MÉTHODE DE L'IA 1 : PROJET D'APPRENTISSAGE NON SUPERVISÉ

COMPTE RENDU : APPRENTISSAGE NON SUPERVISÉ POUR LA COMPRÉHENSION ET LA CLASSIFICATION D'IMAGES DE FEUILLES D'ARBRES

9 janvier 2023

BARILLER Halvard
PIERRON Alex

M1 MIA

Table des matières

1 Introduction

La reconnaissance d'image est un domaine qui connaît de plus en plus d'application pratique au fur et à mesure que les connaissances et outils mathématiques et informatiques se développent. Parmi toutes les applications possibles, la capacité à identifier des feuilles d'arbres est une tâche ardue car soumise aux variations naturelles. En effet, même si deux feuilles d'une même espèce partage des caractéristiques similaires, elles ne sont pas strictement identiques.
Afin de concevoir une base de données d'image annotée automatiquement de différents types de feuilles d'arbres, il est alors intéressant de se tourner vers des méthodes d'apprentissage non supervisé.
Dans cette optique, nous allons étudier un data set qui a été notamment utilisé dans l'article *Evaluation of Features for Leaf Discrimination* publié par Pedro F.B. Silva, André R.S. Marcal et Rubim M. Almeida da Silva.

Nous nous proposons d'utiliser les mêmes données sans avoir recours aux labels des différents échantillons recueillis si ce n'est à des fins de vérifications de nos résultats de nos méthodes non-supervisées.
Pour cela nous aurons besoin dans un premier temps de traiter nos données afin d'obtenir le même jeux de données que dans l'article cité plus haut. Nous pourrons ensuite nous intéresser à la distribution de ces données et faire un traitement par PCA de ces données afin d'étudier la structure interne des ces dernières. Nous terminerons par l'étude de plusieurs approches de clusterings sur ces mêmes données ainsi que sur les variables.

2 Traitement initial des données

Après avoir acquis le jeu de données "leaf.csv", il nous faut trier les données présentes afin d'avoir le même jeu de données que dans l'article étudié.
Pour ce faire, on sait que dans l'article, uniquement certaines variables sont conservées sur les 16 attributs que compte le jeu de données initiales. Les variables conservées sont :

1. *Eccentricity* : prend des valeurs entre 0 et 1
2. *Aspect Ratio* : prend des valeurs positives
3. *Elongation* : prend des valeurs entre 0 et 1
4. *Solidity* : prend des valeurs positives
5. *Stochastic convexity* : prend des valeurs entre 0 et 1
6. *Isoperimetric factor* : prend des valeurs entre 0 et 1
7. *Maximal Indentation Depth* : prend des valeurs positives
8. *Lobedness* : prend des valeurs positives

De plus, nous conservons les première et deuxième colonnes de notre jeu de données initiales (*Class* et *Specimen Number*), ces dernières contenant les labels de chaque feuilles (respectivement classe et numéro au sein de la classe), ce qui nous permettra plus tard de faire des mesures de précision sur nos résultats lors du clustering.
On rappelle que nous n'utiliserons pas ces deux premières colonnes lors des différents traitements des données (cadre non supervisé).
Une fois la sélection des *features* effectuées, il faut à présent retirer les individus, i.e les classes, qui sont absentes de l'article. On retire toutes les lignes de notre data frame qui ont une classe supérieure ou égale à 16 puisque notre article référence analyse uniquement les 15 premières classes de ce jeu de données.

On obtient finalement un jeu de données avec 171 observations et 8 *features* comme dans l'article.

3 PCA

Le jeu de données étant désormais conforme aux attentes, nous commençons notre étude par une analyse en composantes principales. Celle-ci va permettre de considérer une possible réduction de dimensions afin d'obtenir une représentation visuelle réaliste de notre jeu de données.

3.1 Analyse des données

Nous commençons par nous familiariser avec les données retenues. Pour cela, nous procédons tout d'abord à quelques représentations graphiques.

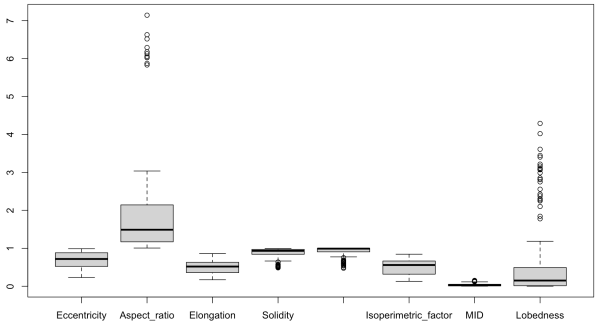


FIGURE 1 – Boxplot des covariables

Nous observons sur les boxplots ci-dessus que les huit covariables considérées sont toutes réparties sensiblement à la même échelle.
Nous observons aussi que les covariables *Aspect ratio* et *Lobedness* possèdent plusieurs outliers. Ces derniers ne déforment pas la distribution, mais il pourrait être intéressant de filtrer les feuilles correspondants à ces caractéristiques et les utiliser comme des individus illustratifs si le résultat de l'ACP ne nous convenait pas.

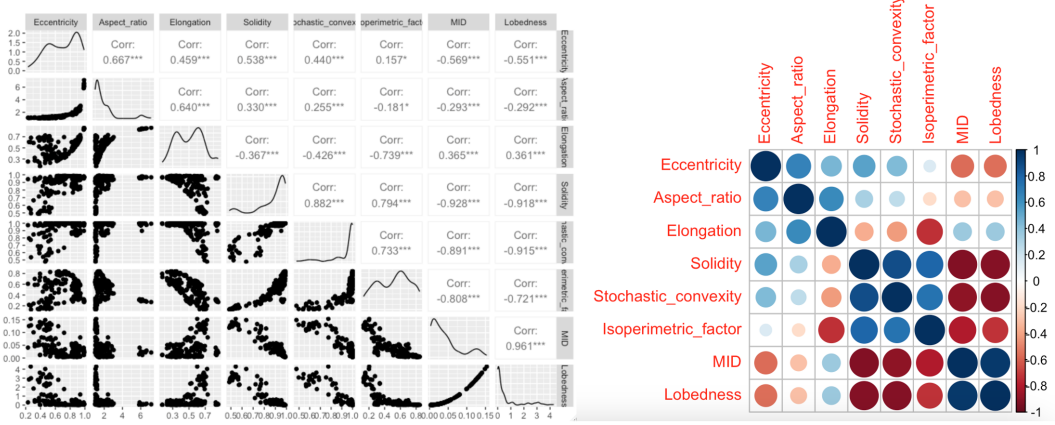


FIGURE 2 – Pairs et covariances des covariables

De manière générale, on observe une forte dépendance des covariables les unes par rapport aux autres : seule *Aspect ratio* semble moins être influencée par les autres paramètres.
Parmi ces covariances, on distingue notamment un bloc très marqué, aussi bien positivement que négativement, composé de *Solidity, Stochastic convexity, Isoperimetric factor, MID* et *Lobedness*. On s'attend

donc à retrouver ces covariables fortement portées par une même composante principale.

