**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 | Execution Time |
| For Loop | 0.013 seconds |
| Apply function | 0.006 seconds |
| Vectorized method | 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.**

|  |  |
| --- | --- |
| Approach | Execution Time |
| For Loop | 0.000202 seconds |
| Apply function | 0.000721 seconds |
| Vectorized method | 0.0000259 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 reveal that Python vectorized operations stay substantially faster than loops (0.001s to 0.013s) but apply functions perform at 0.006s. Both R and Python follow equivalent performance trends yet R execution times are faster than Python execution times at 0.0000259s versus 0.000202s. R proves more efficient than Python for running all computational methods across the language comparison. From an implementation standpoint Python offers better intuitive syntax which makes the language easier to read by programmers whose background includes other language contexts. Given the combination between execution speed and code clarity Python remains my preferred language even though its speeds are marginally slower. Python's easier design along with its extensive library framework and enhanced readability delivers better value than marginal performance losses despite its microseconds of slower execution.

**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 selection between Python and R depends heavily on two critical factors including visual analytics features and practitioner backing. R demonstrates superiority in statistical functionality with its advanced visualization capabilities through ggplot2 library that develop a natural graphic expression system which turns it into an excellent platform for data exploration and high-quality visual presentation. The integration and broad support of other tools available through Python together with its larger developer community let the language serve as a flexible solution for creating multiple data science pipelines. The decision will also depend on your team's skill sets and project requirements because academic researchers or statistical examiners should use R while production systems derived from Python integration are often more effective.