projet Machine Learning 2 - Weather segmentation

Yamina BOUBEKEUR & Amina GHOUL

12/12/2019

Introduction

L'objectif de ce projet est de réaliser une segmentation du territoire français basée sur les séries temporelles de *température* et de *vent* recueillies à $n = 259$ points de grille en utilisant plusieurs méthodes de clustering.

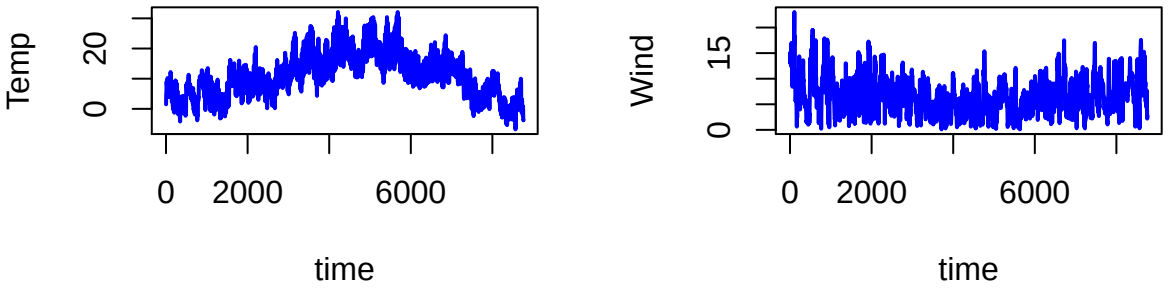
L'ensemble de données *«weatherdata.Rdata»* fournit l'évolution temporelle de la température et du vent pour la grille $n = 259$ pour chaque heure pour une année donnée. Le nombre d'heure ici est de 8760.

- **Temp**: désigne la série temporelle de la Température
- **Win**: représente la série temporelle du vent
- **GPSpos**: contient les positions GPS (longitude et latitude) des points de la grille

I) Préliminaire

Par exemple, la ville de Paris est située à une latitude de $48,51$ et une longitude de $2,20$ et correspond au point $i = 59$ dans la base de données.

On trace les séries temporelles de *température* et du *vent* pour la ville de Paris.



On ajoute les positions GPS de chaque ville, les tableaux de données *Wind* et *Temp*.

Les tableaux de données *Vent* et *Temp* ont tous les deux 259 lignes (qui représentent les 259 villes) et 8762 colonnes (qui représentent les valeurs de la temp/vent pour chaque 8760 heures + 2 colonnes représentant la latitude et la longitude de chaque ville).

On a choisi 3 villes:

- **Paris:** correspond au point $i = 59$ dans le dataset comme nous l'avons mentionné ci-dessus
- **Lyon:** est situé à une latitude de $45,75$ et une longitude de $4,85$ et correspond au point $i = 177$ dans le dataset
- **Perpignan:** est situé à une latitude de $42,70$ et une longitude de $2,90$ et correspond au point $i = 259$ dans le dataset

On représentera ces 3 villes choisies, sur la carte de France.

II) Wind clustering :

II.1) Données brutes

II.1.1) Kmeans :

On utilise la méthode de *kmeans* pour fournir une segmentation en 4 groupes du vent en utilisant la série temporelle brute. Les variables ont les mêmes unités donc on ne normalise pas. Les valeurs sont comprises entre 0 et 30.33

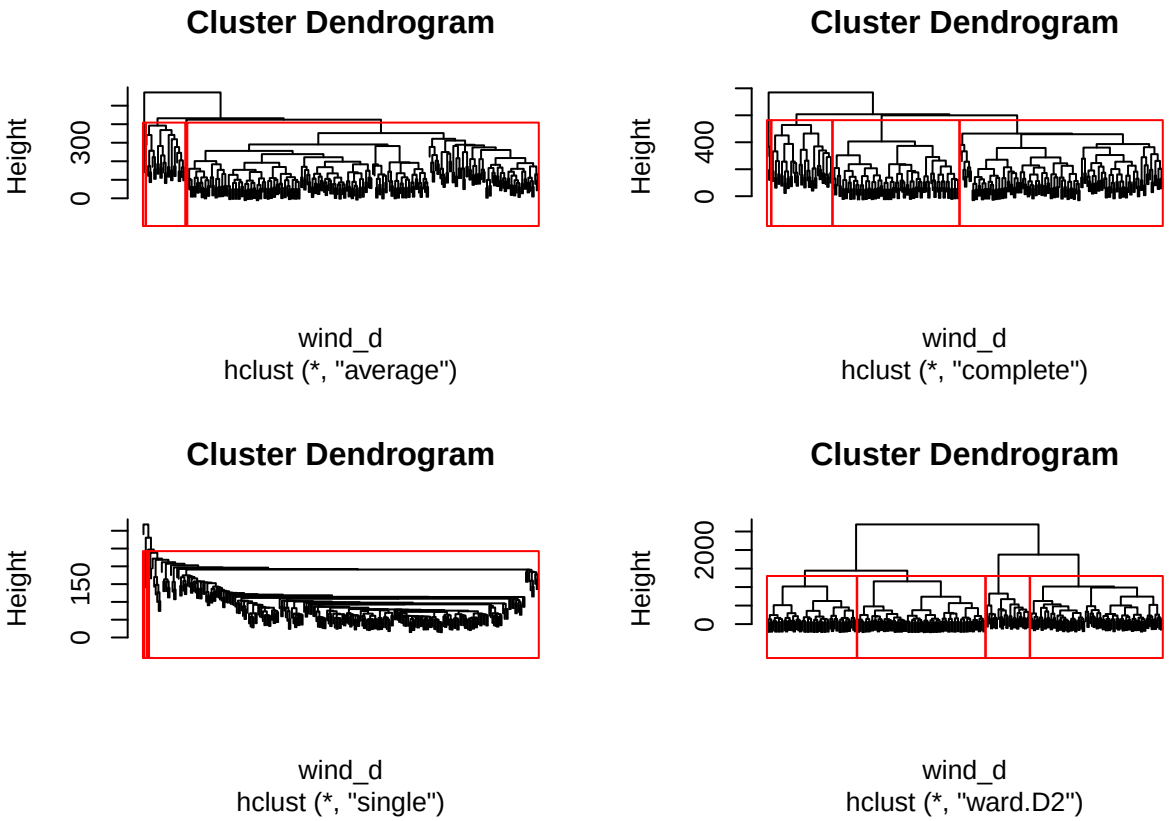
Ici, en utilisant le clustering kmeans, il y a: 82 villes dans le groupe 1, 26 villes dans le groupe 2, 75 villes dans le groupe 3, 76 villes dans le groupe 4. On remarque que les clusters sont plutôt homogènes, mis à part le cluster 2 qui contient un peu moins d'individus que les autres clusters.

