T. D. n° 1
Analyse en Composantes Principales

Résumé

Ce document est le T.D. n° 1 du module intitulé "Analyse exploratoire". Il reprend rapidement des éléments du cours et propose une mise en pratique interactive de l'ACP, de l'AFC et de l'ACM. Dans ce T.D. nous utiliserons une ACP centrée et réduite, appelée ACP normée. L'objectif est d'appliquer différentes types d'analyse en composantes en utilisant les packages *ade4*, *FactoMineR*, *PCAmixdata* sous le logiciel libre R et d'interpréter les résultats.

1 McDonald's

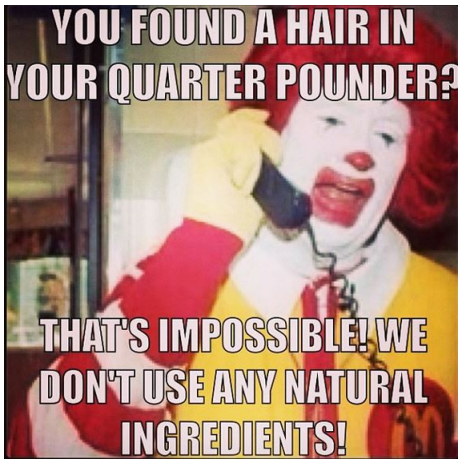


FIGURE 1 – Un principe du tiers exclu
source :`http://thescienceofeating.com`

Cet ensemble de données fournit une analyse nutritionnelle de chaque élément de menu sur le menu US McDonald's, y compris le petit-déjeuner, les hamburgers de bœuf, les sandwichs de poulet et de poisson, les frites, les salades, le soda, le café et le thé, les milk-shakes et les desserts, bref, tout ce qui peux vous faire plaisir un lendemain de soirée arrosée.

