Clustering Example 8: k-means with a Poor Start

Amy Wagaman, Nathan Carter

July 2020

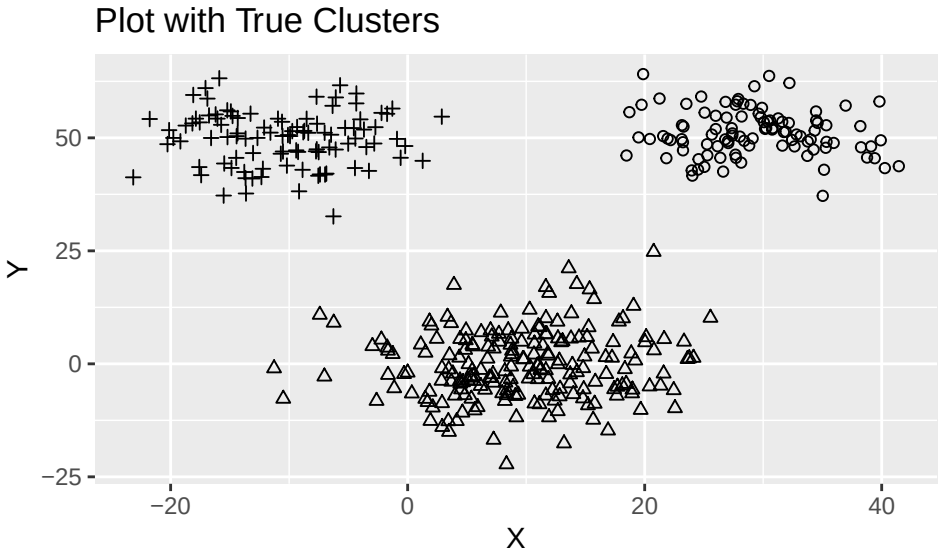
Load required library.

```
library(ggformula)
```

Generate random data in three clusters.

We use three multivariate normal distributions and combine the results.

```
set.seed(300)
X <- MASS::mvrnorm(100, mu = c(30,50), Sigma = matrix(c(30,0,0,30), nrow = 2))
X2 <- MASS::mvrnorm(200, mu = c(10,0), Sigma = matrix(c(50,0,0,50), nrow = 2))
X3 <- MASS::mvrnorm(100, mu = c(-10,50), Sigma = matrix(c(30,0,0,30), nrow = 2))
labels <- c(rep(1,100), rep(2,200), rep(3,100))
Y <- as.data.frame(rbind(X, X2, X3)) # makes the generated data into a data frame
gf_point(V2 ~ V1, data = Y, shape = labels) %>%
  gf_labs(title = "Plot with True Clusters", x = "X", y = "Y")
```



Perform k-means clustering.

But here we intentionally begin with three "unlucky" initial cluster centers, $(10, 5)$, $(20, 0)$, and $(10, 50)$. (However there's no luck about it; we've chosen them to illustrate how the k-means clustering algorithm can suffer from random intial cluster centers that happen to be laid out in an unforunate way relative to the data. Although these weren't chosen at random, points near them might have been.)

```
Ksol2 <- kmeans(Y, centers = matrix(c(10,5,20,0,10,50), nrow = 3, byrow = T))
gf_point(V2 ~ V1, data = Y, shape = Ksol2$cluster) %>%
  gf_labs(title = "Plot with Found Clusters", x = "X", y = "Y")
```

Note that the top two clusters have been united into one and the bottom cluster has been split into two. So even with the correct value of k (in this case 3) the result is not guaranteed to be what you may want.