Clone the repo

- Please download the class repo here (245MB):

  https://agilegeo.s3.amazonaws.com/geocomp-mastery.zip

- After unpacking the ZIP file, navigate to it in the command prompt.

Agile*

https://ageo.co/IlCGdt

Geocomputing mastery

A Python course for digital subsurface scientists and engineers

Matt Hall, Diego Castañeda, John Leeman

https://ageo.co/IlCGdt

Learning objectives

1. Ensure you know how to maintain a coding environment.
2. Know how to read and write object-oriented code.
3. Know how to make a command-line interface.
4. Know how to package a program for others to install.
5. Have a plan for, and make a start on, your own project.

https://ageo.co/IlCGdt

Supporting concepts

- Documentation basics (see notebook)
- Testing basics (see notebook)
- Exceptions and warnings
- Writing data to files or databases (see notebook)
- Quick introduction to `xarray` (see notebook)
- Logging

Let's get set up

The DOS command shell

https://ageo.co/IlCGdt

Open the Anaconda prompt

Useful commands:

`cd`	See the name of the current directory
`cd <dir>`	Change to `<dir>` directory
`cd ..`	Change to parent directory
`dir`	See the files in the current directory
`copy <f1> <f2>`	Copy file `<f1>` to file `<f2>`
`del <f1>`	Delete file `<f1>`

Using conda with a proxy

- **conda** is a package manager and also manages 'virtual environments' (called envs)
- The **.condarc** file initializes some of the environment for conda.
- It's an invisible file (that's what the dot at the start of the name means) so it might not appear to have a name, depending on your OS.
- If you connect to the Internet with a proxy, you'll need to do this (change **eu** to **ua** for Americas):

```
conda config --set proxy_servers.http http://proxy-eu.shell.com:8080
conda config --set proxy_servers.https https://proxy-eu.shell.com:8080
conda config --set ssl_verify false
```

- This edits your **.condarc** file for you.
- More info about **.condarc**: https://conda.io/docs/user-guide/configuration/use-condarc.html

pip.ini file

- **`pip`** is a package manager.
- The **`pip.ini`** file initializes some of the environment for `pip`.
- It's an invisible file (that's what the dot at the start of the name means) so it might not appear to have a name, depending on your OS.
- If you connect to the Internet with a proxy, you'll need to edit the file.
- Add the following to `%APPDATA%\pip\pip.ini` using a text editor (change **eu** to **ua** for Americas):

```
[global]
proxy = https://proxy-eu.shell.com:8080
```

- To check the file from cmd, do the following:

```
C:\> type %APPDATA%\pip\pip.ini
[global]
proxy = proxy-eu.shell.com:8080
```

- Alternatively, use pip like: `pip --proxy https://proxy-eu.shell.com:8080 install welly bruges segyio`
- More info about `pip.ini`: https://pip.pypa.io/en/stable/user_guide/#config-file

Set up **conda** and a new 'env'

https://ageo.co/IlCGdt

To build an environment called *mastery* using Python 3.7 and all the libraries in Anaconda, type the following commands inside the **Anaconda Prompt:**

```
conda create --name mastery python=3 anaconda
conda activate mastery
python -m ipykernel install --user --name mastery
```

Getting into an existing environment

Open an Anaconda prompt, then type (for the `geocomp` env):

```
conda activate geocomp
cd <project folder>
jupyter notebook
```

Change `<project folder>` to wherever your repo is (i.e. where your notebooks are).

Making a new environment

https://ageo.co/IlCGdt

If you start a new coding project, it is probably worth creating a new environment.

Open Anaconda prompt, then type:

```
conda create -n myenv python=3.7 anaconda
conda activate myenv
python -m ipykernel install --user --name myenv
jupyter notebook
```

Change `myenv` to any name you want and the version of Python if you need another version. If you only want a few packages, change `anaconda` to, say, `scipy pandas`

Installing Equinor's segyio package

https://ageo.co/IlCGdt

First, activate the environment you want the software in. Then try this:

```
pip install segyio
```

If it seemed to install but doesn't import when you're in a notebook or Python prompt, then try this:

```
pip uninstall segyio
pip install segyio=1.4.1a
```

Installing other packages

If you find a package you'd like to use, e.g. called *foo*, check if it's in your environment already by typing `import foo` at the Python prompt or in a notebook. If you get an error, you likely need to install the package. Usually you can open a new Anaconda prompt and do this:

```
conda activate myenv

conda install foo
```

If that doesn't work, try (still in the activated environment):

```
pip install foo
```

If that doesn't work either, you'll have to find the package's home page or GitHub page and look for installation instructions.

Interactive widgets

Usually you can just do this in your Anaconda prompt, after activating your environment:

```
conda install ipywidgets
```

Sometimes you also need to type this after installing:

```
jupyter nbextension enable --py widgetsnbextension
```

Using git to clone a repo

https://ageo.co/IlCGdt

Install git with

```
conda install git
```

To use someone else's repo from GitHub, you can clone it (i.e. download it locally):

```
git clone https://github.com/otheruser/project.git
```

Then cd to that directory to use the code or perform git commands on it. If you change the repo, you can commit your own changes:

```
git commit -am "improved fig 1"
```

If you add files, then you must add them to version control with add (change <filename> to your file):

```
git add <filename>
```

Or stash (forget) them:

```
git stash
```

After committing or stashing, you can update the repo from its original source when it changes:

```
git pull
```

Making a new repo of your own

Create an account on GitHub.com (or try GitLab.com)

Create a new, empty, repo. Add a license and README if you want to.

Clone the new empty repo (see previous slide)

Add your file(s) to it, then add them to version:

```
git add -A
```

Commit those changes with something like (from the repo directory):

```
git commit -am "added files"
```

Push to the origin (i.e. online, GitHub-hosted) repo:

```
git push origin master
```

After more changes, do these last two steps again.

