COMP1730/6730 S1 2020 — Project Assignment

Matthew Alger

06-04-2020

Important

- The assignment is due **11:55 PM Sunday in week 11**, May 31.
- This assignment needs to be completed by yourself (i.e, not in a group). You can share ideas (with proper attribution) but your code and report must be your own.
- Include your university ID in every file you submit.
- This assignment is worth 15% of your grade for COMP1730/COMP6730.

Overview

In this assignment you will be doing a short data analysis and modelling task using some real-world geographical data. It is different to the homework assignments you have done up until this point in that for almost all of the questions, there is no single "right" answer. You are also not given any tests against which to check your answers (although you are encouraged to write your own to help you test the correctness of your functions). Because there is no single "right" answer, it will be important to justify the decisions and choices that you make while completing the assignment. This is important since it allows anyone relying on your results and conclusions to understand how they were obtained and whether they are suitable for a particular purpose.

Lake George, or Weereewa (in Ngunnawal), is an intermittent lake to the north-east of Canberra. It is well-known for having dramatic changes in water level. When full, it's around 145 km^2—20 times bigger than Lake Burley Griffin! But most of the time in contemporary history, it's been fairly empty. This unusual behaviour has given Lake George quite the reputation and even some persistent myths and legends, including everything from a secret underground link between Lake George and Mount Gambier, to bunyip and alien sightings. Of course, these legends are (probably) not true, and the fluctuating water levels of Lake George are explained by its unusually shallow geology, evaporation, and rainfall. In this assignment you will investigate the water level of Lake George and develop two models of how it fills and ebbs over time.

The Data

We have provided you with a CSV file `assignment_lake_george_data.csv` containing meteorological data for the Lake George region from 1990–2018, as well as the area and volume of the lake over time. This is made from data from the Bureau of Meteorology, Digital Earth Australia, and the CSIRO.

You may recall CSV files from Labs 6 and 8. They are text files made up of rows of data, with each column separated by commas. There are 9 columns in this dataset:

- Date, in YYYYMM format;
- Volume of Lake George, in litres (as at the end of the month);
- Area of Lake George, in square metres (as at the end of the month);
- Total monthly solar exposure, in megajoules per square metre;
- Total monthly rainfall, in mm;

