final-attempt

2024-12-03

1

(a) The initialization of the cluster centroids does not affect the final result of k-means clustering.
False; k-means clustering is sensitive to initialization [Final Exam Review Slide - Properties of K-Means]

(b) K-means clustering can effectively handle clusters of any shape and/or size.
False; it assumes spherical cluster shapes [Final Exam Review Slide - K-Means]

(c) The K-means algorithm updates cluster centroids by recalculating the mean of all points assigned to each cluster.
True; [Final Exam Review Slide - Properties of K-Means]

(d) K-means clustering guarantees finding the global optimum of the clustering objective function.
False; K-means clustering sometime get stuck in local minimum [Final Exam Review Slide - Properties of K-Means]

(e) K-means is guaranteed to terminate.
True; [Final Exam Review Slide - Solution]

(f) The Elbow Method can always be used to determine the optimal number of clusters for K-means.
False; It might not work in all the case[Final Exam Review Slide - Properties of K-Means and 12/2 Review Session]

(g) Hierarchical clustering does not require the number of clusters to be specified in advance.
True [Final Exam Review Slide - Hierarchical clustering]

(h) Single linkage in hierarchical clustering measures the distance be- tween the closest points of two clusters.
True [Final Exam Review Slide - Group Similarity]

(i) The cutting of the dendrogram at a specific level can help deter- mine the number of clusters.
True [Final Exam Review Slide - Hierarchical Model Example]

(j) A dendrogram is a tree-like diagram that shows the hierarchical relationships between clusters.
True [Final Exam Review Slide - Hierarchical Model Example]

(k) The Expectation-Maximization (EM) algorithm is commonly used to estimate the parameters of mixture models.

True [Final Exam Review Slide - EM Algorithm]

(l) Mixture models provide probabilistic assignments of data points to clusters, unlike K-means clustering, which provides hard as- signments.
True [Final Exam Review Slide - Mixture models can be view as probabilistic modeling]

(m) Poisson Mixture Models assume that the data is generated from a mixture of several Poisson distributions.
True

(n) Mixture models are sensitive to the initialization of parameters.
True [Final Exam Review Slide - Mixing VS K-Means]

(o) The Expectation-Maximization (EM) algorithm for mixture mod- els alternates between an E-step and an M-step.
True [Final Exam Review Slide - EM Algorithm]

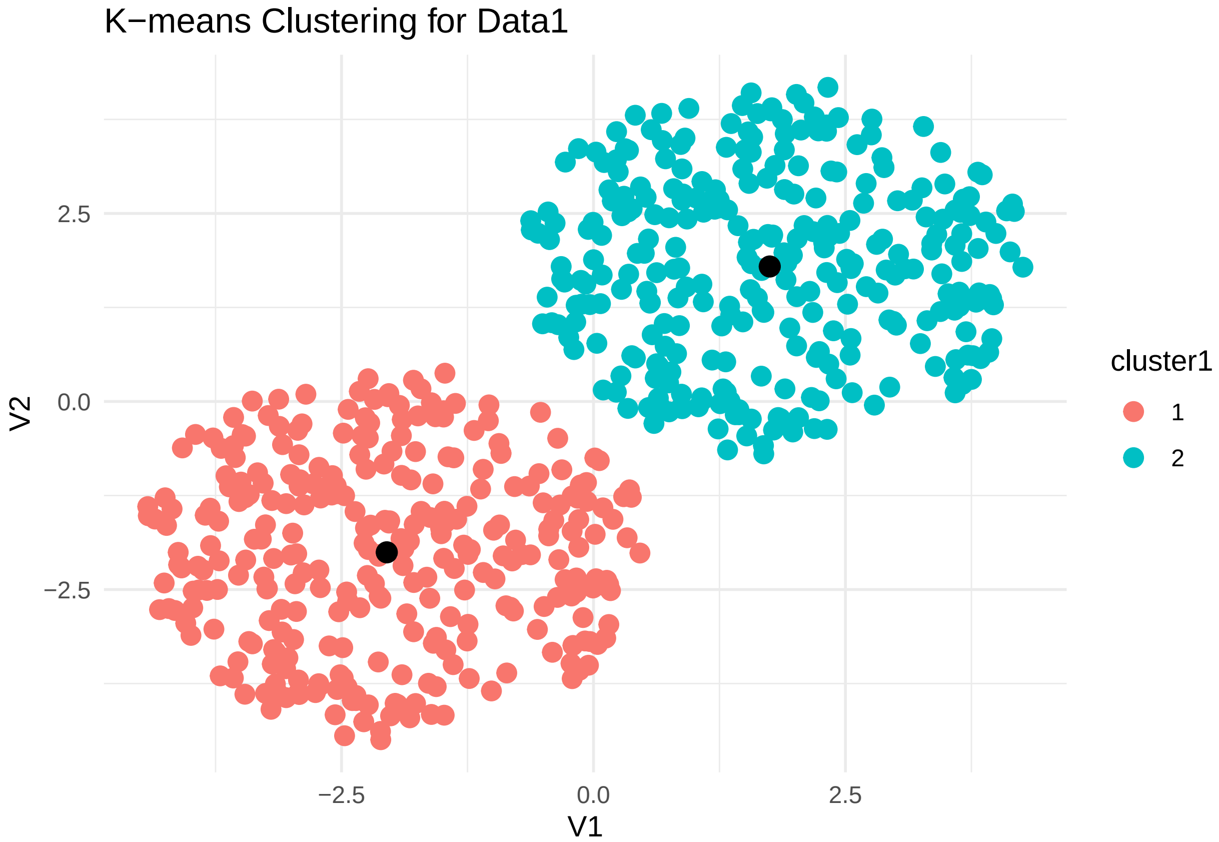
(p) Mixture models require that all components have the same weight.
False; no requiments as long as they add up to 1 [Final Exam Review Slide - Mixture Model]

