K-Means Clustering Lab

Group members: Krystal Ly and Minh Ta

Question 1

How many different types of hieroglyphics do you see?

When we first viewed the unzipped images, we found out several noticeable groups based on our intuition:

+ Birds

+ Parts of human body (eyes, legs, arms, etc)

+ Animals (rabits, deers, snakes)

+ Tools (knifes, swords, axes, ropes)

+ Coins

+ Flowers (?)

=> Intuitively speaking, we thought there were approximately 15 groups, but some groups had fewer images than others. This shouldn't have been an accurate estimation of all the hieroglyph groups because we didn't have a way to document all images throughout the process and the amount of time spent was not significant to go through all 4410 images thoroughly.

To get a basic understanding of hieroglyphs and how to classify them properly, we did spend some time reading the original source. We went through the book Egyptian Grammar: Being an Introduction to the Study of Hieroglyphs by Sir Alan Henderson Gardiner that was cited in the original source. The book classified hieroglyphs into 27 groups with the last group being Others. The groups can be a wide range from maritime tools and vessels, birds, common staples, ground facilities and equipment. We cross checked with the paper and decided the number of clusters should be at around 15-35.

Question 2

Read in all the images in R and store them as a single data frame

Code provided in the Rmd file.

Question 3

Compress the data with PCA

Code provided in the Rmd file.

We choose to use 557 components as those provide 95.006% of the variance in the data. This number of components has shrunk the data by 3750/558 times. It reduces the time needed for computation and retained values that are important. This is especially true because the white space in each image is quite large, and we can remove them for better time usage while still have the data needed in the center for our analysis.

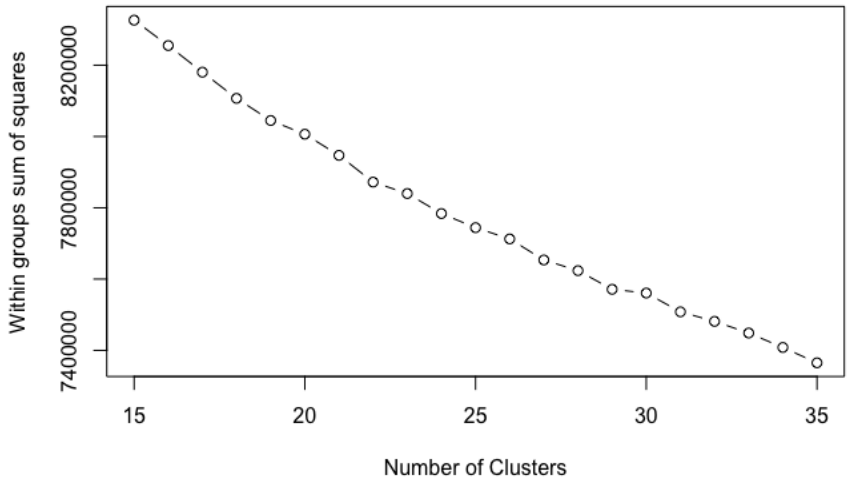
Question 4

Run some k-means clustering algorithm

Code provided in the Rmd file.

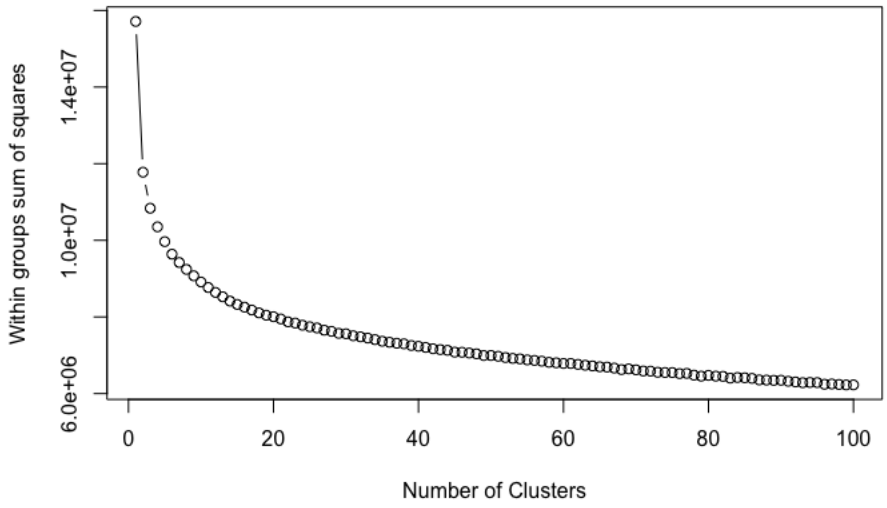
A good k is the one that minimizes within cluster sum of squares (WSS) and maximizes between-cluster sum of squares (BSS). However, there is a trade-off between the value of k and the computation time.

