

Workshop on the Next-Generation Liquefaction Database

Mong Learning Center, UCLA - September 24, 2018



Next Generation Liquefaction (NGL) Database Interactive Tools

Honor Fisher & Allison Lee
Undergraduate Scientists



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

```
1 import mysql.connector
2 cnx = mysql.connector.connect(user='root', password='root', database='ng1')
3 cursor = cnx.cursor()
```

Querying Data with Jupyter Notebook

- Executes the query

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

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

Querying Data with Jupyter Notebook

- Executes the query

```
1 import mysql.connector
2 cnx = mysql.connector.connect(user='root', password='root', database='ng1')
3 cursor = cnx.cursor()

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

Gets CPT data
from database

Querying Data with Jupyter Notebook

- Executes the query

```
1 import mysql.connector
2 cnx = mysql.connector.connect(user='root', password='root', database='ng1')
3 cursor = cnx.cursor()
```

```
1 cursor.execute("SELECT SCPT_DPTH, SCPT_RES, SCPT_FRES, SCPT_PWP FROM SCPT WHERE SCPG_ID=77")
2 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

```
1 import mysql.connector
2 cnx = mysql.connector.connect(user='root', password='root', database='ng1')
3 cursor = cnx.cursor()
```

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

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

Querying Data with Jupyter Notebook

- Converts the data to a pandas data frame

```
1 import mysql.connector
2 cnx = mysql.connector.connect(user='root', password='root', database='ng1')
3 cursor = cnx.cursor()
```

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

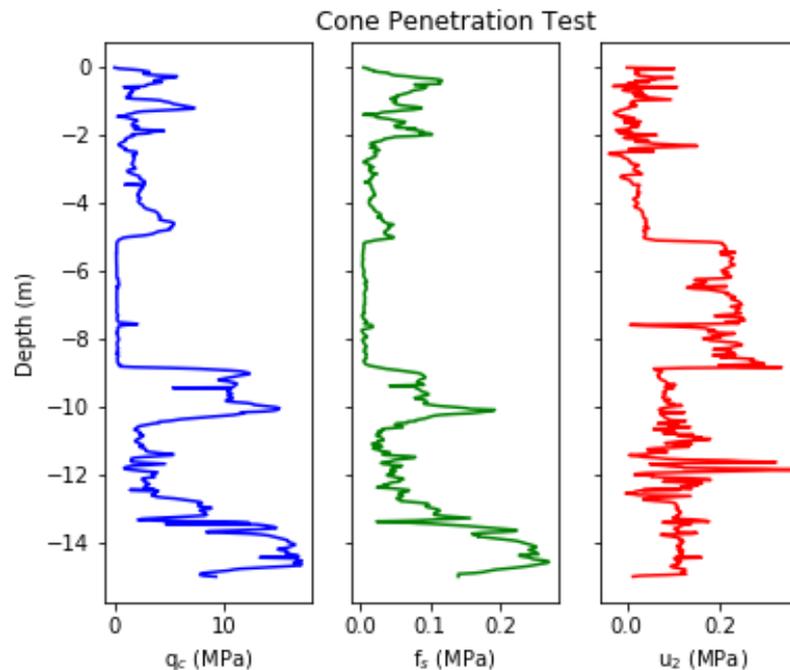
```
1 import pandas
2 SCPTdf = pandas.DataFrame([[ij for ij in i] for i in rows])
3 SCPTdf.rename(columns={0: 'SCPT_DPTH', 1: 'SCPT_RES', 2: 'SCPT_FRES', 3: 'SCPT_PWP'}, inplace=True)
4 display(SCPTdf)
```

	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

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

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

Querying Data with Jupyter Notebook

- Foreign keys can be used to join tables

```
1 cursor.execute('''SELECT TEST_NAME, SCPT_DPTH, SCPT_RES, SCPT_FRES, SCPT_PWP FROM SCPT
2         INNER JOIN SCPG ON SCPG.SCPG_ID = SCPT.SCPG_ID
3         INNER JOIN TEST ON SCPG.TEST_ID = TEST.TEST_ID
4         WHERE SITE_ID=130''')
5 SCPT_rows = cursor.fetchall()
6 SCPTdf = pd.DataFrame([[ij for ij in i]for i in SCPT_rows])
7 SCPTdf.rename(columns={0: 'TEST_NAME', 1: 'SCPT_DPTH', 2: 'SCPT_RES', 3: 'SCPT_FRES', 4: 'SCPT_PWP'}, inplace=True)
8 display(SCPTdf)
```

