**Code Quality**

|  |  |
| --- | --- |
| **CRITERIA** | **MEETS SPECIFICATIONS** |
| Functionality of code | All code cells can be run without error. |
| 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. |
| Use of functions | Functions are used to reduce repetitive code. |
| Use of good coding practices | Docstrings, comments, and variable names enable readability of the code. |

**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. |
| 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 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'. |

def get\_day():

def popular\_month(df):

def popular\_day(df):

def popular\_hour(df):

def popular\_stations(df):

def display\_data(df):

خصوصا هذه الجزئيه filter\_lower, filter\_upper = get\_day()