Analyse du jeu de données cars

Aissatou Signate Jacky Thay Yann Trividic

Introduction

Dans le cadre le cadre du cours Traitement numérique des données dispensé en L3 informatique par Dr Nicoleta Rogovschi, il a été demandé aux étudiants de réaliser un projet d'analyse de jeu données. L'objet de ce rapport est d'analyser un ensemble de voitures dont les caractéristiques sont enregistrées dans le fichier `275-cars.txt` qui, contrairement à son nom, contient les données de 261 voitures.

1. Lecture des données

Les données sont contenues dans un fichier `TXT`, les colonnes sont séparées par le caractère `,` et les lignes par un retour à la ligne. Les valeurs décimales sont séparées des valeurs entières par le caractère `.`.

```
cars = read.table("../data/275-cars.txt", sep=",", header=T)
print(head(cars))
```

```
##    mpg cylinders cubicinches  hp weightlbs time.to.60 year   brand
## 1 14.0         8         350 165      4209         12 1972      US
## 2 31.9         4          89  71      1925         14 1980  Europe
## 3 17.0         8         302 140      3449         11 1971      US
## 4 15.0         8         400 150      3761         10 1971      US
## 5 30.5         4          98  63      2051         17 1978      US
## 6 23.0         8         350 125      3900         17 1980      US
```

2. Descriptions du jeu de données

```
dim(cars)
```

```
## [1] 261   8
```

Le jeu de données cars contient 261 individus définis par 8 variables. Chaque individu représente une liste de caractéristiques d'une voiture. En se basant sur ce lien, on obtient les descriptions suivantes :

1. `mpg` [numérique, réel, continu] : la quantité prédite de gallons par mille (continue, arrondie à l'unité)
2. `cylinders` [numérique, entier, discret] : le nombre de cylindres dans le moteur. Peut être 3, 4, 6 ou 8.
3. `cubicinches` [numérique, entier, continu] : mesure du volume du moteur de la voiture en pouces cube.
4. `hp` [numérique, entier, continu] : puissance réelle du moteur en chevaux.
5. `weightlbs` [numérique, entier, continu] : le poids de la voiture en livres.
6. `time.to.60` [numérique, entier, continu] : durée nécessaire pour aller de 0 à 60 milles par heure.
7. `year` [numérique, entier, discret] : année de fabrication de la voiture.
8. `brand` [textuel, qualitatif, catégorielle] : région géographique de la marque de la voiture.

3. Prétraitement

```
cars[rowSums(is.na(cars)) > 0,] # counts the number of rows with NA values
```

```
## [1] mpg         cylinders   cubicinches hp          weightlbs   time.to.60
## [7] year        brand
## <0 rows> (or 0-length row.names)
```

Le jeu de données ne contient aucune valeur manquante, il peut donc être utilisé tel quel pour la suite de l'analyse.

Seule la variable `brand` est une variable textuelle et catégorielle dans ce jeu de données. Étant donné qu'il s'agit de la seule variable non quantitative, celle-ci peut être considérée comme une variable supplémentaire car la plupart algorithmes que nous utiliserons dans cette analyse demandent en entrée des tables de variables quantitatives. `brand` nous servira donc principalement dans l'interprétation des résultats de ces algorithmes. Nous pouvons par ailleurs changer son type pour le type `factor`, ce qui rendra les manipulations de cette variable plus simples.

Au premier abord, toutes les autres variables doivent être utilisés dans l'analyse, celles-ci ne semblant pas être liées entre elles directement. On peut noter que les unités de mesure associées aux différentes variables sont toutes différentes ; le jeu de données sera analysé en prenant en compte ces caractéristiques.

```
cars$brand = as.factor(cars$brand) # Transformation en factor
cars.scaled = cbind(scale(cars[-8]), cars[8])
```

4. Analyse univariée

4.1 Sommaire, distributions et critères de position

4.1.1 Variables quantitatives Grâce à la fonction `summary`, nous avons un aperçu global des différentes distributions des variables composant notre base de données en calculant des statistiques de base (critères de position et critères de dispersion). Pour les variables quantitatives, `summary` nous retourne le minimum, le maximum, la moyenne et les trois quartiles. La table 1, présente dans le document `tables.pdf`, contient les informations énoncées précédemment.