1.1 Chargement des données

Commencez par charger les données du fichier menu.csv. Associez ces données à un dataframe (fonction `as.dataframe()`) et appliquez la fonction `summary()`. Il est également possible de télécharger le fichier menu.csv sur ce lien :
`https://www.kaggle.com/mcdonalds/nutrition-facts`.

```
              Category                                          Item
                                Serving.Size    Calories
Coffee & Tea       :95   1% Low Fat Milk Jug                    :
   1   16 fl oz cup: 45   Min.   :   0.0
Breakfast          :42   Apple Slices                           :
   1   12 fl oz cup: 38   1st Qu.: 210.0
Smoothies & Shakes:28   Bacon Buffalo Ranch McChicken          :
   1   22 fl oz cup: 20   Median : 340.0
Beverages          :27   Bacon Cheddar McChicken                :
   1   20 fl oz cup: 16   Mean   : 368.3
Chicken & Fish     :27   Bacon Clubhouse Burger                 :
   1   21 fl oz cup:  7   3rd Qu.: 500.0
Beef & Pork        :15   Bacon Clubhouse Crispy Chicken Sandwich:
   1   30 fl oz cup:  7   Max.   :1880.0
(Other)            :26   (Other)
   :254   (Other)      :127
Calories.from.Fat   Total.Fat       Total.Fat....Daily.Value.
   Saturated.Fat    Saturated.Fat....Daily.Value.
Min.   :   0.0    Min.   :  0.000   Min.   :  0.00            Min.
      : 0.000   Min.   :  0.00
1st Qu.:  20.0    1st Qu.:  2.375   1st Qu.:  3.75            1st
   Qu.: 1.000   1st Qu.:  4.75
Median : 100.0    Median : 11.000   Median : 17.00
   Median : 5.000   Median : 24.00
Mean   : 127.1    Mean   : 14.165   Mean   : 21.82            Mean
      : 6.008   Mean   : 29.97
3rd Qu.: 200.0    3rd Qu.: 22.250   3rd Qu.: 35.00            3rd
   Qu.:10.000   3rd Qu.: 48.00
Max.   :1060.0    Max.   :118.000   Max.   :182.00            Max.
      :20.000   Max.   :102.00

  Trans.Fat       Cholesterol     Cholesterol....Daily.Value.
     Sodium       Sodium....Daily.Value.
Min.   :0.0000   Min.   :  0.00   Min.   :  0.00             Min.
      :   0.0   Min.   :  0.00
1st Qu.:0.0000   1st Qu.:  5.00   1st Qu.:  2.00             1st
   Qu.: 107.5   1st Qu.:  4.75
Median :0.0000   Median : 35.00   Median : 11.00
   Median : 190.0   Median :  8.00
Mean   :0.2038   Mean   : 54.94   Mean   : 18.39             Mean
      : 495.8   Mean   : 20.68
3rd Qu.:0.0000   3rd Qu.: 65.00   3rd Qu.: 21.25             3rd
   Qu.: 865.0   3rd Qu.: 36.25
Max.   :2.5000   Max.   :575.00   Max.   :192.00             Max.
      :3600.0   Max.   :150.00

Carbohydrates    Carbohydrates....Daily.Value. Dietary.Fiber
   Dietary.Fiber....Daily.Value.     Sugars
Min.   :  0.00   Min.   : 0.00                  Min.   :0.000   Min
   .   : 0.000                Min.   :  0.00
1st Qu.: 30.00   1st Qu.:10.00                  1st Qu.:0.000   1st
    Qu.: 0.000                1st Qu.:  5.75
Median : 44.00   Median :15.00                  Median :1.000
   Median : 5.000                Median : 17.50
```

```
 Mean   : 47.35   Mean   :15.78                    Mean   :1.631
    Mean   : 6.531                  Mean   : 29.42
 3rd Qu.: 60.00   3rd Qu.:20.00                   3rd Qu.:3.000   3rd
     Qu.:10.000                  3rd Qu.: 48.00
 Max.   :141.00   Max.   :47.00                   Max.   :7.000   Max
    .   :28.000                 Max.   :128.00

    Protein       Vitamin.A....Daily.Value. Vitamin.C....Daily.Value
       . Calcium....Daily.Value.
 Min.   : 0.00   Min.   :  0.00            Min.   :  0.000
              Min.   : 0.00
 1st Qu.: 4.00   1st Qu.:  2.00            1st Qu.:  0.000
              1st Qu.: 6.00
 Median :12.00   Median :  8.00            Median :  0.000
              Median :20.00
 Mean   :13.34   Mean   : 13.43            Mean   :  8.535
              Mean   :20.97
 3rd Qu.:19.00   3rd Qu.: 15.00            3rd Qu.:  4.000
              3rd Qu.:30.00
 Max.   :87.00   Max.   :170.00            Max.   :240.000
              Max.   :70.00

 Iron....Daily.Value.
 Min.   : 0.000
 1st Qu.: 0.000
 Median : 4.000
 Mean   : 7.735
 3rd Qu.:15.000
 Max.   :40.000
```

```
> chisq.test(data_macdo$Calories, data_macdo$Total.Fat)

        Chi-squared test for given probabilities\\

data:  c(data_macdo$Calories, data_macdo$Total.Fat)\\
X-squared = 163710, df = 519, p-value < 2.2e-16
```

À vous !

1. Justifiez l'utilisation d'un test du χ^2 sur le jeu de données.
2. Quelles conditions devez-vous respecter pour utiliser un test du χ^2 ?
3. Concluez sur l'indépendance des variables `Calories` et `Total.Fat`.
4. Testez l'indépendance des variables explicatives (deux à deux) et vous présenterez vos résultats sous forme de tableau pour les variables suivantes : `Calories`, `Total.Fat`, `Cholesterol`, `Sodium`, `Sugars` et `Protein`.

