

```
# Week 6 - Recitation
```

```
# Video 2
```

```
flower = read.csv("flower.csv", header=FALSE)  
str(flower)
```

```
# Change the data type to matrix  
flowerMatrix = as.matrix(flower)  
str(flowerMatrix)
```

```
# Turn matrix into a vector  
flowerVector = as.vector(flowerMatrix)  
str(flowerVector)
```

```
flowerVector2 = as.vector(flower)  
str(flowerVector2)
```

```
# Compute distances  
distance = dist(flowerVector, method = "euclidean")
```

```
# Video 3
```

```
# Hierarchical clustering  
clusterIntensity = hclust(distance, method="ward")
```

```
# Plot the dendrogram  
plot(clusterIntensity)
```

```
# Select 3 clusters  
rect.hclust(clusterIntensity, k = 3, border = "red")  
flowerClusters = cutree(clusterIntensity, k = 3)  
flowerClusters
```

```
# Find mean intensity values  
tapply(flowerVector, flowerClusters, mean)
```

```
# Plot the image and the clusters  
dim(flowerClusters) = c(50,50)  
image(flowerClusters, axes = FALSE)
```

```
# Original image  
image(flowerMatrix, axes=FALSE, col=grey(seq(0,1,length=256)))
```

Video 4

Let's try this with an MRI image of the brain

```
healthy = read.csv("healthy.csv", header=FALSE)
healthyMatrix = as.matrix(healthy)
str(healthyMatrix)
```

Plot image

```
image(healthyMatrix, axes=FALSE, col=grey(seq(0,1,length=256)))
```

Hierarchical clustering

```
healthyVector = as.vector(healthyMatrix)
distance = dist(healthyVector, method = "euclidean")
```

We have an error - why?

```
str(healthyVector)
```

Video 5

Specify number of clusters

```
k = 5
```

Run k-means

```
set.seed(1)
KMC = kmeans(healthyVector, centers = k, iter.max = 1000)
str(KMC)
```

Extract clusters

```
healthyClusters = KMC$cluster
KMC$centers[2]
```

Plot the image with the clusters

```
dim(healthyClusters) = c(nrow(healthyMatrix), ncol(healthyMatrix))
```

```
image(healthyClusters, axes = FALSE, col=rainbow(k))
```

Video 6

Apply to a test image

```
tumor = read.csv("tumor.csv", header=FALSE)
tumorMatrix = as.matrix(tumor)
tumorVector = as.vector(tumorMatrix)
```

```
# Apply clusters from before to new image, using the flexclust
package
install.packages("flexclust")
library(flexclust)

KMC.kcca = as.kcca(KMC, healthyVector)
tumorClusters = predict(KMC.kcca, newdata = tumorVector)

# Visualize the clusters
dim(tumorClusters) = c(nrow(tumorMatrix), ncol(tumorMatrix))

image(tumorClusters, axes = FALSE, col=rainbow(k))
```