Master 2 Machine Learning for Data Science

Projet Apprentissage non supervisé
Human Activity Recognition

Réalisé par:
Nadia RADOUANI – 21911973 FI
Amira KOUIDER - 21904040 FI

Enseignant: Mr ALLOU Samé

Année universitaire: 2020-2021

Table des matières

Table des figures

I. INTRODUCTION:

La reconnaissance de l'activité humaine est un domaine d'étude très utilisé dans la santé. C'est un problème de classification des séquences de données d'accéléromètre enregistrées par des smartphones en mouvements bien définis. Elle a été l'objet des méthodes d'apprentissage supervisé au cours des dernières décennies mais suite à l'augmentation de nombre d'activités à reconnaitre, de nouvelles approches ont été proposées : des méthodes d'apprentissage non supervisé.

L'objectif de ce projet consiste à construire un modèle non supervisé qui prédit les activités humaines telles que marcher, monter et descendre les escaliers, s'asseoir, se tenir debout ou s'allonger. Des lectures de l'accéléromètre et du gyroscope ont été effectué par 30 individus dans une tranche d'âge de 19 à 48 ans au cours de six activités mentionnées précédemment.

Les lectures de l'accéléromètre sont divisées en accélérations de la gravité et du corps, qui ont chacune des composantes x, y et z.

Les lectures du gyroscope sont la mesure des vitesses angulaires qui ont des composantes x, y et z.

Les signaux de Jerk sont calculés pour les lectures d'accélération du corps.

Des transformations de Fourier sont effectuées sur les lectures de temps ci-dessus pour obtenir des lectures de fréquence.

Sur toutes les lectures du signal de base, le mean, max, mad, sma, arcoefficient, engerybands,entropy etc… sont calculés pour chaque fenêtre.

On a donc un vecteur de 561 caractéristiques.

Dans ce projet, on a utilisé dans un premier temps une ACP afin de réduire la dimension de notre jeu de données. Ensuite, on a appliqué une méthode de partitionnement en utilisant Kmeans, une classification hiérarchique (CAH) et la méthode de cartes auto-organisatrice en utilisant l'algorithme SOM.

II. ANALYSE EXPLORATOIRE :

La base contient 561 variables pour 7352 observations. On a ajouté une colonne « ID » pour les id des individus et une colonne « labels » pour les activités.

	labels	ID	tBodyAcc-mean()-X	tBodyAcc-mean()-Y	tBodyAcc-mean()-Z	tBodyAcc-std()-X	tBodyAcc-std()-Y	tBodyAcc-std()-Z	tB m
1	STANDING	1	0.28858451	-0.020294171	-0.13290514	-0.9952786	-0.9831106	-0.9135264	
2	STANDING	1	0.27841883	-0.016410568	-0.12352019	-0.9982453	-0.9753002	-0.9603220	
3	STANDING	1	0.27965306	-0.019467156	-0.11346169	-0.9953796	-0.9671870	-0.9789440	
4	STANDING	1	0.27917394	-0.026200646	-0.12328257	-0.9960915	-0.9834027	-0.9906751	
5	STANDING	1	0.27662877	-0.016569655	-0.11536185	-0.9981386	-0.9808173	-0.9904816	
6	STANDING	1	0.27719877	-0.010097850	-0.10513725	-0.9973350	-0.9904868	-0.9954200	

Showing 1 to 6 of 7,352 entries, 563 total columns

Figure 1 - Aperçu de la base

1- Le taux des activités pour chaque individu :

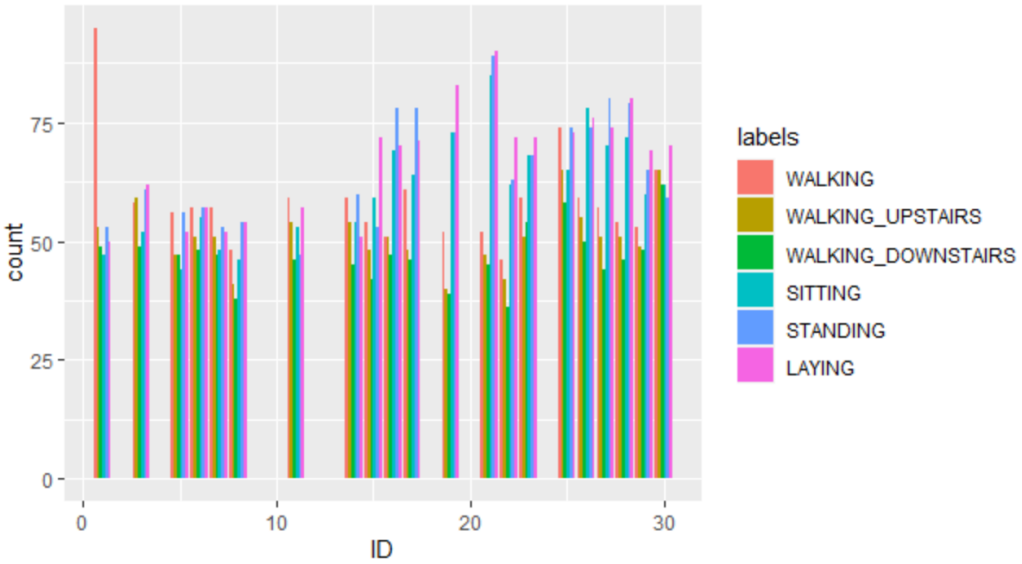

Figure 2 - Taux des activités pour chaque individu

2- Taux de chaque activité :