Merges tables
on foreign key

Querying Data with Jupyter Notebook

- Foreign keys can be used to join tables

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

Merges tables
on foreign key

Querying Data with Jupyter Notebook

- Foreign keys can be used to join tables

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

All CPT data
for one site

Querying Data with Jupyter Notebook

- Foreign keys can be used to join tables

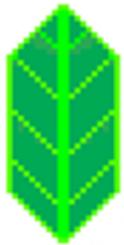
```
1 cursor.execute('''SELECT TEST_NAME, SCPT_DPTH, SCPT_RES, SCPT_FRES, SCPT_PWP FROM SCPT
2                 INNER JOIN SCPG ON SCPG.SCPG_ID = SCPT.SCPG_ID
3                 INNER JOIN TEST ON SCPG.TEST_ID = TEST.TEST_ID
4                 WHERE SITE_ID=139''')
5 SCPT_rows = cursor.fetchall()
6 SCPTdf = pd.DataFrame([[ij for ij in i]for i in SCPT_rows])
7 SCPTdf.rename(columns={0: 'TEST_NAME', 1: 'SCPT_DPTH', 2: 'SCPT_RES', 3: 'SCPT_FRES', 4: 'SCPT_PWP'}, inplace=True)
8 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:

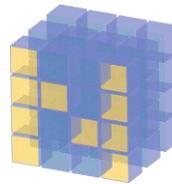


Folium

[ipy]widgets

[bq]plot

matplotlib

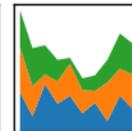
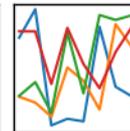


NumPy



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

Select the site.

SITE

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

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

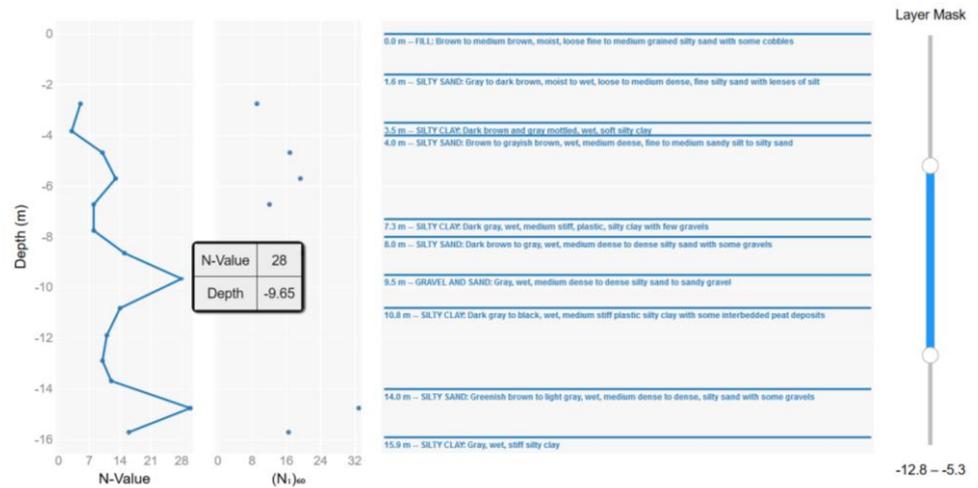
GWD Other:

Click button to plot soil data.

