

COMP1730/6730 Assignment Description

Matthew Alger

26-04-2018

Introduction

The Atlas of Living Australia is a project to collate biodiversity data from around Australia. It records information on where and when different species have been observed, and currently lists over 120,000 species. The data are sourced from a variety of projects and made publicly available online, where they can be used by anyone for environmental monitoring, conservation planning, education, and citizen science.

In this assignment you will write a program to analyse biodiversity data derived from the Atlas of Living Australia and the ANU tree database. `ala-acton.txt` is part of the Atlas centred on the Australian National University's Acton campus, containing every recorded sighting of a species at the University since 1960. You will also write a short report about your design decisions and some analysis using your program.

Important

- The assignment is due at the end of week 11 on Sunday 20th May, 11:55 PM.
- The code section of the assignment can be completed in groups of 1–3. These groups must be registered on WATTLE by Friday of Week 9.
- The report section of the assignment must be completed individually. You can share ideas (with proper attribution) but your write-ups must be your own.
- Do not include your names anywhere on the assignment. Include your university IDs in every file you submit.

Requirements and marking criteria

You need to submit two files: `assignment.py`, the Python script containing your assignment submission; and `report.pdf`, a PDF version of your report.

- Your code must be syntactically correct or it won't be marked.
- Your code should be good-quality:
 - You should use docstrings and comments where appropriate.
 - 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 functions where necessary.
 - Your code should be reasonably efficient: Don't make the computer do too much unnecessary work.
- You can only import the following modules (if you want them):
 - `numpy`
 - `matplotlib`
 - `os`
 - `random`

- `itertools`
- `functools`
- `collections`
- `math`
- `csv`
- `time`
- `datetime`
- In particular you cannot import `pandas`.
- You shouldn't change the arguments or names of functions in the template, except for `function` for Question 3.
- You can add your own functions if you like.
- You should not use any global variables or have any code outside of your functions unless it is in an `if __name__ == '__main__':` block.
- Your report should be:
 - clear, concise, and well-organised, using headings and subheadings where appropriate;
 - relevant to your code; and
 - 1–3 pages.
- Your code shouldn't raise any unintentional exceptions or warnings.

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 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 have a “correct” answer, so it is especially important to justify the decisions, assumptions, and solutions you’ve made.

The assignment is divided into roughly 60% code and 40% report, with marks allocated for:

- Code quality: 20%
- Functionality: 40%
- Design: 20%
- Explanation and understanding: 20%

There are six questions with uneven weightings:

- Question 1: 20%
- Question 2: 10%
- Question 3: 10%
- Question 4: 20%
- Question 5: 20%
- Question 6: 20%

Please note that these percentages are indicative only.

The data

You are provided with `ala-acton.txt`, a table of species sightings in Acton. There are 11 columns:

- *Name* — The name of the species sighted.
- *Latitude, Longitude* — The location of the sighting.
- *Locality* — A description of the location.
- *Date* — When the sighting was, in year-month-day format.
- *Class, Order, Family, Genus, Species* — The taxonomy of the species sighted.
- *Duplicate* — Whether this sighting is a duplicate.

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

Name	Latitude	Longitude	Locality	Date	Class	Order	Family	Genus	Species	Duplicate
Noisy Miner	-35.28111	149.1222	ANU School of Art	2012-09-21	Aves	Passeriformes	Meliphagidae	Manorina	Manorina melanocephala	False
Noisy Miner	-35.28111	149.1222	ANU School of Art	2012-09-19	Aves	Passeriformes	Meliphagidae	Manorina	Manorina melanocephala	False
Australian Magpie	-35.27386	149.1126251		2009-09-21	Aves	Passeriformes	Artamidae	Cracticus	Cracticus tibicen	False
Crimson Rosella	-35.27917	149.1095	Australian National Botanic gardens	2001-12-08	Aves	Psittaciformes	Psittacidae	Platycercus	Platycercus elegans	False