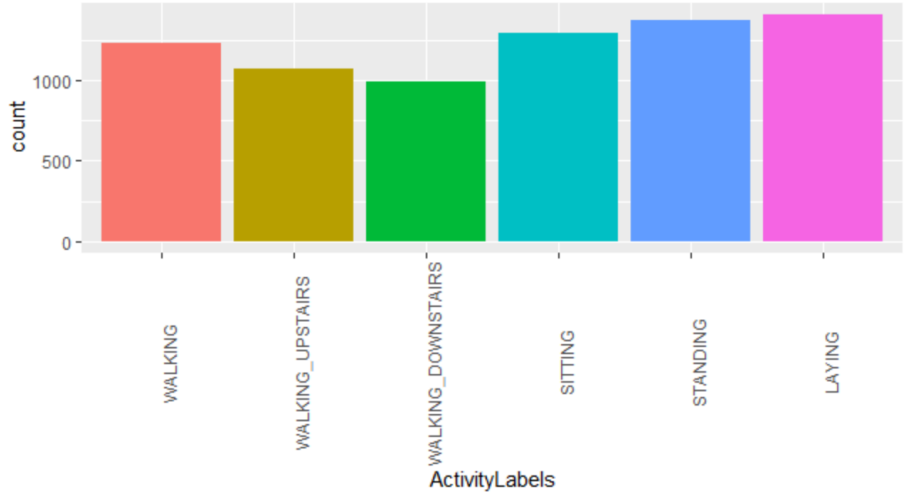

Figure 3 - Taux de chaque activité

Des figures 2 et 3, on peut remarquer que la base est bien équilibrée.

3- Courbe des activités stationnaires et activités mobiles :

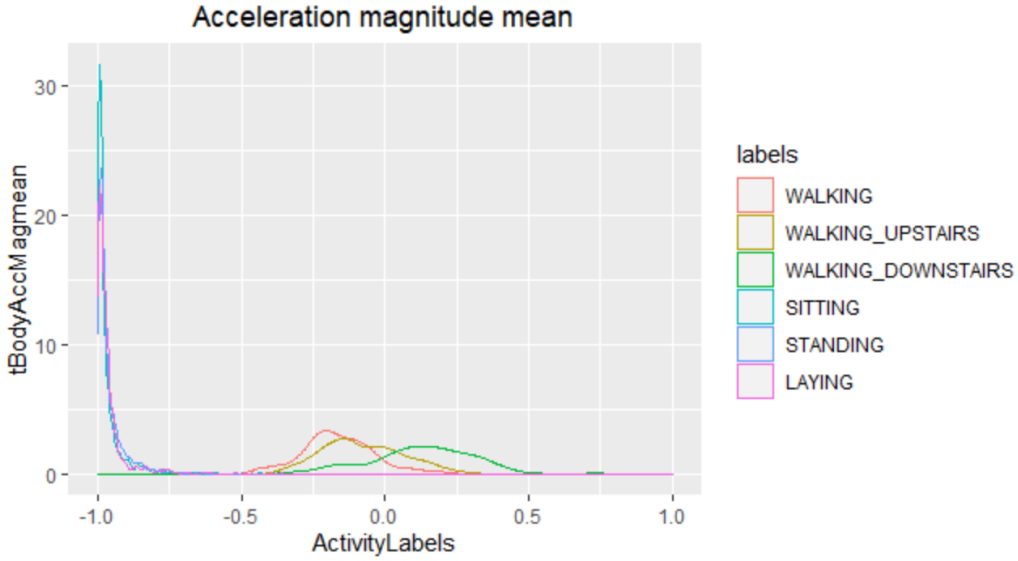

Figure 4 - Courbe des activités stationnaires et activités mobiles

De la figure 4, on peut distinguer deux activités différentes :

- stationnaires (quand l'individu s'assoie, se met debout et s'allonge).
- mobiles (quand l'individu marche, monte et descends les escaliers).

4- Boxplot de quelques variables :

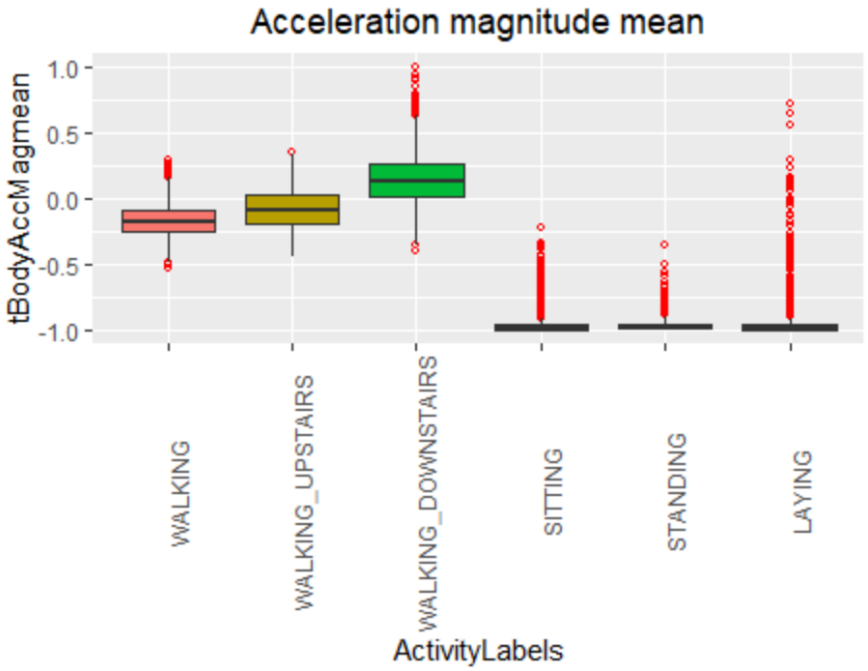

Figure 5 - Boxplot de la variable "tBodyAccMagmean"

La variable "tBodyAccMagmean" correspond à l'ampleur de l'accélération.

- Cas tBodyAccMagMean < -0,5 : Activités stationnaires.

