COMP1730/6730 S2 2018 Project Assignment

Semester 2, 2018

Introduction

In 2003, western Canberra was hit by a bushfire that famously devastated Mount Stromlo Observatory and a number of homes in Weston Creek.

The spread of bushfires is affected by the type and density of vegetation, as well as the wind speed. The ACT has a fire danger notification system based on these factors along with the local humidity, temperature, and other weather data.

The ACT Vegetation Subformation Map and the ACT Average Wind Speed Map are maps of the Canberra region developed by the ACT Government that describe the vegetation and wind speed respectively. In this assignment we will model the spread of bushfires using these maps and compare our model to the 2003 bushfire.

Important

- The assignment is due at the end of semester week 11, on **Sunday the 21st October, 11:55 PM**. This deadline is **hard**. No submissions will be accepted after the deadline. Submissions **must** be uploaded to Wattle (as described below). Submissions sent by other means (such as email) will not be accepted.
- The assignment consists of two components: code and a report. For the code part, you may work in groups of up to three students. A group selection activity will be available on Wattle. If you are working in a group, you **must** register this on Wattle, no later than by the end of Friday in semester week 10. If you are not going to work in a group, please add yourself to the “I will do the assignment on my own” group.

  No collaboration between groups is allowed. If you have not declared that you are working in a group, by signing up to one on Wattle, you are requried to do all parts of the assignment by yourself.
- The report **must** be done individually. You can discuss ideas (within your group) but all writing must be your own. Sharing report documents, in draft or final form, before the submission deadline is absolutely not permitted.
- Deadline extensions can be granted in cases of unforeseeable and unavoidable circumstances that prevent you from working effectively, such as accidents or illness. Extensions can only be given if requested **before** the deadline.

  Extensions apply only to individual components of the assignment, not to group work. This means that if one person in a group is absent (for example, sick) and is granted an extension, the other group members must still submit the group’s code by the deadline. If you choose to work in a group, it is your responsibility to organise your work so that the whole group can not be held up by one persons absence.