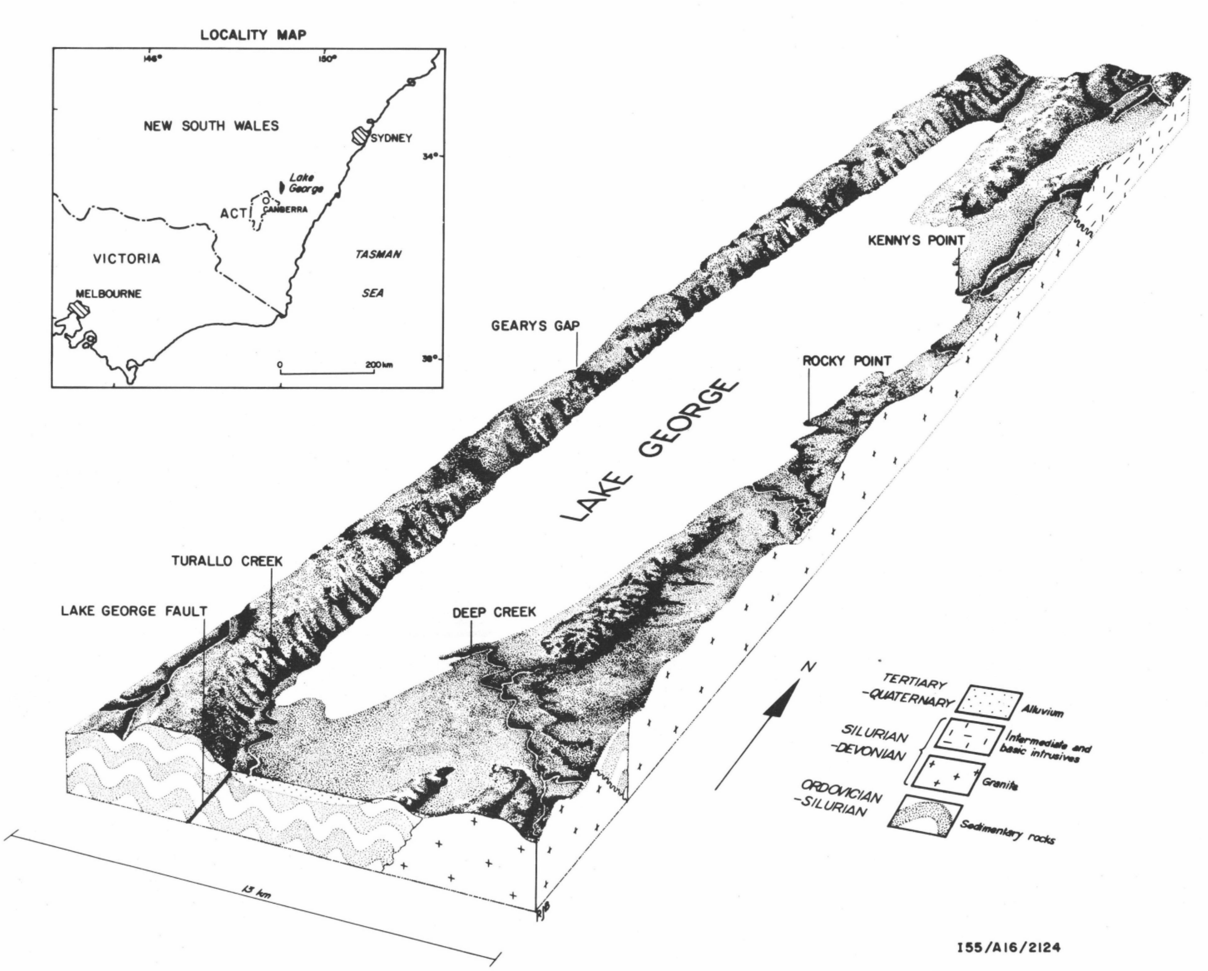

Figure 1: Geology of Lake George (Jacobson & Schuett, 1979).

- Average daily maximum temperature, in degrees Celsius;
- Average daily minimum temperature, in degrees Celsius;
- Average daily relative humidity, as a percentage; and
- Average monthly wind speed, in metres/second.

The file looks like this (but with more rows):

```
date,volume,area,solar_exposure,rainfall,max_temperature,min_temperature,humidity,wind_speed
199001,190850270907.59,141952429.50,22.10,22.80,22.82,8.44,37,1.64
199002,184683568988.20,141499648.55,19.40,83.30,22.32,8.41,40,1.26
199003,180741119308.03,141298750.00,16.30,17.60,21.86,8.38,42,1.39
```

This would correspond to the following table:

Date	Volume	Area	Solar exp.	Rainfall	Max temp.	Min temp.	Humidity	Wind
199001	190850270907.59	141952429.50	22.10	22.80	22.82	8.44	37	1.64
199002	184683568988.20	141499648.55	19.40	83.30	22.32	8.41	40	1.26
199003	180741119308.03	141298750.00	16.30	17.60	21.86	8.38	42	1.39

The Task

You are provided with assignment_template.py, which contains the basic functions of the assignment. The functions are incomplete. In this assignment, you will fill in the blanks and complete the missing functions. You will also write a short report about your functions and decisions.

Question 1: Reading the Data

Write a function that takes the file path of the dataset as input, reads the data, and returns the dataset in a suitable format. The assignment template contains a function for you to fill in:

```
def read_dataset(filepath):
    pass
```

`pass` means "do nothing", and you should remove it when you fill in this function. To load the data, you can then run

```
data = read_dataset('assignment_lake_george_data.csv')
```

as long as the CSV file is in the same directory as your assignment file.

You should read the data from `filepath`, and return it in an easy-to-use format. This can be any data type or data structure that you like, as long as it makes sense for the tasks you will be doing later in this assignment. You will be using this returned value in all other questions of the assignment, so make sure your choices here support your later solutions!

Hint - have a look at the remaining questions before deciding on what format to load your data in!

Question 2: Statistics about Lake George

Write four functions that output (return) the answers to the following questions:

a. What is the largest area covered by the lake?
b. What is the average volume of the lake?
c. What month and year had a rainfall closest to average?
d. Which month is the hottest on average?

The corresponding four functions are outlined in the assignment template:

```
def largest_area(data):
    pass
```

```
def average_volume(data):
    pass

def most_average_rainfall(data):
    pass

def hottest_month(data):
    pass
```

All of these functions should take as an argument the Lake George data in the same format you loaded it in Question 1 (for example, if you loaded the data as a list, then `data` is a list here).

