Exam 3

Alex Whang

2024-12-6

Part 1

a. False
b. False
c. True
d. False
e. True
f. False
g. True
h. True
i. True
j. True
k. True
l. True
m. True
n. True
o. True
p. False
q. False

Part 2

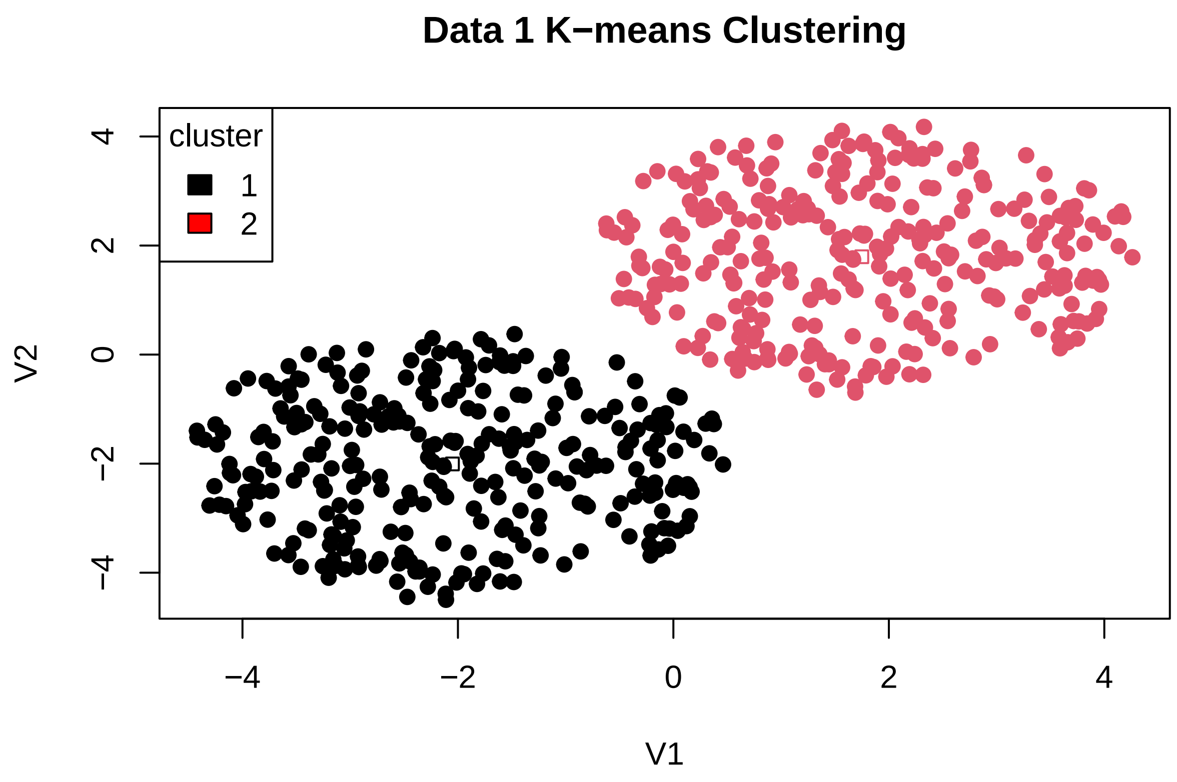
```r
# load data into environment
data1 <- read.csv("data/data1.csv", header = FALSE)
data2 <- read.csv("data/data2.csv", header = FALSE)
```

a. i.

