Workshop on the Next-Generation Liquefaction Database

Mong Learning Center, UCLA - September 24, 2018

UCLA Samueli School of Engineering

Next Generation Liquefaction (NGL) Database Interactive Tools

Honor Fisher & Allison Lee
Undergraduate Scientists

NGL NEXT GENERATION LIQUEFACTION

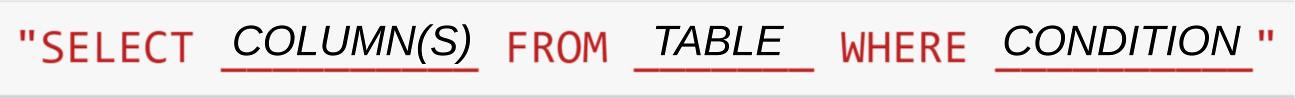
Engineer Change.

Outline

- Querying Data
  - Using Jupyter Notebook
- Creating Post-Processing Tools
  - Interactive Soil Test Plot
  - Digital Elevation Model

Querying Data

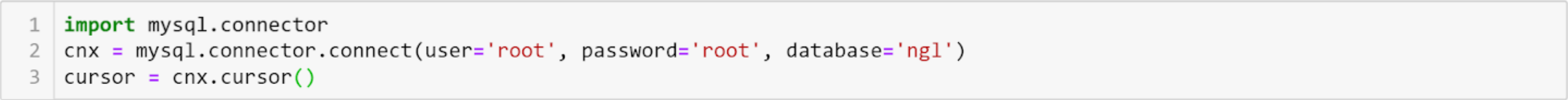
- Allows further interaction with queried data
- Uses the following format:



```
"SELECT COLUMN(S) FROM TABLE WHERE CONDITION "
```

Querying Data with Jupyter Notebook

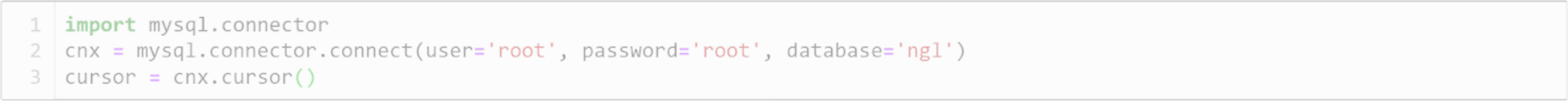
- Establishes a connection with the database



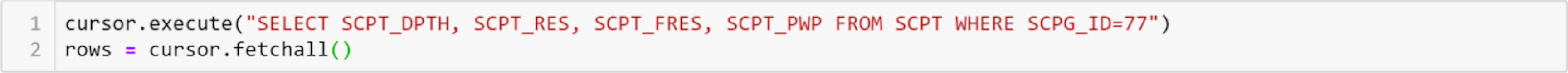
```
import mysql.connector
cnx = mysql.connector.connect(user='root', password='root', database='ngl')
cursor = cnx.cursor()
```

Querying Data with Jupyter Notebook

- Executes the query



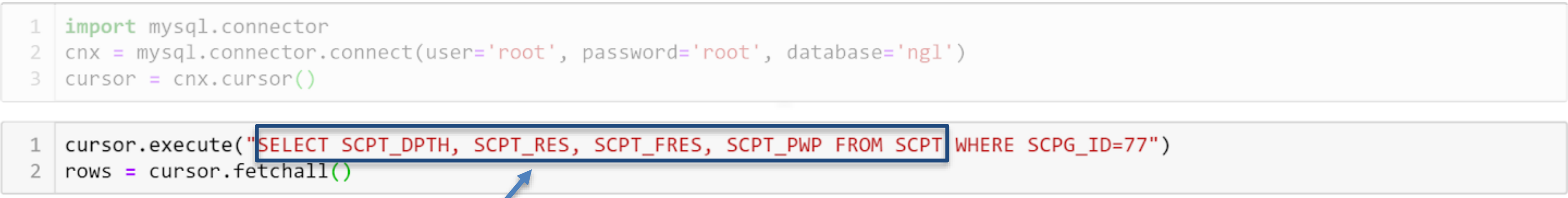
```
import mysql.connector
cnx = mysql.connector.connect(user='root', password='root', database='ngl')
cursor = cnx.cursor()
```



```
cursor.execute("SELECT SCPT_DPTH, SCPT_RES, SCPT_FRES, SCPT_PWP FROM SCPT WHERE SCPG_ID=77")
rows = cursor.fetchall()
```

Querying Data with Jupyter Notebook

- Executes the query



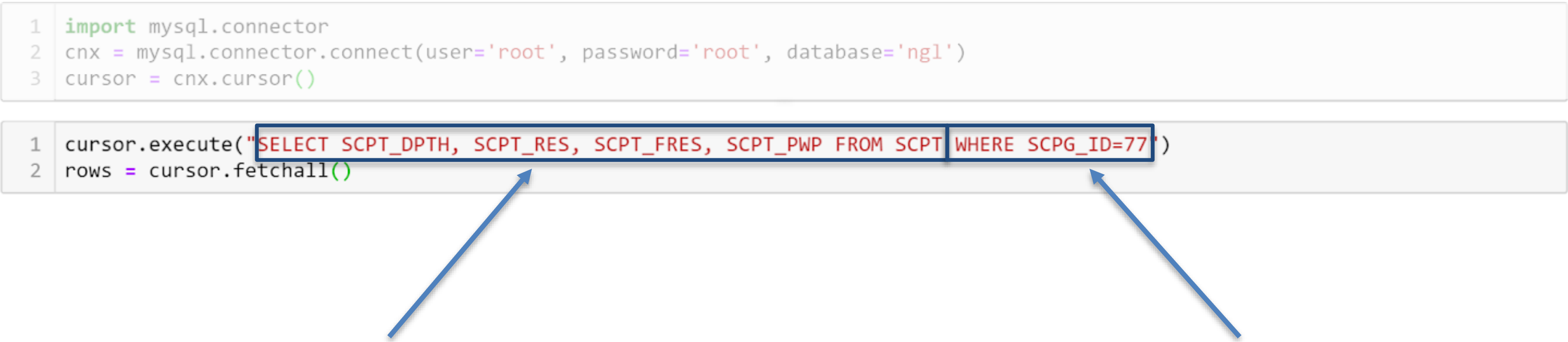
```python
import mysql.connector
cnx = mysql.connector.connect(user='root', password='root', database='ngl')
cursor = cnx.cursor()
```

```python
cursor.execute("SELECT SCPT_DPTH, SCPT_RES, SCPT_FRES, SCPT_PWP FROM SCPT WHERE SCPG_ID=77")
rows = cursor.fetchall()
```

Gets CPT data
from database

Querying Data with Jupyter Notebook

- Executes the query



```python
import mysql.connector
cnx = mysql.connector.connect(user='root', password='root', database='ngl')
cursor = cnx.cursor()
```