On procède également à une vérification de la bonne répartition des 15 classes étudiées dans le jeu de données (cf. code).

3.2 Préparation à l'ACP

Parmi les *features* conservés, on distingue deux catégories :

- les variables actives : *Eccentricity*, *Aspect Ratio*, *Elongation*, *Solidity*, *Stochastic convexity*, *Isoperimetric factor*, *Maximal Indentation Depth* et *Lobedness* qui vont servir à déterminer les composantes principales (variables quantitatives).
- les variables illustratives : *Class* et *Specimen number* qui sont deux variables qualitatives n'ayant pas d'incidence sur la détermination des composantes principales, mais qui pourront être utilisées pour distinguer les feuilles. (En pratique, on ne se servira que de la variable *Class* pour distinguer les feuilles).

En ce qui concerne les individus, le jeu de données étant réduit, nous allons effectuer l'ACP en considérant tous les individus comme actifs. Si les résultats ne sont pas satisfaisants, nous considérerons quelques individus *outliers* comme illustratifs (cf. ci-dessus).

Enfin, nous centrons toutes les covariables actives avant de commencer l'ACP, puis nous les normons afin de les considérer à des échelles semblables et sans unité scientifique.

3.3 Analyse en composantes principales

Une fois les transformations précisées en 3.2 effectuées, nous procédons à l'analyse en composantes principales à l'aide de la libraire *FactoMineR*.
On commence par analyser l'éboulis des valeurs propres.

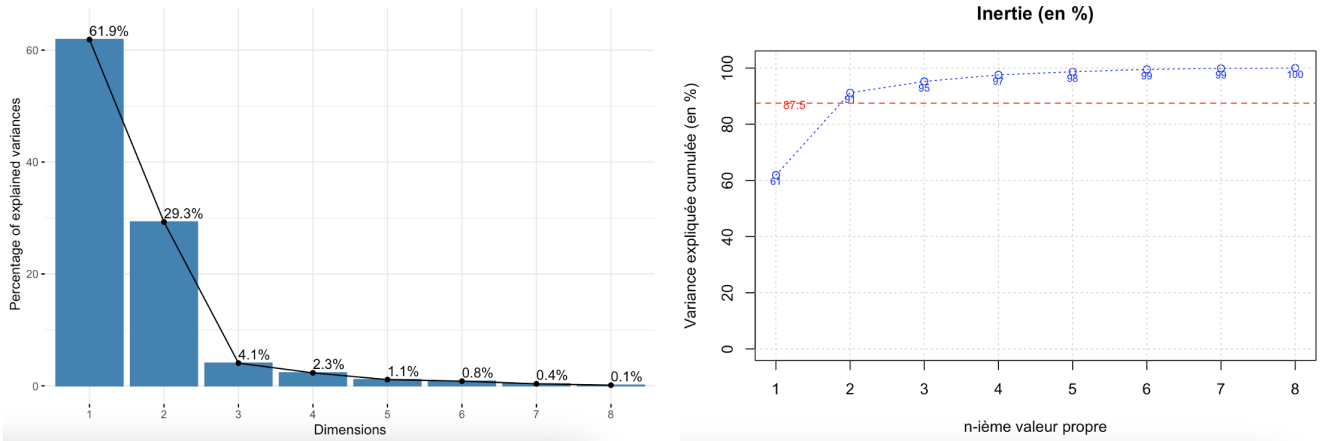


FIGURE 3 – Éboulis des valeurs propres et variance expliquée cumulée (en %)

On voit clairement sur les figures que deux composantes principales suffiront dans notre étude : les deux premières composantes obtenues expliquent plus de 91% de la variance.

Nous poursuivons avec la représentation du nuage d'individus ainsi que la représentation des variables.

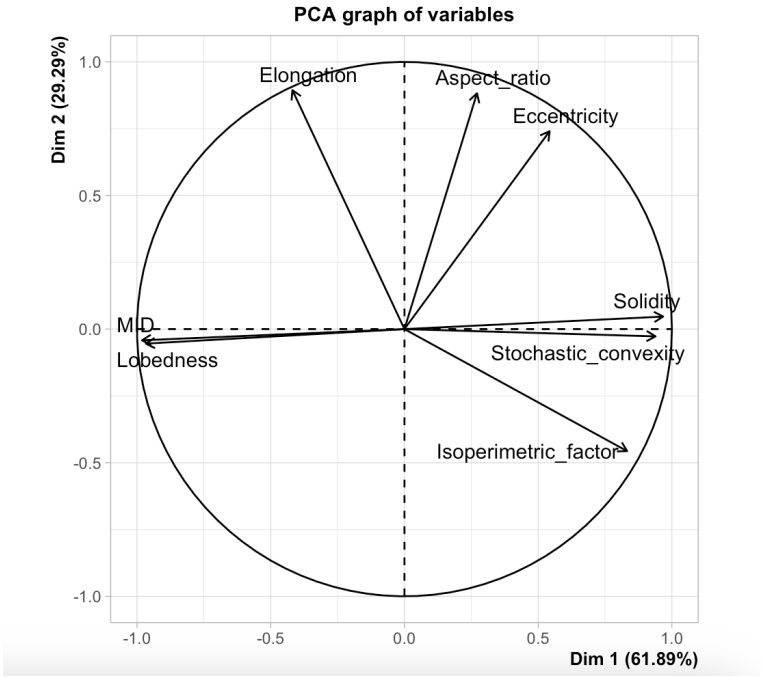


FIGURE 4 – Cercle des corrélations

Comme attendu après l'analyse des covariances, nous retrouvons bien un premier axe sur lequel les covariables *Solidity, Stochastic convexity, Isoperimetric factor, MID* et *Lobedness* sont très bien représentées. Comme mentionné dans l'article, on peut interpréter cet axe comme une moyenne des contributions des covariables relatives à la forme de la feuille. Ces paramètres sont séparés en deux groupes en fonction des propriétés géométriques qu'ils représentent, avec un impact positif pour *eccentricity, aspect ratio, solidity, stochastic convexity, isoperimetric factor*, et un impact négatif pour *elongation, maximal indentation depth, lobedness*.

