

Question 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 Elapsed
For loop	844 μ s \pm 138 μ s per loop (mean \pm std. dev. of 7 runs, 1000 loops each)
Iterrows	2.03 ms \pm 95.8 μ s per loop (mean \pm std. dev. of 7 runs, 100 loops each)
Apply function (with lambda)	1.42 ms \pm 304 μ s per loop (mean \pm std. dev. of 7 runs, 1000 loops each)
Vectorized (on Pandas series)	1.4 ms \pm 214 μ s per loop (mean \pm std. dev. of 7 runs, 1000 loops each)
Vectorized (on NumPy array)	169 μ s \pm 12.2 μ s per loop (mean \pm std. dev. of 7 runs, 10000 loops each)
Cythonized (with dtypes)	827 μ s \pm 167 μ s per loop (mean \pm std. dev. of 7 runs, 1000 loops each)

Question 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 Elapsed
For loop	Time difference of 0.094872 secs
Vectorized approach with data.table	Time difference of 0.07317901 secs
Foreach Parallelized Loop	Time difference of 0.09297013 secs

Question 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?

Answer: Based on both the computational efficiency and ease of implementation, I prefer the Python approaches speeding up code more than the R ones. I find that Python has a larger number of libraries to support speeding up my code, and it is relatively easy to implement.

Question 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.

1. Syntax - I believe that Python syntax is a bit easier to understand. It may differ from person to person.
2. Industry - Depending on the industry I am working in, I might prefer R over Python actually. Working in clinical trials, it's important to have an easily validated program/environment, and R is more easily validated than Python. Also, the plots produced by ggplot may be preferred to Python matplotlib plots.