Computing for Mathematics: Handout 7

This handout contains a summary of the topics covered and an activity to carry out prior or during your lab session.

At the end of the handout is a specific coursework like exercise.

For further practice you can do the exercises available at the sequences chapter of Python for Mathematics.

1 Summary

The purpose of this handout is to cover sequences which corresponds to the probability chapter of Python for Mathematics.

The main topic covered here is recursion.

2 Activity

We will be tackling the problem from the tutorial of the sequences chapter of Python for Mathematics.

A sequence $a_1, a_2, a_3, ...$ is defined by:

$$\begin{cases} a_1 = k, \\ a_{n+1} = 2a_n - 7, n \geq 1, \end{cases}$$

where k is a constant.

1. Write down an expression for a_2 in terms of k.
2. Show that $a_3 = 4k - 21$
3. Given that $\sum_{r=1}^{4} a_r = 43$ find the value of k.

There are instructions for how to do all of this is in the probability chapter of Python for Mathematics.

1. Define a python function `generate_a` which uses recursion to give the values of the sequence a_n.
2. Use a symbolic variable for k to obtain a_1, a_2, a_3 and a_4.
3. Obtain the sum of these four values to get an equation for k.

3 Coursework like exercise

Consider this recursive definition for the sequence a_n:

$$a_n = \begin{cases} c & \text{if n = 1} \\ 3a_{n-1} + \frac{c}{n} \end{cases}$$

1. Output the sum of the 15 terms.
2. Given that $c = 2$ output $\frac{df}{dx}$ where:
$$f(x) = a_1 + a_2 x + a_3 x^2 + a_4 x^3$$
3. Given that $c = 2$ output $\int f(x)dx$

4 Summary examples

Define the following sequence:

$$a_n = \begin{cases} 1 & \text{if } n = 1 \\ \frac{1}{a_{n-1}+1} & \text{otherwise} \end{cases}$$

```python
def generate_a(n):
    """
    Generate the sequence a_n using recursion
    """
    if n == 1:
        return 1
    return 1 / (generate_a(n - 1) + 1)
```