Share your awesome new repo with others!

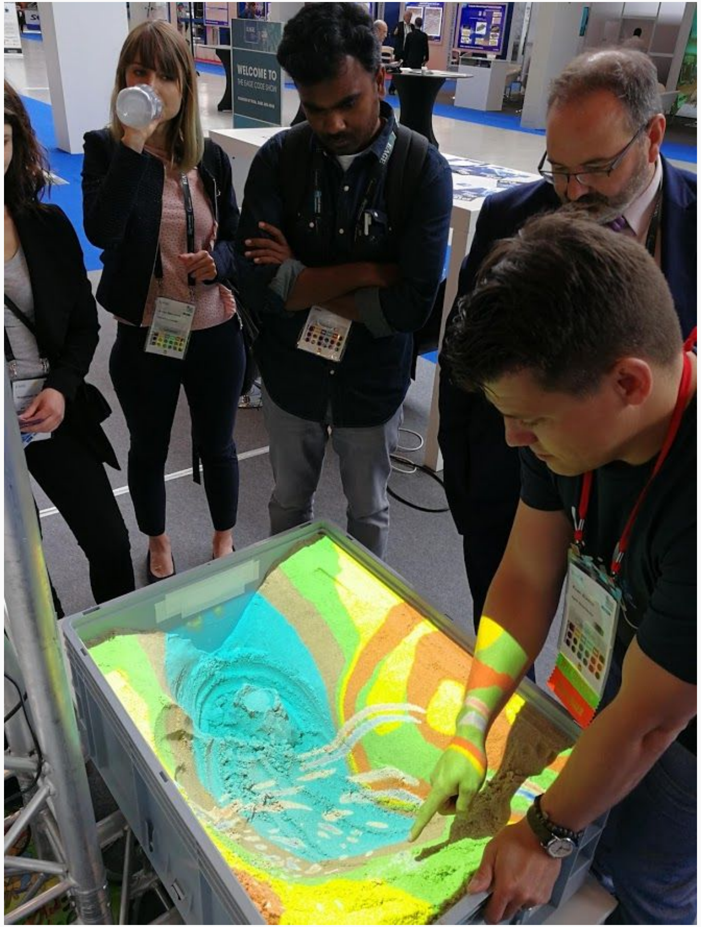
Make a file called `.gitignore` to list any files to ignore (e.g. data files