Pour calculer des quantiles d'un jeu d'observations stocké dans un vecteur v, nous utilisons la fonction `quantile`. Celle-ci calcule les quantiles à 0 %, 25 %, 50 %, 75 % et 100 %. Pour avoir les quantiles à d'autres ordres, il faut manipuler le paramètre `probs`. Dans notre cas , nous les calculons par intervalle de 10 %. Ces résultats sont trouvables dans la table 2.

On ajoute à cela les diagrammes en boîte, il s'agit de la figure 1 du document `figures.pdf`.

Ces diagrammes en boîtes permettent de visualiser plus facilement les résultats obtenus précédemment grâce à la fonction `summary`. Par exemple, pour la variable `mpg`, on peut lire que sa valeur médiane est 22, et que les valeurs sont un peu plus dispersées au-dessus de cette valeur médiane. Pour la variable `cylinders`, on remarque une symétrie parfaite : la distribution en-dessous de la médiane est très similaire à celle au-dessus de celle-ci. Son coefficient d'asymétrie est proche de 0. La même observation peut être faite concernant la variable `year`.

En ce qui concerne la variable `time.to.60`, on peut noter la présence de valeurs aberrantes. Une valeur aberrante est une valeur qui s'écarte fortement des valeurs des autres observations, anormalement faible ou élevée. Ici, ces valeurs correspondent à des voitures ayant une accélération anormalement élevée (8 secondes pour atteindre 60 mph pour la plus rapide) ou anormalement faible (25 secondes pour la plus lente). Ces valeurs ne semblent pas être des erreurs de mesure mais simplement des voitures sortant de la norme. Dans certains cas, il est nécessaire d'effectuer un traitement particulier sur ces valeurs. Avec ces valeurs en particuliers, nous considérons qu'il est possible de continuer l'analyse en l'état, celles-ci n'allant pas poser pas de problèmes.

4.1.2 Variables qualitatives Le jeu de données `cars` contient, en plus des sept variables quantitatives déjà abordées, une variable qualitative : la variable `brand`. Cette donnée, par sa nature, doit être traitée séparément du reste. La répartition de occurrences des différentes marques peut être appréciée dans la figure 2, en annexes, de même que la table 3, illustrant la table de contingence des différentes marques dans le jeu de données.

Comme on peut le voir, la distribution des individus en fonction de la variable brand est tout sauf uniforme : près de deux tiers sont des voitures américaines (62 %) tandis que les voitures japonaises et européennes se partagent de manière à peu près égale le dernier tiers des données, soit respectivement environ 20 et 18 %. Cette surreprésentation des voitures américaines sera à prendre en compte tout au long de l'analyse.

Il nous est aussi possible de résumer les différentes distributions des variables en fonction des valeurs de `brand`. Ces résultats sont disponibles dans la table 4.

Le résultat obtenu nous permet d'avoir une idée claire des différences de distribution des valeurs en fonction de `brand`. On peut voir notamment une hiérarchie entre les provenances des voiture et leur consommation en carburant : en règle générale, une voiture japonaise consommera moins qu'une européenne, qui elle-même consommera moins qu'une américaine.

D'autres interprétations peuvent être effectuées à partir de ces sommaires :

- Les moteurs des voitures japonaises et européennes ont généralement moins de cylindres que ceux des américaines. La majorité des voitures européennes et japonaises auront quatre cylindres, tandis que l'écart-type est plus important pour les voitures amércaines ($\sigma = 1,62$).
- Le volume des moteurs des voitures américaines est en moyenne beaucoup plus important que celles des japonaises et des européennes. On peut noter une différence d'un facteur $2,5$.
- Les voitures américaines sont beaucoup plus puissantes que les voitures européennes et japonaises en moyenne.
- Les voitures américaines sont en moyenne beaucoup plus lourdes que les voitures européennes, qui sont elles-même légèrement plus lourdes que les voitures japonaises.
- L'accélération des voitures américaines est globalement meilleure que celle des japonaise, qui elle-même est légèrement meilleure que celle des européennes.

