APPM 5515: High Dimensional Probability– Fall 2020 — Homework 4

Brandon Finley

1 Introduction

In this assignment, we learned how to implement the Johnson-Lindenstrauss Transform (JL) and Fast Johnson-Lindenstrauss Transform (FJL). Throughout the report, you will see numerous plots that go over the different combinations used for our dataset of biological genes. We will go over their pairwise distances, nearest neighbor accuracy, and finally compare the accuracy and time of the JL compared to FJL.

2 Pairwise Distance Plots

This next group of plots computes the pairwise distance between the patients. This is done before and after the transformation as well as with both types of transformations. Along with the scatter plot, which plots the orignal data vs. the transformed, we also see a histogram that shows the distortion ratio (reduced dataset / old dataset). We would expect a proper transformation would be tightly concentrated around a value of $r = 1$.

2.1 k = 25

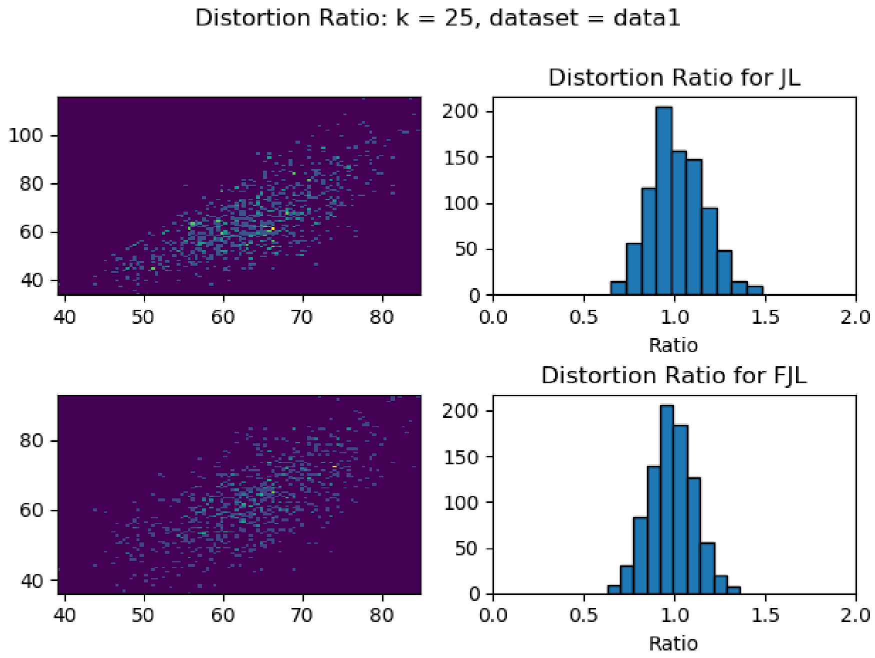

Figure 1: Data 1, k = 25

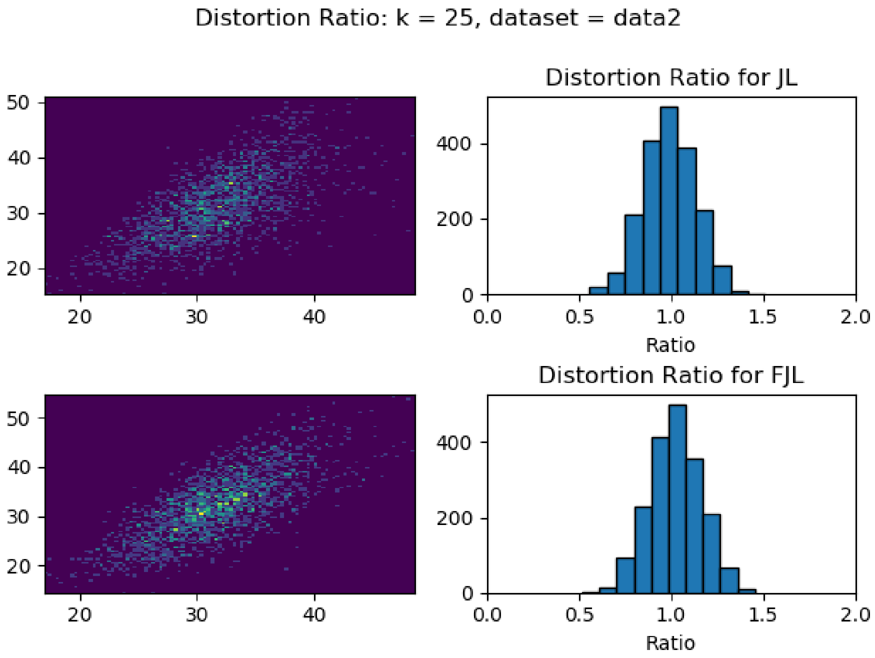

Figure 2: Data 2, k = 25

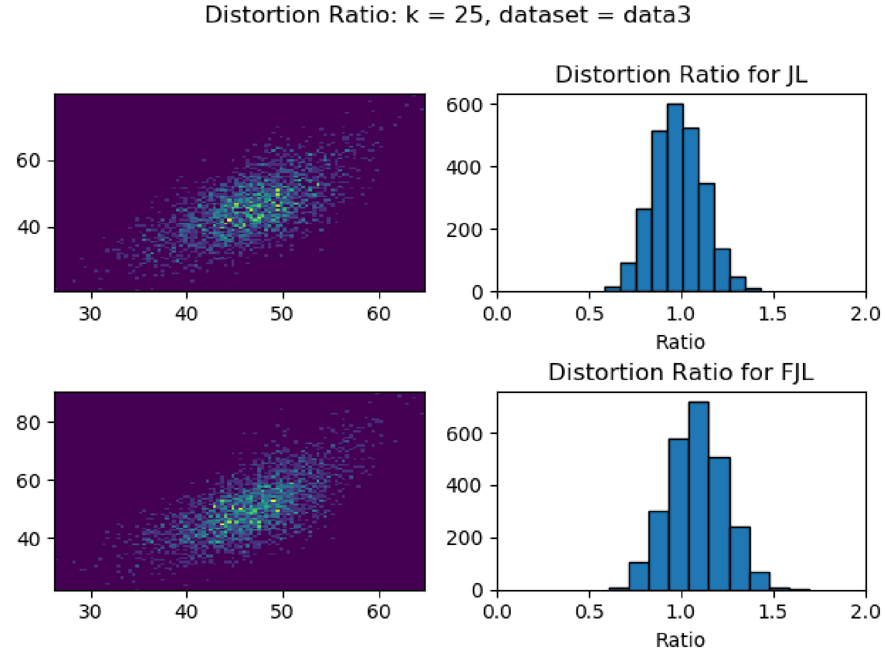

Figure 3: Data 3, k = 25

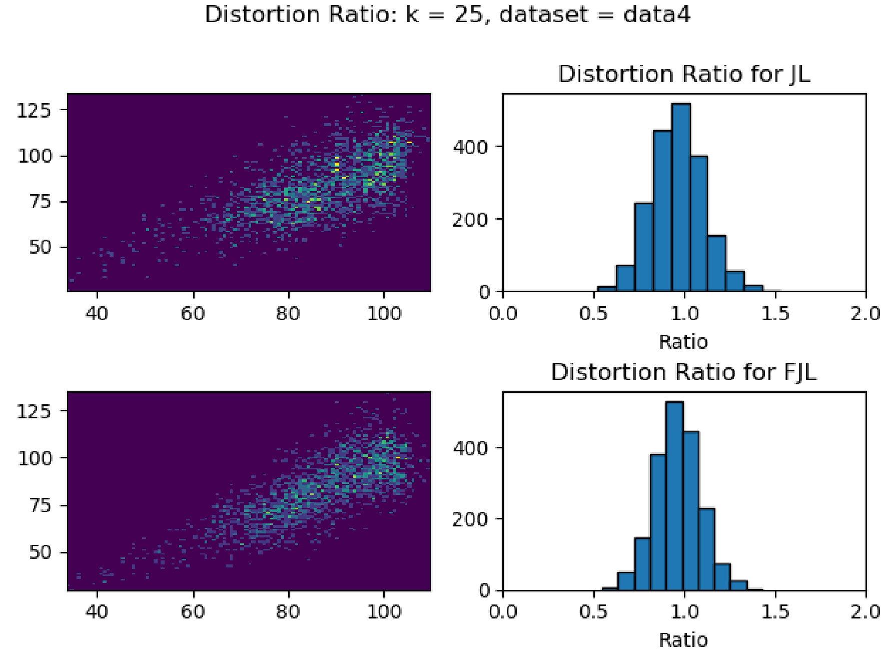

Figure 4: Data 4, k = 25

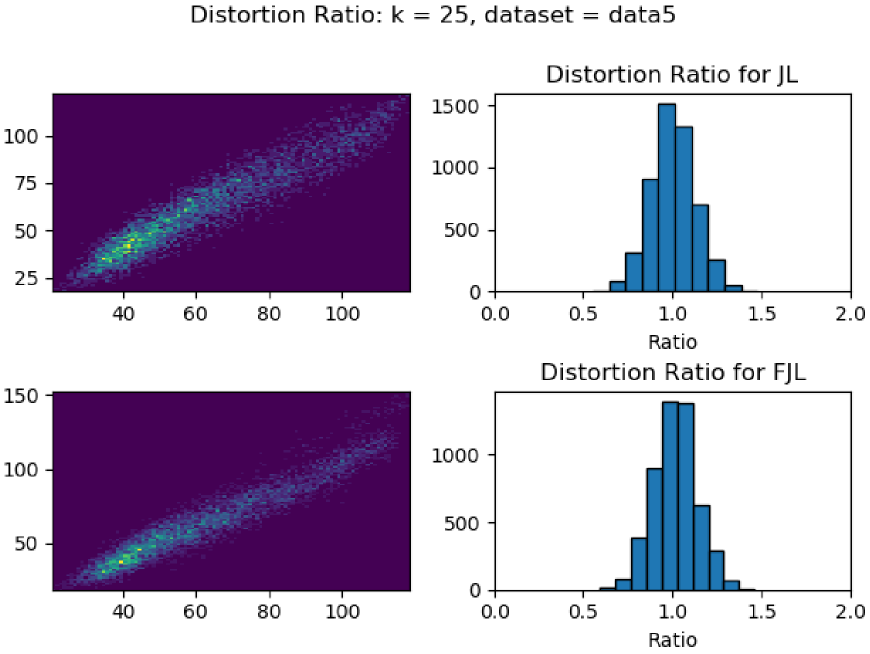