1.2 Corrélation linéaire entre deux variables

Une ACP se fait sur des variables quantitatives continues. Commencez par afficher le coeffcient de corrélation linéaire de Pearson entre les variables quantitatives à l'aide de la fonction `cor()`. Ce coefficient permet de détecter la présence ou l'absence d'une relation linéaire entre deux caractères quantitatifs continus. En principe, le coefficient de Pearson n'est applicable que pour mesurer la relation entre deux variables X et Y ayant une distribution gaussienne et ne comportant pas de valeurs exceptionnelles. Si ces conditions ne sont pas vérifiées (cas fréquent) l'emploi de ce coefficient peut aboutir à des conclusions erronées sur la présence ou l'absence d'une relation linéaire entre les deux variables. Nous noterons également que l'absence d'une relation linéaire ne signifie pas l'absence de toute autre type de relation entre les deux variables étudiées.

```
> list <- c("Calories","Total.Fat","Cholesterol","Sodium","Sugars
   ","Protein")

> round(cor(data_macdo[, list]),2)
            Calories Total.Fat Cholesterol Sodium Sugars Protein
Calories        1.00      0.90        0.60   0.71   0.26    0.79
Total.Fat       0.90      1.00        0.68   0.85  -0.12    0.81
Cholesterol     0.60      0.68        1.00   0.62  -0.14    0.56
Sodium          0.71      0.85        0.62   1.00  -0.43    0.87
Sugars          0.26     -0.12       -0.14  -0.43   1.00   -0.18
Protein         0.79      0.81        0.56   0.87  -0.18    1.00
```

À vous !

5. Déterminez deux groupes d'attributs qui présentent des corrélations linéaires entre eux ($r > 0,5$).
6. Justifiez l'utilisation d'une ACP.
7. Expliquez les différences obtenues entre une ACP normée et non normée ?

1.3 Représentation en trois dimensions

Chargez le package `rgl`.
Faites une représentation en trois dimensions des attributs `Calories`, `Total.Fat`, `Cholesterol`. Voici les lignes de commande qui pourront vous aider à faire cette représentation en 3D.

```
## Representation en 3D des trois variables Calories, Total.Fat,
   Cholesterol
> library(rgl)
> plot3d(data_macdo$Calories,data_macdo$Total.Fat,
   data_macdo$Cholesterol,
type="s")
```

