**Pandas DataFrames and Functions**

[00:00:00.48] [MUSIC PLAYING]

[00:00:08.65] RYAN AHMED: Hello, everyone, and welcome to this lesson on Pandas with functions. In this lesson, we will learn how to apply a function along an axis in a Pandas DataFrame. Recall that functions are a block of code that performs a specific task. A function is executed with code. You can send the data, and you can get the results out of it.

[00:00:30.37] You can apply a function to specific elements in Pandas DataFrame using the apply function. You can choose to apply Python built-in functions such as sum, min, max, or use any custom function that you create. Here are the key learning objectives of this lesson-- define a Python function, apply a custom function along that axis in a Pandas DataFrame, apply Python built-in functions to Pandas DataFrame.

[00:00:58.59] So let's head over to our Jupyter Notebook and get started.

[00:01:01.59] [MUSIC PLAYING]

[00:01:09.16] All right. So right now, we are in the DataFrames with Functions Jupyter Notebook. And let me go ahead and first introduce you to Pandas DataFrames with Functions. So if you recall in previous couple of lessons, we learned how to define a function. And we used the DEF keyword to define a function. And we covered also what function was. Simply, a function is a block of code that can perform a specific task. You can send it data, and you can receive data out of it.

[00:01:44.89] What I wanted to do right now is I wanted to show you how we can apply a function to a specific column in a Pandas DataFrame. So Pandas DataFrame apply function comes in really handy to apply a specific function along an axis in the Pandas DataFrame. You can also use either built-in Python functions or your own custom-based functions as well.

[00:02:11.99] So let me show you an example. Let's assume that here I have portfolio\_update function. That's how you define a function. So you say DEF, define, portfolio, underscore, update. The function receives value x, and this function simply increases the portfolio of a given client by 10%. It takes x and multiplies it by 1.1. That's all it does.

[00:02:39.61] And please note that when you write this code, you just define this function. Nothing will happen because you haven't invoked that function yet. You haven't called it yet. And if you wanted to take this function and apply it to a specific column in a Pandas DataFrame, this is simply the syntax here. First, you need to specify the column that you want to apply the function to. Here, I'm going to pick portfolio size.

[00:03:04.38] And then you say .apply. You apply the apply function on it, and then you specify the name of the function. You open parentheses, and you specify the name of the function. And here, the name is portfolio\_update. And the function here is going to be applied to that specific column and is going to return the output. And the output here, I'm going to put it in a new column. Call it Updated Portfolio Size.

[00:03:33.85] So let me go ahead and show you how we can do that in code. This is simply the results that we're going to get out of it, so this is my-- let me zoom in a little bit. So here, I have the original raw Pandas DataFrame. This is the input column, the portfolio size. After I apply the portfolio update function and write the new data in the updated portfolio size column, this is simply what you get. So for example, $80,000 as an example here-- let me go up. So the $80,000 right now becomes $88,000 as an example. So now we increase the value by 10%.

[00:04:16.18] So let me show you how we can do that in code. First, I'm going to import Pandas, and I'm also going to import another library called NumPy. And NumPy is used for numerical analysis and arrays manipulation. And I'm just going to use a function within NP. It's going to be the np.squareroot, which is what I got here. And that's why I need to define it first.

[00:04:43.12] I'm going to come to that later on. Next, I'm going to use Pandas to read my data. So I'm going to say pd.read\_csv. I'm going to specify the name of the CSV file. That is going to be investors\_data. I'm going to put that data in investor\_dfpandasdataframe. Next, I'm going to apply a function to a specific column. So first, I need to define the function. This is the function definition. You say DEF. And please note that DEF is a reserved word, and it's highlighted in green.

[00:05:18.73] You give the function a name. Here I called it portfolio\_update. The function receives x. The function returns x times 1.1. That's all it does. It takes the input, increases it by 10%. If you press Shift-Enter, nothing will happen-- we learned that before-- because we didn't call the function yet. Now let's go ahead and call the function and apply it to a specific column in a Pandas DataFrame.

[00:05:43.90] So here, i'm going to grab my investor\_df. I'm going to grab my portfolio size column. And if I say .apply, you open parentheses, and here you specify the name of the function. That is going to be Portfolio Update. Basically, I'm going to apply this function to the Investor DF Portfolio Size column, and then I'm going to write the data in a new column titled Updated Portfolio Size. Press Shift-Enter. You want to check it out. Here we go.

[00:06:13.43] Now you will see there is a new column that has been added. And this is simply a 10% increase over the original value, which was 80,000 as an example. Now it becomes 88,000. 95, for example-- or 950,000, actually, becomes 1 million, approximately, and so on. What you could do as well is you can use the applied method to simply apply a built-in function in Python to a given column within a Pandas DataFrame.

[00:06:47.21] For example, here I'm going to grab my investor\_df. I'm going to grab the Age column. And let's assume that maybe I wanted to obtain the square root, for example. So if I say .apply-- and you don't need to define the square root function beforehand. You can just say NumPy or np.squareroot. This is already a function that has been written for you.

[00:07:10.58] And that is going to calculate the square root for you and place it in a new column titled Square Root Amount. If you push Shift-Enter, here we go. And you will notice that the square root amount has been captured in here. So I showed you how to use Python built-in functions, or you can also use your own custom-based functions as well.

[00:07:31.43] And that's it. That's all I have for this lesson. I hope you enjoyed it. In the next lesson, let's go ahead and cover our practice opportunity. And I'm going to show you a detailed video explanation of the practice opportunity solution. I hope you enjoyed it, and see you in the next lesson.

[00:07:47.29] [MUSIC PLAYING]