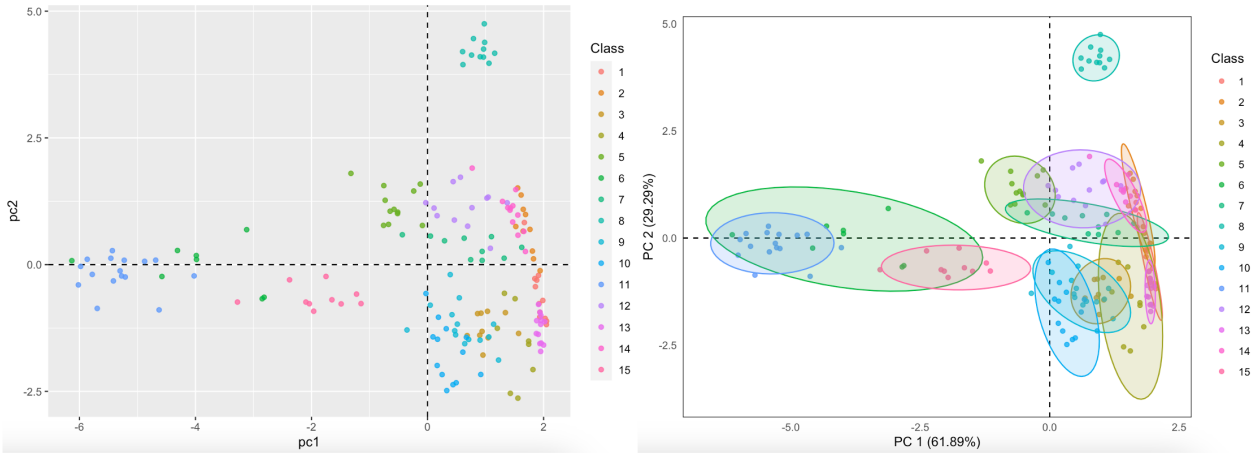


FIGURE 5 – Nuage des individus avec régions de confiance

On remarque sur la représentation du nuage d'individus qu'exception faite d'un ou deux groupes un peu large, on semble avoir une représentation assez distincte et sans trop de chevauchements entre les classes de feuilles.

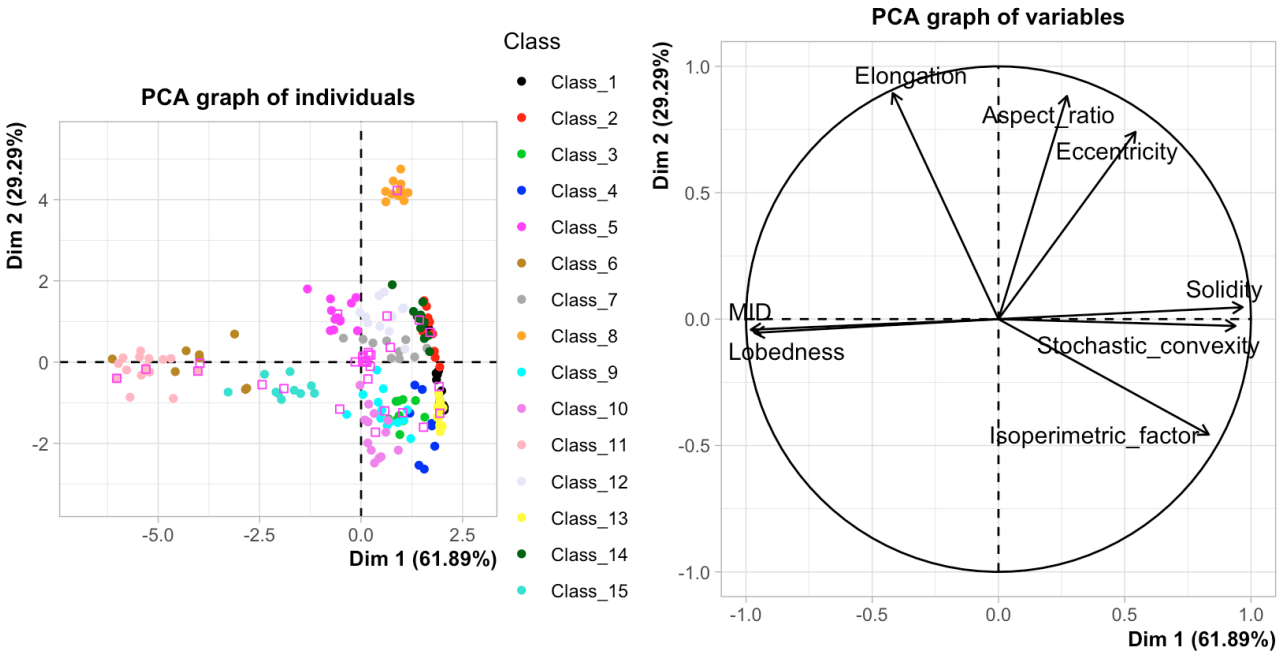


FIGURE 6 – Comparaison simultanée individus et plans principaux

On observe un amas de classes autour de l'origine de l'axe.
Certains groupes de feuilles semblent quand même se distinguer, comme la classe 8 qui présente des caractéristiques d'Eccentricity et d'Aspect ratio plus importants que les autres par exemple.