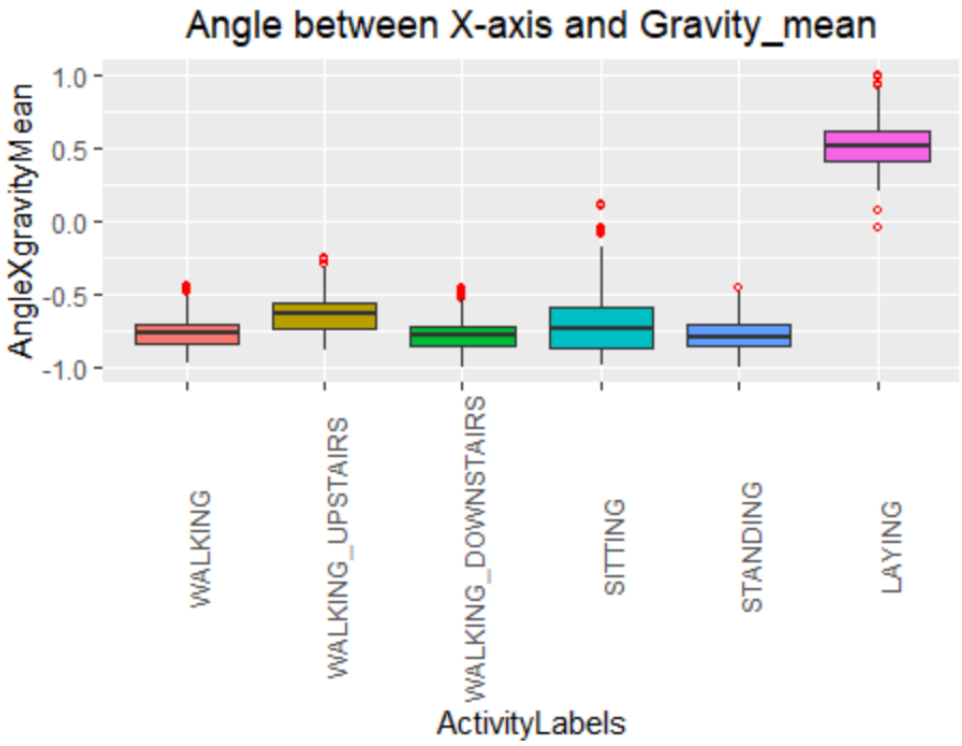

Figure 6 - Boxplot de la variable "AngleXgravityMean"

La variable "AngleXgravityMean" correspond à La composantes Accélération-Gravité (l'angle entre l'axe X et la moyenne de la gravité). Si elle est supérieure à 0, l'individus est allongé.

5- Visualisation avec TSNE :

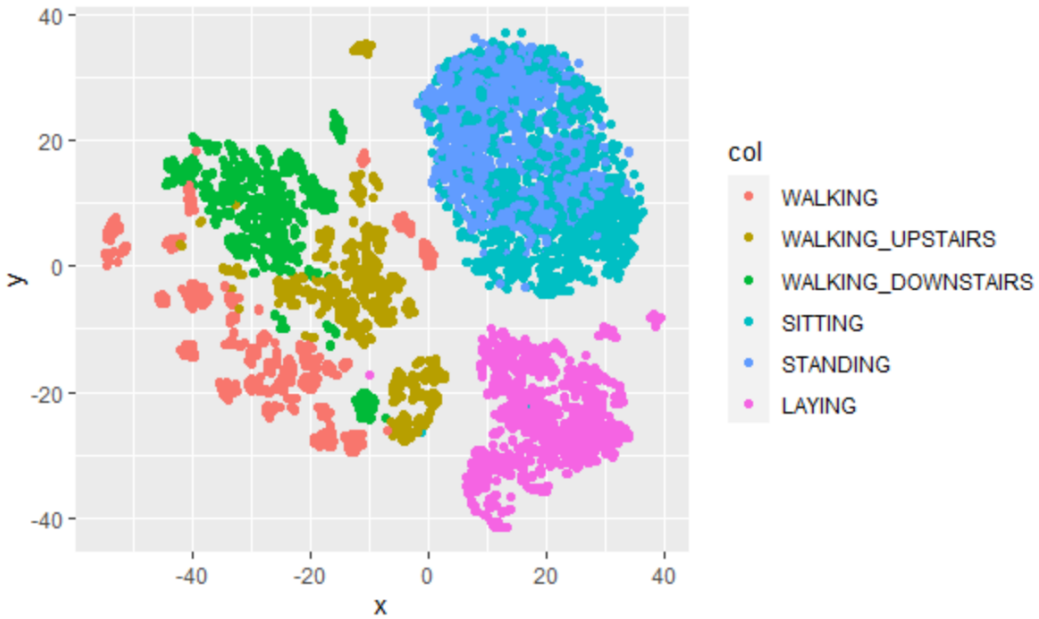

Figure 7 - TSNE vizualisation

On remarque que les classes sont bien séparées sauf les deux activités "s'asseoir" et "se mettre debout" qui chevauchent.

III. Modèles de la classification automatique :

Notre jeu de données contient des individus décrits par un nombre important de variables (561 toutes quantitatives) d'où l'intérêt d'utiliser une ACP (Analyse en Composantes Principales) afin de garder les variables les plus pertinentes à notre étude.

1- ACP :

Pour faire une ACP sur notre base, on a utilisé la fonction *prcomp()* et on a affiché le plot suivant (figure 8) de la variance cumulée.