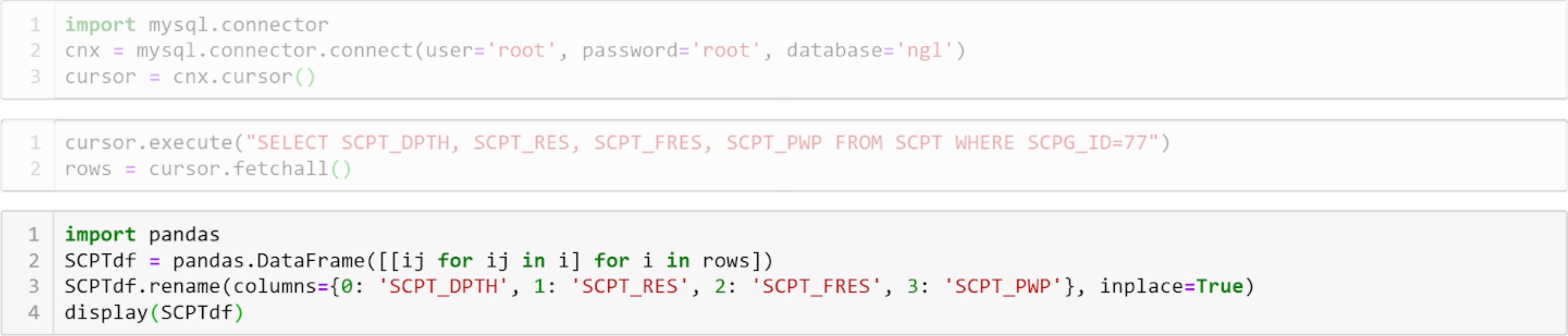
```python
cursor.execute("SELECT SCPT_DPTH, SCPT_RES, SCPT_FRES, SCPT_PWP FROM SCPT WHERE SCPG_ID=77")
rows = cursor.fetchall()
```

Gets CPT data
from database

For one specific
test

Querying Data with Jupyter Notebook

- Converts the data to a pandas data frame



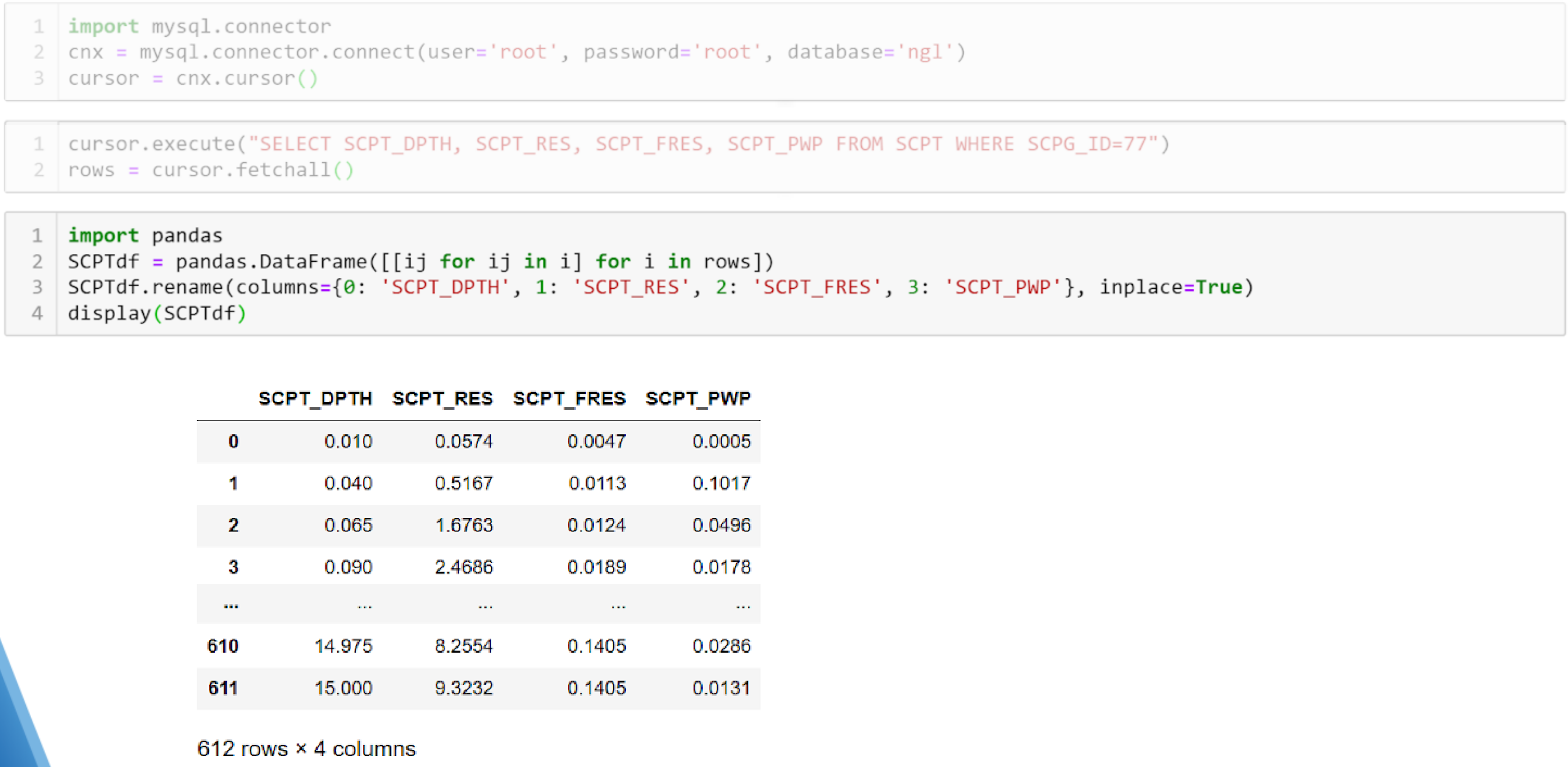
```python
import mysql.connector
cnx = mysql.connector.connect(user='root', password='root', database='ngl')
cursor = cnx.cursor()
```

```python
cursor.execute("SELECT SCPT_DPTH, SCPT_RES, SCPT_FRES, SCPT_PWP FROM SCPT WHERE SCPG_ID=77")
rows = cursor.fetchall()
```

```python
import pandas
SCPTdf = pandas.DataFrame([[ij for ij in i] for i in rows])
SCPTdf.rename(columns={0: 'SCPT_DPTH', 1: 'SCPT_RES', 2: 'SCPT_FRES', 3: 'SCPT_PWP'}, inplace=True)
display(SCPTdf)
```

Querying Data with Jupyter Notebook

- Converts the data to a pandas data frame



```
import mysql.connector
cnx = mysql.connector.connect(user='root', password='root', database='ngl')
cursor = cnx.cursor()
```

```
cursor.execute("SELECT SCPT_DPTH, SCPT_RES, SCPT_FRES, SCPT_PWP FROM SCPT WHERE SCPG_ID=77")
rows = cursor.fetchall()
```

```
import pandas
SCPTdf = pandas.DataFrame([[ij for ij in i] for i in rows])
SCPTdf.rename(columns={0: 'SCPT_DPTH', 1: 'SCPT_RES', 2: 'SCPT_FRES', 3: 'SCPT_PWP'}, inplace=True)
display(SCPTdf)
```

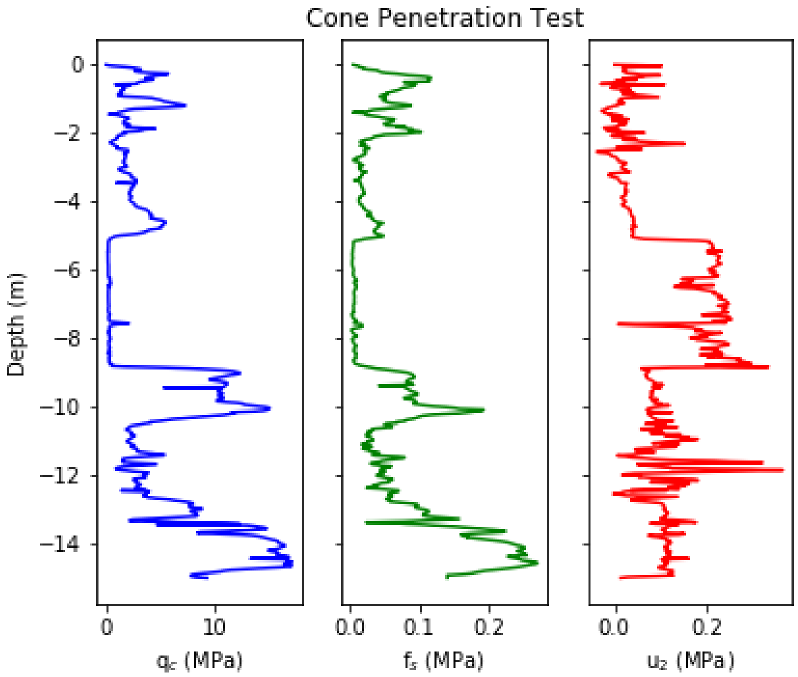
```
     SCPT_DPTH  SCPT_RES  SCPT_FRES  SCPT_PWP
  0      0.010    0.0574     0.0047    0.0005
  1      0.040    0.5167     0.0113    0.1017
  2      0.065    1.6763     0.0124    0.0496
  3      0.090    2.4686     0.0189    0.0178
...        ...       ...        ...       ...
610     14.975    8.2554     0.1405    0.0286
611     15.000    9.3232     0.1405    0.0131

612 rows × 4 columns
```

UCLA Samueli

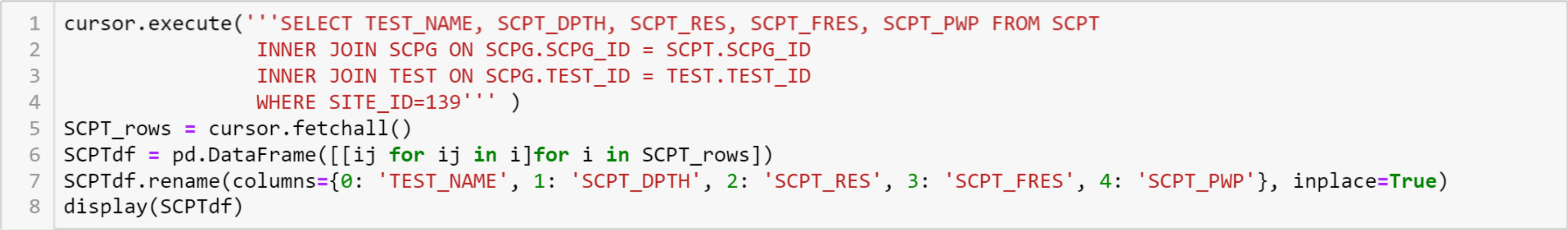
Querying Data with Jupyter Notebook

- The data frame is used to easily manipulate the data in a variety of ways, such as plotting.



Querying Data with Jupyter Notebook

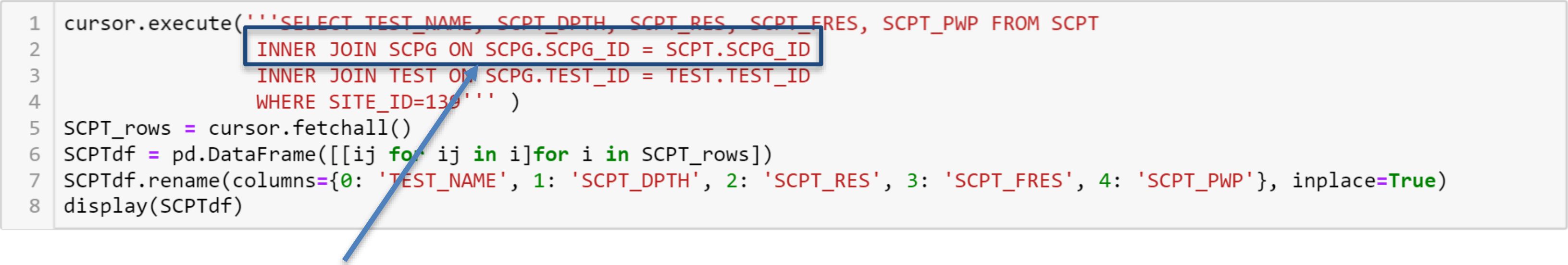
- Foreign keys can be used to join tables



```
cursor.execute('''SELECT TEST_NAME, SCPT_DPTH, SCPT_RES, SCPT_FRES, SCPT_PWP FROM SCPT
                INNER JOIN SCPG ON SCPG.SCPG_ID = SCPT.SCPG_ID
                INNER JOIN TEST ON SCPG.TEST_ID = TEST.TEST_ID
                WHERE SITE_ID=139''' )
SCPT_rows = cursor.fetchall()
SCPTdf = pd.DataFrame([[ij for ij in i]for i in SCPT_rows])
SCPTdf.rename(columns={0: 'TEST_NAME', 1: 'SCPT_DPTH', 2: 'SCPT_RES', 3: 'SCPT_FRES', 4: 'SCPT_PWP'}, inplace=True)
display(SCPTdf)
```

Querying Data with Jupyter Notebook

- Foreign keys can be used to join tables

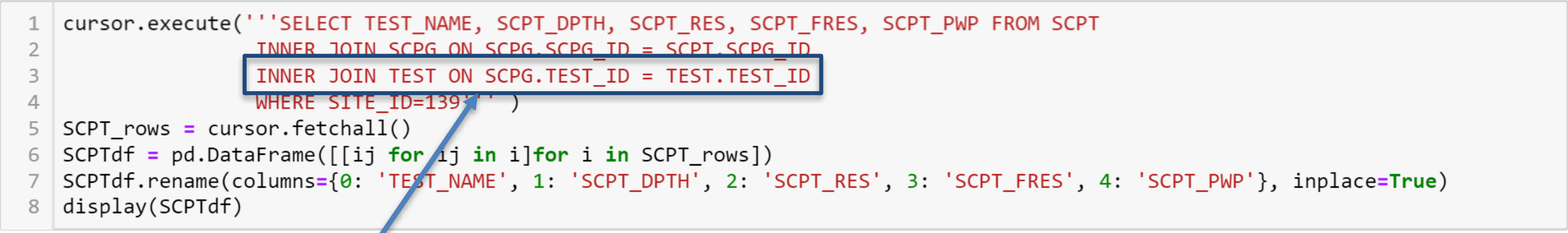


```
cursor.execute('''SELECT TEST_NAME, SCPT_DPTH, SCPT_RES, SCPT_FRES, SCPT_PWP FROM SCPT
                 INNER JOIN SCPG ON SCPG.SCPG_ID = SCPT.SCPG_ID
                 INNER JOIN TEST ON SCPG.TEST_ID = TEST.TEST_ID
                 WHERE SITE_ID=139''' )
SCPT_rows = cursor.fetchall()
SCPTdf = pd.DataFrame([[ij for ij in i]for i in SCPT_rows])
SCPTdf.rename(columns={0: 'TEST_NAME', 1: 'SCPT_DPTH', 2: 'SCPT_RES', 3: 'SCPT_FRES', 4: 'SCPT_PWP'}, inplace=True)
display(SCPTdf)
```

Merges tables
on foreign key

Querying Data with Jupyter Notebook

- Foreign keys can be used to join tables

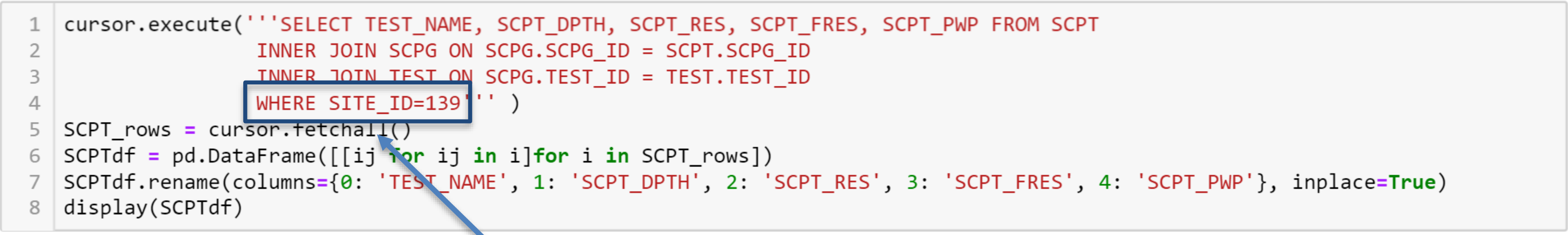


```
cursor.execute('''SELECT TEST_NAME, SCPT_DPTH, SCPT_RES, SCPT_FRES, SCPT_PWP FROM SCPT
                INNER JOIN SCPG ON SCPG.SCPG_ID = SCPT.SCPG_ID
                INNER JOIN TEST ON SCPG.TEST_ID = TEST.TEST_ID
                WHERE SITE_ID=139''')
SCPT_rows = cursor.fetchall()
SCPTdf = pd.DataFrame([[ij for ij in i]for i in SCPT_rows])
SCPTdf.rename(columns={0: 'TEST_NAME', 1: 'SCPT_DPTH', 2: 'SCPT_RES', 3: 'SCPT_FRES', 4: 'SCPT_PWP'}, inplace=True)
display(SCPTdf)
```

Merges tables
on foreign key

Querying Data with Jupyter Notebook

- Foreign keys can be used to join tables

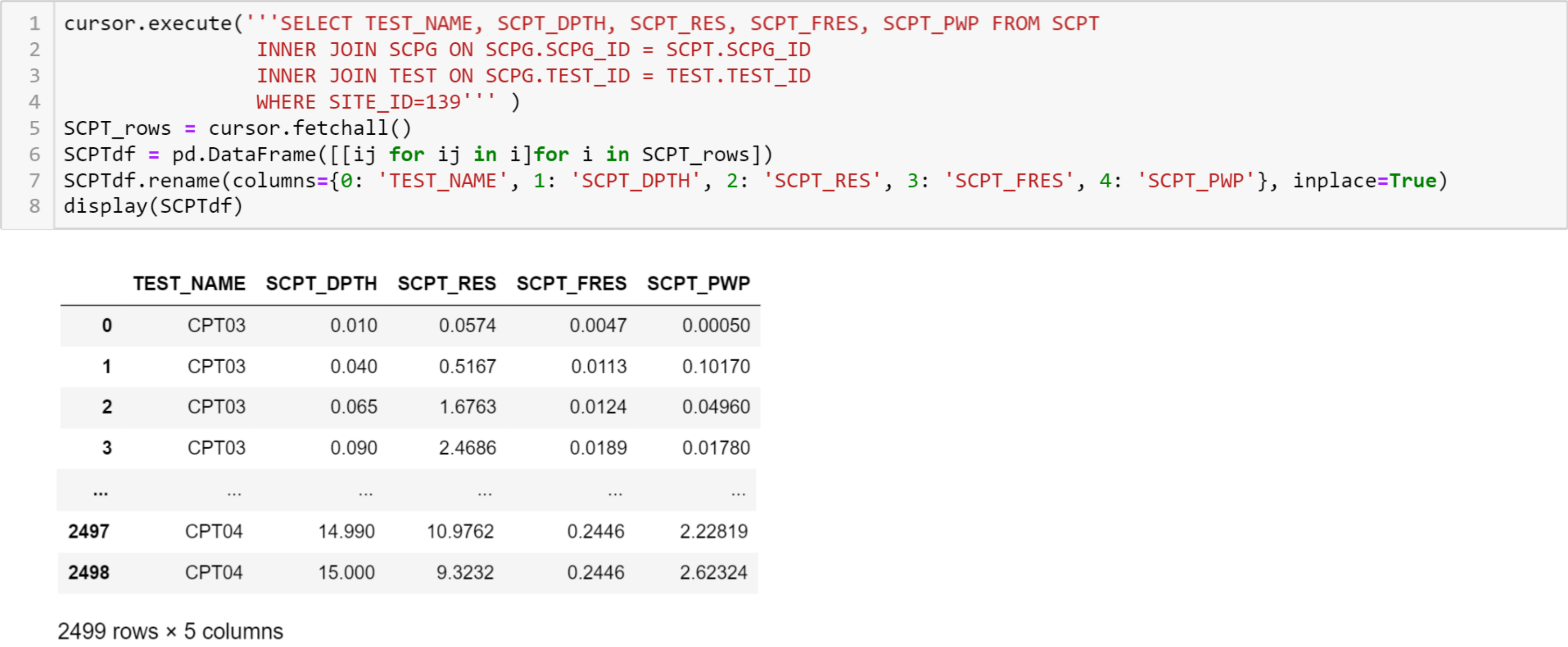


```
cursor.execute('''SELECT TEST_NAME, SCPT_DPTH, SCPT_RES, SCPT_FRES, SCPT_PWP FROM SCPT
                INNER JOIN SCPG ON SCPG.SCPG_ID = SCPT.SCPG_ID
                INNER JOIN TEST ON SCPG.TEST_ID = TEST.TEST_ID
                WHERE SITE_ID=139''' )
SCPT_rows = cursor.fetchall()
SCPTdf = pd.DataFrame([[ij for ij in i]for i in SCPT_rows])
SCPTdf.rename(columns={0: 'TEST_NAME', 1: 'SCPT_DPTH', 2: 'SCPT_RES', 3: 'SCPT_FRES', 4: 'SCPT_PWP'}, inplace=True)
display(SCPTdf)
```

