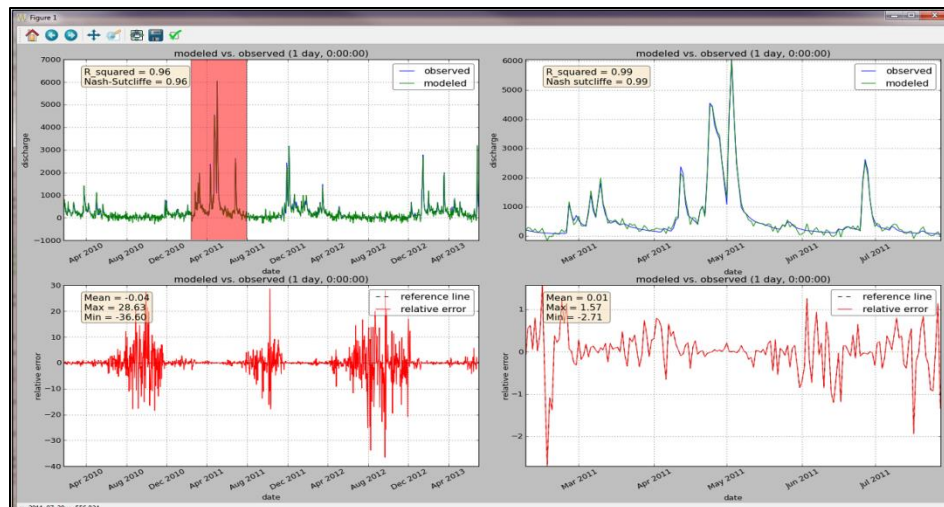
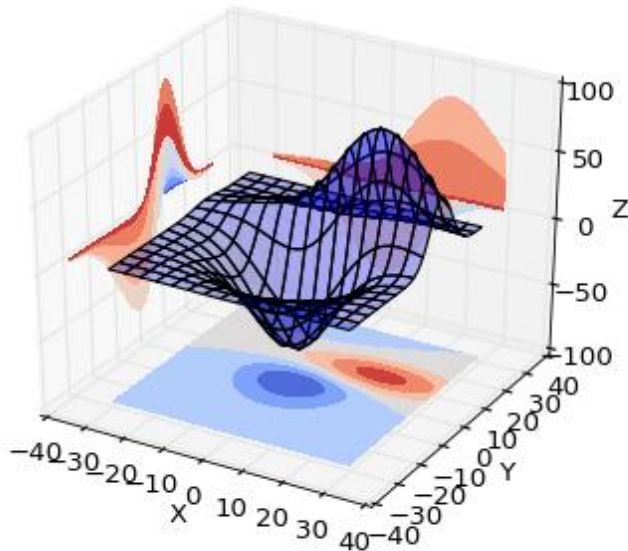


Scientific Computing Group

Programming with Python:

Functions, Read Measurements Project



```
# 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

- Completed a python script that reads a sample data file using `sys.argv` and displays simple statistics to the screen.
- Improved the `read_measurements.py` script by adding the ability to process multiple data files.
- Introduced functions

Today's Objective

- Write functions and show the various ways to write the parameters
- Start to implement functions in `read_measurements.py` script

Review – Functions

- Functions are used to help to modularize your code.

Keyword **def**
defines start
of a function

Function **arguments/parameters**

Colon to signify
block of
operations will
follow

```
def func_name(param1, param2, ...):  
    """ Document the function """  
    statements  
    return return_values
```

Indent 4
spaces

Optional **return** specifies value
to be returned

Some common built-in Python Functions

```
>>> abs(-5)
```

```
>>> max([1, 2, 3])
```

```
>>> min([1, 2, 3])
```

```
>>> print("Hello")
```

<code>abs(a)</code>	Absolute value of <code>a</code>
<code>max(sequence)</code>	Largest element of <i>sequence</i>
<code>min(sequence)</code>	Smallest element of <i>sequence</i>
<code>round(a,n)</code>	Round <code>a</code> to <code>n</code> decimal places
<code>cmp(a,b)</code>	Returns $\begin{cases} -1 & \text{if } a < b \\ 0 & \text{if } a = b \\ 1 & \text{if } a > b \end{cases}$

Video – Python



- Software Carpentry, Greg Wilson
 - Python: Functions
 - <http://software-carpentry.org/v4/python/func.html>

Practice – writing functions

- Write a function called **subtract()** that takes 2 numbers and adds them together and returns the result.
- Write a function called **greet()** that takes no arguments and prints a friendly greeting to the screen.
- Rewrite the function **greet()** so that it takes a persons name as a parameter and prints a friendly greeting which includes the persons name.
- Write a function called **average()** that takes a list of numbers and returns the average.

Next meeting

- Write functions for the `read_measurements.py` script.