De manière générale, on voit que les voitures européennes ont des caractéristiques proches des voitures japonaises. Les voitures américaines sont quant à elles très différentes des deux autres catégories.

4.2 Corrélations

```r
cars.correlations <- cor(cars[-8]) # on inclut pas la dernière variable
```

Une matrice de corrélation est utilisée pour visualiser la liaison linéaire entre plusieurs variables. Il s'agit d'un tableau contenant les coefficients de corrélation entre chaque variable. Pour l'obtenir, on peut utiliser la fonction `cor`, présente dans les fonctions de base de R. La table des corrélations est affichée dans la table 5 document `tables.pdf`.

Nous pouvons également visualiser et représenter graphiquement notre matrice avec un corrélogramme. Ce corrélogramme peut être apprécié avec la figure 3. Cet outil permet de mettre en évidence les variables les plus corrélées : les coefficients de corrélation sont colorés en fonction de leur valeur.

La valeur 1 indique que les deux variables sont exactement corrélées, c'est le cas avec une relation parfaitement linéaire entre deux variables. À l'inverse, une corrélation de -1 entre deux variables indique une parfaite anti-corrélation entre ces dernières. Les corrélations dont la valeur absolue sont supérieures à $0,5$ peuvent être considérées comme des corrélations fortes. Les autres, celles comprises entre $-0,5$ et $0,5$, peuvent être considérées comme des corrélations faibles.

À la lecture de ce corrélogramme, on peut noter que la variable `mpg` est très fortement anti-corrélée avec les variables `cylinders`, `cubicinches`, `hp` et `weightlbs`, avec des valeurs allant de $-0,78$ à $-0,82$. De manière générale, on remarquera de très fortes corrélations positives entre les quatre dernières variables citées (toutes supérieures à $0,85$). Ces observations peuvent s'interpréter de la manière suivante : plus un moteur est gros, plus il est puissant et la voiture est lourde, et inversement. Plus la voiture est grosse et puissante, et plus sa consommation en carburant sera élevée.

Des corrélations existent aussi entre ces variables et la variable `time.to.60`, bien qu'elles soient moins importantes. On pourra noter que plus une voiture a une consommation de carburant élevée et plus elle aura une forte accélération ; plus son moteur est gros et la voiture lourde, et plus ses capacités d'accélération seront importantes.

5. Classification

La section qui suit est basée sur le travail disponible à ce lien.

5.1 Tendance à la classification

Avant d'utiliser quelconque algorithme de classification, il peut être intéressant de quantifier la propension du jeu de données à contenir des classes distinctes et pertinentes. Cette quantification peut être effectuée grâce au critère d'Hopkins. Ce critère permet de mettre une valeur sur le degré d'uniformisation de la distribution des valeurs d'un jeu de données. Si la valeur prise par le critère est proche de 0, alors les distributions sont parfaitement uniformes. Si la valeur approche $0,5$, alors distribution des valeurs est proche de celle d'une série générée par une loi de Poisson. Les cas qui nous intéressent sont ceux où le critère approche 1, cette valeur indiquant des classes très clairement définies.

```
library(factoextra)
```

```
## Loading required package: ggplot2
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
get_clust_tendency(cars[-8], 100, graph=FALSE)$hopkins_stat
```

```
## [1] 0.8325368
```

En utilisant la fonction `get_clust_tendency` de la bibliothèque `factoextra`, on remarque que la valeur prise par le critère de Hopkins pour le jeu de données cars est très forte. On peut donc continuer le travail de classification.

5.2 K-moyennes

L'algorithme des K-moyennes, très simple à mettre en place et très utilisé dans le domaine de la classification, permet de distinguer des groupes d'individus en minimisant la somme totale des carrés des distance. Cette méthode suppose cependant de préciser en paramètre le nombre de classes du jeu de données ; cependant, dans notre cas, l'information est manquante.