Each row of the table has its own line in the file, and every column is separated by a pipe character (|). The above table would look like this in the file:

```
Name|Latitude|Longitude|Locality|Date|Class|Order|Family|Genus|Species|Duplicate
Noisy Miner|-35.28111|149.1222|ANU School of Art|2012-09-21|Aves|Passeriformes|Meliphagidae|Manorina|Manorina melanocephala|False
Noisy Miner|-35.28111|149.1222|ANU School of Art|2012-09-19|Aves|Passeriformes|Meliphagidae|Manorina|Manorina melanocephala|False
Australian Magpie|-35.27386|149.1126|2009-09-21|Aves|Passeriformes|Artamidae|Cracticus|Cracticus tibicen|False
Crimson Rosella|-35.27917|149.1095|Australian National Botanic gardens|2001-12-08|Aves|Psittaciformes|Psittacidae|Platycercus|Platycercus elegans|False
```

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.

1: Loading the Atlas of Living Australia

Write a function `load_atlas` which reads the Atlas data and returns it in a suitable format. The assignment template includes the basic outline of this function:

```
def load_atlas(filename):
    pass
```

(`pass` means “do nothing” in Python. You should remove it when you write the function.)

You should read the data from `filename`, parse it, and return it in an easy-to-use format. You will be using this returned value in all other questions of the assignment, so make sure your choices here support your later solutions.

In your report, justify your choice of types and format of the return value.

2: Removing duplicate sightings

Write a function `remove_duplicates` that takes the Atlas data you loaded in Question 1 and removes all duplicate sightings. Whether a sighting is a duplicate or not is shown in the “Duplicate” column. The assignment template includes the basic outline of this function:

```
def remove_duplicates(atlas_data):
    pass
```

In your report, explain how this function works, and how you could write it in another way.

3: Mapping the sightings

Matthew has written a function that shows a map of sightings. It takes a list of tuples representing the coordinates of each sighting and plots them on top of a map from OpenStreetMap. However, because Matthew didn't take COMP1730, their code is hard to read and doesn't use good code quality conventions. Rewrite the function in a better, neater way, and make it take your Atlas data as input.

The function currently looks like this:

```
def function(x):
    i = 0
    y = []
    z = []
    while i < len(x):
        y.append(x[i][0])
        z.append(x[i][1])
        i = i + 1

    plt.imshow(mping.imread('map.png'), extent=MAP_EXTENT)
    plt.scatter(z, y)
    plt.show()
```

In your report, include a screenshot of the map and explain what you changed and why you changed it.

4: Looking for birds

Jeff likes to sit and drink coffee at a local café while he marks assignments. His favourite seat is at latitude -35.276159, longitude 149.120893. If Jeff sits in his favourite seat, what kind of birds is he likely to see?

Write a function `nearest_bird` that finds the bird sighting closest to a given location. The outline of this function is in the template:

```
def nearest_bird(atlas_data, latitude, longitude):
    pass
```

The approximate distance in kilometres between two coordinates is given by the following formula:

$$d = \frac{6371\pi}{180} \sqrt{(\phi_1 - \phi_2)^2 + (\lambda_1 - \lambda_2)^2}$$

d is the distance, ϕ_1 and ϕ_2 are the latitudes of point 1 and point 2 in degrees, and λ_1 and λ_2 are the longitudes of point 1 and point 2 in degrees.

Birds are in the taxonomic class Aves.

Here are some example outputs of the function:

```
>>> nearest_bird(atlas_data, -35.283705, 149.122700) # Shine Dome
'Australian Coot'
>>> nearest_bird(atlas_data, -35.283334, 149.116765) # Ursula Hall
'Red Wattlebird'
>>> nearest_bird(atlas_data, -35.276733, 149.125674) # Kinloch Lodge
'Magpie-lark'
```

In your report, explain how your function works and include the name of the bird closest to Jeff's favourite seat.