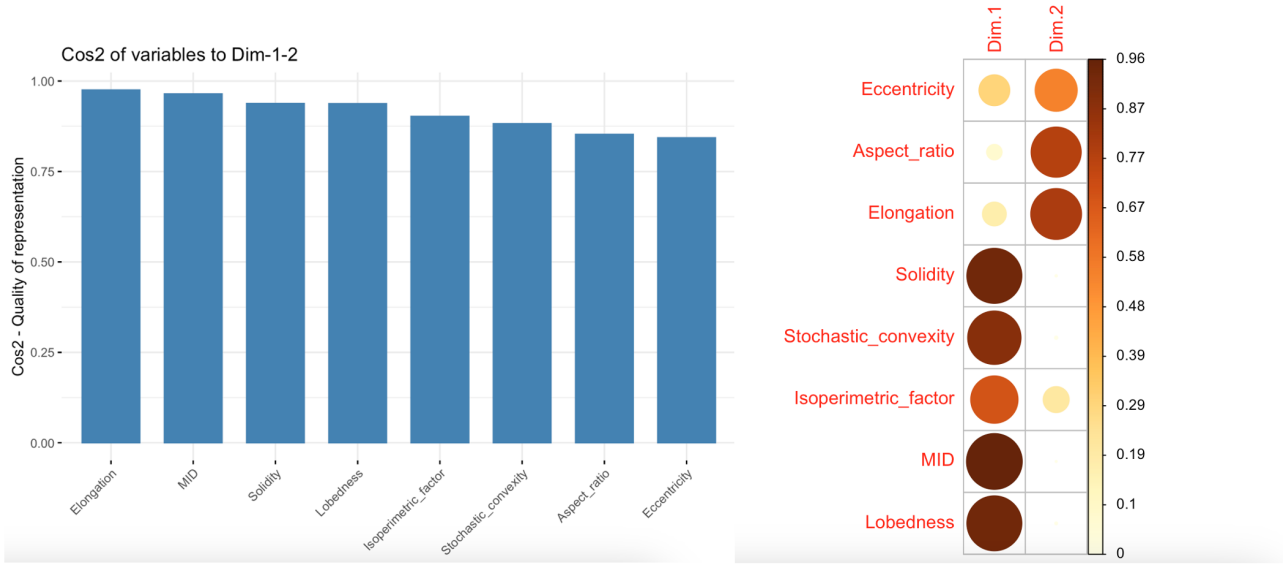


FIGURE 7 – Qualité de la représentation des variables

4 Clustering

Après avoir réduit la dimension du problème dans la partie précédente, on souhaite à présent procéder à un clustering de nos données. On utilisera majoritairement dans cette partie des jeux de données normalisés. Nous préciserons lorsque les données auront un format différent.
On utilise ici les données normalisées sans passer par un PCA car il n'y a pas besoin de réduire la dimensionnalité des données au regard de la faible taille du data frame originale et du nombre limité de features et d'individus. Cela nous permet de garder un maximum d'information.

4.1 Clustering par K-Means

Nous allons réaliser un premier clustering non supervisé par K-Means. Pour cela nous allons tester plusieurs valeurs de K, nous avons choisi de prendre $K \in \{3, 6, 9, 12, 15, 18, 21\}$ et d'effectuer un clustering par K-Means pour chacune de ces valeurs.

En utilisant la fonction *AdjustedRandIndex* de R, nous pouvons accéder à la précision de chacun de ces clusterings par rapport à nos données initiales afin de comparer, on obtient le tableau suivant :

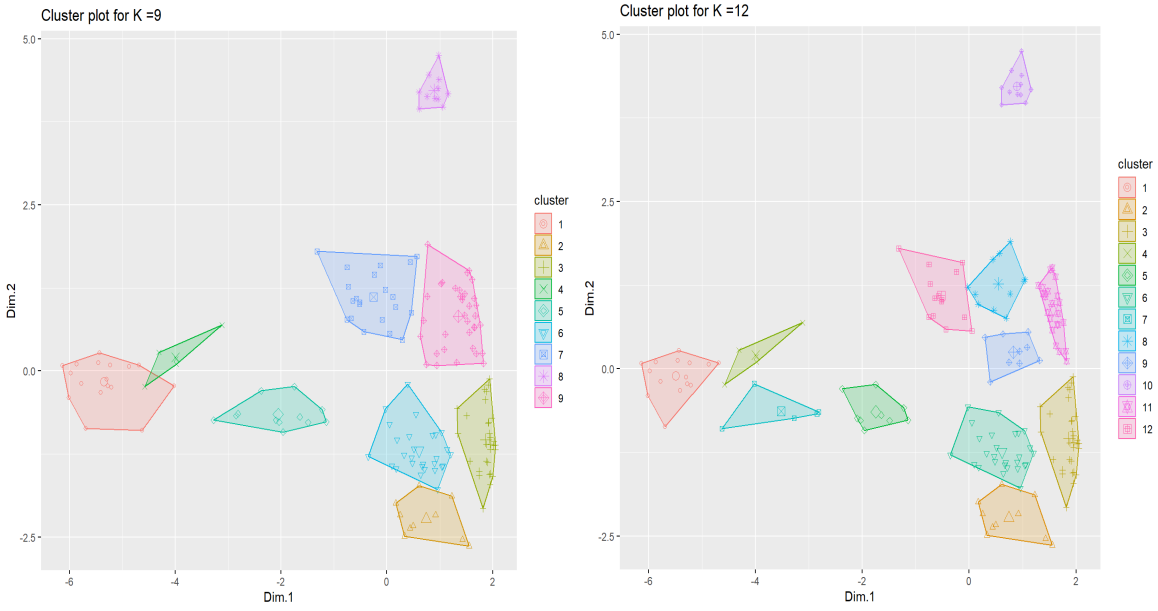
	Nombre de cluster	precision
1	3	0.1766334
2	6	0.4035743
3	9	0.4613611
4	12	0.5245116
5	15	0.5193283
6	18	0.5211095
7	21	0.4944087

FIGURE 8 – Précision du clustering par K-Means pour différentes valeurs de K

On constate que la précision maximale est ici atteinte pour $K = 12$. Toutefois on observe un écart relatif inférieur à 5% avec les cas où $K \in A = \{15, 18\}$. On précise que chacun des clusterings a été réalisé avec la même graine pour le générateur de nombre aléatoire. Ici chacun des clusterings par K-Means utilise les mêmes centres pour l'initialisation. On peut donc nuancer ces résultats par le fait que la méthode K-Means choisie ici utilise des centres choisis au hasard, donc les résultats dépendent de la graine choisie dans notre programme. Le faible écart relatif de précision n'est donc pas suffisant pour dire que K=12 est vraiment le meilleur nombre de cluster puisqu'en fonction de la graine, il est possible que les valeurs incluses dans A aient une meilleure précision.