(q) In a Gaussian Mixture Model, the probability of each data point belonging to a component is fixed over all iterations of the EM algorithm (and never updated).
False; the probability of each point is updated throughout EM algorithm [Final Exam Review Slide - EM Algorithm Code]

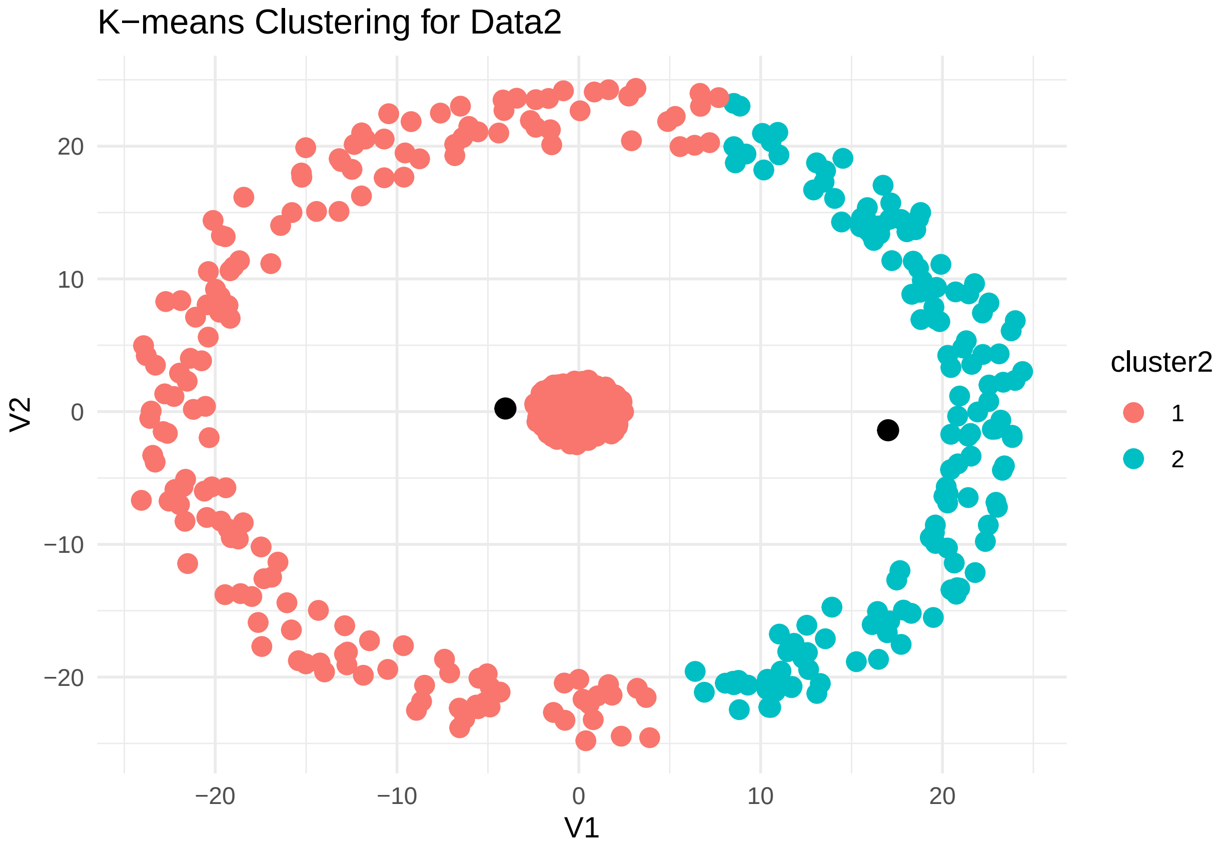
2
a)i.

```
library(ggplot2)
set.seed(1234)
data1 <-read.csv('~/data-clean/final-attempt/data1.csv', header=FALSE)
data2 <-read.csv('~/data-clean/final-attempt/data2.csv', header=FALSE)
```

```
#use kmeans function
kresult1 <- kmeans(data1,  centers = 2,
                   nstart = 20, algorithm ='Lloyd', iter.max=100)
kresult2 <- kmeans(data2,  centers = 2,
                   nstart = 20, algorithm ='Lloyd', iter.max=100)

# combine dataframe for plotting
kdata1 <- cbind(data1, cluster1 = as.factor(kresult1$cluster))
kdata2 <- cbind(data2, cluster2 = as.factor(kresult2$cluster))

# plot
ggplot(kdata1, aes(x = V1, y = V2, color = cluster1)) +
  geom_point(size = 3) +
  geom_point(data = as.data.frame(kresult1$centers),
             aes(x = V1, y = V2),
             color = "black",
             size = 5,
             shape = 20) +
  labs(title = "K-means Clustering for Data1",
       x = "V1",
       y = "V2") +
  theme_minimal()
```



```
ggplot(kdata2, aes(x = V1, y = V2, color = cluster2)) +
  geom_point(size = 3) +
  geom_point(data = as.data.frame(kresult2$centers),
             aes(x = V1, y = V2),
             color = "black",
             size = 5,
             shape = 20) +
  labs(title = "K-means Clustering for Data2",
       x = "V1",
       y = "V2") +
  theme_minimal()
```



