University of Glasgow

A WORLD TOP 100 UNIVERSITY

Education Strategies in the AI-era for Statistics and Data Analytics

Dr Craig Alexander & Dr Jenn Gaskell

WORLD CHANGING GLASGOW

THE TIMES THE SUNDAY TIMES GOOD UNIVERSITY GUIDE 2024 SCOTTISH UNIVERSITY OF THE YEAR

Staff Usage

- Exam moderation

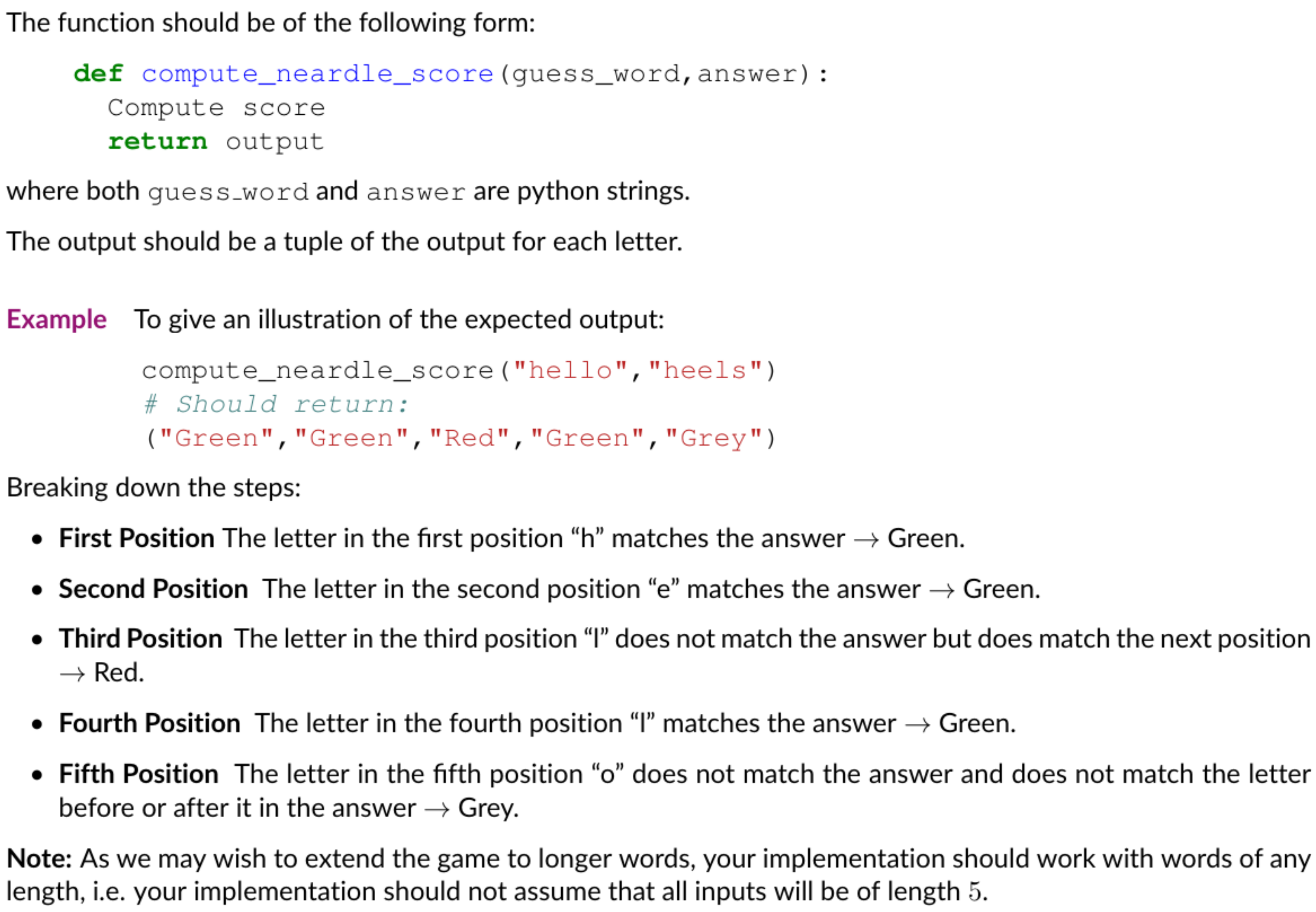
Cohorts

- Online MSc Statistics & Data Analytics
- In-person MSc Statistics & Data Analytics
- 3rd year undergraduates Statistics
- Analysing the cohort difference with our summer student

ODL Python Assessment

- Designed to test basic Python programming skills, such as writing a class and defining a function
- Students may be new to programming or have experience in R
- Take-home, open book assignments with a generous time window

ODL Python Assessment: Neardle



The function should be of the following form:

```
def compute_neardle_score(guess_word,answer):
  Compute score
  return output
```

where both `guess_word` and `answer` are python strings.

