Dalton Murray

Python for Data Analysis and Visualization

Professor Shams

January 19, 2024

**Assignment #1**

\*Please note that I include comments when initially writing the code as this helps me think through the process in my head, so these comments will be in pictures here and in the final submission of the .py file, I do not think I leave much out of the comments and that they are pretty thorough initially so some of the content here may be repeated on the .py file, however, I will try to explain it in a different way here.

With all of my code, I also follow a standard which I set for myself and enjoy using. This standard is using functions for all of the code. Given this code doesn’t need to be in functions and they haven’t been taught in class I have decided to put the code in one main function. My standard code start is as follows:

A computer screen with white text

Description automatically generated

What this is doing is importing the sys library which gives some extra system functions, such as sys.exit() which allows a smooth exiting of the Python program.

Next, I create a function called \_\_main\_\_ which is where I will put all of my base code into for these tasks. Then, outside of this function, I have an if check which checks if \_\_name\_\_ is equal to “\_\_main\_\_”, which, of course, it is. It then wraps calling the main function into a sys.exit() function which will start the main function and then once the main function is done it will exit. In this picture I have pass in \_\_main\_\_ because it allows the code to be valid and not be an error, essentially it’s a placeholder telling it to ignore the function in this case.

**Task 1: Extracting Top 3 Values**

Write a Python program that extracts the three largest values from an integer list and prints these values in descending order.

We were initially given the following as a base code to adapt to extract the top 3 values instead of the top 2 values:

A computer screen shot of numbers and letters

Description automatically generated

Because of it being taught this way in class and being given this as a base code I decided to complete this task with two different methods, the first method by following the base code and the second method in a more simple and efficient way.

Given that we have a base set of numbers I was initially under the impression that in-class we were to design our initial solution with the base set in numbers in mind which is why I originally had a variable set maxNum = 0 but I was then told that this won’t work because if there are negative numbers, so this changed my impression that we were supposed to be under the assumption that we didn’t know the base set of numbers (for example, taking in a user supplied list of inputs.

Continuing the base supplied code and adapting it to list the top 3 rather than top 2 numbers is incredibly easy as we were already given everything we needed to do so, however, I will also explain the base code rather than just my added code.

A screenshot of a computer code

Description automatically generated

This first block of code is primarily comments, however, all it is doing is defining and setting a variable called nums to be a list with the contents of 4, 3, 2, 78, 32, 12, 11, 10, 7.8. I have other comments in this block of code that define and set the variable nums, this is what I used to validate my method of ensuring that it will pick the top 3 regardless of if the numbers are all negative or a mixed set of numbers as I don’t want to assume to know the base set of the list of nums. I then put comments on the lines of each set of the nums which show the outputs I get for running the code which helps ensure accuracy for me.

After this block I have this:

A black screen with yellow and green text

Description automatically generated

This is defining and setting the initial 3 maxNum variables to specific indexs of the list nums, and in the case of our supplied base list is 4, 3, 2. This is important for keeping track of the highest number and is what the variables for the final output will be.

This next block of code will be our base block of code we were supplied for getting the actual top 2 numbers so I can explain it before my correction to top 3:

A black background with white text and numbers

Description automatically generated

What this is doing is creating a for loop that is incrementing based on i which is in the range of the starting position of 2, and the stopping position, which is the length of the list nums, and with the base supplied numbers is equal to 9. In some cases, this starting position should change, however, if we are trying to get the top 3 numbers it doesn’t necessarily need to change. I would actually change the starting position to 1 as it may work better in more scenarios than setting it to 2, however, there should be not much of an actual difference, which will be shown by my validation later on.

As this for loop goes through the list of numbers it will run this if check everytime it runs, it is asking if the current list item is greater than the maxNum1 variable and if it is to then set maxNum2 equal to maxNum1 and then set maxNum1 to the current incremented number. It is in this order of 2 and then 1 because if it was 1 and then 2 maxNum2 would not be getting set to the second highest number and would be equal to the first highest number. Continuing this if check for each iteration of the for loop if the current iterated item of nums is not greater than maxNum1 we check if the current iterated item is greater than maxNum2 and if it is we set maxNum2 equal to the current iterated nums item.

In order to make sure this be the top 3 I just need to expand the code a little further. So, instead of the base for loop I made the following:

A screen shot of a computer program

Description automatically generated

A black screen with yellow text

Description automatically generated

This takes the base for loop, keeping the starting position of 2 (which, as said previously, will still work perfectly even with looking for top 3, and setting it to 3 actually fails my validation test of mixed nums, ideally what the starting position should actually be set to is 1 as this will pass every validation test and work correctly and has the following outputs:



Base nums:

A black background with white text

Description automatically generated

All negative nums:

A black background with white text

Description automatically generated

Mixed nums:

A black background with white text

Description automatically generated

And in order to prove that setting it to 3 breaks with my mixed nums set here is the output of it:

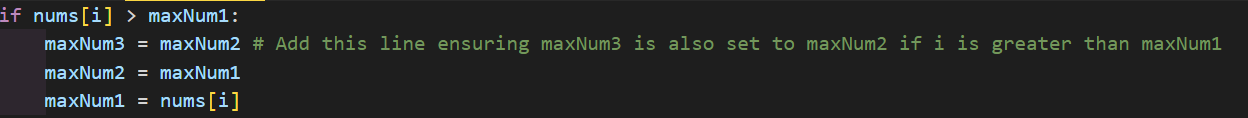
A close up of a black background

Description automatically generated

A black background with white text

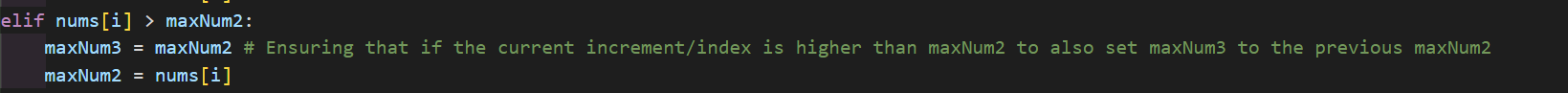
Description automatically generated

Which is not correct, setting it to 1 and 2 however work.)

Continuing looking at my for loop, what I have done is taken the additional maxNum3 variable I set and added it before setting maxNum2 in the if check:

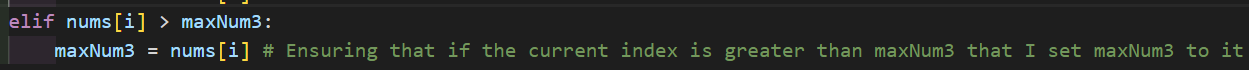
This now allows for in each for loop if the current increment of the item is greater than maxNum1 to set maxNum3 to 2, 2 to 1, and 1 to the current item, if it were in the reverse order it would not be setting the top 3 numbers correctly.

Following this, I added to the first elif:



Now if the current increment of I is not greater than 1 and if the current increment is greater than 2 we set maxNum3 to maxNum2 and then set maxNum2 to the current incremented item.

After this elif I added another one for 3, following the same format:



This now, if failing the previous if and elif checks if the current incremented item is greater than maxNum3 and if it is to set maxNum3 to the current incremented item in the list. Both of the elif’s are needed so that I can make sure that the maxNum is being properly set, for example, without the second elif the output is:

A black background with white text

Description automatically generated

This is incorrect as the third highest number is 12 and not 4.  
 Finally, my last change to the base code is the following:



I personally prefer not using print(f””) and formatting it using the standard print statement. This print statement prints out on the first line the text Max Num 1: and then it has a \n, the \ escapes the literal string printout and the n tells it to print a new line, then I have it print maxNum1, and then I continue with formatting new lines and the maxNum’s, and is in descending order because when I think of a list of the top 3 numbers I think the first highest then second and then the third.

Moving on to my second method of printing out the top 3 numbers is a very simple, easy to use and efficient 1 line solution.

A black screen with green and yellow text

Description automatically generated

I still have the base nums and my random other negative and mixed nums validation lists:

A screenshot of a computer code

Description automatically generated

After this I have the following:



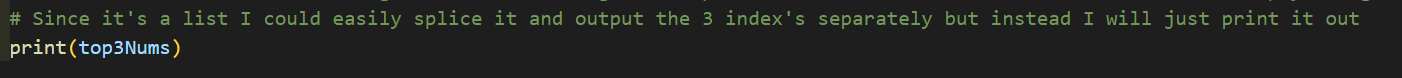


This one line solves everything the previous if’s and elif’s solved, and is more easily configured.

What I am doing is setting the variable top3Nums to the sorted list of nums which is in reverse order. The normal order of the sorted function is lowest to highest, and although I could do it that way and it would still be just as simple, I prefer this way. Once it gets this list of sorted highest to lowest numbers it then splices, or cuts out the first 3 items, which are guaranteed to be the top 3 highest numbers. After this, it takes these 3 items and sets it to the variable. This can be easily changed by setting the [:3] to let’s say [:2] and getting the top 2, or [:10] and getting the top 10. I also don’t have to worry about in this case adding an elif to each nth top number I want and adding setting it equal to the previous one in the initial if statement.

Then, I simply print out this list:





I could have, if I chose to, splice this list into 3 separate variables and printed it out and formatted it like the first method, however, I wanted to instead just print it out, it is also possible to format it by not splicing it into 3 separate variables but just doing a loop through each item in the list and printing it out directly for each item.

**Task 2: Moving Average Calculation**

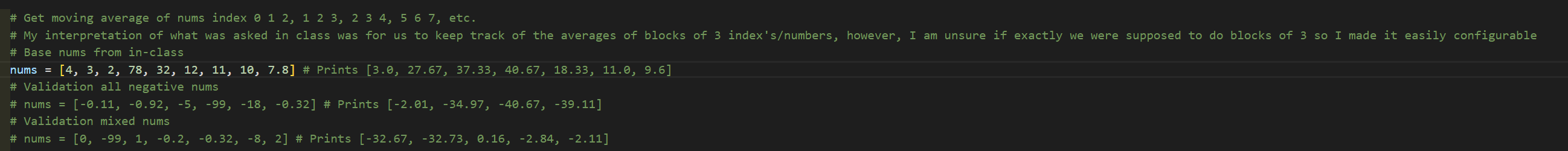
Write a Python program that generates a new float list representing the moving average of an integer list. Use a window size of 3 for the moving average.

