**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 | 0.0035 seconds |
| Apply time | 0.0010 seconds |
| Vectorized | 0.0010 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 | 0.02788401 seconds |
| Apply time | 0.0002241135 seconds |
| Dplyr time | 0.002485037 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 computational efficiency results show R outperforms Python because its Apply method achieves execution time of 0.0002 seconds compared to Python's vectorized approach which requires 0.001 seconds. The implementation process for Python becomes more accessible through its natural syntax structure that simplifies general programming tasks. In Python the vectorized NumPy implementation provides an effective trade-off between performance strength and code readability thus becoming my default solution for this dual performance and development-based selection. The execution times between these code implementations reveal an insignificant difference which supports Python as the superior choice due to its better syntax and detailed documentation.

**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.**

Two primary variables determining the selection between Python and R comprise seed integration capabilities and built-in data visualization tools. Python exceeds R in its ability to work with various systems because its broad ecosystem enables seamless integration with web services and machine learning frameworks which enhances value in complete data science workflows. Statistical packages within R along with its advanced built-in visualization features especially ggplot2 establish R as an exceptional tool for statistical research and making professional-quality graphical output. Projects that demand either statistical analysis or academic research will benefit from R because Python proves stronger for integration needs and complete application development.