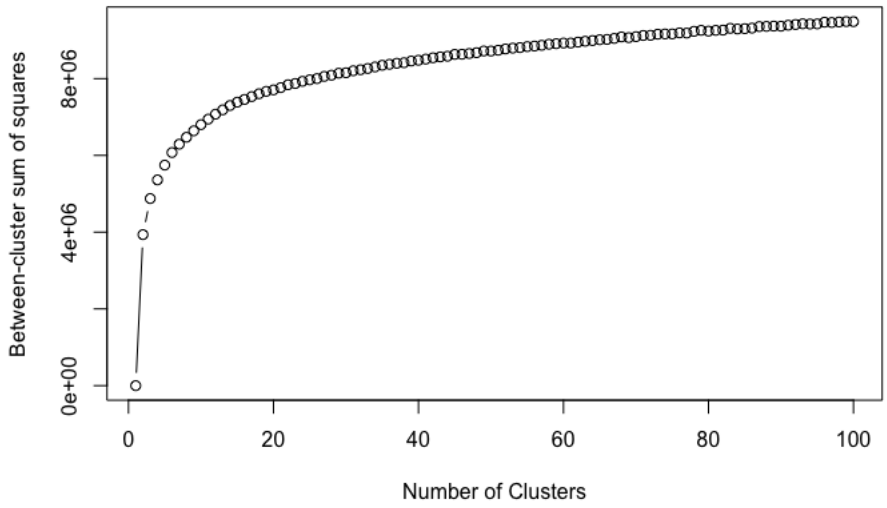
We try ranging k from 15 to 25 as we stated in Q1. The plot of the WSS against k is as below:



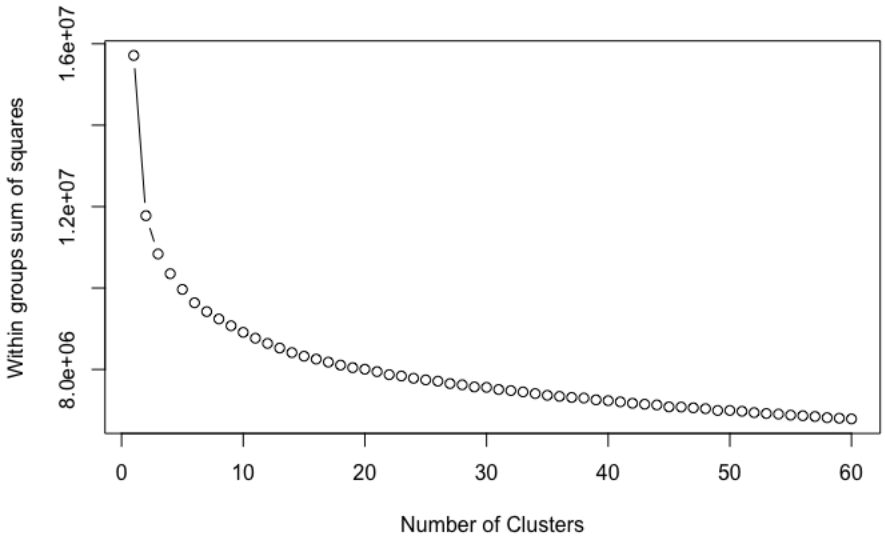
As k increases, WSS decreases gradually with no major improvements. Given that it doesn't take too much time to run from k =15 to 35 and the fact that the plot above doesn't give us enough information to consider the value of k, we try ranging k from 1 to 100. Along with the WSS, we also save and plot the BSS.

