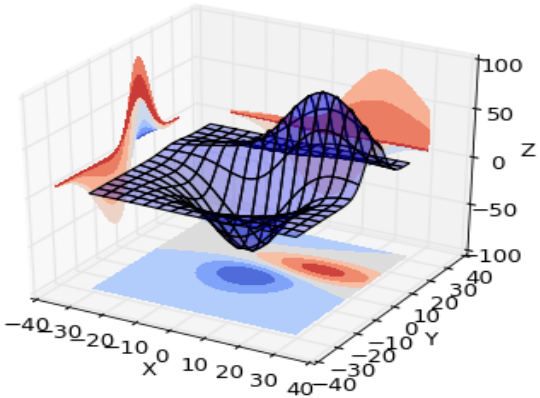
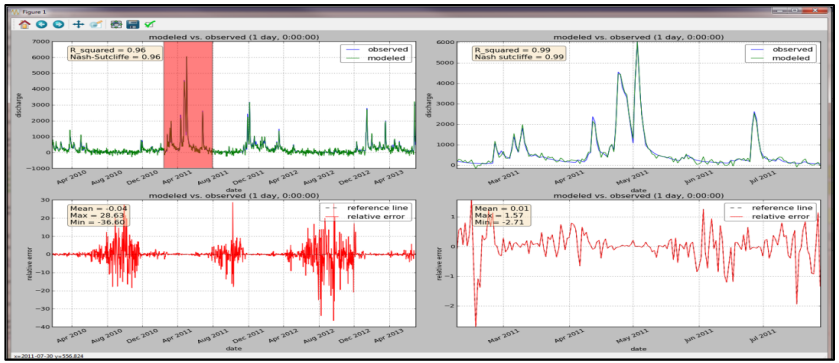
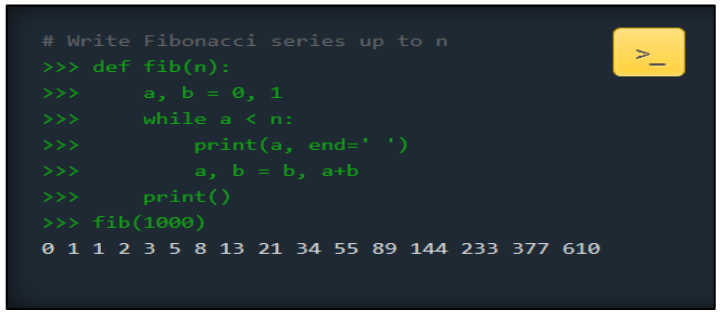
Scientific Computing Group
Programming with Python:
Read Measurements Project, Refactoring







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

Review of notes and questions from last meeting

- https://github.com/jlant-usgs/scientific-computing-group

Today’s Objectives

1. Refactor the read_measurements.py file with functions.

Convert script to a module like structure

read_measurements.py script

```
# import useful modules
import sys

# get user arguments (measurement files) using sys.argv
python_filename = sys.argv[0]
all_files = sys.argv[1:]

for measurements_file_str in all_files:

    # read the file and its contents
    file_obj = open(measurements_file_str, "r")
    file_lines = file_obj.readlines()

    # extract the column names and put into a list
    column_names = file_lines.pop(0)
    column_names = column_names.rstrip().split("\t")

    # get the data organized into lists - loop through the file_lines
    dates = []
    discharge = []
    stage = []
    temperature = []
    for line in file_lines:
        line_list = line.rstrip().split("\t")
        dates.append(line_list[0])
        discharge.append(float(line_list[1]))
        stage.append(float(line_list[2]))
        temperature.append(float(line_list[3]))

    # write a for loop to loop through each parameter and compute stats and print output to screen
    print("{}".format(measurements_file_str))
    counter = 1
    for parameter in [discharge, stage, temperature]:

        # compute some simple statistics
        avg_param = sum(parameter) / len(parameter)
        max_param = max(parameter)
        min_param = min(parameter)

        # find the dates for the max and min values
        max_param_index = parameter.index(max_param)
        min_param_index = parameter.index(min_param)

        max_param_date = dates[max_param_index]
        min_param_date = dates[min_param_index]

        # print in a nice format
        print("")
        print("\t{}".format(column_names[counter]))
        print("\t\t Average: {0}".format(avg_param))
        print("\t\t Maximum: {0} occurred on {1}".format(max_param, max_param_date))
```

read_measurements.py module

```
import sys

def read_file(file_str):
    """ Return file lines for a file """

    file_obj = open(file_str, "r")
    file_lines = file_obj.readlines()

    return file_lines

def get_data_parameters(file_lines):
    """ Return lists for each data parameter (column); date, discharge, stage, temperature"""
    dates = []
    discharge = []
    stage = []
    temperature = []
    for line in file_lines:
        line_list = line.rstrip().split("\t")
        dates.append(line_list[0])
        discharge.append(float(line_list[1]))
        stage.append(float(line_list[2]))
        temperature.append(float(line_list[3]))

    return dates, discharge, stage, temperature

def compute_stats(parameter_list):
    """ Return average, max, min for a data parameter list"""

    avg_param = sum(parameter_list) / len(parameter_list)
    max_param = max(parameter_list)
    min_param = min(parameter_list)

    return avg_param, max_param, min_param

def extract_column_names(file_lines):
    """ Return a list of column names extracted from a list of file lines. Mutates the file lines list by extracting first element (first row of data file) """
    column_names = file_lines.pop(0)
    column_names = column_names.rstrip().split("\t")

    return column_names

def find_date(date_list, parameter_list, value):
    """ Return the string date that corresponds to a particular value in a parameter list"""
    value_index = parameter_list.index(value)       # find where the value occurs in the parameter list
    date = date_list[value_index]                   # get the date that corresponds to the value index

    return date

def print_results(column_names_list, loop_counter, avg_value, max_value, min_value, max_date, min_date):
    """ Print the results in a nice format """
    print("")
```

Next Meeting

- Refactor read_measurements.py with numpy arrays
- Plot data with matplotlib