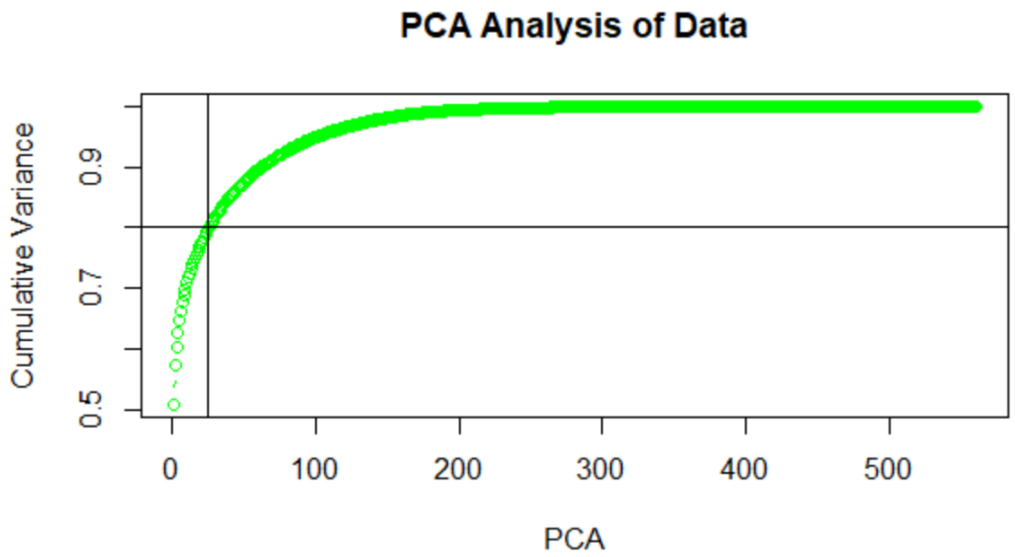

Figure 8 - Plot ACP

On remarque que 25 variables expliquent 80% de la variance totale, tandis que 100 expliquent 95%. On déduit qu'au-delà de 25 variables, chaque variable ajoutée contribue à moins de 1% à l'augmentation de la variance.

Pour le reste de l'étude, on a décidé alors de garder que 25 variables stockées dans la base *pca_data*.

2- Modélisation avec K-Means

Dans un premier temps, on a choisi d'appliquer le K-Means en choisissant 6 clusters (en se basant sur les 6 activités mentionnées précédemment).

Le résultat est comme suit :

```
> table(km$cluster)

   1    2    3    4    5    6
1210  944 1337 1716 1910  235
```

Figure 9 - Les 6 clusters obtenus de la K-Means

On a donc obtenu une qualité de répartition de 70% en devisant *betweeness* : la somme des carrées entre les clusters (la moyenne des distances entre les centres de cluster) par *totss* la somme totale des carrées.

On peut toutefois vérifier si le nombre de clusters choisi donne la meilleure répartition et ceci en appliquant les deux méthodes : Ward et Silhouette (figure 10).

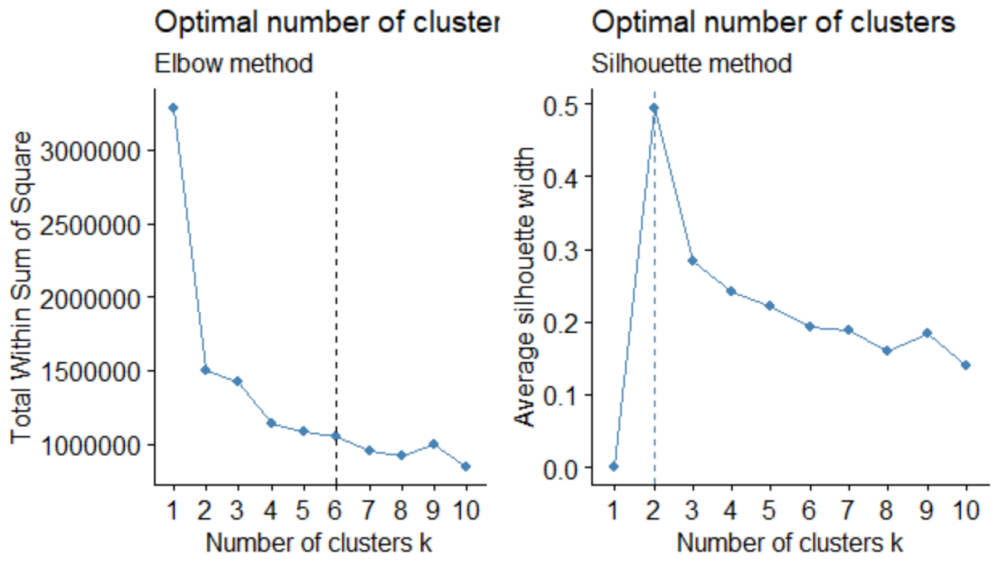

Figure 10 - Choix du nombre de clusters

On remarque alors que Ward nous propose 6 clusters tandis que Silhouette propose 2. On visualise donc nos données en utilisant 2 et 6 clusters (figure 11).

Le nombre de clusters proposé semble être évident, car, on peut détecter deux classes d'activités humaines : stationnaire et mobile. Comme on peut détecter 6 : marcher, monter et descendre les escaliers, s'asseoir, se tenir debout ou s'allonger.