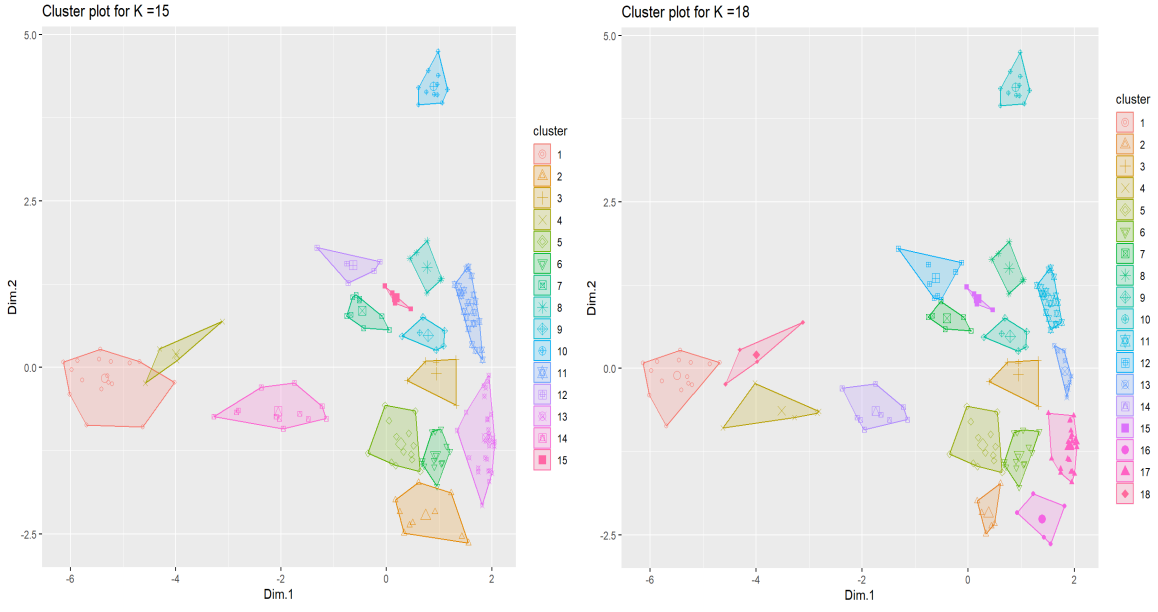
Pour obtenir une réponse plus tranchée, il faudrait répéter l'opération plusieurs fois pour différentes valeurs de la graine et prendre la moyenne de la précision pour chacun des nombres de clusters fixé. Nous n'allons pas implémenter cette fonctionnalité car elle s'avère gourmande en calcul et que le tableau précédent nous donne déjà un ordre d'idée de l'intervalle optimale du nombre de classe à choisir. On retient donc que le nombre de classe optimal K_{opt} semble être dans l'intervalle $[12, 18]$ avec ici un résultat empirique optimal pour $K = 12$.

On représente les données avec les clusters obtenus pour K $\in \{9, 12, 15, 18\}$, on obtient les graphiques suivants :



(a) K=9

(b) K=12



(c) K=15

(d) K=18

FIGURE 9 – Représentation graphiques des données et de leurs clusters pour plusieurs valeurs du nombre de clusters K

4.2 Clustering par méthode hiérarchique ascendante (CAH)

On souhaite à présent effectuer un clustering par méthode hiérarchique ascendante.
Pour cela nous utilisons la fonction *hclust()* de la librairie **FactoMiner**.
Nous commençons par faire un clustering par CAH sans spécifier de méthode spécifique, on obtient le dendogramme et le diagramme des hauteurs suivants :

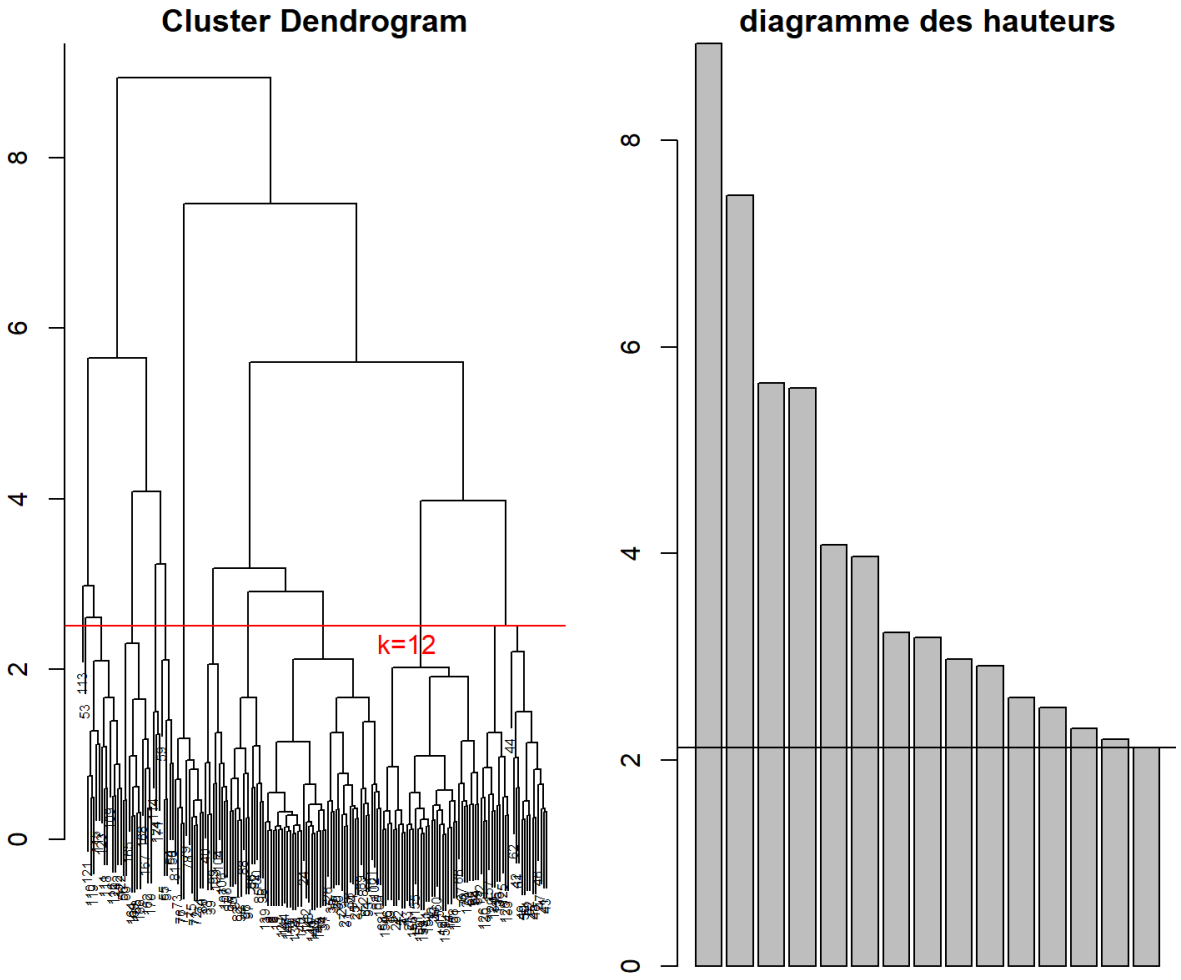


FIGURE 10 – Dendogramme(gauche) et diagramme des hauteurs(droite)

Nous avons représenté sur le dendogramme la hauteur à laquelle il faudrait le couper pour obtenir précisément 12 classes. Nous avons retenu ce nombre de manière arbitraire car il appartient à l'intervalle A décrit précédemment. On peut retrouver cette valeur en se référant au diagramme des hauteurs à gauche qui donne pour chaque nombre de cluster la hauteur correspondante sur le dendogramme.
Nous représentons les données après le clustering pour K $\in \{9, 12, 15, 18\}$, on obtient les figures suivantes :

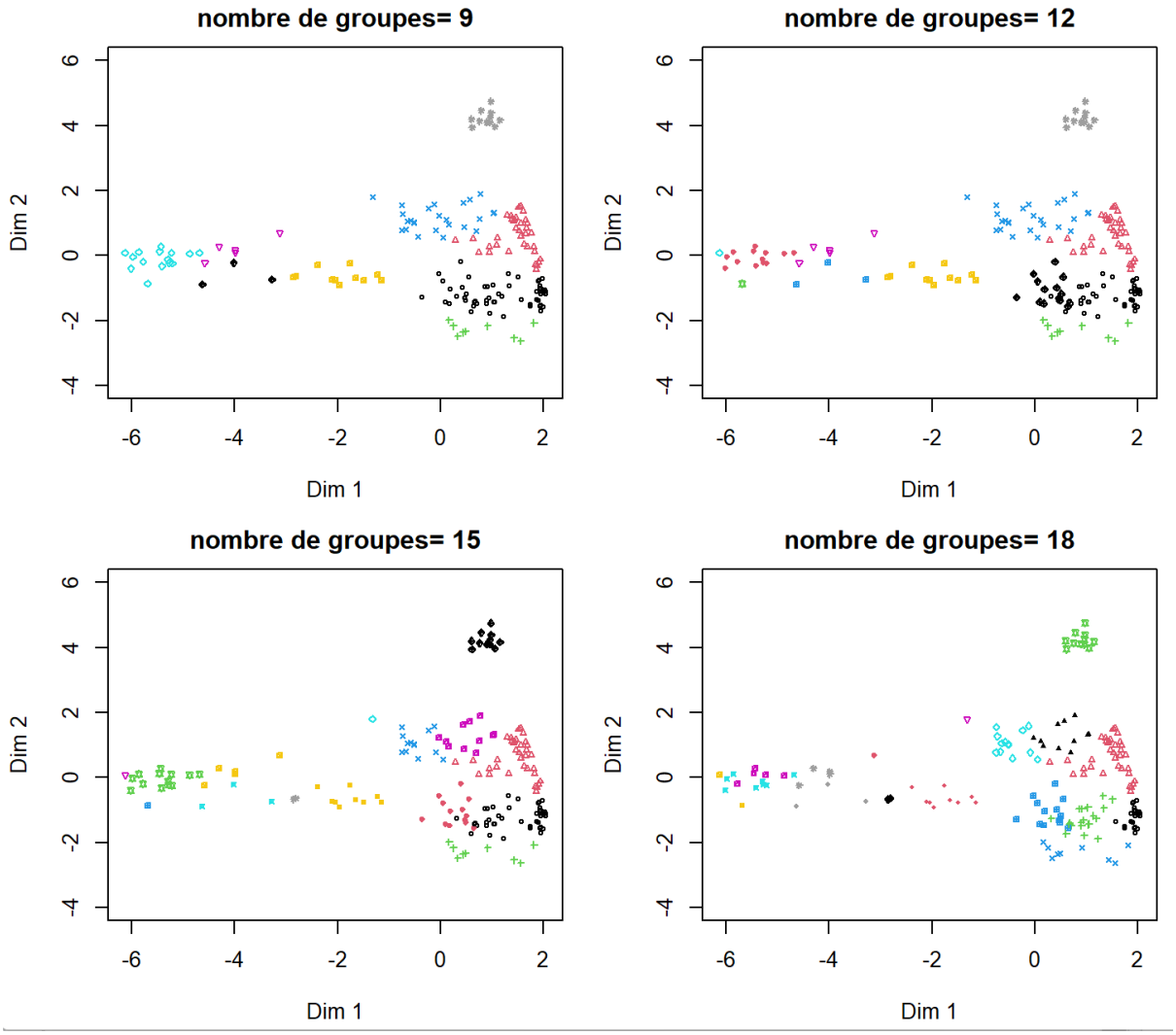


FIGURE 11 – Représentation du clustering par CAH par la méthode *complete* pour plusieurs valeurs du nombre de clusters

Nous souhaitons désormais comparer plusieurs méthodes possibles pour cette méthode. Nous retenons ainsi 5 méthodes à comparer :

1. *complete* : méthode par défaut de la fonction *hclust()*. Elle calcule toutes les dissimilarités par paire entre les éléments d'un cluster 1 et les éléments d'un cluster 2, et considère la valeur maximale de ces dissimilarités comme la distance entre les deux clusters.

2. *ward.D* : cette méthode minimise la variance intra-groupe (somme des erreurs). Les clusters sont combinés en fonction de la plus petite distance entre les clusters.

3. *average* : Cette méthode calcule toutes les dissimilarités par paire entre les éléments d'un cluster 1 et les éléments d'un cluster 2, et considère la moyenne de ces dissimilarités comme la distance entre les deux clusters.

4. *ward.D2* : Similaire à la méthode ward.D sauf que la somme des erreurs est élevée au carré.

5. *centroid* : Elle calcule la distance entre le centre d'un cluster 1 et le centre d'un cluster 2.

En comparant ces méthodes pour différentes nombres de clusters, on obtient :

	Nombre de Cluster	complete	ward.D	average	ward.D2	centroid
1	3	0.09198651	0.1789346	0.06230964	0.1693500	0.06230964
2	4	0.09063211	0.2212072	0.09327023	0.2108840	0.06259779
3	5	0.25210172	0.2973185	0.25573311	0.2900279	0.09377461
4	6	0.25046031	0.3989109	0.26043632	0.3043804	0.09309752
5	7	0.30813677	0.4457744	0.25160325	0.4094834	0.08988113
6	8	0.30972075	0.4850954	0.24692732	0.4201806	0.08939147
7	9	0.33578066	0.4823596	0.29589626	0.4966604	0.08985631
8	10	0.32967820	0.5025238	0.29716829	0.5179134	0.24574095
9	11	0.37162721	0.5143230	0.29894172	0.4988991	0.23802368
10	12	0.37372462	0.5131072	0.32235139	0.5105493	0.23772667
11	13	0.39993139	0.5307384	0.42166725	0.5245513	0.23780133
12	14	0.40316829	0.5472483	0.41952873	0.5785808	0.24172441
13	15	0.39718395	0.5502103	0.41396846	0.5844607	0.28771791
14	16	0.48127397	0.5542172	0.46741906	0.5833799	0.29822684

FIGURE 12 – Précision du clustering par CAH pour différentes méthodes et nombres de clusters

La précision maximale de 58,44607% est ici atteinte pour $K = 15$ et avec la méthode *ward.D2*, ce qui nous pousse à retenir une classification avec 15 groupes et à privilégier la méthode *ward.D2* pour annoter automatiquement les feuilles.
Il faut préciser que nous pouvons déterminer la meilleure méthode car nous possédons les vraies classifications des données pour vérifier nos classifications. Un deuxième est induit par le fait que nous savons qu'il y a 15 classes dans le jeu de données, ce qui peut pousser à vouloir obtenir ce nombre de clusters.

On réitère le même procédé, cette fois pour les données non normalisées et on obtient :

	Nombre de Cluster	complete	ward.D	average	ward.D2	centroid
1	3	0.04477695	0.06938614	0.06938614	0.09198651	0.06938614
2	4	0.08875469	0.21898669	0.06549395	0.24617398	0.06973213
3	5	0.15694019	0.25876809	0.13708073	0.24893994	0.06579073
4	6	0.15291474	0.27952792	0.17511137	0.25042815	0.06376510
5	7	0.15305257	0.38850709	0.17534214	0.27000351	0.08949520
6	8	0.27453918	0.38099381	0.17211533	0.30477707	0.08347742
7	9	0.26394239	0.44173200	0.16470146	0.44381471	0.24205301
8	10	0.26468279	0.43630673	0.27815755	0.42304372	0.24212867
9	11	0.24922607	0.47544208	0.27719005	0.41678365	0.24086709
10	12	0.24169273	0.46913215	0.28432518	0.40194642	0.24132078
11	13	0.24937597	0.48571741	0.29293730	0.43741351	0.23533692
12	14	0.31369964	0.48356275	0.27629511	0.45133654	0.25234401
13	15	0.32148247	0.46605517	0.34767316	0.44643865	0.24040038
14	16	0.32027411	0.43865070	0.33823050	0.44498931	0.25943044

FIGURE 13 – Précision du clustering par CAH pour différentes méthodes et nombres de clusters pour les données non normalisées

On constate une perte en précision lorsqu'on utilise les données non normalisées puisque la précision maximale atteinte ici est de 48.5%. Le fait de normaliser les données a donc une grande influence sur les résultats.

4.3 Méthode mixte et nombre optimal d'axes principaux

4.3.1 Méthode mixte

Dans cette partie nous allons dans un premier temps tenter un clustering par méthode mixte.
Nous utilisons ainsi d'abord un clustering par K-Means pour un très grand nombre de clusters. On extrait alors la liste des centres pour effectuer un clustering avec cette dernière pour le nombre de clusters final qu'on désire (ici K=15). On termine par effectuer un clustering par K-Moyenne avec nos données initiales et en y mettant comme centre les centres des classes trouvées à l'étape précédente. En utilisant les données normalisées sans effectuer de PCA et pour un nombre de clusters K = 15, on obtient une précision de 52,6%, ce qui est nettement moins que le clustering par CAH mais reste comparable et légérement meilleur que le clustering par K-mean. Toutefois on constate une amélioration de la précision de l'ordre de 2% par rapport au clustering par K-Means avec les mêmes paramètres.
On ne peut donc pas affirmer avec certitude que cette méthode est ici plus efficace que le clustering par K-Means car la graine du générateur de nombre aléatoire joue un rôle suffisant à relativiser cette apparente meilleure précision.

On peut de plus émettre une remarque : comme nous avons conservé l'entiéreté des données sans passer par un PCA car le jeu de données est faible en taille, cela réduit l'intérêt de cette méthode. Dans le cas

de données trop lourdes ou massives et nécessitant un PCA, il peut très vite devenir intéressant d'utiliser cette méthode afin de fluidifier l'éxécution tout en gardant une précision fiable.

4.3.2 Nombre optimal d'axes principaux

On effectue un clustering sur les $j \in \{1, 2, 3, 4, 5, 6, 7\}$ premières composantes principales pour déterminer le nombre optimal de composantes principales. Pour ce clustering, on utilise un clustering par CAH avec la méthode *Ward.D2* car ce dernier s'est avéré le plus précis dans les parties précédentes. On décide de prendre K = 15 clusters. On obtient les résultats suivants :

	précision par CAH
1	0.3010735
2	0.4604790
3	0.4922588
4	0.5548952
5	0.5425128
6	0.5748683
7	0.5921704

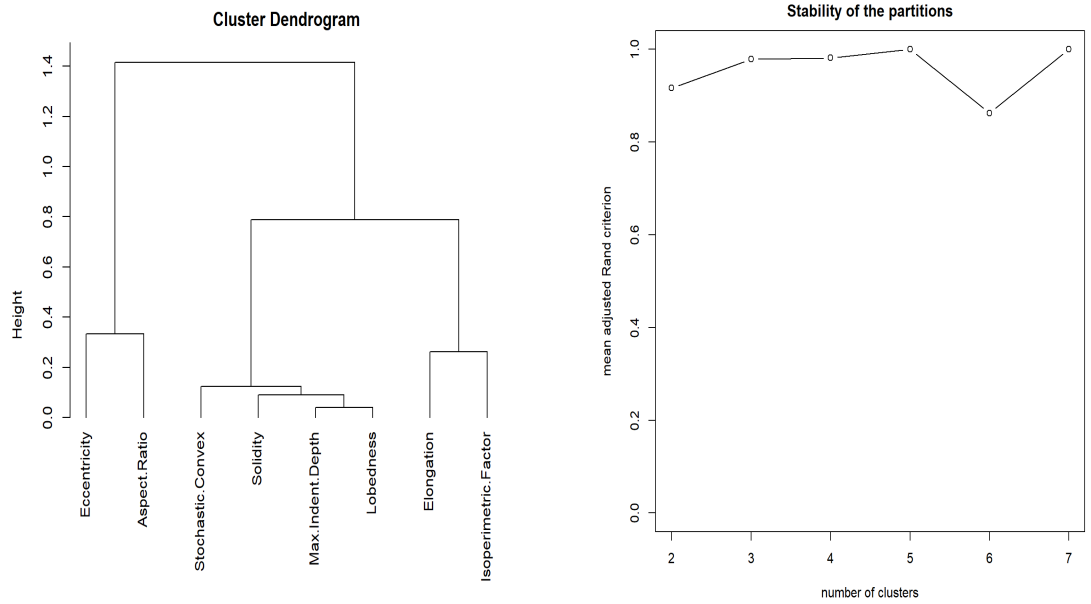
FIGURE 14 – précision du clustering pour différents nombres de composantes principales

On constate qu'on obtient une précision maximale si l'on garde les 7 premières composantes principales. Ce résultats n'est pas étonnant au vu de la méthode utilisée qui à recours à la somme des erreurs. La précision dépend alors directement de la quantité d'information disponible notamment sur la variance. En augmentant au maximum le nombre de composantes principales, on maximise l'information disponible et donc par corollaire la précision du clustering par CAH en utilisant *ward.D2*.

Une autre approche a été tentée dans notre code R, nous avons voulu utiliser une méthode mixte. Cependant cette méthode étant moins précise que le clustering par CAH, nous nous contentons de le mentionner aussi. Cette approche a été conservé dans le fichier .R associé à notre étude.

4.4 Clustering des variables

Dans cette partie, on souhaite faire un clustering des variables. Pour cela on a recours à la librairie *ClustOfVar*en y utilisation la fonction *hclustVar* pour obtenir le dendogramme des variables et la fonction *stability* pour déterminer le nombre de cluster de variable nécessaire. On obtient alors facilement le dendogramme du clustering des variables ainsi que le graphique de stabilité des partitions :



(a) Dendogramme du clustering des variables

(b) Graphique de stabilité des partitions

FIGURE 15 – Dendogramme et graphique de stabilité des partitions

On constate une stabilité extrêmement proche de 1 pour K $\in \{5, 7\}$, ce qui nous indique que ces deux choix de nombres de cluster de variables sont particulièrement pertinents. Vu que l'on souhaite diminuer au minimum le nombre de cluster, on sélectionne K=5 et on obtient alors comme clustering des variables :

```
        Eccentricity         Aspect.Ratio           Elongation
                   1                    1                    4
           Solidity    Stochastic.Convex Isoperimetric.Factor
                   2                    3                    4
    Max.Indent.Depth            Lobedness
                   5                    5
> res_k_mean_var$E
[1] 79.23541
```

FIGURE 16 – Clustering des variables

On obtient une homogénéïté des clusters de 79%, ce qui est un niveau satisfaisant dans le cadre d'une simplification des variables.
On peut interpréter ce résultat comme le fait qu'il est possible de se ramener à 5 variables au lieu des 8 initiales. Pour cela, on utilise les coefficients disponibles pour chaque cluster de variable en sortie de la fonction *kmeansvar* si l'on souhaite effectivement transformer nos données pour réduire le nombre de variables.

5 Conclusion

En conclusion, l'analyse menée sur ce jeu de données se révèle prometteuse.
En effet, l'ACP a donné de très bons résultats avec une réduction de dimension de 8 paramètres à 2 composantes principales tout en conservant 91% de variance expliquée. La représentation du nuage d'individus dans ce même plan nous a permis d'obtenir des répartitions assez distinctes dans l'espace des différents types de feuilles.
Nous retrouvons ensuite une représentation assez similaire à ce nuage de points lors du clustering effectué sur ce dataset. Nous avons également testé d'autres méthodes de classification non supervisée, le CAH s'avérant être celui avec la meilleure précision (58.44 pour 15 clusters).

Ces résultats, loin d'être optimaux, semblent cependant être satisfaisants compte tenu de la taille du jeu de données (seulement 171 feuilles, à peine plus de 10 exemples par classe) et du caractère non supervisé pour autant de classes.
Le but de cette étude étant de considérer un possible étiquetage automatique des données, il semblerait cependant prématuré de procéder à cette technique avant d'avoir amélioré le modèle, soit en augmentant la taille du jeu de données, soit en considérant de l'apprentissage semi-supervisé : les résultats obtenus par LDA dans l'article avoisine en effet les 88% de précision.

6 Annexes

6.1 Librairies R utilisées

Voici la liste des librairies R utilisées :

1. **corrplot**
2. **car**
3. **GGally**
4. **FactoMineR**
5. **factoextra**
6. **mclust**
7. **ClustOfVar**