All CPT data
for one site

Querying Data with Jupyter Notebook

- Foreign keys can be used to join tables



```
cursor.execute('''SELECT TEST_NAME, SCPT_DPTH, SCPT_RES, SCPT_FRES, SCPT_PWP FROM SCPT
                INNER JOIN SCPG ON SCPG.SCPG_ID = SCPT.SCPG_ID
                INNER JOIN TEST ON SCPG.TEST_ID = TEST.TEST_ID
                WHERE SITE_ID=139''' )
SCPT_rows = cursor.fetchall()
SCPTdf = pd.DataFrame([[ij for ij in i]for i in SCPT_rows])
SCPTdf.rename(columns={0: 'TEST_NAME', 1: 'SCPT_DPTH', 2: 'SCPT_RES', 3: 'SCPT_FRES', 4: 'SCPT_PWP'}, inplace=True)
display(SCPTdf)
```

	TEST_NAME	SCPT_DPTH	SCPT_RES	SCPT_FRES	SCPT_PWP
0	CPT03	0.010	0.0574	0.0047	0.00050
1	CPT03	0.040	0.5167	0.0113	0.10170
2	CPT03	0.065	1.6763	0.0124	0.04960
3	CPT03	0.090	2.4686	0.0189	0.01780
...	...	...	...	...	...
2497	CPT04	14.990	10.9762	0.2446	2.22819
2498	CPT04	15.000	9.3232	0.2446	2.62324

2499 rows × 5 columns

Creating Post-Processing Tools

- Jupyter notebook allows users to interface directly with the NGL database using MySQL.

- We used this small sample of python modules to create post-processing tools:

Foliu
m

[ipy]widgets

[bq]plot

matplotlib

NumPy

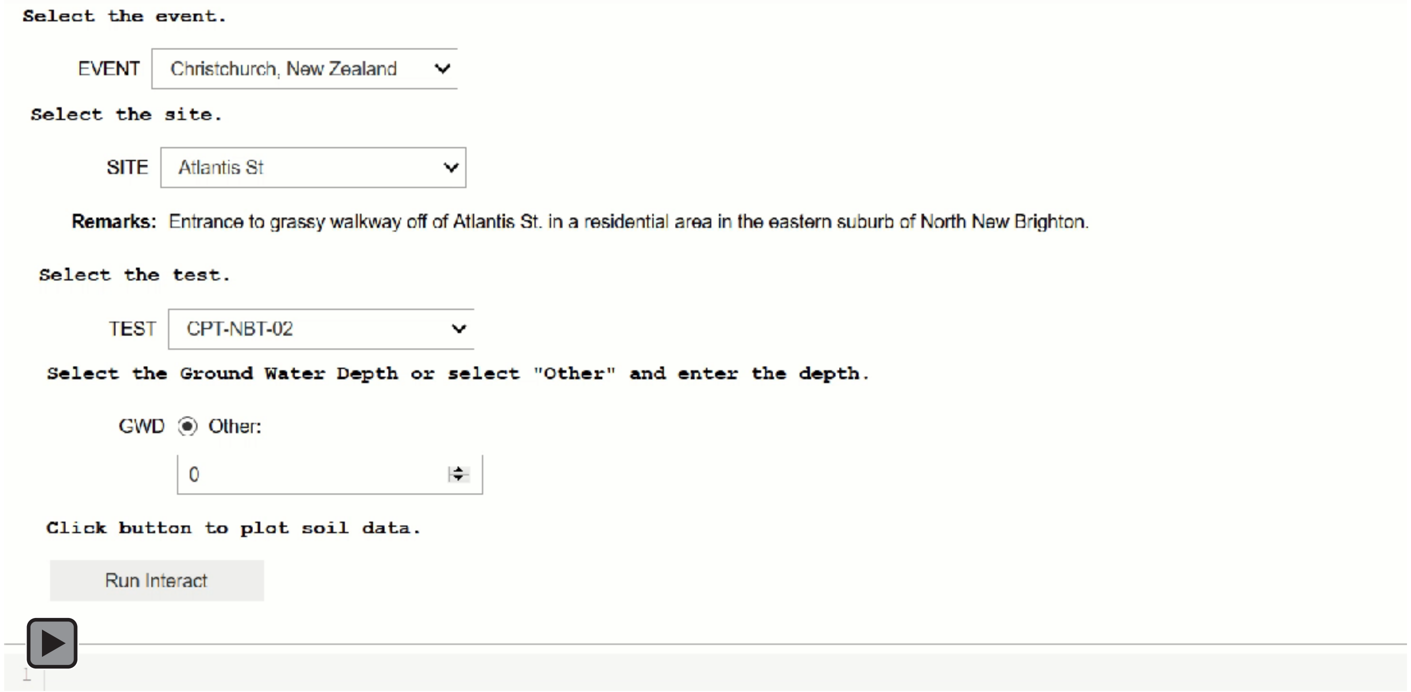
GDAL

pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

Interactive Soil Test Plot

- Allows selection of related event, sites, and tests from the NGL database
- Enables user input of GWD



Select the event.

EVENT Christchurch, New Zealand

Select the site.

SITE Atlantis St

Remarks: Entrance to grassy walkway off of Atlantis St. in a residential area in the eastern suburb of North New Brighton.

Select the test.

TEST CPT-NBT-02

Select the Ground Water Depth or select "Other" and enter the depth.

GWD ● Other:

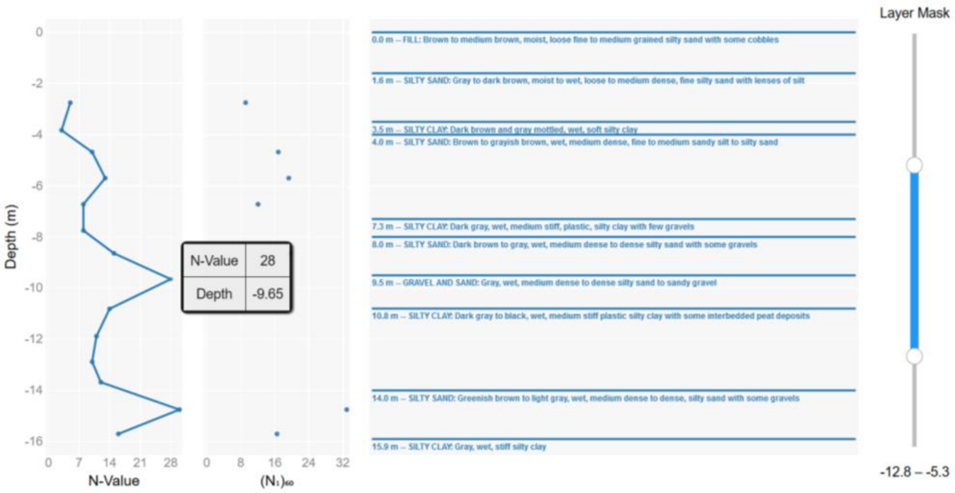
0

Click button to plot soil data.