II.1.2) Clustering Hierarchique:

On utilise le *clustering hiérarchique* pour fournir une segmentation en 4 groupes du vent en utilisant la série temporelle brute. Étant donné que toutes les valeurs ici sont des valeurs numériques continues, on utilisere la méthode de la distance euclidienne pour le calcul du tableau des distances entre individus.

Ensuite l'algorithme de classification hiérarchique va détecter les 2 groupes les plus proches, puis les agréger pour n'en former qu'un seul. On considère alors différentes stratégies d'agrégation: stratégie **linkage average** qui calcule la distance moyenne entre les clusters avant la fusion, la stratégie **Complete linkage** qui calcule la distance maximale entre les clusters avant la fusion, la stratégie **Single linkage** qui calcule la distance minimale entre les clusters avant la fusion, et la stratégie **wards** qui cherche à minimiser l'inertie intra-classe et à maximiser l'inertie inter-classe afin d'obtenir des classes les plus homogènes possibles

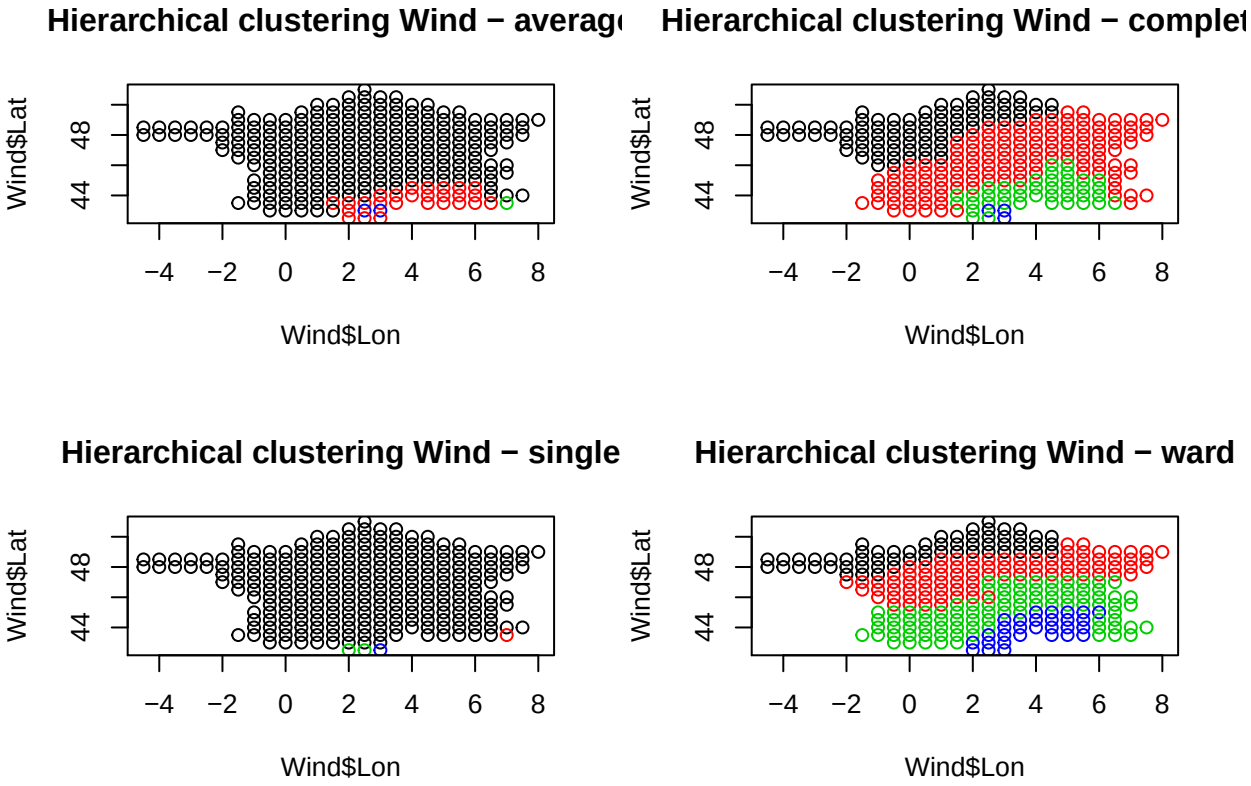
On trace le Dendrogramme pour chaque méthode :



On remarque d'après les graphiques que le clustering n'est pas le même pour chaque méthode. On remarque un gros déséquilibre entre les clusters pour les trois premières méthodes, par exemple pour la méthode single,

on remarque un effet de chaine seule la méthode de Ward a présenté 4 clusters de taille à peu près homogènes.