ii). The Kmeans did significantly better for data1 compared to its performance for data2. For data1, the two clusters have clear shape that aligns with the pattern of the true clustering and the centers labeled by the algorithm resembles the true centers of the true clusters. However, for data2, it looks like the algorithm simply split the data points into left half and right half, which is not reflective of the data clustering pattern. Moreover, the centers of the two clusters do not assemble the true center of the data.
iii).It went wrong because kmeans assumes the spherical clustering. Since data1 has spherical clustering pattern and data2 has ring clustering pattern, kmeans performed significantly better for data1 but worse for data2.
b)
i).[Final Exam Review - Two Component mixture]

```
library(mixtools)
```

```
## mixtools package, version 2.0.0, Released 2022-12-04
## This package is based upon work supported by the National Science Foundation under Grant No. SES-051
```

```
mdata1 <- unlist(data1)
mdata2 <- unlist(data2)
model1 <- normalmixEM(mdata1, k = 2)
```

```
## number of iterations= 345
```

```
summary(model1)
```

```
## summary of normalmixEM object:
##          comp 1    comp 2
## lambda 0.660628  0.339372
## mu     1.127161 -2.571966
## sigma  1.622304  0.989746
## loglik at estimate:  -2618.101
```

```
model2 <- normalmixEM(mdata2, k = 2)
```

```
## number of iterations= 14
```