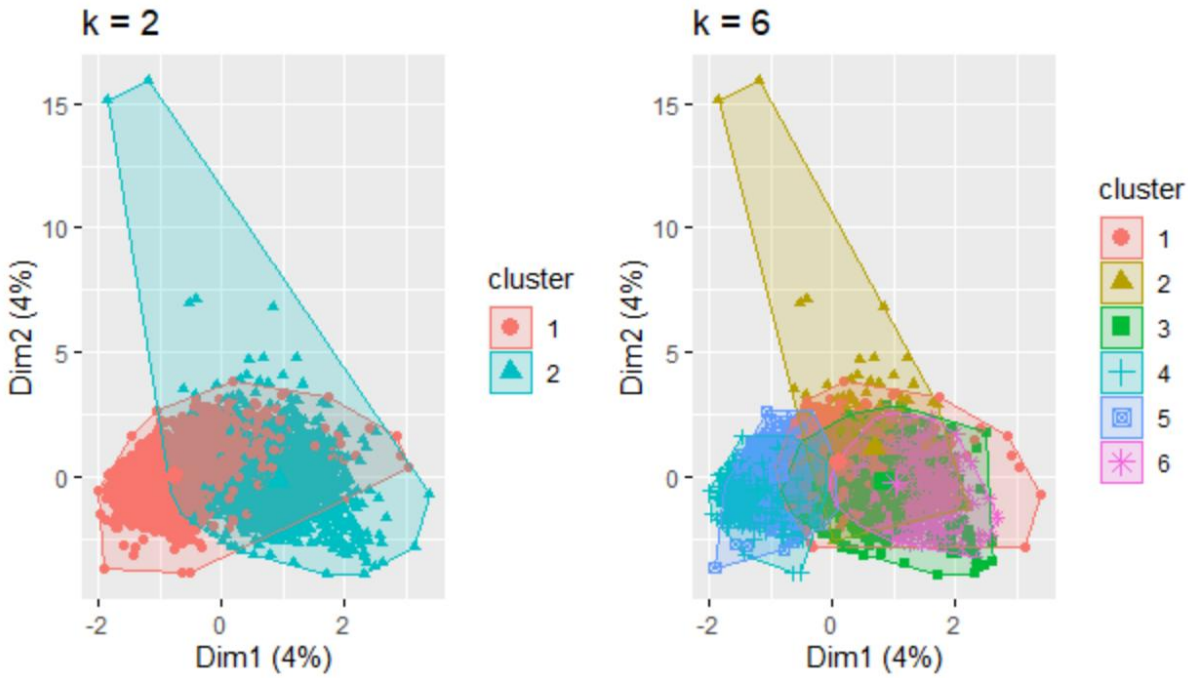

Figure 11 - K-Means avec 2 et 6 clusters

3- Modélisation avec CAH

Le choix du nombre des clusters est 6, en partant toujours du principe qu'il existe 6 activités, et on affiche le dendrogramme (Figure 12).

Pour cet affichage, on a utilisé plusieurs méthodes d'agrégation : Ward, Single, Complet…, et c'est avec Ward qu'on a obtenu le meilleur dendrogramme.

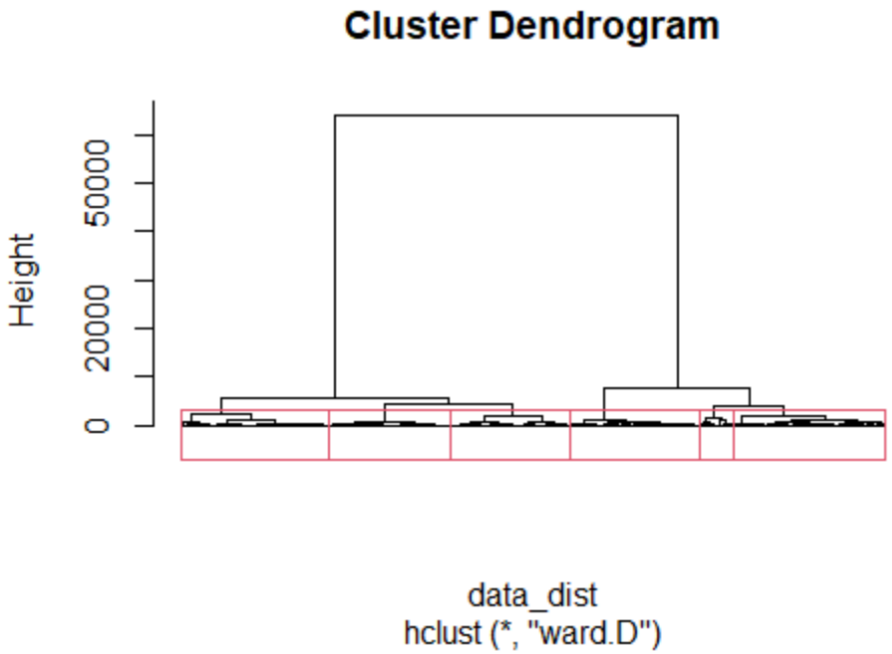

Figure 12 - Dendrogramme avec Ward

Afin de vérifier le nombre de clusters qui découpe parfaitement le dendrogramme en classes homogènes, on se base sur les sauts d'inertie.

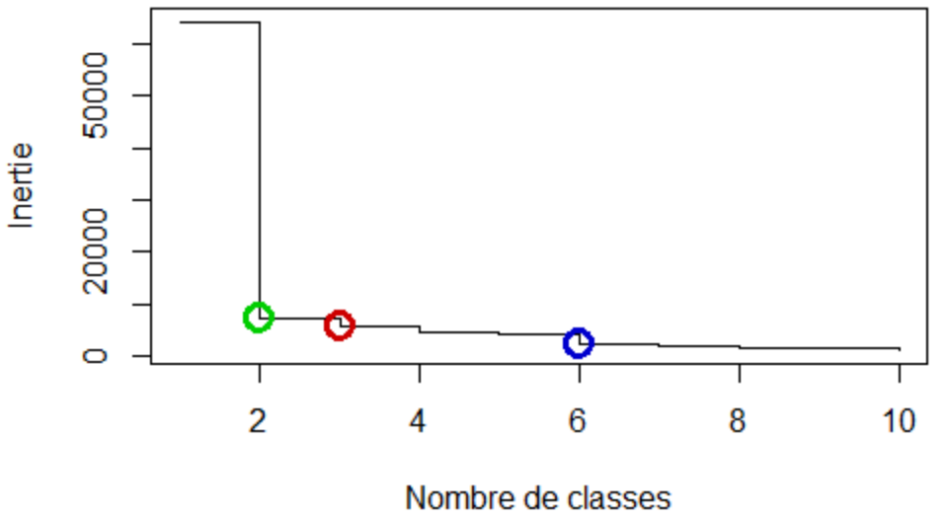

Figure 13 - Les sauts d'inertie

On remarque qu'il y a des sauts assez nets en 2, 3 et 6 classes, mais un découpage en deux classes minimise le critère d'inertie. Cependant, si l'on souhaite réaliser une analyse un peu plus fine, un nombre de classes plus élevé serait pertinent.

