



Analyse et Traitement des Données (ATD) — Examen pratique

Master 1 DSC — Saint-Étienne, France

Remarques préliminaires

Examen pratique d'Analyse et Traitement des Données (ATD) avec  d'une durée de deux heures.

Pour cet examen, vous pouvez utiliser votre ordinateur personnel ou celui de la salle informatique avec , RStudio et les packages usuels installés. Tous les documents numériques du cours sont autorisés (fichiers des diapositives des cours, énoncés des TD et TP, codes des TP). Attention, les recherches sur Internet (forums ou autres), la communication par messagerie électronique ou par messagerie instantanée (*webchat*), ou toute autre forme de communication sont interdites. L'examen est noté sur 20 points. Répondez aux questions dans un fichier de code en  à votre nom et votre prénom. Pour les réponses dans votre fichier, utilisez des commentaires, avec une phrase précédée par le caractère croisillon (le caractère # qui ressemble au symbole musical dièse). Avant la fin de l'épreuve, après avoir enregistré votre fichier, déposez votre fichier de code sur le répertoire Travaux de la plate-forme *Claroline*/Sciences du cours d'ATD (pas *Claroline Connect*) en tant que nouvelle soumission sur le répertoire de Travaux « Examen pratique d'ATD 2021-2022 ».

Cet examen correspond à 30 % de la note finale d'ATD.

Le jeu de données utilisé pour cet examen est le fichier `tourbillon.csv` présent sur la plate-forme *Claroline*, dans « Documents et liens », dans le répertoire `Examen_pratique_2021_2022`. Téléchargez ce fichier et importez-le (avec "Import Dataset") dans RStudio.

Questions

- 2 pts [1] Affichez le résumé des informations sur le jeu de données. Affichez un histogramme des valeurs pour chaque variable avec la fonction `qplot` du package `ggplot2`. Testez avec différentes valeurs du paramètre `binwidth` si nécessaire pour plus de précision dans le graphique. Est-ce que les variables semblent suivre une distribution normale ? Pourquoi ?
- 2 pts [2] Réalisez une ACP sur votre jeu de données. Représentez vos données avec les deux (premières) composantes principales de l'ACP. En commentaire, indiquez quel est l'intérêt d'une représentation effectuée avec l'ACP en général ainsi que l'intérêt d'une ACP pour ce jeu de données en particulier.
- 2 pts [3] Affichez un graphique avec la variable x pour l'axe des abscisses et la variable y pour l'axe des ordonnées. À votre avis, est-il possible de réaliser un modèle de régression linéaire afin de prévoir la valeur de y en fonction de x ? Si oui, donnez les paramètres a et b du modèle $y = a.x + b$. Si non, expliquez pourquoi en commentaire.
- 2 pts [4] Créez un nouveau jeu de données d'un millier de points de la façon suivante :
- soit a une variable entière allant de 1 à 1000
 - pour chaque valeur de a , calculez les valeurs des variables suivantes :
 - soit b une variable réelle égale à $a/100$
 - soit x une variable réelle égale à $b \times \sin(b)$
 - soit y une variable réelle égale à $b \times \cos(b)$

Les valeurs de x et y seront stockées dans un nouveau dataframe que vous afficherez. En vous inspirant des résultats obtenus, indiquez en commentaire comment le jeu de données `tourbillon` a pu être généré.

- 3 pts [5] Réalisez des classifications avec la méthode des k -means en faisant varier le nombre de clusters de 2 à 10. Pour cela, vous utiliserez la fonction `kmeans` avec les deux paramètres que sont le jeu de données (c'est-à-dire le dataframe `tourbillon`) et k (c'est-à-dire le nombre de clusters à obtenir). Pour chaque classification, affichez dans une nouvelle fenêtre les résultats obtenus, c'est-à-dire en faisant un plot du jeu de données et en utilisant pour chaque point la couleur correspondant au numéro du cluster issu de la classification (c'est-à-dire la composante `cluster` issue de l'objet retourné par la fonction `kmeans`). Après observation graphique de ces résultats, quel est le meilleur résultat obtenu sur ces 9 classifications par la méthode des k -means ? Est-ce pour autant un résultat satisfaisant ? Indiquez en commentaire pour quelle(s) raison(s).

3 pts [6] Réalisez des classifications avec la méthode de la classification ascendante hiérarchique (CAH).

Pour cela, vous utiliserez la fonction `hclust` qui prend en paramètres la matrice de distance et une méthode d'agglomération.

Pour la matrice de distance, vous utiliserez la fonction `dist` appliquée au jeu de données `tourbillon` (sans préciser de paramètre sur la méthode de distance, c'est la distance euclidienne qui est employée par défaut).

Pour la méthode d'agglomération, vous testerez les résultats obtenus au moyen des trois méthodes suivantes :

- agglomération avec le lien moyen (utilisation de la valeur moyenne des distances aux points issus du cluster, `method="average"`),
- agglomération avec saut minimal (ou méthode du lien simple, c'est-à-dire la distance au point le plus proche du cluster, `method="single"`),
- et agglomération avec saut maximal (méthode des liens complets, c'est-à-dire la distance au point le plus éloigné du cluster, `method="complete"`).