Notre jeu de données étant en sept dimensions (sept variables quantitatives), il est difficile de visualiser les résultats de l'algorithme des K-moyennes pour estimer le nombre de classes le plus approprié. Afin de surmonter ce problème, il existe de nombreuses approches mathématiques pour quantifier la qualité des résultats de l'algorithme K-moyennes. La plus connue est celle du coefficient de silhouette. Nous l'utiliserons dans un premier temps pour obtenir des résultats préliminaires, une représentation est disponible avec la figure 4.

Le coefficient de silhouette mesure la qualité de la partition obtenue après avoir appliqué un algorithme de classification. Ici, on lance une série d'algorithmes K-moyennes avec le nombre de classe minimum à 1 et le

maximum à 15. On obtient les scores moyens obtenus par les différentes partitions en fonction du nombre de classes.

On remarque que la meilleure partition a obtenu un score d'environ $0,62$ (celui-ci pouvant monter jusqu'à 1) avec deux classes. Cependant, les scores obtenus par les partitionnements à trois et quatre classes sont relativement proche du meilleur score. La différence entre ces scores n'est pas suffisante pour en garder un seul.

Il est donc préférable de garder les partitionnements à deux, à trois, et à quatre classes comme effectué ci-dessous. Il est à noter que, bien que ce ne soit pas précisé explicitement, 10 itérations de l'algorithme sont effectués pour chaque partitionnement afin de pallier à l'instabilité de K-moyennes à l'initialisation.

```
cars.kmeans.cluster2 <- kmeans(cars[-8], 2) # K-moyennes pour partitionnement à deux classes,
cars.kmeans.cluster3 <- kmeans(cars[-8], 3) # à trois classes,
cars.kmeans.cluster4 <- kmeans(cars[-8], 4) # à quatre classes.
```

Une visualisation de ces différents partitionnements couplés aux distributions normalisées des variables sont disponibles avec les figures 5, 6 et 7.

On peut voir sur ces différents graphiques les classes se séparer clairement. On remarque que les variables corrélées voient la distributions de leurs classes être similaires, ce qui était prévisibles. Dans la même idée, on observe que les distributions des classes sont en miroir lorsque qu'on regarde deux variables anti-corrélées. Ces points sont tous cohérents avec les interprétations données dans la section *4.2 Corrélations*.

Un point intéressant à noter est que la séparation des classes est plus prononcés pour certaines variables que pour d'autres. Par exemple, la variable `weightlbs` (index 5) a dans les trois cas des classes parfaitement séparées. Au contraire, `cubicinches` (index 1) voit la distribution de ses classes moins distincte.

En addition à cela, on peut comparer ces classes à la classe explicitement définie par le jeu de données : la variable `brand`. La distribution de ces variables est disponible dans la figure 8.

Les voitures américaines prennent bien entendu la plus grande part des individus (voir section *4.1.2 Variables qualitatives* [**Vérifier section**]). On remarque que les voitures américaines prennent presque systématiquement un côté du spectre des valeurs, tandis que les japonaises et les européennes se partagent la seconde extrémité de manière plus ou moins distincte selon les variables.

En regardant conjointement la distribution des variables colorées par le résultat de K-moyennes à trois classes et ce dernier graphique, on peut tenter d'assimiler les classes obtenues avec K-moyennes avec les valeurs de `brand` dans une table de contingence, disponible dans le document `tables.pdf` avec la table 7.

En observant les résultat d'une table de contingence correspondant à cette hypothèse, cette dernière s'en voit invalidée ; les résultats ne soutiennent pas l'idée que les classes trouvées correspondraient de manière acceptable aux valeurs de `brand`. En effet, la première classe contient autant de voiture américaines que japonaises ou européennes malgré le fait qu'elle soit la plus importante des trois classes (46 % des individus).

Cependant, il est à noter que cette même classe (la première) contient l'extrême majorité des voitures européennes et japonaises. De plus, elle contient une minorité des voitures américaines. Remarquons aussi que la troisième classe ne contient qu'une seule voiture non-américaine tandis qu'elle contient 61 américaines. Une interprétation similaire, bien que moins prononcée, peut être faite sur la classe 2. Les classes 2 et 3 caractérisent relativement bien les voitures américaines.