Figure 5: Data 5, k = 25

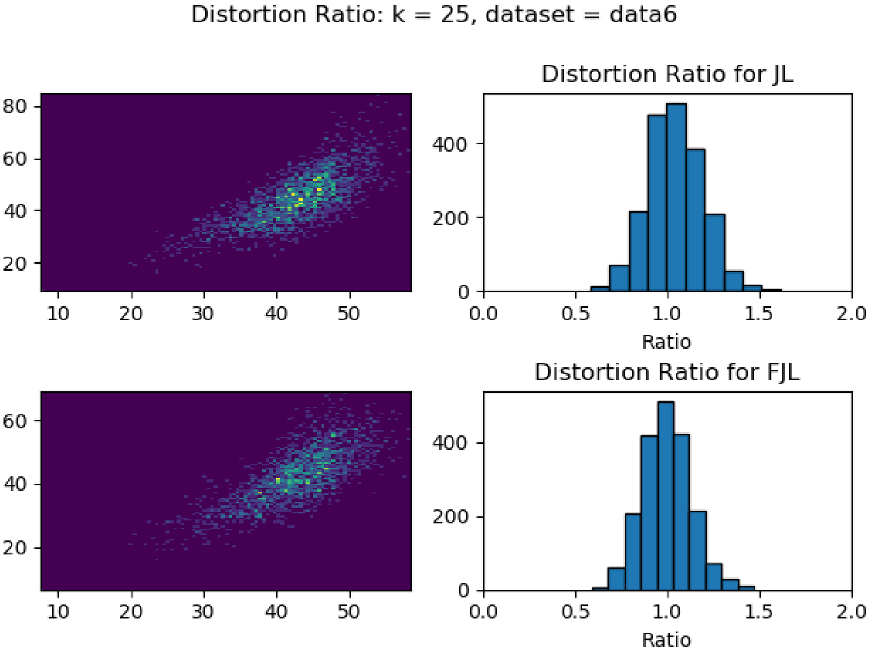

Figure 6: Data 6, k = 25

2.2 k = 100

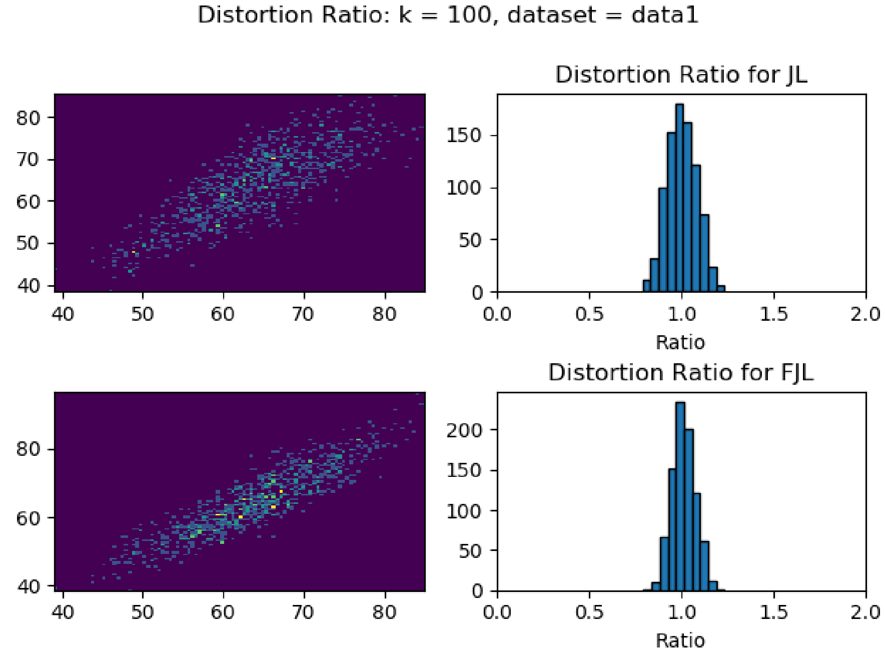

Figure 7: Data 1, k = 100

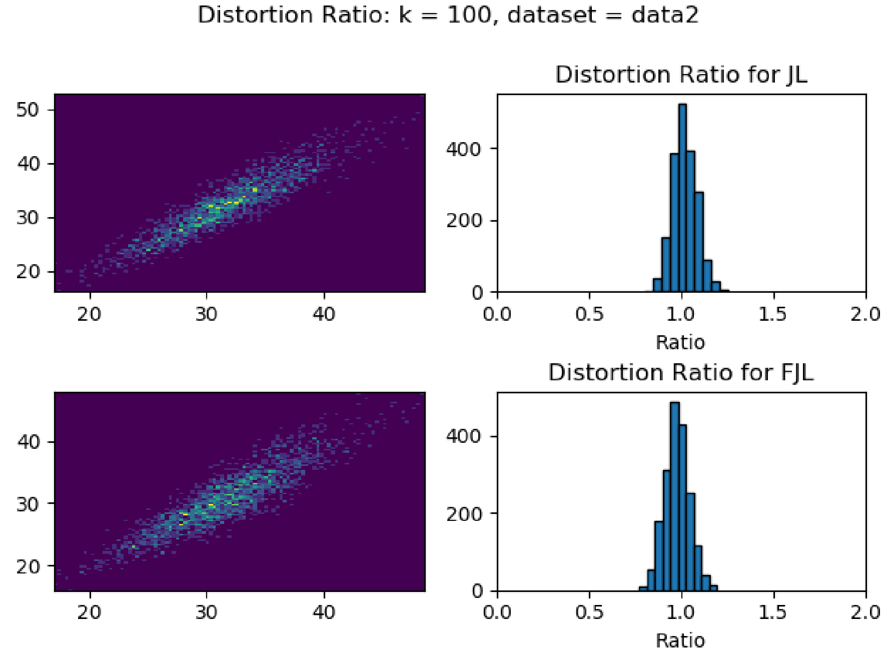

Figure 8: Data 2, k = 100

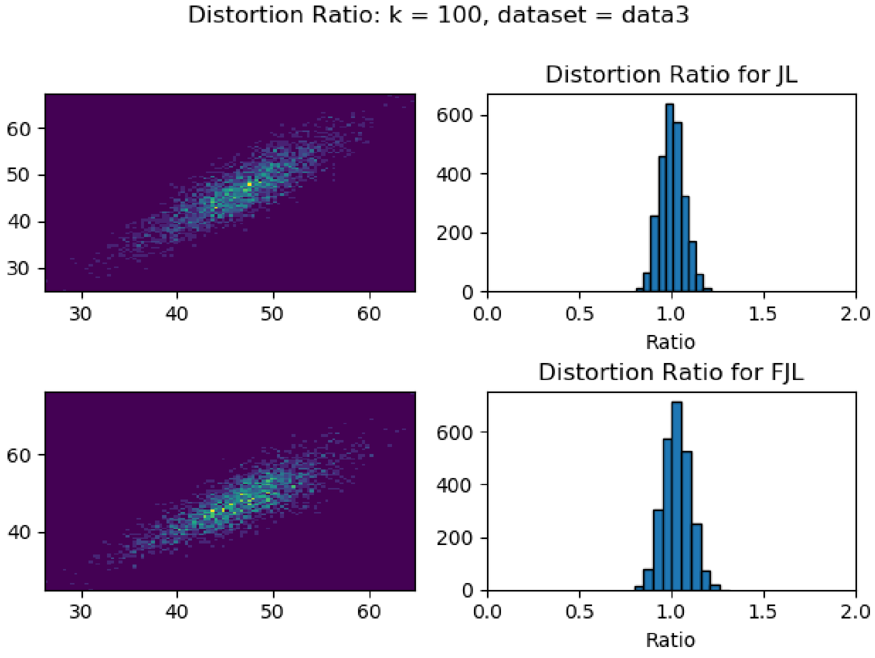

Figure 9: Data 3, k = 100

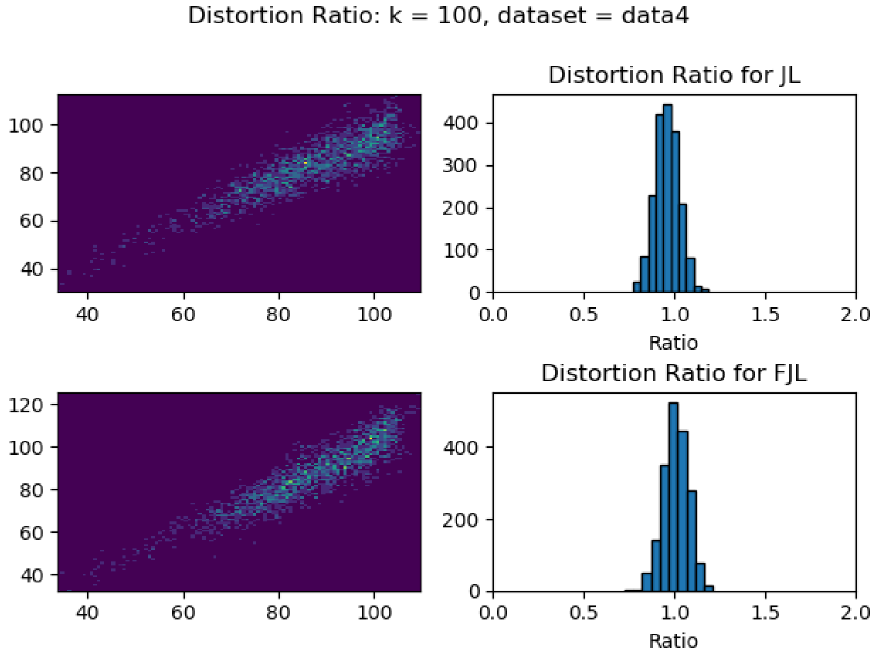

Figure 10: Data 4, k = 100

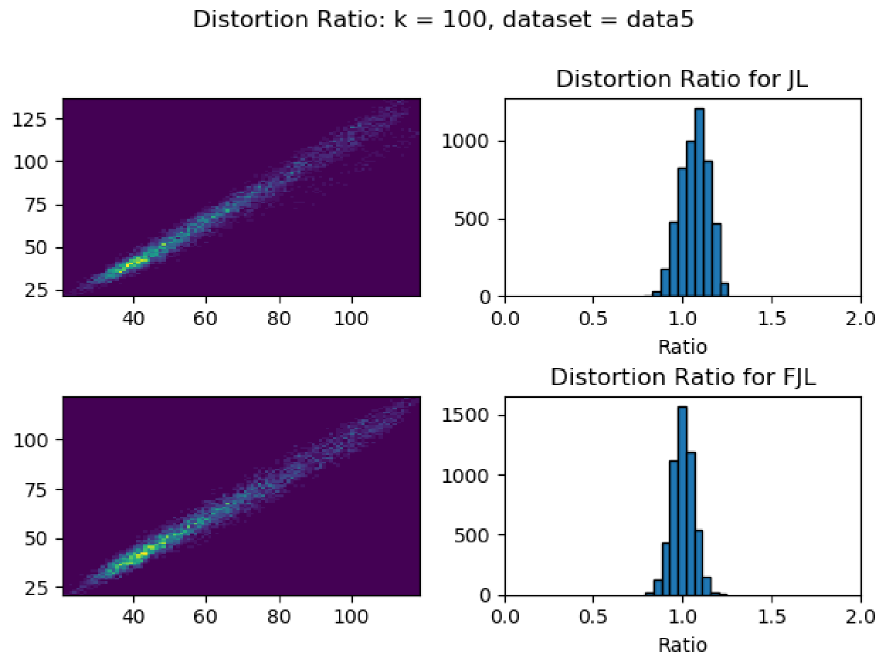

Figure 11: Data 5, k = 100

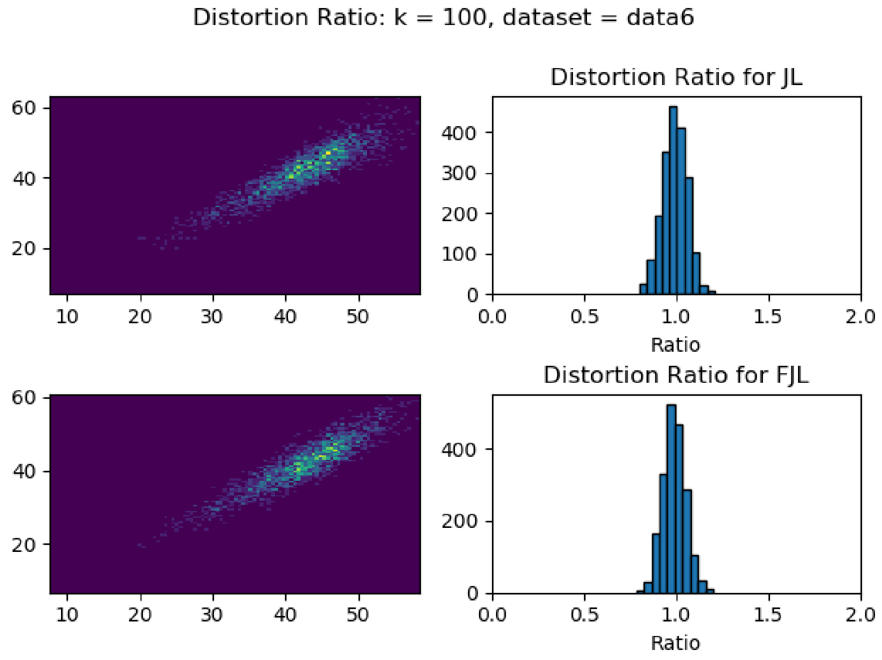

Figure 12: Data 6, k = 100

2.3 k = 225

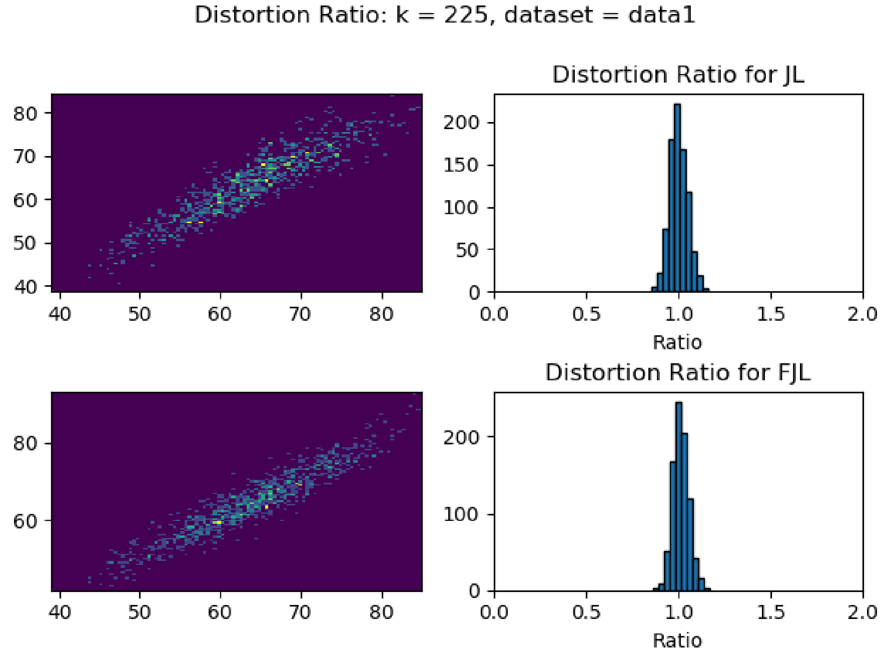

Figure 13: Data 1, k = 225

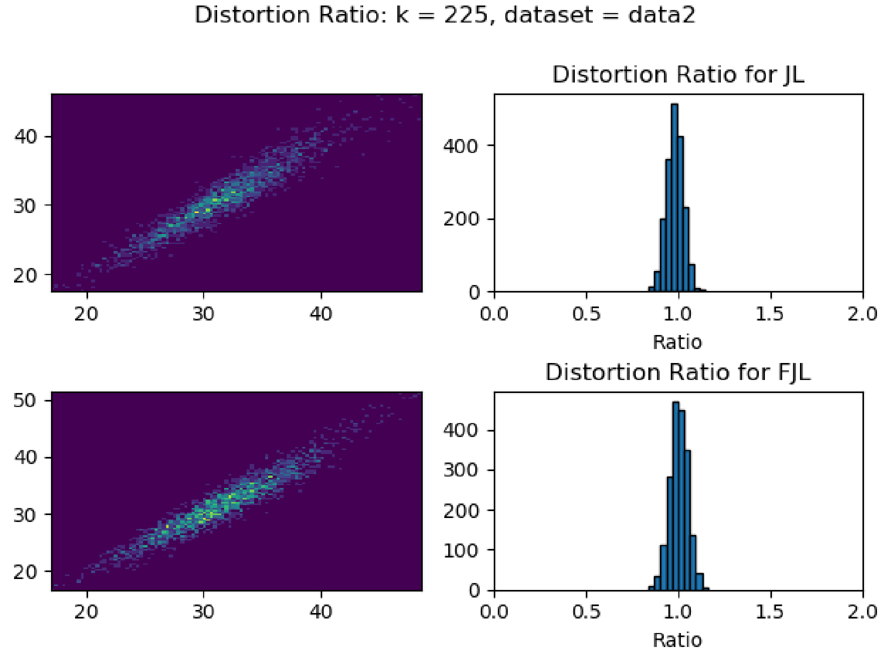