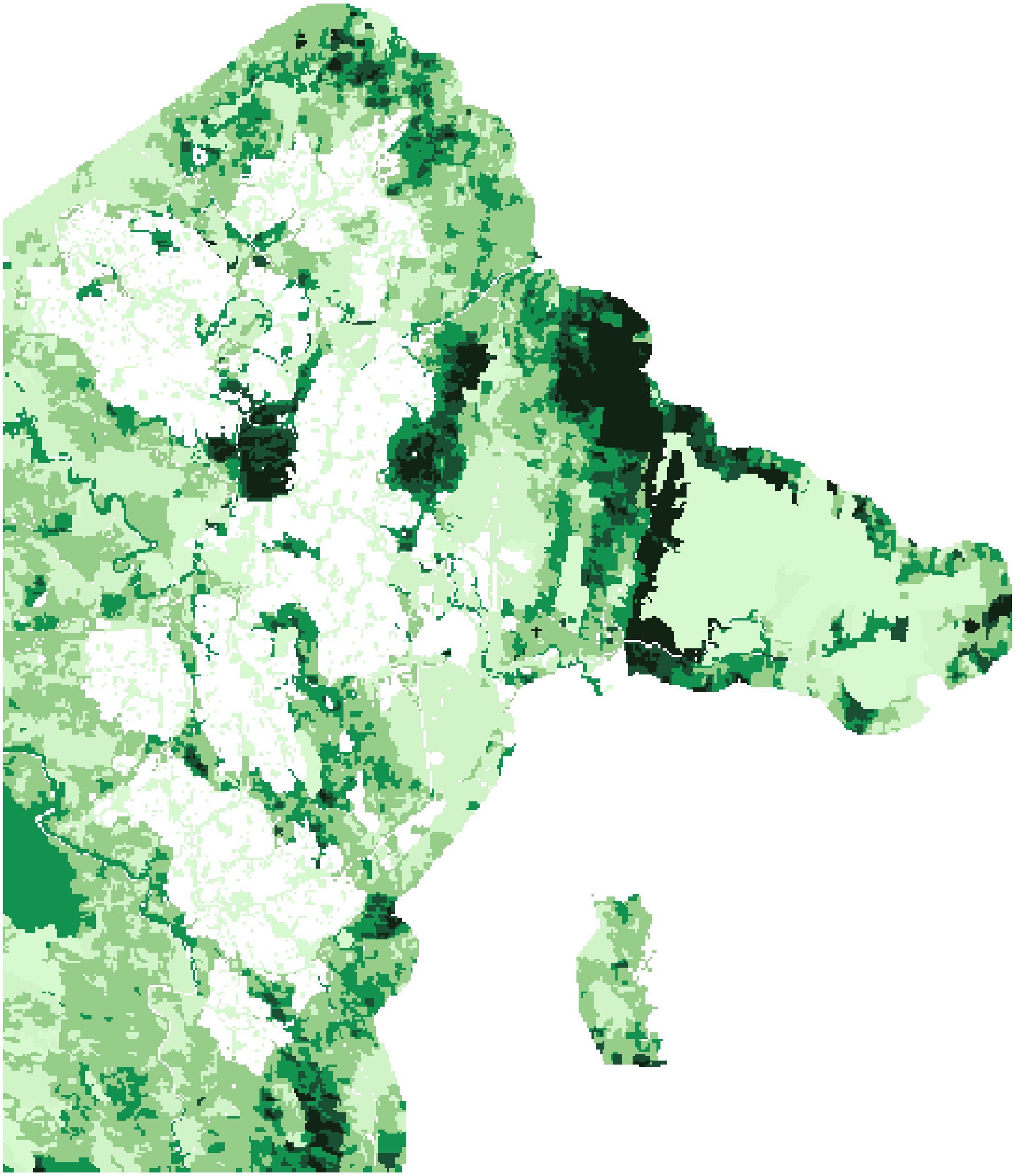


Figure 1: ACT vegetation density map.

- The assignment has an extra question (question 8) for students in COMP6730 (masters students) only. Both the code and report parts of Question 8 **must** be completed individually.

  Note that it is fine for a group to consist of a mix of COMP1730 and COMP6730 students. The masters students in the group must complete Question 8 individually.

Requirements and marking criteria

What to submit

You must submit two files: `assignment.py`, the Python file containing your assignment code; and `report.pdf`, a PDF version of your report. Students in COMP6730 must also submit an additional file, `question8.py`, with their code for Question 8. All files should be packaged together in a single zip file which must be submitted through the assignment submission link on Wattle (https://wattlecourses.anu.edu.au/mod/assign/view.php?id=1427051).

You must include your university ID (all authors' IDs, for code files that are joint work) in every file you submit.

General requirements

- **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. Remember that docstrings should provide a high-level summary of the purpose, arguments and return value, and limitations of each function; this high-level documentation should not assume that the reader knows the assignment problem specification. Detail (in-line) comments should be used to explain or clarify parts of the code whose functioning is not obvious.
  - 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, defining additional functions where it improves code organisation and reuse. Avoid repeating identical or very similar code.
  - Your code should be reasonably efficient: Don't make the computer do too much unnecessary work.
- For Questions 1-7, you cannot use any modules other those in the python standard library. (Most likely, you will not have use for any modules other than `math`, `csv`, and `random`; `random` is needed only for Question 7.)

  Masters students also need to import `matplotlib` for Question 8, but *only* for this question.
- You may not change the names or arguments of functions in the code template.
- You can define additional functions if you want. Indeed, you should do this when it helps improve the organisation of your code and avoid repetition.
- You must not use any global variables and you should not have any code outside of your function definitions (unless it is in the `if __name__ == '__main__':` suite).
- Your report should be:
  - clear, concise, and well-organised, using headings and subheadings as appropriate to indicate where the answer to each question is found;
  - relevant to your code; and

  - 1–3 pages.

- Your code must not raise any runtime errors (exceptions) when run.

In this assignment, you will have to make some choices as to how you design your solution to problems, and you will be asked to justify these choices in your report. A justification should demonstrate that you understand not only your chosen solution method, but also what the alternatives were, and the advantages and disadvantages of your choice compared to those alternatives. 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.

Marking

The assignment accounts for 20% of your final course mark. It will be marked out of 100 (110 for COMP6730 students), and the result scaled to a mark out of 20. The weighting of the questions in the assignment is as follows:

- Question 0 (10 marks)
- Question 1 (15 marks)
- Question 2 (10 marks)
- Question 3 (10 marks)
- Question 4 (15 marks)
- Question 5 (15 marks)
- Question 6 (10 marks)
- Question 7 (15 marks)

COMP6730 (i.e. masters students) have an additional question, Question 8. This is worth an additional 10 marks and the assignment as a whole will be marked out of 110.

For each question, marks are divided into roughly 70% for the code and 30% for the report, with marks allocated for:

- Functionality 45%
- Code quality 25%
- Report 30%

The data

You will use several different data files in this assignment. The first two are based on the ACT Vegetation Subformations dataset and describe the vegetation of the ACT. The third data file is based on the ACT Average Wind Speed Map. The fourth data file is based on the 2003 Bushfire (Affected Areas) dataset and shows which areas of the ACT were affected by the 2003 bushfire.