Il est donc intéressant maintenant de voir si, en groupant les voitures européennes et japonaises, on obtient un meilleur partitionnement avec $k = 2$. Vous pouvez appréciez ces résultats avec la figure 9 et la table 8.

```
cars.combined.brand <- cars$brand
levels(cars.combined.brand)[1:2] <- 'Non-US' # remplace Japon et Europe par Non-US
```

En comparant visuellement les résultats obtenus avec la fonction `stripplot_clusters`, on observe que même s'il y a une certaine correspondance entre les résultats obtenus, elle reste approximative.

Cette même observation peut-être faite en lisant la table de contingence. On peut lire que plus de 95 % des voitures non américaines sont concentrées dans la classe 1. Cependant, les 95 voitures non américaines de la classe 1 ne représente que 60 % du nombre de voitures totales de la classe 1. En contrepartie, 60 % des voitures américaines sont dans la classe 2, et les 40 % restants étant dans la classe 1.

La tendance est assez claire : si une voiture est non américaine, il y a de grandes chances pour qu'elle soit dans la classe 1. Cependant, les voitures américaines peuvent appartenir aux deux classes, bien que leur appartenance soit plus prononcées pour la classe 2.

En conclusion, l'interprétation des résultats de l'algorithme des K-moyennes permet donc de soutenir l'hypothèse qu'il existe bel et bien des différences quantitatives entre les différentes marques de voitures, et que celles-ci peuvent être partitionner au moins en partie. Les différences les plus grandes sont entre les voitures américaines et les non américaines. Les voitures européennes et japonaises sont différentiables, mais dans une moindre mesure. Ces différences ne permettent cependant pas d'avoir un partitionnement clair et sans équivoque entre les marques à partir de l'algorithme des K-moyennes ; une part importante des voitures américaines est indifférenciée des voitures non américaines. Il faut donc continuer l'analyse.

5.3 Classification Ascendante Hiérarchique (CAH)

5.3.1 Critère de Ward Avant de commencer à travailler sur le jeu de données avec l'algorithme de la classification ascendante hiérarchique, il est important de noter que le nombre d'individus du jeu de données (261) est conséquent pour cet algorithme. Cela signifie que le dendrogramme résultant de la CAH comportera 261 feuilles ; la visualisation sera difficilement lisible et interprétable avec le dendrogramme seulement.

Une possibilité pour réduire le nombre d'individus serait d'appliquer un prétraitement comme K-moyennes avec par exemple $k = 50$, puis d'appliquer la CAH sur les parangons des cinquante centroïdes trouvés. Ici, nous choisirons une autre approche pour interpréter les résultats, encore une fois basée sur la matrice de contingence entre `brand` et les classes obtenues. Dans cette partie, nous utiliserons le critère de Ward par défaut, aucune raison ne permettant de prioriser un autre critère pour l'instant. Le critère du lien minimum et celui du lien maximum sont utilisés dans la section suivante.

```
cars.dist = dist(cars.scaled[-8]) # matrice des distances euclidiennes entre individus

# "ward.D2" correspond au vrai critère de Ward utilisant le carré de la distance
cars.hc.ward = hclust(cars.dist, method="ward.D2")
cars.hc.ward.inertia = sort(cars.hc.ward$height, decreasing = T)
```

Avant même de s'intéresser au dendrogramme en lui-même, il est intéressant de visualiser son inertie. Celle-ci est disponible dans la figure 10. Ici, deux sauts se distinguent nettement. Les deux variations d'inertie les plus grandes sont à 2 classes et à 4 classes, mis en valeur dans la figure 11. La figure 12, quant à elle, montre le dendrogramme avec les différents niveaux de coupe séparés.