Figure 14: Data 2, k = 225

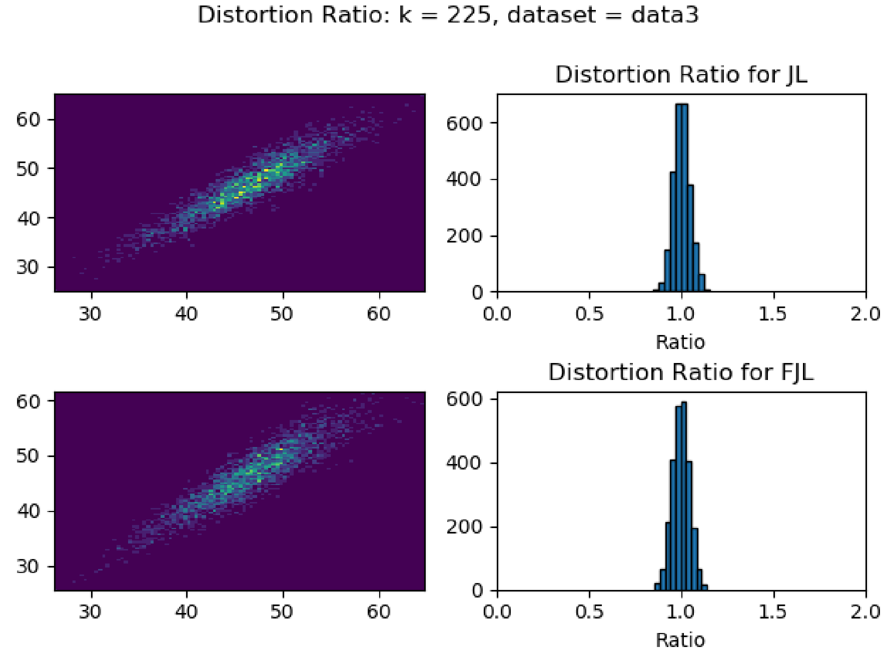

Figure 15: Data 3, k = 225

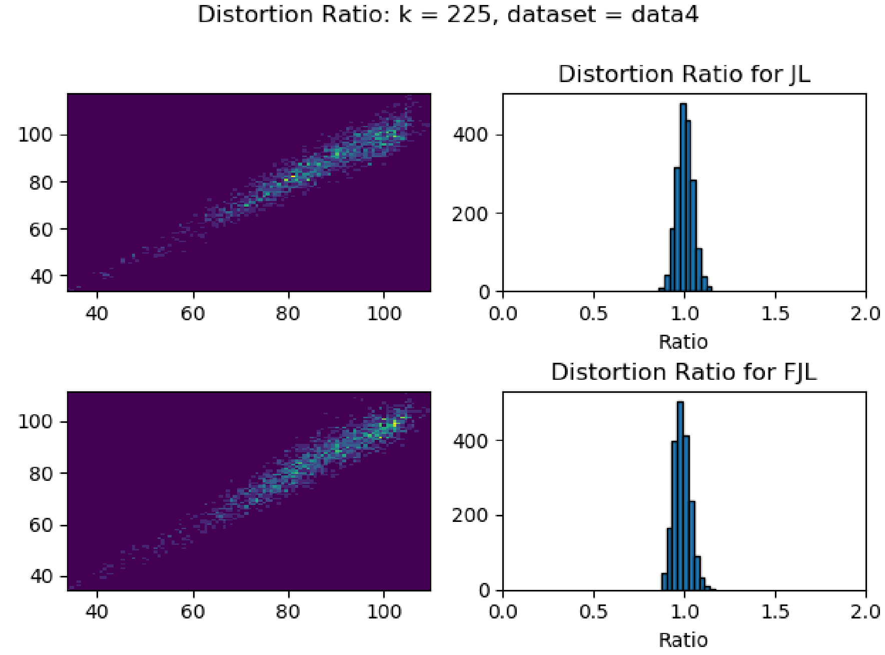

Figure 16: Data 4, k = 225

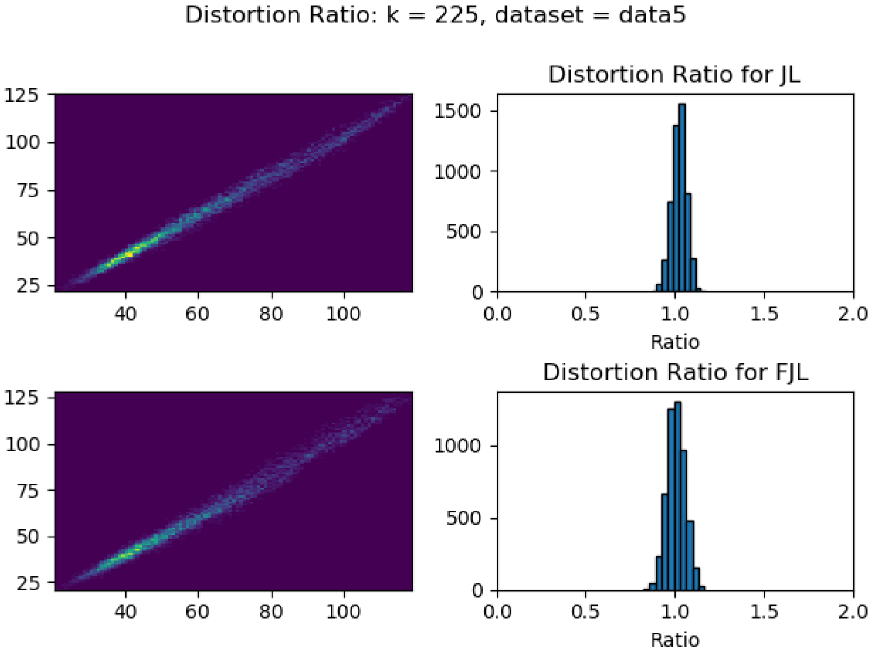

Figure 17: Data 5, k = 225

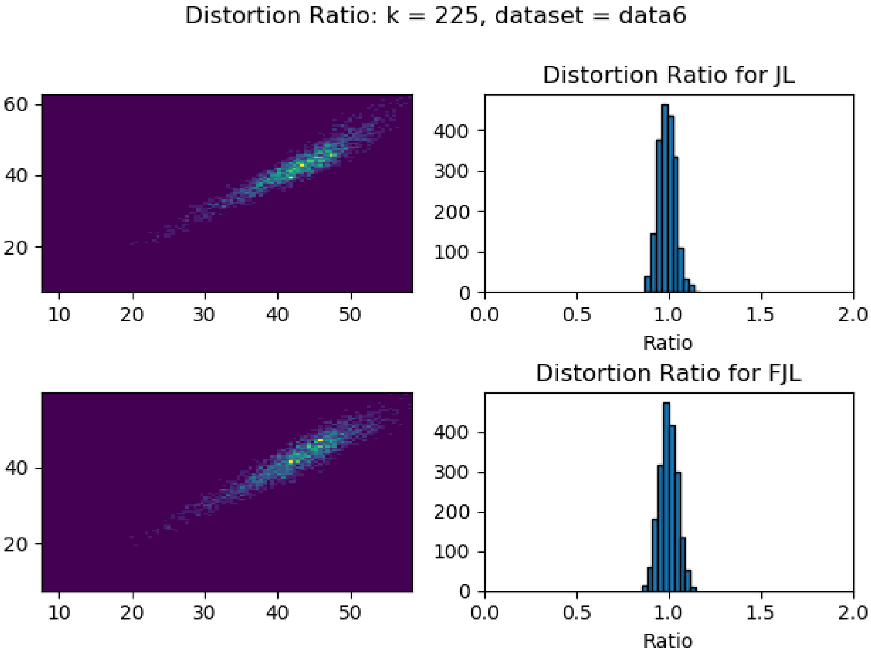

Figure 18: Data 6, k = 225

2.4 k = 400

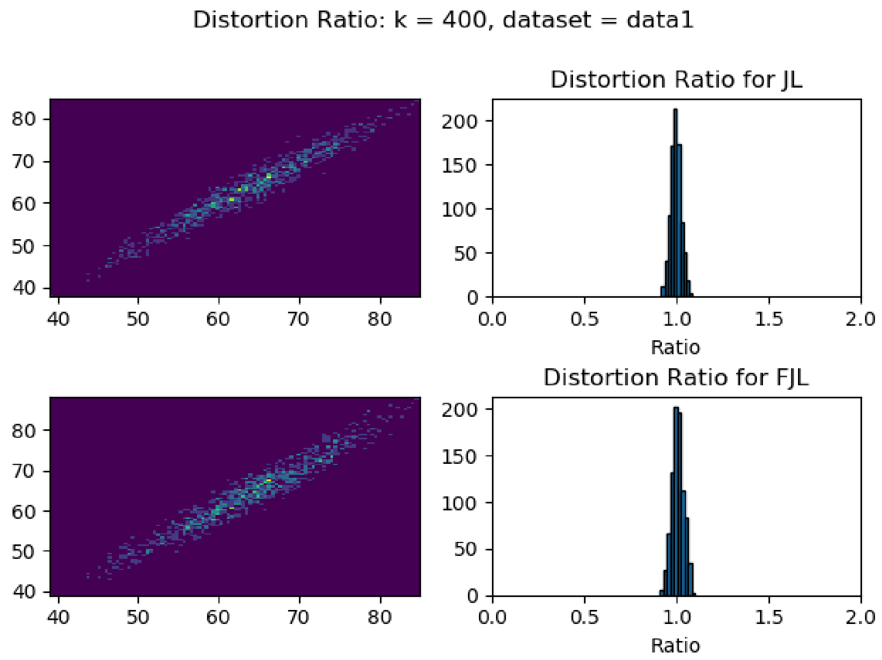

Figure 19: Data 1, k = 400

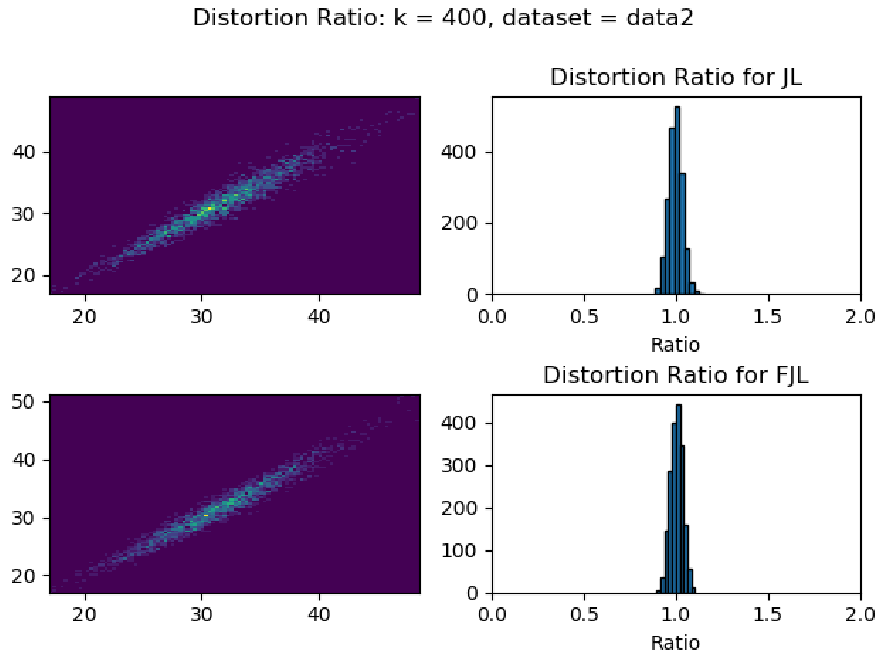

Figure 20: Data 2, k = 400

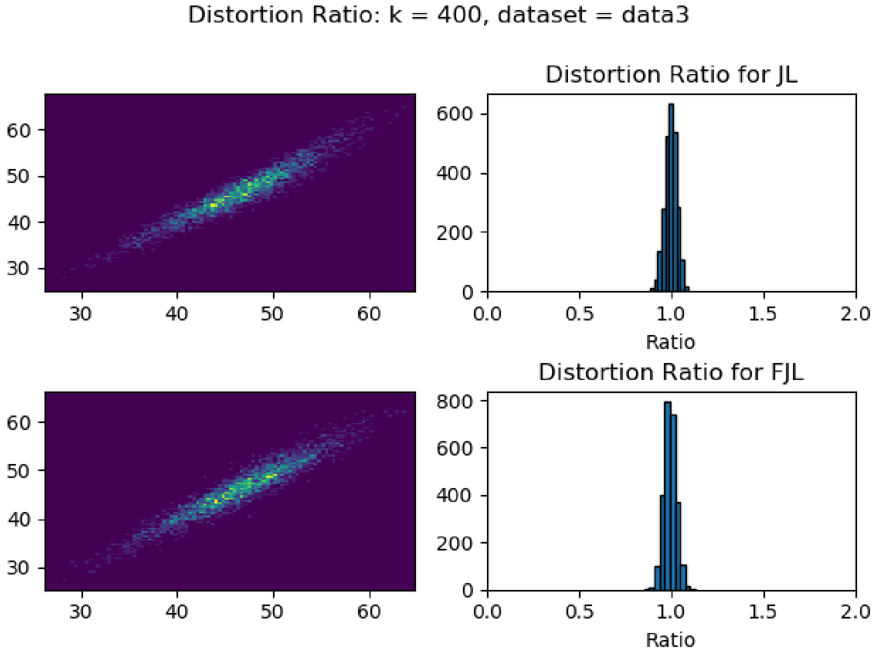

Figure 21: Data 3, k = 400

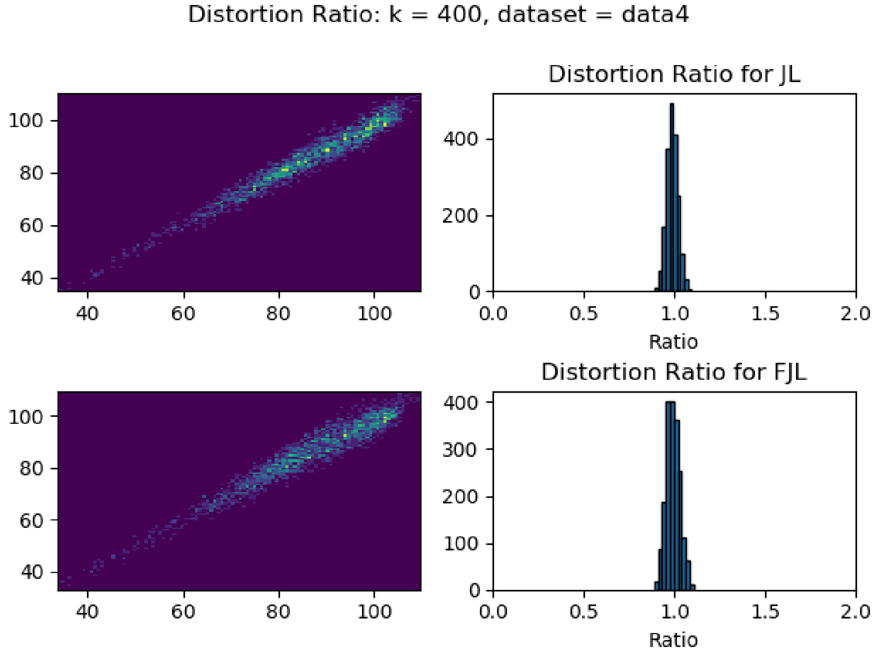

Figure 22: Data 4, k = 400

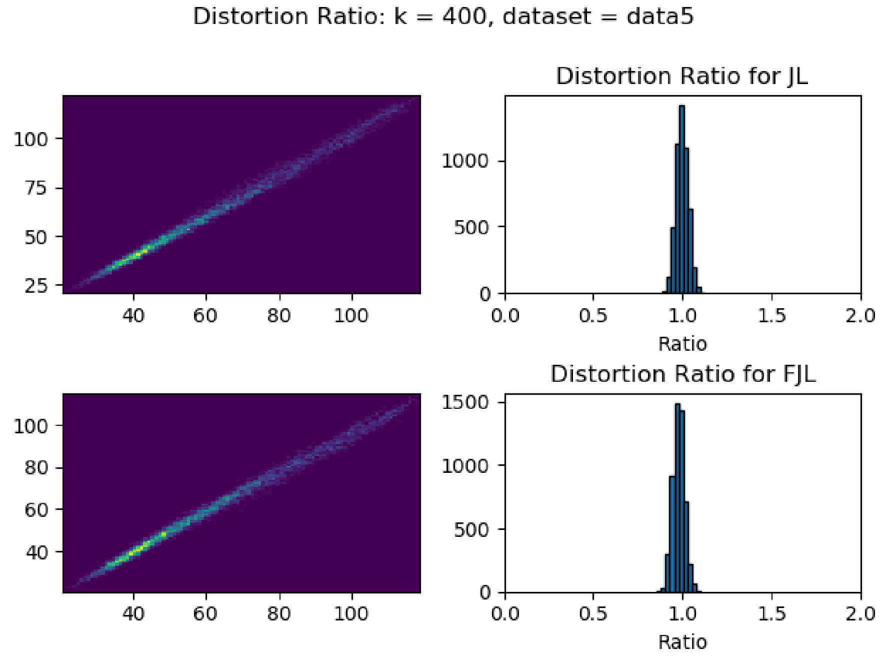