5: Looking for more birds

The function in Question 4 doesn't take into account that some birds are more common than others, instead showing only the nearest bird to a given coordinate. In this question, you need to write the function `most_common_birds` which returns the most common birds in a location. The function is outlined in the assignment template:

```
def most_common_birds(atlas_data, latitude, longitude, distance):  
    pass
```

Return the names of the most common birds within `distance` km of the given location. If there are multiple birds that are equally common, return a list of all their names. If there is just one bird, return a list with one bird in it.

Here are some example outputs of the function:

```
>>> most_common_birds(atlas_data, -35.287413, 149.116274, 0.1) # Crawford  
['Noisy Miner']  
>>> most_common_birds(atlas_data, -35.274142, 149.120336, 0.03) # Brian Anderson  
['Common Starling', 'Swift Parrot']  
>>> most_common_birds(atlas_data, -35.275662, 149.126026, 0.1) # Lena Karmel  
['Australian Raven', 'Galah', 'Willie Wagtail', 'Crested Pigeon', 'Silver Gull']
```

In your report, explain how your implementation of `most_common_birds` works, and write down the most common birds within 80 m of Jeff's favourite seat at the café. Use `most_common_birds` to find out what the most common bird at ANU is, and include this too. Explain how you found the most common bird at ANU. You will need to decide what constitutes "ANU" and how to use your code to solve the problem.

6: The Eucalypt Tree Club

You are the vice-president of the ANU Eucalypt Tree Club (ETC). The president has tasked you with finding a good location for your new club common room. Naturally, you want to make sure that your common room is near as many eucalyptus trees as possible. Write a function `eucalypt_club_location` that decides where your common room should be, outputting a tuple containing the latitude and the longitude of your new common room location. This function is outlined in the template:

```
def eucalypt_club_location(atlas_data):  
    pass
```

In your report, state the location of the ETC common room. Give the latitude and longitude as well as a brief description of the location. Explain your function, justify the method you chose to find the location, and justify why the location that you wrote in your report suits your needs. Be specific! The code and report components of this question are equally weighted.

The report

Alongside your code, you also have to submit a written report that answers the following questions:

1. Justify your choice of types and format of the return value for `load_atlas`.
2. Explain how `remove_duplicates` works, and how you could write it in another way.
3. Show your map of sightings and explain what improvements you made to `map_sightings`.
4. Explain your implementation of `nearest_bird` and name the bird closest to -35.276159, 149.120893.
5. Explain your implementation of `most_common_bird` and name the most common bird within 80 m of -35.276159, 149.120893. Name the most common bird at ANU.
6. Decide where to put the Eucalypt Tree Club (ETC) common room. Include the latitude and longitude as well as a brief description of the location. Explain your implementation of `eucalypt_club_location` and justify your method. Justify the location you chose for the ETC common room. Be specific.

Submission

- Submit the assignment on WATTLE.
- We will mark the most recent submission before the deadline.
- Every group member must submit a zip folder containing `assignment.py` (the code) and `report.pdf` (the report).
- The code should be identical for every member of the group.
- The report must be entirely your own work.
- No late submissions will be accepted without prior arrangement with the convenor.
- Students will only be granted an extension on the submission deadline in extenuating circumstances as defined by ANU policy. If you think you have grounds for an extension, you should notify the course convener 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. Extensions will only apply to the individual reports.

Plagiarism

The only groupwork in this assignment should be on the code. All reports must be individual submission. We do encourage you to discuss your reports, but we expect you to do the write-up by yourself. Reports will be considered under 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.

Data sources

Data are principally sourced from the Atlas of Living Australia, downloaded at <https://biocache.ala.org.au/occurrences/search?q=%3A835.279446&lon=149.11945&radius=1.0>, accessed on Sun Apr 15 14:10:57 AEST 2018. Data augmented by the Australian National University Tree Database, accessed on Wed April 25th AEST 2018. For full citation see [here](#).