

Count by new **month-yr** column

Write a function **count_month_yr** to create the following dataframe using your new column **month-yr**,

	Timestamp
month-yr	
Apr-2018	28
Feb-2018	2
Jan-2018	148
Jan-2019	57
Mar-2018	41
Oct-2018	6
Sep-2017	74
Sep-2018	130

Notice that the order of the dates is incorrect. We will fix that later. Remember to include your **add_month_yr** code from the previous part, as your new function needs the output from it.

Here is the function signature: **count_month_yr(x)** where **x** is a **pd.DataFrame** that returns a **pd.DataFrame**.

Hint: If you added **add_month_yr** inside your **count_month_yr** and you changed the index in Part 2 then you might have trouble with this question. As the autograder calls **count_month_yr(add_month_yr(x))** But if you did not modify the index in Part 2, you need not worry. As calling **add_month_yr** a second time within your function will not change the dataframe.

Please put your Python code in a Python script file and upload it. Please retain your submitted source files! Remember to use all the best practices we discussed in class. You can use any module in the Python standard library, but third-party modules (e.g., Numpy, Pandas) are restricted to those **explicitly** mentioned in the problem description.

Tips:

- After you have submitted your file, do **not** use the browser back or reload buttons to navigate or open the page in multiple browser tabs, as this may cause your **attempts** to decrease unexpectedly. It may take up to thirty seconds for your code to be processed, so please be **patient**.
- If you find yourself back at the main page without any feedback or change in your **attempts** then it means that your code timed out or crashed in some unexpected way.
- Ensure that your development environment does not presume the existence of certain packages for the autograder. The autograder does not have anything other than the standard library and those third-party libraries **explicitly** named in the problem description.
- Do not leave extraneous statements in your code like test cases, print statements, or anything else besides what is needed to evaluate your submission because the the autograder will spend its limited time executing those lines, which may result in unexpected crashes or timeouts.

浏览... 0703.py

Upload Python source code file

Correct! Back to assignments. functional points = 10 /10 and validation points = 5/5