The files are in comma-separated values (CSV) format, and store values in a grid of locations, or "cells". Each cell represents a 100 m × 100 m area. Each row in the files is a number of grid cells delimited by commas "," and each row is on a new line. Each row of the file represents a west-east line and each column represents a north-south line. For example, if we have a grid of data:

```
A A A A B
A A A B
```

```
A A A B
A A B
A B   C C
B
```

it is represented as

```
A,A,A,A,A,B
A,A,A,A,B,
A,A,A,B,,
A,A,B,,,
A,B,,C,C,
B,,,,,
```

in the csv file.

In `vegetation_type.csv`, the values are types of vegetation. In `vegetation_density.csv`, the values are the percentage density of vegetation. In `wind.csv`, the values are the wind speed in km/h. In `2003_bushfire.csv`, the values are either a 1 or a 0 depending on whether the location was affected or not by the 2003 bushfire respectively. `initial_2003_bushfire.csv` is exactly the same format, but it contains a map of bushfire locations before the 2003 bushfire had fully spread.

Some locations don't have data (for example, they might not be in the ACT). These locations are left blank in each file.

There are three versions of these files, of increasing size. These are stored in separate directories. The smallest set of files is in the directory `anu`, and shows just the area around ANU and Black Mountain.

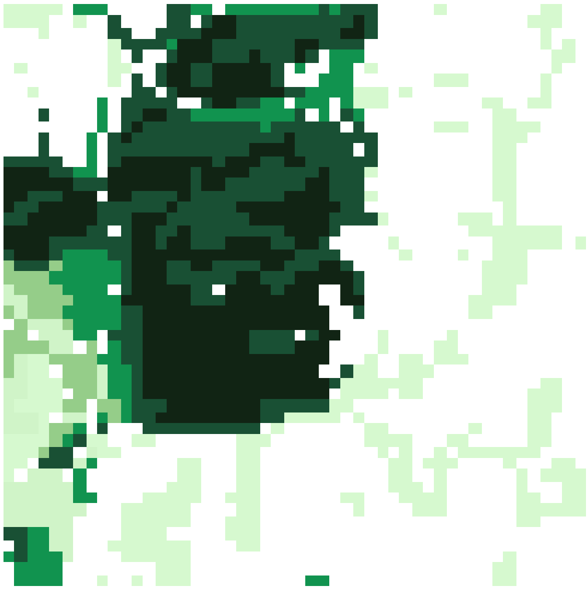


Figure 2: ANU vegetation density map.

The next-largest files are `south` which show Woden, Weston Creek, and Tuggeranong, centred on Mount Taylor.

Finally, the `act` data set contains all available data.

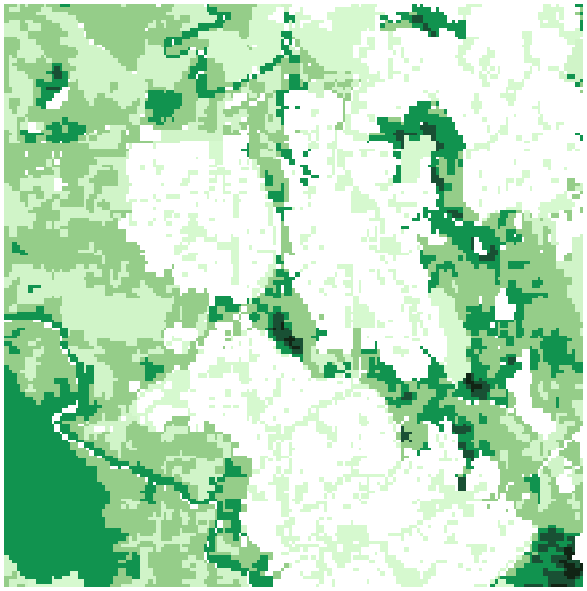


Figure 3: Woden vegetation density map.

All together, there are fifteen files:

- /anu/
  - /vegetation_type.csv
  - /vegetation_density.csv
  - /wind.csv
  - /2003_bushfire.csv
  - /initial_2003_bushfire.csv
- /south/
  - /vegetation_type.csv
  - /vegetation_density.csv
  - /wind.csv
  - /2003_bushfire.csv
  - /initial_2003_bushfire.csv
- /act/
  - /vegetation_type.csv
  - /vegetation_density.csv
  - /wind.csv
  - /2003_bushfire.csv
  - /initial_2003_bushfire.csv

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 0: Data and design

Examine the data files, making sure you understand what they represent. (This is a report-only question. If you like, do some of the later questions and then come back to this question.)

Report

In your report, answer the following questions:

a. The maps are stored as CSV files. This is not the only way that the data can be stored. Suggest another way, and explain what its advantages and disadvantages would be.
b. In Question 1, we will load each of these files into a lists of lists. Describe some advantages and some disadvantages of using a list of lists to represent the data.
c. Suggest another data structure we could use to represent these files. Compare it to the list of lists. What are the advantages and disadvantages of your proposed data structure?
d. We have provided data about the vegetation and wind speeds in the ACT, which we will use later on to model bushfire risk. data.act.gov.au provides many ACT Government datasets such as these. Suggest (and provide a URL to!) another dataset that could be useful for modelling bushfire risk, and explain why this would be a useful addition. (You don't have to (and shouldn't) use this data - just suggest a dataset that could be useful.)

