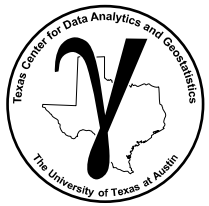


Introduction to GeostatsPy

Conclusions

- Concluding Remarks
- Plus / Delta Feedback

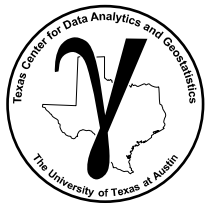
Motivation

“Promote use of and contributions to the GeostatsPy open source package for enabling spatial data analytics and geostatistics in Python workflows.”

“Support education, students, working professionals and potential students”

“Expand the use of spatial data analytics and machine learning in the subsurface.”

“Make data analytics and machine learning workflows accessible to all scientists and engineers.”



Introduction to GeostatsPy

Conclusions

- Concluding Remarks

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 tests for Travis continuous integration

Michael Pyrcz – GSLIB and geostats modules

I would love to see this expand

Current State of GeostatsPy

What is good?

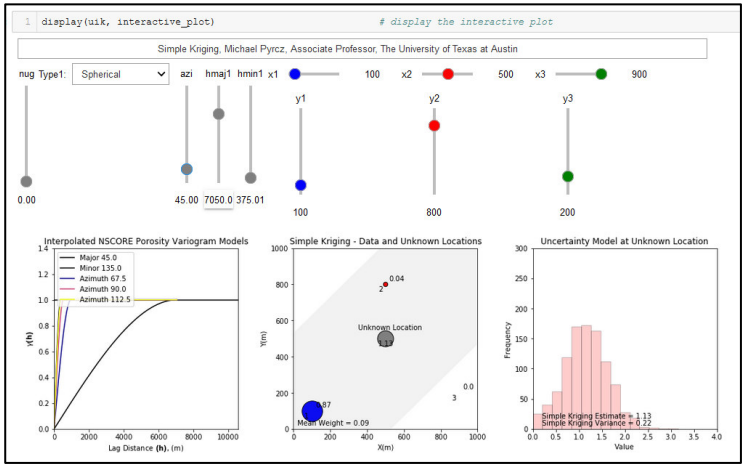
docs passing

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.

Michael Pyrcz, The University of Texas at Austin

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.

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
```

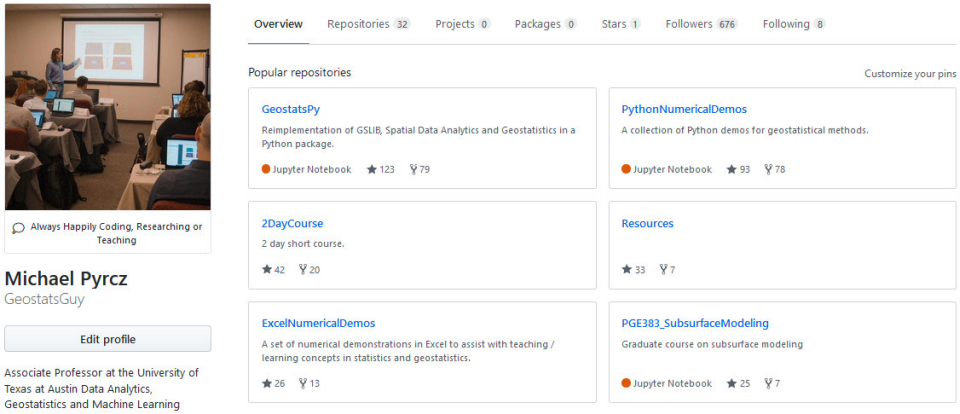
Expanded spatial data analytics, spatial statistics.

```
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
```

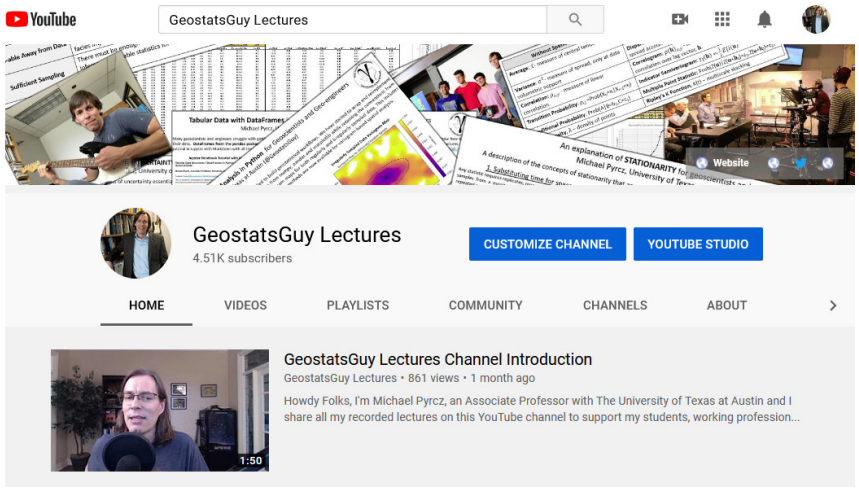
Support for 3D workflows.

Michael Pyrcz, The University of Texas at Austin

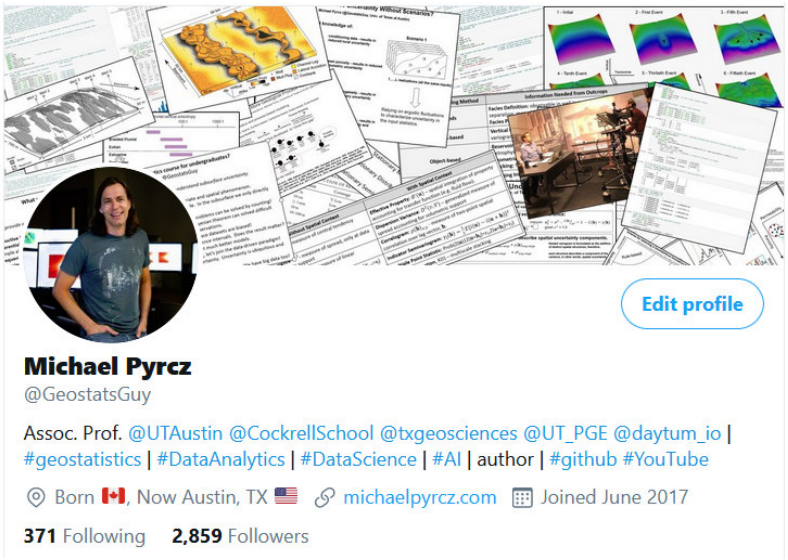
More Resources on Subsurface Data Analytics and Machine Learning



For a lot of content, demonstrations in Python, R and Excel, follow on GitHub.



For all my lectures, subscribe on YouTube.



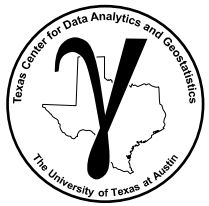
Daily technical sharing, follow on Twitter.

Ask Me About:

DIRECT Consortium for Subsurface Data Analytics and Machine Learning

daytum for Energy Data Science education.

Michael Pyrcz, The University of Texas at Austin



Introduction to GeostatsPy

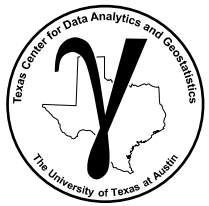
Conclusions

- **Plus / Delta Feedback**

Plus / Delta Feedback

Plus	Delta

Michael Pyrcz, The University of Texas at Austin



Introduction to GeostatsPy

Conclusions

- Some Comments
- Plus / Delta Feedback