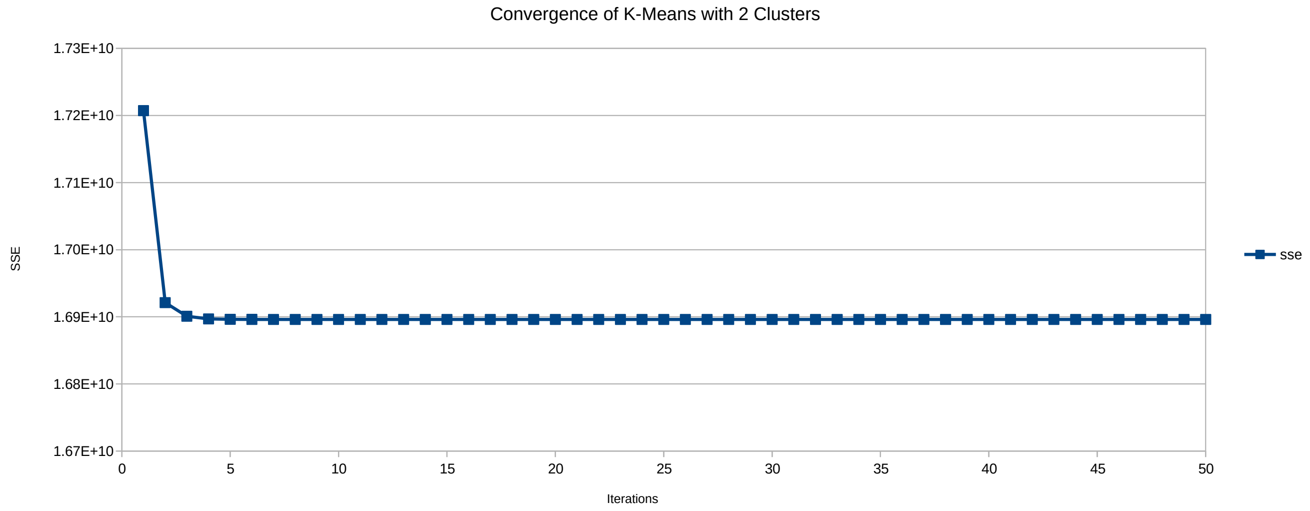
Implementation Assignment #3

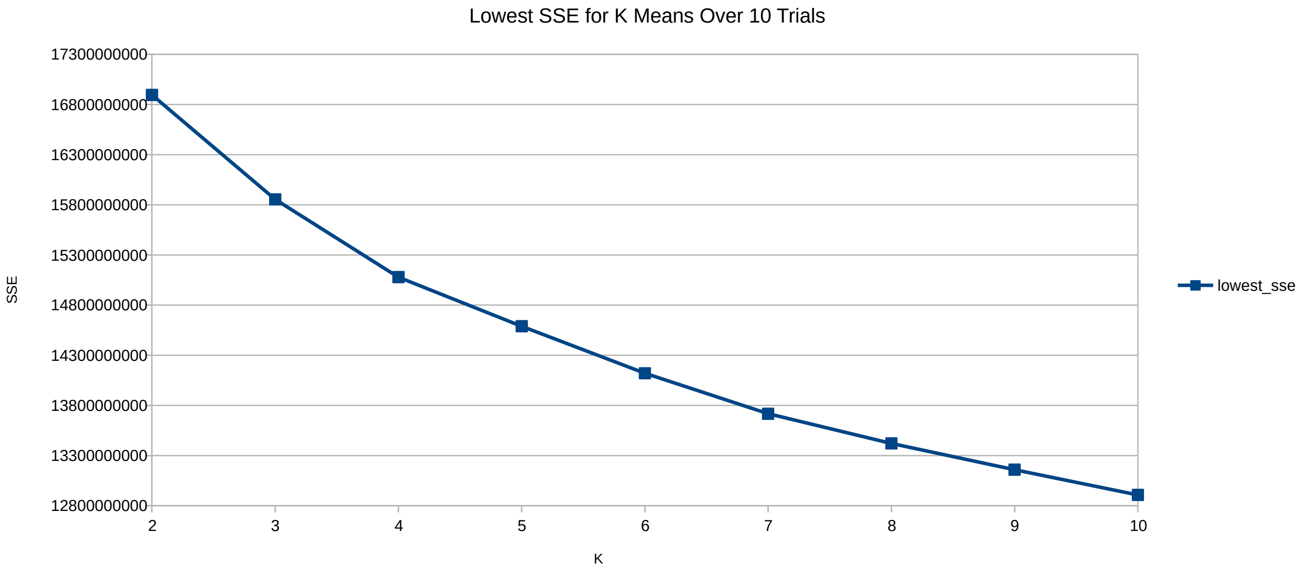
Trevor Hammock, Charles Koll

## Part 2

1. *Implement the K-means algorithm. Run your K-means algorithm with k = 2. To verify that your algorithm actually converges, please plot the objective of the K-means algorithm (i.e., the SSE) as a function of the iterations. From one run to another run, this curve may look different. Just present the results of a typical run.*



2. *Now apply your K-means implementation to this data with different values of k (consider values 2, 3, · · · , 10). For each value of k, please run your algorithm 10 times, each time with a different random initialization, record the lowest SSE value achieved in these 10 repetitions for each value of k. Plot the recorded SSE values against the changing k value. What do you think would be a proper k value based on this curve? Please provide justification for your choice.*



The larger the k value, the lower the SSE. Since k only changes how many clusters the data is binned into, more of them allows for tighter groupings, which leads to lower SSEs. In the scenarios presented, we would choose 10 because it is the largest k value.

