§*R* for Clustering: *K*-Means Clustering

Dolores Romero Morales

The aim of this workshop is to work on building *K*-Means clusterings using a cars dataset that is familiar to you (mtcars). This dataset can be found in R. Please use the command 'data()' to load the dataset, and get familiar with it before starting to answer the questions below.

Note: Luckily, 'kmeans()' and related commands to do K-means clustering are included in the 'stats' package, which is available in the basic download of R.

Step 1. Load the 'mtcars' dataset and get the dimension of the dataset.
Answer:

```
data(mtcars)
dim(mtcars)
```

Step 2. Normalize the data and get summary statistics.
Answer:

```
minimum <- apply(mtcars,2,min)
maximum <- apply(mtcars,2,max)
mtcarsNORM <- scale(mtcars,center=minimum,scale=(maximum-minimum))
summary(mtcarsNORM)
```

Note:
You have seen the use of the scale function already twice in this course. We have used it previously to standardize the data and here to normalize the data.

Step 3. Using the normalized dataset, derive a *K*-means clustering, with *K*=3.
Answer:

```
set.seed(1)
myKmeans3 <- kmeans(mtcarsNORM, 3)
```

Note1:
Recall that in *K*-means clustering as in other Data Science tools there is a random component, and therefore we fix the seed used to generate random numbers to make sure we can reproduce the tool again.
Note2:
If you would like to plot the clustering you can use

```
with(as.data.frame(mtcarsNORM), pairs(as.data.frame(mtcarsNORM),
col=c(1:11)[myKmeans3$cluster]))
with(as.data.frame(mtcarsNORM), pairs(as.data.frame(mtcarsNORM)[,1:2],
col=c(1:11)[myKmeans3$cluster]))
```

where we had to use the command 'as.data.frame(mtcarsNORM)' to treat 'mtcarsNORM', which is currently a matrix, as a data frame.

Step 4. Obtain the size of the clusters, the sum of squares vector, and the total sum of squares.
Answer:

```
myKmeans3$size
myKmeans3$withinss
myKmeans3$tot.withinss
```

Note:

Observe that 'myKmeans3$tot.withinss' is the sum of the 'myKmeans3$withinss'. Recall that 'myKmeans3$tot.withinss' is the function that we are optimizing. Therefore, if we have two runs of kmeans() with the same K, then we will choose for the clustering with the smallest 'myKmeans3$tot.withinss'.

Step 5. Choose the seed for the random generator equal to 2, and repeat Step 3. Compare both clusterings in terms of sum of squares.

Answer:

```
set.seed(2)
myKmeans3 <- kmeans(mtcarsNORM, 3)
```

Note:

Although we did not see a change in the clustering from seed=1 to seed=2, we will see that there are values of the seed for which the clustering changes.

Step 6. Choose the seed for the random generator equal to 1345555, and repeat Step 3. Compare both clusterings in terms of sum of squares.

Answer:

```
set.seed(1345555)
myKmeans3 <- kmeans(mtcarsNORM, 3)
```

Note:

To *get close* to the best 3-means clustering, one needs to repeat Step 3 several times for different values of the seed and take the clustering with the best total sum of squares. We could rerun the 3-means 50 times, and take the best output. This is what the parameter nstart does.

Step 7. Repeat Step 3 with a multi-start approach with nstart=50. Compare both clusterings in terms of sum of squares.

Answer:

```
set.seed(1)
myKmeans3multistart <- kmeans(mtcarsNORM, 3, nstart=50)
```

Note:

To *get close* to the best 3-means clustering, we have repeated Step 3 several times for different values of the seed, and taken the best output as measured by the total sum of squares.

Step 8. Experiment with several values of *K*, and make an attempt to choose a good *K*.

