5CCP211C-Introduction to numerical modelling,
Physics Dept., King's College London

Introduction into Python

Muhammad Haider, Zhi Jun Cheung

January 17, 2024

Date performed .... October 2, 2018
Group 1
Pair .... Pair#2

1 Introduction

1.1 Functions and Classes

In Python, functions are reusable constructors, organized as a block structure when written by the user. A different type of function can be used when importing modules, which are usually pre-installed. These functions are ones that are commonly used, for example *numpy*, which contains many mathematical functions.

A class is a coding template used to create objects. Objects have member variables that as associated to the object, such as age and weight for a dog. An object is created using the class as a constructor. The object created by the class is known as an instance.

1.2 Modules for Linear Algebra and Graph Plotting

The two modules used in this code for linear algebra and graph plotting were *numpy* and *matplotlib*.

Numpy is a fundamental package for scientific computing with Python and is very useful for linear algebra, random number generators, Fourier transforms as well as analysing N dimensional array objects.

Matplotlib is a 2D plotting library for Python that can generate different types of plots. It is used for plotting a histogram.

1.3 Structure of Code

1.3.1 Arithmetic average, data read and histogram

The first part of the exercise required the implementation of a code which would read a data file containing three columns of information. The first column comprised of single characters whilst the other two contained numbers. The average values of the second and the third columns would need to be found. Lastly, a histogram would need to be generated with the numbers from the third column.

Firstly, the supplied data file would need to be read and classified as something which can be used in the code later. Since the file was already neatly organised and separated by regular intervals, this was easily achieved by using the *genfromtxt* function through the numpy module. This function allows numerical data to be converted into an array which can be easily read and manipulated. The code starts by specifying what the names of the generated lists will be, these were sorted as the names, heights and months for the three columns in the data, respectively. An important feature in this function is the *delimter*. This corresponds to the string used to separate the variables, which were spaces in the file supplied. Another important part of this function was the *dtype*, this specifies the type of data that the resulting array will be converted to. For this code, the data type was chosen to be a string. Lastly the array is transposed by using the *.T* function. This was done so the subsequent code would work.

The elements in the height and months array were converted into a *float*, this was to ensure that when the mean was calculated, it would be a decimal. The transposed forms of the array meant that the function *astype* could be used to achieve this. Now that the data supplied was converted into an array, the calculation of an arithmetic mean would be trivial. This was achieved by using the *mean* function, also from the numpy module. The two values of the averages were then printed allowing for verification.

The histogram was generated by using the *hist*function from the *matplotlib* module. Since the histogram was to be made using only the third data column, the 'months' array was inputted. The number of *bins* was also selected so that all data would be visible on the histogram.

1.3.2 Birthday Problem

The probability that two people share a birthday out of a group of thirty is 70.6%. A Python code is written to prove this by using a numerical method based on iterations.

The initial conception how the code should be build is based on three main processes. The first one, a function, should generate a list of thirty random birthdays as numerical values between 1 and 365. The second function should check for duplicates and the third will execute both functions while iterating 10000 times. The final result should provide the number of sets of birthdays that contains two or more same birthdays.

The first part of the code uses the a function from the *random* module to generate a random integer in between 1 and 365. This number is added to an empty list. Looping over thirty times provides a list of randomly generated integers between 1 and 365. Running the function returns this list.

The second checks whether the list contains duplicate values or not. The initial idea was to sort the list in order and compare check whether each element was equal to its subsequent element by indexing. However, a better solution was found when looking on the internet.

The logic behind this solution is to use the *set*() function, which returns a collection with no duplicate elements. An *if else* statement is used to check whether the number of elements in the set is smaller than the original list. Since the problem requires counting the number of lists of birthdays that contain duplicates, a value of 0 or 1 is returned depending on whether there are duplicates or not.

The final part of the code iterates through both functions 10000 times, each time generating a set of birthdays and checking for duplicates. Each time both functions are ran, a value of 0 or 1 is returned and added to the total number of lists that contain duplicate birthdays. After looping, the code prints the result as a fraction out of 10000 and as a percentage.

2 Discussion of Data

2.1 Birthday Problem

The birthday problem showed different ways of developing a solution, especially with regard to the method of checking for duplicates. A standard, more common approach for this problem would be to iterate through the list. The code in this case provided a simpler, more elegant solution. However, it contains a limitation in that it does not check for multiple duplicate birthdays and may not work in the case where the birthday problem may be extended. For example, the probability of 2 pairs of people who share 2 different birthdays.