U UDACITY

PROJECT SPECIFICATION

Explore US Bikeshare Data

Code Quality

CRITERIA	MEETS SPECIFICATIONS
Functionality of code	All code cells can be run without error. *Tips*: Implement safeguards against invalid user inputs that can potentially break the codes. Please refer to the "Solicit and handle raw user input" rubric item for further details.
Choice of data types and structures	Appropriate data types (e.g. strings, floats) and data structures (e.g. lists, dictionaries) are chosen to carry out the required analysis tasks.
Use of loops and conditional statements	Loops and conditional statements are used to process the data correctly.
Use of packages	Packages are used to carry out advanced tasks.

CRITERIA	MEETS SPECIFICATIONS
Use of functions	Functions are used to reduce repetitive code.
Use of good coding practices	Docstrings, comments, and variable names enable the readability of the code. *Tips*: Please refer to the Python's documentation PEP 257 -- Docstring Conventions. Example of docstring conventions: ```python\ndef function(a, b):\n """Do X and return a list."""\n```

Script and Questions

CRITERIA	MEETS SPECIFICATIONS
Solicit and handle raw user input	Raw input is solicited and handled correctly to guide the interactive question-answering experience; no errors are thrown when unexpected input is entered. User inputs should be made **case insensitive**, which means the input should accept the string of "Chicago" and its case variants, such as "chicago", "CHICAGO", or "cHicAgo". You should also implement **error handlings** so your program does not throw any errors due to invalid inputs. For example, if the user enters "Los Angeles" for

CRITERIA	MEETS SPECIFICATIONS
	the city, the error handling should reject the user input and avoid breaking the codes.
Use descriptive statistics to answer questions about the data. Raw data is displayed upon request by the user.	Descriptive statistics are **correctly computed** and used to answer the questions posed about the data. Raw data is displayed upon request by the user in the following manner: • Your script should prompt the user if they want to see 5 lines of raw data, • Display that data if the answer is 'yes', • Continue iterating these prompts and displaying the **next** 5 lines of raw data at each iteration, • Stop the program when the user says 'no' or there is no more raw data to display. Tips: you can implement the `while` loop and track the row index in order to display the continuous raw data.

Suggestions to Make Your Project Stand Out!

- Change the structure of bikeshare.py to make the code more efficient or in better style.
- Ask and answer additional questions about the data beyond the questions already provided.
- Make the interactive experience wow-worthy! Add images, make it into a web app, etc. (If you do create a web app, make sure to include clear directions how to execute it.) Make it your own!