Pour chacune des classifications hiérarchiques employant une méthode d'agglomération donnée, affichez dans une nouvelle fenêtre le dendrogramme associé au résultat de la classification. À l'aide de la fonction `rect.hclust`, affichez des rectangles autour du nombre de clusters que vous aurez choisi. En commentaire, indiquez pour quelle(s) raison(s) vous avez décidé de couper votre dendrogramme à tel ou tel endroit ou en tel ou tel nombre de clusters.

Pour chacune de ces trois classifications, affichez dans une nouvelle fenêtre les résultats obtenus, c'est-à-dire en faisant un plot du jeu de données et en utilisant pour chaque point la couleur correspondant au numéro du cluster issu de la classification (c'est-à-dire la valeur de l'objet retourné par la fonction `cutree` appliquée sur l'objet retourné par `hclust`, en indiquant un nombre de clusters donné).

Y a-t-il une méthode d'agglomération qui donne des meilleurs résultats de classification que les autres sur ce jeu de données particulier ? Pour vous aider à répondre (sous forme de commentaire), rappelez-vous de la dernière question de l'examen théorique de la semaine dernière où il fallait comparer les classifications avec agglomération par lien simple et agglomération par lien complet.

3 pts [7] Réalisez des classifications avec la méthode DBSCAN, c'est-à-dire *Density-Based Spatial Clustering of Applications with Noise* (regroupement spatial basé sur la densité pour des applications présentant des données bruitées).

Pour cela, installez le package `dbscan` et chargez-le.

La méthode DBSCAN se lance avec la fonction `dbscan` du package du même nom avec 3 paramètres :

- `x` = le nom du jeu de données sur lequel appliquer la méthode (ici, le dataframe `tourbillon`),
- `eps`, la taille (rayon) du voisinage *epsilon* (ici, vous ferez varier *epsilon* de 0.1 à 0.9)
- `minPts`, le nombre de points minimum requis dans le voisinage *eps* pour constituer des clusters (et ne pas considérer qu'il s'agit de bruit) autour des points centraux (ici, nous garderons la valeur du paramètre par défaut qui est de 5 points).

Pour chacune des 9 classifications obtenues en faisant varier *epsilon* de 0.1 à 0.9 (avec un pas de 0.1), affichez dans une nouvelle fenêtre les résultats obtenus, c'est-à-dire en faisant un plot du jeu de données et en utilisant pour chaque point la couleur correspondant au numéro du cluster issu de la classification (c'est-à-dire la valeur de l'objet retourné par la fonction `dbscan` appliquée sur le jeu de données).

Y a-t-il des valeurs pour *epsilon* qui donnent des meilleurs résultats de classification que d'autres sur ce jeu de données ? Si oui, indiquez en commentaires pour quelles valeurs et essayez de comprendre pourquoi.

3 pts [8] Installez le package `cluster` et chargez-le. Installez le package `clusterSim` et chargez-le également.

Sélectionnez quelques classifications obtenues lors des trois questions précédentes, avec différents paramètres (différentes valeurs de *k* pour la méthode des *k*-means, différentes méthodes d'agglomération en CAH, différentes valeurs de *epsilon* pour DBSCAN) et affichez les résultats obtenus pour différentes mesures de qualité d'un partitionnement :

- le coefficient de silhouette, qui est un indice moyen calculé, pour chaque point, à partir de la différence entre la distance moyenne avec les points du même groupe que lui (cohésion) et la distance moyenne avec les points des autres groupes voisins (séparation). Si cette différence est négative, le point est en moyenne plus proche du groupe voisin que du sien : il est donc mal classé. À l'inverse, si cette différence est positive, le point est en moyenne plus proche de son groupe que du groupe voisin : il est donc bien classé. Le coefficient de silhouette proprement dit est la moyenne du coefficient de silhouette pour tous les points ;
- l'indice de Davies-Bouldin, qui est une mesure de qualité d'une partition d'un ensemble de données, introduite par David L. Davies et Donald W. Bouldin en 1979, et qui se calcule à partir de la moyenne du rapport maximal entre la distance d'un point au centre de son groupe et la distance entre deux centres de groupes.

La fonction `silhouette` du package `cluster` prend en paramètre le vecteur des numéros du cluster auquel appartient le point à l'issue de la classification ainsi que la matrice de distance du jeu de données. Le coefficient silhouette est la moyenne (`mean`) des valeurs de silhouette des points, c'est-à-dire la 3ème composante de l'objet retourné par la fonction `silhouette`. Le coefficient de silhouette varie entre -1 (pire classification) et 1 (meilleure classification).

La fonction `index.DB` du package `clusterSim` prend en paramètre le jeu de données et le vecteur des numéros du cluster auquel appartient le point à l'issue de la classification. L'indice de Davies-Bouldin varie entre 0 (meilleure classification) et $+\infty$ (pire classification).

Indiquez en commentaire les résultats obtenus et si ces indices se prêtent bien à une aide à l'interprétation des résultats de classification sur un jeu de données tel que `tourbillon`.