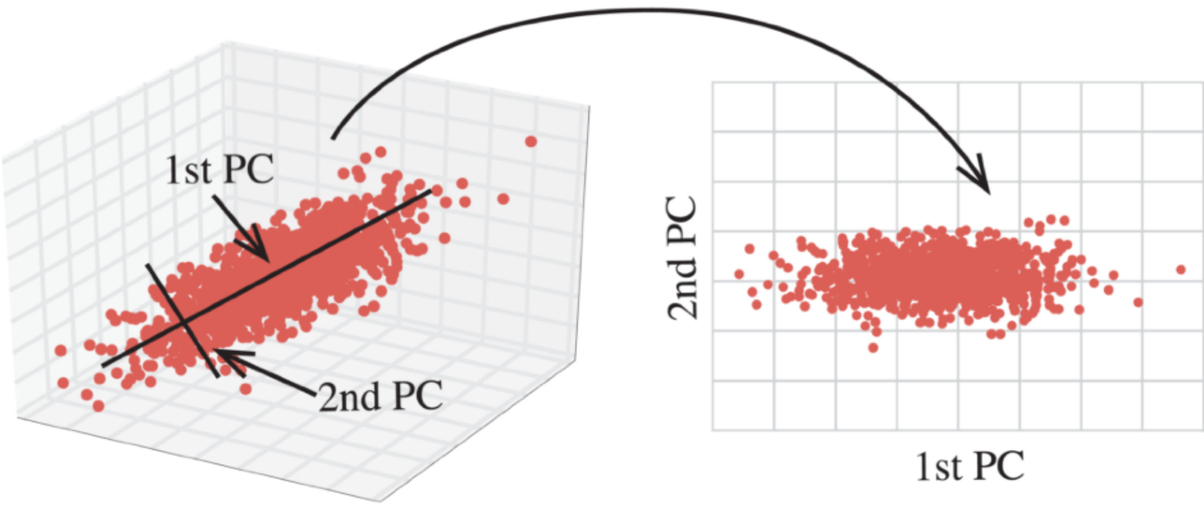
# Principal Component Analysis

## DSAA Assignment 2, Question 4




Abhinav Gupta

20171059

# Report

## What is PCA?

Principal Component Analysis is a technique that helps us in Dimension Reduction. It finds the combinations of variables that help explain the most amount of variance. Mathematically, it is an orthogonal linear transform that transforms the data to a new coordinate system. In this new system, the greatest variance by some projection of the data lies on the first coordinate - the first principal component, the second greatest variance on the second coordinate and so on and so forth.

## The Implementation

We first read all of the images from the dataset given and store them in arr. And then we calculate the covariance of its transpose, and then find its eigenvectors and eigenvalues.

Here is a snippet of the code:

```
transpose_mat = arr';
covariance_mat = cov(double(transpose_mat));
[vectors, values] = eig(covariance_mat);
```

Then we sort the diagonal of the eigenvalues in descending order and map to corresponding eigenvector, to compute the norm_mat.
520 - number of dimensions (65x7)
Code Snippet:

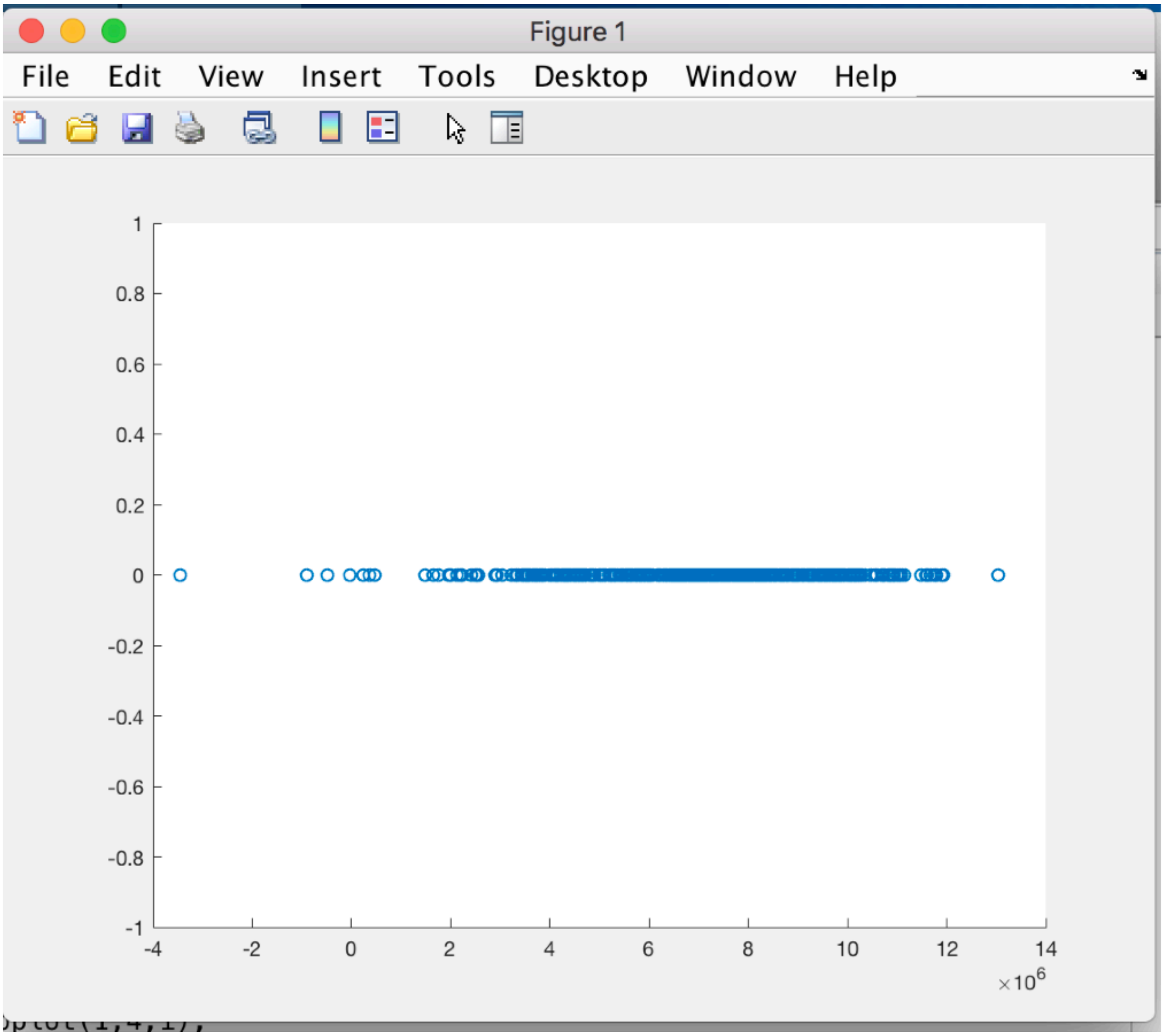
```
[a,b] = sort(diag(values), 'descend');
norm_mat = normalize( double(transpose_mat) * vectors(:, b(1:520, :)));
```

So now, **if we multiply with norm_mat, we can compress the image**. And if we multiply by its transpose, we can decompress it back.

```
compress_mat = double(arr) * norm_mat;
decompress_mat = compress_mat * norm_mat';
```

We then use scatterplots to study how these images are clustered in the 1D, 2D and 3D spaces, using required number of principal components. Here the screenshots of the scatterplots.

1D:




2D:

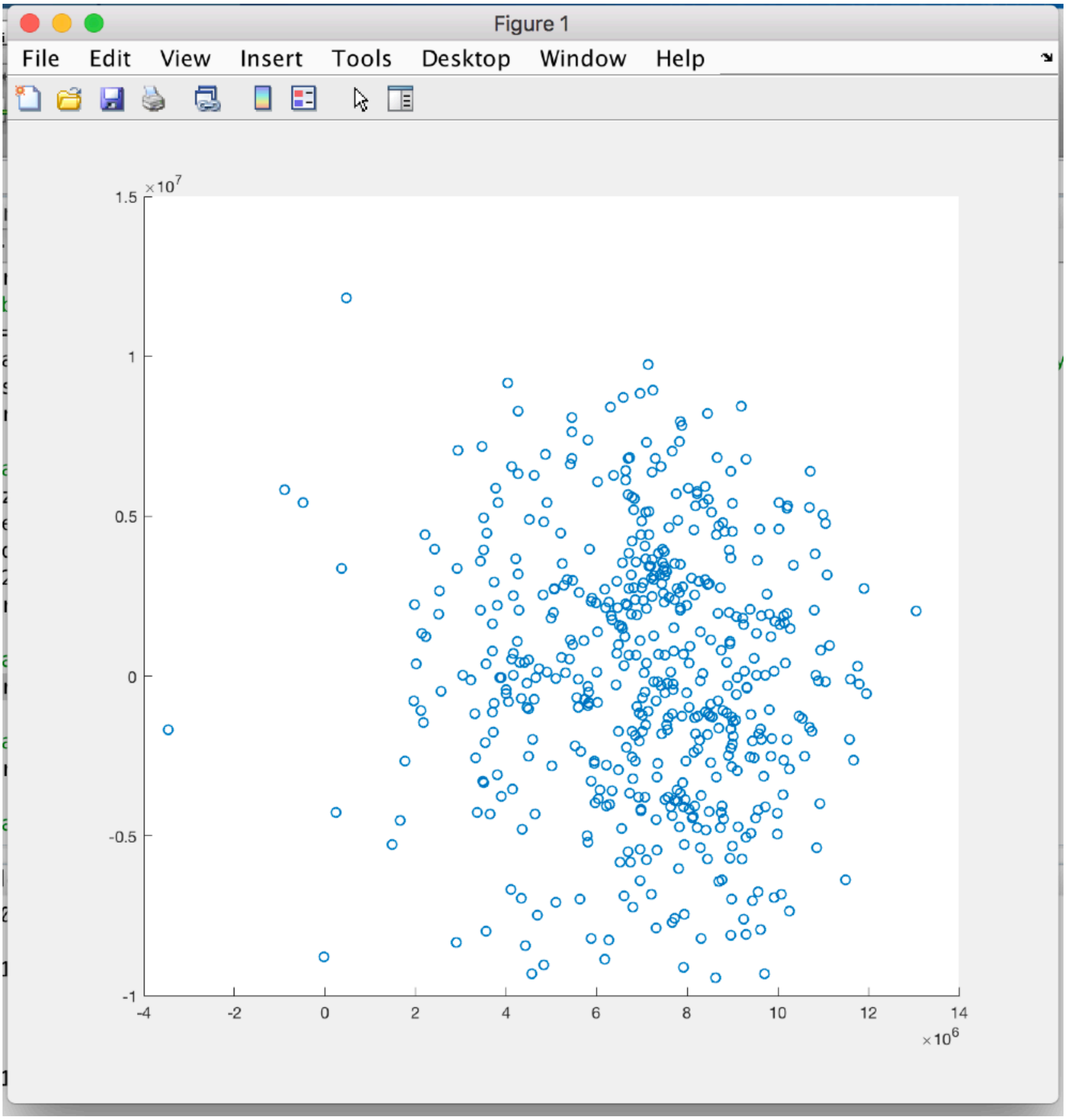
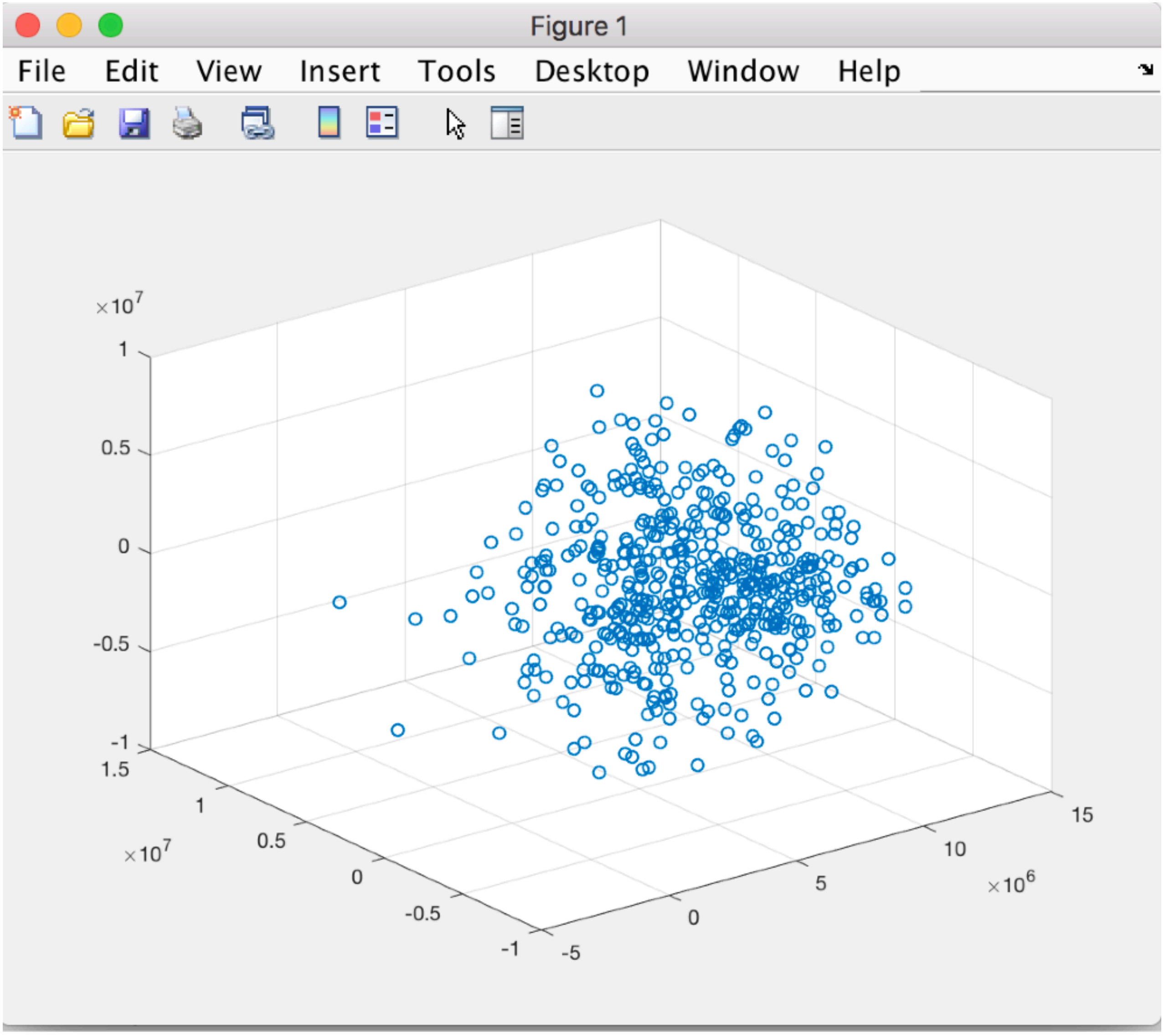