Question 1: Loading the datasets

In this question, you need to write a function that loads each map from a file into a list of lists. For example, the 3 × 3 grid

```
1,2,3
3,3,3
1,2,3
```

would be loaded into a list of 3 lists `[[1,2,3], [3,3,3], [1,2,3]]`.

The assignment template contains placeholder functions for you to fill in:

```python
def load_vegetation_type(filename):
    pass

def load_vegetation_density(filename):
    pass

def load_wind_speed(filename):
    pass

def load_bushfire(filename):
    pass
```

`pass` means "do nothing" in Python. Remove it when you fill in the function.

In each of the four functions, you can assume that the `filename` parameter is a string, which is the path to the file to be read. Each function should return lists of lists, with each sublist representing one row of cells. You can choose what the type of the value in each cell is.

We have provided functions that will let you visualise the different maps to make sure you've loaded them correctly:

```
from visualise import show_vegetation_type
from visualise import show_vegetation_density
from visualise import show_wind_speed
from visualise import show_bushfire
```

To use one of the visualisation functions, call it with the data structure returned by your corresponding loading function. For example,

```
veg_density_map = load_vegetation_density("anu/vegetation_density.csv")
show_vegetation_density(veg_density_map)
```

should reproduce the image shown in Figure 2 above.

Report

In your report, answer the following questions:

a. What type does each of your file-loading functions return?
b. Explain how you accounted for blank values, and why you chose to account for them in the way you did.
c. How many blank values are there in each file?

Question 2: The maximum wind speed

To get into the data, let's answer a simple question: What's the highest wind speed in the dataset? You need to write the function `highest_wind_speed`, for which we have provided a placeholder in the assignment template:

```
def highest_wind_speed(wind_speed_map):
    pass
```

`wind_speed_map` is the map, in the form of a list of lists, returned by `load_wind_speed`. The function should return the highest wind speed in each map as a float.

Report

In your report, answer the following questions:

a. What is the highest wind speed in each map?
b. What is the time complexity of your implementation of this function? (Remember that time complexity is given as a function of the size of the input. Be careful to specify what you consider to be the size of the wind speed map; for example, is it the number of cells, or the size of each side of the map?)

Question 3: The most common vegetation

What's the most common type of vegetation in the dataset? We can define "most common" in two ways:

- Is in the most cells, or
- Covers the most area.

To calculate the former, we need to count how many cells of each type of vegetation there are. To calculate the latter we need to calculate the area covered by vegetation of each type, taking into account the density of each cell (i.e. multiplying the area of each cell by the density of that cell). Remember that each cell is 100 m × 100 m.

You need to write a function `count_cells` that counts the number of cells filled by each kind of vegetation, and a function `count_area` that counts the area covered by each kind of vegetation. These functions should **print** the number of cells or area occupied by each kind of vegetation. `count_cells` should print the kind of vegetation, then a `:`, then the number of cells. For example, the output for the `anu` dataset should look like this:

```
Open Forest: 368
Forest: 370
Open Woodland: 50
Woodland: 125
Pine Forest: 0
Arboretum: 26
Grassland: 65
Shrubland: 0
Golf Course: 0
Urban Vegetation: 315
```

`count_area` should print the kind of vegetation, then a `:`, then the area in m^2. For example, the output for the `anu` dataset should look like this:

```
Open Forest: 2944000.00 sq m
Forest: 3700000.00 sq m
Open Woodland: 100000.00 sq m
Woodland: 625000.00 sq m
Pine Forest: 0.00 sq m
Arboretum: 0.00 sq m
Grassland: 13000.00 sq m
Shrubland: 0.00 sq m
Golf Course: 0.00 sq m
Urban Vegetation: 0.00 sq m
```

The assignment template contains placeholder functions for you to fill in:

```python
def count_cells(vegetation_type):
    pass

def count_area(vegetation_type, vegetation_density):
    pass
```

These functions should take as arguments the map (lists of lists) returned by `load_vegetation_type` and `load_vegetation_density`. Neither function should return anything (they should only print).

Note that Urban Vegetation, Arboretum, Golf Course, Shrubland, and Pine Forest have no defined vegetation density, so they will all have zero area.

Report

In your report, answer the following questions:

a. How many cells are covered by each vegetation type?
b. What is the total area covered by each vegetation type?
c. If you input a `vegetation_type` list containing n lists (each containing n elements themselves), then how many times do you index the list (in terms of n)?

Question 4: Fire risk

The risk of fire is dependent on the vegetation type, vegetation density, and the wind speed. In this question we will estimate the risk of fire in each cell.

