Explore US Bikeshare Data

Meets Specifications

**Code Quality**

All code cells can be run without error.

Appropriate data types (e.g. strings, floats) and data structures (e.g. lists, dictionaries) are chosen to carry out the required analysis tasks.

The analysis makes use of appropriate data types and data structures. In the next courses you will learn how to use pandas Dataframe that provides fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. <https://pandas.pydata.org/pandas-docs/stable/index.html>

Loops and conditional statements are used to process the data correctly.

Packages are used to carry out advanced tasks.

It is excellent that you are making use of functions that make the code more efficient but also easier to follow and understand.

Functions are used to reduce repetitive code.

Docstrings, comments, and variable names enable readability of the code.

The code is well formatted and appropriately documented with comments that explain the functionality.

**Script and Questions**

Raw input is solicited and handled correctly to guide the interactive question-answering experience; no errors are thrown when unexpected input is entered.

The code provides an interactive experience for the users. The code also provides a readable output that is easy to follow and understand. For the statistics of the duration, it is awesome that you transform the seconds into days hours and minute which are more natural and intuitive.

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 this manner: Script should prompt the user if they want to see 5 lines of raw data, display that data if the answer is 'yes', and continue these prompts and displays until the user says 'no'.

As mentioned above, pandas provide you with more efficient computation but the code is also simpler. For example, to implement the “popular\_trip”, you need a single code line.

result = df[['Start Station', 'End Station']].groupby(['Start Station', 'End Station']).size().nlargest(1)

You will learn more about pandas in the following courses.