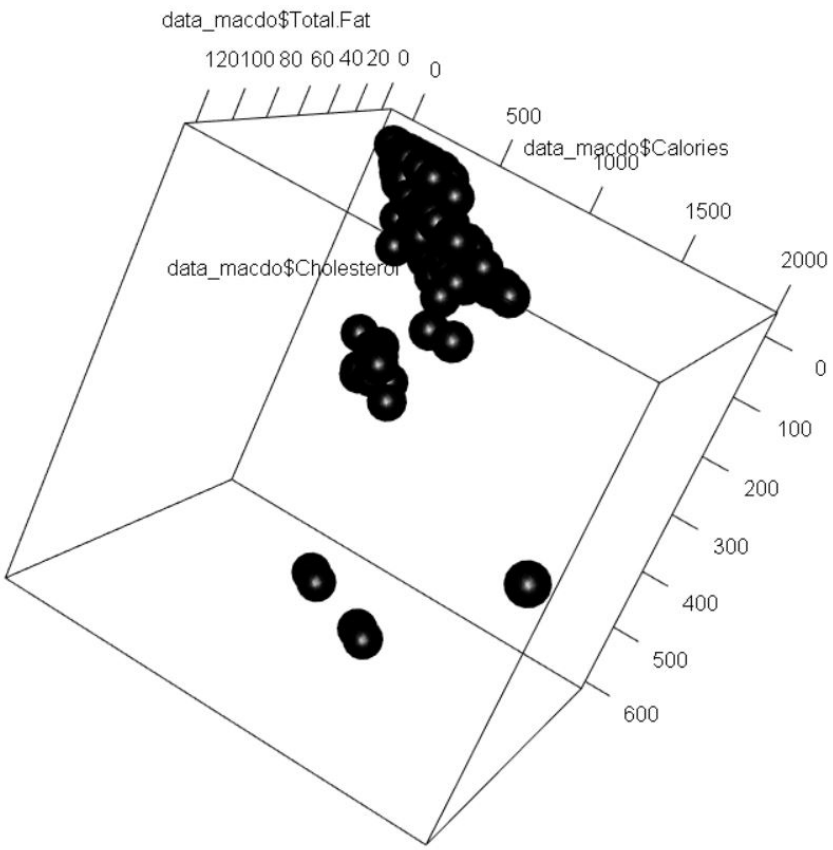


FIGURE 2 – Valeurs : Calories, Total.Fat, Cholesterol

La fonction `scale()` permet de centrer les données puis divise les données centrées par l'écart-type.
Appliquez cette fonction sur votre `dataframe`.

```
> list <- c("Calories", "Total.Fat", "Cholesterol")
> data_macdo.cr <- scale(data_macdo[, list])
> lims <- c(min(data_macdo.cr),max(data_macdo.cr))
> plot3d(data_macdo.cr, type = "s", xlim = lims, ylim = lims,zlim =
    lims)
```

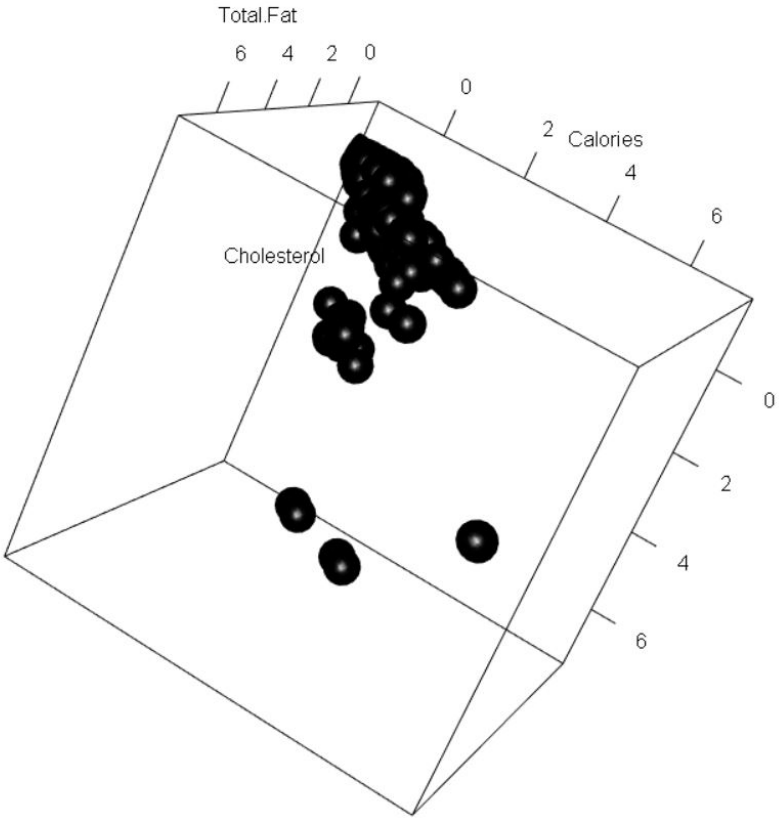


FIGURE 3 – Plot3D après la fonction `scale()` : Calories, Total.Fat, Cholesterol

À vous !

8. Quelles différences voyez-vous entre ce graphique et le plot 3D d'avant ?

La fonction `ellipse3d()` permet de représenter une ellipse de concentration. L'ellipse de concentration d'un sous–nuage de points est l'ellipse d'inertie telle qu'une distribution uniforme à l'intérieur de l'ellipse a une variance égale à celle du sous–nuage.

```
> data_macdo.cr_df <- as.data.frame(data_macdo.cr)
> plot3d(data_macdo.cr, type = "s", xlim = lims, ylim = lims,zlim =
    lims)
> plot3d(ellipse3d(cor(cbind(data_macdo.cr_df$Calories, data_macdo.
   cr_df$Total.Fat,data_macdo.cr_df$Cholesterol))),
       col="grey",add=TRUE)
```