The output should be a tuple of the output for each letter.

Example To give an illustration of the expected output:

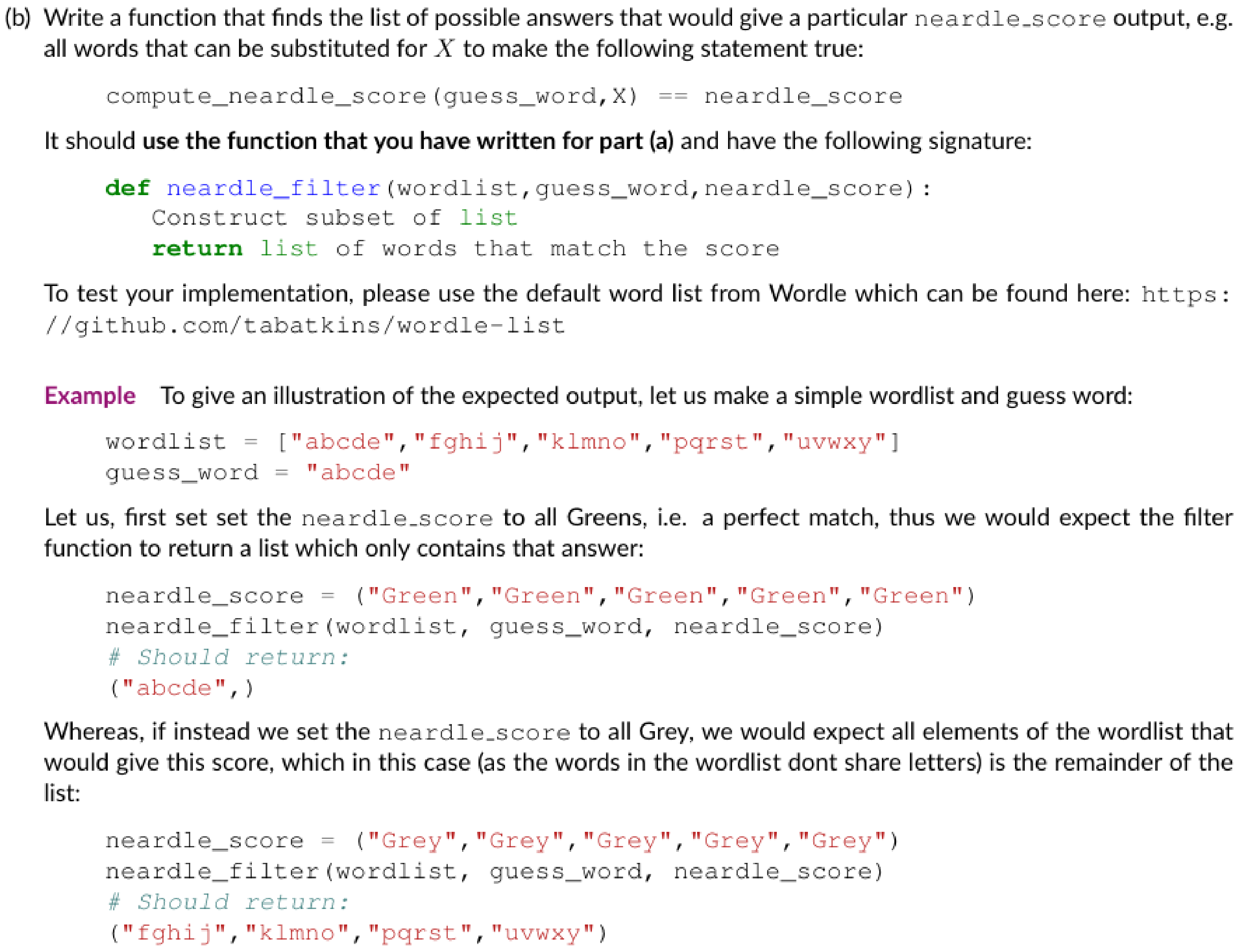
```
compute_neardle_score("hello","heels")
# Should return:
("Green","Green","Red","Green","Grey")
```

Breaking down the steps:

- **First Position** The letter in the first position “h” matches the answer → Green.
- **Second Position** The letter in the second position “e” matches the answer → Green.
- **Third Position** The letter in the third position “l” does not match the answer but does match the next position → Red.
- **Fourth Position** The letter in the fourth position “l” matches the answer → Green.
- **Fifth Position** The letter in the fifth position “o” does not match the answer and does not match the letter before or after it in the answer → Grey.

