Question 3

Yufei Yin

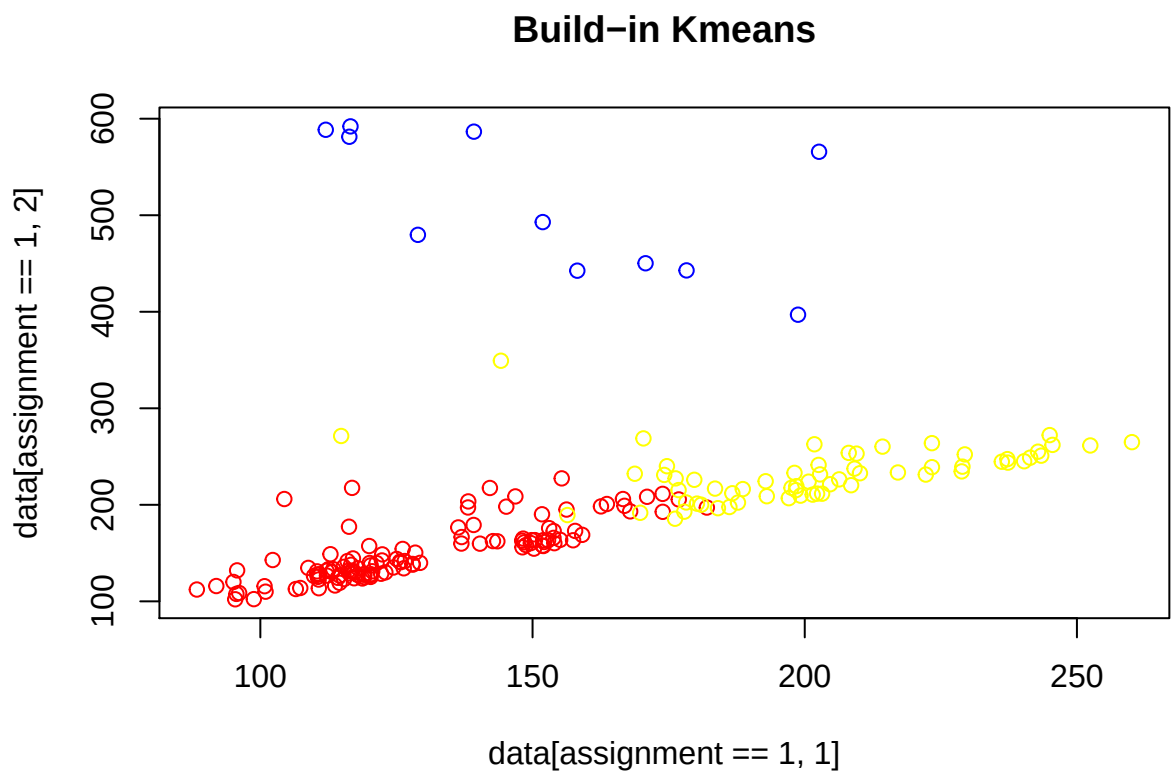
Question 3

- Modify your code to match the usual kmeans algorithm (and rename the function to mykmeans).
- Run your mykmeans algorithm with K = 3 on the parkinsons data.
- Run R's built in kmeans function on the parkinsons data, with K = 3.
- Explore the differences between the two implementations with some figures or examinations, and explain why these differences may arise.