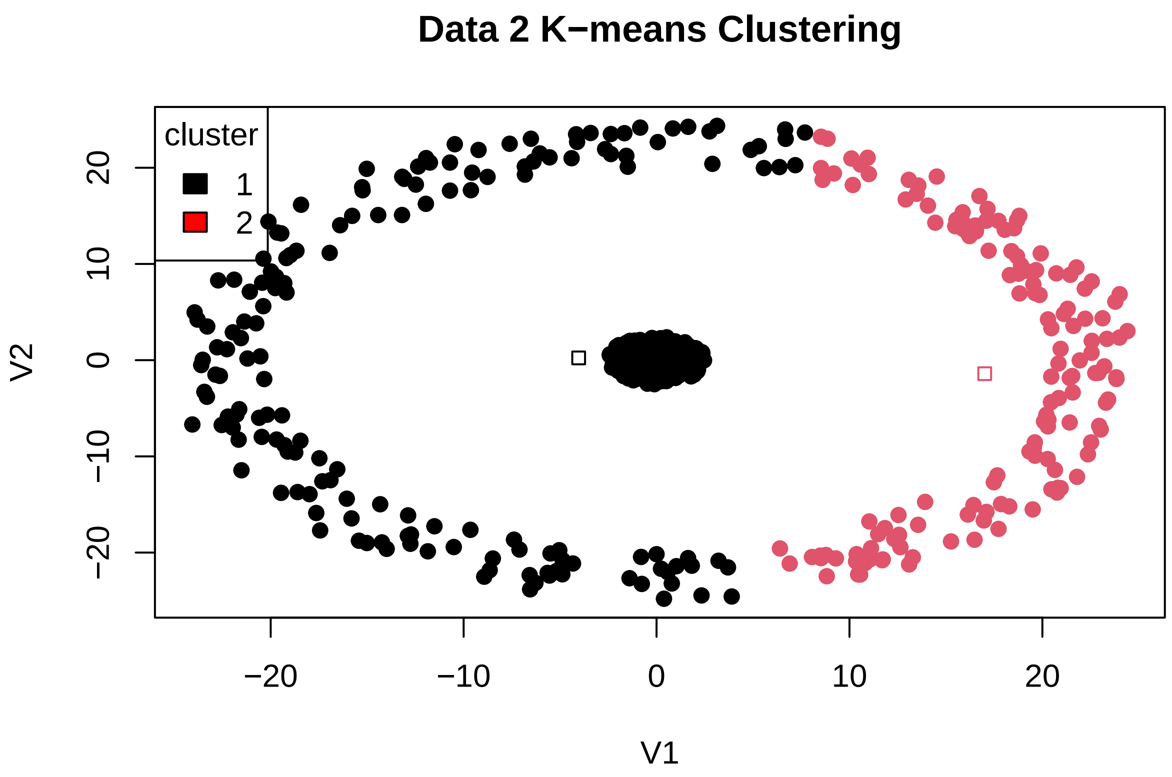
```
# set seed to 1234
set.seed(1234)

# apply k-means clustering function to both data sets
kmeans1 <- kmeans(data1, centers = 2, nstart = 20, algorithm = "Lloyd", iter.max = 100)
kmeans2 <- kmeans(data2, centers = 2, nstart = 20, algorithm = "Lloyd", iter.max = 100)

# plot results for data sets
plot(data1, col = kmeans1$cluster, pch = 19, main = "Data 1 K-means Clustering")
legend("topleft", legend = c("1", "2"), fill = c("black", "red"), title = "cluster")
points(kmeans1$centers, col = 1:2, pch = 22)
```



```
plot(data2, col = kmeans2$cluster, pch = 19, main = "Data 2 K-means Clustering")
legend("topleft", legend = c("1", "2"), fill = c("black", "red"), title = "cluster")
points(kmeans2$centers, col = 1:2, pch = 22)
```



Clusters are indicated by color and their centers are indicated with a square. In these data sets, black represents one cluster and red represents the other cluster.

ii. For data1, k-means performed very well. The clusters are clearly identified and separated and the data is clustered in spherical shapes which k-means will be effective for.

For data2, k-means performed poorly. The clusters aren't very spherical as the data has an outer and inner region almost like a ring. K-means incorrectly divides the clusters as all of the middle region and a lot of the outer region form to make one cluster while the rest of the outer region makes the other cluster. This leads to incorrect assignments of data points as k-means fails to separate the regions.

iii. For data2, k-means failed to correctly cluster the points into meaningful groups. Because this algorithm relies on Euclidian distance to assign points to the nearest centroid, data2 is not suited for k-means because clusters have non-convex shapes instead of spherical shapes. This ring like structure doesn't work with the k-means assumption about cluster shape because of its weird shape.

b. i.

```r
# load mclust package
library(mclust)
```

```
## Package 'mclust' version 6.1.1
## Type 'citation("mclust")' for citing this R package in publications.
```

```r
# fit GMM to data sets
gmm1 <- Mclust(data1, G = 2)
gmm2 <- Mclust(data2, G = 2)
```

ii.

```r
# show gmm1 parameters
gmm1$parameters
```

```
## $pro
## [1] 0.4995303 0.5004697
##
## $mean
##        [,1]      [,2]
## V1 1.739569 -2.038016
## V2 1.782681 -1.990026
##
## $variance
## $variance$modelName
## [1] "EII"
##
## $variance$d
## [1] 2
##
## $variance$G
## [1] 2
##
## $variance$sigma
## , , 1
##
##          V1       V2
## V1 1.575518 0.000000
## V2 0.000000 1.575518
##
## , , 2
##
##          V1       V2
## V1 1.575518 0.000000
## V2 0.000000 1.575518
##
##
## $variance$Sigma
##          V1       V2
## V1 1.575518 0.000000
## V2 0.000000 1.575518
##
## $variance$sigmasq
## [1] 1.575518
##
## $variance$scale
## [1] 1.575518
##
##
```

```
## $Vinv
## NULL
```

