

Mahdeen Ahmed Khan Sameer

Professor Allen Harper

CS 152 A

14 February 2022

Abstract

Our first project uses Python Interference in Visual Studio code to create a data program. Our main goal is with the concept of "Expressions, Variables, and Functions" taught in the class; we will be plotting a graph using the ideas on the following physics formula $pf = pi + vi * t + 0.5at^2$. It is a model equation for finding the al positions of an object acted on by gravity. We will use the ballistic function throughout the project to print out the required data to make a trajectory curve.

Method

To evaluate the value of the given formula, $pf = pi + vi * t + 0.5at^2$, we used different arguments in the first case to print the direct pf. Here, pf is the final position of an object, pi is the initial position, vi is the initial velocity, and a is the acceleration acting on the object. We have assumed and as a constant will be influenced by the gravity 9.8 ms^{-2} .

Using that given equation, we defined function "ballistic1" with parameter t and printed out the value of the final position, pf with $t = 0.5$, $t = 1$. Here, we learned about the importance of return in Python. It stood as one of the confusing aspects for the students to understand.

After that, we used the more general ballistic function to define the new function "compute an output." It had four parameters. We later called out 10 trajectory points using this concept with the fourth argument as " $t_1 + N \cdot 0.1$ ". Here, we used our computational thinking as we used Python to manipulate the information into a few calls to get 10 points without calculating the trajectories separately. It made our task simple and less time-consuming. We called the function "trajectory100" to compute 100 points on a trajectory using the same concept. We used the arguments 1, 50, 10, 0. It gave us more than 100 points, and with that, we plotted the points on our "data.csv" file to plot the graph inside that.

Results

Our target of creating data using the given ballistic function to the trajectory was successful as we made our graph using excel. We got the excel file by redirecting the program's output into the "data.csv" file by using the terminal in the VS code.