Figure 23: Data 5, k = 400

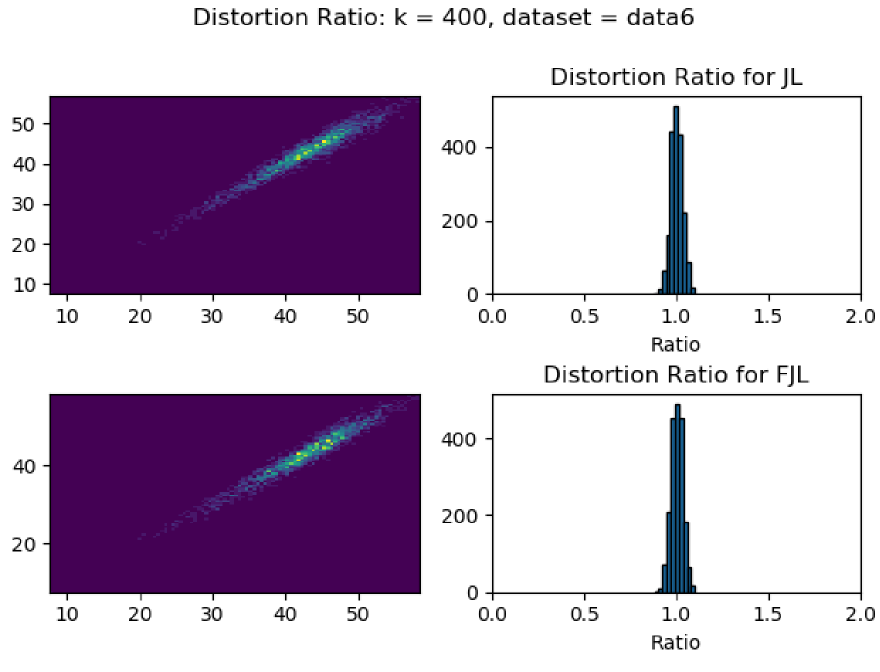

Figure 24: Data 6, k = 400

2.5 Analysis

Overall, we observed that as the dimension of k increased, the histogram became sharper and more concentrated around a ratio of $r = 1.0$. Additionally, all resembled a normal Gaussian.

Another thing to note, which was interesting to me, was that the scatter plot of the distances before and after the transformation seemed to depend on the dataset rather than the dimension. Namely, data1 had a very hazy plot while data5 was very concentrated. This makes sense since some data might have some global topology that varies significantly between datasets as the space is deformed. I believe this would be an interesting topic to see why some geometries behaved this way.

Another very obvious trend was the that scatter plot concentrated around $y = x$. For all of them, even hazy ones, there was a clear linear relationship.

Lastly, notice that certain datasets concentrate not only on the axis but have the highest density in different parts of the curve. For instance, in data5, the highest density are the ones nearest to the origin while in data4, it is the furthest away. Once again, this depends on the structure of the dataset as we could rearrange the set to get different results.