Finding a project

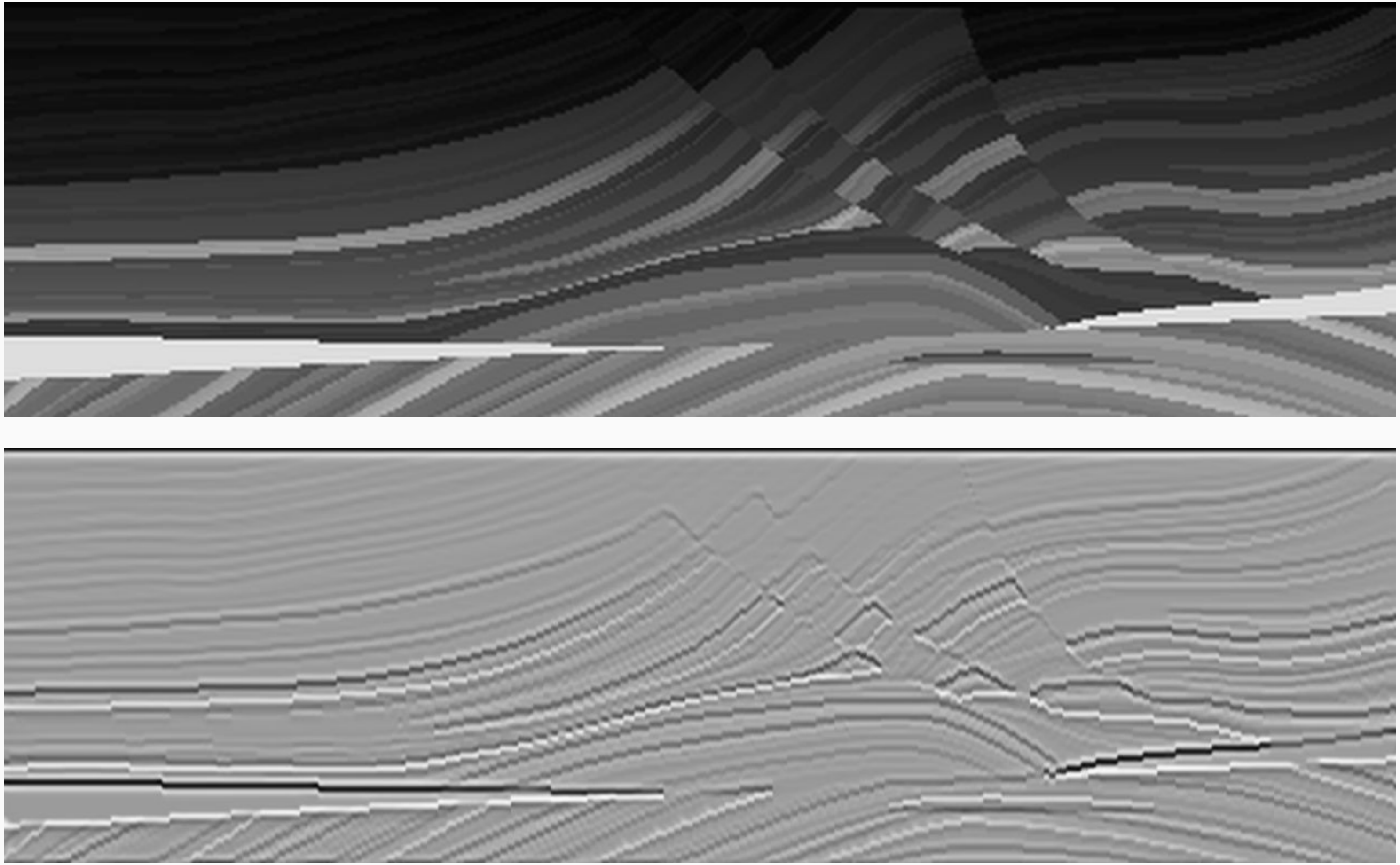
https://ageo.co/IlCGdt

Objects to model

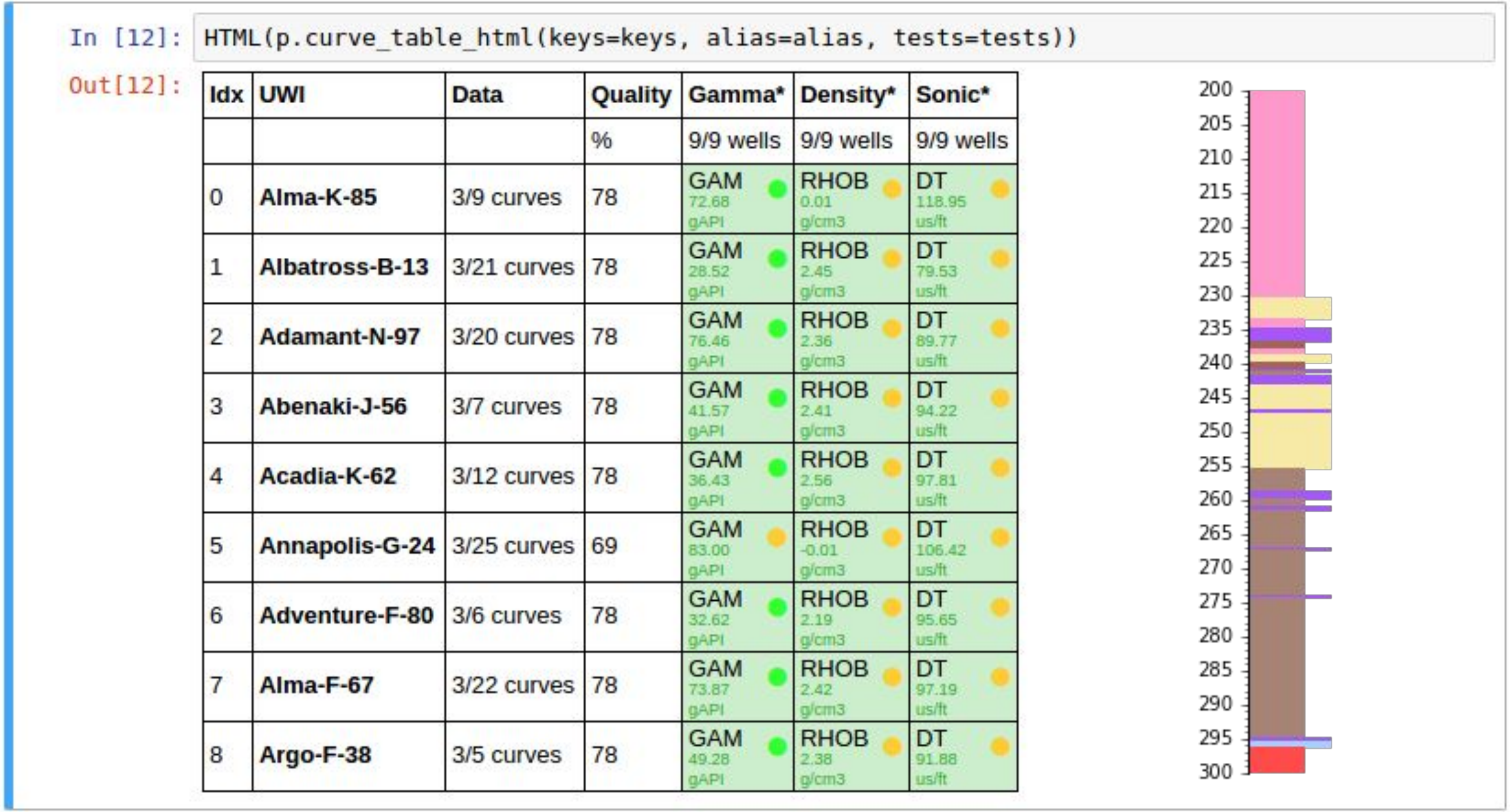
- Core
- Core sample (SWC, SCAL)
- Well log
- Well (borehole)
- Seismic shot record
- Seismic trace (or line, or vol)
- Seismic survey plan
- Synthetic seismogram
- Well test or production log
- Basin or play
- Stratigraphic column
- Formation (top pick, map)
- Prospect or field
- Well report
- Seismic processing report
- Outcrop