Note: As we may wish to extend the game to longer words, your implementation should work with words of any length, i.e. your implementation should not assume that all inputs will be of length 5.

ODL Python Assignment: Neardle



(b) Write a function that finds the list of possible answers that would give a particular `neardle_score` output, e.g. all words that can be substituted for X to make the following statement true:

```
compute_neardle_score(guess_word,X) == neardle_score
```

It should **use the function that you have written for part (a)** and have the following signature:

```
def neardle_filter(wordlist,guess_word,neardle_score):
   Construct subset of list
   return list of words that match the score
```

To test your implementation, please use the default word list from Wordle which can be found here: `https://github.com/tabatkins/wordle-list`

Example To give an illustration of the expected output, let us make a simple wordlist and guess word:

```
wordlist = ["abcde","fghij","klmno","pqrst","uvwxy"]
guess_word = "abcde"
```

Let us, first set set the `neardle_score` to all Greens, i.e. a perfect match, thus we would expect the filter function to return a list which only contains that answer:

```
neardle_score = ("Green","Green","Green","Green","Green")
neardle_filter(wordlist, guess_word, neardle_score)
# Should return:
("abcde",)
```

Whereas, if instead we set the `neardle_score` to all Grey, we would expect all elements of the wordlist that would give this score, which in this case (as the words in the wordlist dont share letters) is the remainder of the list:

```
neardle_score = ("Grey","Grey","Grey","Grey","Grey")
neardle_filter(wordlist, guess_word, neardle_score)
# Should return:
("fghij","klmno","pqrst","uvwxy")
```

Performance

- Excellent code
- Well-written
- Neater than the lecturer!

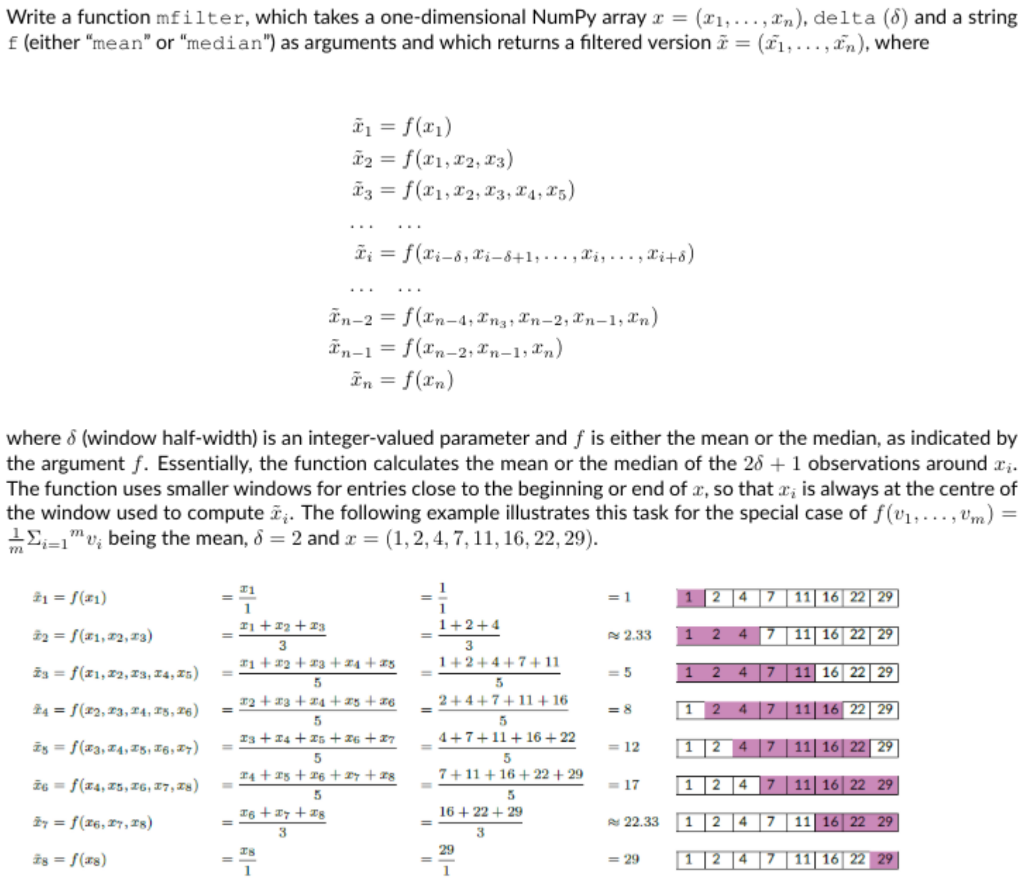
Problems & Opportunites

- Can solve the old assignments!
- Opportunity to update our assessments and learning outcomes to match the modern working world

What to do?

- Still want to teach basics of programming/statistics/mathematics
- Also need to ensure we teach skills relevant to today’s workplace