1

Interactive Soil Test Plot

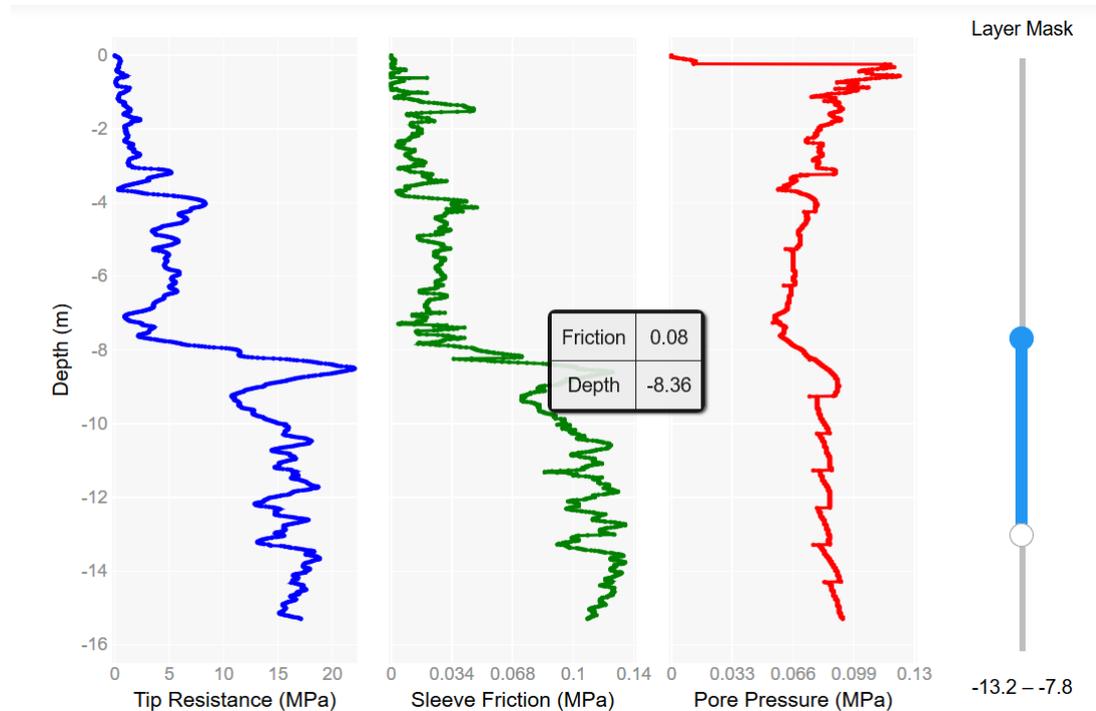
- Plots data and displays stratigraphic layer information
- Allows layer selection with toggle
- Displays coordinates of point when hovered



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

Interactive Soil Test Plot

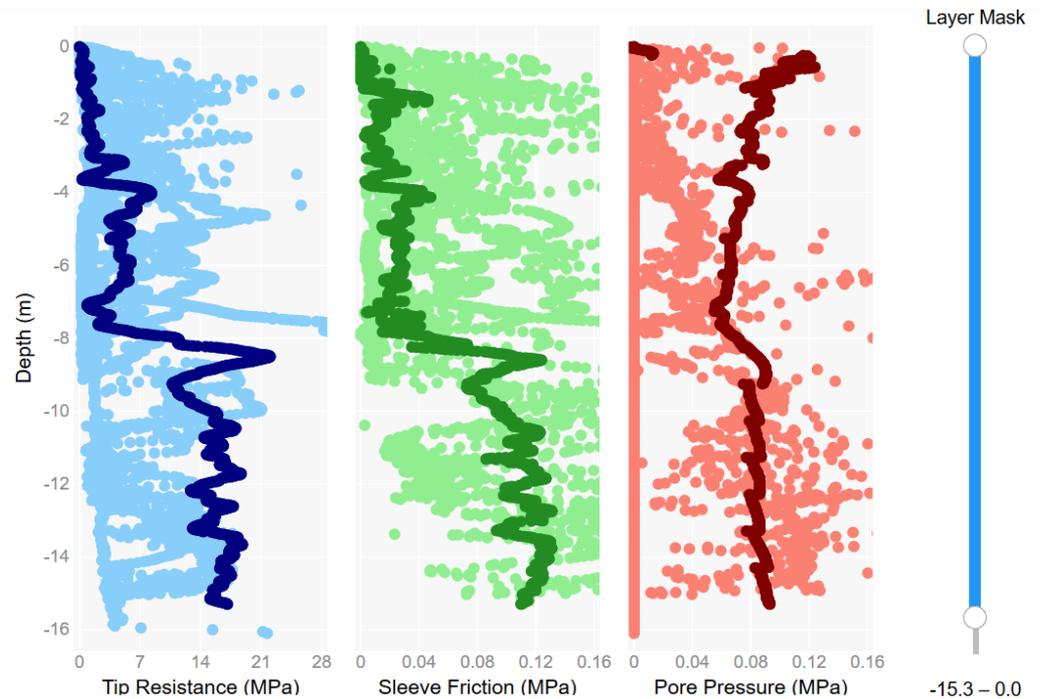
- Provides same features for other test types in database



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

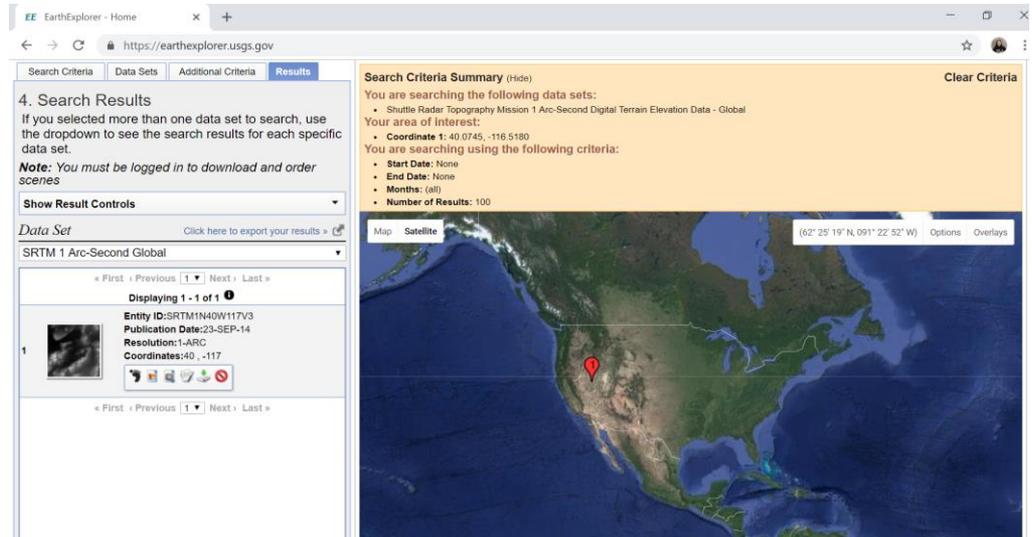


Plots from test *CPT-NBT-02* at *Atlantis St.*

Digital Elevation Model

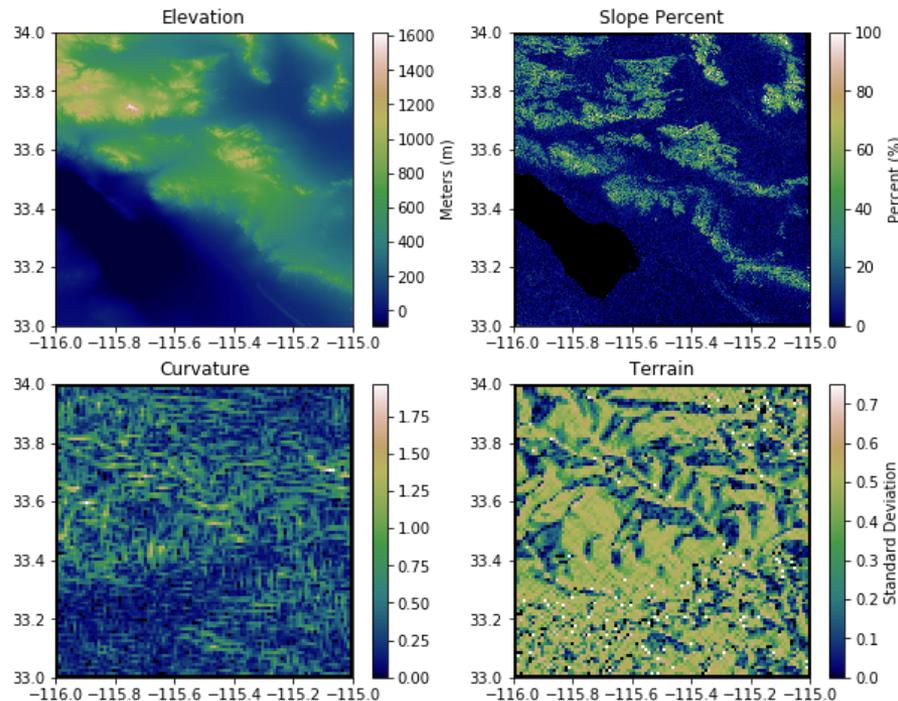
- Allows user to input a DEM map in the form of a *.tif* file
 - File selected from the user's computer