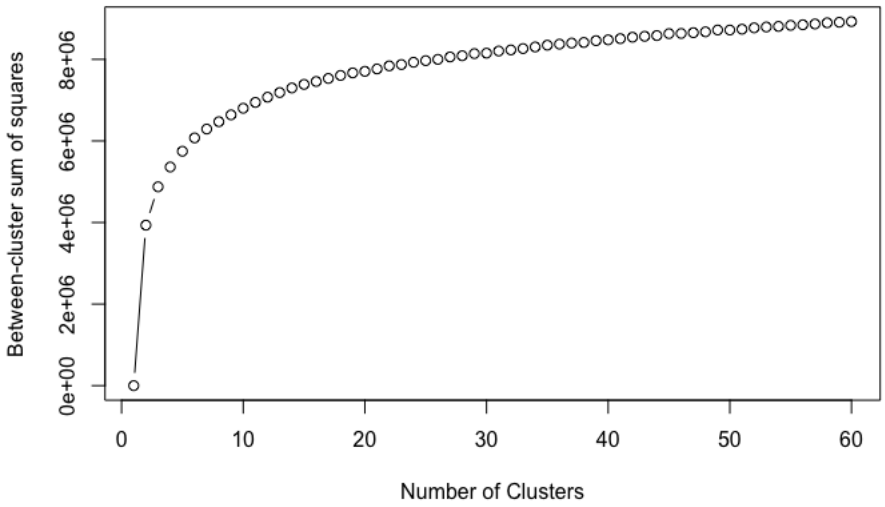
Note: We ran into a problem with the iter.max parameter. Particularly, we got a warning saying that it didn't converge in 10 iterations. The kmeans function doesn't iterate the process of relocating centers until convergence but only iterating 10 times (the default iter.max) so we have to change to iter.max = 20





It appears that starting from k = 60, WSS doesn't improve much. The BSS plot also shows that BSS improves most from lower values of k so we zoom in to see k from 0 to 60.





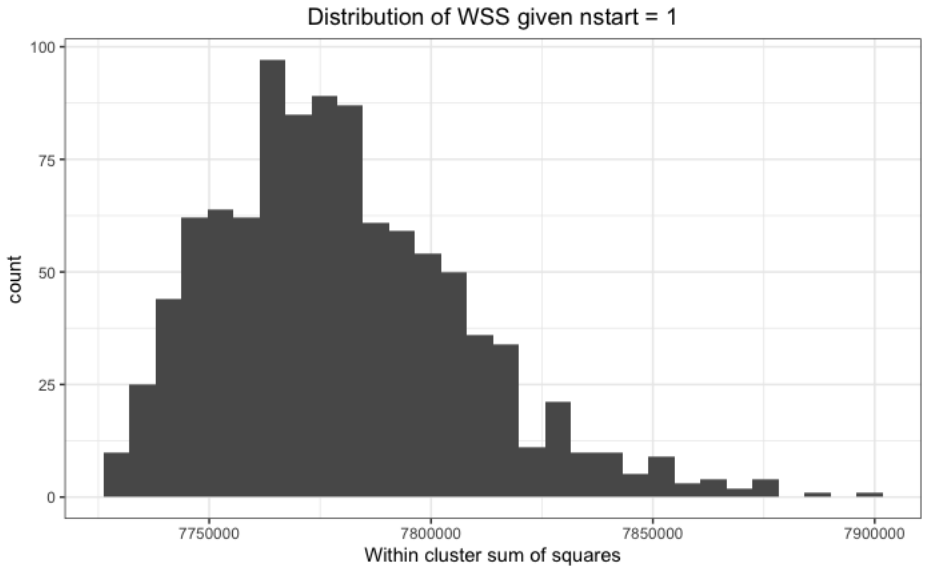
The plots show that k = 3 has the greatest improvement. However, that number would be too small to classify 4410 images. We finally decide on k = 25 because:

- This is a fairly large enough number of k to classify 4410 images.
- Although the improvement is not as great as with smaller number of k, there is still great improvement of WSS when k = 25.
- The book cited in the original source reasonably classified hieroglyphs into 27 classes.