Other Question Performance

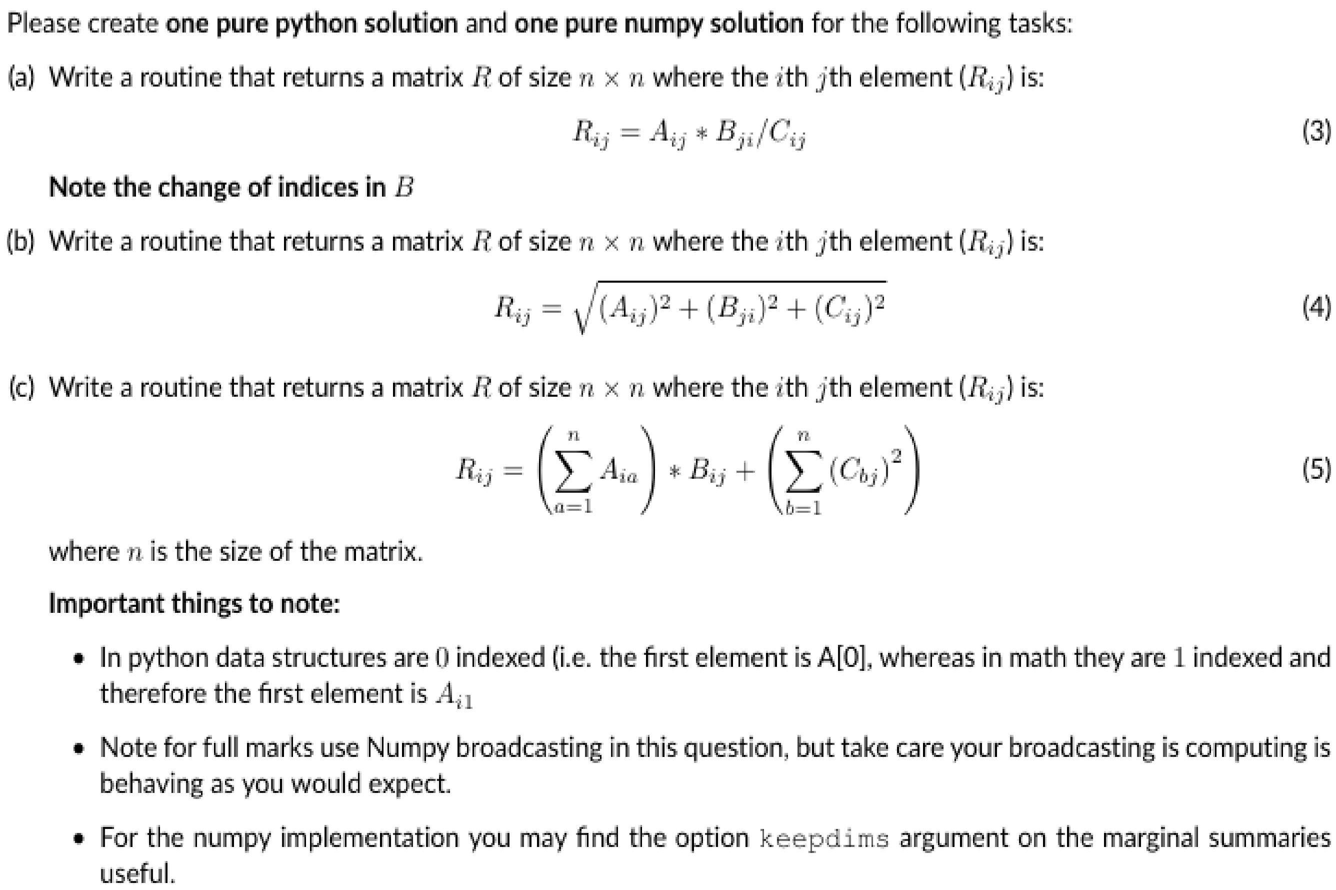


Write a function `mfilter`, which takes a one-dimensional NumPy array $x = (x_1, \dots, x_n)$, `delta` (δ) and a string `f` (either "`mean`" or "`median`") as arguments and which returns a filtered version $\tilde{x} = (\tilde{x_1}, \dots, \tilde{x_n})$, where

$$\begin{aligned}
\tilde{x}_1 &= f(x_1)\\
\tilde{x}_2 &= f(x_1, x_2, x_3)\\
\tilde{x}_3 &= f(x_1, x_2, x_3, x_4, x_5)\\
\dots &\quad \dots\\
\tilde{x}_i &= f(x_{i-\delta}, x_{i-\delta+1}, \dots, x_i, \dots, x_{i+\delta})\\
\dots &\quad \dots\\
\tilde{x}_{n-2} &= f(x_{n-4}, x_{n_3}, x_{n-2}, x_{n-1}, x_n)\\
\tilde{x}_{n-1} &= f(x_{n-2}, x_{n-1}, x_n)\\
\tilde{x}_n &= f(x_n)
\end{aligned}$$

where δ (window half-width) is an integer-valued parameter and f is either the mean or the median, as indicated by the argument f. Essentially, the function calculates the mean or the median of the $2\delta + 1$ observations around x_i. The function uses smaller windows for entries close to the beginning or end of x, so that x_i is always at the centre of the window used to compute $\tilde{x}_i$. The following example illustrates this task for the special case of $f(v_1, \dots, v_m) = \frac{1}{m}\Sigma_{i=1}{}^m v_i$ being the mean, $\delta = 2$ and $x = (1, 2, 4, 7, 11, 16, 22, 29)$.

$\tilde{x}_1 = f(x_1)$	$= \frac{x_1}{1}$	$= \frac{1}{1}$	$= 1$	1	2	4	7	11	16	22	29
$\tilde{x}_2 = f(x_1, x_2, x_3)$	$= \frac{x_1 + x_2 + x_3}{3}$	$= \frac{1+2+4}{3}$	≈ 2.33	1	2	4	7	11	16	22	29
$\tilde{x}_3 = f(x_1, x_2, x_3, x_4, x_5)$	$= \frac{x_1 + x_2 + x_3 + x_4 + x_5}{5}$	$= \frac{1+2+4+7+11}{5}$	$= 5$	1	2	4	7	11	16	22	29
$\tilde{x}_4 = f(x_2, x_3, x_4, x_5, x_6)$	$= \frac{x_2 + x_3 + x_4 + x_5 + x_6}{5}$	$= \frac{2+4+7+11+16}{5}$	$= 8$	1	2	4	7	11	16	22	29
$\tilde{x}_5 = f(x_3, x_4, x_5, x_6, x_7)$	$= \frac{x_3 + x_4 + x_5 + x_6 + x_7}{5}$	$= \frac{4+7+11+16+22}{5}$	$= 12$	1	2	4	7	11	16	22	29
$\tilde{x}_6 = f(x_4, x_5, x_6, x_7, x_8)$	$= \frac{x_4 + x_5 + x_6 + x_7 + x_8}{5}$	$= \frac{7+11+16+22+29}{5}$	$= 17$	1	2	4	7	11	16	22	29
$\tilde{x}_7 = f(x_6, x_7, x_8)$	$= \frac{x_6 + x_7 + x_8}{3}$	$= \frac{16+22+29}{3}$	≈ 22.33	1	2	4	7	11	16	22	29
$\tilde{x}_8 = f(x_8)$	$= \frac{x_8}{1}$	$= \frac{29}{1}$	$= 29$	1	2	4	7	11	16	22	29



Please create **one pure python solution** and **one pure numpy solution** for the following tasks:

(a) Write a routine that returns a matrix R of size $n \times n$ where the ith jth element (R_{ij}) is:

$$R_{ij} = A_{ij} * B_{ji} / C_{ij} \tag{3}$$

Note the change of indices in B

(b) Write a routine that returns a matrix R of size $n \times n$ where the ith jth element (R_{ij}) is:

$$R_{ij} = \sqrt{(A_{ij})^2 + (B_{ji})^2 + (C_{ij})^2} \tag{4}$$

(c) Write a routine that returns a matrix R of size $n \times n$ where the ith jth element (R_{ij}) is:

$$R_{ij} = \left(\sum_{a=1}^{n} A_{ia}\right) * B_{ij} + \left(\sum_{b=1}^{n} (C_{bj})^2\right) \tag{5}$$

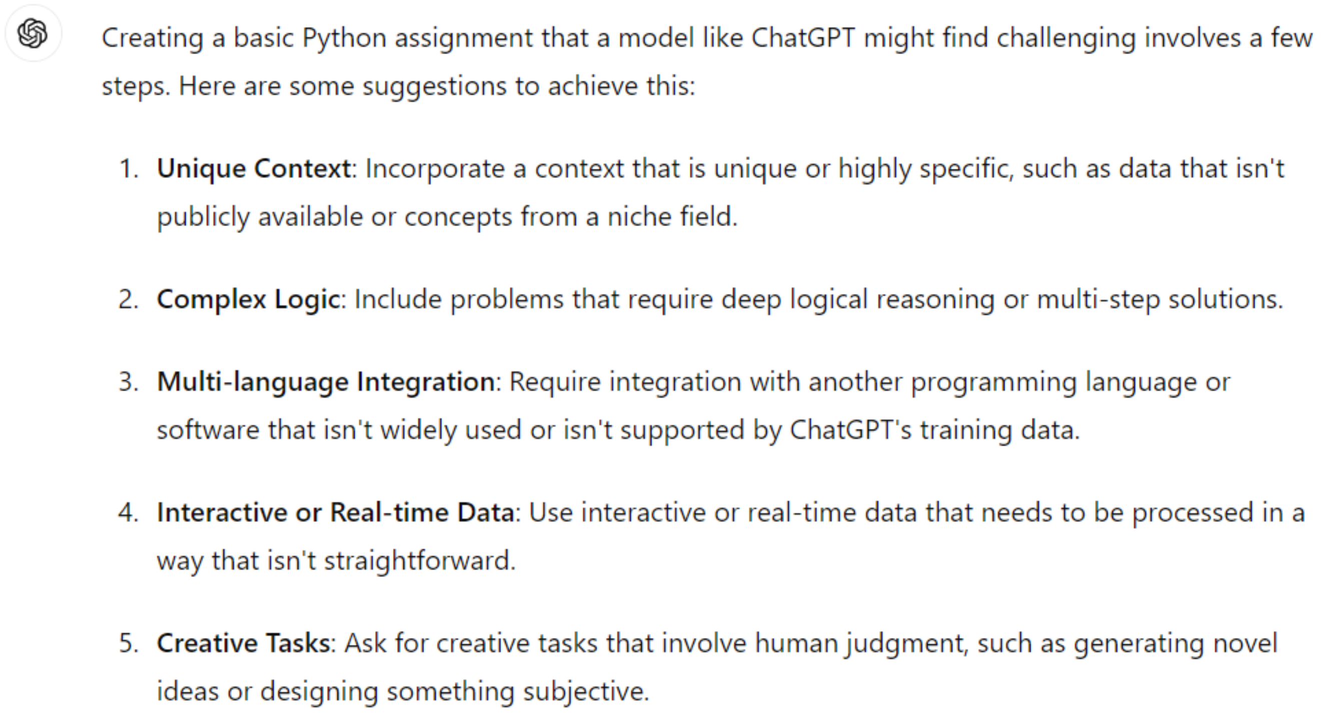
where n is the size of the matrix.

Important things to note:

- In python data structures are 0 indexed (i.e. the first element is A[0], whereas in math they are 1 indexed and therefore the first element is A_{i1}
- Note for full marks use Numpy broadcasting in this question, but take care your broadcasting is computing is behaving as you would expect.
- For the numpy implementation you may find the option `keepdims` argument on the marginal summaries useful.

What to do?