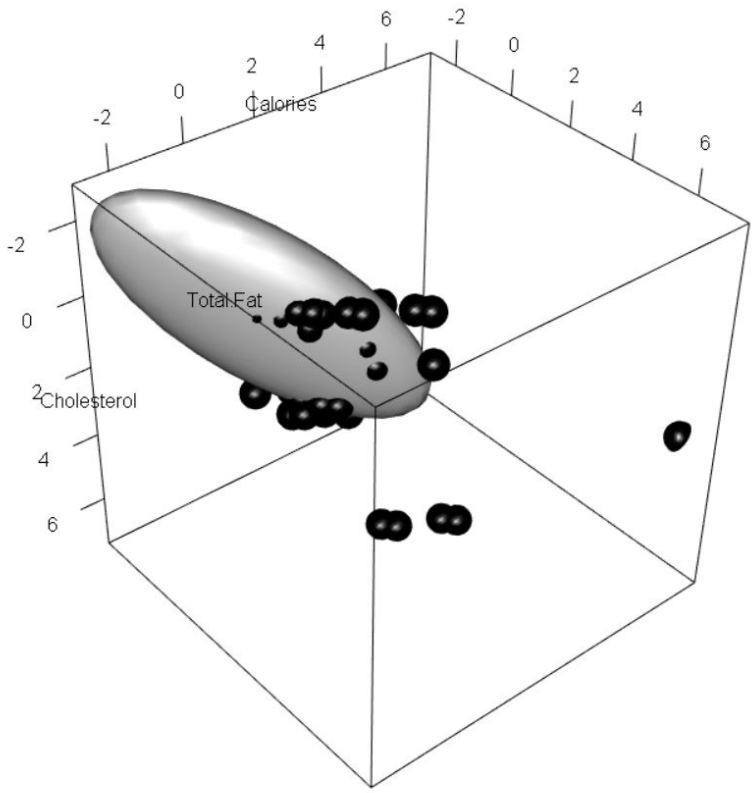


FIGURE 4 – Ellipse de corrélation linéaire : Calories, Total.Fat, Cholesterol

À vous !

9. Commentez la répartition des points dans l'ellipse.
10. Affichez l'ellipse de corrélation linéaire dans la représentation en 3D pour les attributs *Sodium*, *Sugars* et *Protein*.
11. Expliquez les différences entre les ellipses obtenues dans les deux nuages.

1.4 Analyse en Composantes Principales

Le package `ade4` permet de réaliser une ACP. Il est téléchargable sur : `https://cran.r-project.org/web/packages/ade4/index.html`. D'autre part, il est possible de télécharger le package `FactomineR` qui permet également de faire des ACP. Utiliser la fonction *dudi.pca()* le package ade4 pour exécuter une ACP centrée réduite :

```
> library(ade4)
> list <- c("Calories", "Total.Fat", "Cholesterol")
> acp <- dudi.pca(data_macdo[, list], center=TRUE, scale=TRUE,
   scannf = FALSE, nf = 3)
> names(acp)
```

```
 [1] "tab"  "cw"   "lw"   "eig"  "rank" "nf"   "c1"   "li"   "co"
      "l1"   "call" "cent" "norm"
```

À vous !

12. Que contient le `dataframe tab` ?
13. Comparez avec le tableau de données `data_macdo.cr`, expliquez la légère différence.
14. Quelle manipulation devez-vous réaliser pour retrouver exactement le tableau utilisé dans `dudi.pca()` ?

Le vecteur `cw` donne le poids des colonnes (*column weight)*, c'est-à-dire le poids de chaque variable. Par défaut, chaque variable a un poids égal à 1.

```
> acp$cw
[1] 1 1 1
```

Le vecteur `lw` donne le poids des lignes (*line weight*), c'est-à-dire le poids de chaque individu. Par défaut, chaque individu a un poids égal à $1/n$.

```
 head(acp$lw)
[1] 0.003846 0.003846 0.003846 0.003846 0.003846 0.003846
head(acp$lw)*nrow(data_macdo)
[1] 1 1 1 1 1 1
```

Les valeurs propres renseignent la part de l'inertie totale prise en compte par chaque axe.

```
> (pve <- 100*acp$eig/sum(acp$eig))
[1] 60.944152 23.661817 13.075336  2.318694
> pve <- 100*acp$eig/sum(acp$eig)
> cumsum(pve)
[1]  82.08  97.07 100.00
```

1.5 Informations associées à une ACP

Dans l'exemple, le premier axe factoriel extrait 82,08 % de l'inertie totale, le deuxième axe factoriel 14,99 % de l'inertie totale. Le premier plan factoriel représente donc 97,07 % de l'inertie totale. Ceci signifie que lorsque nous projetons le nuage de points initial dans le plan défini par les deux premiers axes factoriels, il y a peu de perte d'informations.