On représente les clusters des différentes stratégies sur la carte de France :



On remarque la même chose qu'avec les dendogrammes, dans la méthode Ward les 4 clusters sont répartis de manière plus équilibrée comparant aux autres méthodes.

La fonction *agnes{cluster}* peut également être utilisée pour calculer le dendrogramme. Nous pouvons obtenir le coefficient d'agglomération, qui mesure la quantité de structure de clustering trouvé (des valeurs plus proches de 1 suggèrent une structure de clustering forte). D'après l'aide de R, on sait que pour chaque observation i, notons $m(i)$ sa dissimilarité avec le premier cluster avec lequel elle est fusionnée, divisée par la dissimilarité de la fusion à l'étape finale de l'algorithme. Le paramètre ac est la moyenne de tous les $1 - m(i)$.

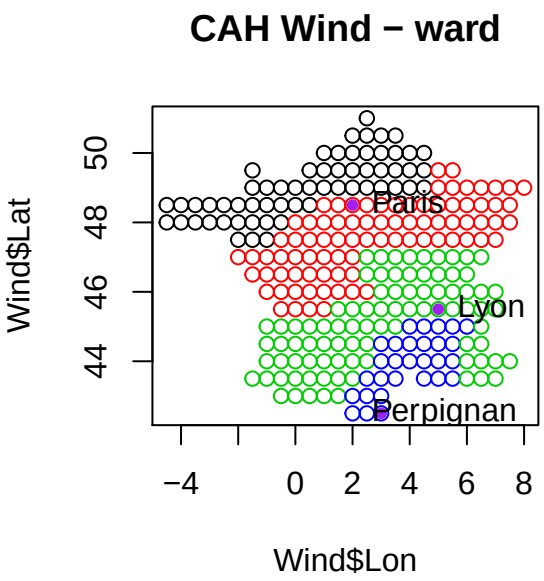
```
##    [,1]                [,2]                [,3]
## m  "average"           "single"            "complete"
## ac "0.83265561465872" "0.719202072126348" "0.873859383314128"
##    [,4]
## m  "ward"
## ac "0.964116984299966"
```

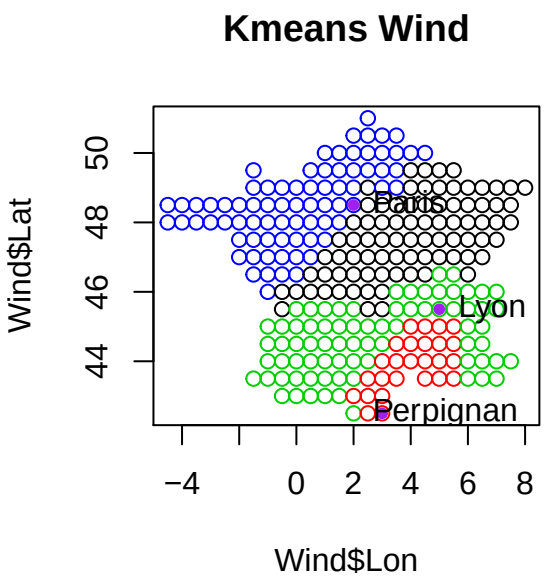
Cela nous affirme que la méthode de Ward identifie la structure de regroupement la plus solide des quatres méthodes évaluées.

II.1.3) CAH VS Kmeans :

- Comparaison des temps d'exécution de deux méthodes de clustering

Pour l'algorithme kmeans, la différence de temps est de 1.54933547973633 alors que pour le clustering hiérarchique elle est de 2.26418852806091. On remarque que le temps d'exécution de la CAH plus grand que celui des Kmeans. K-means est donc moins coûteux en termes de calcul que le clustering hiérarchique et peut être exécuté sur de grands ensembles de données dans un délai raisonnable.





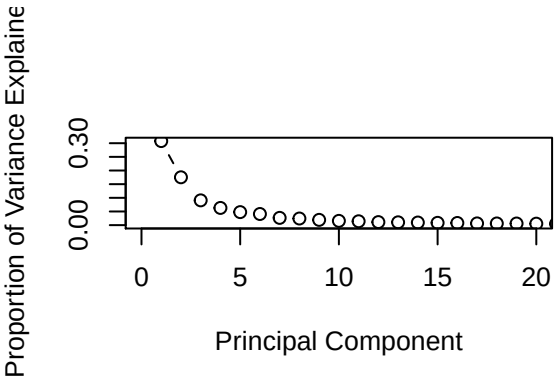
Le territoire français est segmenté presque de la même façon pour les deux méthodes. étant donné que l'algorithme Kmeans est plus rapide que l'algorithme CAH, il est préférable d'utiliser l'algorithme Kmeans.