UCLA Samueli

Interactive Soil Test Plot

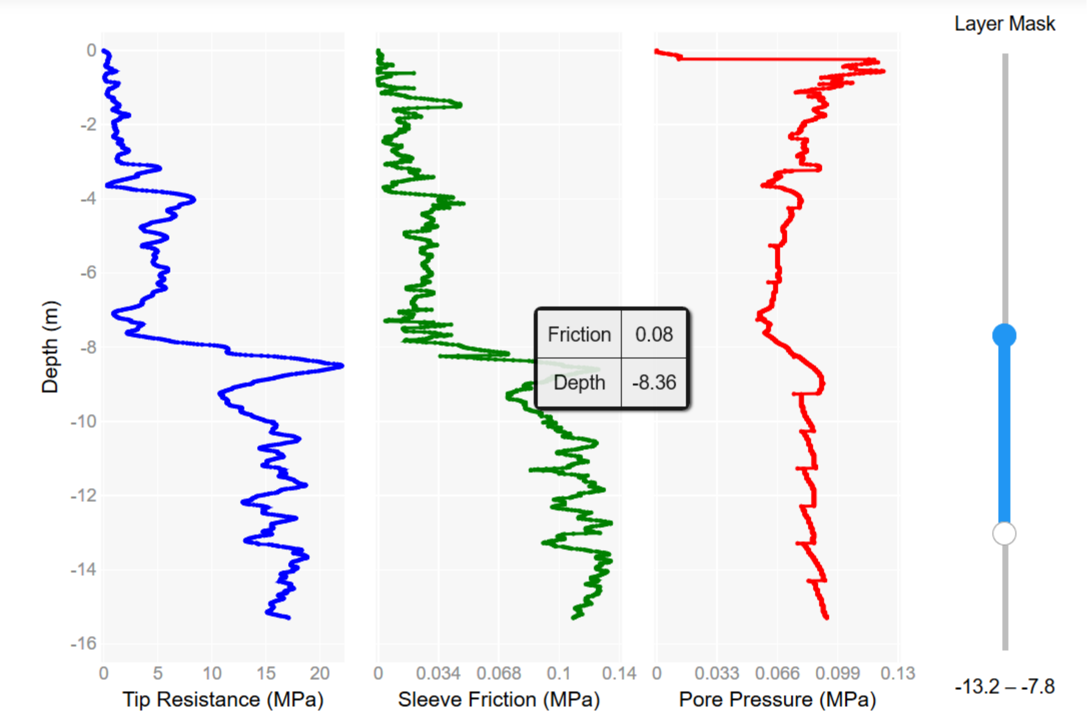
- Plots data and displays stratigraphic layer information
- Allows layer selection with toggle
- Displays coordinates of point when hovered


Layer Mask
Depth (m)
0
-2
-4
-6
-8
-10
-12
-14
-16
0 7 14 21 28
N-Value
0 8 16 24 32
(N₁)₆₀
N-Value 28
Depth -9.65
0.0 m – FILL: Brown to medium brown, moist, loose fine to medium grained silty sand with some cobbles
1.6 m – SILTY SAND: Gray to dark brown, moist to wet, loose to medium dense, fine silty sand with lenses of silt
3.5 m – SILTY CLAY: Dark brown and gray mottled, wet, soft silty clay
4.0 m – SILTY SAND: Brown to grayish brown, wet, medium dense, fine to medium sandy silt to silty sand
7.3 m – SILTY CLAY: Dark gray, wet, medium stiff, plastic, silty clay with few gravels
8.0 m – SILTY SAND: Dark brown to gray, wet, medium dense to dense silty sand with some gravels
9.5 m – GRAVEL AND SAND: Gray, wet, medium dense to dense silty sand to sandy gravel
10.8 m – SILTY CLAY: Dark gray to black, wet, medium stiff plastic silty clay with some interbedded peat deposits
14.0 m – SILTY SAND: Greenish brown to light gray, wet, medium dense to dense, silty sand with some gravels
15.9 m – SILTY CLAY: Gray, wet, stiff silty clay
-12.8 – -5.3

Plots from test *NBS-1* at *Chi-Chi (Taiwan) SITE B*

Interactive Soil Test Plot

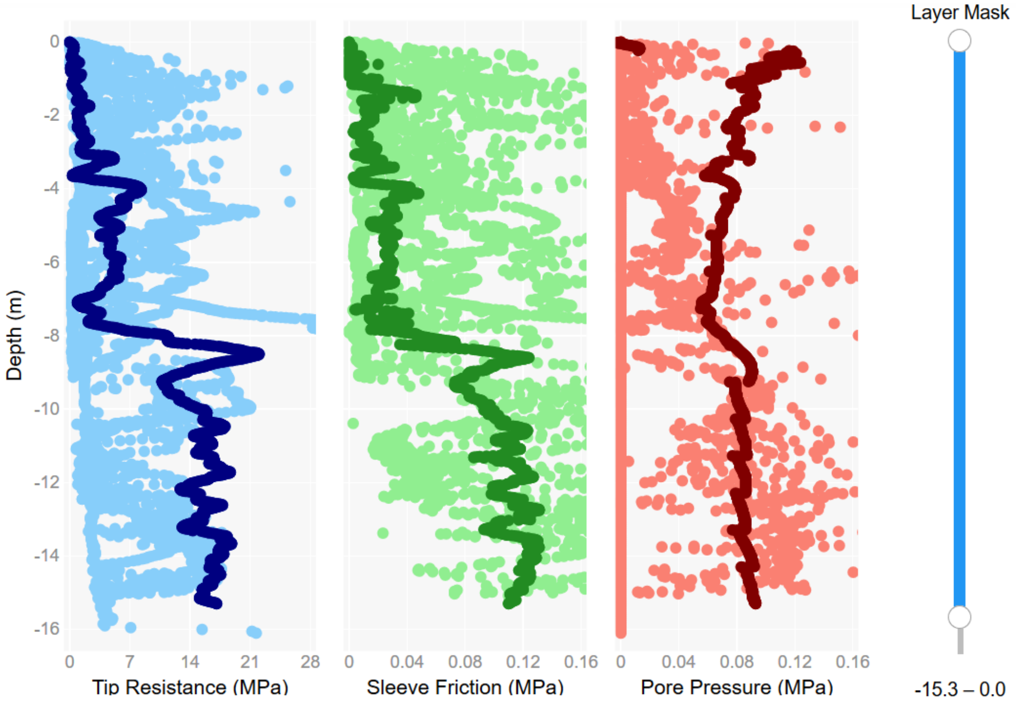
- Provides same features for other test types in database