The AR sandbox at EAGE 2018



GAN-based modeling and inversion, Lukas Mosser et al., Paris 2017

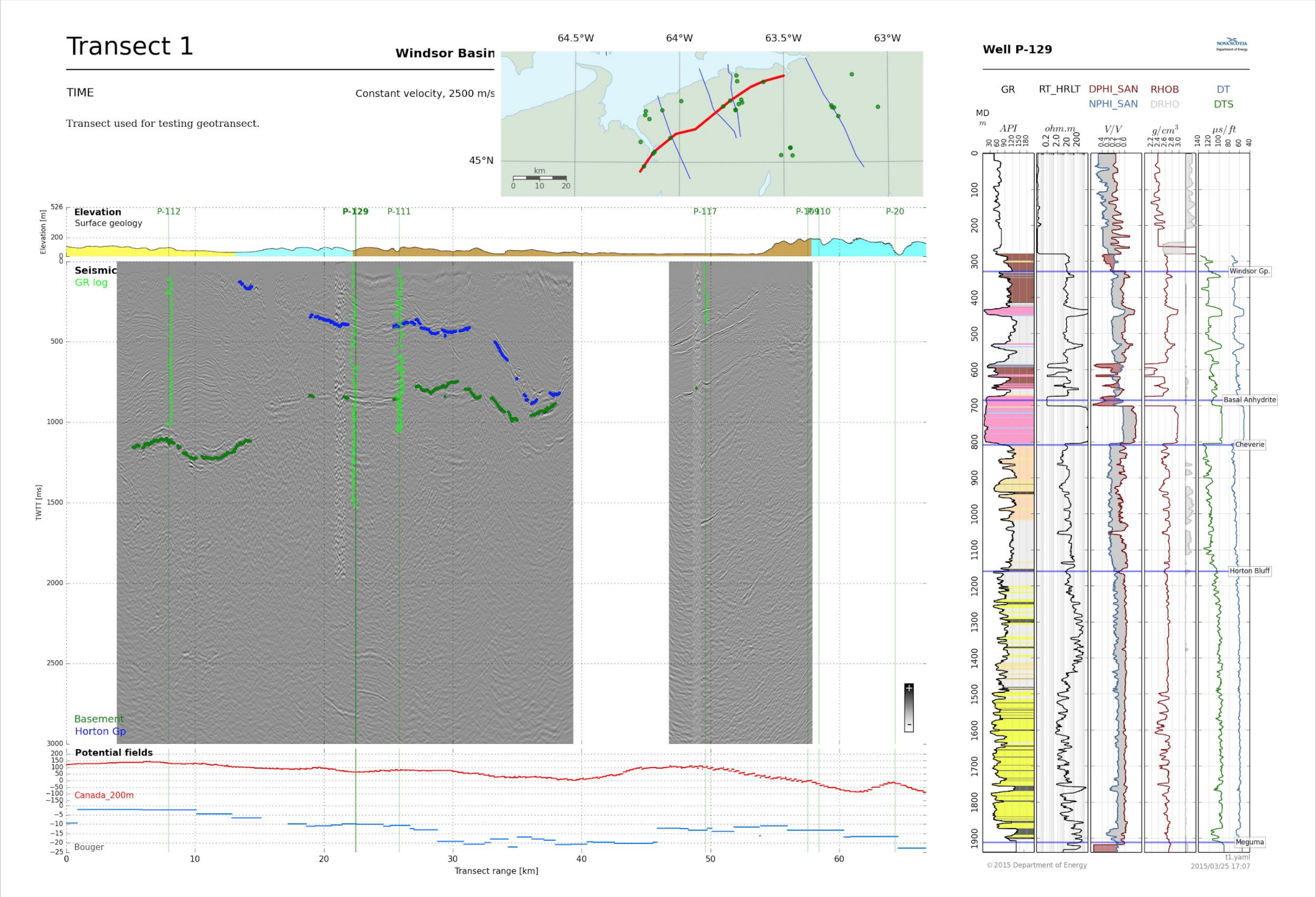


```
In [12]: HTML(p.curve_table_html(keys=keys, alias=alias, tests=tests))
```

Out[12]:

Idx	UWI	Data	Quality	Gamma*	Density*	Sonic*
			%	9/9 wells	9/9 wells	9/9 wells
0	**Alma-K-85**	3/9 curves	78	GAM 🟢 72.68 gAPI	RHOB 🟠 0.01 g/cm3	DT 🟠 118.95 us/ft
1	**Albatross-B-13**	3/21 curves	78	GAM 🟢 28.52 gAPI	RHOB 🟠 2.45 g/cm3	DT 🟠 79.53 us/ft
2	**Adamant-N-97**	3/20 curves	78	GAM 🟢 76.46 gAPI	RHOB 🟠 2.36 g/cm3	DT 🟠 89.77 us/ft
3	**Abenaki-J-56**	3/7 curves	78	GAM 🟢 41.57 gAPI	RHOB 🟠 2.41 g/cm3	DT 🟠 94.22 us/ft
4	**Acadia-K-62**	3/12 curves	78	GAM 🟢 36.43 gAPI	RHOB 🟠 2.56 g/cm3	DT 🟠 97.81 us/ft
5	**Annapolis-G-24**	3/25 curves	69	GAM 🟠 83.00 gAPI	RHOB 🟠 -0.01 g/cm3	DT 🟠 106.42 us/ft
6	**Adventure-F-80**	3/6 curves	78	GAM 🟢 32.62 gAPI	RHOB 🟠 2.19 g/cm3	DT 🟠 95.65 us/ft
7	**Alma-F-67**	3/22 curves	78	GAM 🟢 73.87 gAPI	RHOB 🟠 2.42 g/cm3	DT 🟠 97.19 us/ft
8	**Argo-F-38**	3/5 curves	78	GAM 🟢 49.28 gAPI	RHOB 🟠 2.38 g/cm3	DT 🟠 91.88 us/ft

Welly and striplog projects, Agile, 2015–17



Transect 1

Windsor Basin

TIME

Constant velocity, 2500 m/s

Transect used for testing geotransect.

