

# Projet d'Analyse de données : *Vélib*

Arthur BOIVERT, Paul CORBALAN, Alexia GRENOUILLET, Hugo LELIEVRE, Alexandre LIBOUREL

Mardi 17 mai 2022

## Introduction

Pour cette analyse, nous travaillons avec le jeu de données des *Vélib'* parisiens. Afin de construire le dataset *Vélib'*, des données ont été récoltées à Paris, à chaque heure de la journée, du lundi 1er au dimanche 7 septembre 2014. Il est constitué de :

- **data** : le taux de remplissage des 1189 stations, (nombre de vélos disponibles / nombre de points d'accueil pour les vélos), à 181 intervalles de temps.
- **position** : la longitude et la latitude des 1189 stations.
- **bonus** : indique si la station est sur une colline (**bonus = 1**).
- **names** : le nom des stations.

Les données dont nous disposons correspondent à des "profils d'occupation" des stations *Vélib'* sur la période du lundi 2 au dimanche 7 septembre 2014.

Dans le document qui suit, on définit par "profil d'occupation" d'une station le rapport entre le nombre de vélos disponibles et le nombre de points d'accueil pour les vélos. Autrement dit, une occupation valant 1 signifie que la station est entièrement occupée (*ie* tous les vélos sont disponibles pour les usagers), tandis qu'une occupation valant 0 traduit le fait que la station est vide (*ie* tous les vélos ont été loués).

Si l'on se place d'un point de vue d'analyse des données, on identifie les individus de notre jeu de données comme étant les 1189 stations *Vélib'*. Les variables sont quant à elles les heures de la semaine (sachant qu'une heure correspond à un pas de temps).

L'objectif principal de ce projet est de détecter la présence de clusters dans les données. Ces clusters correspondent en pratique aux locations habituelles des usagers. Dans l'idéal, nous voudrions réussir à dégager certains "profils" de station afin de pouvoir anticiper le comportement des utilisateurs de *Vélib'*. Par exemple, si certaines stations ont un taux d'occupation très faible à certaines heures de la journée, et d'autres sont au contraire quasiment pleines, on pourrait suggérer de déplacer une partie des vélos des secondes stations afin de les rajouter dans les premières.

Pour ce faire, nous commencerons par mener une analyse descriptive du jeu de données afin d'identifier les premières tendances des usagers. Nous effectuerons ensuite une Analyse en Composante Principale (ACP) afin de réduire la dimension de notre problème et ainsi le rendre plus interprétable. Dans un troisième temps, nous réaliserons 3 types de clustering :

- *K-means*
- Clustering à Ascendante Hiérarchique (CAH), aussi appelé *agglomerative clustering* (clustering agglomératif)
- Gaussian Mixture

Chacun de ces clusterings sera mené à la fois sur les données brutes du problème et sur les données décomposées dans la base de l'ACP. Enfin, nous intégrerons la variable qualitative binaire *bonus* à notre étude par une Analyse des Correspondances Multiples (ACM), puis effectuerons un *K-means* sur les 5 composantes principales de l'ACM. Ceci nous permettra de déterminer si la variable *bonus* a une réelle importance pour le clustering sur le jeu de données *Vélib'*.

Enfin, nous terminerons par une interprétation des différents résultats obtenus en proposant des solutions pour la redistribution des vélos entre les stations.

# 1 Analyse descriptive des données

## 1.1 Etude préliminaire des individus

On commence par afficher les graphes des occupations pour chaque station afin d'essayer d'identifier d'éventuelles tendances et ressemblances entre les stations.

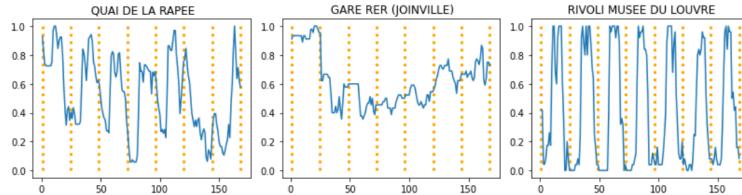


FIGURE 1 – Occupation en fonction de l'heure pour 3 stations choisies pour leurs profils différents

On remarque alors plusieurs tendances générales :

- certaines stations ont un profil d'occupation quasiment périodique (où une période correspond à une journée), comme à Rivoli-Musée du Louvre ;
- certaines présentent un profil plutôt chaotique ; aucune régularité ne ressort, comme à la Gare RER (Joinville) ;
- d'autres se situent entre les 2 précédentes catégories : leur périodicité n'est pas clairement identifiable, mais elle n'est pas non plus totalement chaotique, comme au Quai de la Rapée.

On trace ensuite un boxplot pour observer l'occupation des stations à chaque heure de la semaine (avec les heures ordonnées dans le temps).

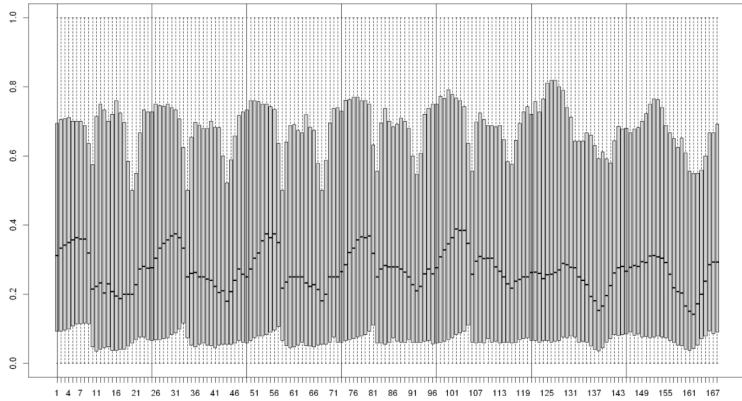


FIGURE 2 – Boxplot des stations par heure

On peut remarquer plusieurs points importants :

- de manière générale, ces données sont plutôt dissymétriques : la différence entre le 0,75-quantile et la médiane est plus élevée qu'entre la médiane et le 0,25-quantile.
- la dispersion des boîtes est relativement importante, ce qui signifie que l'occupation varie beaucoup selon la station considérée. Si l'on observe plus précisément, certaines variations quotidiennes sont remarquables. D'une part, la dispersion est plus faible lorsque la position est faible ; d'autre part, elle semble plus importante le week-end.
- si l'on s'intéresse plus précisément à la courbe en escalier obtenue à partir des médianes des boîtes, on s'aperçoit que pour les jours "ouvribles" (*i.e.* du lundi au vendredi inclus), la courbe est périodique, avec une période valant 24h. Les usagers ont également tendance à emprunter les vélos sur une plage horaire comprise entre 10h et 21h. En revanche, pour les jours de week-end, la courbe est presque symétrisée par rapport aux jours ouvrables ; les usagers ont plutôt tendance à pédaler entre 14h et 22h.

## 1.2 Partitionnement des stations en fonction de leur localisation

On utilise à présent les coordonnées géographiques des stations (latitude et longitude) pour les localiser sur une carte 2D. Notre objectif dans cette section est de déterminer si l'occupation d'une station Velib' est significativement différente en fonction de l'altitude à laquelle elle se situe (est-elle sur une colline ou sur du plat ?). Pour cela, on utilise la variable *bonus* du jeu de données Velib', qui vaut 1 si la station est située sur une colline, 0 sinon. On obtient la carte 2D suivante :

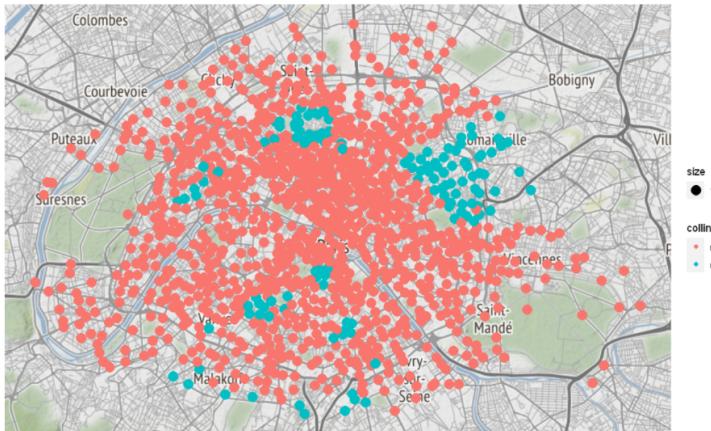


FIGURE 3 – Localisation des stations Velib' sur une carte 2D

On reprend ensuite notre étude préliminaire ; on analyse disjointement les tendances pour les deux types de stations afin de déterminer si le fait qu'une station soit située sur une colline change significativement son occupation.

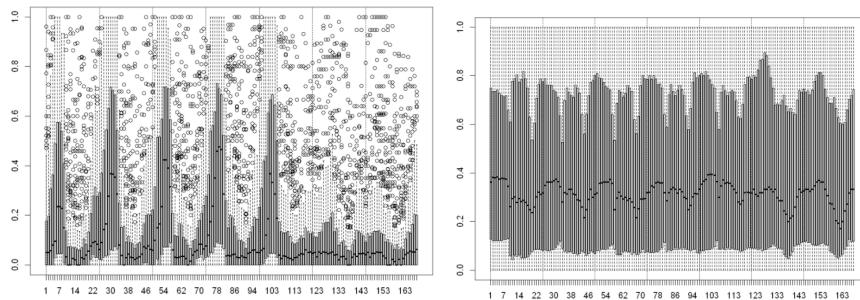


FIGURE 4 – Boxplot de l'occupation par heure des stations Velib' situées sur une colline (à gauche) et sur du plat (à droite)

La première remarque que l'on peut soulever en comparant ces deux boxplots est que l'occupation des stations situées sur une colline est beaucoup plus irrégulière que celle des stations situées sur du plat. Pour les "stations collines", on distingue deux tendances :

- la semaine, l'occupation est très faible à partir du milieu de la matinée (vers 9h) jusqu'à la fin d'après-midi (vers 18h), et redevient assez élevée du début de soirée jusqu'au lendemain matin. On peut donc raisonnablement supposer que de telles tendances résultent du comportement des utilisateurs, qui prennent le vélo pour se rendre sur leur lieu de travail dans la matinée, et le ramènent le soir une fois leur journée terminée.
- le week-end, l'occupation est toujours très basse, ce qui laisse penser que les Parisiens vivant sur une colline sont férus d'activités en extérieur lors de leurs jours de repos.

Pour les stations situées sur du plat, on ne peut pas rajouter grand-chose à nos observations préliminaires sur le comportement général des stations. Le motif périodique obtenu à partir du tracé des médianes semble correspondre à celui obtenu précédemment, tout comme les paramètres de dispersion et de dissymétrie.

## 2 Analyse en Composantes Principales

Afin de réduire la dimension du problème, et ainsi rendre nos résultats plus interprétables, on met en place une Analyse en Composantes Principales.

### 2.1 Choix du nombre de composantes

On commence par tracer 3 graphes qui permettent de sélectionner le nombre de composantes à garder :

- le graphe du pourcentage d'inertie expliquée par composantes ;
- le graphe du pourcentage cumulé croissant d'inertie expliquée par les composantes ;
- les boxplots des inerties selon chaque composante.

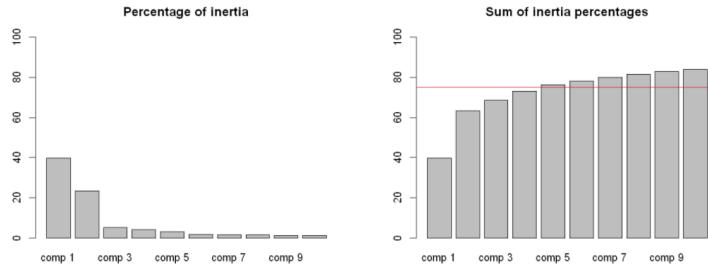


FIGURE 5 – Pourcentages d'inertie en fonction des composantes de l'ACP

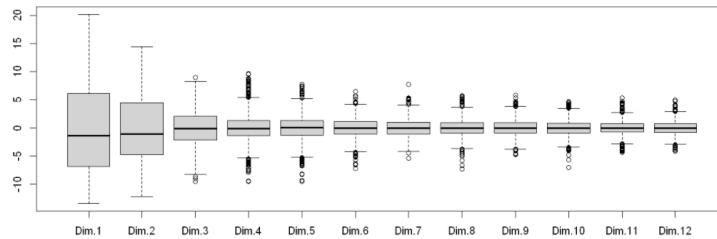


FIGURE 6 – Boxplot des coordonnées selon chaque composante

Afin de trouver le meilleur compromis possible entre un nombre de composantes relativement faible et un pourcentage de variance expliquée élevé, on utilise la méthode du "coude". Les 2 premières composantes expliquent une grande partie de la variance et doivent être gardées. Ensuite, on observe une diminution de variance expliquée assez significative entre la 5ème et la 6ème composante. De plus, à partir de 5 composantes, on explique plus de 75 % de la variance. Enfin, la dispersion des boxplots des coordonnées sur les axes principaux de l'ACP semblent se stabiliser à partir du sixième. Finalement, garder 5 composantes semble donc un choix raisonnable.

On utilise ensuite le cercle de corrélation afin d'essayer de trouver une interprétation aux 2 premiers axes de l'ACP.

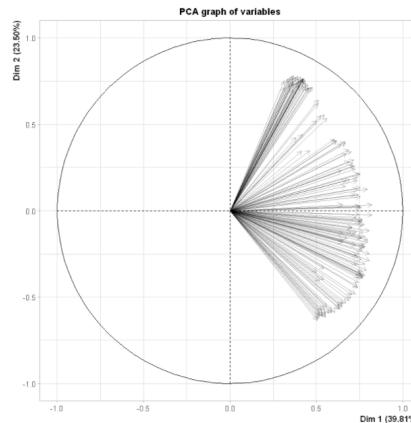


FIGURE 7 – Graphe des variables dans les deux premiers axe de l'ACP

En premier lieu, on remarque que toutes les flèches (qui représentent les heures), sont proches du bord du cercle. Cela signifie que toutes les variables sont proches de leur projection sur le plan des 2 premiers axes ; on peut donc en déduire que l'ACP permet de réduire efficacement la dimension en expliquant les données efficacement. Interprétons maintenant ces axes :

- **1er axe** : on remarque que les abscisses de toutes les variables sont positives et comprises entre 0,3 et 0,8. On peut l'interpréter comme étant le "chargement" moyen des stations ;
- **2ème axe** : ce second axe peut s'interpréter comme le contraste de chargement entre le jour et la nuit durant les jours ouvrés.

Enfin, on représente les stations sur le plan des 2 premières composantes de l'ACP.

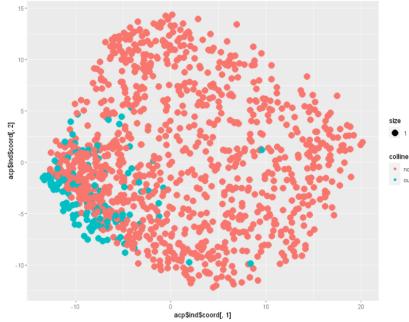


FIGURE 8 – Graphe des stations sur le plan des 2 premières composantes de l'ACP

On remarque alors que les stations situées sur une colline sont regroupées dans une même zone. Cela ne semble pas étonnant ; avec l'analyse descriptive, nous avions déjà noté que leur comportement se distinguait de celui des autres stations. Toutefois, d'autres stations sont situées dans cette même zone. Il semble donc intéressant d'effectuer un clustering afin de vérifier si ces stations ont les mêmes types de boxplots que celles sur une colline.

## 3 Clustering

On va maintenant chercher à identifier des "clusters" dans notre jeu de données. On va partitionner l'ensemble des stations de manière à avoir des groupes de stations au profil de chargement hebdomadaire similaire. On cherche ainsi à obtenir dans chaque groupe une variance plus petite que la variance totale du jeu de données. Si la variance est assez petite dans chaque groupe, on sera alors capable de prévoir le chargement d'une station à n'importe quelle heure.

Nous allons dans la suite essayer de déterminer le meilleur partitionnement possible de notre jeu de données via différents algorithmes de clustering. Pour l'ensemble des algorithmes de clustering présentés, on utilise la métrique de *Ward* : ce critère induit, à chaque étape de regroupement, une minimisation de la décroissance de la variance interclasse.

### 3.1 K-means

Le clustering *K*-means permet de partitionner notre jeu de données en *K* clusters et fonctionne de la façon suivante :

- On initialise *K* centroïdes comme *K* points (ici des stations) tirés de façon aléatoire parmi les différents points du jeu de données et sans remise.
- Pour chaque points du jeu de données, on associe le point au plus proche centroïde. Cela crée alors *K* classes distinctes.
- On calcule le nouveau centroïde de chaque groupe comme la moyenne de tous les points de la classe obtenue. Puis on recommence depuis l'étape 2 jusqu'à ce que l'algorithme converge.

On cherche ici à minimiser la variance intraclasses, définie comme la moyenne des distances au carré entre les individus et le centroïde de leur classe d'appartenance. Cet algorithme converge toujours, mais pas toujours vers la même solution (exemple Figure 9) ; en effet, *K*-means converge vers des minimums locaux, et non pas vers un minimum global. Cela est dû au fait que le nombre de partitions possibles pour l'ensemble des points du jeu de données est très conséquent. Il n'est donc pas question de chercher à optimiser la variance intraclasses sur toutes les partitions possibles. L'algorithme se contente donc de converger vers un minimum local, qui varie aléatoirement selon le choix des centroïdes lors de l'initialisation de *K*-means.

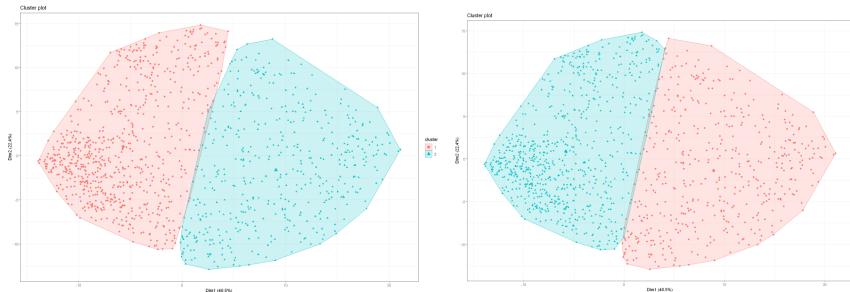


FIGURE 9 – Différents clusters obtenus selon le choix des centroïdes lors de l’initialisation de  $K$ -means.

Pour le choix du nombre de clusters nous avons tracé l’inertie observée en fonction du nombre de clusters. Pour décider combien de clusters on va considérer, on utilise la ”méthode du coude”. Cependant il est difficile de voir une variation évidente de l’inertie dans le graphique. Le nombre de cluster optimal semble cependant être entre 4 et 6.

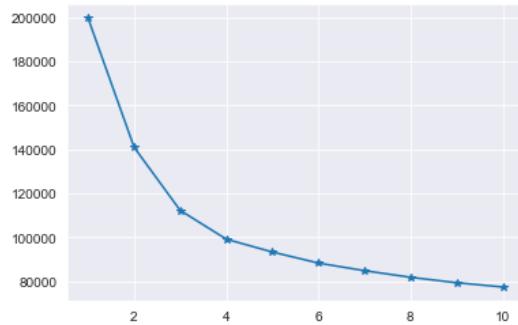


FIGURE 10 – Évolution de l’inertie en fonction du nombre de clusters

Les méthodes de clustering que l’on va employer après  $K$ -means nous donne un nombre optimal de 5 clusters (voir section Agglomerative Clustering) et c’est donc le nombre de clusters que nous allons considérer.

Nous obtenons alors le découpage suivant :

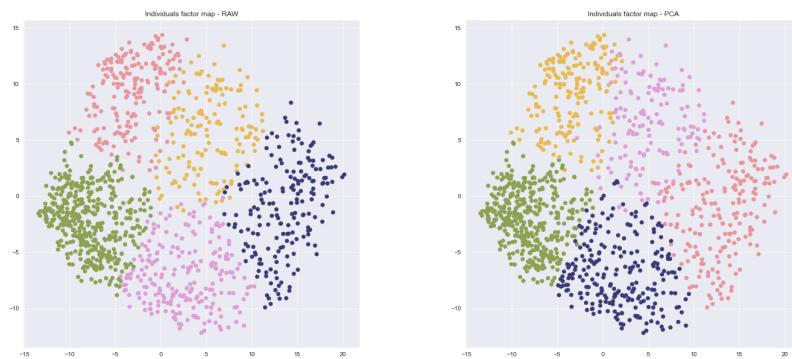


FIGURE 11 – Clustering du jeu de données sur les données brutes (à gauche) et sur le coordonnées de l’ACP (à droite), selon cinq classes.

On remarque une petite différence sur les clusters créés en fonction de si l’on considère les coordonnées de l’ACP ou si l’on considère les données brutes. Cependant, les clusters créés sont sensiblement identiques.

On peut maintenant regarder la dispersion observée dans chacun des clusters créés :

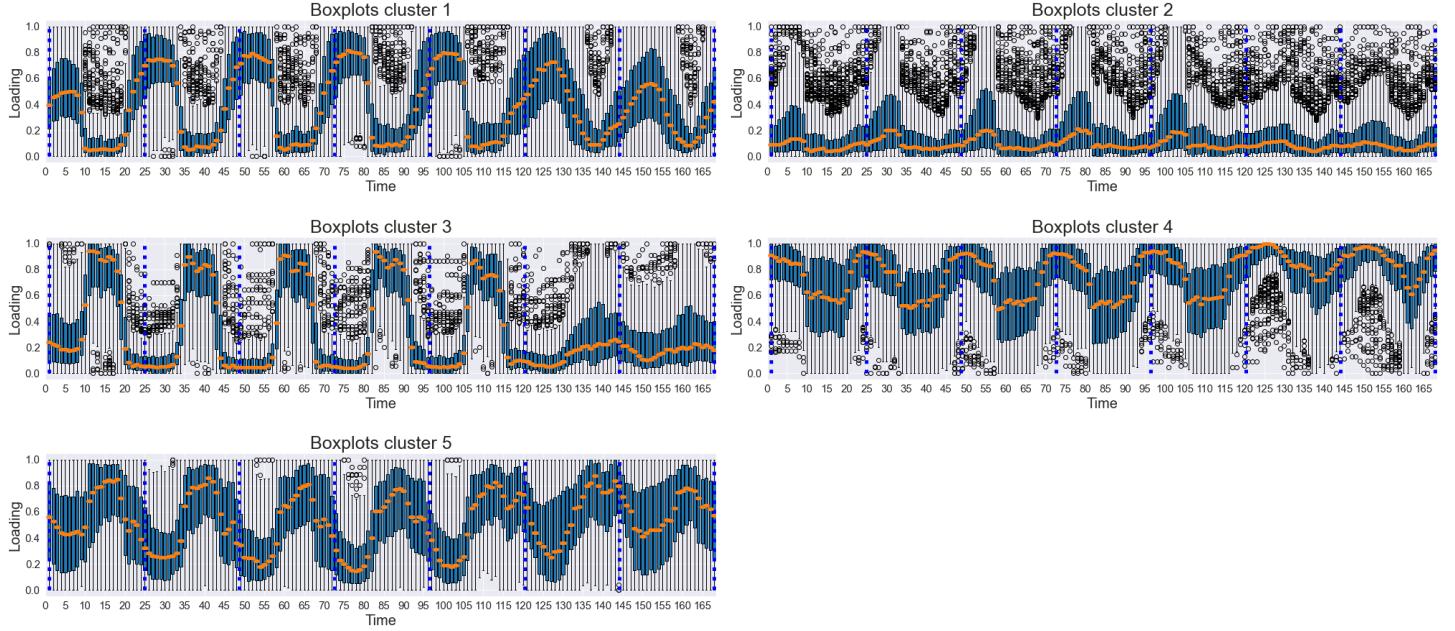


FIGURE 12 – Boxplots représentant la répartition de la capacité des stations *Vélib* à chaque heure de la semaine dans les différents clusters.

Les 5 clusters obtenus présentent chacun un profil de boxplots différents au fil des heures de la semaine. Chacun de ces clusters est caractérisé par une habitude des utilisateurs de *Vélib* où une caractéristique des stations. Par exemple, les clusters 1 et 3 sont en opposition de phase en semaine : tandis que les stations du cluster 1 sont remplies la nuit et vides le jour, les stations du cluster 3 sont vides la nuit et pleines le jour. On interprète cela comme le fait que les stations du premier cluster sont les stations où les travailleurs habitent et celles du troisième cluster sont les stations proches de leur lieu de travail. On reconnaît également la répartition obtenue pour les collines dans le cluster 2. On peut s'en rendre compte avec le graphique suivant :



FIGURE 13 – Clustering *K*-means selon 5 classes et mise en évidence des stations situés sur des collines

### 3.2 Agglomerative Clustering

L’Agglomerative Clustering, ou classification hiérarchique (CAH), est une autre méthode de partitionnement. Nous considérons ici la classification hiérarchique **ascendante** (*CAH*). L’initialisation de cet algorithme consiste à calculer un tableau de distances entre les individus du jeu de données. Le CAH commence ensuite par partitionner trivialement les  $n$  individus ; on a donc autant de classes que d’individus. A chaque étape, il cherche ensuite à constituer des classes par agrégation des deux éléments les plus proches de la partition de l’étape précédente. Le CAH s’arrête lorsque l’on obtient une seule et unique classe.

- Cette méthode de classification automatique se base sur une mesure de dissimilarité entre les individus : la distance euclidienne dans notre cas.
- Nous avons aussi besoin de préciser un *linkage criterion* (critère de liaison), qui récupère les données renvoyées par la mesure de dissimilarité, et groupe les individus en clusters selon leur similarité. Nous utilisons ici le linkage *ward*, qui minimise l'inertie intraclasse.

On peut ensuite tracer un *dendrogramme*, une représentation très visuelle de la hiérarchie obtenue par l'algorithme.

**Cluster Dendrogram**

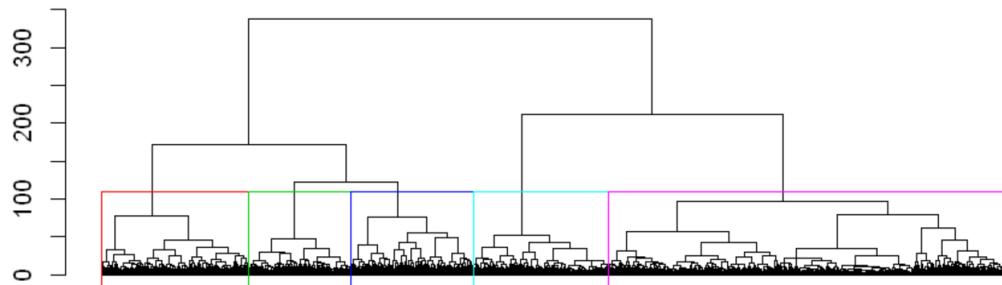


FIGURE 14 – Dendrogramme permettant de déterminer le nombre de clusters idéal pour notre jeu de données.

Sur la précédente figure, chaque feuille représente un individu, et au fur et à mesure que l'on remonte dans l'arbre, les individus les plus similaires sont regroupés sur une même branche. Finalement, on coupe cet arbre à une certaine hauteur afin de séparer nos individus en un certain nombre de clusters (de notre choix).

Le choix le plus logique et qui permet d'analyser au mieux les données est de choisir 5 clusters. Nous obtenons alors la séparation suivante :

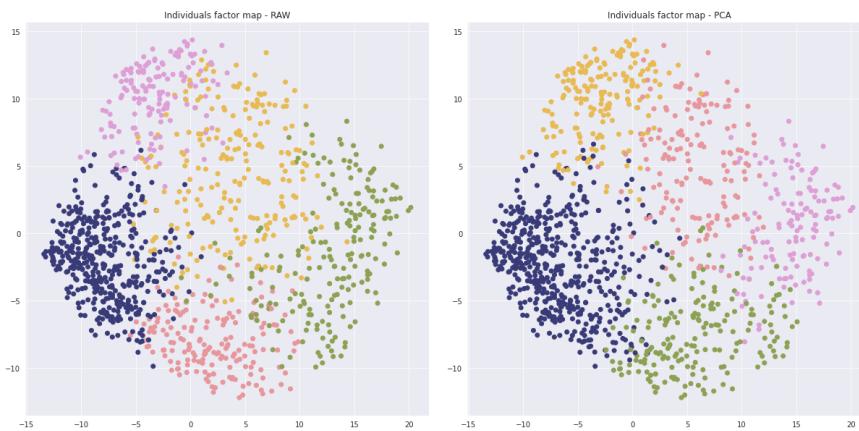


FIGURE 15 – Clustering du jeu de données sur les données brutes (à gauche) et sur les 5 premières composantes de l'ACP (à droite), selon cinq classes.

On remarque quelques différences entre les clusters selon les données sur lesquelles on les définit. Cependant, ces derniers ayant une forme similaire, on en déduit que la plupart des points sont dans les mêmes clusters, et que seuls certains points sont amenés à en changer. Comme pour le clustering *K-means*, on peut regarder la dispersion dans chacun des clusters créés à l'aide des boxplots suivants.

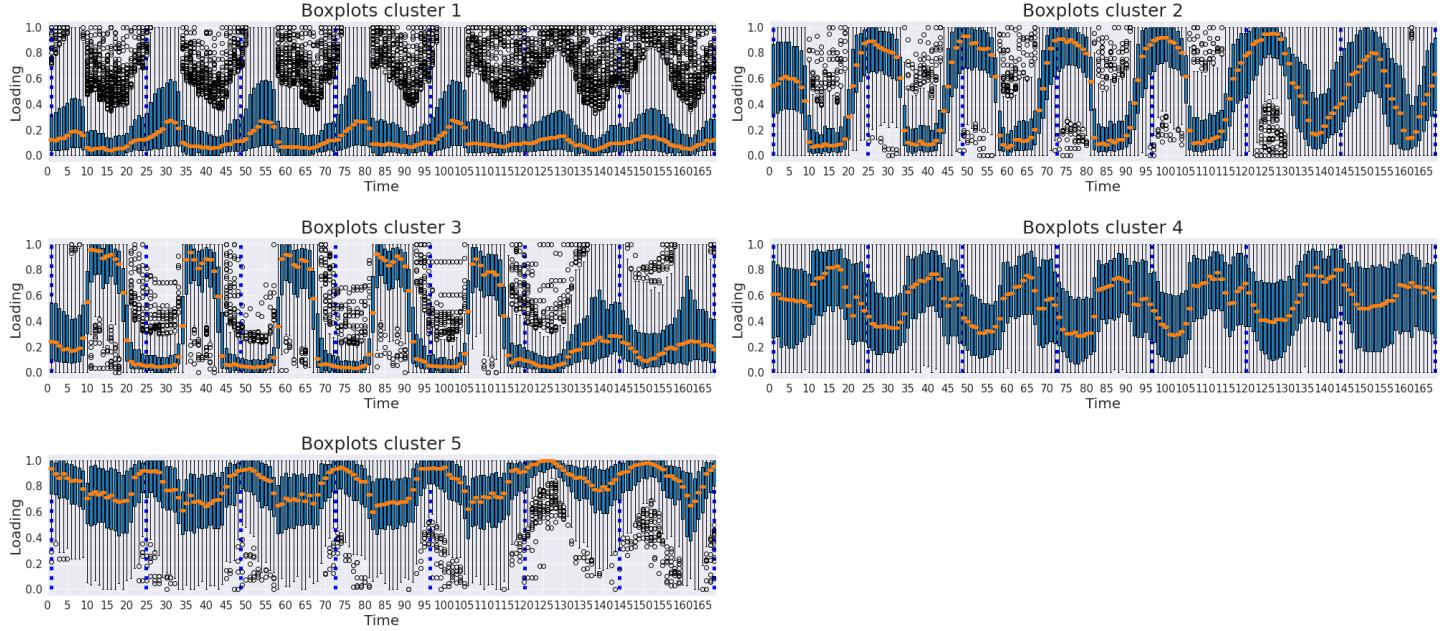


FIGURE 16 – Boxplots représentant la répartition de la capacité des stations *Vélib* à chaque heure de la semaine dans les différents clusters.

Les boxplots obtenus par la classification hiérarchique ascendante sont très similaires à ceux du clustering *K-means*. On en déduit donc qu'ils représentent la même chose. Seuls les numéros de cluster changent, mais cela n'a strictement aucun impact, et on peut immédiatement faire la correspondance entre les clusters Agglomerative Clustering et les clusters *K-means* (1-2, 2-1, 3-3, 4-5 et 5-4).

Finalement, on peut à nouveau remarquer que les stations sur les collines sont principalement dans le cluster 1, comme le montre le graphique suivant :



FIGURE 17 – Classification hiérarchique ascendante selon 5 classes et mise en évidence des stations situées sur des collines.

### 3.3 Gaussian Mixture Model (GMM)

#### 3.3.1 Description de Gaussian Mixture Model (GMM)

Gaussian Mixture Model (GMM) est une méthode de clustering plus "douce" que les méthodes précédentes. Au lieu de déterminer à chaque itération à quel cluster appartient un point, on calcule la vraisemblance d'appartenir à chacun des différents clusters. Pour cela on introduit une famille de distribution  $(f_\theta)_{\theta \in \Theta}$  et des poids  $(\pi_k)_{1 \leq k \leq K}$  tel que  $\sum_{i=1}^K \pi_i = 1$ , pour calculer la probabilité que  $x$  soit dans la classe  $C_k$ .

L'algorithme derrière le fonctionnement de la méthode est le suivant :

1. Initialisation les paramètres ( $\pi^0, \theta^0$ ).
2. Itération jusqu'à convergence des paramètres :
  - (a) Calcul, pour chaque point, de la vraisemblance de provenir de la distribution des  $K$  clusters.
  - (b) Mise à jour des paramètres ( $\pi^t, \theta^t$ ).

En pratique, on suppose que les points suivent une distribution normale  $\mathcal{N}(m, \Sigma)$  dans leurs clusters respectifs. Le paramètre  $\theta$  qui varie à chaque itération est donc le tuple  $(m, \Sigma)$ .

Cependant, si on considère les données brutes pour notre algorithme,  $\Sigma$  sera de taille  $168 \times 168$  car les individus sont dans  $\mathbf{R}^{168}$ . On risque de rencontrer des erreurs de compilation. C'est pour cela que l'on va uniquement considérer les 5 premières composantes de l'ACP.

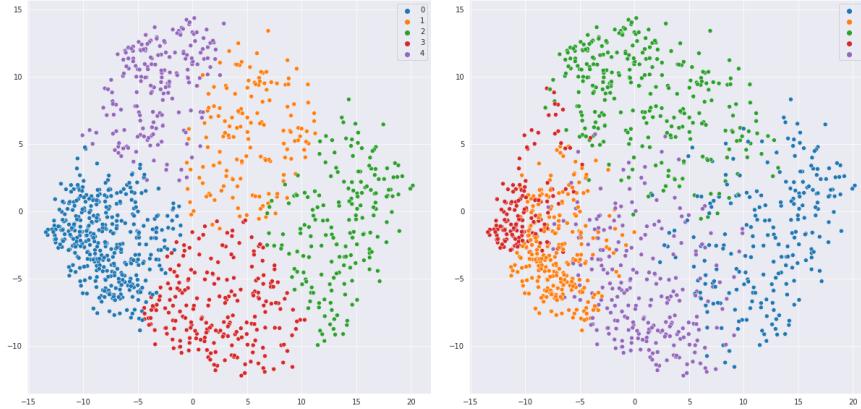


FIGURE 18 – Clustering du jeu de données sur les données brutes (à gauche) et sur les coordonnées de l'ACP (à droite), selon cinq classes.

Nous avons tout de même décidé de lancer l'algorithme sur nos données brutes en Python. Cependant, nous obtenons des résultats peu convaincants. Python doit automatiquement traiter les données afin que l'algorithme converge, ce qui n'est pas le cas de R. On obtient des résultats meilleurs en ne prenant que les 5 premières composantes de l'ACP. Par rapport aux deux précédentes méthodes, on remarque un découpage un peu différent. Le cluster vert sur la figure de droite regroupe deux clusters des précédentes méthodes, tandis qu'on trouve une séparation en deux clusters en bas à gauche, alors que ces variables n'étaient pas séparés lorsque l'on appliquait  $K$ -means et  $CAH$ .

De ce fait, on observe des boxplots assez différents (Figure 19).

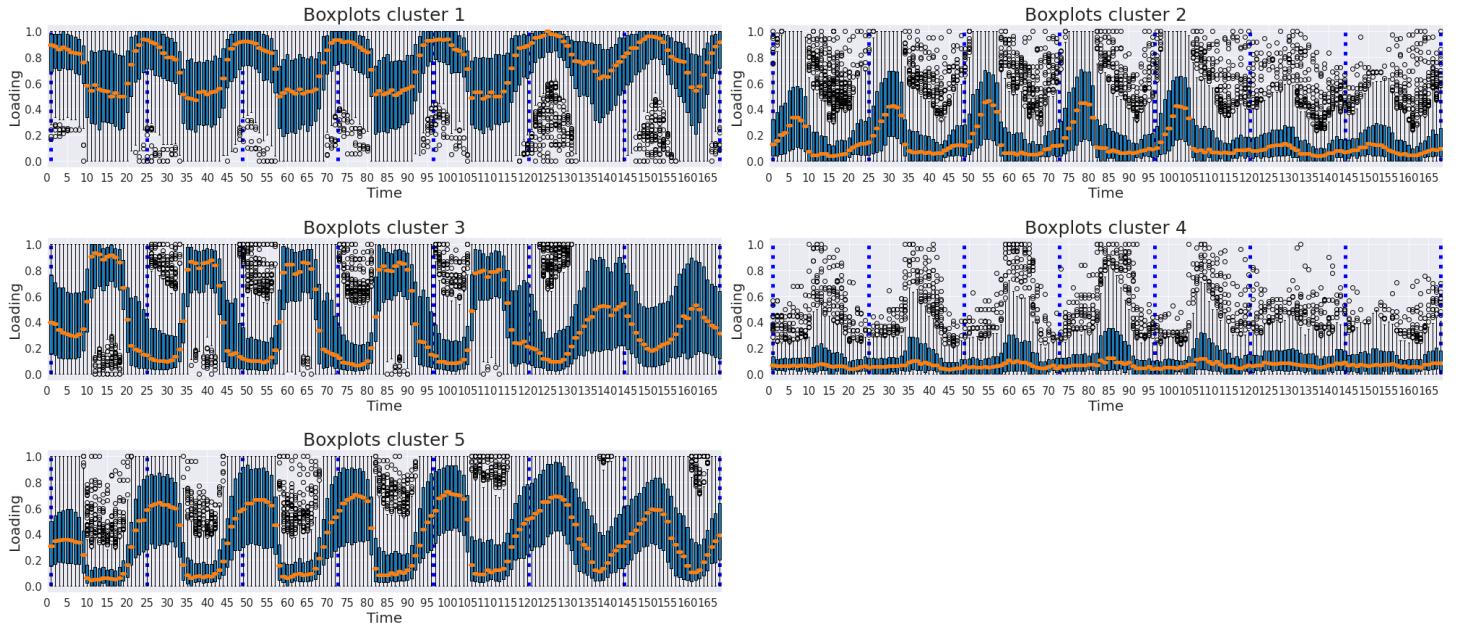


FIGURE 19 – Boxplots représentant la répartition de la capacité des stations de Vélib à chaque heure de la semaine dans les différents clusters (selon les 5 premières composantes de l'ACP)

- Le boxplot du cluster 2, représenté par le cluster orange sur la Figure 18 à droite, correspond principalement aux stations sur les collines. On retrouve effectivement la répartition en sous-figure 4 de la figure 19 ;
- Le troisième cluster correspondrait aux stations situées proches de lieux de travail : elles sont chargées durant la journée, et très peu le week-end.
- Le cluster 5 pourrait représenter les stations proches des lieux d'habitation des personnes qui travaillent, étant peu chargées durant les jours "actifs". Cependant, contrairement aux deux méthodes précédentes, on différencie moins cette catégorie de population. Sur ce même boxplot, on remarque beaucoup d'activités pendant le week-end en soirée : cela pourrait correspondre à ces mêmes personnes qui sortent dans Paris durant le week-end.
- Le cluster 4 représente des stations souvent vides, et principalement sur des collines comme on peut le voir en Figure 19. On peut penser que ces stations ont été mal implantées : une station presque tout le temps vide est inutile et ne répond pas aux besoins des utilisateurs.
- Le premier cluster pourrait correspondre aux populations qui utilisent le moins le vélo, peut-être situées en périphérie de la ville : les stations sont très peu utilisées, et souvent pleines.

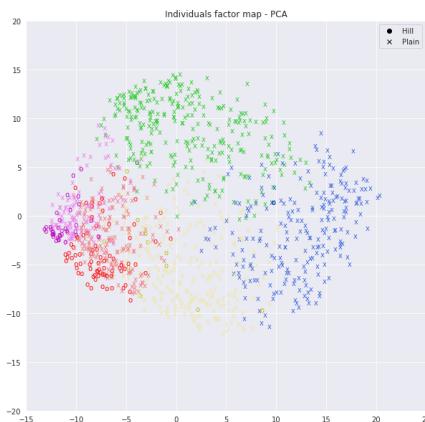


FIGURE 20 – GMM selon 5 classes sur les 5 premières composantes de l'ACP et mise en évidence des stations situés sur des collines

Finalement, on retrouve notre hypothèse précédente sur le cluster orange (en rouge sur la Figure 20), qui représente effectivement principalement les stations sur les collines.

## 4 Analyse des correspondances multiples (ACM)

Nous cherchons à présent à affiner l'analyse factorielle réalisée par le biais de l'ACP en prenant en compte la variable *hill* (*ie* l'appartenance ou non d'une station à une colline). On choisit de mettre en place une analyse des correspondances multiples (ACM) ; pour cela ; il nous faut préalablement transformer toutes les variables quantitatives en variables catégorielles (afin d'avoir des indicatrices à la place de variables quantitatives continues). On découpe ainsi les variables d'occupation en 3 catégories :

- **Catégorie "A"** : taux d'occupation compris dans l'intervalle  $[0, \frac{1}{3}[$ , *ie* peu de Vélib' sont encore disponibles ;
- **Catégorie "B"** : taux d'occupation compris dans l'intervalle  $[\frac{1}{3}, \frac{2}{3}[$  ;
- **Catégorie "C"** : taux d'occupation compris dans l'intervalle  $[\frac{2}{3}, 1]$ , *ie* la station dispose d'un grand nombre de Vélib'.

Avant d'effectuer notre ACM, il est indispensable de réaliser une analyse préliminaire de chaque variable, afin de voir si toutes les classes sont aussi bien représentées ou s'il existe un déséquilibre. En effet, l'ACM étant sensible aux effectifs faibles, il est préférable de regrouper les classes peu représentées le cas échéant. Dans notre cas, les 3 classes sont bien représentées ; il n'est donc pas nécessaire d'en regrouper certaines.

Le principe consiste à conduire une Analyse Factorielle des Correspondances (AFC) à partir d'un tableau disjonctif complet (avec les stations en lignes et les variables d'occupation en colonnes). Ce tableau disjonctif est une présentation de tous les tableaux de contingence des variables (prises deux à deux) et réunis en une seule matrice. De cette manière, on peut déterminer des proximités entre des modalités de variables différentes ou entre individus. On compare ensuite le tableau des observations avec un tableau théorique de totale indépendance. On rappelle ici que la distance considérée pour mesurer l'écart entre ces deux tableaux est celle du **khi-deux**, définie comme :

$$d^2 = \sum_{i=1}^n \frac{(O_i - T_i)^2}{T_i} \quad (1)$$

où  $O_i$  et  $T_i$  sont, respectivement, les valeurs observée et théorique. On décide de mettre en place deux ACM :

- Une première ACM comprenant la variable qualitative binaire *bonus* ;
- Une seconde ACM pour laquelle on enlève *bonus* du tableau disjonctif (on ne traite alors que les variables quantitatives) ; on rajoute ensuite *bonus* comme variable supplémentaire pour l'ACM.

Pour chaque ACM, on extrait ensuite les 5 composantes principales et on effectue un clustering *K-means* sur ces 5 composantes. L'objectif est de déterminer si l'ajout de la variable *bonus* produit des résultats significativement différents pour l'ACM. Les figures 18 et 19 représentent la projection des variables du tableau disjonctif sur les deux principales composantes de l'ACM. Pour la première ACM (Figure 21), le plan engendré par les deux composantes principales explique 36% de l'inertie totale ; pour la seconde (Figure 22), le plan engendré expliqué quasiment 40% de l'inertie totale.

La première ACM permet de mettre en relief deux tendances principales :

- La première composante (horizontale) permet de dissocier les variables en 3 groupes, qui correspondent exactement au niveau de chargement des stations : plus la coordonnée selon la première composante est élevée, plus on se situe dans une station dont le taux d'occupation est élevé (*ie* appartenant à la catégorie "C"). A l'inverse, plus la coordonnée selon la première composante est faible, plus on se situe dans une station dont le taux d'occupation est faible (*ie* de classe "A"). On pourrait même tracer 2 droites sur le plan pour délimiter les 3 niveau d'occupation ;
- La seconde composante (verticale) permet de mettre en relief la différence entre jours de semaine / jours de week-end. En effet, les jours de week-end (*ie* 6 et 7) ont une coordonnée selon cette composante proche de 0, tandis que les jours de semaine (*ie* de 1 à 5) sont plus éloignés de l'axe des abscisses.

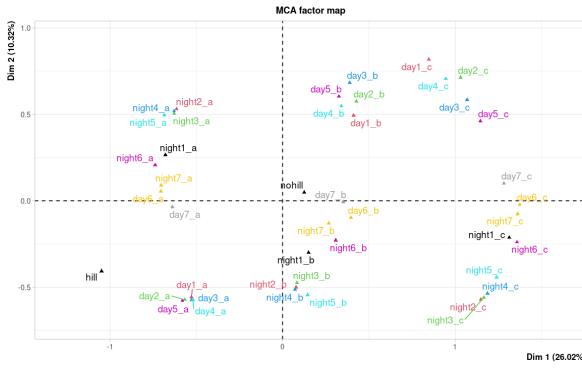


FIGURE 21 –  
ACM intégrant *bonus* dans le tableau disjonctif

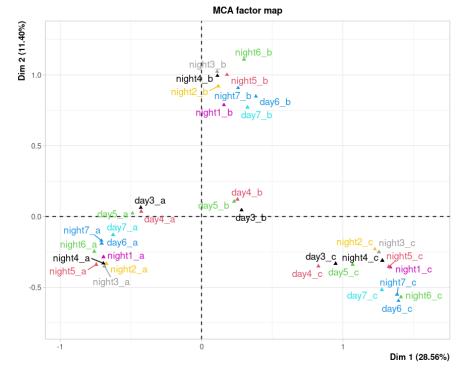


FIGURE 22 –  
ACM sans *bonus* dans le tableau disjonctif

La seconde ACM permet également de dissocier les 3 catégories de chargement selon la première composante. Rajouter *bonus* après avoir construit le tableau disjonctif n'apporte donc rien de particulier par rapport à la précédente ACM. En revanche, cette ACM semble distinguer les jours et les nuits : les jours sont plus proches (en distance du khi-deux) de l'origine que les nuits.

Une fois les 5 composantes principales de l'ACM extraites, on effectue un  $K$ -means dessus. On obtient les 6 clusters suivants (Figures 23 et 24).

