

Week 10 Mini Project Clustering

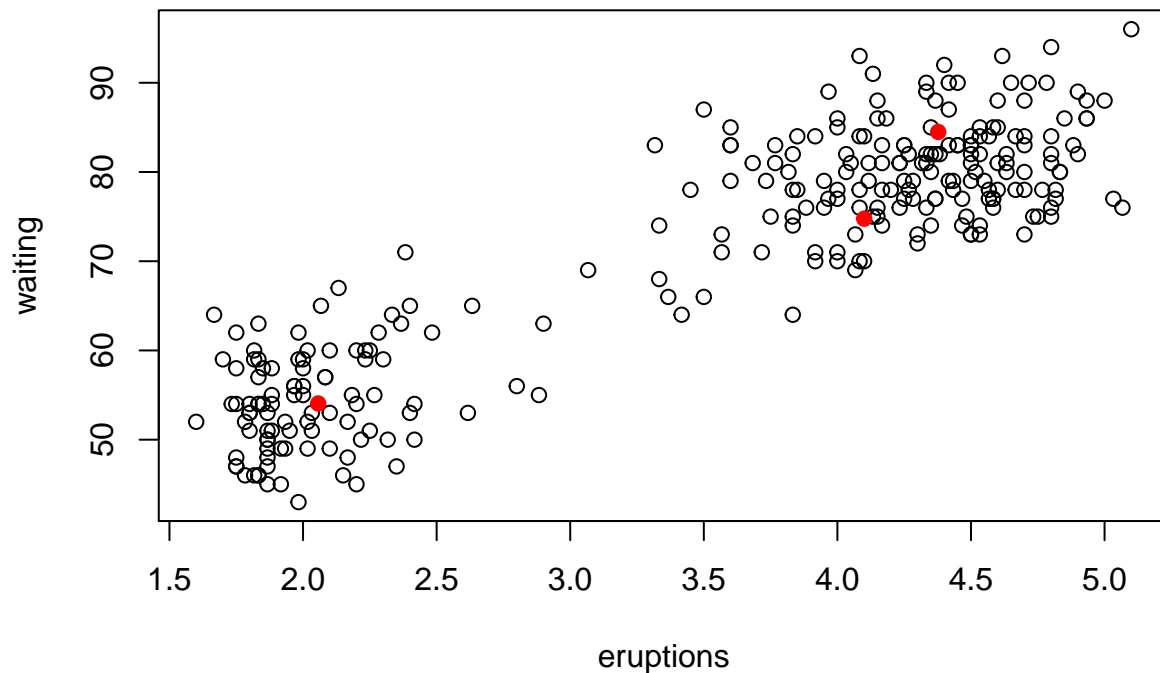
Ben Arancibia

10/31/2015

This project will focus on taking data from the R package `cluster.datasets` and performing a cluster analysis on the data. The first thing to do is setup the appropriate environment using the following.

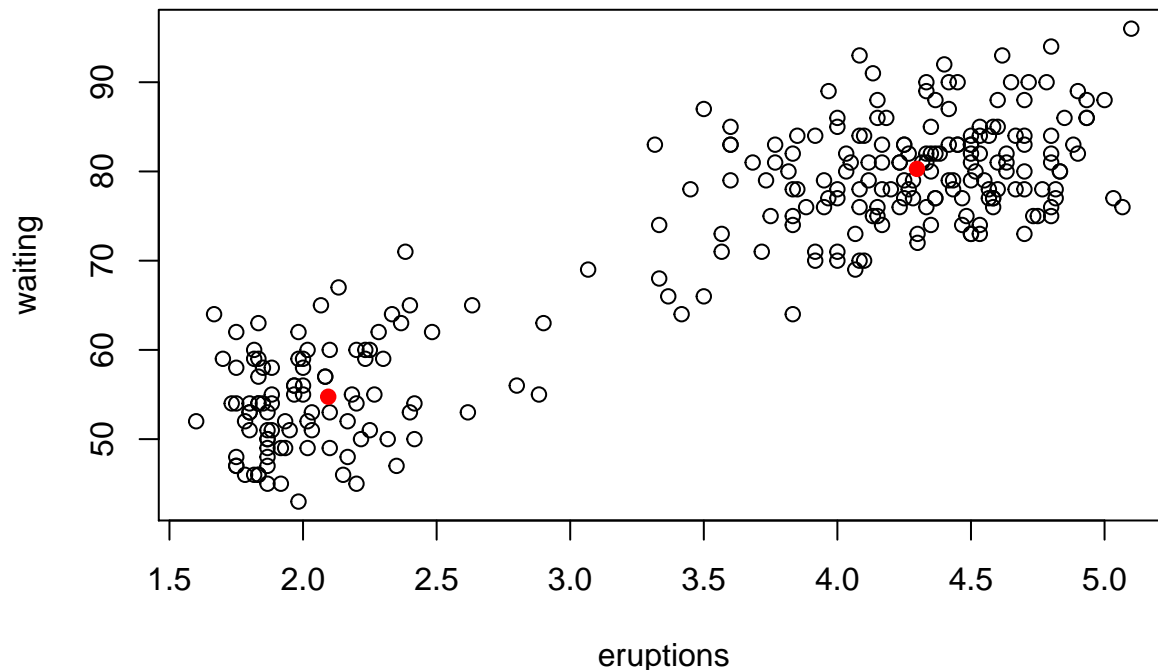
I am going to cluster waiting time between eruptions and the duration of eruptions for the Old Faithful geyser in Yellowstone. This dataset `faithful` is part of base R.

```
data <- faithful  
  
kmeans.data<- kmeans(data, 3)  
  
plot(data)  
points(kmeans.data$centers, pch=19, col="red")
```



Looking at the initial datasets there are two clusters.

```
kmeans.data<- kmeans(data, 2)  
  
plot(data)  
points(kmeans.data$centers, pch=19, col="red")
```



Next do this in Spark. I did this using Python in terminal on the dataset. I used the same commands from the Spark documentation and it worked for me. Below are the following commands.

```
from pyspark.mllib.clustering import KMeans, KMeansModel
from numpy import array
from math import sqrt

# Load and parse the data
data = sc.textFile("data/mllib/faithful.csv")
parsedData = data.map(lambda line: array([float(x) for x in line.split(' ')]))

# Build the model (cluster the data)
clusters = KMeans.train(parsedData, 2, maxIterations=10,
                        runs=10, initializationMode="random")

# Evaluate clustering by computing Within Set Sum of Squared Errors
def error(point):
    center = clusters.centers[clusters.predict(point)]
    return sqrt(sum([x**2 for x in (point - center)]))

WSSSE = parsedData.map(lambda point: error(point)).reduce(lambda x, y: x + y)
print("Within Set Sum of Squared Error = " + str(WSSSE))

# Save and load model
clusters.save(sc, "myModelPath")
sameModel = KMeansModel.load(sc, "myModelPath")
```

The clustering is the same as using Kmeans in R. See the plot below which is similar to the plot above.