Pour les répartitions en 2 et 6 clusters, le raisonnement reste le même que celui du K-means, et pour la répartition en 3 clusters, en se basant sur les résultats de l'analyse exploratoire (Figure 5), on peut déduire que les 3 classes peuvent être : une classe pour les activités stationnaire, une autre pour les activités mobiles nécessitant moins d'efforts et une troisième classe pour les activités mobile augmentant l'accélération (descendre de l'escalier) .

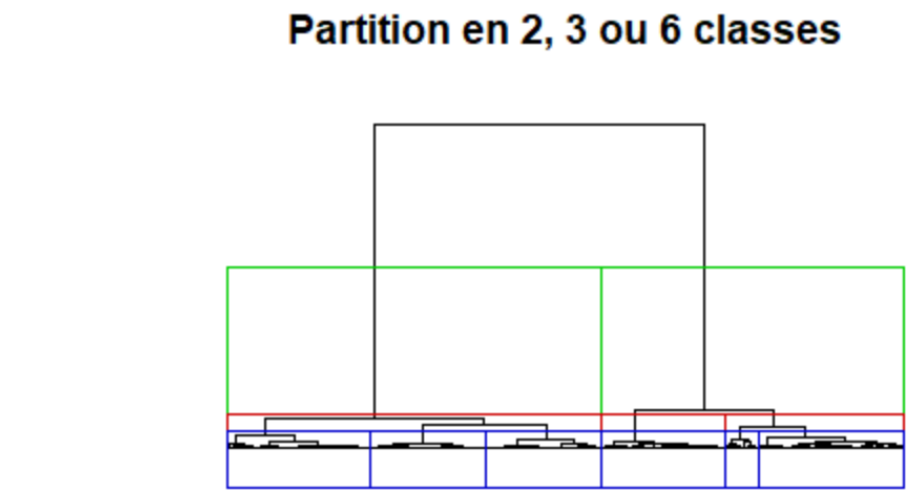

Figure 14 - Partition en 2,3 ou 6 classes

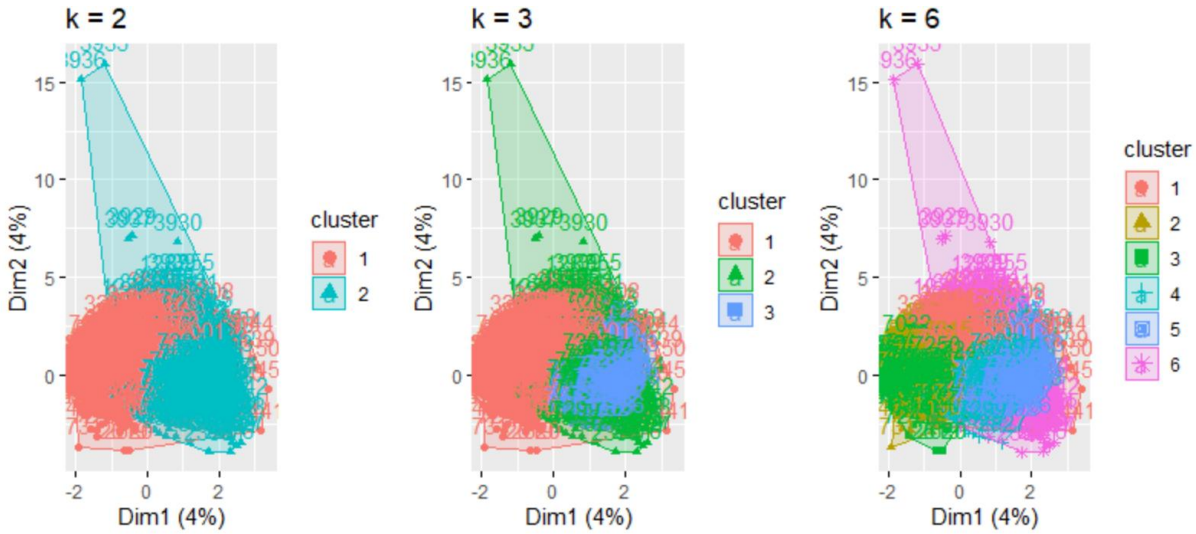

Figure 15 - Différents clusters obtenus du CAH

Une partition en 2 groupes donne le meilleur résultat, ce qui est évident.

Une partition en 3 groupes donne une classe dominante par rapport aux autres.

Enfin, une partition en 6 groupes donne des résultats plus ou moins cohérents.

```
> # See the number of observations in each cluster
> table(sub_grp_2)
sub_grp_2
   1    2
4067 3285
> table(sub_grp_3)
sub_grp_3
   1    2    3
4067 1941 1344
> table(sub_grp_6)
sub_grp_6
   1    2    3    4    5    6
1546 1255 1266 1589 1344  352
```

Figure 16 - Nombre d'observations dans les classes

Remarque : la liste des individus par groupes est donnée par la fonction :

sapply(unique(sub_grp_6),function(g)data$ID[sub_grp_6 == g])

4- Modélisation avec SOM

Dans un premier temps, on a choisi la taille de la grille, ensuite on a fait l'apprentissage.

```
> print(summary(som.model))
SOM of size 15x15 with a hexagonaltoroidal topology and a bubble
 neighbourhood function.
The number of data layers is 1.
Distance measure(s) used: sumofsquares.
Training data included: 7352 objects.
Mean distance to the closest unit in the map: 9.344.
NULL
```