Encore une fois, comme avec K-moyennes, il pourrait être intéressant d'interpréter les résultats de la table de contingence entre `brand` et les classes résultant de la CAH, ce paramètre n'entrant pas en compte dans le calcul de la matrice de dissimilarités. Pour ce faire, on utilise la fonction `afficher_table_contingence_clusters`, dont le résultat est rendu disponible dans la table 9.

Pour le partitionnement à deux classes, on peut observer que 100 % des voitures non américaines sont contenues dans la classe 2. Cela constitue une différence majeure avec le partitionnement en deux classes obtenus avec K-moyennes, qui n'atteignait que 95 %.

La classe 2 comporte 54 % de voitures non américaines et 46 % de voiture américaines. La première classe comporte 100 % de voitures américaines. Le même ratio pour K-moyennes était de 60 contre 40 %.

Ces observations renforcent la validité des interprétations effectuées grâce aux résultats de K-moyennes : les voitures japonaises et européennes sont assez proches en ce qui concerne notre jeu de données, et dans

l'ensemble assez différentes des voitures américaines. Certaines voitures américaines sont cependant plus proches des voitures européennes et japonaises.

Pour le partitionnement à quatre classes, on remarque que les voitures européennes et japonaises sont chacune présentes dans les trois mêmes classes dans des proportions relativement proches. Voici le résultat obtenu lorsque l'on décide de les regrouper. Vous pouvez apprécier cette table dans la table 10.

Comme avec le partitionnement à deux classes, on observe qu'il existe une classe seulement constituées de voitures américaines (il s'agit rigoureusement de la même classe, voir le dendrogramme), et les autres constituées d'un mélange de voitures américaines et non américaines. Cependant, à la différence du partitionnement précédent, les valeurs de `brand` sont plus finement séparées.

En effet, la classe 3 contient 78 % de voitures non américaines, et la classe 4 contient 85 % de voitures américaines. Ces deux dernières classes caractérisent donc dans des proportions concluantes la différence entre les voitures américaines et non américaines. La classe 2, quant à elle, contient 58 % de voitures non américaines contre 42 % de voitures américaines. C'est dans cette classe que réside le plus d'incertitude quant à la détermination de la valeur de `brand`. Cette classe compte pour 21,5 % du nombre d'individus total.

Le partitionnement à quatre classe est donc un meilleur partitionnement que celui à deux classes en ce qui concerne la catégorisation par marques. Celui-ci est globalement plus fin pour différencier les voitures américaines des non américaines. Ce partitionnement surclasse aussi de loin tous les résultats obtenus avec l'algorithme des K-moyennes.

```
cars.hc.ward.cluster4 <- cutree(cars.hc.ward, 4)
```

Etant donné que seule la classe 2 présente des résultats peu concluant, il est intéressant de vérifier la structure de l'arbre qui la constitue pour peut-être la diviser en sous-classes.

On recommence donc le même procédé avec uniquement les individus de la classe 2 (voir l'annexe 1). Bien que le partitionnement en trois sous-classes puisse affiner les résultats de la classification au sein de la classe 2, cet affinage est trop faible pour mériter l'ajout de trois nouvelles classes.

5.3.2 Critères du lien minimum et du lien maximum Des CAH ont aussi été effectuées en utilisant les critères du lien minimum et du lien maximum. Les résultats n'ayant pas été concluants, nous avons décidé de laisser ces résultats en annexe. Voir les annexes 2 et 3.

6. Analyse en composantes principales pour interpréter les classes

Grâce à l'ACP, nous partons des corrélations entre les variables pour les résumer dans des plans factoriels. L'information originellement contenue dans les sept variables quantitatives du jeu de données va être résumée dans de nouveaux sous-espaces générés en maximisant l'inertie des données. En d'autres termes, l'analyse en composantes principales permet de condenser les informations des variables originelles dans de nouvelles variables moins nombreuses pour en faciliter le traitement et l'analyse.

