Emma Horsfield
Randy Truong

a) Introduction
    i) The purpose of this study is to determine whether selection sort, insertion sort, or quick sort is the best and therefore, the fastest sorting method in R.
b) Methodology

    i) Using the appropriate sorting method, we created three different samples that each contain 10 measurements (in seconds) of how long it takes to sort a vector containing 10 elements, thus we have three different samples involving quick sort, insertion sort, and selection sort methods. Each sample was sorted 100 times in order to reduce systemic imprecision, resulting in a single, more precise measurement. The vectors being sorted were of random permutation using the sample() function.
    ii) We decided to try ANOVA test because we had three samples and then tested the assumptions to see if we could use it or if we had to do Kruskal Wallis instead. The assumptions for ANOVA are the data involved must be interval or ratio level data, the populations from which the samples were obtained must be normally or approximately normally distributed, the samples must be independent, and the variances of the populations must be equal. We didn't have evidence that the samples were not independent so we assumed that they were. Further, time is ratio level data because it has an absolute zero (we can't have negative time and 0 minutes means that we don't have any time) and because the 2 minutes is twice as much time as 4 minutes so we can divide values. Thus, that assumption is satisfied. In order to check normality, we removed outliers and checked skewness. We then did the homogeneity of variance test to see if the standard deviations were equal. Then, we constructed a scatter plot of the sorting times (for insertion sort only) against vector size, which followed a positive correlation in the form of a quadratic, so in order to meet assumptions of a bivariate normal distribution, we removed any influential outliers and took the square root of all of the y-values in order to demonstrate linearity. As a result, we have a scatterplot in a linear form with a line of regression drawn.

c) Findings: The various steps/calculations needed in conducting the hypothesis test(s) involved.

    In checking the homogeneity of variance, we assured that the larger variance was in the numerator, keeping in mind to cut in half the alpha level, so 0.05 becomes 0.025. We used the ANOVA test to decide if there was a significant difference between the mean times of selection sort,

insertion sort, and quick sort. This resulted in an F-value of 119.55. Using the P-Value method, we got (Pr(>F)) = $1.593*10^{-13}$ and after comparing it with an alpha level of 0.001 which we got from the ANOVA chart that R gives you after running the test and we reject the null hypothesis that all of the mean times are equal and make an inference that at least one mean is different. Due to this, we must proceed with the Scheffe test to determine which mean(s) are different. By doing so, we found that quick sort is significantly different because when comparing the Fs values with the critical value 6.770380. By doing so, we reject the pair of insertion sort and quick sort and selection sort and quick sort. Noticing that quick sort was the common sample in both, we can make an inference that quick sort is significantly different from the other two.

d) Discussion and Interpretation: i.e., make an inference for the hypothesis test(s).
   i) From the ANOVA test, we rejected the null hypothesis since the p-value (Pr(>F)) = $1.593*10^{-13}$ < 0.001 (alpha). There is enough evidence to reject the claim that there is no significant difference between the mean times of selection sort, insertion sort, and quick sort. From Scheffe's test, the test statistics (206.98 and 139.38) for when the sample was compared against quick sort were within the rejection region with critical value +/- 6.770, thus rejecting the null hypothesis; however, the test statistic was not in the rejection region for insertion sort vs. selection sort (t.s. = 6.33), so in this case, it failed to reject, so these two sorting methods are not significantly different from each other. There is evidence that quick sort is different.

e) Predictions and Comments: (address any questions asked in the directions)
   i) Is this inference surprising? This is not surprising because quick sort is probably labelled "quick" sort due to its faster sorting ability than other sorting methods, thus it makes sense for it to be different. Further, from our scatter plot and best fit line, we predict that an insertion sort on a vector of size 1000 should take 0.677 seconds; we are not confident that that this is a good predictor because we rejected the previous t-test, thus determining that the correlation between insertion sort and vector size is significant. Additionally, since we took the square root of the y-values to demonstrate linearity within the scatterplot, this influences the line as a predictor. Using the insertion.sort() method, we found that it actually takes 0.444 seconds to sort a vector size of 1000, so our prediction is off by ~ 0.14 seconds. Are we surprised that R allows for quick sort in its sorting methods? Yes and no. Yes, because for smaller 'n' values (n = size of

vector), insertion sort is faster than quick sort at sorting values, but as ‘n’ increases, quick sort becomes more effective and faster. No, because quick sort is overall faster, especially as vector sizes increases, thus it makes the most sense to offer the most cost-time efficient sorting method, rather than resorting to slower methods.