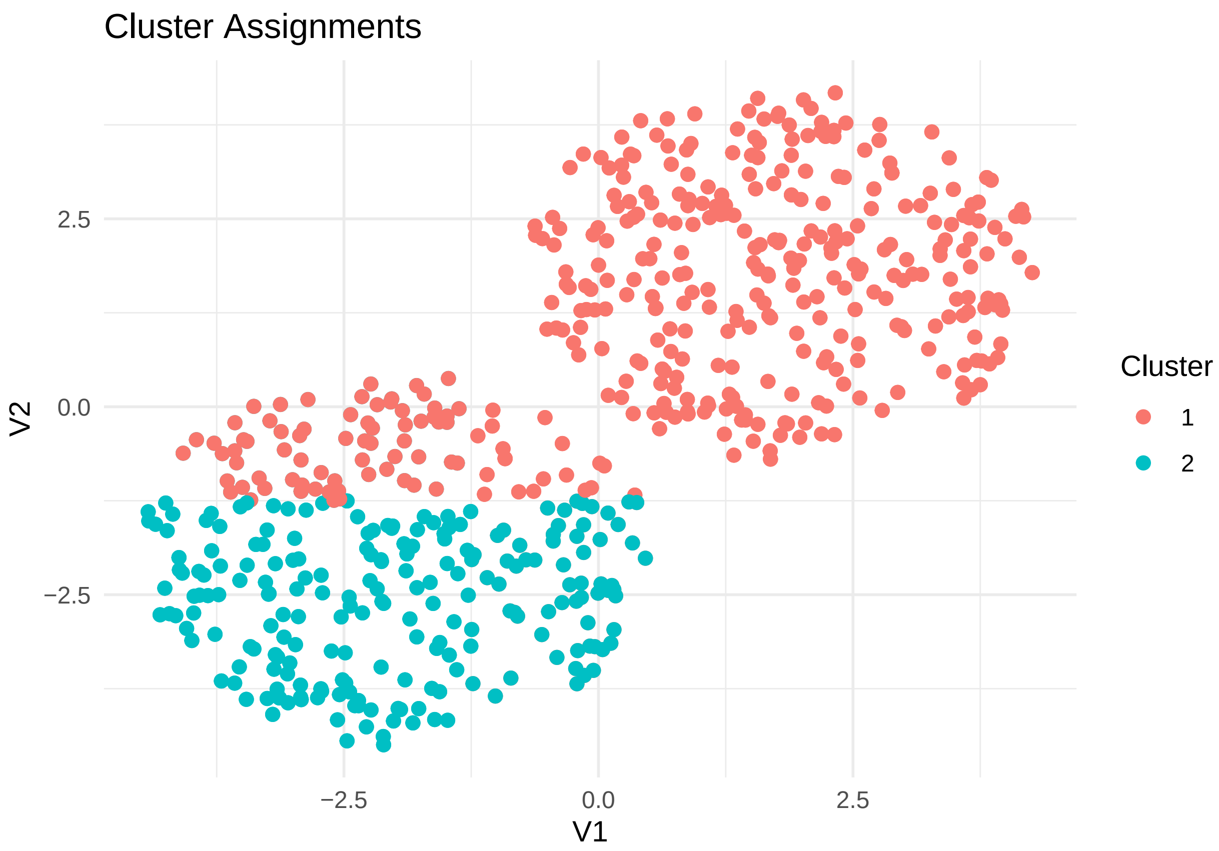
```
summary(model2)
```

```
## summary of normalmixEM object:
##             comp 1    comp 2
## lambda  0.4559817  0.544018
## mu     -0.0366967  0.796870
## sigma   1.2239174 15.337517
## loglik at estimate:  -4204.165
```

