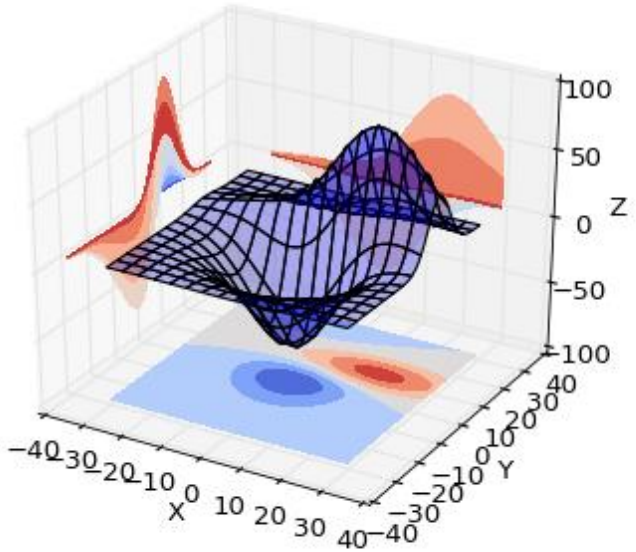
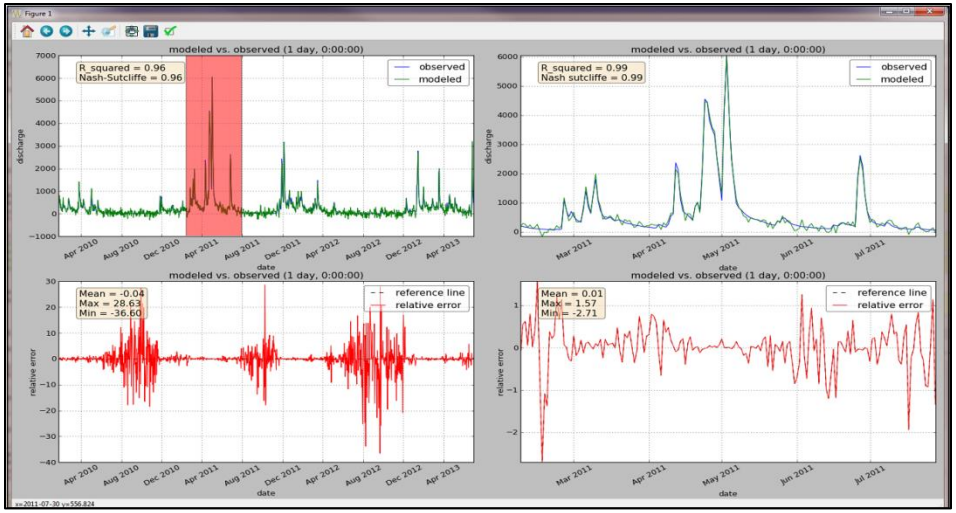
Introduction to Scientific Computing
Meeting 27
Programming with Python



python™

```
# Write Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610
```



Jeremiah Lant, Hydrologist
USGS Kentucky Water Science Center
jlant@usgs.gov

Last Meeting

- Showed Spyder IDE (Scientific Python Development Environment).
- Mentioned reorganization of GoogleDrive material and making a GitHub repo for the material.
- Talked about possibility of using GoogleDocs or Etherpad (http://etherpad.org/) to keep a record/log of meeting notes and questions.
- Continued to write a python script to read a sample data file using sys.argv and display simple statistics to the screen.

Today's Objective

- Continue to work on python script to read a sample data file using sys.argv and display simple statistics to the screen.
- Show how to develop a basic test for the code using an external test file.

Let’s continue coding!

date	discharge (cfs)	stage (ft)	temperature (celsius)
01/05/2014	100	12.2	5
02/08/2014	110	12.8	3
03/07/2014	105	12.5	10
04/01/2014	98	11.9	20
05/04/2014	92	11.5	25
06/01/2014	104	12.3	28
07/02/2014	97	11.8	32
08/03/2014	95	11.7	33
09/04/2014	96	11.7	27
10/05/2014	101	12.0	20
11/02/2014	112	13.2	15
12/03/2014	109	12.8	7

```
$ python read_measurements.py 2014_measurements_bob.txt
2014_measurements_bob.txt

  discharge (cfs):
     Average: 101.583
     Maximum: 112.0 occurred on 11/02/2014
     Minimum: 92.0 occurred on 05/04/2014

  stage (ft):
     Average: 12.200
     Maximum: 13.2 occurred on 11/02/2014
     Minimum: 11.5 occurred on 05/04/2014

  temperature (celsius):
     Average: 18.750
     Maximum: 33.0 occurred on 08/03/2014
     Minimum: 3.0 occurred on 02/08/2014
```

Basic test using external file

- Why?
  - Helps ensure that our code is doing what we think it is doing.
- First step in testing your software.
  - Will learn more about automated testing using a library called **nose**.