Data Manipulation Script

For Pandas Tables

Karthik Nadimpally

September 13, 2022

I. Setup

This is a pandas table manipulation that is designed by Karthik Nadimpally for basic table manipulation. In this chapter I am explaining how to setup the table manipulator and how to use it.

Begin by downloading the files and then opening the file. Go to the input folder and input the csv you would want to modify. Then go to the config.json folder, and change the file path to “Input/[YOUR FILE NAME].csv”. There are other settings involved, which will be covered within later chapters. Next, open up the terminal/console/command prompt, and navigate to the file directory using cd. For this part, you need to have python and pip installed. Now, enter “pip install -r requirements.txt”. This will download the required modules for the program to run. Finally using your code editor, open up the main.py file. Find the location within the file where there is hashtag spams, and that will be the only section of the code that the user will need to modify. I will explain the commands you can set in detail within the next couple of chapters

II. Quick Notes (IMPORTANT)

Overall, this script is designed to have an aspect of complexity involved, however I hope this manual is comprehensive on how to use the table manipulator. More versions will be released over time, with more and more features.

Remember to enter commands only in the areas that are marked with a bunch of hashtags. Be sure to use the examples below for help when entering commands. Remember order of operations matter throughout the script.

If anyone discovers any bugs within the program, please contact me immediately. My contact is karthiknadim@gmail.com, or to reach me faster, my phone number is 5025537131.

III. Add Columns Together Function

Usage: pandasController.addColumns(self, columns, columnName = "")

Example: pandasController.addColumns(['Open', 'High', 'Low'], columnName = 'Add Columns')

	Date	Open	High	Low	Close	Adj Close	Testing Set	Add Columns
0	1980-12-12	0.100000	0.200000	0.300000	0.128348	0.100600	469033600	0.600000
1	1980-12-15	11.000000	2.000000	3.000000	4.000000	5.000000	175884800	16.000000

To use the add columns function, you will enter the columns that you would like to add together in an array as you can see within the example to select the columns that you would like to add together. After that, you have an option to set which column values you would like to modify or create based on the columnName parameter.

ColumnName —> False: It adds the columns and creates a new table with the added values separate from the main table. To use this table, you can store it into a variable and run [VARIABLE_NAME].to_csv('Output/[SAVE_FILE_NAME].csv', index = False, header = True). This will save the added values within a separate csv file.

ColumnName —> True: It adds or modifies columns within the input csv and adds columns or changes columns to the result of adding the columns together.

IV. Subtract Columns Function

Usage: pandasController.subtractColumns(self, columns, columnName = "")

Example: pandasController.subtractColumns(['Open', 'High', 'Low'], columnName = 'Subtract Columns')

	Date	Open	High	Low	Close	Adj Close	Testing Set	Subtract Columns
0	1980-12-12	0.100000	0.200000	0.300000	0.128348	0.100600	469033600	-0.400000
1	1980-12-15	11.000000	2.000000	3.000000	4.000000	5.000000	175884800	6.000000

ORDER OF COLUMNS MATTERS!!

To use the subtract columns function, you will enter the columns that you would like to subtract from each other in an array as you can see within the example to select the columns that you would like to subtract from each other. After that, you have an option to set which column values you would like to modify or create based on the columnName parameter.

ColumnName —> False: It adds the columns and creates a new table with the subtracted values separate from the main table. To use this table, you can store it into a variable and run [VARIABLE_NAME].to_csv('Output/[SAVE_FILE_NAME].csv', index = False, header = True). This will save the added values within a separate csv file.

ColumnName —> True: It adds or modifies columns within the input csv and adds columns or changes columns to the result of subtracting the columns from each other.

V. Multiply Columns Function

Usage: pandasController.multiplyColumns(self, columns, columnName = "")

Example: pandasController.multiplyColumns(['Open', 'High', 'Low'], columnName = 'Multiply Columns')

	Date	Open	High	Low	Close	Adj Close	Testing Set	Multiply Columns
0	1980-12-12	0.100000	0.200000	0.300000	0.128348	0.100600	469033600	6.000000e-03
1	1980-12-15	11.000000	2.000000	3.000000	4.000000	5.000000	175884800	6.600000e+01

To use the multiply columns function, you will enter the columns that you would like to multiply together in an array as you can see within the example to select the columns that you would like to subtract from each other. After that, you have an option to set which column values you would like to modify or create based on the columnName parameter.

ColumnName —> False: It adds the columns and creates a new table with the subtracted values separate from the main table. To use this table, you can store it into a variable and run [VARIABLE_NAME].to_csv('Output/[SAVE_FILE_NAME].csv', index = False, header = True). This will save the added values within a separate csv file.

ColumnName —> True: It adds or modifies columns within the input csv and adds columns or changes columns to the result of multiplication with each other.

VI. Divide Columns Function

Usage: pandasController.divideColumns(self, columns, columnName = "")

Example: pandasController.divideColumns(['Open', 'High', 'Low'], columnName = "Divide Columns')

	Date	Open	High	Low	Close	Adj Close	Testing Set	Divide Columns
0	1980-12-12	0.100000	0.200000	0.300000	0.128348	0.100600	469033600	1.666667
1	1980-12-15	11.000000	2.000000	3.000000	4.000000	5.000000	175884800	1.833333

ORDER OF OPERATIONS MATTERS!!

To use the divide columns function, you will enter the columns that you would like to divide from each other in an array as you can see within the example to select the columns that you would like to subtract from each other. After that, you have an option to set which column values you would like to modify or create based on the columnName parameter.

ColumnName —> False: It adds the columns and creates a new table with the subtracted values separate from the main table. To use this table, you can store it into a variable and run [VARIABLE_NAME].to_csv('Output/[SAVE_FILE_NAME].csv', index = False, header = True). This will save the added values within a separate csv file.

ColumnName —> True: It adds or modifies columns within the input csv and adds columns or changes columns to the result of division from each other.

VII. Change Column Name Function

Usage: pandasController.changeColumnHeader(self, oldName, newName)

Example: pandasController.changeColumnHeader('Volume', 'Testing Set')

ORIGINAL:

	Date	Open	High	Low	Close	Adj Close	Volume
0	1980-12-12	0.100000	0.200000	0.300000	0.128348	0.100600	469033600
1	1980-12-15	11.000000	2.000000	3.000000	4.000000	5.000000	175884800

NEW:

	Date	Open	High	Low	Close	Adj Close	Testing Set
0	1980-12-12	0.100000	0.200000	0.300000	0.128348	0.100600	469033600
1	1980-12-15	11.000000	2.000000	3.000000	4.000000	5.000000	175884800

To use this function, the user enters the command, and then puts the column that they would like to change, and then enters a new name for the column.