- Make question longer – not ideal
- Make uncopiable – not good for accessibility
- Ask ChatGPT:
- Integrate GenAI into assessment collaborate



Creating a basic Python assignment that a model like ChatGPT might find challenging involves a few steps. Here are some suggestions to achieve this:

1. **Unique Context**: Incorporate a context that is unique or highly specific, such as data that isn't publicly available or concepts from a niche field.
2. **Complex Logic**: Include problems that require deep logical reasoning or multi-step solutions.
3. **Multi-language Integration**: Require integration with another programming language or software that isn't widely used or isn't supported by ChatGPT's training data.
4. **Interactive or Real-time Data**: Use interactive or real-time data that needs to be processed in a way that isn't straightforward.
5. **Creative Tasks**: Ask for creative tasks that involve human judgment, such as generating novel ideas or designing something subjective.

ODL Python Assignment: Buffalo

- Created a new Python library, with many functions and files, hosted on GitHub
- Hosted documentation on a separate website
- Writing a class, subclass and plotting

- https://uofganalytics.github.io/PythonODLAssessInternal/API.html
- https://github.com/UofGAnalytics/PythonODLAssessInternal/tree/main/bufflib

University of Glasgow

ODL Python Assignment: Buffalo

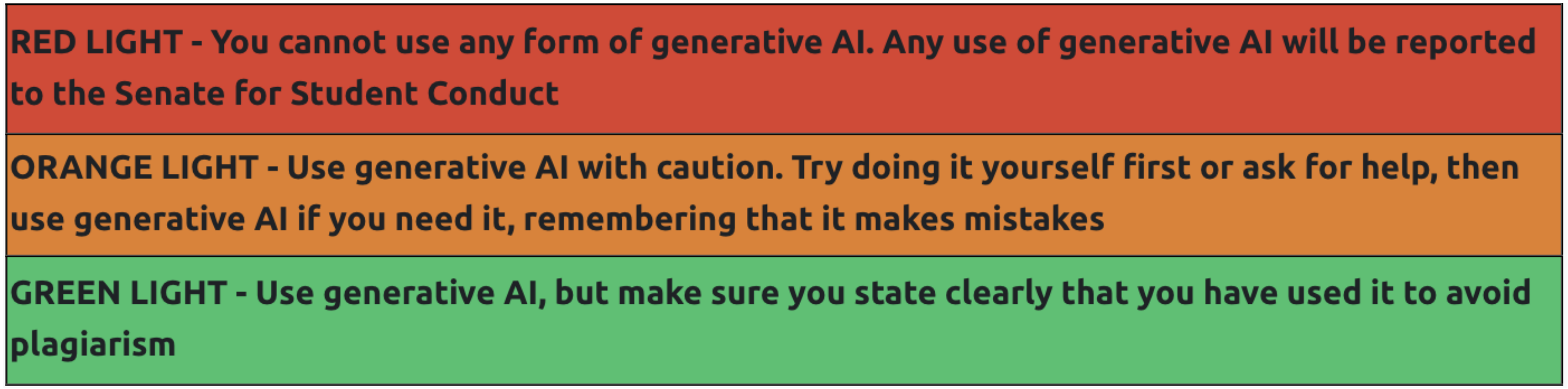
- Higher start-up cost
- Similar grades to previous years
- Students learnt new skills: GitHub integration, using external libraries, reading docs
- Still tested basic Python programming
- Genuinely written by students

The Synoptic Project

- Merged two assignments into a larger project
- Formative peer feedback at half way, with students marking the feedback they receive
- Future: more interaction with external libraries and datasets
- Future: interaction with AI – collaborative assessments

Maintaining Academic Integrity

- Adapt our assessments to be more genAI ‘proof’
  - but it takes time!
- In Statistics, we have introduced a traffic light system (inspired by Michael Grove) to highlight to students when usage is appropriate



RED LIGHT - You cannot use any form of generative AI. Any use of generative AI will be reported to the Senate for Student Conduct

ORANGE LIGHT - Use generative AI with caution. Try doing it yourself first or ask for help, then use generative AI if you need it, remembering that it makes mistakes

GREEN LIGHT - Use generative AI, but make sure you state clearly that you have used it to avoid plagiarism

Creating a Responsible Usage Course

- Our team are looking to design a framework and course which can help people self-evaluate where and how they use GenAI.