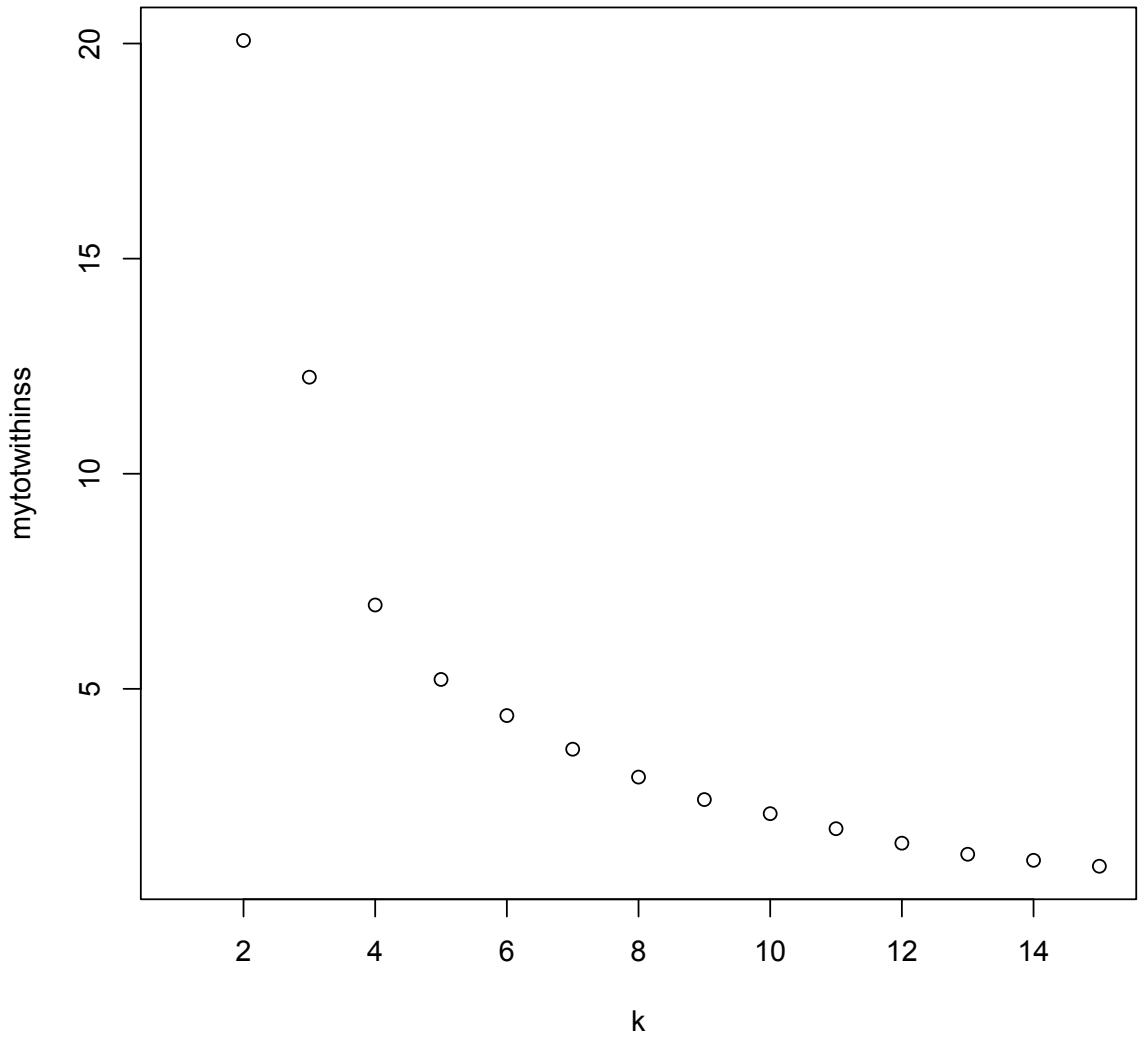
Answer:

```
mytotwithinss <- NULL
for(auxk in  2:15){
set.seed(1)
myKmeansauxkmultistart <- kmeans(mtcarsNORM, auxk, nstart=50)
mytotwithinss[auxk] = myKmeansauxkmultistart$tot.withinss
plot(mytotwithinss, xlab='k')
```

Note1:

Clearly, the total within sum of squares keeps decreasing when the number of clusters *K* keeps increasing. However, a high number of clusters may not be manageable for the user. Think about designing different pricing mechanisms for each of the clusters. A common thing

to do is to choose the K using the elbow rule. In our plot the elbow is around K=4, 5, 6. I would have personally chosen K=4.



Note2:
Once you make the decision on the number of clusters, for instance, K=4, you would need to build the clustering with K=4.

Note3:
It would be good if you plot this clustering, as we have done in Step 3.