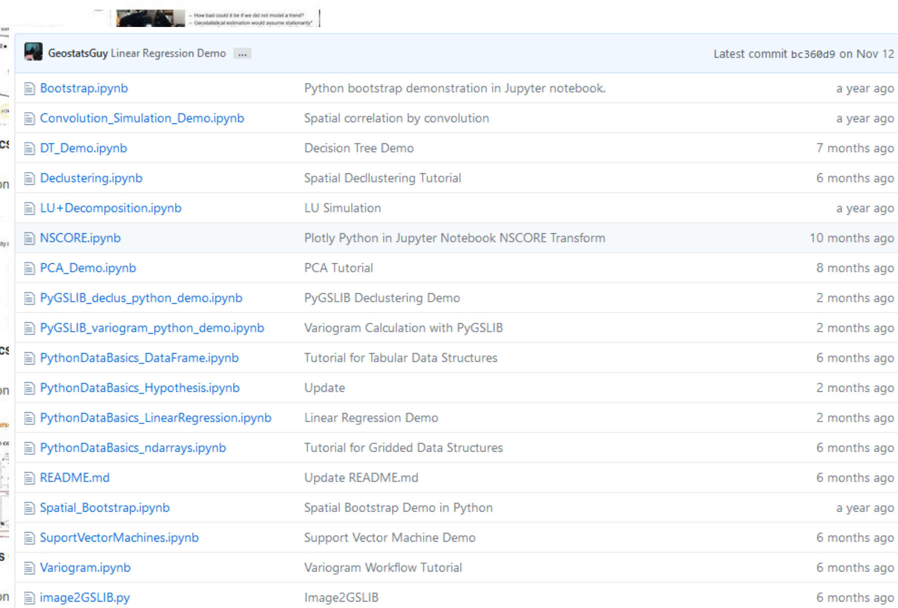
Really, How Will We Learn All of That?

Calibrating Our Goal Today:

1. The goal is to showcase GeostatsPy, not to teach spatial data analytics.
2. Check out my YouTube for greater depth (GeostatsGuy Lectures Channel)
3. All demos, datasets and workflows are available (GitHub GeostatsGuy).



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
Declustering.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_declus_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
SuportVectorMachines.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

Michael Pyrcz, The University of Texas at Austin

Introduction to GeostatsPy

Lecture outline . . .

- **Getting Setup**

Getting Setup on Your Laptop

Time to Get Technical, For Next Class

1. Install Anaconda 3.7 on your Laptop, that's Python!

2. Install the GeostatsPy, my geostatistics python package

3. Bring your laptop to class from now on

We are going to start working through methods / workflows in class

Getting Setup

To participate in this Tutorial.

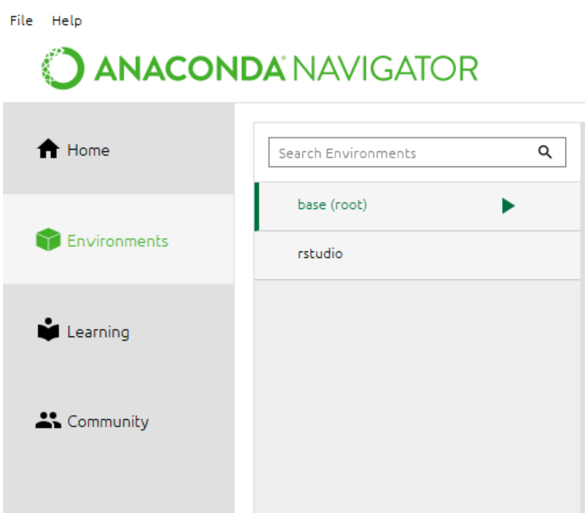
- Install Anaconda 3.7 and the GeostatsPy package.
- Install Anaconda from: https://www.anaconda.com/products/individual

Anaconda installers from anaconda.com.

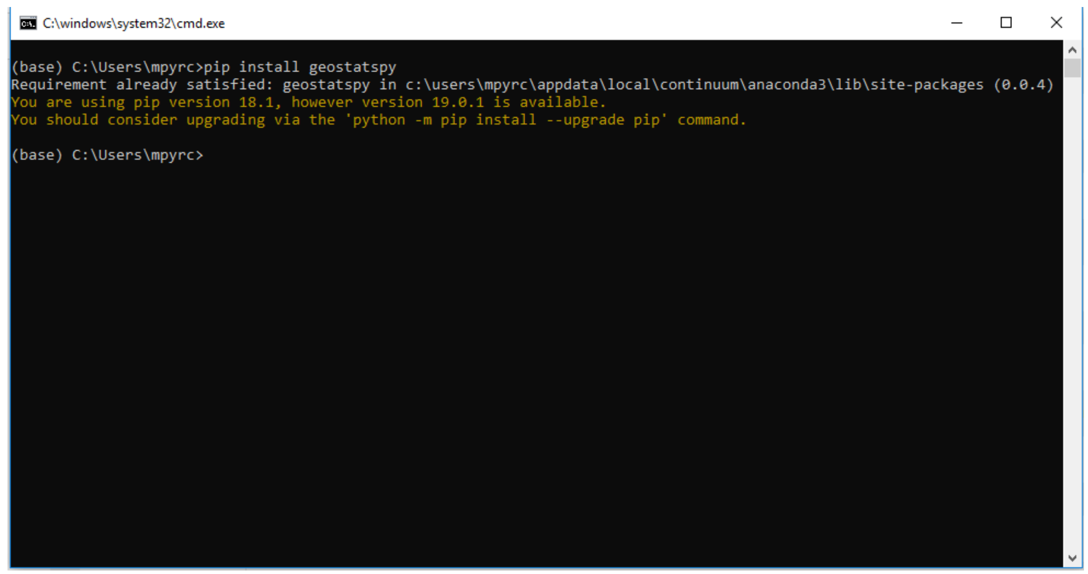
Getting Setup

To participate in this Tutorial.

- Install Anaconda 3.7 and the GeostatsPy package.
- Install GeostatsPy package from PyPI repository.
- One method to ensure GeostatsPy is available from Jupyter notebook.



1. Open anaconda navigator, select Environments and click green arrow besides base and select 'open terminal'.



2. In the terminal type 'pip install geostatspy'.

Introduction to GeostatsPy

Lecture outline . . .

- **Call to Action**

Call to Action

Contributions are welcome, consider:

- docs
- optimization
- coding standards
- code readability
- testing
- examples
- expansion

Introduction to GeostatsPy

Lecture outline . . .

- **Who am I?**
- **Introduction to GeostatsPy**
- **The Plan**
- **Getting Setup**
- **Call to Action**