Plots from test *CPT-NBT-02* at *Atlantis St.*

Interactive Soil Test Plot

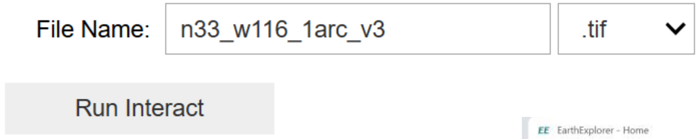
- Provides option to plot current profile over summary of all profiles in the database
- Helps to check for outliers



Plots from test *CPT-NBT-02* at *Atlantis St.*

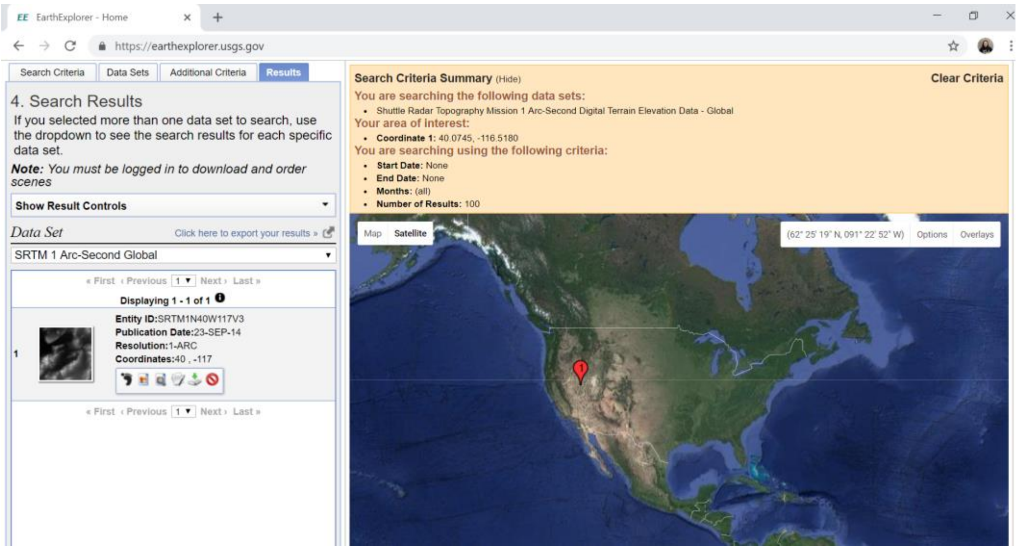
Digital Elevation Model

- Allows user to input a DEM map in the form of a *.tif* file
  - File selected from the user's computer



File Name: n33_w116_1arc_v3

.tif



https://earthexplorer.usgs.gov

Search Criteria
Data Sets
Additional Criteria
Results

4. Search Results

If you selected more than one data set to search, use the dropdown to see the search results for each specific data set.

Note: *You must be logged in to download and order scenes*

Data Set

Click here to export your results »

SRTM 1 Arc-Second Global

« First ‹ Previous 1 Next › Last »

Displaying 1 - 1 of 1

1

Entity ID:SRTM1N40W117V3
Publication Date:23-SEP-14
Resolution:1-ARC
Coordinates:40 , -117

« First ‹ Previous 1 Next › Last »

Search Criteria Summary (Hide)

You are searching the following data sets:

- Shuttle Radar Topography Mission 1 Arc-Second Digital Terrain Elevation Data - Global

Your area of interest:

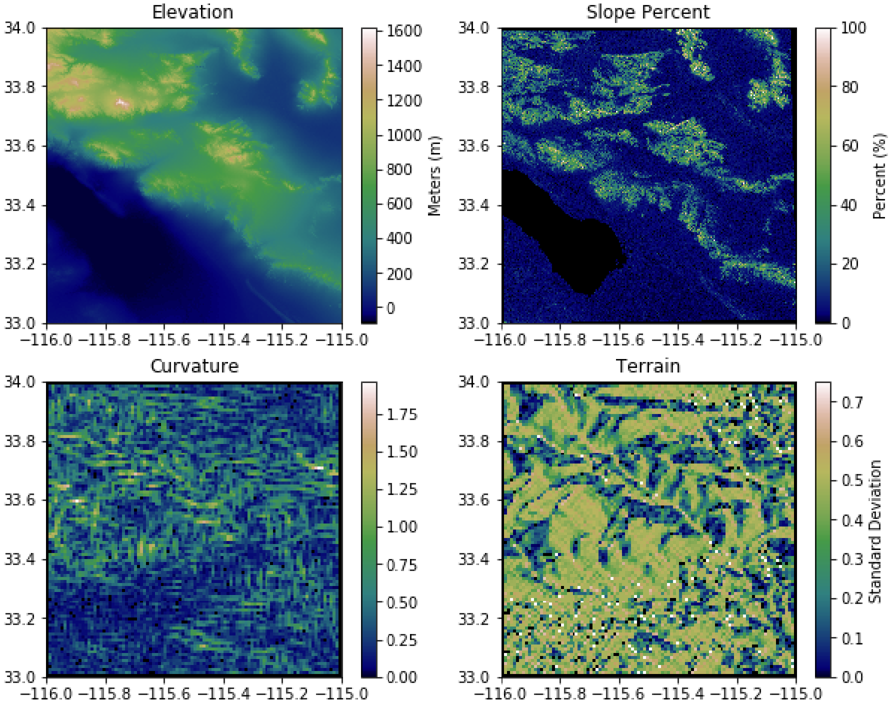
- **Coordinate 1:** 40.0745, -116.5180

You are searching using the following criteria:

- **Start Date:** None
- **End Date:** None
- **Months:** (all)
- **Number of Results:** 100

Digital Elevation Model

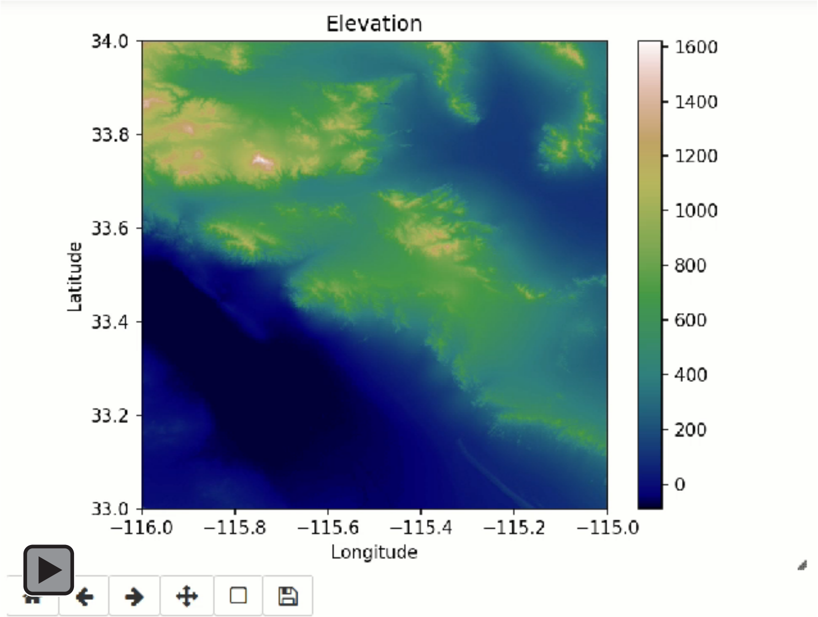
- Outputs colored maps of elevation, slope, curvature, and terrain