3 Theoretical vs. Experimental

From last homework, we know

$$P\left(\mathbf{G} = (g_{i,j}); \forall 1 \leq k < l \leq N, \left|\frac{\|\mathbf{Q}(x_k - x_l)\|^2}{\|x_k - x_l\|^2} - 1\right| \geq \epsilon\right) \leq \delta$$

We also learned that theoretically, the dimension that we choose to reduce to does not depend on the original dimension. From above and below, you can see that the accuracy retained, indeed, did not depend on the dataset we used (original dimension). In the above histograms, regardless of the dataset we used, the concentration around the center was the same. This affirms this theoretical result.

Specifically, for its accuracy, we see that as k increases, ϵ decreases as the concentration increases around $r = 1$. For its probability, we see that with high probability, its pairwise distance is maintained. Furthermore, this also increases with k.

Lastly, FJL compared to FL seems to not make a difference with the structure of the local distances.

4 Accuracy from k values

Here we graphed a plot for each dataset. Within each dataset's plot, we compared the different k values. You will also notice that I plotted JL as well as FJL here. Feel free to go through the plots; at the end I will do an analysis between the variations.

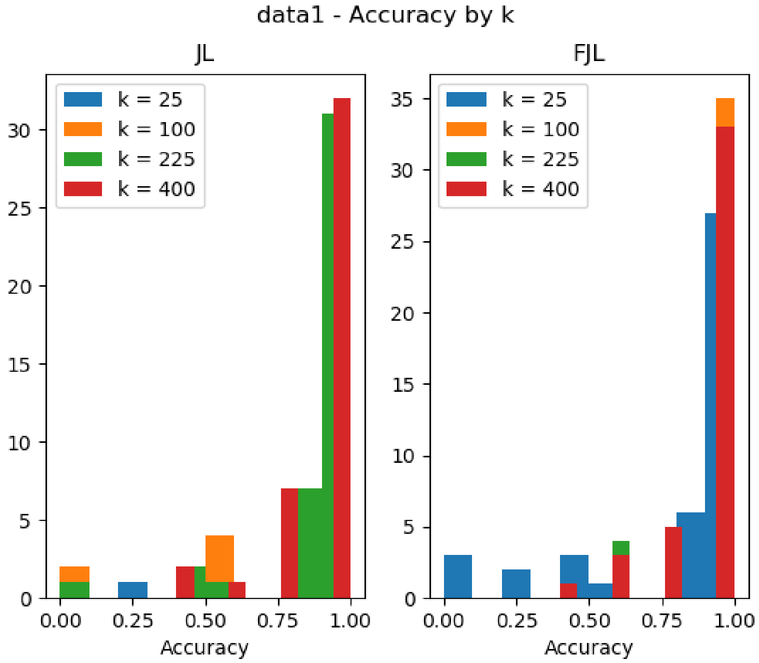

Figure 25: Data 1

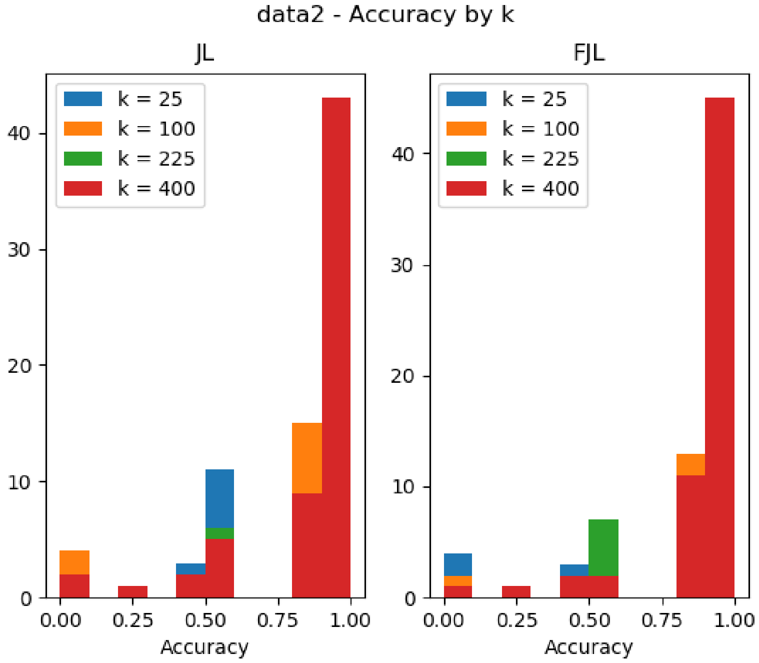

Figure 26: Data 2

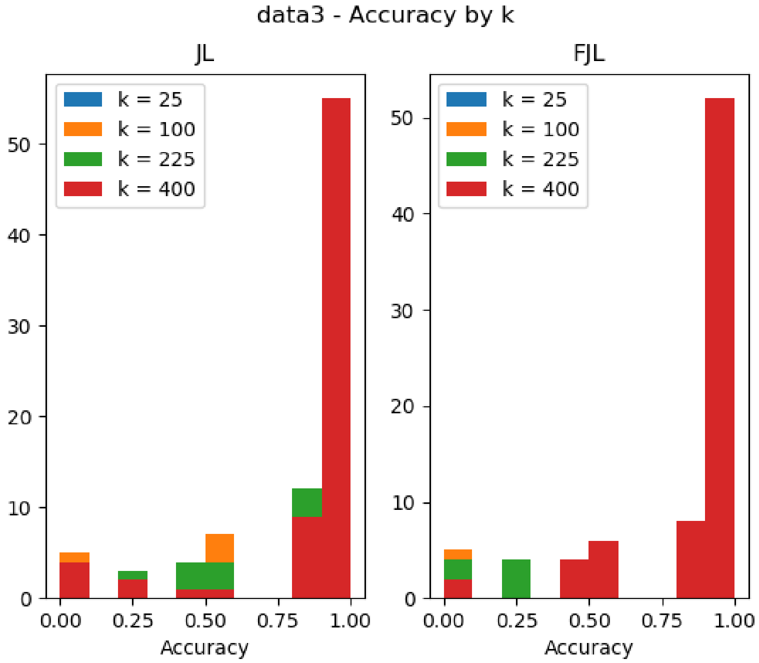

Figure 27: Data 3

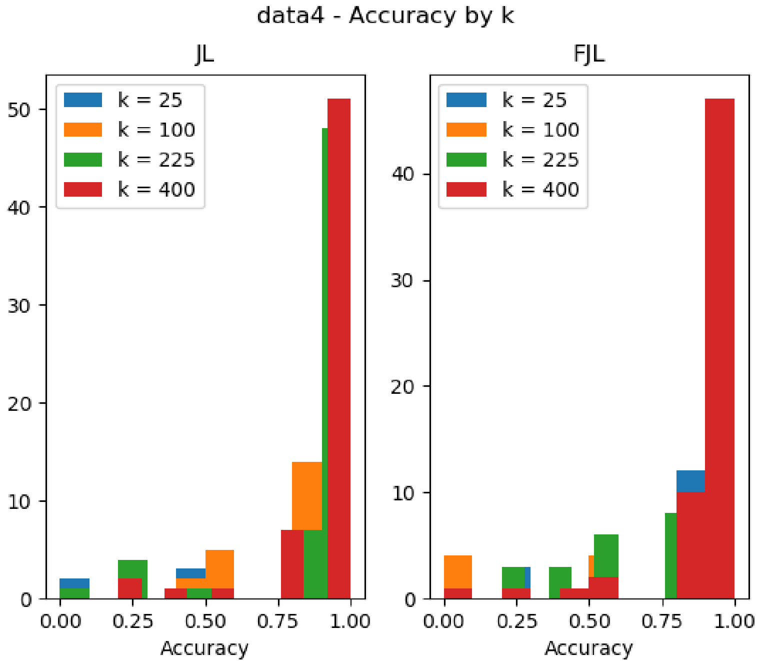

Figure 28: Data 4

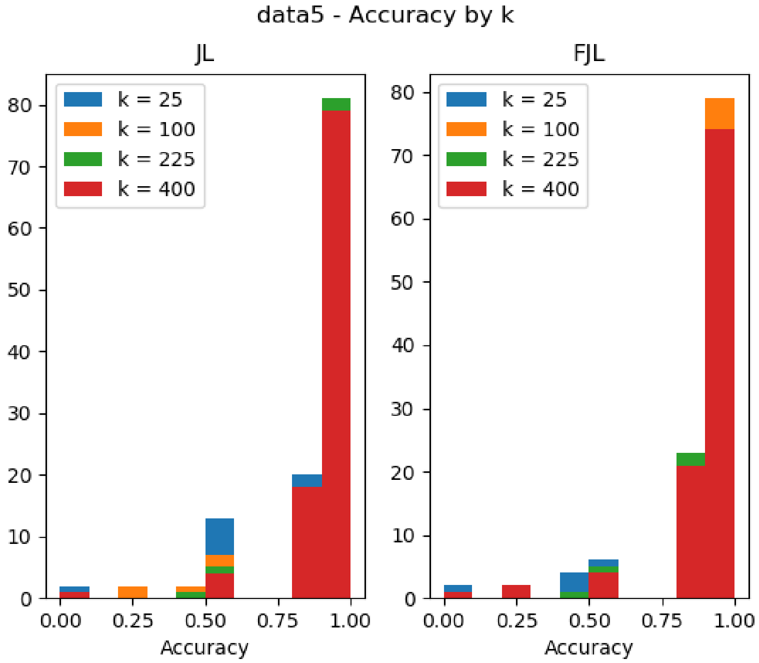

Figure 29: Data 5

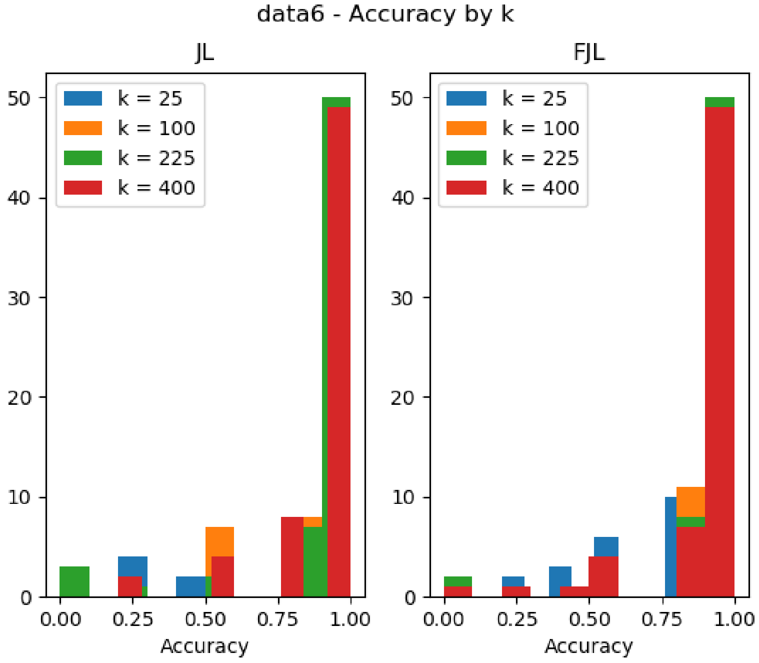

Figure 30: Data 6

4.1 Analysis