À vous !

15. Quel pourcentage de l'inertie total avec 3 axes ?
16. Cherchez la signification du vecteur rank.
17. Cherchez la signification du vecteur nf.
18. Cherchez la signification du vecteur c1.
19. Cherchez la signification du vecteur l1.

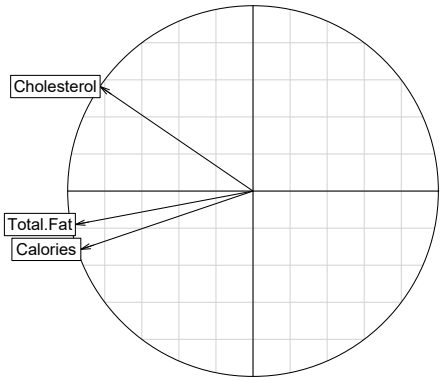
20. Cherchez la signification du vecteur co.
21. Cherchez la signification de l'objet call.
22. Cherchez la signification du vecteur cent.
23. Cherchez la signification du vecteur norm.
24. Donnez le nombre de facteurs retenus.

1.6 Analyse des variables

Observez les attributs de notre `data.frame` sur trois axes obtenus par l'ACP. La représentation des attributs se fait à travers un cercle de corrélation linéaire et nous pouvons voir aisément la proximité des attributs dans le cercle.

```
> inertie <-inertia.dudi(acp, col.inertia=TRUE)
> inertie
Inertia information:
Call: inertia.dudi(x = acp, col.inertia = TRUE)

Decomposition of total inertia:
    inertia     cum  cum(%)
Ax1 2.46246   2.462   82.08
Ax2 0.44959   2.912   97.07
Ax3 0.08795   3.000  100.00

Column contributions (%):
   Calories   Total.Fat Cholesterol
      33.33       33.33       33.33

Column absolute contributions (%):
            Axis1(%) Axis2(%) Axis3(%)
Calories      35.06   22.006   42.935
Total.Fat     37.31    7.245   55.448
Cholesterol   27.63   70.748    1.617

Signed column relative contributions:
              Axis1   Axis2   Axis3
Calories     -86.33  -9.894  3.7760
Total.Fat    -91.87  -3.257 -4.8765
Cholesterol  -68.05  31.808  0.1422

Cumulative sum of column relative contributions (%):
              Axis1 Axis1:2 Axis1:3             Axis4:3
Calories      86.33   96.22     100 -0.0000000000001332
Total.Fat     91.87   95.12     100 -0.0000000000001110
Cholesterol   68.05   99.86     100 -0.0000000000001332
> # Coordonnees des attributs
> round(acp$co,2)
            Comp1 Comp2 Comp3
Calories    -0.93 -0.31  0.19
Total.Fat   -0.96 -0.18 -0.22
Cholesterol -0.82  0.56  0.04

>s.corcircle(acp$co,xax=1,yax=2)
```

FIGURE 5 – Cercle des corrélations linéaires



À vous !

