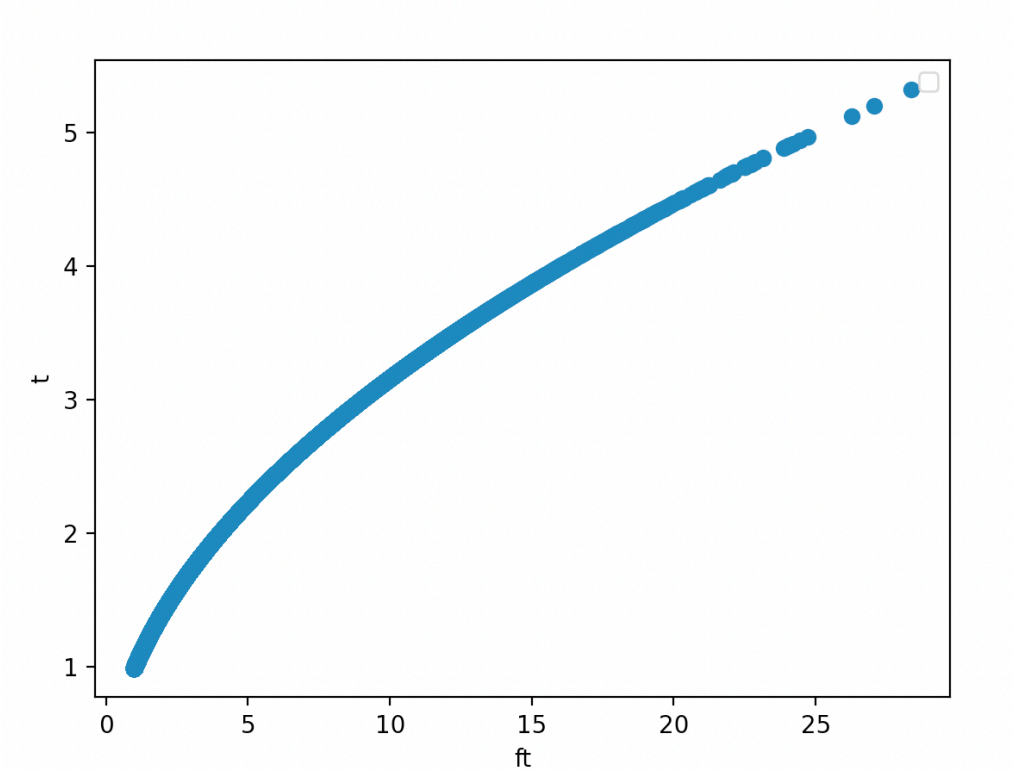
2022-2023 Fall Semester
IE452 Project 1 Report

Efe Umut Ateş 21801611
Doruk Barkın 21802255
Sena Erçoban 21501172

Question 1

For question 1 we were asked to generate a dataset of 100,000 vectors that are in R^100. This meant that we needed a dataset with 100,000 rows and 100 columns to end up with an MxN matrix of 100,000x100. We chose the data points t between 0 and 1 as randomly generated values. After doing so, we compared every value according to the equation $(100 - 100t, 100 + 100t)$ and we found the appropriate $f(t)$ values. The following graph is the generated graph of $f(t)$ against t, for further details we advise to check the python code:

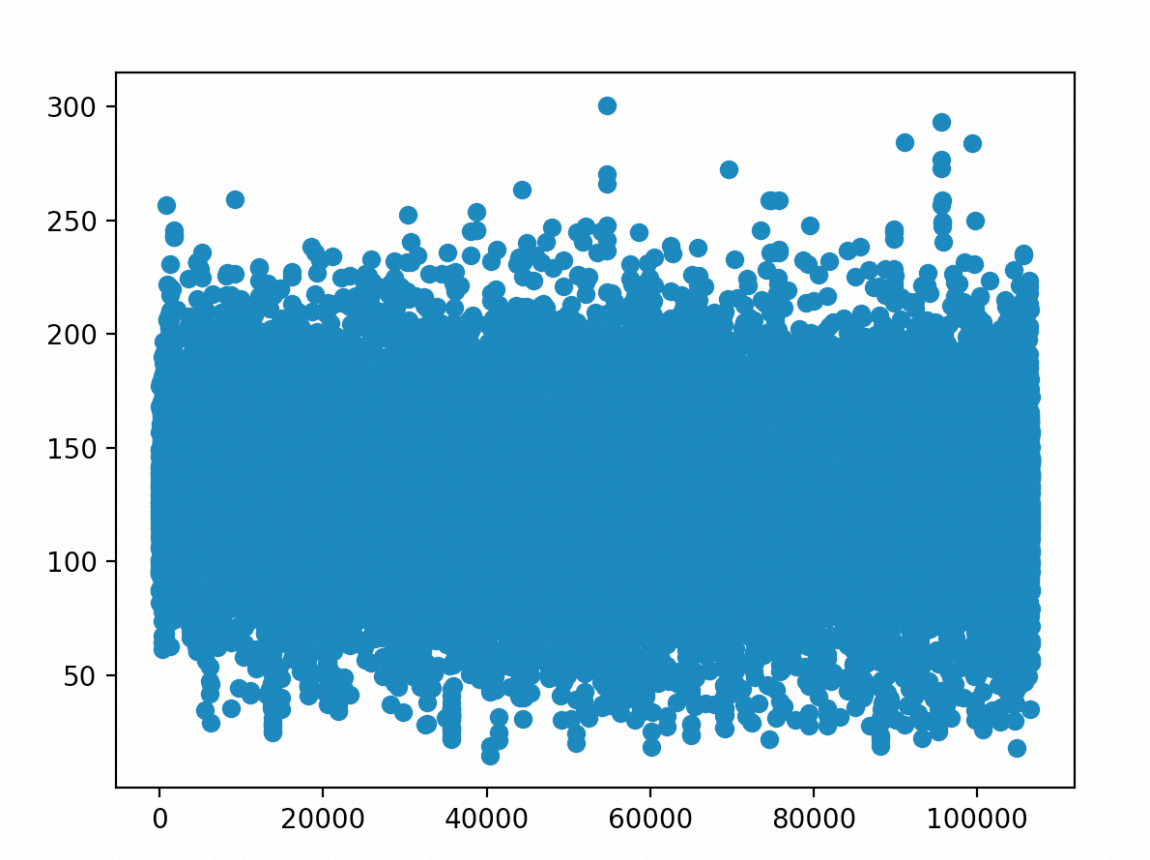


As seen in the graph, after a certain threshold which is 5, data is becoming scarce and the increasing trend in the graph seems to decline.

Question 2

We treated our dataset as an MxN matrix with 106379x519 (we have used the features.csv provided in the GitHub link in fml.txt). First, to apply the Johnson-Lindenstrauss lemma, the error rate (ε) too small with a value of 0.05 so we went with an alternative route.We used an R matrix that was initialized with Gaussian Random Projection method. The N component of the projection was

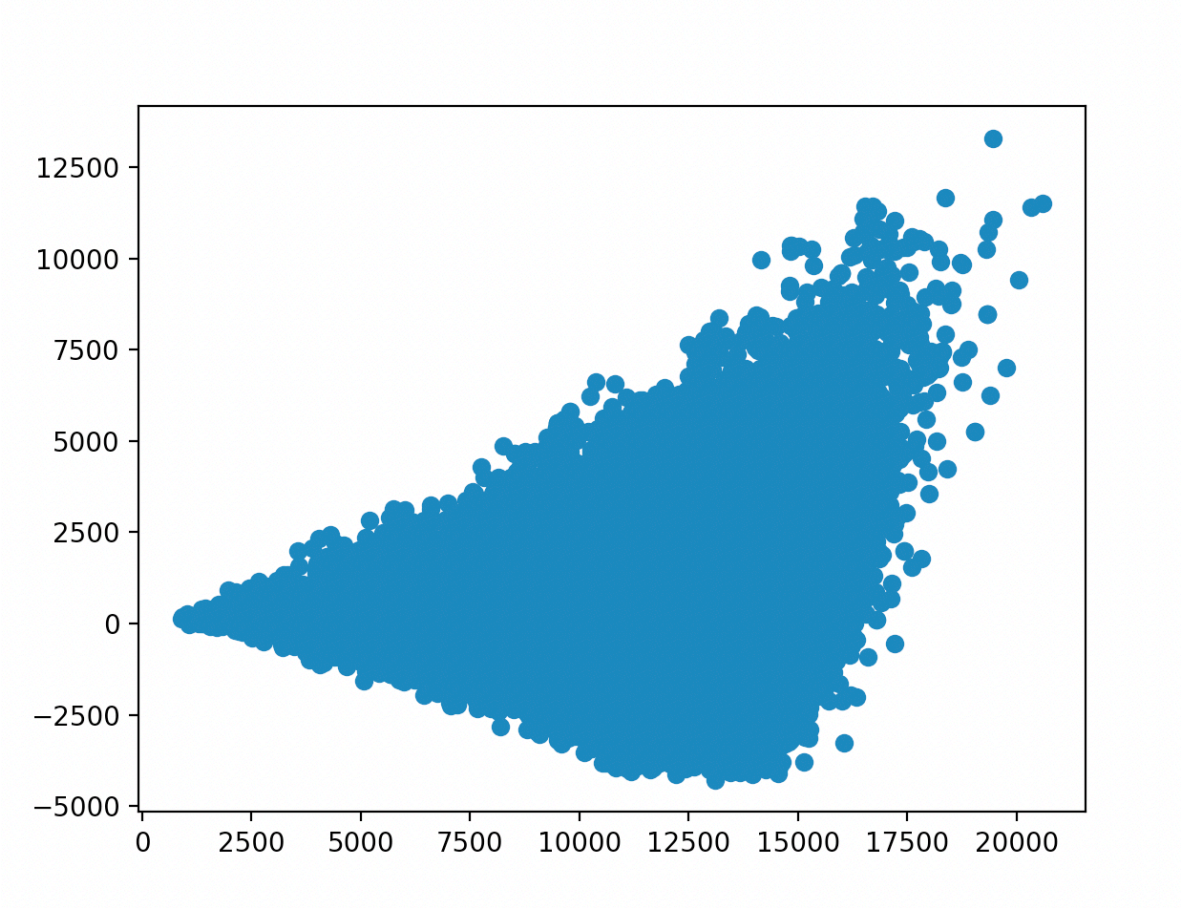
344 which was calculated from the Big O Notation in the project description pdf. After normalizing the R matrix, we performed a matrix multiplication with the R matrix and the dataset D. We couldn't tell the projection was sufficiently accurate because we were reducing 519 components to 344. The following graph is the best fit of dataset D on the subspace S:



From this graph, we can deduce that the best fit line is along the 150 component level. Moreover we can also see that as the component numbers go above and below this level the data points in the projection get gradually low. As the sample size increases it becomes easier to reduce the dimensions of the data using Random Projection method that is based on Johnson-Lindenstrauss lemma.

Question 3

We performed a singular value decomposition using the truncated SVD method that is available in the Python's Sci-kit Learn library. We chose the components (N) of the transform matrix as 2 because it was easier to plot a fitting line. The following graph shows our results:



As we can see from the graph, if our samples increase, the dimensionality interval also tends to increase. We also think that SVD is a better algorithm because the graph in SVD was much more clear when compared to Question 2 and was easier to understand.