Above, the first thing you probably noticed is that for a higher k (less compression), there was an increase in accuracy. This makes sense. You also noticed that it trended downward as k decreased (more compression). Between datasets, all seemed to be equal in terms of their distribution.

Overall, this affirms our intuition about dimensionality deduction. It also shows that the datasets generally do not matter.

5 Accuracy and Timing of Nearest Neighbor

Below we graphed the average average and calculated time for each dataset and k combination. This was done for both the JL transform as well as the FJL transform. Thus, we have have total plots.

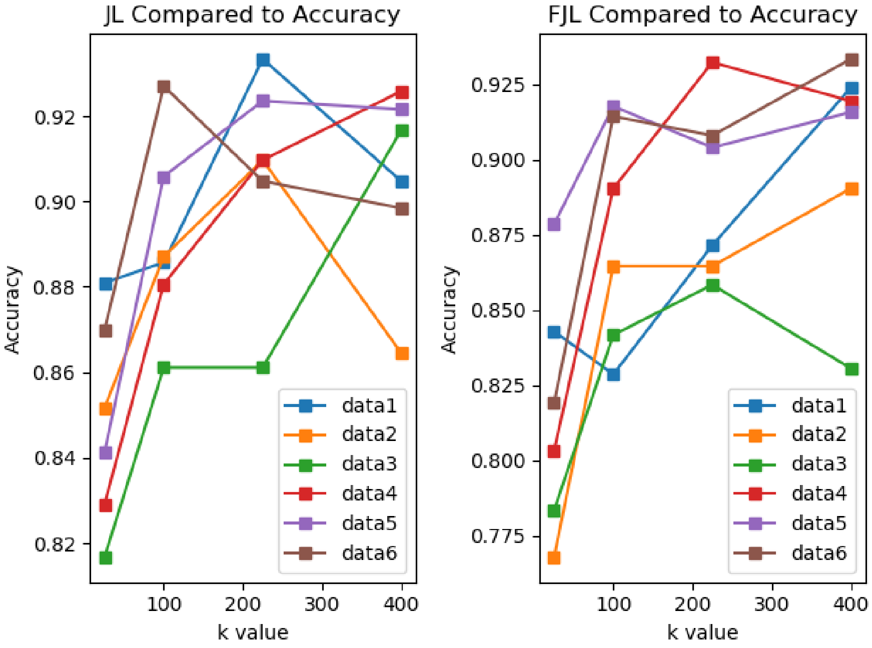

Figure 31: Accuracy Plot

5.1 Accuracy Analysis

For the accuracy plot, you will notice that both have pretty similar accuracy which is good since that means it retained most of the information. Looking at the graph, it appears that the dataset is fairly irrelevant in the calculation of the average accuracy. However, as k increases (which means the dataset was less compressed), accuracy increased dramatically.

In short we conclude that compressing your 1) dataset decreases accuracy, 2) datasets are irrelevant on the method, and 3) JL and FJL do not make a noticeable difference.

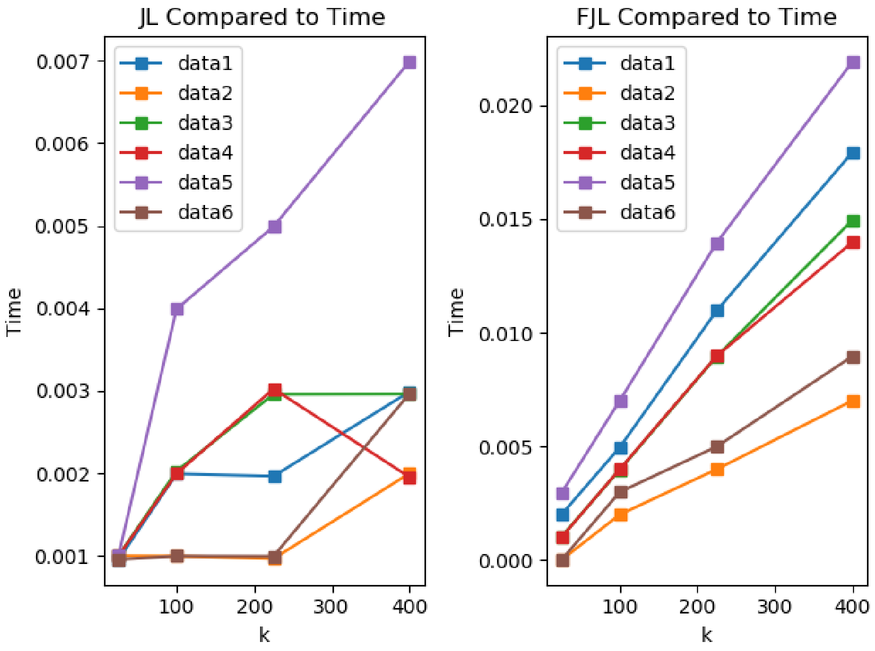

Figure 32: Time Plot

5.2 Time Analysis

For the time plot, we notice many differences. The first prominent feature is that when FJL is applied, the time scales linearly with k. When FJL is not applied, it generally increases but there is not as distinct of a pattern. We also notice that data5 is the most computationally expensive while data2 is the least. Another thing to notice is that FJL is actually slower than JL. This seems counter intuitive and wrong but I believe I implemented the code correctly, so I am not sure what happened.

— END —