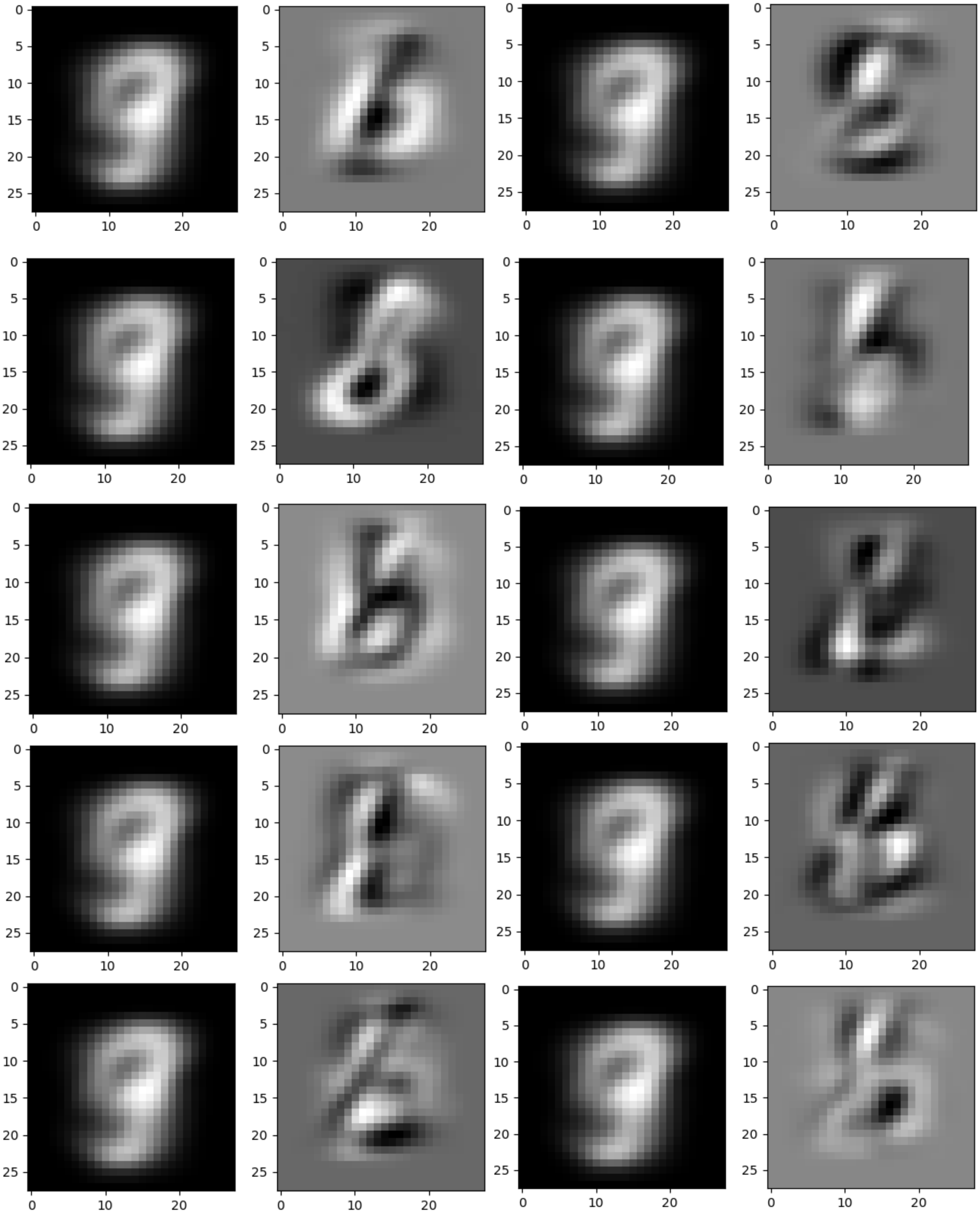
**Part 3**

1. *Implement Principal Component Analysis for dimension reduction. Specifically, your program needs to compute the mean and covariance matrix of the data, and compute the top ten eigen-vectors with ten largest eigen-values of the Covariance matrix (you can use existing functions in numpy to compute the eigen-values and eigen-vectors). Report the eigen-values in decreasing order.*

| Top 10 Eigen Values |
|---|
| 12347508 |
| 2614365 |
| 1934458 |
| 1349383 |
| 1140525 |
| 937976 |
| 814087 |
| 695348 |
| 683781 |
| 552380 |