You need to write a function `fire_risk` that calculates the fire risk for a single cell. This function will take as arguments the indices of the cell to calculate fire risk for (as integers), the vegetation type grid, the vegetation density grid, and the wind speed grid (as lists of lists). The assignment template contains a placeholder function to fill in:

```
def fire_risk(x, y, vegetation_type, vegetation_density, wind_speed):
    pass
```

To calculate the fire risk for a cell, we add up the *fire risk factors* for each nearby cell. The fire risk factor for a cell is $\sqrt{a + density}$, where

- $a = 0.2$ for Shrubland and Pine Forest,
- $a = 0.1$ for Arboretum,
- $a = 0.05$ for Urban Vegetation and Golf Course, and
- $a = 0$ for everything else.

How far away a nearby cell is depends on the wind speed. If the wind speed is n, then a cell is nearby if it is closer than $\lfloor n \rfloor$ cells away. If $\lfloor n \rfloor = 0$, or if the wind speed is blank, then only consider the current cell.

($\lfloor x \rfloor$ is the mathematical floor function, which means "round x down to the nearest integer.")

We've provided a function that will let you visualise the fire risk map:

```
from visualise import show_fire_risk
```

This function takes as arguments your `fire_risk` function, the vegetation type grid, the vegetation density grid, and the wind speed grid. That is, if you call it like this:

```
density_map = load_vegetation_density("anu/vegetation_density.csv")
type_map = load_vegetation_density("anu/vegetation_type.csv")
wind_speed_map = load_wind_speed("anu/wind.csv")
show_fire_risk(fire_risk, type_map, density_map, wind_speed_map)
```

it should produce the fire risk plot for the ANU map, which is shown in Figure 4 below.

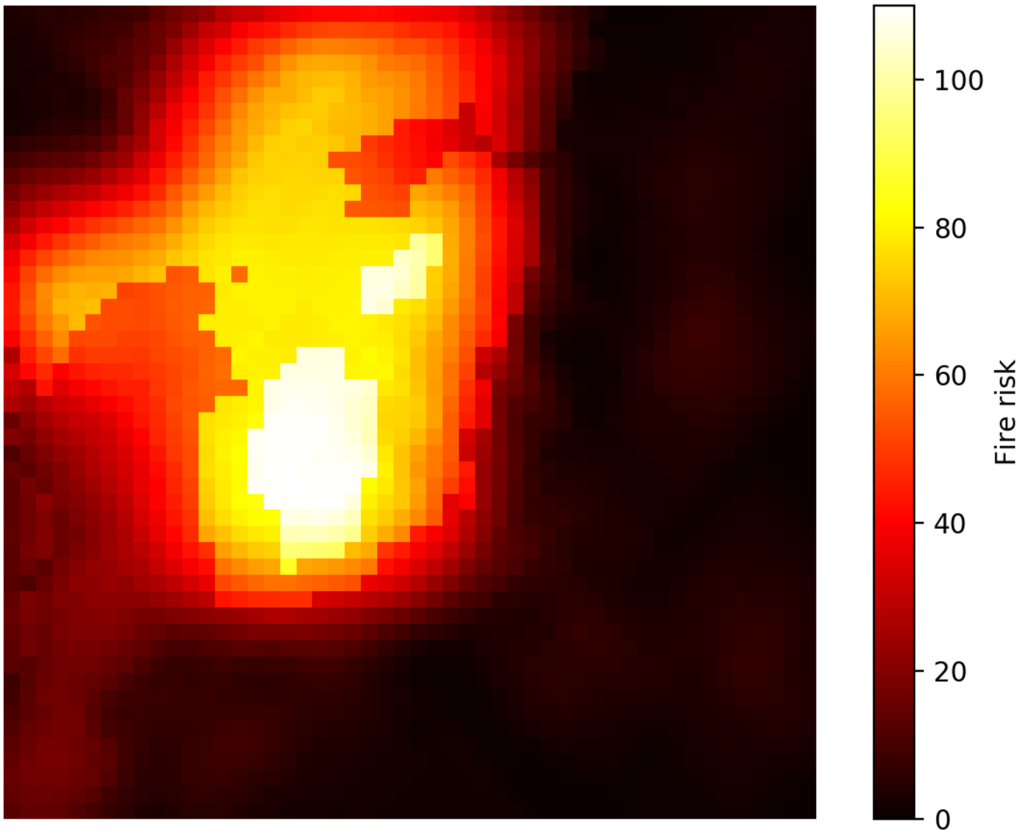


Figure 4: ANU fire risk map.

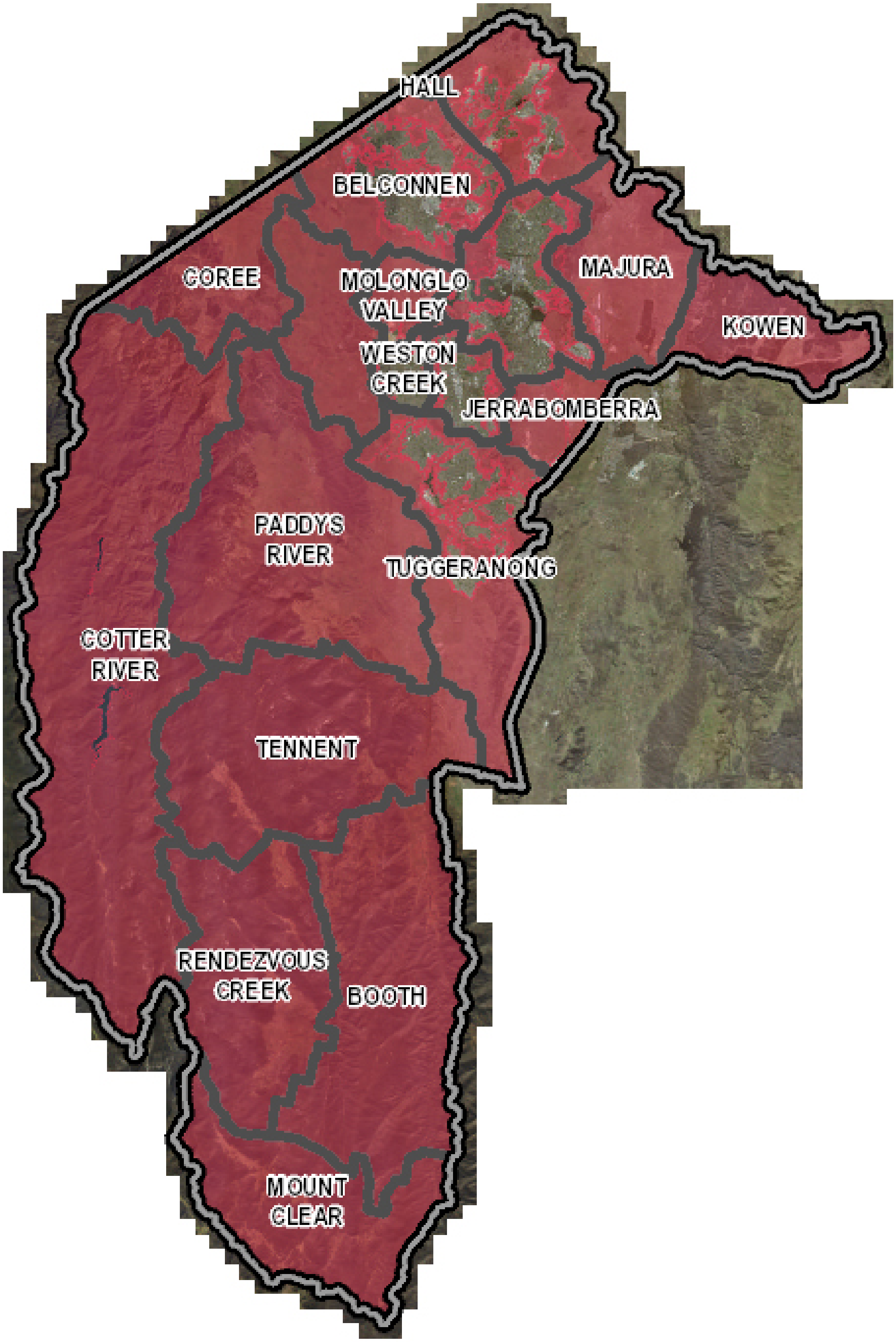


Figure 5: Map of bushfire prone areas from the ACT Emergency Services Agency.

Report

Compare your fire risk map to the ACT Emergency Services Agency's map of bushfire prone areas (BPA), which is shown in Figure 5 below.

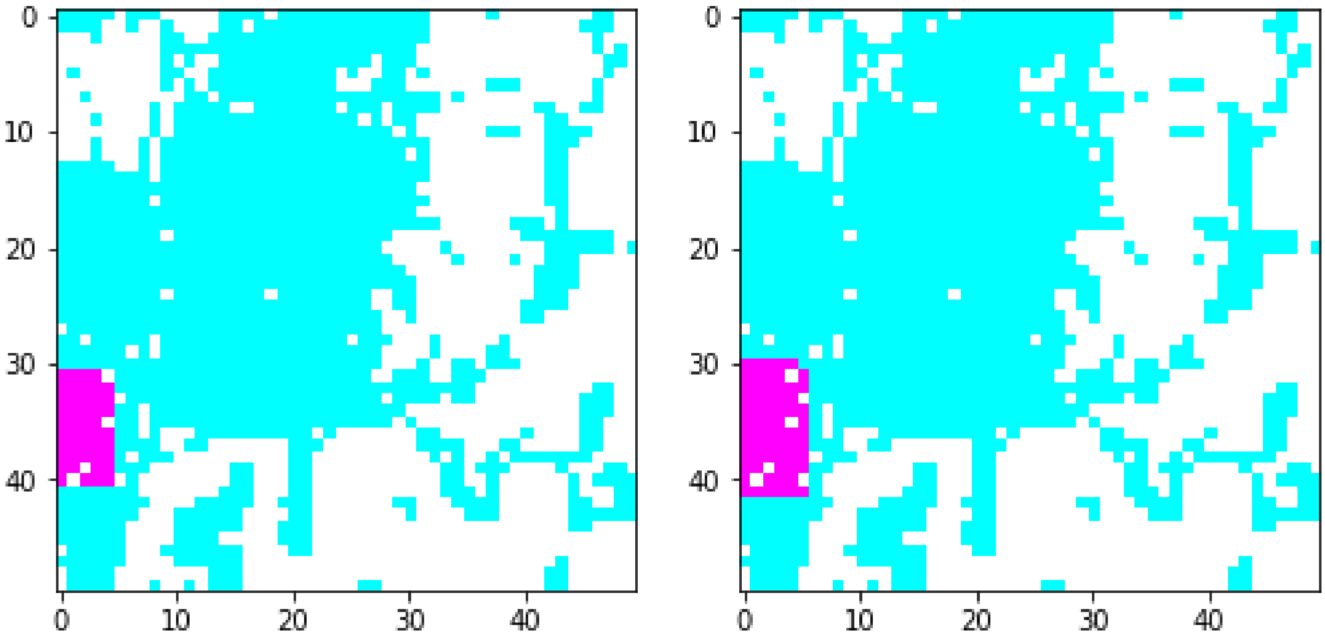
In your report, you should answer the following questions:

a. Describe the similarities and differences between your fire risk map and the BPA map. Explain how your calculation causes these similarities and differences.
b. Also in your report, explain how you handled the calculation of fire risk at the edges of the map and why you chose to handle the edges in this way.

Question 5: Simulating the spread of bushfire

We will implement a simple simulation of the spread of a bushfire. We'll represent the map as a grid just like the bushfire data, so a list of lists of booleans. For example, if we have a 3×3 grid with fire in the centre-top, it would be represented by the list `[[False, True, False], [False, False, False], [False, False, False]]`.

In every step of the simulation, the fire spreads from all locations already affected by fire to all 8 neighbouring locations that aren't blank. For example, starting from the map on the left and simulating for one step should produce the map on the right:

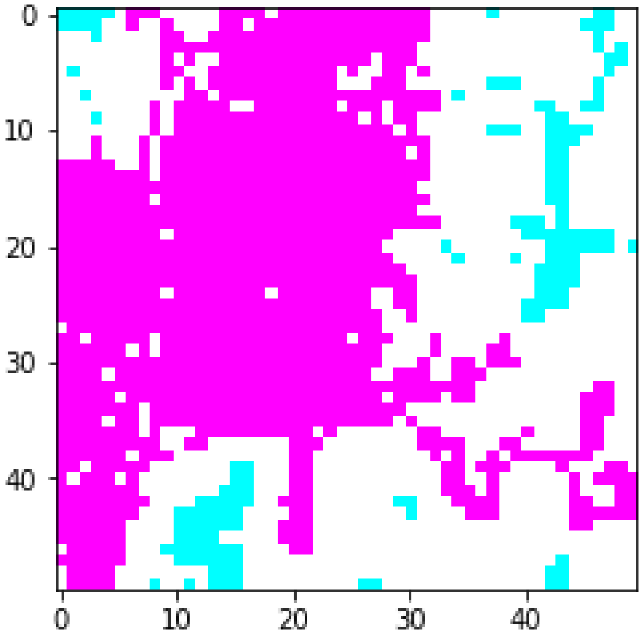


The bushfire has expanded by one pixel, all along the frontier. Eventually, if you repeat this over lots of steps, the fire will spread to all connected locations in the map.

Write a function `simulate_bushfire` that takes as arguments an initial map of fire affected areas (a list of lists from `load_bushfire`) and a number of steps (an integer). Simulate that number of steps and return the resulting bushfire map (again as another list of lists).

The assignment template contains a function for you to fill in:

```python
def simulate_bushfire(initial_bushfire, vegetation_type, vegetation_density, steps):
    pass
```



Report

We have provided you with a map of the initial conditions for the 2003 bushfire in `initial_2003_bushfire.csv`. Use your function to spread this bushfire until it covers approximately half the map.

In your report, answer the following questions:

a. Write down how many steps you simulated.
b. Include an image of the resulting bushfire (remember, you can use `show_bushfire` to make an image).
c. Compare the spread of fire produced by your simulation qualitatively to the provided map of the real 2003 bushfire. By comparing qualitatively, we mean that you should compare the two plots and summarise in your own words in what ways they are similar and in what ways they are different.

Question 6: Comparing to the 2003 bushfire

While we have previously compared our simulation to the real bushfire by eye, it would be nice to see how accurate we are quantitatively. One way to do this is to write a function that shows how much the result of the simulation overlaps with the real bushfire.

Write a function `compare_bushfires` that takes two bushfire maps (as lists of lists) and returns the percentage of cells that are the same (as a float). Ignore blank cells entirely.

For example, comparing the initial 2003 bushfire map with the final 2003 bushfire map should give approximately `0.435` on the `south` dataset, and `1.0` on the `anu` dataset.

There is a placeholder function for you to fill in in the assignment template:

```
def compare_bushfires(bushfire_a, bushfire_b):
    pass
```

Report

Use your function to compare your simulation from Question 5 to the real 2003 bushfire.

In your report:

a. Write down the accuracy, as measured by your implementation of `compare_bushfires`, of your simulation result from Question 5.
b. How good would you say your simulation is?

Question 7: Simulation with fire risk

The speed of bushfire spread depends on the vegetation density and type. In this question, we will extend our simulation based on the fire risk function we made in Question 4.

Write a function `simulate_bushfire_stochastic` that makes use of the `fire_risk` function to *stochastically* simulate fire spreading for a number of steps. *Stochastically* means that instead of fire necessarily spreading to nearby cells, it instead spreads with a chance based on the `fire_risk` of the cell. There is once again a placeholder function in the assignment template:

```
def simulate_bushfire_stochastic(
        initial_bushfire, steps,
        vegetation_type, vegetation_density,
        wind_speed):
    pass
```

This function should take the initial bushfire as a list of lists, the number of steps to simulate as an integer, and the vegetation type, density, and wind speed as lists of lists, and return a list of lists.

Your function should be *efficient*.

Report

Use your function to spread the initial bushfire (`initial_2003_bushfire.csv`) until it covers approximately half the map.

In your report:

a. Write down how many steps you simulated.
b. Show an image of the resulting bushfire (remember, you can use `show_bushfire` to make an image).
c. Compare the resulting map of fire affected areas to the real 2003 bushfire both qualitatively (by eye) and quantitatively (using `compare_bushfires`).
d. Is your simulation realistic?

Question 8 (COMP6730 only)

This question is for COMP6730 (i.e. masters students) only. COMP1730 students may attempt it if they want, but it will not be marked.

You must complete this question **individually** in a separate file, `question8.py`. You can assume it will be run from the same directory as `assignment.py` is located in.

Using your stochastic simulation, plot the number of cells on fire over time for each vegetation type. You should write a function `plot_fire_spread` that does this plotting. There is a placeholder in the file `question8_template.py`:

```
def plot_fire_spread(vegetation_type, vegetation_density, wind_speed):
    pass
```

This function should take three lists of lists and not return anything (just produce a plot).

Report

a. Include your plot(s) in your report.
b. What does the plot tell you about the rate of fire spreading?

You will be marked on how well your plot(s) address the question, i.e. the quality of your plots. Make sure you use include appropriate labels, colours, and so on.

Submission

- The assignment submission deadline is **11:55 PM on Sunday the 21st October**.
- Every group member must submit a zip folder containing `assignment.py` (the code) and `report.pdf` (the report). Students in COMP6730 (masters students) should additionally submit `question8.py`, with their code for Question 8.
- The complete zip file must be submitted through Wattle: https://wattlecourses.anu.edu.au/mod/assign/view.php?id=1427051.
- You can update your submission as many times as you like. However, we can only see, and will only mark, the latest submission that you made.
- If you are working in a group, the code that you submit should be identical for every member of the group.
- The report must be entirely your own work.
- No late submissions will be accepted without an extension being approved in advance.
- 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 documentation to support your case (for example, a medical certificate in case of illness). The course convener will then decide whether to grant an extension and inform you as soon as practical. Extensions can only be given to individuals, not to groups. If a member of a group is granted an extension, that applies only to that student's individual report.

Plagiarism

The only group work 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. Note that discuss does not include sharing documents, or drafts of documents. Reports will be considered under usual individual plagiarism rules. If you are unsure about what constitutes plagiarism, please read the ANU Academic Honesty Policy.

If you 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

- *ACT Vegetation Subformation*, ACTmapi.
- *2013 ACT Wind Map*, ACTmapi.
- *Bushfire Prone Areas*, ACT Emergency Services Agency.
- *2003 Bushfire (Affected Areas)*, ACT Environment and Sustainable Development Directorate.