Vinny Davies

Jake Lever

Craig Alexandar

Jenn Gaskell

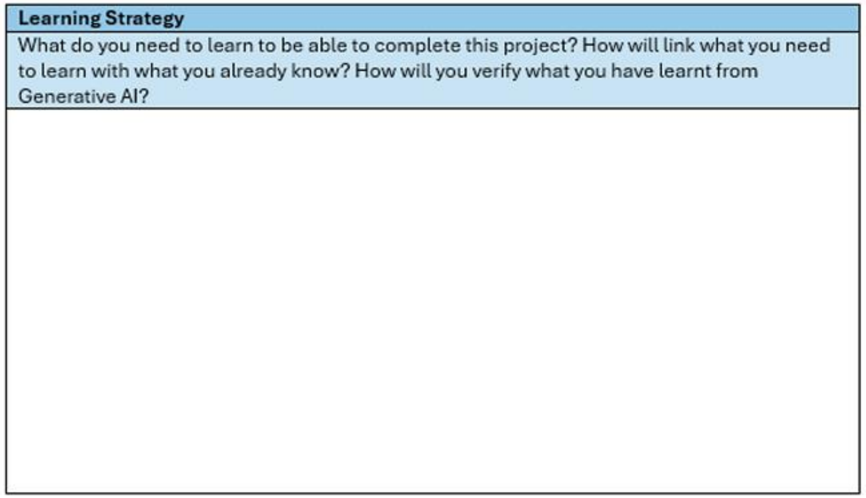
University of Glasgow

Course Structure

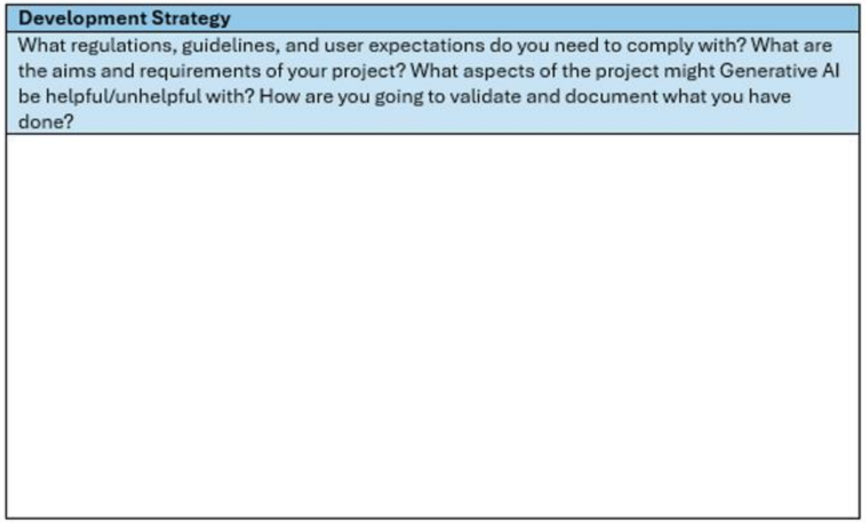
- The course is structured around four key themes:
  - **Understanding** – Understanding the basics of LLMS and the strengths/weaknesses of GenAI for your project.
  - **Development** – How to use GenAI to learn and develop models & code.
  - **Reporting** – Considering when and where you should document your GenAI usage.
  - **Safety** – Thinking about things to consider when setting guidelines within your workplace.

Engagement - Worksheet

- Using a worksheet to encourage critical thinking and evaluation.
- People can consider their own tasks and evaluate these against our four key themes.
- Allows us to provide things to consider over definitive solutions.



Learning Strategy
What do you need to learn to be able to complete this project? How will link what you need to learn with what you already know? How will you verify what you have learnt from Generative AI?



Development Strategy
What regulations, guidelines, and user expectations do you need to comply with? What are the aims and requirements of your project? What aspects of the project might Generative AI be helpful/unhelpful with? How are you going to validate and document what you have done?

Course Plans

- We are aiming to explore running the course in several different ways
  - **Students** – Using materials within courses – currently reviewing honours programme and can fit within skills based courses.
  - **Staff** – Training offered within school and beyond. Provide seminars to other institutions.
  - **Industry** – Workshops and seminar sessions.
  - **MOOC** – Funding to create a Coursera MOOC, taking elements of the course with guided examples.

Thanks for Listening

Let us know if you have any questions

Or feel free to reach out at:

Jennifer.Gaskell@Glasgow.ac.uk

Craig.Alexander2@Glasgow.ac.uk