After trying out the 1k, 10k, and 100k text files, the time differences were more drastic than I thought that they were going to be. First of all, the implementation of bubble sort was the easiest and shortest for me to do, so I assumed that bubble sort would be the fastest in regards to the time it takes for the function to execute, but this was not the case. Bubble sort was the slowest one to run in comparison to merge sort and selection sort. Furthermore, I did not anticipate the difference in times for merge and selection to be so drastically different to the point where merge was always faster. If choosing to use the merge sort, although it is fastest, making a temporary array for it causes more space to be taken up which is not the most efficient. If choosing to use the selection sort, then it is very slow in comparison to the merge sort when running large lists of numbers. In addition, if choosing to use the bubble sort, then the sorting operations are carried out the slowest so in regards to time efficiency, bubbles would not be the wisest choice, especially for large lists of data. Furthermore, the choice of programming language, the only language that I tested was with C++ so I do not have any other language to compare it to to see how the results are affected. According to this website that I found after doing research on different languages,

https://www.geeksforgeeks.org/top-10-fastest-programming-languages/,

 I found that C++ is one of the fastest and most efficient languages, so this could have improved the results of the sorting algorithms that I ended up getting. If I chose a slower programming language, the time results may have been slower as well. The shortcomings of this empirical analysis may include the fact that more testing must be done in order to have a better understanding of all the sort functions. So far the data types that were tested were only large sets of doubles, but smaller sets as well as much larger sets of data could also be tested in order to see how that affects the sort. In addition, other data types could also be tested in order to test out the efficiency of the sorting algorithms in further detail.