Figure 17 - Résultats du SOM

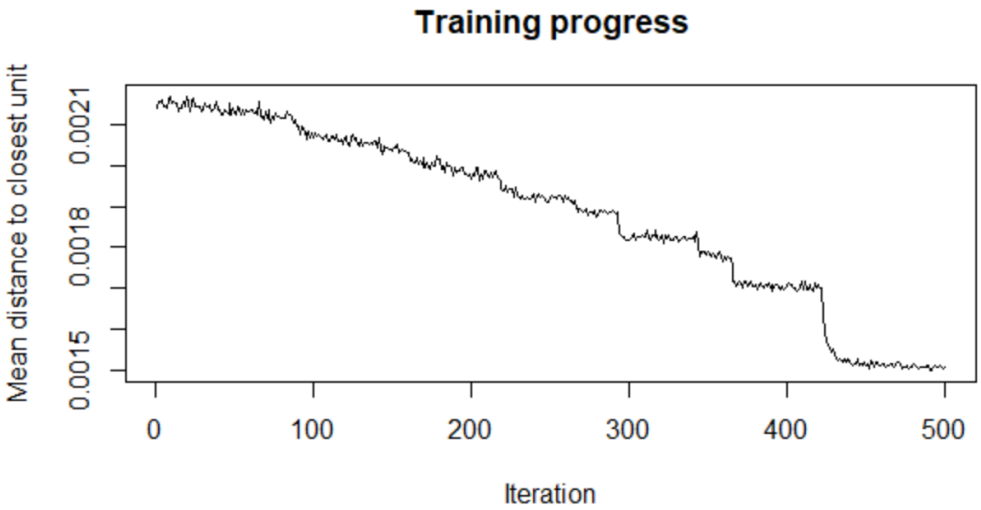

Figure 18 - Progression de l'apprentissage

La figure 18 montre la progression de l'apprentissage du modèle, on remarque que la courbe diminue continuellement ensuite elle stationne, ce qui signifie qu'on a choisi un nombre d'itérations adéquat.

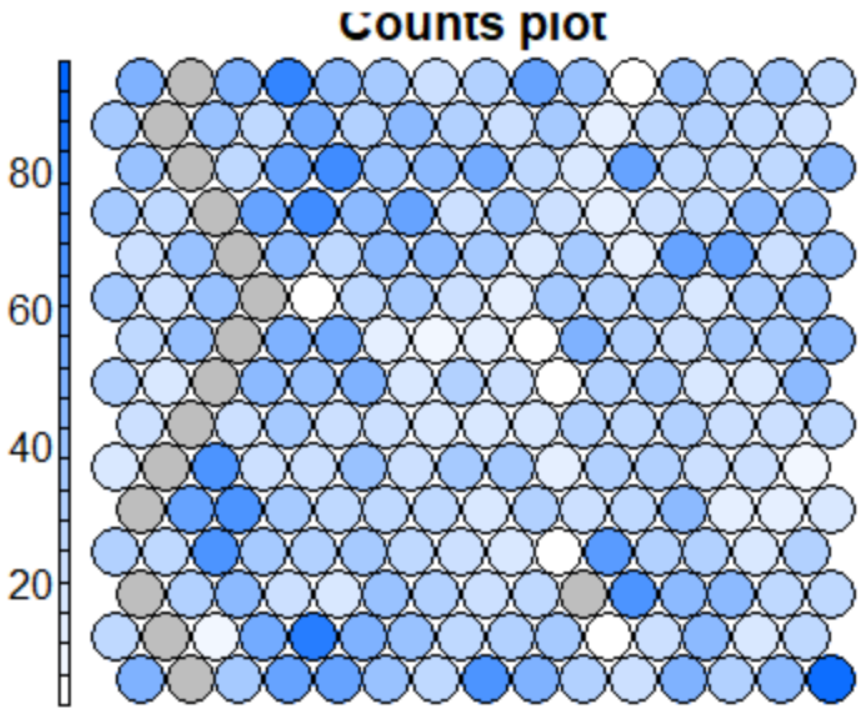

Figure 19 - Counts plot

La figure 19 montre le nombre d'instances dans les cellules, ceci permet d'identifier les zones à forte densité.

Il est important de vérifier si on a des nœuds vides, cette information est accessible via le composant *unit.classif* du modèle SOM.

```
> # number of instances assigned to each node
> nb <- table(som.model$unit.classif)
> View(sort(nb, decreasing=F))
> #check if there are empty nodes
> print(length(nb))
[1] 210
> sum(is.na(nb))
[1] 0
```

Figure 20 - Vérification des nœuds vides

Pour les 7352 individus, l'algorithme SOM les a répartis sur 210 nœuds où chaque nœud contient au moins une observation.