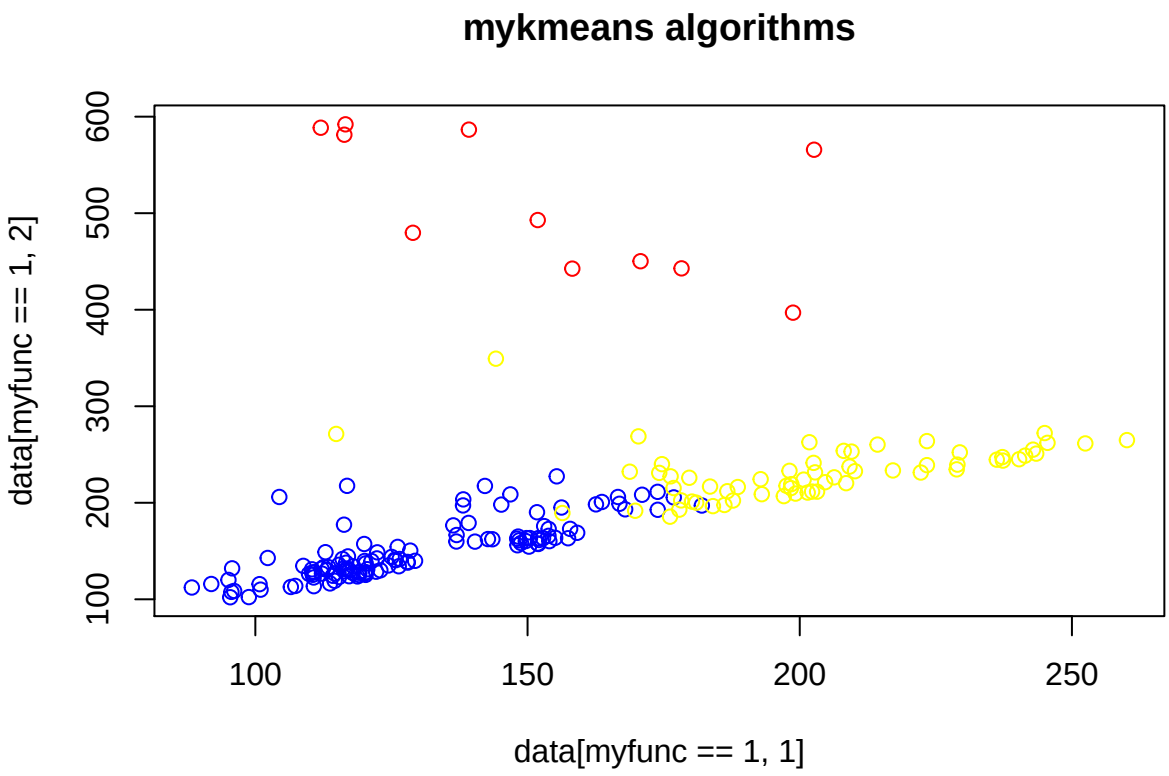
```
mykmeans = function(x, k, iters){
  N = dim(x)[1]
  D = dim(x)[2]

  centres = matrix(NA, k, D)
  clusters = rep(NA, N) # each entry between 1 and K

  for (i in 1:N){
    clusters[i] = sample.int(k, 1)
  }

  for (iter in 1:iters){
    for (k in 1:k){
      for (d in 1:D){
        centres[k, d] = mean(x[clusters == k, d])
      }
    }
    distanceMatrix <- matrix(NA, nrow=N, ncol=k)
    for(i in 1:k) {
      distanceMatrix[,i] <- sqrt(rowSums(t(t(x)-centres[i,])^2))
    }
    clusters <- apply(distanceMatrix, 1, which.min)
    centres <- apply(x, 2, tapply, clusters, mean)
  }
  return(list(locations=centres, assignment=clusters))
}
```

```
data = read.table(file = 'parkinsons.data', sep = ',', header = TRUE)
data = data[,-1]
result = kmeans(data, 3)
assignment = result$cluster

res = mykmeans(data, 3, 1000)
myfunc = res$assignment
```

```r
plot(data[assignment == 1, 1], data[assignment == 1, 2], col = "blue",
     xlim = range(data[,1]), ylim = range(data[,2]),
     main = "Build-in Kmeans")
points(data[assignment == 2, 1], data[assignment == 2, 2], col = "red")
points(data[assignment == 3, 1], data[assignment == 3, 2], col = "yellow")
```




```r
plot(data[myfunc == 1, 1], data[myfunc == 1, 2], col = "blue",
     xlim = range(data[,1]), ylim = range(data[,2]),
     main = "mykmeans algorithms")
points(data[myfunc == 2, 1], data[myfunc == 2, 2], col = "red")
points(data[myfunc == 3, 1], data[myfunc == 3, 2], col = "yellow")
```



The two figures used different colors for clusters. The difference is caused by the random generator.