File Name:



Digital Elevation Model

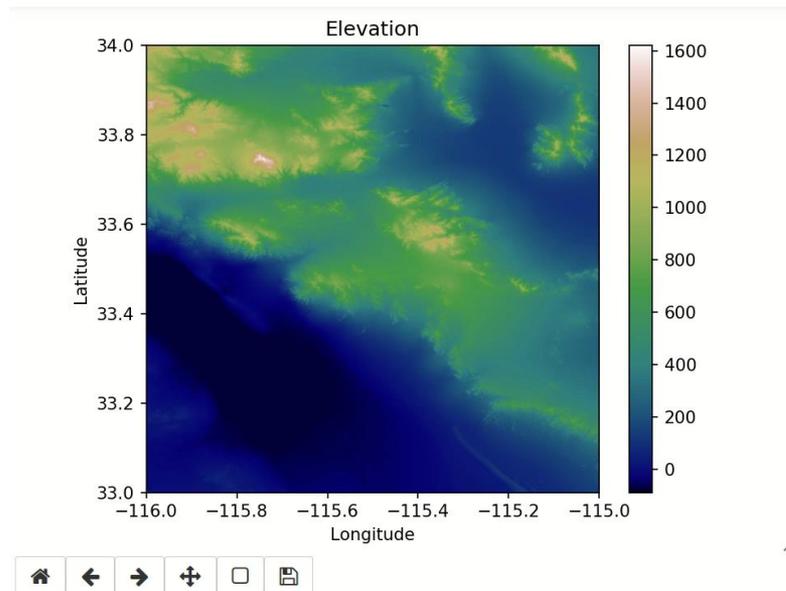
- Outputs colored maps of elevation, slope, curvature, and terrain



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:

Longitude:

Enter a latitude between -114.00013888888888 and -112.99986111111111 degrees and a longitude between 40.99986111111111 and 42.000138888888884 degrees (in decimal format).

Run Interact

Elevation: 1399 meter

Slope: 4.7124505043029785 %

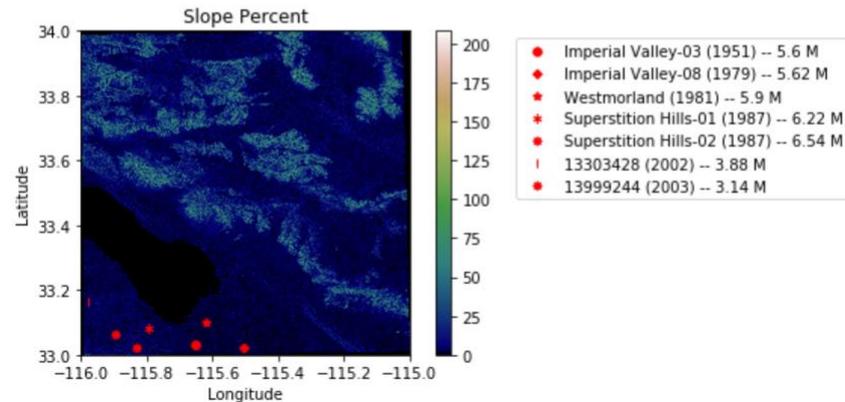
Digital Elevation Models

- 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 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>



Project homepage:

<https://uclageo.com/NGL/>

Database:

www.nextgenerationliquefaction.org