Data1:

Mixing proportions: first component is 0.4995303 while second component is 0.5004697

Mean: first component is 1.739569, 1.782681 while second component is -2.038016, -1.990026

Variance values can be seen above

```r
# show gmm2 parameters
gmm2$parameters
```

```
## $pro
## [1] 0.5157589 0.4842411
##
## $mean
##           [,1]         [,2]
## V1  1.9692741 -0.052864050
## V2 -0.2993706 -0.004360617
##
## $variance
## $variance$modelName
## [1] "VII"
##
## $variance$d
## [1] 2
##
## $variance$G
## [1] 2
##
## $variance$sigma
## , , 1
##
##         V1      V2
## V1 246.713   0.000
## V2   0.000 246.713
##
## , , 2
##
##          V1       V2
## V1 1.518602 0.000000
## V2 0.000000 1.518602
##
##
## $variance$sigmasq
## [1] 246.713031   1.518602
##
## $variance$scale
## [1] 246.713031   1.518602
##
##
## $Vinv
## NULL
```

Data2:

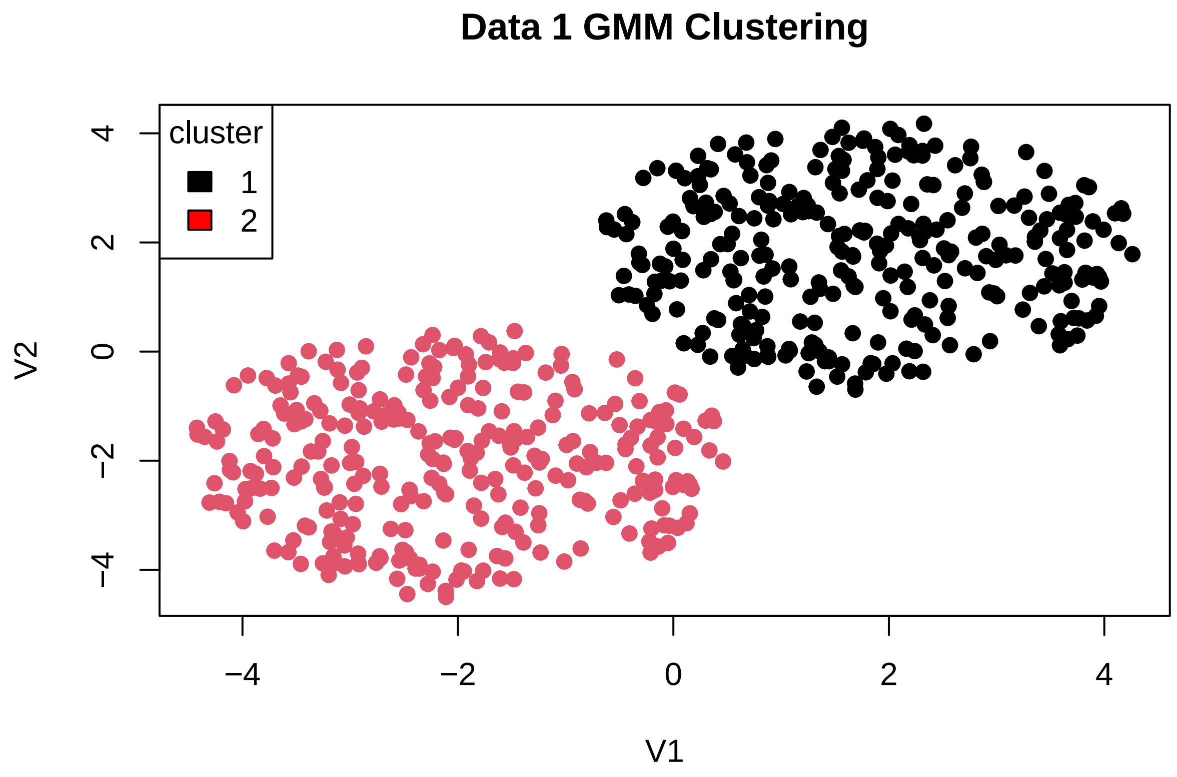
Mixing proportions: first component is 0.5157589 while second component is 0.4842411