Question 3: Lake George Topography

Lake George is notable for being very shallow. We want to identify points where the area increases quickly with little volume added, and vice-versa. We have written a function `area_vs_volume` which should plot the area of Lake George against its volume. Our function doesn't work though, and even if it did, the plot is so poor it doesn't really help us answer our question. Fix the function so that it works on your data, and improve the plot so it makes it easy for someone looking at it to identify the points we are interested in.

The function to fix is in the template:

```
def area_vs_volume(data):
    areas = []
    volumes = []

    plt.plot(areas)
    plt.plot(volumes)
    plt.show()
```

You will be marked on how good the plot looks (and how appropriate it is for the task) as well as whether the code works. You are free to pick a completely different plot type if you think it is more appropriate.

You should include a copy of your plot in your report.

Hint - have a look at some of the visualisation tips included in Lecture 12.

Question 4: Modelling Lake George

Lake George is an interesting lake because it is *endorheic*, meaning that it has no outflows: water only leaves the lake through evaporation. While the physical process of evaporation can be very complicated, in this question we will assume a much simpler situation. We will also assume that Lake George has only a single inflow, i.e. rainfall (thus the secret underground passage to Mount Gambier is outside the scope of the assignment).

We want to develop a *model* for how Lake George fills and ebbs over time. A model is a programmatic description or simulation of a physical process, like how much water is in the lake after it receives some amount of rain. In this question, you will build two different models of Lake George: a simple one which assumes that evaporation is constant, and a more complex one which assumes evaporation is dependent on some other variables.

Evaporation and rainfall are both measured in millimetres (mm)! 1 mm of rainfall corresponds to 1 litre of rain falling per square metre of area. Similarly, 1 mm of evaporation corresponds to 1 litre of water evaporating per square metre of area.

Part (a)

Use the monthly rainfall data to estimate the volume of Lake George for each month in the data, assuming a constant evaporation rate each month. Write a function `lake_george_simple_model` which takes the Lake George data in your format, as well as the assumed rate of evaporation (in mm/month), and returns a list of modelled lake volumes (in litres). We have outlined this function in the template:

```
def lake_george_simple_model(data, evaporation_rate):
    pass
```

Hint - in order to convert both water gained from rainfall (in mm/month) and water lost to evaporation (also in mm/month) into volume, you will need to make assumptions about both the catchment area and surface area of Lake George. You can always start by assuming they are both equal to the maximum surface area of the lake (which you calculated in Question 2) and then refining them from there if you want.

Another hint - you will need a starting point for your calculations. You can use the first record in the data set as the starting point for your models.

We have also given you a function `plot_volumes` to help visualise the results. You can use it like this:

```
plot_volumes(lake_george_simple_model(data, evaporation_rate))
```

Try plotting your model for different values of the evaporation rate. For example, if you set the evaporation rate to 55 mm/month and use `plot_volumes`, you should see something like this:

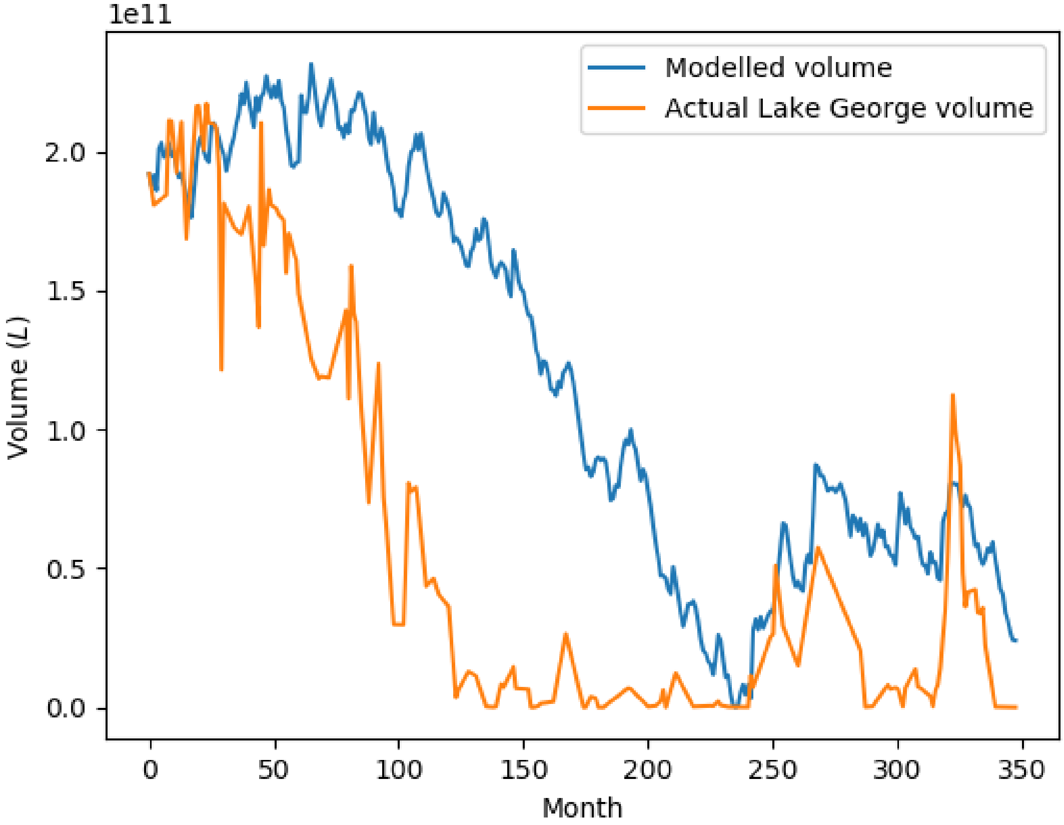

Figure 2: A simple model for Lake George with 55 mm/month evaporation.

Part (b)

While our simple model will give some very rough answers, it might be too simple to be of much use for real analysis tasks. For this question, instead of assuming evaporation is constant, use the following equation to calculate the evaporation rate E (in mm/month):

$$E = -3T_{\text{min}} + 1.6T_{\text{max}} - 2.5W + 4.5S - 0.4H$$

where T is the temperature (in Celsius), S is the solar exposure (in MJ/month/m^2), W is the wind speed (in m/s), and H is the humidity (as a percentage, i.e. as a number between 0 and 100). Write a function `lake_george_complex_model` that uses this evaporation rate instead of assuming the rate is constant. We have outlined this function in the template:

```
def lake_george_complex_model(data):
    pass
```

Question 5: Evaluating a Model

We can make as many models as we like, but if there's no way to *quantify* how good those models are, then we have no way to choose which model we should use.

Write a function `evaluate_model` that takes in your Lake George data and a list of volumes (in litres) and returns a float that indicates how bad your model is at estimating the real values. This float is called the *model error* and it should be equal to zero if and only if the model volumes are all exactly the same as the real volumes.

We have provided an outline in the assignment template:

```
def evaluate_model(data, volumes):
    pass
```

There are lots of different ways to calculate the error of a model like this - you might need to do a little bit of research to find something that is appropriate. Make sure you reference any ideas or material you use that is sourced from somewhere else.

Written Report

Answer the following questions in your written report, `answers.pdf`. Justify your answers by making reference to the code you've written.

1. Choose part of the code you have written for this assignment and explain it. Make sure to explain it at the right level: we don't want a line-by-line description.
2. Which model (simple or complex) is the best? Why?
3. What assumptions did you have to make in order to solve Question 4? Are they realistic? How would you improve your Lake George model?

In addition to answering these three questions, please provide a copy of the plot you generated in Question 3.

Requirements, Expectations, and Marking Criteria

You need to submit two files:

- `assignment.py`, the Python script containing your implementation of the assignment; and
- `answers.pdf`, a PDF version of your written report.

You can optionally submit a third file:

- `assignment_tests.py`, which is a Python script containing any tests you have written to verify the correctness of your functions.

Note that while the report component accounts for a small part of the assignment mark, **the report is required**. If you fail to submit an individual report, your mark for the assignment may be reduced (even as far as to zero), regardless of the mark of your code.

For your code, we have the following requirements:

- Your code must be syntactically correct: it must run in Python 3.
- You can use any modules that are available in the Anaconda 3 Python distribution.
- You must not change the names of the functions in the template, or change their parameters (i.e. you can't change their names or types). You may add new functions if you wish (indeed, appropriate use of functional problem decomposition is part of the marking criteria).
- You should not use any global variables or have any code outside of function definitions unless it is in the `if __name__ == '__main__'` block.
- Your code shouldn't raise any unintentional exceptions or warnings.