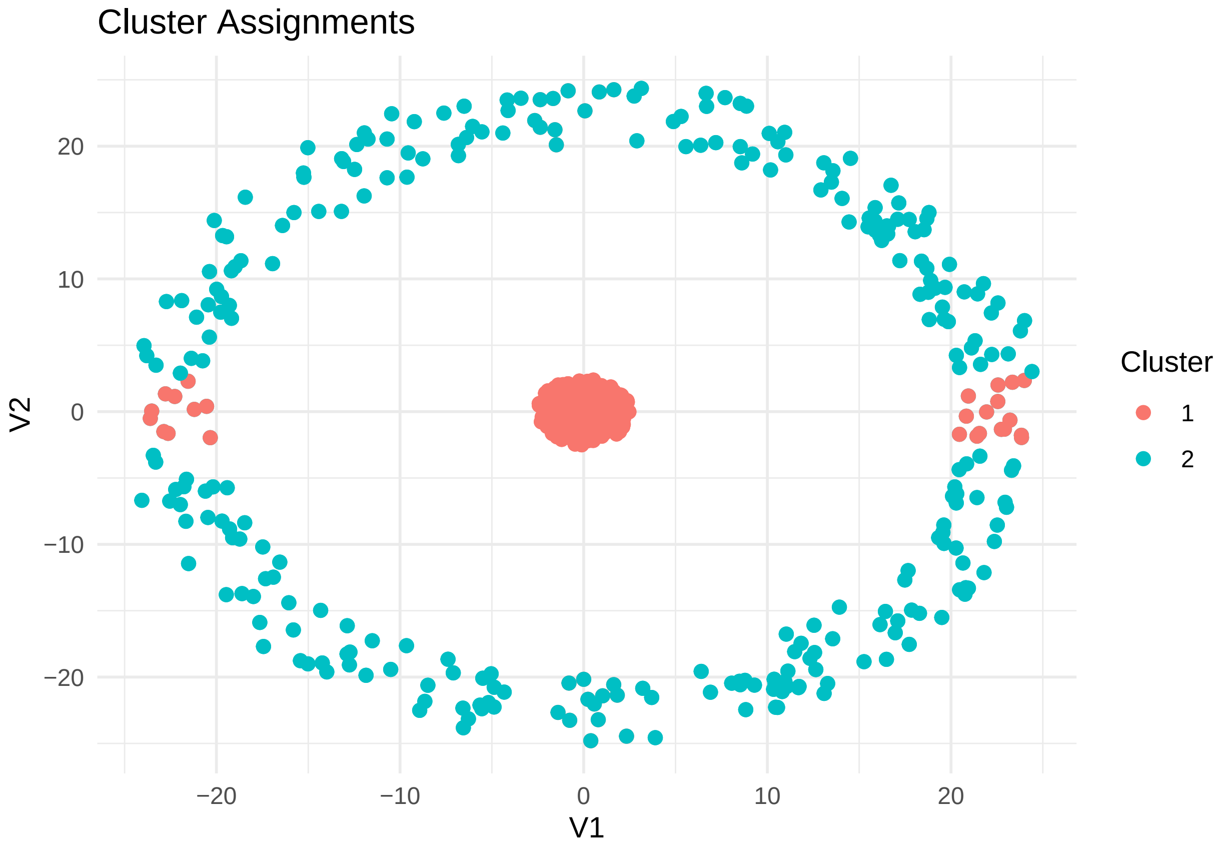
Model 1 estimates 66.0628% of the data is associated with cluster 1 and the est. mean of the cluster 1 is 1.127161 and the est. standard deviation of cluster 1 is 1.622304. For cluster 2, 33.9372% of data is associated with cluster 2 and the est. mean of the cluster 2 is -2.571966 and the est. standard deviation of cluster 2 is 0.989746.
Model 2 estimates 45.59817 % of the data is associated with cluster 1 and the est. mean of the cluster 1 is -0.0366967 and the est. standard deviation of cluster 1 is 1.2239174. For cluster 2, 54.4018% of of data is associated with cluster 2 and the est. mean of the cluster 2 is 0.796870 and the est. standard deviation of cluster 2 is 15.337517.

iii)

```
prepdata1 <- data.frame(data1, cluster1 = factor(apply(model1$posterior, 1, which.max)))
prepdata2 <- data.frame(data2, cluster2 = factor(apply(model2$posterior, 1, which.max)))

ggplot(prepdata1, aes(x = V1, y = V2, color = cluster1)) +
  geom_point(size = 2) +
  theme_minimal() +
  labs(title = "Cluster Assignments", x = "V1", y = "V2", color = "Cluster")
```



```r
ggplot(prepdata2, aes(x = V1, y = V2, color = cluster2)) +
  geom_point(size = 2) +
  theme_minimal() +
  labs(title = "Cluster Assignments", x = "V1", y = "V2", color = "Cluster")
```



c) Second dataset has a pattern of data clustering around a ring and data clustering at the center of the ring. However, the EM algorithm still assigns the center of the ring as well part of the ring as cluster 1 and the rest as cluster 2, which suggests something is incorrect.