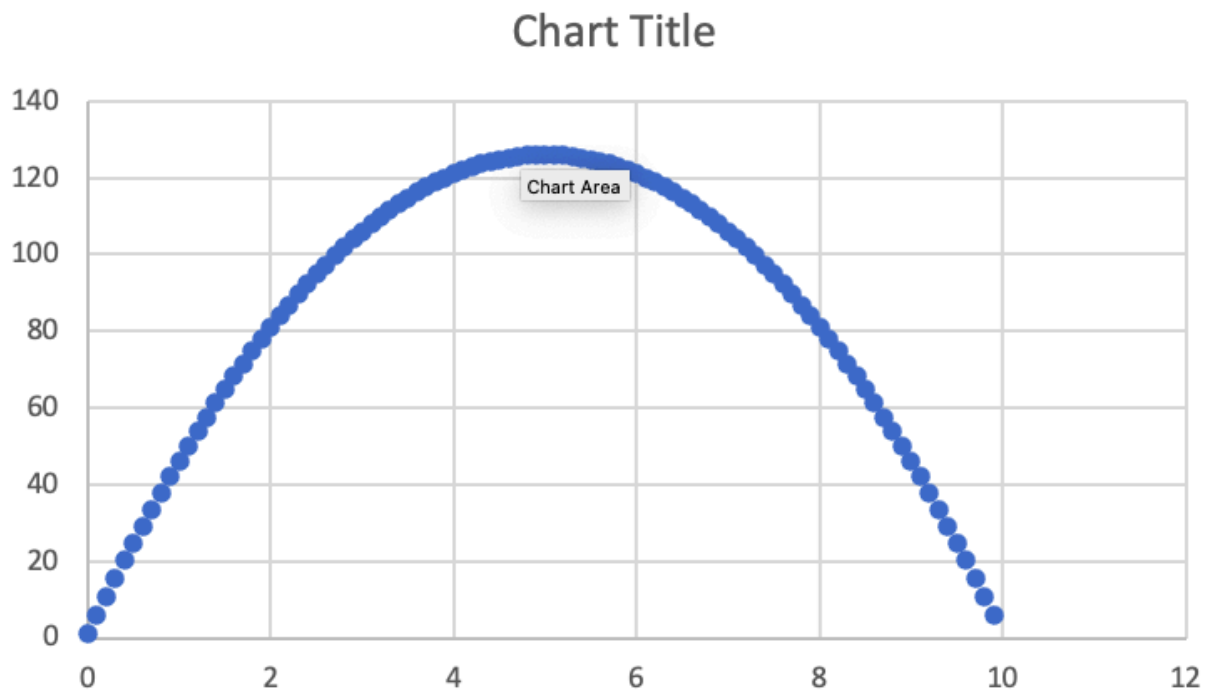


Figure: Graph plotted in data.csv file by using the 100 points of the trajectory. Here, the x-axis is 0 to 12, and the y-axis is 0 to 140.

The graph started with the point (0.0, 1.0) and ended in (9.9, 5.95). We had to extract some extra values from excel to make the graph more reasonable to express the trajectory plotting.

Besides, we also did the 10 points trajectories, which get directly included in the 100-point trajectory. So, for example, the final point of the "trajectory10" is (0.9, 6.85). This output means that our graph matches the equation of the given formula that makes our project successful.

Extensions

I tried to do the extensions given in the simplest way I know. For the first extension, I just followed the loop method to use the time steps by calling "for N in range (10) and then called the output. This loop method quickly demonstrates our seventh extension, which asks about loops. It also does our extension about writing 1000 codes instead of 100. It is easy. With this process, we do not need to write any additional code. We just need to manipulate trajectory 100 to 1000 points.

I imported the python math package for our second extension by putting "import math as np" and then printed the value from the equation, $y = \sqrt{x}$. Then, again, we imported matplotlib to plot a graph which is another extension.

In the end, I demonstrated the way to control a floating-point number. First, we took $a = 3.4588$ and then printed the function using `("%.2f"%a)` to limit the decimals.

Follow up Questions:

1. What is a variable?

We can call a variable a container or a block to store data and information. For example, if x contains a value, $x = 5$, then x is a variable because it has an assigned value. In this project, we got pf as a variable too. But a variable cannot start with a number or a unique character (*, %, #). So again, we cannot have commas in our variables. So we also call variable identifier.

Other examples: **_str**, **str**, **num**, **_num** are all valid names for the variables.

2. What is a function, and how are they useful?

A function is a group of codes to perform a specific action or task according to instructions. Its role is to take some inputs and give out some outputs according. We can call the inputs parameters. In our project, we had "def ballistic1(t): "where "ballistic1" was a function where "def" was defining the function. In the parenthesis, t was an input for the ballistic1 function, and that's why t is the given parameter. Operating a task saves a lot of time.

Other examples: def **Liza()**:

3. What is a parameter, and how do they affect the generality of a function?

The input we assign into the bracket of the functions is called parameters. A parameter is a variable given inside the parentheses in the function, which passes the reference value to the given object to which the variable refers (the value). By the way, it is not a variable itself. For example, in this project, we had "def ballistic1(t): "where "t" was a parameter assigned inside the function "ballistic1". Without well-defined parameters, we will not get the required values after calling the function.

4. What is the difference between a print statement and a return statement?

A return statement is used at the end of the function's execution to call the result (value) back to the caller. On the other hand, a print statement shows the value stored by a variable after executing the function.

Reflection

In our CS 152 course, we learned Python Programming by grasping some concepts like functions, variables, and Expressions. In addition, we were taught to learn small coding to enforce our computational thinking in a competitive, engaging environment. In this project, we interestingly approached our new skills as we had to find out the coding solutions by our reflections. Moreover, engaging with professors and teaching assistants made the experience more adventurous. Besides, I believe the concept I learned in this project will be my lifelong friend.

References

Going to TA and office hours helped me grasp concepts I was hesitant with. But it helped me understand concepts like loops, return statements, etc. I also had a problem loading the Python interference as I had my anaconda interference installed. Both were colliding or making a combined problem. It was solved by my friend Sam. Besides, I watched some random YouTube videos about executing random functions, which was fun to learn.