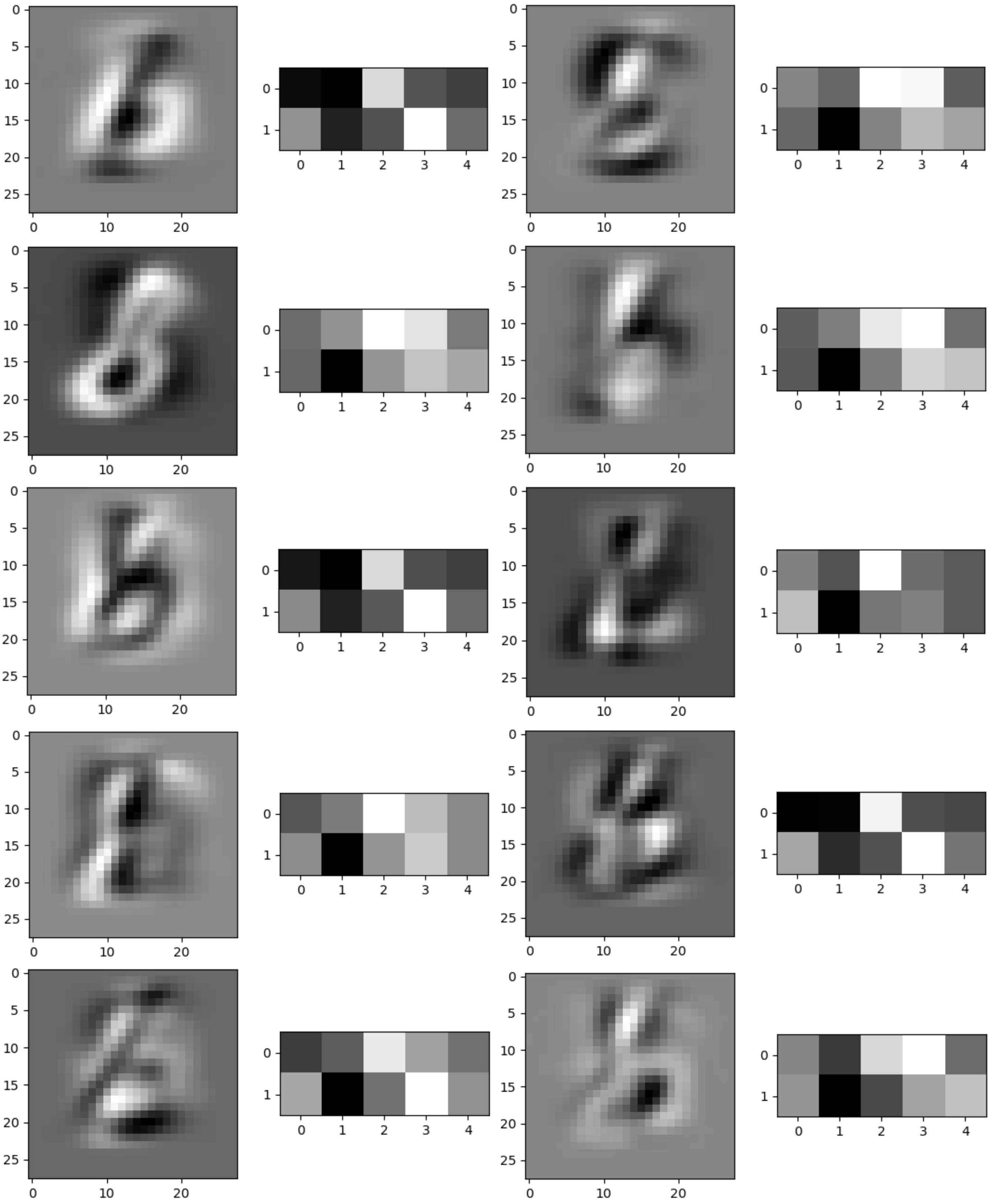
2. *Plot the mean image, and each of the top ten eigen-vectors. To make the image for eigen-vectors viewable, you should re-scale each eigen-vector by its maximum value. Inspect the resulting images, what do you think they each capture?*

The images capture the various features of the digits and groups them into separate categories.


0
5
10
15
20
25
0
10
20

3. *Use the top 10 eigen-vectors to project each image to 10 dimensions. Identify for each dimension the image that has the largest value in that dimension and plot it. Compare the image with its corresponding eigen-vector image, what do you observe? What do you think the reduced 10-dimensional representation is capturing in this case?*

The 10-dimensional representations look like compressed versions of their corresponding eigen-vector images. We believe they portray the most commonly used colors for their paired images, which represent features.


0
5
10
15
20
25
0
10
20
0
1
0
1
2
3
4