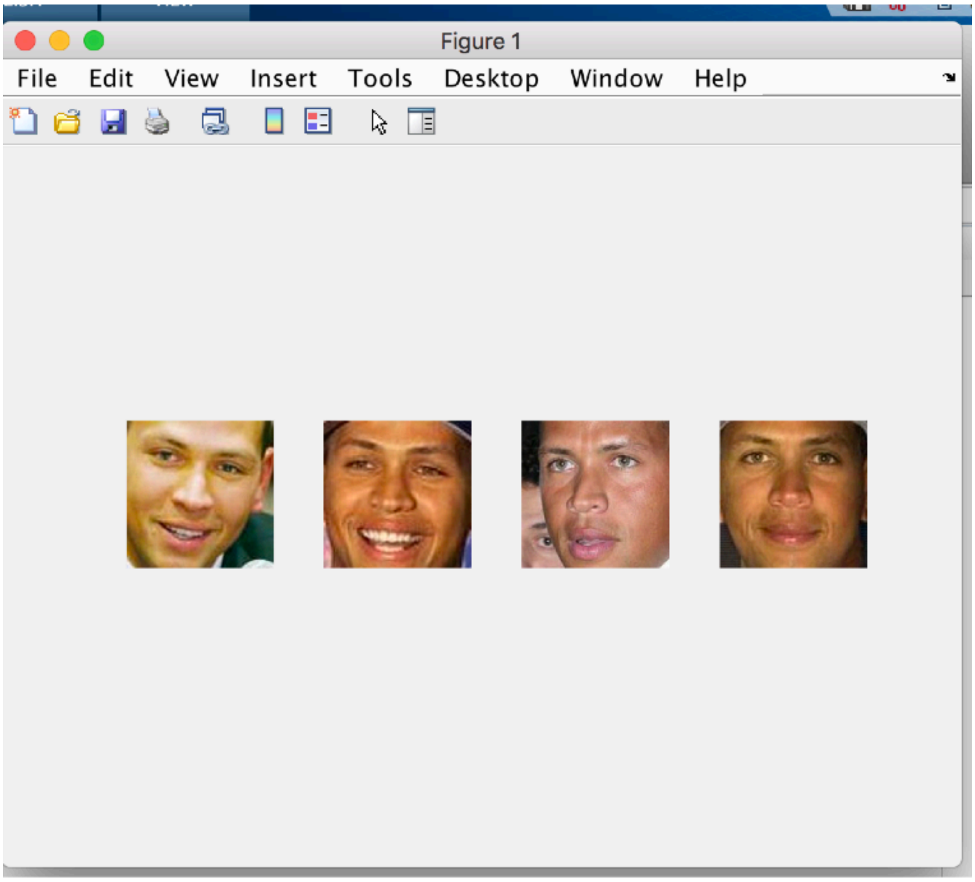
Figure 1
File Edit View Insert Tools Desktop Window Help
×10^7
1.5
1
0.5
0
-0.5
-1
-4
-2
0
2
4
6
8
10
12
14
×10^6

3D:




If we keep number of dimensions as 520, we would not lose any data, because we are breaking a 520 array into 520 eigenvectors itself. So we can retrieve it back completely. But if we reduce the number of dimensions, then it will be LOSSY compression. So when we construct the images, they will not be very clear as some information has been lost.

When number of dimensions is 520:




When we reduce it to 30: