Geog0111 Scientific Computing

Coursework (Assessed Practical) Part A
Instructions and marking grids

M. Mokros
(m.mokros@ucl.ac.uk)

P. Lewis

2nd October 2023

URL: https://github.com/UCL-EO/geog0111

1. Introduction

1.1 Task overview

The coursework for Geog0111 Scientific Computing consists of two parts (Part A and Part B). The course is assessed entirely using these two submissions. This document describes the requirements for Part A (50% of the total marks), which is due for submission on the first Monday after Reading Week in Term 1. Part A covers data preparation and presentation, and part B covers environmental modelling (using these and other geospatial data).

In this task, you will be writing codes that download, interpret (convert units), and plot two datasets associated with the Del Norte catchment in Colorado, USA for the years 2016-2019 inclusive (i.e. 4 years of data).

The main coding exercise is a set of Python functions that you should develop in a Jupyter notebook and run (in a notebook cell). The code should download the data for a given year (from a URL) and store them in CSV format files (for each year). You need to demonstrate running the codes, illustrate the resultant datasets and produce appropriate plots.

1.2 Submission

The due dates for the two formally assessed pieces of coursework are:

- Part A (this piece of work):   13 Nov, 2023 (50% of final mark) - first Monday after reading week.
- Part B (the next piece of work): 8 Jan, 2024 (50% of final mark) - first day of term 2

Submission is through the usual Turnitin link on the course Moodle page.

You must develop and run the codes in a single Jupyter notebook, and submit the work in a single notebook as a PDF file.

2 Background

2.1 Model

The hydrology of the Rio Grande Headwaters in Colorado, USA is snowmelt dominated. It varies considerably from year to year and may very further under a changing climate. One of the tools we use to understand monitor processes in such an area is a mathematical ('environmental') model describing the main physical processes affecting hydrology in the catchment. Such a model could help understand current behaviour and allow some prediction about possible future scenarios.

In part B of your assessment you will be using, calibrating and validating such a model that relates temperature and snow cover in the catchment to river flow.

We will use the model to describe the streamflow at the Del Norte measurement station, just on the edge of the catchment. You will use environmental (temperature) data and snow cover observations to drive the model. You will perform calibration and testing by comparing model output with observed streamflow data.

2.2.Del Norte

Further general information is available from various websites, including NOAA. You can visualise the site Del Norte 2E here. This is the site we will be using for river discharge data.

3. Coursework Background

3.1 Basic requirements for this submission
In Part A of the assessment (this part), you will need to write computer codes to read, interpret and plot the various environmental datasets we will be using in Part B.

To complete this practical you will need to develop and run a set of Python functions to access, read, save and plot the following datasets:

- Stream discharge from the USGS https://waterservices.usgs.gov.
- Temperature data from a URL on the coursenotes giuthub site.

The temperature dataset is derived from a dataset for Del Norte 2E from the Colorado State Climate data site.

3.1 Access
You should examine the data on the site links and make sure you understand the file format and data characteristics.

The data links and formats are:

Stream discharge from the USGS

https://waterservices.usgs.gov/nwis/dv/?sites=08220000&format=rdb&startDT=2000-01-01&endDT=2000-12-31&parameterCd=00060 for the year 2000.

Use the startDT and endDT fields in this URL to specify other years of data.

The first 30 lines of data (those starting # and the 2 lines after) are comment (header) lines.
The 3rd column of the data fields gives the date, the 4th column the streamflow (discharge) at the site (units of cubic feet per second), and the 5th column a qualification code (see file header for further details).

Temperature data from Colorado State Climate data site

The climate data for 2000-2022 are derived from the Colorado State Climate data site and accessible via https://raw.githubusercontent.com/UCL-EO/geog0111/master/notebooks/data/delNorteT.dat

The first line of the file gives a header. The first column of data is the date field, then maximum temperature (Fahrenheit), minimum temperature (Fahrenheit), and snowfall (units unspecified). Note that missing data are specified with M or T.

3.2 Units
You need to make a note of the *units* that the physical parameters (the temperature and discharge datasets) are presented in, as you will need to convert them to the units specified for the task.

3.3 Header
Make a note of the number of 'header' lines in the files (if any) and the column headings (if given) as you may need to specify these when reading the dataset.

3.4 Getting started with data access

Before going further, you should use a Jupyter notebook to make sure you can access and interpret these files using the methods you have been taught (or other appropriate methods) for accessing ASCII and interpreting datasets from a URL. There are only two files, so this should not take too long, and you need to make sure you know how to do this before you attempt to do the coursework.

You will most likely want to use pandas to do this, and should have gained experience in this already in the course. You may need to try out several options sin reading the data into pandas if it does not automatically load correctly into pandas from a URL. In that case, remember that you can skip header lines, and get pandas to read only the data lines. You should be able to print the pandas dataframe to see that the data in your table corresponds to what you can see when you access data with a URL.

4. Coursework Instructions

4.1 Introduction
For this task, you must write a series of *documented* and *commented* Python functions within cells in a Jupyter notebook, show the documentation (`help()`), run the functions for multiple years, and demonstrate the results.

The datasets you must access are <u>daily</u> datasets of: **stream discharge** from the USGS and **temperature data** from Colorado State Climate data site that can be accessed as described above.

4.2 Tasks
The three required elements of the notebook are given below. You must *label* them as separate sections in the report. Percentage marks for each is given in brackets.

4.2.1 Write a function to access and save the data [50%]

You must write a function with docstring and appropriate comments that:
- takes the `year` (specified as integer) as argument
- accesses the **stream discharge** from the USGS and **temperature data** from Colorado State Climate data site described above for *each day* in that year
- calculates:
  - the day of year
  - the *mean* temperature in Celcius and
  - stream discharge in ML/day (i.e. units of 1000000 litres a day) for each day
- saves these 3 values for each day in the year as a dataset in a 3 column CSV-format file called `work/delNorteYYYY.csv`, where `YYYY` is the year, with appropriate headers for columns of day of year, temperature, and stream discharge
- returns a dictionary (or pandas dataframe) containing the dataset.

You must show the docstring for the function, by running `help()` for your function.

4.2.2 Demonstrate running the function to access and save the data [20%]

You must write code in a Jupyter notebook cell to:
- run the data access function defined above for the years 2016-2019 inclusive (i.e. 4 years of data)
- for each of these, demonstrate that the dictionary (or pandas dataframe) returned from the function gives appropriate data by printing some part of the dataset (e.g. as we did in the notes for pandas dataframes)
- demonstrate that the file `work/delNorteYYYY.csv` has been written for each year (where `YYYY` is the year) by showing the **file size** and **modification date** of the CSV files

In the submission, You **MUST** show that you have run the notebook (from start to finish) to prove that the codes work, so we will expect to see cells starting at [1].

4.2.2   Visualise the data [30%]

You must write code in a Jupyter notebook cell to plot mean temperature (Celcius) and Discharge (ML/day) as a function of day of year for the years 2016-2019 inclusive in the Jupyter notebook <u>and</u> save the plot to a graphics (PNG) file.

The plots must have suitable axis labels and titles. You should choose whether to use one or more plot or sub-plots for the datasets and years, and choose appropriate styles for the plots. Whilst you might try out several styles in deciding this, you should only submit the ones you decide to use in the end.

4.3 Submission

The due date for Part A (this piece of work) is 14 Nov, 2022 (first Monday after reading week). Part A represents 50% of final mark for the course. Submission is through the usual Turnitin link on the <u>course Moodle page</u>.
You must develop and run the codes in a single Jupyter notebook, and submit the work in a single notebook as a PDF file.

You must work individually on these tasks. If you do not, it will be treated as plagiarism. By reading <u>these instructions</u> for this exercise, we assume that you are aware of the UCL rules on plagiarism. You can find more information on this matter in your student handbook. If in doubt about what might constitute plagiarism, ask the course conveners.