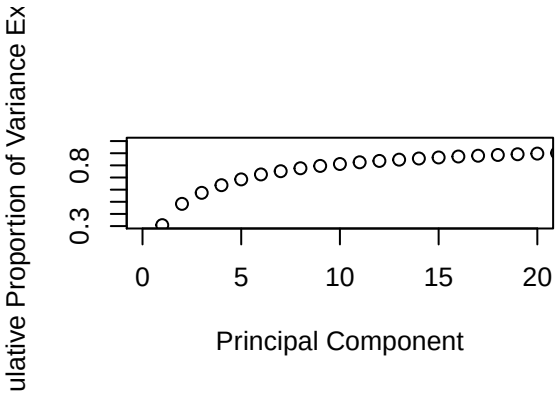
II.2) Feature extracion

ACP

Nous utilisons une Analyse en Composantes Principales (ACP) pour réduire la dimension de la série temporelle pour les données du vent. Et avoir une grande variabilité, c'est avoir un sous-espace qui résume au mieux les données. On centre les données, par contre, on n'a pas besoin de normaliser car les variables ont les mêmes unités.

On représente la proportion de la variance expliquée pour un certain nombre de composante.





Ainsi, on peut voir que les 10 premières composantes expliquent plus de 80% de la variance cumulée, donc on peut choisir de conserver 10 composantes principales. On remarque également que à partir de 10 composantes principales, la variance cumuléen'augmente pas significativement.

II.3) Clustering :

On fait une segmentation en 4 clusters du vent en France, basée sur la représentation de l'ACP en ne gardant que 10 composantes principales à l'aide de Kmeans et du clustering hiérarchique comme on a fait précedemment.

II.3.1) Kmeans :

Dans cette partie, on applique l'algorithme de Kmeans comme précedemment et on affiche les résultats dans la partie comparaison.

II.3.2) CAH:

On applique le clustering hiérarchique (méthode Ward) sur 10 composantes principales.

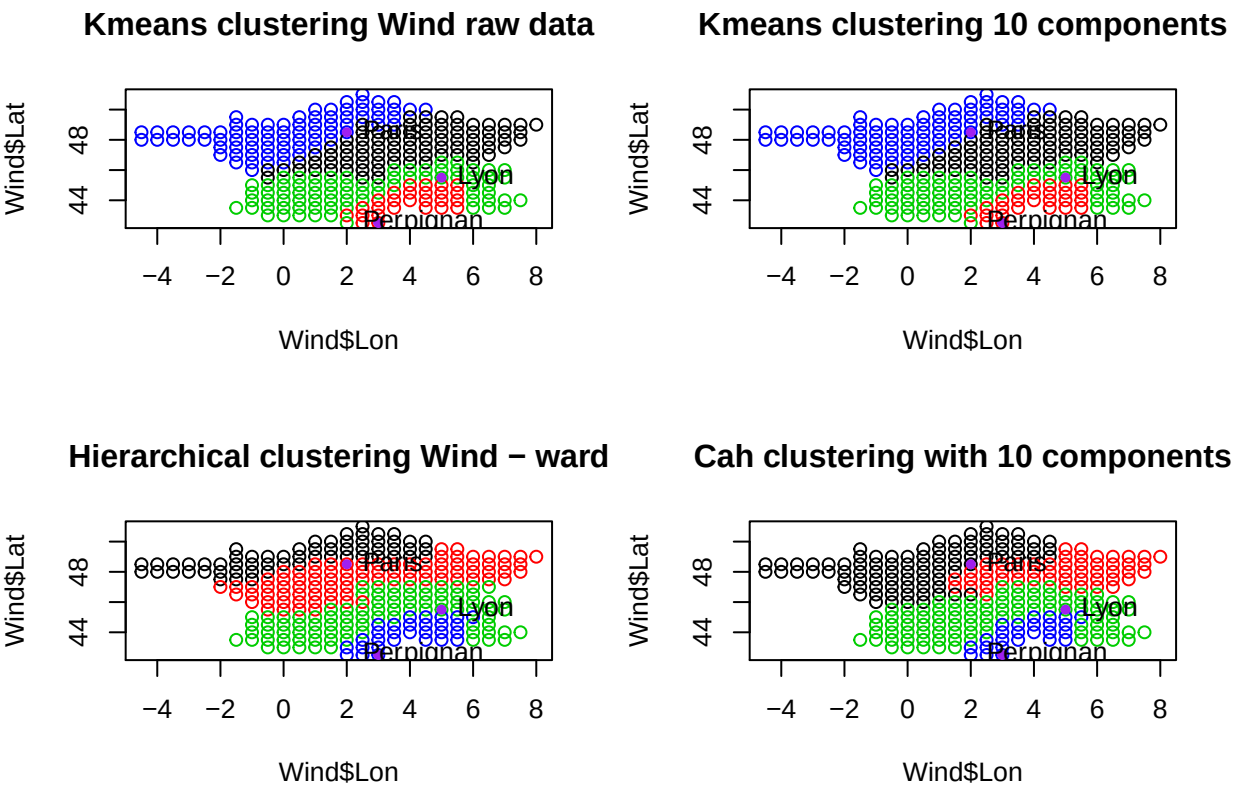
II.3.3) Kmens VS CAH

- Comparaison des temps d'exécution

	Kmeans	CAH
données brutes	1.54933547973633	2.26418852806091
ACP	0.00293159484863281	0.00432038307189941

On remarque que le temps d'exécution pour la méthode des kmeans est moins important après avoir gardé que les 10 composantes principales contrairement à la différence de temps d'exécution dans la méthode CAH.

On représente les clustering des deux algorithmes pour les données brutes et les 10 composantes principales.



