**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).**

|  |  |
| --- | --- |
| **Approach** | **Time** |
| For – Loop | 0.005632 |
| Apply() | 0.001584 |
| Vectorized | 0.002905 |

**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.**

|  |  |
| --- | --- |
| **Approach** | **Time** |
| For – Loop | 0.01464 |
| Apply() | 0.007636 |
| Vectorized | 0.000487 |

**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?**

R achieves the fastest computations through vectorized operations which defeat the execution of for-loop and apply functions both in R and Python. Python shows strong performance in general operations although its apply function and for-loop execution times work faster than other competitors. Python provides better implementation capabilities through its Panda and NumPy libraries that simplify large dataset coding while resulting in more compact code structures. I select Python as my preferred language since its combination of execution speed and user-friendly attributes makes it most suitable for implementable real-world systems which demand scalability along with machine learning or automation tool integration.

**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.**

Both R and Python differ in their adoption based on the scope of their system networks together with interface compatibility features. Programmers should select R as their primary platform because it provides exceptional capabilities for data analysis both in statistical computations and spatial work along with visual presentation. Since machine learning and AI together with automation and system integration are strengths of Python it demonstrates superior capability for full data science processes. Python stands out for large-scale applications due to its unified database and web application and big data processing tool integration features even though R still provides excellent capabilities for specialized statistical research within academic settings.