**1. Tabulate the execution times of each of the individual approaches for computing distance in Python (i.e., run the shared code on your computer, note the times, and tabulate them).**

|  |  |
| --- | --- |
| For Loop Time | 0.010 seconds |
| Apply function time | 0.006 seconds |
| Vectorized time | 0.001 seconds |

**2. Next, replicate the for-loop based approach (the first one) and two different ways to make that version more efficient, in R. Profile these three approaches, and tabulate the results.**

|  |  |
| --- | --- |
| For Loop Time | 0.0124540328979492 seconds |
| Apply function time | 0.00394701957702637 seconds |
| Vectorized time | 0.00129795074462891 seconds |

**3. Based on the computational efficiency of implementations in Python and R, which one would you prefer? Based on a consideration of implementation (i.e., designing and implementing the code), which approach would you prefer? Taking both of these (run time and coding time), which approach would you prefer?**

The execution times show that Python delivers higher speed than R through its three different implementations which achieves minimal runtime in vectorized methods. Since speed matters significantly in processing large data Python would be my choice for speed purposes. Implementation of code proves easy for both Python and R while vectorized approaches represent the most readable and concise method for coding in these languages. Vectorized implementation in Python emerges as the optimal solution because it combines high performance execution with easy maintainable code.

**4. Identify and describe one or two other considerations, in addition to these two, in determining which of the two environments – Python or R – is preferable to you.**

The decision between Python and R requires thorough evaluation of their user community support along with their integration capabilities in addition to execution speed and programming efficiency. The Python ecosystem includes databases which surpass data analysis capabilities thus making it attractive across different development applications such as machine learning and automation and web development. Data analysis requirements demand R because it includes extensive statistical packages which perfectly serve statistical modeling needs. Wider industrial adoption of Python has resulted in robust documentation alongside extensive libraries and various application support from different communities.