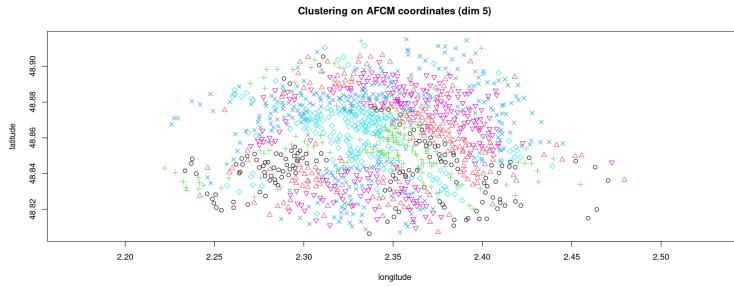


FIGURE 23 –  
 $K$ -means à partir des 5 composantes principales de la  
1ère ACM

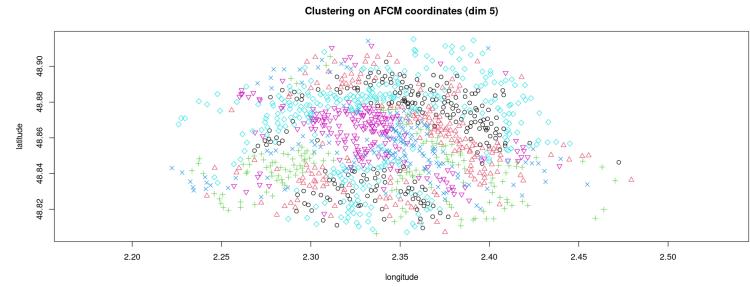


FIGURE 24 –  
 $K$ -means à partir des 5 composantes principales de la  
2ème ACM

Même si les couleurs de clusters des deux  $K$ -means ne sont pas les mêmes, on remarque néanmoins une correspondance :

Cluster $K$ -means à partir de la 1ère ACM	Cluster $K$ -means à partir de la 2nde ACM
Bleu ciel	Rose
Rouge	Rouge
Noir	Vert
Bleu foncé	Bleu ciel
Rose	Noir
Vert	Bleu foncé

On peut donc en conclure que, si le fait d'ajouter la variable binaire *bonus* au tableau disjonctif apporte une information différente au niveau de l'ACM (*ie* distinction semaine / week-end au lieu de jour / nuit), le clustering résultant n'est pas significativement différent de celui obtenu en rajoutant *bonus* comme variable supplémentaire de l'ACM. Ainsi, la variable *bonus* semble avoir une importance minime si l'on souhaite réaliser un clustering sur notre jeu de données.

## Interprétation générale et conclusion

Pour la conclusion, nous nous appuierons sur les boxplots obtenus à partir des clusters de  $K$ -means (page 7). On précise que chaque boxplot correspond à l'un de ceux obtenus avec une autre méthode de clustering. Les clusters n'ont simplement pas le même nom. Pour éviter les biais d'interprétation, notre analyse portera sur les habitudes des Parisiens utilisant le service *Vélib* ; elle ne représente donc pas la totalité des comportements des Parisiens.

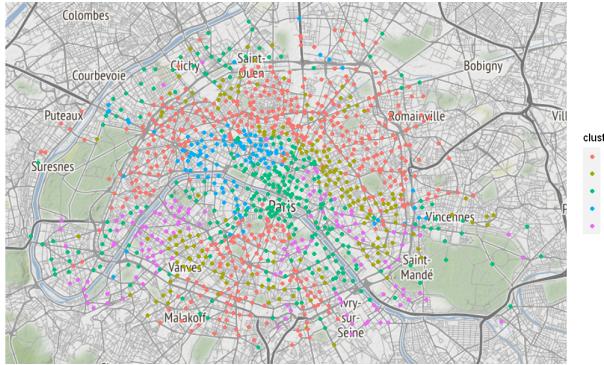


FIGURE 25 – Clustering  $K$ -means sur les données dans le plan de Paris.

**Attention :** Le numéro des clusters sur la carte ne correspond pas au numéro des clusters de la Figure 11. Nous préciserons à chaque fois la couleur correspondante sur la carte.

Afin d'utiliser la variable bonus, nous avons procédé à une ACM. Cependant, les clusters que nous obtenons avec  $K$ -means sur les 5 composantes principales de l'ACM sont relativement similaires aux clusters que nous avions obtenus précédemment. Il y a 2 explications à cela :

- Tout d'abord, il y a 168 variables quantitatives dans notre jeu de données. Lors de l'ACM, la variable bonus ne représente qu'une variable sur 169, et son importance est donc minime.
- Ensuite, nous avons pu remarquer que dans les clusterings précédents, les stations situées sur des collines étaient en grande majorité placées dans le même cluster. La variable bonus n'apporte donc que peu d'information supplémentaire, ce qui explique son effet relativement négligeable.

Cependant, au delà de l'objectif de clustering des stations, cette variable nous permet d'interpréter l'un des clusters (le **cluster 2, cluster rouge**) ; nous pouvons ainsi mieux comprendre le chargement des stations de ce cluster car elles sont essentiellement placées sur des collines. En effet, descendre une colline à vélo n'est pas fatigant, et c'est très rapide. A l'inverse, la grimper en vélo est long et difficile ; on peut donc supposer qu'il y a plus de gens qui prennent un vélo pour descendre (et donc déchargent les stations en haut des collines), que de gens qui prennent un vélo en bas et le déposent en haut (et donc remplissent les stations des collines). Il n'est donc pas étonnant que le chargement en haut des collines soit faible (presque tout le temps sous 0,5, quelle que soit l'heure de la journée).

Nous pouvons également interpréter les autres clusters en fonction de leur profil d'occupation (visible sur la Figure 11) :

- Le **cluster 1 (cluster doré)** correspond à un profil de station située dans une zone d'habitation : les stations de ce cluster se vident à 10h du matin et se remplissent à nouveau vers 20h. La plage de 10h à 20h coïncide quasiment avec les horaires de travail des Parisiens ;
- Le **cluster 3 (cluster bleu)** correspond à un profil de station proche d'un lieu de travail des Parisiens : les stations de ce cluster se remplissent vers 10h du matin (*ie* à l'heure d'arrivée des Parisiens sur leur lieu de travail) et se vident vers 20h (*ie* à l'heure où les travailleurs rentrent à leur domicile). Avec la carte de Paris ci-dessus, on remarque que le lieu de travail des utilisateurs de *Vélib* se trouve vers le centre ville, un lieu où la circulation en voiture est difficile ;
- Le **cluster 4 (cluster violet)** est plutôt beaucoup rempli toute la semaine (taux d'occupation toujours supérieur en moyenne à 0,5). Sur la carte de Paris, ce cluster est plutôt situé au centre de Paris. On peut donc supposer qu'il correspond aux stations localisées à proximité des stations de métro : les Parisiens empruntent les *Vélib*, mais la présence de moyens de transport alternatifs explique que ces stations soient toujours très remplies.

- Le **cluster 5** (*cluster vert*) a un taux de chargement qui oscille, mais ne part jamais dans les extrêmes : elles ont plutôt tendance à se vider en journée et à se remplir la nuit.

Ainsi, avec pour objectif de rendre optimale l'utilisation des *Vélib*, une stratégie qui pourrait permettre de rééquilibrer la répartition des *Vélib* dans les stations parisiennes serait de déplacer une partie des vélos du **cluster 4** (qui est en moyenne toujours bien rempli) pour les mettre dans les stations du **cluster 1** (dont le taux de chargement est toujours assez faible). Cela permettrait ainsi d'homogénéiser le chargement des stations afin d'éviter que certaines soient quasiment pleines, tandis que d'autres manquent de *Vélib* quasiment toute la semaine. En déplaçant certains *Vélib*, on éviterait que certains habitants soient confrontés à des stations vides, et donc dans l'incapacité d'emprunter un vélo. Les vélos pourraient ainsi être utilisés par tous les habitants, à tous les endroits de la ville.