My understanding of this question was a little fuzzy, I assumed that it means it wants a complete list of a moving block of 3 averages and for it to go through the whole list of numbers.

My overall code for this task was quite simple, however, I will still break it down and explain each part:

A black screen with green and yellow lines

Description automatically generated

I, of course have my base 3 sets of random numbers at the top, 1 provided in class, 1 fully negative and 1 mixed numbers:

I also included their outputs when running at the end, which I have confirmed are all accurate with a calculator:

A screen shot of a computer

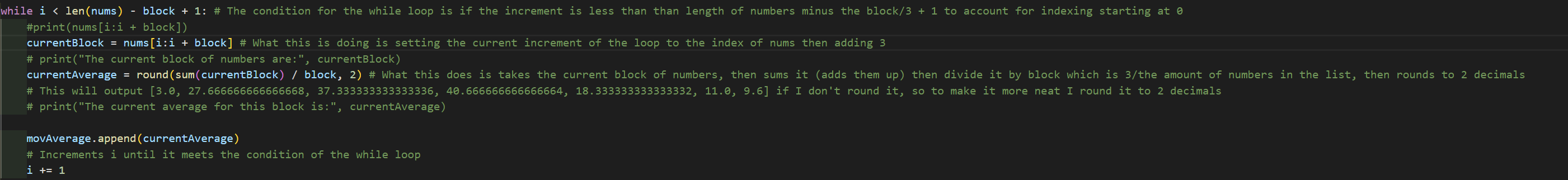
Description automatically generated

After the lists being set, I have 3 base variables being set, the block, or window which is 3, a moving average list, and setting i = 0 for incrementation.

A black background with white text and numbers

Description automatically generated

Next, I have the while block:



A computer screen shot of a code

Description automatically generated

This while block includes some additional testing information and some additional print statements for if I want to change how I am outputting the information.

To start the while loop I have the condition:

A black background with white text

Description automatically generated



This is a pretty basic condition, all that it is saying is that while the variable i is less than the length of the list nums subtracted the block variable, or 3, and + 1 to continue looping through the statements. I have it subtracting the block because it needs to stick to how many items are in the block, and I have the addition of 1 to take into account indexing of lists starting at 0.

In this while loop I have this print statement:



This outputs:

A screenshot of a computer screen

Description automatically generated

This allows me to verify that it is properly looping through the list with the set block and is starting and stopping at the correct positions.

Then, I have this:



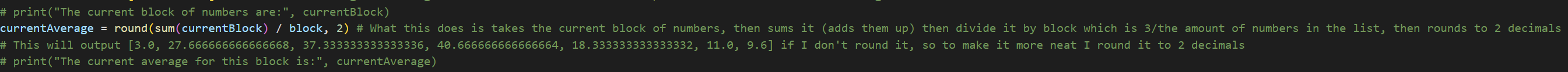


What this does is sets a new variable currentBlock, and it is setting it to the list nums and splicing to the current increment/variable i through the increment + block or 3, which is also the output of my previous print statement. Essentially, it takes the current incrementation number, then takes it again and adds 3 which is setting the block or the window, which gives the start and stop position of the block/window.

Next, this block of code:

A computer screen shot of numbers

Description automatically generated



I will explain the commented out prints after I explain the actual calculation. I define a new variable now, currentAverage, which is taking the previously set currentBlock list of numbers and adding them together and then dividing it by the block, or 3, which is how to calculate an average. After this, this average is being rounded to the second decimal place for neatness and organization, which then gets set to the currentAverage.

Now, those two print statements, I have these commented out just because they’re not necessary but depending on how someone wants to see the output they could be useful. What this does is prints out the current block of numbers and then the average for the block, and then it repeats this for every time it loops through:

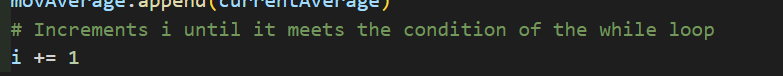
A screen shot of a computer

Description automatically generated

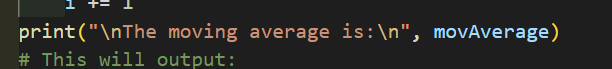
In this loop, I of course have to create a final list of all of the averages of the blocks:

A close up of a computer screen

Description automatically generated

This takes the movAverage I set before and appends onto it the currentAverage. I then at the end increment i by 1 until it meets the condition for the loop I set:  


I then have, outside of the loop:



Which is a simple print statement with new lines printing out the list of the averages of the blocks:

A screenshot of a computer

Description automatically generated

Should I choose to, I could then take this outputted list and splice it to different variables to then perform different functions with or perform it directly on the list.

I have neither given nor received unauthorized aid in completing this work, nor have I presented someone else's work as my own.

*Dalton Murray*