Task 2

Unfortunately, we do not have the time to run this but we expect the Rcpp version of the "fundamental algorithm" to be a lot faster than the Python implementation.

Task 3

Rerunnable

R: 4/5

Python: 4/5

Rcpp: 3/5

We believe that R and Python have a 4/5 compared to a 3/5 of the Rcpp because of the fact that R and Python are more standard and work by themselves. So even in the future running the code we have written should be quite straightforward as the objects we used are very common.

However, Rcpp is essentially translating code that only C++ understand into code that both R and C++ can understand which makes it even easier for something to go wrong. Essentially if the software or libraries of either R or C++ change an issue might rise. The key in Rcpp is compatibility of the functions of the two languages we used.

Repeatable

R: 5/5

Python: 5/5

Rcpp: 5/5

Since we follow the same algorithm and principles in every language, given a certain input we shall receive the same output no matter where we run it. For this reason we choose 5/5 here.

Reproducible

R: 4/5

Python: 4/5

Rcpp: 4/5

For similar reason to before, since the code we have written in every language follows a deterministic procedure and there is no randomness the algorithm will always return the same

output. Even if the behaviour of any of the languages change, the output will be the same (for a given input). The reason for putting 4 instead of 5 is that we should have included some automated tests, which would indicate the algorithm is working as intented.

Reusable

R: 4/5

Python: 4/5

Rcpp: 3/5

The reason we put 4/5 here for R and Python and 3/5 for Rcpp is that overall we believe that the code we have written is quite clear without being more complicated than it should. More comments can be added to facilitate the understanding, especially for the Rcpp version. Additionally, the use of some mechanisms can be used to aid the use of the code by a third-party (i.e. type hints). Documentation **must** be added for the Rcpp library to help the user understand how and what does this package actually do. This way it is easier for an error or an improvement to be spotted by someone.

Replicable

R: 2/5

Python: 2/5
Rcpp: 1/5

The reason for 2/5 here is that we haven't followed a proper procedure to ensure replicability. We just provided references to the pseudocode in the book that writes the Fundamental Algorithm. We put 1/5 in Rcpp because we **should** include this in the documentation (which we have not).

Some general comments, given more time, we emphasise again the importance of writing a documentation for the package. The most important thing is to include what the input in R needs to be as well as which R version and Rcpp version this package was written in.