3.

```
library(circular)
```

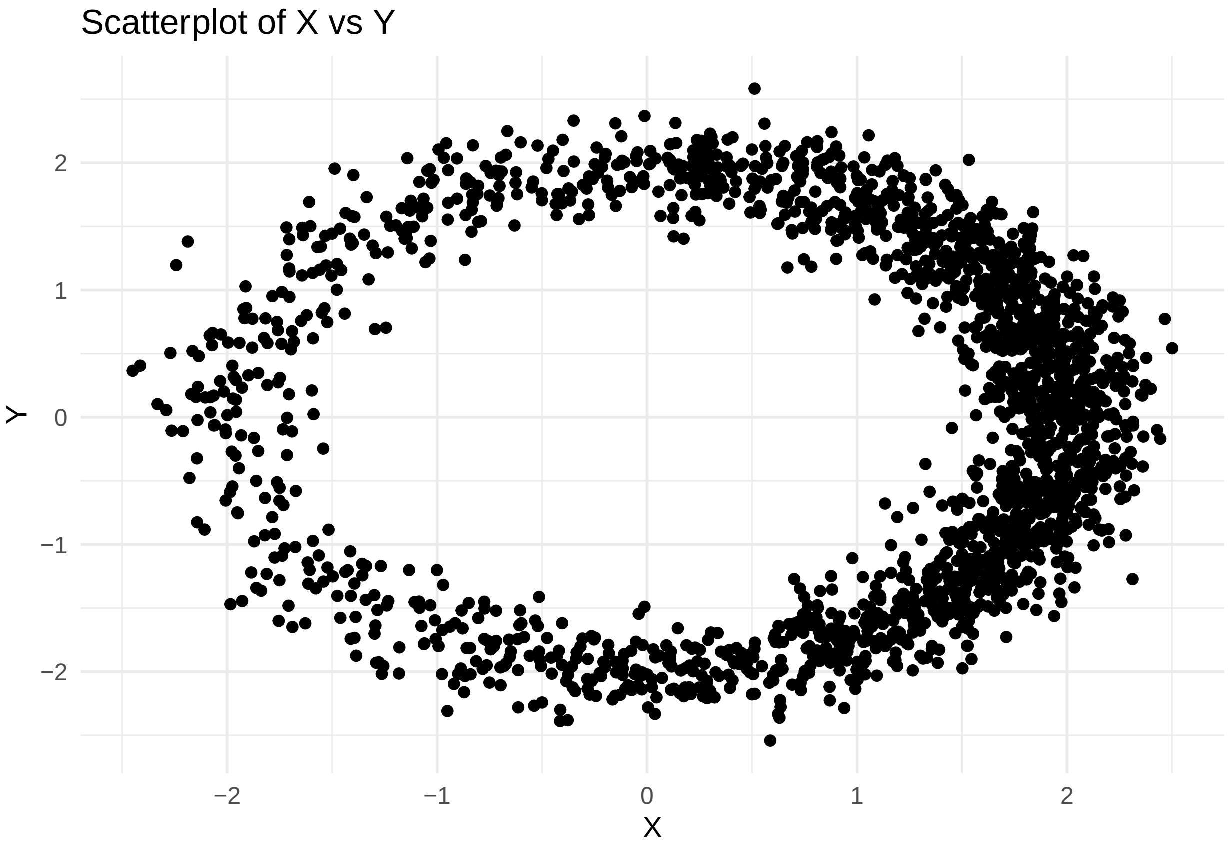
```
##
## Attaching package: 'circular'

## The following objects are masked from 'package:stats':
##
##     sd, var
```

```
# Function to generate von Mises data
vdist <- function(n, mu_r, sigma_r, mu_theta, kappa) {
  # Radial Distance
  r <- abs(rnorm(n, mean = mu_r, sd = sigma_r))

  # Angular Component
  if (kappa > 0) {
    theta <- rvonmises(n, mu = mu_theta, kappa = kappa)
  } else {
    theta <- runif(n, 0, 2 * pi)
  }

  # Conversion to Cartesian Coordinates
  x <- r * cos(theta)
  y <- r * sin(theta)


  return(data.frame(x = x, y = y))
}
```

```
n <- 2000
mu_r <- 2
sigma_r <- 0.2
mu_theta <- 0
kappa <- 1
vdata <- vdist(n, mu_r, sigma_r, mu_theta, kappa)
```

```
## Warning in as.circular(x): an object is coerced to the class 'circular' using default value for the
##   type: 'angles'
##   units: 'radians'
##   template: 'none'
##   modulo: 'asis'
##   zero: 0
##   rotation: 'counter'
## conversion.circularmuradians0counter
```

```
ggplot(vdata, aes(x = x, y = y)) +
  geom_point() +
  labs(title = "Scatterplot of X vs Y",
       x = "X",
       y = "Y") +
  theme_minimal()
```



The sampled data closely matches the outer ring shape of Data2. Data2 appears to follow a von Mises two-component mixture model, with one component at a larger radius and another at a smaller radius near the center. This likely explains why the Gaussian Mixture Model (GMM) with EM failed, as GMM assumes elliptical clusters, which do not align with the circular structure of the data.