CS CM121 Final Project

Ellen Wei

March 17, 2022

K-means clustering

```
source("project_function.R")
```

```
## Loading required package: colorspace
```

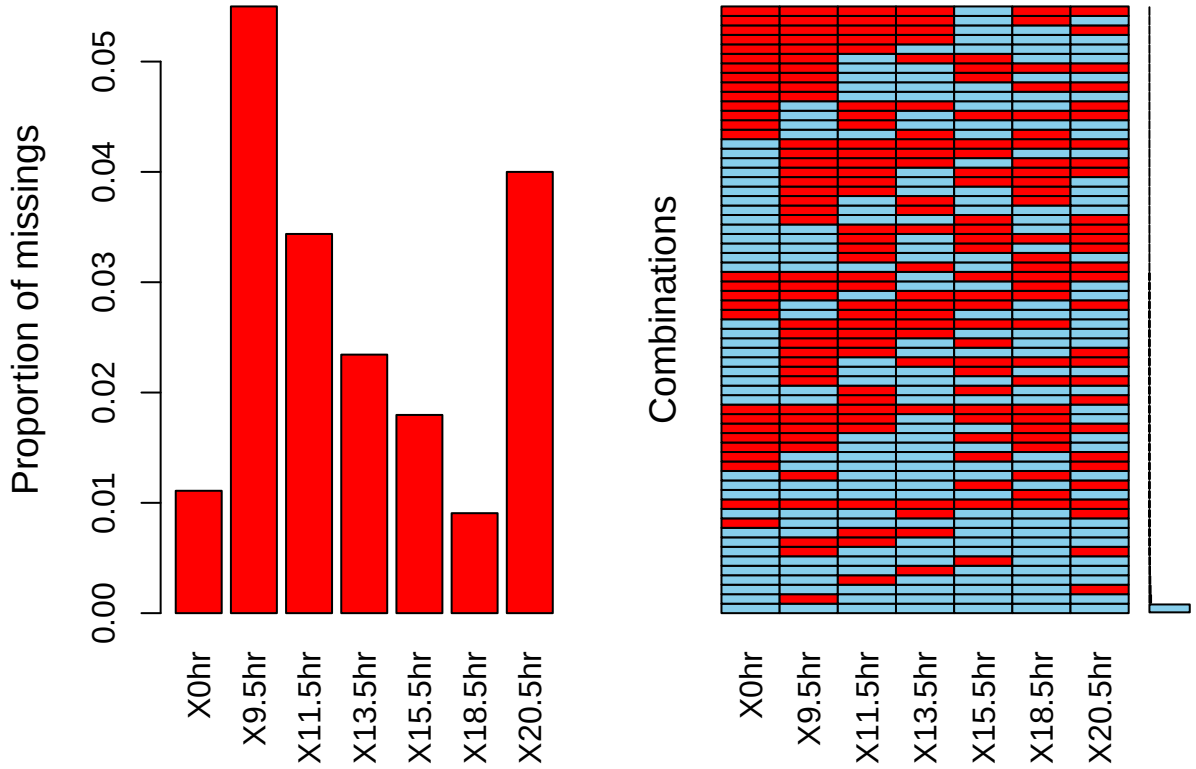
```
## Loading required package: grid
```

```
## VIM is ready to use.
```

```
## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues
```

```
##
## Attaching package: 'VIM'
```

```
## The following object is masked from 'package:datasets':
##
##     sleep
```




```
## Number of NA's in  X0hr = 71
## Number of NA's in  X9.5hr = 352
## Number of NA's in  X11.5hr = 220
## Number of NA's in  X13.5hr = 150
## Number of NA's in  X15.5hr = 115
## Number of NA's in  X18.5hr = 58
## Number of NA's in  X20.5hr = 256
```

Description of algorithm

The data has missing values. One method would have been to remove all the rows including NA values, but I chose to replace the NA's with the column mean so that I could still keep those observations.

First, I initialized k randomly sampled centroids. Then, using Euclidean distance, the distance is calculated between each observation and the centroids. The centroid with the minimum distance between the observation is assigned to the observation. After these two steps are initialized, I iterate between recalculating the centroid values taken as the mean of its assigned data points and reassigning the cluster to each observation. I monitor the change between distance between the centroid and its corresponding data points between iterations and once this change is less than my predetermined tolerance level of 10^{-8} or complete the predetermined number of iterations, I consider the algorithm to be complete and take the centroid values.

