Math 656 Assignment 9

Jeff Gould

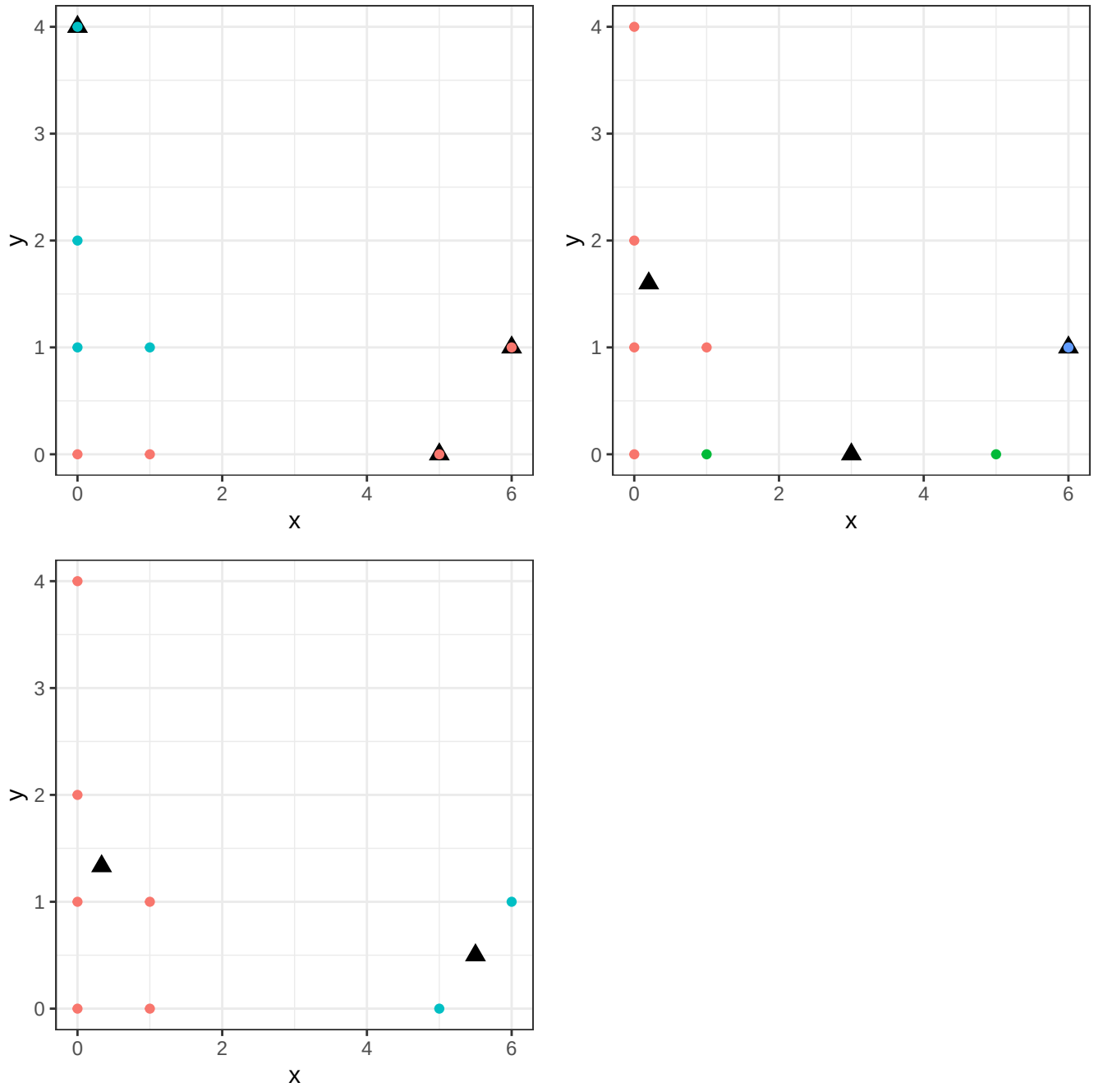
10/26/2020

```
Data <- data.frame(
  ID = c(1:8),
  x = c(0,0,0,0,1,1,5,6),
  y = c(4,2,1,0,1,0,0,1)
)


x <- Data[,2:3]

distance <- function(a,b){
  dist <- sum((a-b)^2)
  return(sqrt(dist))
}




MyKmeans <- function(x, K, start_mu, plots = F){

  assign_mu <- function(point){
    dists <- apply(mu, MARGIN = 1, FUN = distance, b = point)
    new_assign <- which.min(dists)
    return(new_assign)
  }

  new_mu <- function(j){
    x_j <- x[new_assignments == j,]
    new_mu <- colMeans(x_j)
    return(new_mu)
  }

  mu <- start_mu

  init_assignments <- sample(1:K, size = nrow(x), replace = T)

  if(plots == T){
      plot_data <- data.frame(x,
                              assn = as.factor(init_assignments))
      mu_plot <- data.frame(mu)
      print(
        ggplot(data = plot_data, aes(x = x, y = y)) +
          geom_point(data = mu_plot, color = "black",
```

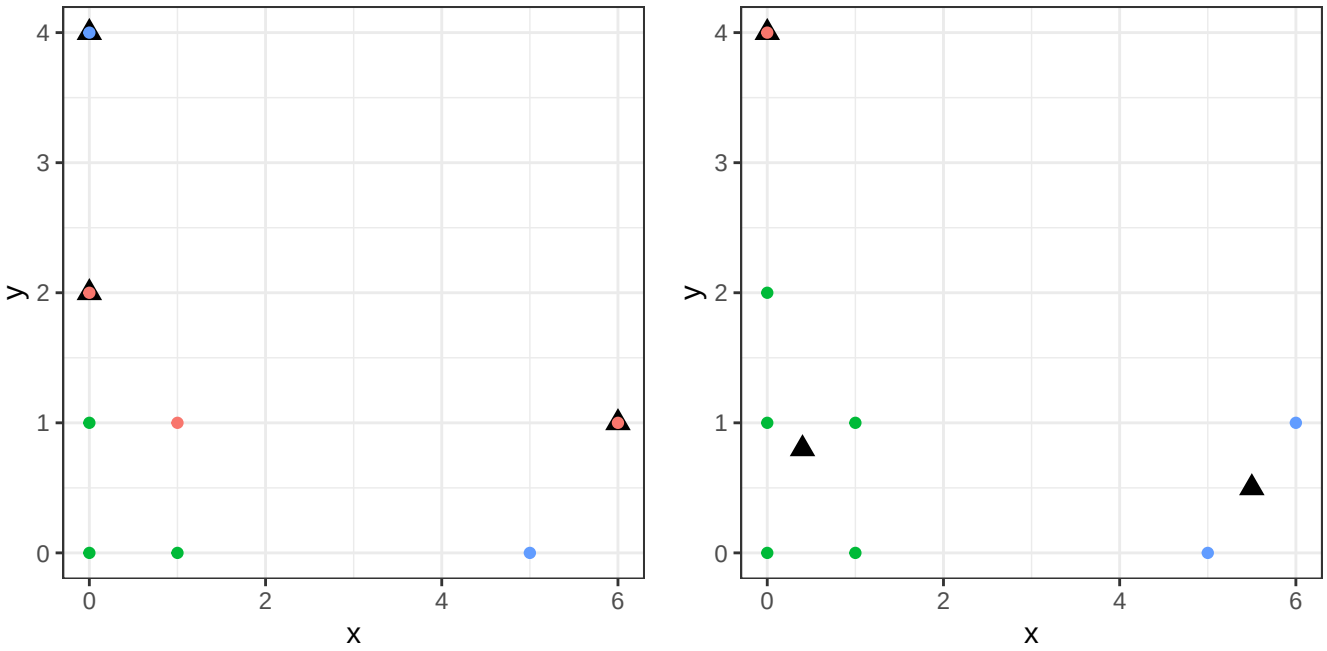
```
                    size = 3, shape = 17) +
          geom_point(aes(color = assn), show.legend = F) +
          theme_bw()
      )
    }

  new_assignments <- apply(x, 1, assign_mu)
  mu <- t(sapply(c(1:K), new_mu))
  iteration <- 1


  while (identical(init_assignments, new_assignments) == F) {

    if(plots == T){
      plot_data <- data.frame(x,
                              assn = as.factor(new_assignments))
      mu_plot <- data.frame(mu)
      print(
        ggplot(data = plot_data, aes(x = x, y = y)) +
          geom_point(data = mu_plot, color = "black",
                     size = 3, shape = 17) +
          geom_point(aes(color = assn), show.legend = F) +
          theme_bw()
      )
    }

    init_assignments <- new_assignments

    new_assignments <- apply(x, 1, assign_mu)
    mu <- t(sapply(c(1:K), new_mu))

    iteration <- iteration + 1

  }

  final_assignment <- new_assignments

  output <- list(iterations = iteration,
                 assignment = final_assignment,
                 mu = mu)
  return(output)


}
```

a)

```
set.seed(123)
start_mu <- Data[c(1,7,8), c(2,3)]

KmeansA <- MyKmeans(x = x, K = 3, start_mu = start_mu, plots = T)
```




```
start_mu <- Data[c(1,2,8), c(2,3)]

KmeansB <- MyKmeans(x = x, K = 3, start_mu = start_mu, plots = T)
```

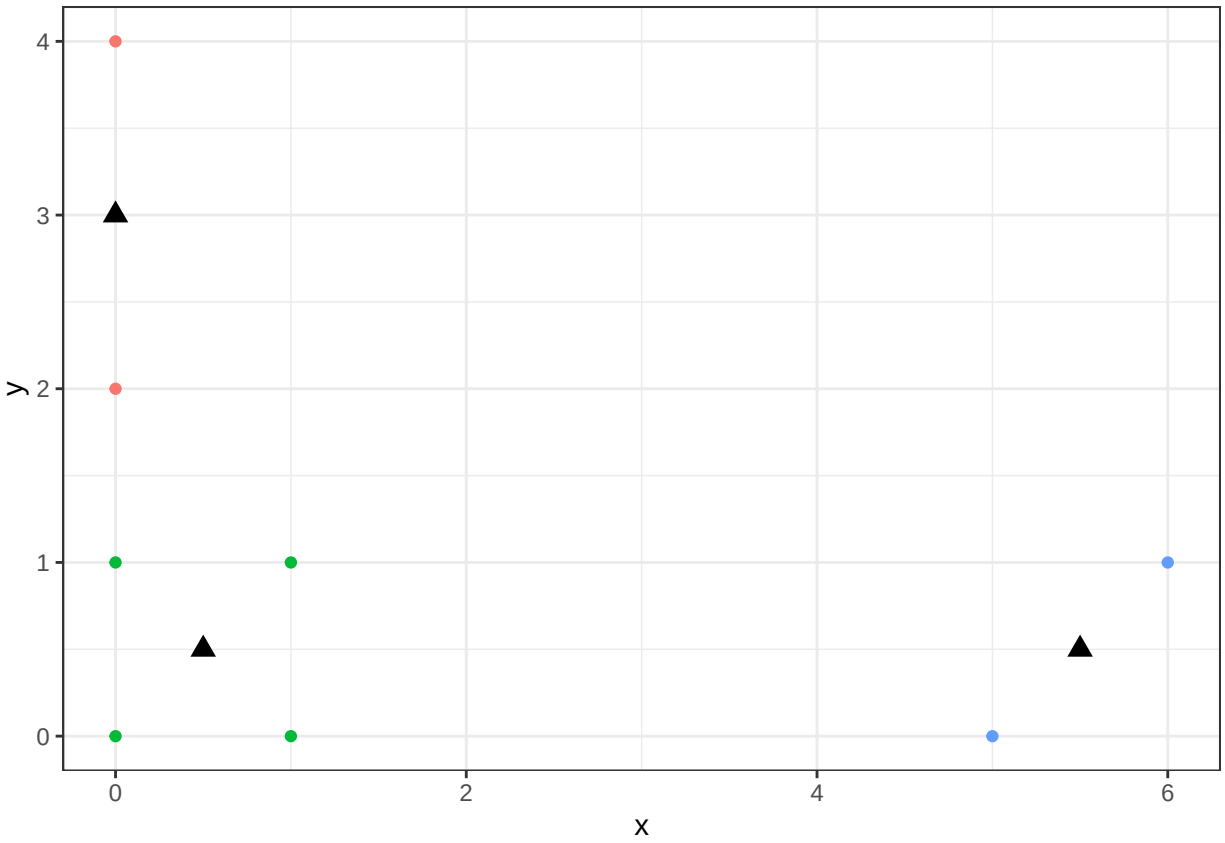



```
cl1 <- RWeka::SimpleKMeans(Data[,2:3], Weka_control(N = 3))
cl1
```

```
##
## kMeans
## ======
##
## Number of iterations: 3
## Within cluster sum of squared errors: 0.2604166666666667
##
## Initial starting points (random):
##
## Cluster 0: 1,0
## Cluster 1: 0,2
## Cluster 2: 0,0
##
## Missing values globally replaced with mean/mode
##
## Final cluster centroids:
##                            Cluster#
## Attribute    Full Data          0          1          2
##                  (8.0)      (2.0)      (2.0)      (4.0)
## ============================================================
## x                1.625        5.5          0        0.5
## y                1.125        0.5          3        0.5
```

```
cl1$class_ids
```

```
## 1 2 3 4 5 6 7 8
## 1 1 2 2 2 2 0 0
```



d)

The key observation from this exercise is that the starting μ is a key factor in determining the final μ and assignments. As we saw in exercises a and b, with different starting μ's but the same function, we ended up with different results. In fact, in a we actually ended up with only two centers, as one of them ends up getting dropped becuase it ends up not being the closest centroid for any of the data points. Meanwhile, `RWeka` ends up getting a different result than either of our starting points.

We are also doing this exercise on a very small dataset. This potentially allows for much greater variation in output with minor tweaks to inputs. We might expect more consistency with output should a larger data sample be available