Mean: first component is 1.9692741, -0.2993706 while second component is -0.052864050, -0.004360617

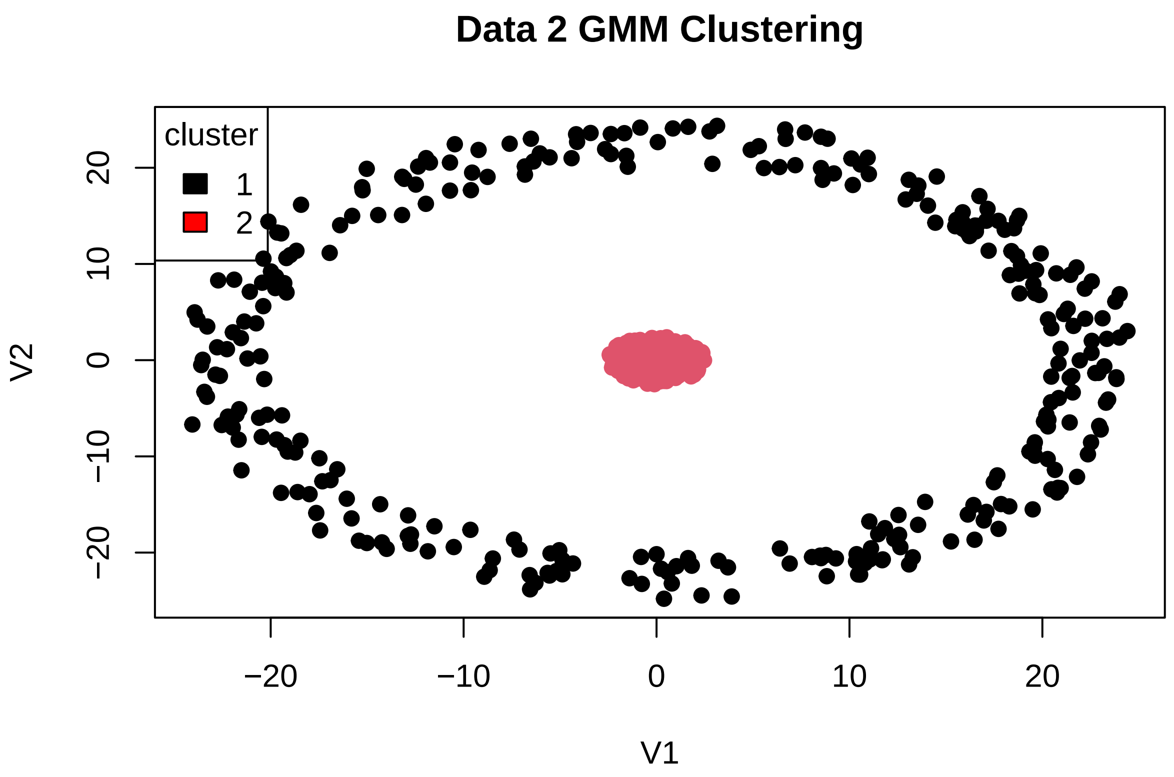
Variance values can be seen above

iii.

```r
# plot results for data sets
plot(data1, col = gmm1$classification, pch = 19, main = "Data 1 GMM Clustering")
legend("topleft", legend = c("1", "2"), fill = c("black", "red"), title = "cluster")
```



```r
plot(data2, col = gmm2$classification, pch = 19, main = "Data 2 GMM Clustering")
legend("topleft", legend = c("1", "2"), fill = c("black", "red"), title = "cluster")
```



c. Yes, there are multiple things that suggest that modeling the second data set as a mixture of Gaussians is unrealistic. Although the GMM package I use modeled the data well, GMMs typically cannot handle circular data like there is in data set 2. Additionally, because each cluser is modeled by a Gaussian distribution with parameters in a GMM, the EM algorithm might try to force one Gaussian to the inner cluster and one to part of the outer ring. This might lead to unrealistic parameters. Points in the same actual cluster might be incorrectly assigned to different Gaussian components due to the model assigning different parts of the outer ring into different sections. Lastly, the variance of the outer Gaussian might be very large to accommodate the width of the ring which could become an issue for parameter estimation.