Comme énoncé dans la section *3. Prétraitement*, la variable `brand` peut être utilisée comme variable supplémentaire lors de cette opération, celle-ci étant la seule variable catégorielle du jeu de données.

```
library(FactoMineR)
cars.pca <- PCA(cars.scaled, scale.unit=F, ncp = 10, graph = F, quali.sup = 8)
```

On obtient alors sept composantes principales allant de `Dim1` à `Dim7`. Les résultats qui suivent découlent de l'interprétation de la table 11 contenant un sommaire des résultats de l'ACP et du cercle de corrélation obtenu en figure 13.

Le tableau des valeurs propres est un outil permettant de déterminer quels sont les axes à prendre en compte pour l'ACP. Il est important que les valeurs propres des axes retenus restituent une part importante de la variance totale du jeu de données. Cela signifie que la somme de l'inertie expliquée par chacun des axes

représente une partie importante de l'inertie totale. C'est pour cette raison que nous allons retenir `Dim1` et `Dim2`, représentant respectivement 72 et 13 % d'inertie, avec un total de 85 % d'inertie. On peut dire que ces deux axes restituent 85 % des informations du jeu de données d'origine.

15 % de l'inertie totale du jeu de données n'est donc pas considérée lors des interprétations que nous émettons. Nos interprétations sont cependant assez larges et solides pour ne pas pouvoir être remises en question par cette marge d'erreur.

Le graphique nous montre que sur les sept variables représentées, six sont corrélées avec `Dim1`, une observation cohérente puisque cette composante a une inertie de 72 % dans l'ensemble du data set. Parmi les variables corrélées négativement avec `Dim1` nous trouvons `mpg` (-89 %) et `time.to.60` (-71 %). Les variables corrélées positivement avec `Dim1` sont `hp` (95%), `cylinders` (93%), `cubicinches` (96%) et `weightlbs` (92%). La principale information à extraire de cette composante est donc principalement le gabarit du véhicule et du moteur, ainsi que sa puissance. Plus une voiture est grosse et puissante, plus elle sera représentée sur la partie droite du plan des individus, et inversement.

Pour `Dim2`, l'unique variable qui lui est corrélée est `year` à 86 %. `Dim2` caractérise donc l'âge du véhicule : celui-ci est anti corrélé avec `Dim2`, ce qui implique que sur le graphique des individus, les voitures les plus vieilles seront plutôt en bas du graphique.

On voit sur la figure 14 que l'ensemble des variables se trouvent relativement loin de l'origine, elles sont donc bien représentées par ce plan factoriel. On remarque que la variable `time.to.60`. est représentée seulement à hauteur de 50 % par ce plan. 49 % de sa variance est contenue dans la troisième composante. Quelconque interprétation basée sur uniquement la moitié de la variance ne serait pas assez rigoureuse dans notre cas.

7. Classification hiérarchique sur composantes principales

```
cars.hcpc <- HCPC(cars.pca, nb.clust = -1, consol=T, iter.max=50,
                  min = 3,
                  metric = "euclidean", method = "ward",
                  graph=F)
```

La HCPC suggère un partitionnement à trois classes, en plus du dendrogramme disponible dans la figure 15, on peut observer ces classes sur le graphique des individus du premier plan factoriel dans la figure 16.

Les résultats de la table 12 indiquent que les individus dans les groupes 1 et 3 ont des coordonnées élevées sur l'axe 1 (`Dim1`) et l'axe 2 (`Dim2`) . Les individus du groupe 2 ont des coordonnées élevées uniquement sur le deuxième axe.

La table 13, quant à elle, indique les cinq meilleurs individus les plus proches du centre de la classe ; ce sont les parangons de ces différentes classes. La distance entre chaque individu et le centre du groupe est fournie. Par exemple, les individus représentatifs pour le groupe 2 inclus les suivants : 87, 34, 229, 208 et 159.

Conclusion

L'analyse de du jeu de données `cars` a permis de mettre en exergue d'un côté les différences entre les voitures américaines et les voitures non américaines, et d'un autre côté les similarités entre les voitures européennes et japonaises. Pour aller plus loin, il serait intéressant de travailler avec des outils de régression pour mettre d'établir des modèles de classification, ce jeu de données ayant toutes les caractéristiques permettant de classer efficacement les voitures selon leurs origines.