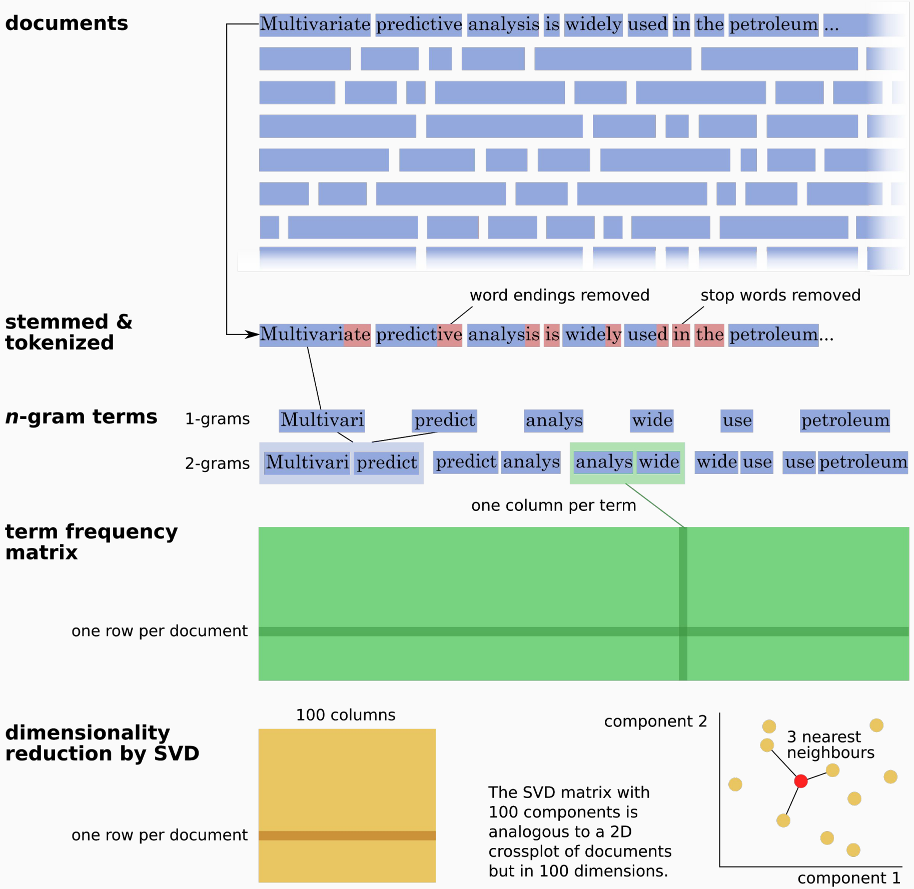
Well P-129

© 2015 Department of Energy

t1.yaml

2015/03/25 17:07

Geotransect project, Agile, 2015



documents

Multivariate predictive analysis is widely used in the petroleum ...

stemmed & tokenized

word endings removed

stop words removed

Multivariate predictive analysis is widely used in the petroleum...