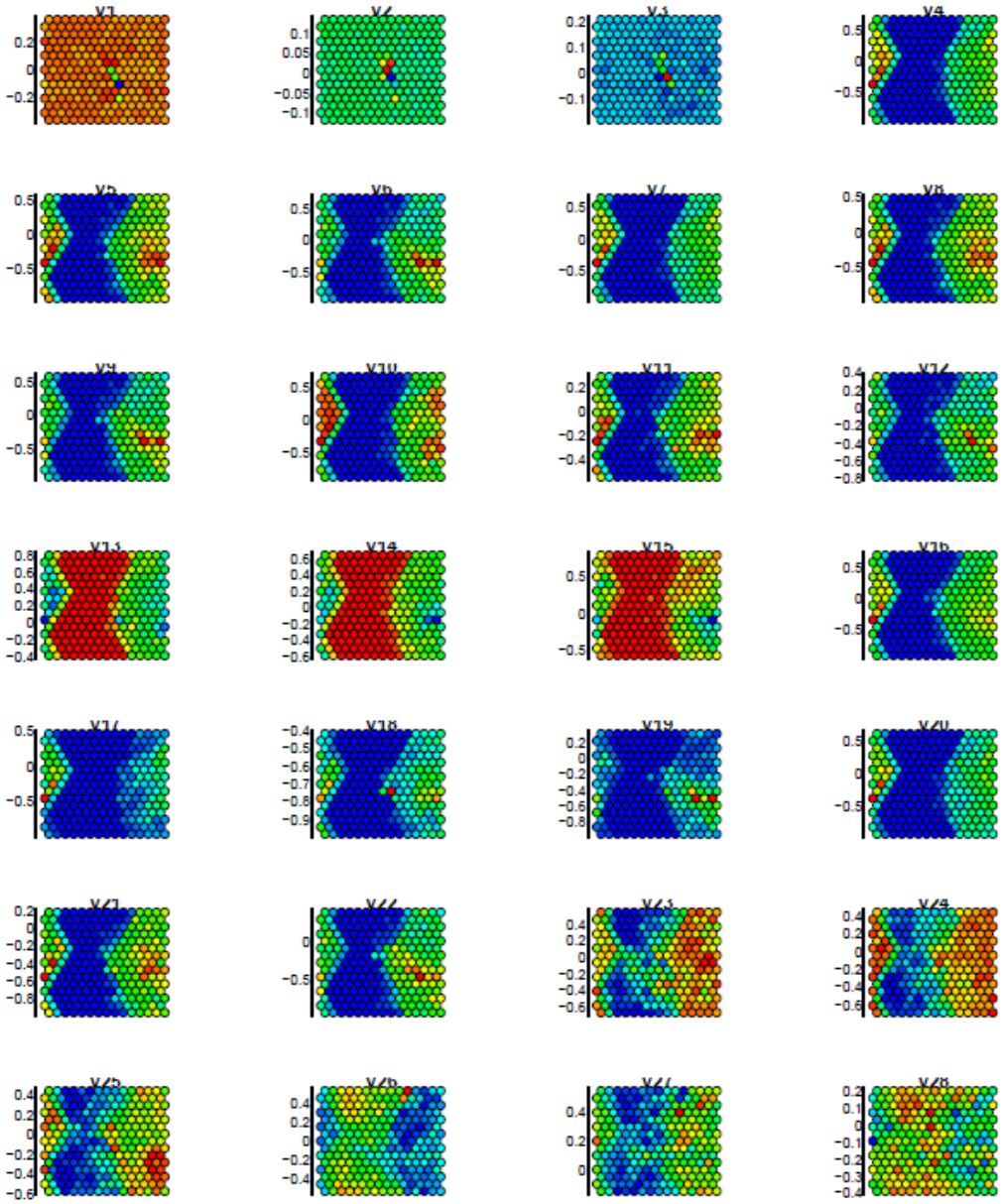

Figure 21 – Heatmaps des 28 premières variables

Les Heatmaps nous permettent de faire un graphique pour chaque variable, en mettant en évidence les contrastes entre les zones de valeurs élevées et faibles. Cette description univariée est plus facile à comprendre. Le schéma de couleurs utilisé combine des couleurs rouges (respectivement bleues) avec des valeurs élevées (respectivement valeurs faibles).

Cette représentation nous permet également d'avoir une idée sur le nombre de clusters qu'on peut construire et les variables caractérisant chacun d'eux.

En effet, pour l'application du SOM, on a utilisé la totalité des variables (561 variables). Du fait, en observant les heatmaps générées à partir de ces variables, il est possible d'avoir des informations importantes sur les classes en interprétant les relations entre les différentes heatmpas.

Par exemple, les variables ayant des heatmaps similaires sont fortement corrélées. On peut donc voir que la SOM est un outil puissant et efficace pour valider la corrélation attendue entre les différentes variables et pour prédire des corrélations inconnues.

L'utilisation de la SOM pour générer ces heatmaps présente l'avantage de permettre une compréhension rapide, visuelle et qualitative des relations entre les variables. Celles présentant une forte corrélation auront des heatmaps similaires, avec des régions rouges et bleues similaires.

Dans l'algorithme SOM, les cellules adjacentes ont des codebooks similaires. On peut donc à partir de ces cellules effectuer une classification comme la CAH qui est souvent utilisée dans ce contexte. (Figure 22)

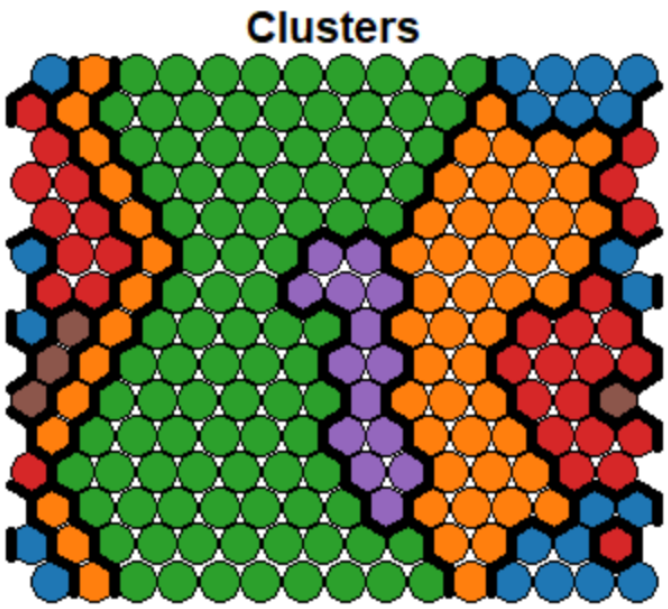

Figure 22 - Clusters

En effet, il était difficile de détecter les régions en se basant seulement sur les heatmaps, l'affichage de 561 heatmaps prendra énormément de temps. Du fait, le choix de 6 clusters vient d'une connaissance préalable des classes construisant notre base,

Toutefois, en observant les heatmaps des 28 premières variables de notre base de données (Figure 21), on peut déjà constater la présence de la classe verte, et qui est caractérisée par les variables V4 jusqu'à V22.

IV. Conclusion:

On a appliqué la K-mean et la CAH en ne gardant que les variables les plus pertinentes après application de l'ACP, et on a obtenu pour chacune d'elles des résultats plus ou moins cohérents avec les classes constituant notre base de départ.

Pour la SOM, on l'a appliqué en incluant toutes les variables (561), ce n'était pas évident d'afficher les heatmaps de toutes ces variables afin de détecter des classes. Par conséquent, le choix de 6 classes était basé sur la connaissance préalable des classes de départ et les résultats des méthodes CAH et K-means qu'on a appliqué précédemment.