On remarque que le partionnement avec la méthode Kmeans est presque exactement le même en gardant que les 10 composantes principales, par contre, le partionnement avec la méthode CAH est différent dans les deux cas, par conséquent les villes n'appartiennent pas à chaque fois au même cluster.

III) Temperature Clustering

III.1) Raw data

III.1.1) Kmeans :

On utilise la méthode de *kmeans* pour fournir une segmentation en 4 groupes du *Temp* en utilisant la série temporelle brute. Les variables ont les mêmes unités donc on ne normalise pas. Les valeurs sont comprises entre -23.1 et 38.3

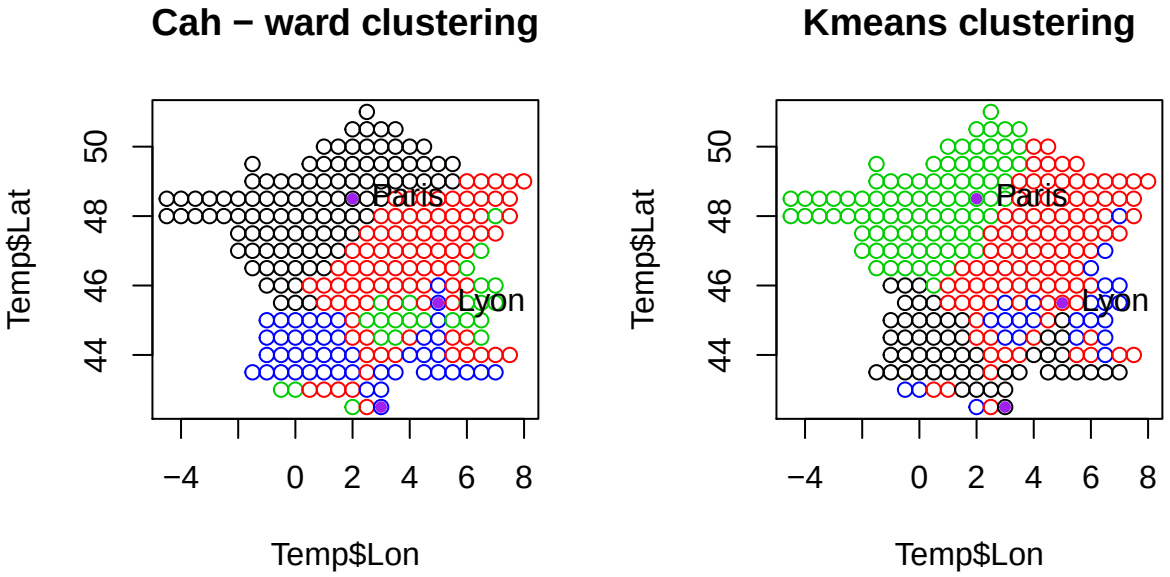
III.1.2) Clustering Hierarchique

Dans cette partie, on va utiliser directement la méthode wards

III.1.3) Kmeans VS CAH

- Comparaison des temps d'exécution

Pour l'algorithme kmeans, la différence de temps est de 1.85798740386963 alors que pour le clustering hiérarchique elle est de 2.24903726577759.



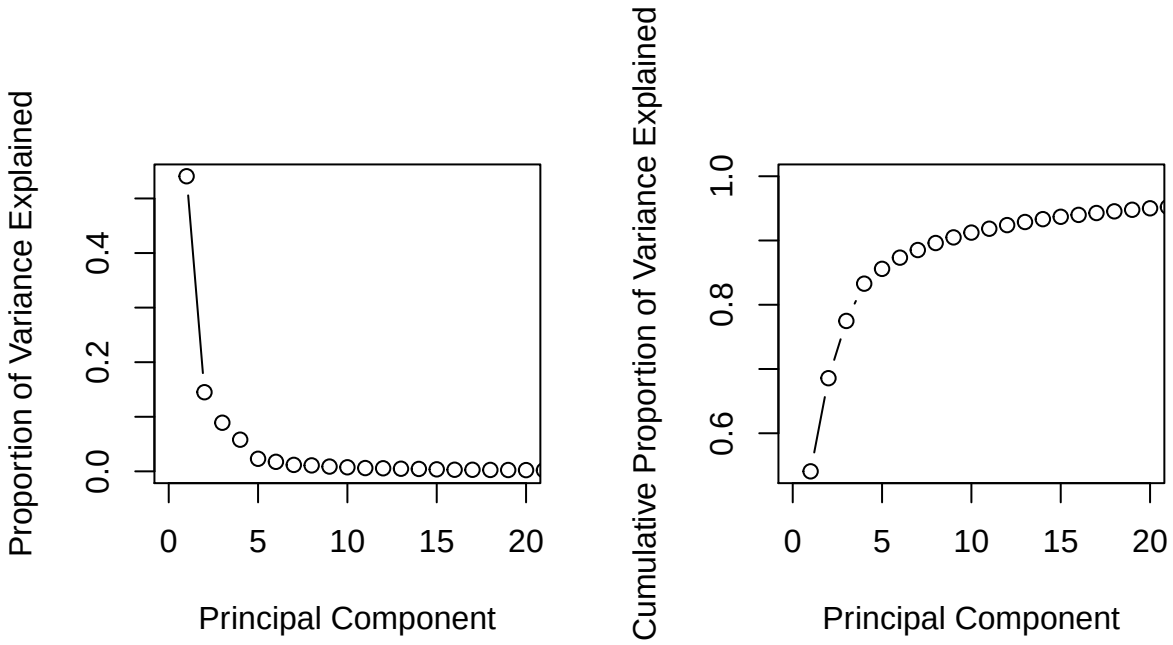
On remarque que les 2 partitionnements sont presque identiques et contrairement au clustering des données Wind, certaines partitions ici ne sont pas très compactes (partie sud).

III.2) Feature extraction

ACP :

On fait une ACP pour réduire les dimensions des données. On centre les données, par contre, on n'a pas besoin de normaliser car les variables ont les mêmes unités.

On représente la proportion de la variance expliquée pour un certain nombre de composante.



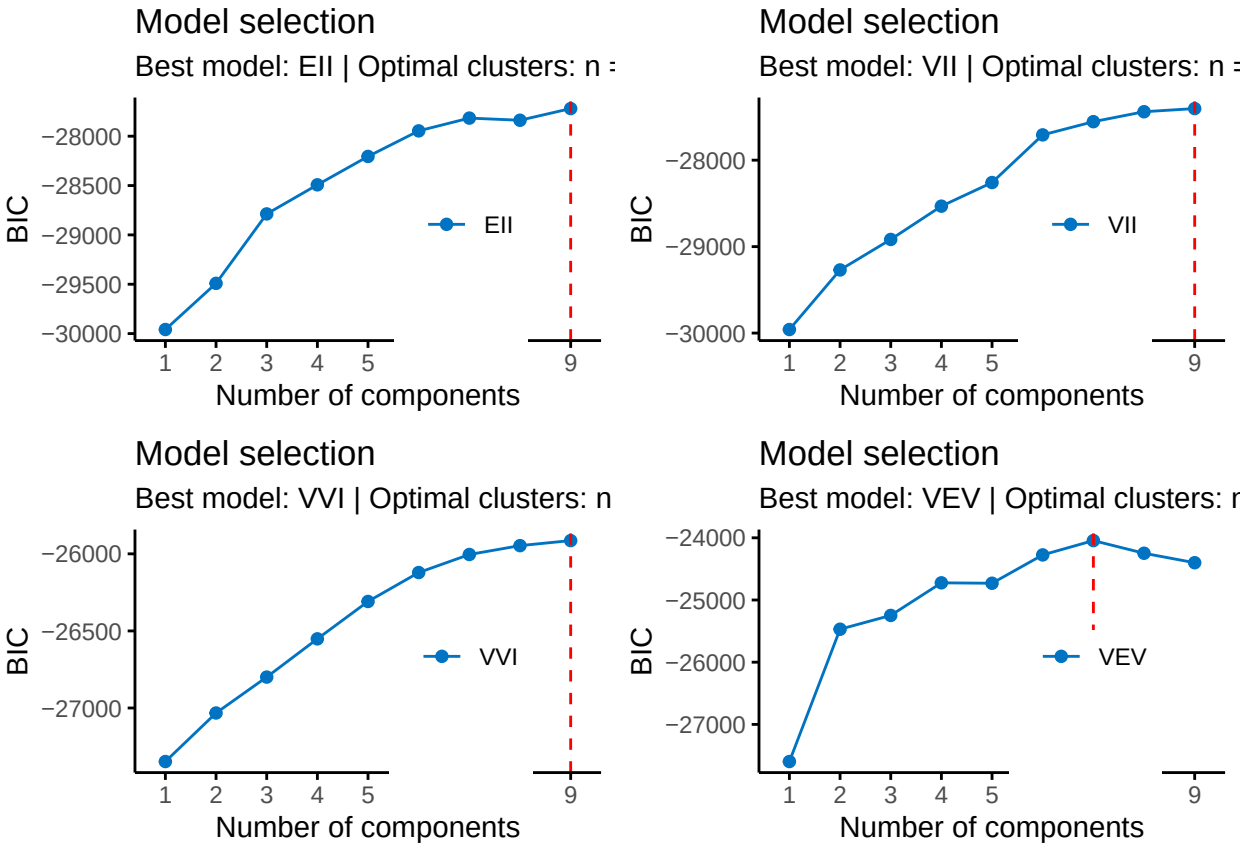
Ici, on peut voir que les 10 premieres composantes expliquent plus de 90% de la variance, donc on peut choisir de conserver 10 composantes principales. On observe par ailleurs que à partir de 10 composantes pricipales, la variance expliquée cumulée n'augmente pas significativement.

III.3) Clustering using model based

On étudie une segmentation de la série temporelle de température, basée sur la représentation ACP, en ne gardant seulement que 10 composantes principales utilisant la méthode de clustering basée sur un modele.

On va regrouper les données *Temp* en utilisant la fonction *mclust* qui utilise BIC (Critère d'Information Bayésien) comme critère de sélection de modèle de cluster. Dans un premier temps, on va essayer différents modèles: modèle sphérique, volume égal: **EII**, modèle sphérique, volume variable: **VII**, modèle diagonale, volume et forme variables: **VVI**, et modèle ellipsoïdale, volume variable et forme égale: **VEV**.

On trace la courbe du BIC pour chaque modèle. Pour trouver le meilleur modèle et le nombre de cluster optimal il faut maximiser le BIC qui vaut ici : $BIC = 2\log(L) - k\log(N)$. avec L a vraisemblance du modèle estimée, N le nombre d'observations dans l'échantillon et k le nombre de paramètres libres du modèle



On affiche le nombre de clusters et la valeur du BIC pour chaque modèle:

```
##               EII      VII    VVI       VEV
## nbclust      9.00      9.0      9      7.00
## Bic     -27720.05 -27403.2 -25914 -24044.87
```

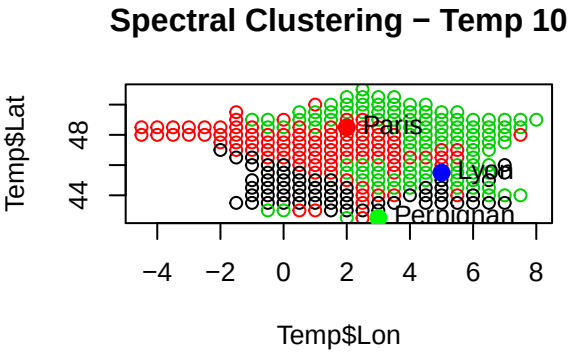
On peut voir que la valeur la plus élevée de BIC est -2.4044874×10^4, donc le meilleur modèle est "VEV" avec 7 groupes.

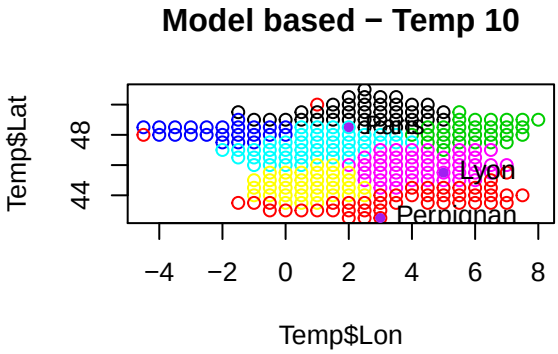
4) Clustering using spectral clustering

On utilise la méthode de spectral clustering pour regrouper les observations de température en utilisant les 10 principales composantes de l'ACP. On trace la courbe du nombre de clusters en fonction de l'inertie intra-class qu'on divise par le nombre de cluster.

D'après la règle du coude, on remarque qu'il n'y pas d'amélioration significative à partir de 3 clusters. Donc on peut considérer que le nombre de clusters optimal vaut 3.

5) Représentation des clusterings sur la carte





Pour mclust : on remarque que les partitions obtenues sont bien compactes, mise à part la zone à l'extrême sud, on observe deux points aberrants. Pour le spectral clustering : on remarque que les partitions trouvées ne sont pas du tout homogènes, ni compactes.

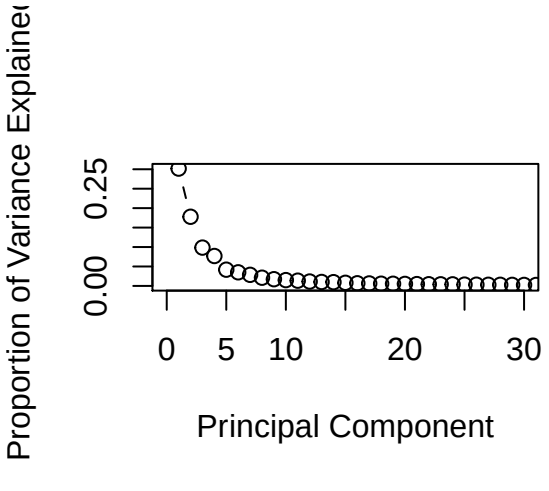
IV) Temperature and Wind Clustering

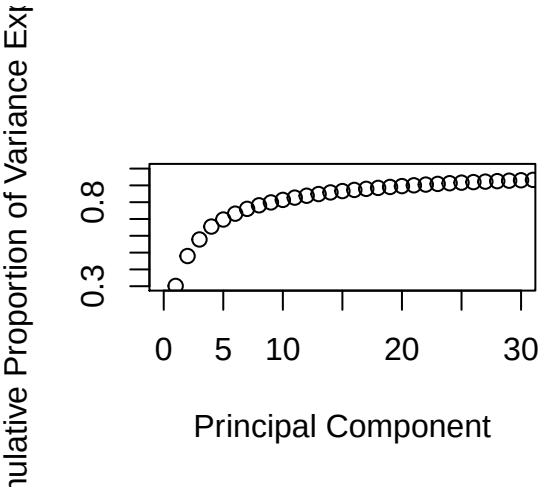
On utilise la fonction cbind() pour fusionner les données température et les données du vent. Par ailleurs, comme *Temp* et *Wind* ont des unités différentes, on doit les mettre à l'échelle en les normalisant.

Notre nouveau jeu de données contient 259 villes et 17520 variables quantitatives et 2 colonnes en plus qui représentent la latitude et longitude de chaque ville.

IV.1) ACP

On fait une ACP pour réduire la dimension de ces données.

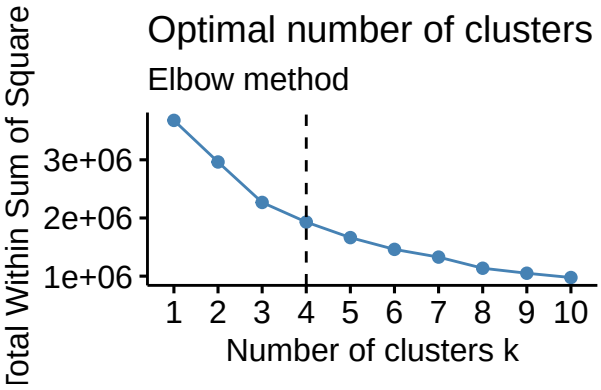




Ici, on peut voir que les 10 premieres composantes expliquent environ 80% de la variance, et à partir de 10 composantes pricipales, la variance expliquée cumulée n'augmente pas significativement. Donc on peut choisir de conserver 10 composantes principales.

IV.2) Kmeans

On utilise l'algorithme des kmeans comme précédemment.



Avec la méthode du coude, on choisit k=4 car l'adjonction d'un groupe supplémentaire n'augmente pas «significativement» la part d'inertie expliquée par la partition.

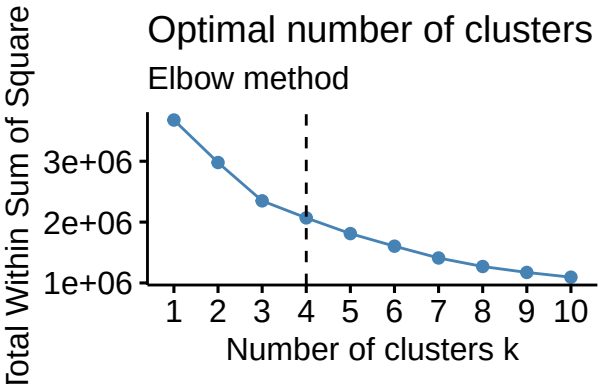
IV.3) Model based

On regroupe les données en utilisant la fonction *mclust*.

Pour ces données, la mise en cluster basée sur un modèle a sélectionné un modèle avec 9 composantes. Le meilleur modèle choisi est le "VEV", modèle ellipsoïdale, volume variable et forme égale.

IV.4) Classification hiérarchique

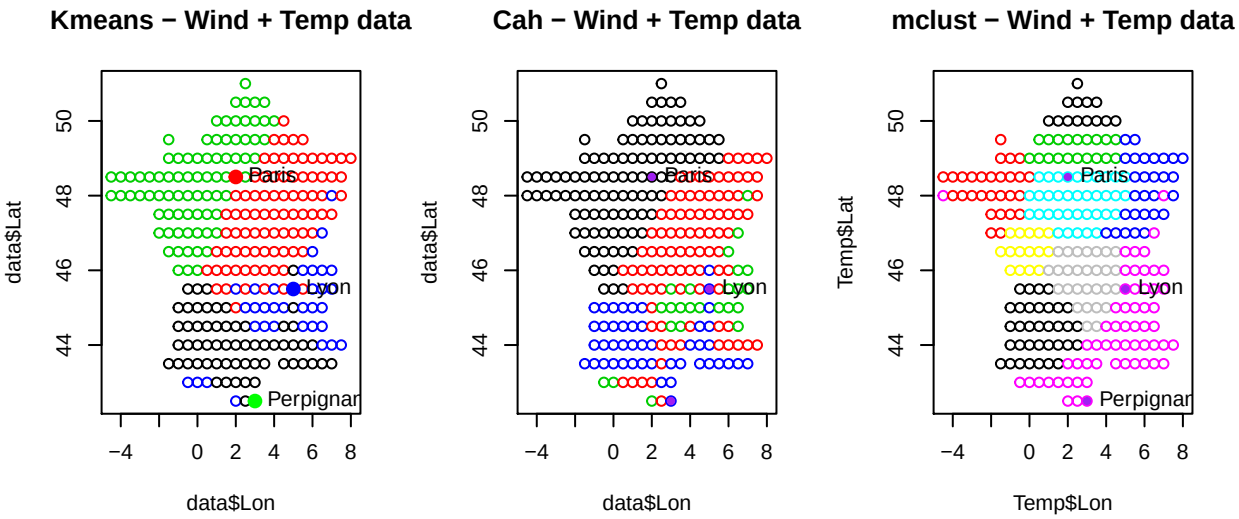
Enfin, on fait un classification hiérarchique pour regrouper nos données.



Avec la méthode du coude, on choisit k=4 car l'adjonction d'un groupe supplémentaire n'augmente pas «significativement» la part d'inertie expliquée par la partition.

IV.5) Comparaison des 3 algorithmes

- On représente la segmentation du territoire français selon les trois algorithmes utilisés.



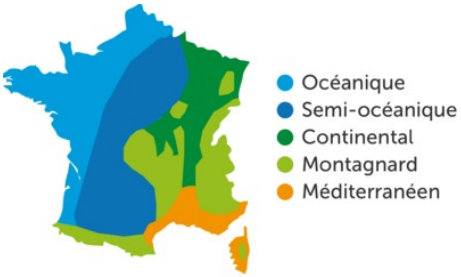
- Temps d'exécution des 3 algorithmes

```
## [1] "temps kmeans : 0.00833892822265625"
```

```
## [1] "temps CAH : 0.00534176826477051"
```

```
## [1] "temps mclust : 0.155179500579834"
```

On remarque que l'algorithme "Mclust" met plus de temps d'exécution que les autres. On observe que les villes n'appartiennent pas au même cluster (mis à part Lyon et Perpignan dans le mclust) ce qui signifie qu'elle n'ont pas le même profil météorologique. on observe aussi que les clusters sont globalement compacts pour les 3 algorithmes et que les Kmeans et le CAH se rapproche de la répartion des 5 climats de la France comme présenté ci-dessous.



On remarque que le clustering CAH c'est qui ressemble plus à la carte du climat de France.

Conclusion :

Selon les algorithmes utilisés, on obtient des segmentations différentes. Chaque cluster représente un profil météorologique particulier et ce genre de segmentation peut être utile pour par exemple prévoir un projet de production d'énergie.