***n*-gram terms**

1-grams: Multivari predict analys wide use petroleum

2-grams: Multivari predict, predict analys, analys wide, wide use, use petroleum

one column per term

term frequency matrix

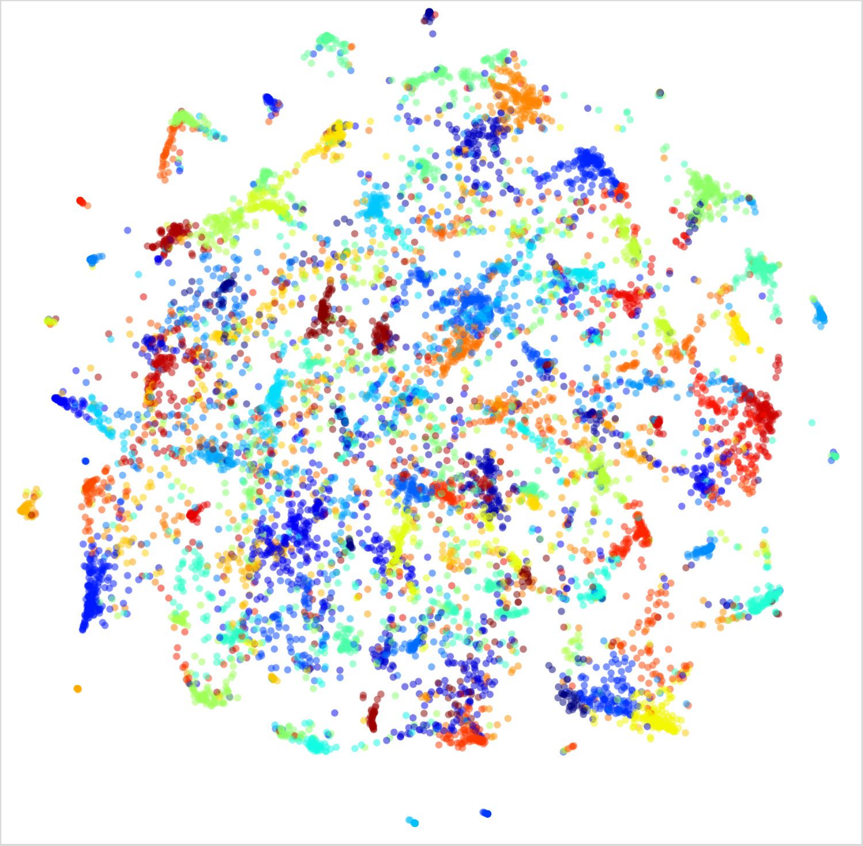
one row per document

dimensionality reduction by SVD

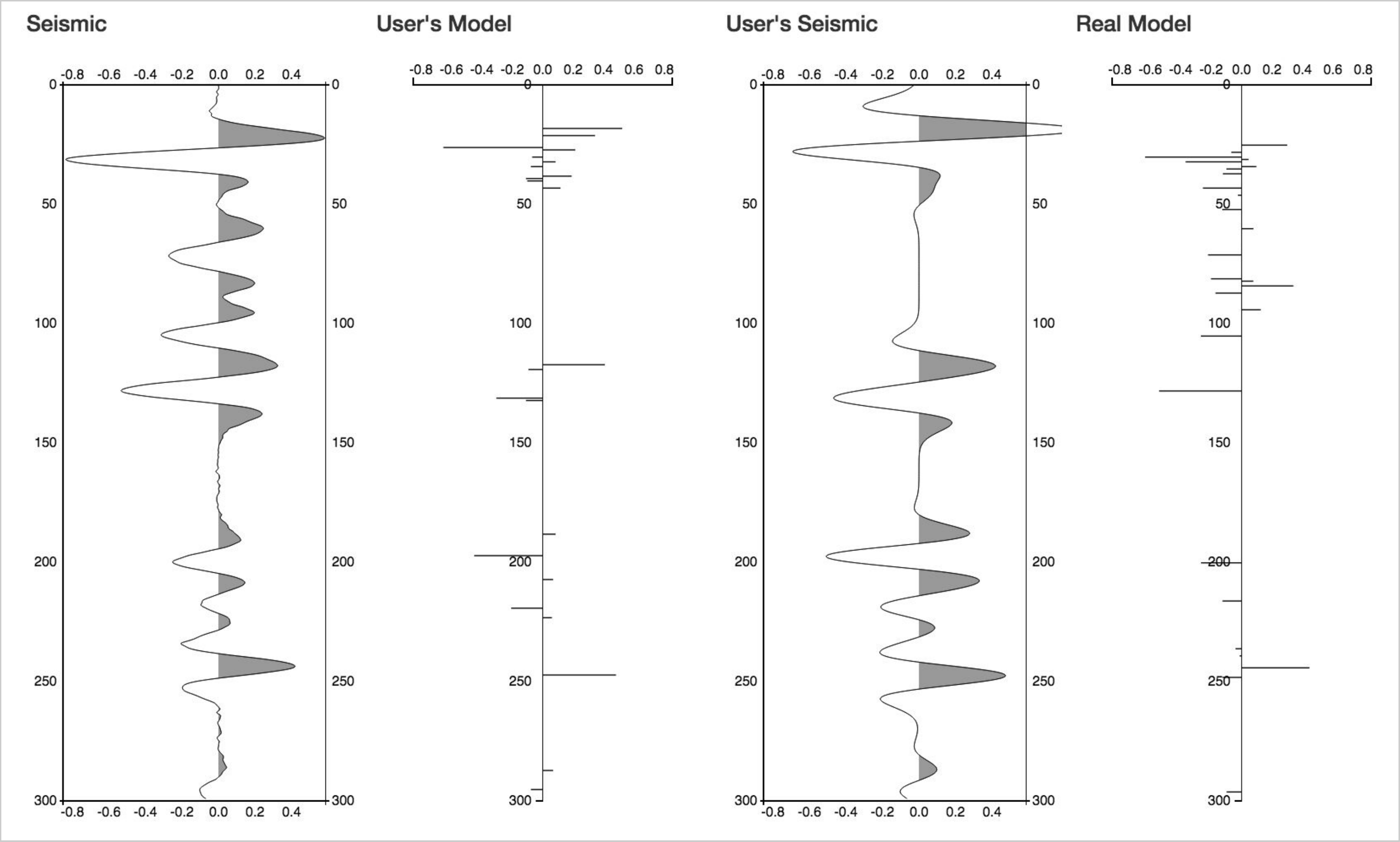
100 columns

one row per document

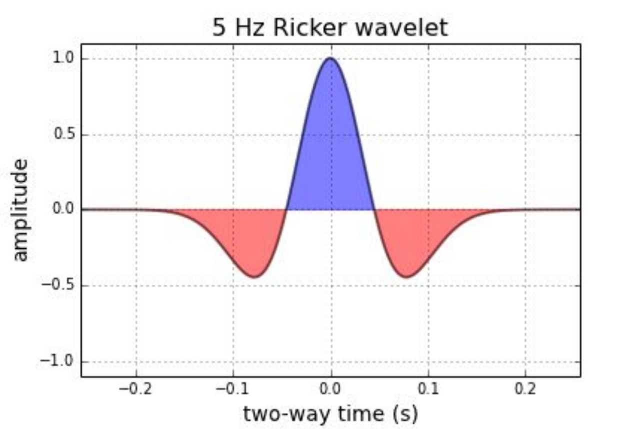
The SVD matrix with 100 components is analogous to a 2D crossplot of documents but in 100 dimensions.

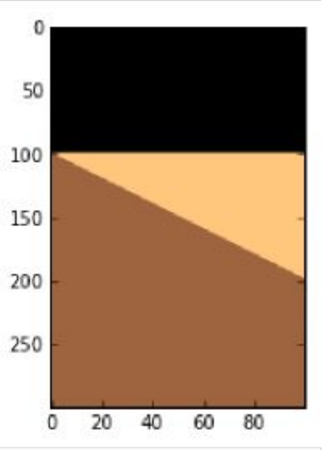


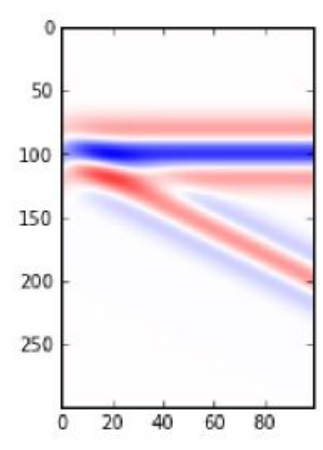
Georx project, Agile, 2017

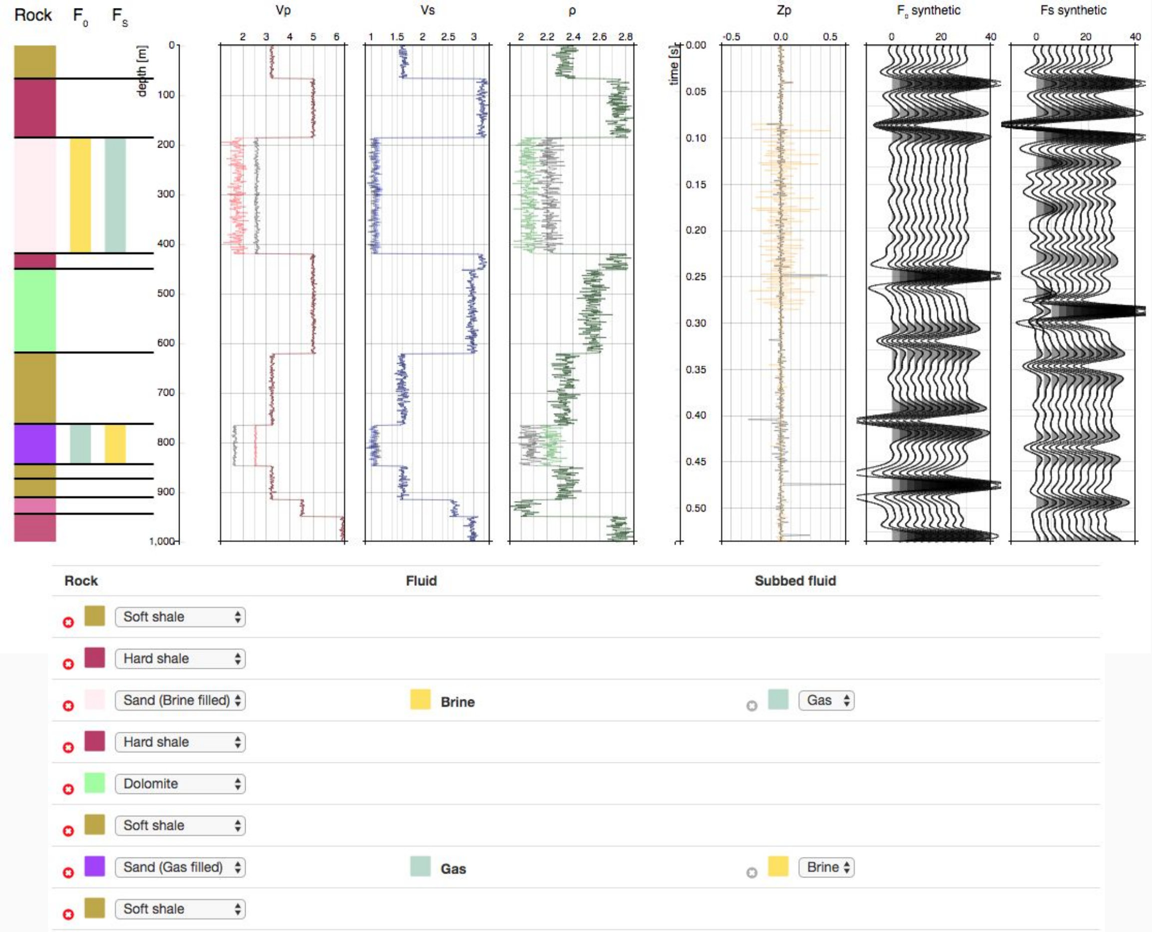


Human inversion game, Steve Purves et al, Vienna 2016



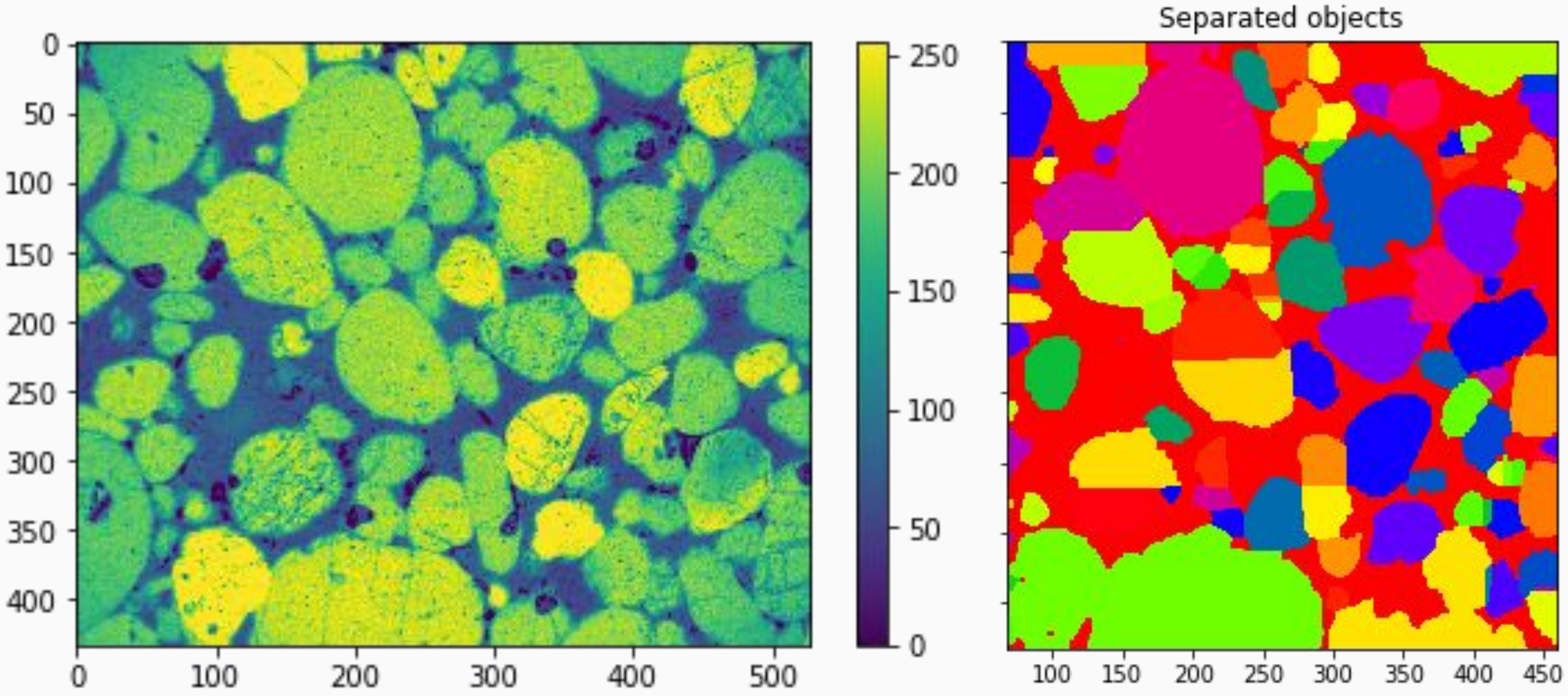






Rock	Fluid	Subbed fluid
⊗ Soft shale		
⊗ Hard shale		
⊗ Sand (Brine filled)	🟨 Brine	○ 🟩 Gas
⊗ Hard shale		
⊗ Dolomite		
⊗ Soft shale		
⊗ Sand (Gas filled)	🟩 Gas	○ 🟨 Brine
⊗ Soft shale		

Forward modeling tools, Agile



https://www.dropbox.com/s/cqa1a7ip45cpi57/Image_segmentation.ipynb?dl=0

Resources

Python tutorials

- A Whirlwind Tour of Python – The essentials from the essential Jake Vanderplas.
- Learn X in Y minutes — If you just want to get cracking.
- Stavros — If you want to know a bit more.
- Robert Johansson's lectures
- Tutorials Point — Another option.
- Code Academy — A more sedate pace.
- Udacity Intro to Computer Science — Fantastic but a serious undertaking.
- All the tutorials!

Geoscience Python tutorials

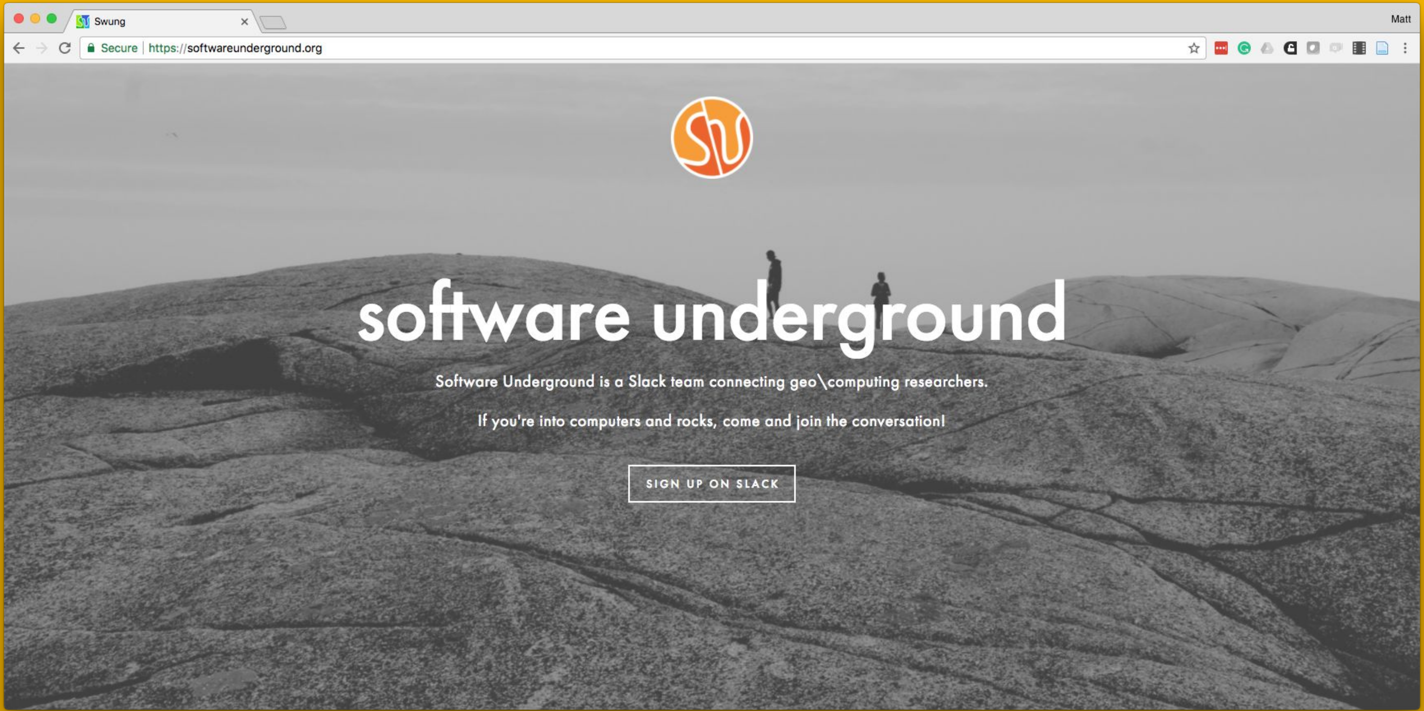
- Agile's: x-lines of Python series – GitHub repo of notebooks doing something neat in less than X lines of code (where x < 10 or so).
- SEG's Geophysical Tutorials – GitHub repo of notebooks and papers implementing geophysical workflows, mostly in Python.
- Geology and Python – an awesome new blog about... geology and Python!
- Geosci.xyz – a collection of geoscience software and widgets, in Python and JavaScript.

Recommended MOOCs and books

- Udacity, Computer Science 101
- Coursera, Applied machine learning in Python
- Coursera, Deep learning specialization

- *Data Science from Scratch*, by Joel Grus (brilliant intro to machine learning)
- *Effective Computation in Physics*, by Anthony Scopatz and Kathryn Huff (not really just about physics, this is a great book about Python and other tools for computational scientists)

Software Underground is a Slack group for geoscientists who use computers in their work or research.



Software Underground is a Slack team connecting geo\computing researchers.

If you're into computers and rocks, come and join the conversation!

Go to **softwareunderground.org** or **swung.slack.com** to join in!