Question 5

Perform the kmeans with your chosen k 1000 times, each time with 1 start

Code provided in the Rmd file



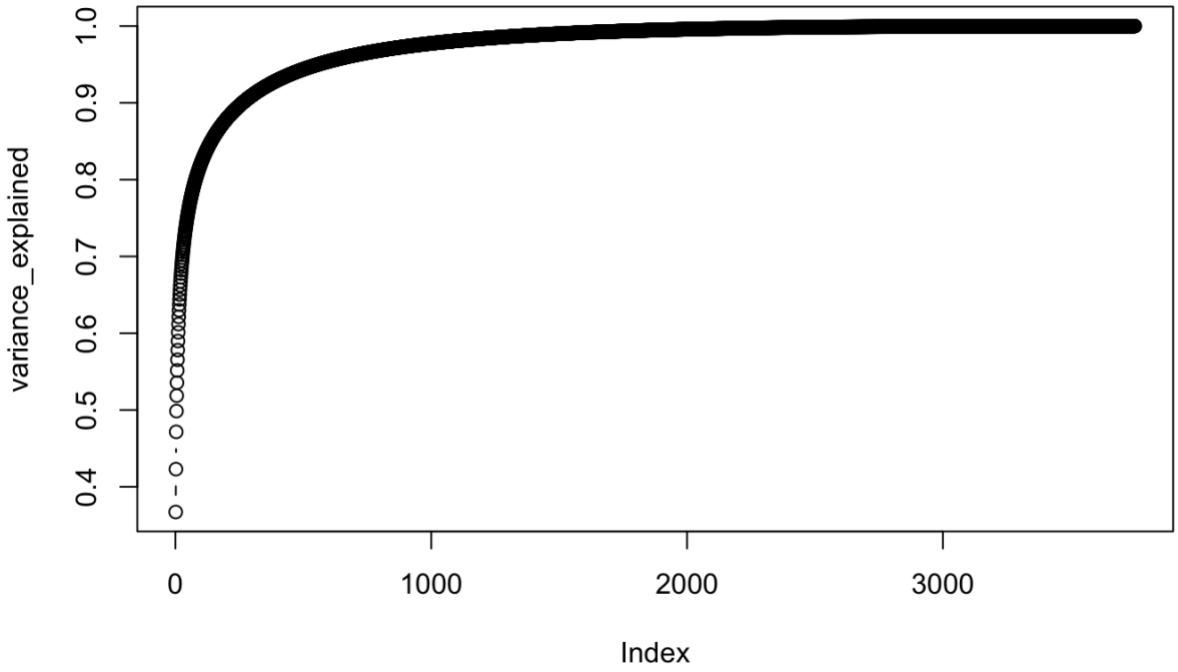
When nstart = 1, WSS has a normal distribution with the most frequent WSS not the smallest WSS possible. This means that if we train the model again with nstart = 1, the WSS that we get is highly likely not to be the lowest possible one that we could get. Therefore, nstart = 1 is not enough. We will finalize our model with nstart = 20.

Question 6

Write each image to the folder corresponding to it's assigned cluster.

Code provided in the Rmd file

When we did Q7, we received bad predictions so we tried changing the parameters in the model and trained it again multiple time with different number of k (k = 15, 20, 25) and nstart. k = 25 yields the best result when we looked at the folder of each clusters. However, prediction was still really bad. Therefore, we tried increasing the number of principal components (PC).



2000 PCs seem to be a good number as larger than that, the proportion of variance explained doesn't increase much. 2000 PCs explain 99.59% of the variance of the data which is quite a decent amount of data.

The prediction improved significantly, and the computation time wasn't too bad. Using a smaller number of PCs may help us save the computation time because the white space of the images is quite large. However, as the original images are quite blurred and the background is not entirely white but grey, too few PCs may capture some noise (background) instead of the hieroglyphs but still have a high percentage of variance explained.

Our final model was trained with 2000 PCS, k = 25, nstart = 20 and iter.max = 20.

Folders of clusters were submitted along with this lab report

Overall, there are some clusters that looks great but some looks awful with visually different hieroglyphs. Also, some clusters can be grouped together into a larger cluster, e.g. bird, knife. The problem seems to be the brightness of the images. It affects how images are clustered. Images that have the same level of brightness tend to be grouped together. The algorithm can't distinguish the background from the hieroglyphs well. It may be better if we could find a way to extract the hieroglyphs out of the background.

Question 7

Write a function to predict which cluster new images belong to.

Code provided in the Rmd file

We pick a cluster for new points by comparing the Euclidean distance between that point and all cluster centers. The point would be added to the cluster with lowest distance.

We came across challenges when the cluster centers were stored in the form of PCs instead of the original pixel information. Therefore, we had to scale the new data using the loading vectors using predict function. The prediction results are as follows:

Image	Cluster
030036.png	19
030135.png	16
050188.png	5
050189.png	5
200081.png	10
200084.png	8
410399.png	11
410400.png	14
410401.png	20
410402.png	4

Clustering is doing a good job. In each cluster the images assigned to, we can see some images that are identical to the images used for prediction. Still, there are some clusters that have a wide range of visually different hieroglyphs affected by the brightness level that makes the prediction less accurate.

Contributions of group member

We have two members in our group: Krystal Ly and Minh Ta. We work together in most of the problem. One was coding, while the other checked the code before running and vice versa. Some complicated coding part was done by Krystal. However, when Krystal was working on these parts, Minh worked on the interpretation part to save time. We both read through the report and finalized it.