Elevation and slope plots from *.tif* file downloaded from USGS Earth Explorer at SRTM 1 Arc-Second Global in the Imperial Valley region of California

Digital Elevation Model

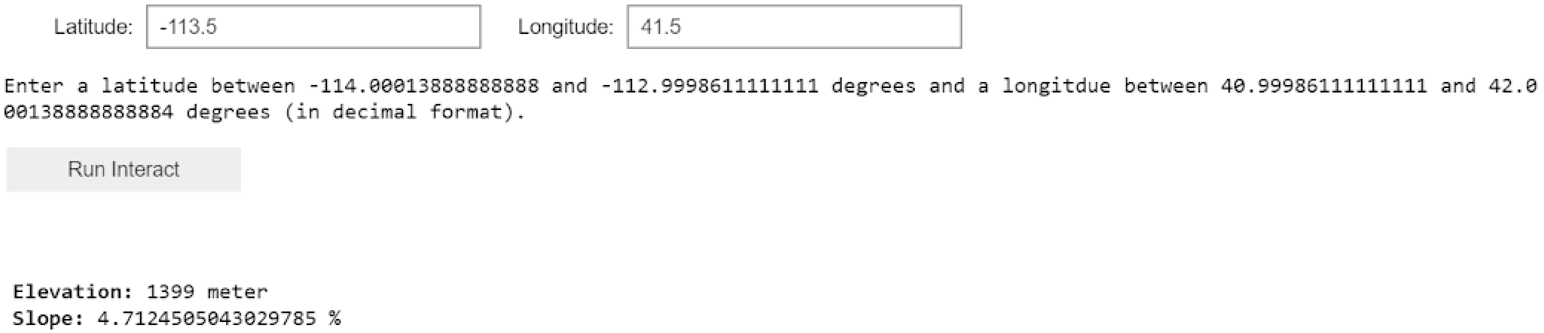
- Allows interaction with color map



Interactive elevation and slope plots from *.tif* file downloaded from USGS Earth Explorer at SRTM 1 Arc-Second Global covering the Imperial Valley region of California

Digital Elevation Model

- Allow user to input a set of latitude and longitude to see specific elevation and slope values



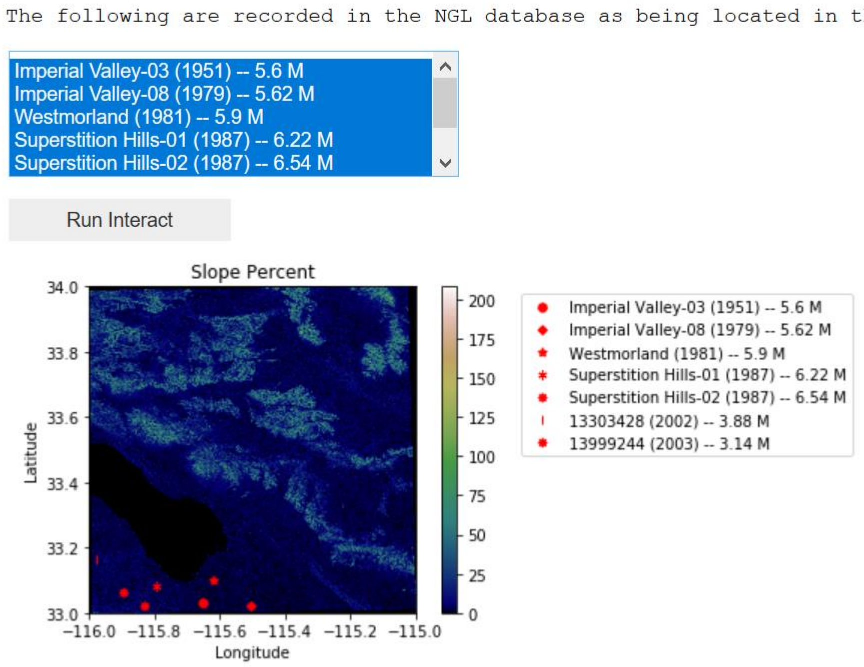
Latitude: -113.5 Longitude: 41.5

```
Enter a latitude between -114.00013888888888 and -112.9998611111111 degrees and a longitdue between 40.99986111111111 and 42.0
00138888888884 degrees (in decimal format).
```

```
Elevation: 1399 meter
Slope: 4.7124505043029785 %
```

Digital Elevation Models

- Offers selection of events, sites, and tests from the NGL database that fall within range of *.tif* latitude and longitude
- Plots selected locations on slope raster



```
The following are recorded in the NGL database as being located in the .tif
```

Imperial Valley-03 (1951) -- 5.6 M
Imperial Valley-08 (1979) -- 5.62 M
Westmorland (1981) -- 5.9 M
Superstition Hills-01 (1987) -- 6.22 M
Superstition Hills-02 (1987) -- 6.54 M

Run Interact
Slope Percent
34.0
33.8
33.6
33.4
33.2
33.0
Latitude
−116.0
−115.8
−115.6
−115.4
−115.2
−115.0
Longitude
200
175
150
125
100
75
50
25
0
● Imperial Valley-03 (1951) -- 5.6 M
◆ Imperial Valley-08 (1979) -- 5.62 M
★ Westmorland (1981) -- 5.9 M
✱ Superstition Hills-01 (1987) -- 6.22 M
● Superstition Hills-02 (1987) -- 6.54 M
| 13303428 (2002) -- 3.88 M
● 13999244 (2003) -- 3.14 M

Slope plot from *.tif* file downloaded from USGS Earth Explorer at
SRTM 1 Arc-Second Global containing events from the Imperial Valley

Conclusion

- Analyzing data is simpler and more efficient with the NGL database and Jupyter Notebook
- Interacting with data is feasible by using tools we made or by making your own!

Thank you!

Questions?

Relevant References

Next-Generation Liquefaction Database. http://uclageo.com/NGL/database/index.php

Earth Explorer (USGS). https://earthexplorer.usgs.gov

NGL NEXT GENERATION LIQUEFACTION

Project homepage:
https://uclageo.com/NGL/
Database:
www.nextgenerationliquefaction.org

UCLA Samueli School of Engineering