We will mark your code based on correctness and quality. "Correctness" means that your functions run without error and return acceptable answers for all valid inputs. "Quality" means your code is readable, good, and efficient:

- You should use docstrings and comments where it is appropriate. The content of docstrings and comments should be clear and accurate.
- Your function and variable names should make sense and be descriptive.
- You should use suitable data types to solve problems.
- You should organise your code appropriately, using additional functions where it is helpful to do so. Avoid code repetition. In particular, although the assignment specifies a number of different functions for you to implement, these are not (always) meant to be self-contained. If there is functionality that is common between the questions and that can be isolated into one or more separate functions that are reused in several places in your code, we expect you to do so.
- Your code should be reasonably efficient: don't make the computer do too much unnecessary work.

We will also mark the answers in your written report based on correctness and clarity. **If we cannot understand or find your answers in the PDF file you submit, you may receive 0 marks.** Your written answers should be:

- clear (it should be easy to find and understand your answers);
- concise (write what is relevant to answer the questions; do not overcomplicate);
- well-organised (use headings and subheadings where appropriate);
- relevant to the rest of your assignment submission; and
- 1–2 pages. **We will stop reading after 2 pages! (Including your plot for Question 3).**

In this assignment, you will have to make some choices on how to design your solution to problems, and you will be asked to justify these choices in your written report. You should show understanding of the problem and your solution, and convince your marker that your solution solves the problem in an appropriate way. Much like real life, many questions in this assignment do not always have a single correct answer, so it is especially important to justify the decisions, assumptions, and solutions you've made.

The marking of the assignment is divided into roughly 40% code quality and organisation, 50% code and functionality for specific questions (10% per question) and 10% written report. A full marking rubric will be made available.

Note that although your written report is only worth 10% of the assignment marks, **the report is required**. If you fail to submit an individual report, your mark for the assignment may be reduced by much more than 10% (even as far as to zero).

Submission

You must submit the assignment through the submission link on wattle. You must submit a zip folder containing `assignment.py` (the code) and `answers.pdf` (the written report) and optionally `assignment_tests.py` (any testing functions you used to test your code).

You can upload new versions of your submission up to the deadline. However, remember that we can only see your latest submission, and that is what we will mark.

You must write your ANU ID in your report and in the comment at the beginning of the code file(s).

The assignment is due **May 31, 2020, at 11:55 pm**. This deadline is hard. Late submissions will not be accepted, unless you have received an approved extension **before** the deadline.

Extensions can only be given in extenuating circumstances as defined by ANU policy; this means accident, illness, or other things that you could not reasonably have anticipated or avoided. Failure to plan in advance to spend sufficient time working on the assignment is neither unforseeable nor unavoidable. If you think you have grounds for an extension, you should contact the course conveners as soon as possible and provide written evidence in support of your case (such as a medical certificate). The course convener will then decide whether to grant an extension and inform you as soon as practical.

Please note that although there has been significant disruption this semester, the assignment scope and deadlines have already been adjusted to reflect the current situation with the COVID-19 pandemic. As a result, unless you have been particularly severely impacted, a general statement about difficulties due to the COVID-19 pandemic will not be considered.

Plagiarism and collusion

You cannot work together to develop your solution to this assignment. You can of course seek help from others to learn and improve your understanding, but not to obtain solutions to the specific assignment problems.

Your assignment must be entirely your own work. Both the report and code will be considered under the usual individual plagiarism rules. If you are unsure about what constitutes plagiarism, please read through the ANU Academic Honesty Policy.

If you do include ideas or material from other sources (in your code or your report), then you clearly have to make attribution by providing a reference to the material or source in your report. We do not require a specific referencing format, as long as you are consistent and your references allow us to find the source, should we need to while we are marking your assignment. Marking will be based on original content; if you have borrowed extensively from other sources, we may consider that in determining your mark.

If you are found to have have engaged in plagiarism you will usually receive a mark of 0 for the **entire assignment**. If you assist another student in engaging in plagiarism, for example by giving them your code or instructing them on how to solve the problem, you will also normally receive a mark of 0 for the **entire assignment**. In either case, you may also receive additional penalties as appropriate under the ANU Academic Honesty Policy.

References

- McVicar, Tim (2011): Near-Surface Wind Speed. v10. CSIRO. Data Collection.

  https://data.csiro.au/dap/landingpage?pid=csiro%3AWind_Speed
- Jacobson, G. & Schuett, A. W. (1979): Water levels, balance, and chemistry of Lake George, New South Wales. BMR Journal of Australian Geology & Geophysics, 4, 25-32.
- Bureau of Meteorology (accessed 2020): Climate Data Online.