Testing with different values of k

Using my k-means clustering function, I tried it with k = 2,4,7.

```
k_means(data, k=2)
```

```
## Iterations:  24

##          X0hr      X9.5hr     X11.5hr    X13.5hr    X15.5hr    X18.5hr
## 1 -0.11262280 -0.06523613  0.14963904  0.1648802  0.1834103  0.9426264
## 2  0.03335309  0.05328242 -0.05564685 -0.1323035 -0.1913817 -0.3663951
##      X20.5hr cluster
## 1  0.8554267       1
## 2 -0.3433324       2
```

```
k_means(data, k=4)
```

```
## Iterations:  36

##          X0hr      X9.5hr     X11.5hr       X13.5hr     X15.5hr    X18.5hr
## 1 -0.06968871 -0.02602243  0.02725316 -0.0071734920 -0.05323494  0.3075413
## 2  2.56088889 -0.36278704 -0.07030291 -0.0001473778  0.04039136  0.2793415
## 3 -0.10497912 -0.02028513  0.23592830  0.3803533155  0.49419060  1.5486506
## 4  0.09620242  0.08286854 -0.06763645 -0.2219185151 -0.27920759 -0.8574460
##       X20.5hr cluster
## 1  0.26060265       1
## 2  0.03789974       2
## 3  1.54965395       3
## 4 -0.84711157       4
```

```
k_means(data, k=7)
```

```
## Iterations:  21

##           X0hr      X9.5hr     X11.5hr     X13.5hr     X15.5hr    X18.5hr
## 1 -0.888556428  4.93097438 -4.63882526 -2.71447672  3.57664248  1.7197512
## 2 -0.127905663 -0.09546805  0.13369320  0.09571026  0.07757575  0.7644258
## 3 -0.032085164 -1.49492361  5.16000000  0.00091840  0.48831527  0.2204147
## 4 -4.491085345  4.15911118 -1.40303978  4.30508328 -6.30181297 -3.1176104
## 5 -0.094461264  0.03557859  0.24173416  0.51971446  0.69250674  1.9762311
## 6  0.144238352  0.10011149 -0.05930518 -0.29446238 -0.37892107 -1.2268836
## 7  0.007241726  0.04510661 -0.05346449 -0.08043914 -0.12615740 -0.1124159
##       X20.5hr cluster
## 1 -4.96206170       1
## 2  0.62055623       2
## 3  1.40800000       3
## 4  5.39254180       4
## 5  2.08940977       5
## 6 -1.24647661       6
## 7 -0.06977956       7
```

For k = 2, the cluster means look quite different. The values are quite far from each other. For k = 4, the cluster means still show vast differences. The means span the entire range, and none of them are particularly close to each other. For k = 7, the cluster means show some similarities but still with a large range of values. Considering that each column is a point in time, we could plot these as a time graph and follow the trajectories. In this sense, the trajectories are still very varied and different from each other.

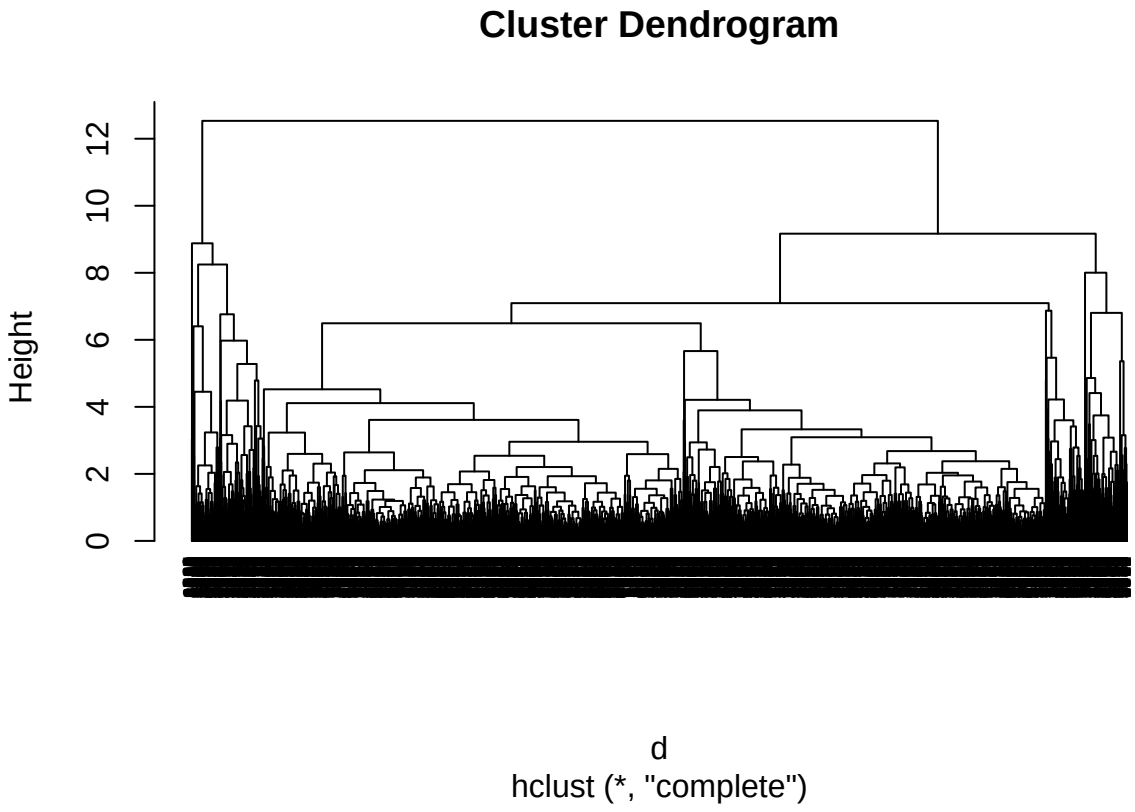
Different Clustering Method

A different clustering method is hierarchical clustering, implemented below.

```
# Dissimilarity matrix
d <- dist(data, method = "euclidean")

# Hierarchical clustering using Complete Linkage
hc1 <- hclust(d, method = "complete" )

# Plot the obtained dendrogram
plot(hc1, cex = 0.6, hang = -1)
```



Comparing the results

Hierarchical clustering can roughly show the relative distances between centres when using different values of k. It is useful for deciding how many k-centroid values to use. The k-means clustering method is useful when the k is known or given and can group observations together intuitively. These can be used in conjuction: first apply hierarchical clustering to decide how many k's should be used, then k-means clustering with that k value. Hierarchical is definitely faster in this implementation. However, one is not better than the other since we have slightly different usages for them.