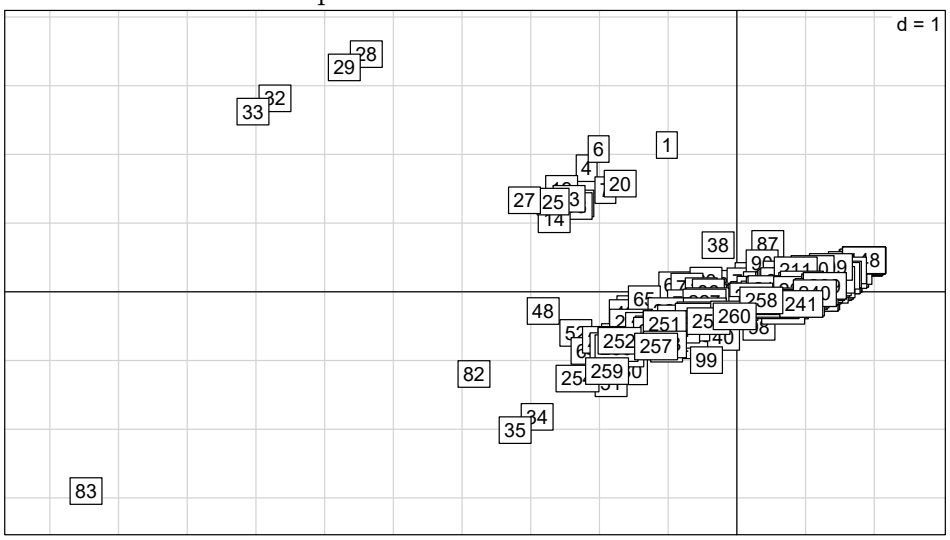
25. Comment reconnaissez-vous sur la figure qu'un attribut est bien représenté ?
26. Quel est l'attribut le moins bien représenté dans le cercle ? Justifiez votre réponse.
27. À l'aide de la figure précédente (figure 4), précisez l'attribut le plus corrélé positivement à *Calorie* ?
28. Quels sont les attributs qui ont contribué à l'axe F1 ? Justifiez votre réponse.
29. Donnez une signification à cet axe.
30. Quels sont les attributs qui ont contribué à l'axe F2 ? Justifiez votre réponse.
31. Donnez une signification à cet axe.

1.7 Conclusion

La fonction `s.label()` permet de représenter les individus sur les différents plans factoriels, par exemple sur le premier plan factoriel :

```
s.label(acp$li, xax = 1, yax = 2)
```

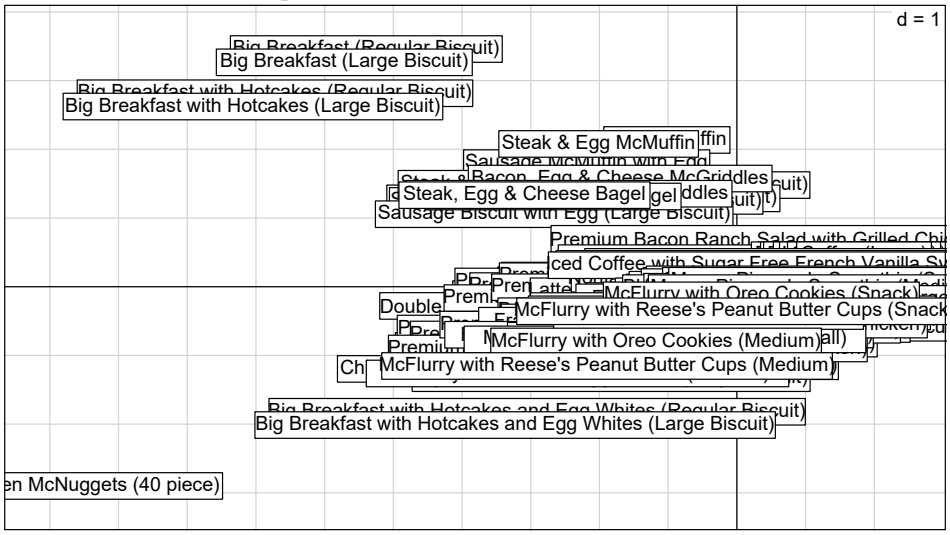
FIGURE 6 – Représentation des individus avec l'ACP.



Afin de bien interpréter les données, il est préférable d'utiliser comme étiquette d'un produit son `Item`.

```
s.label(acp$li, xax = 1, yax = 2, label=as.character(
    data_macdo$Item), clabel=1.5)
```

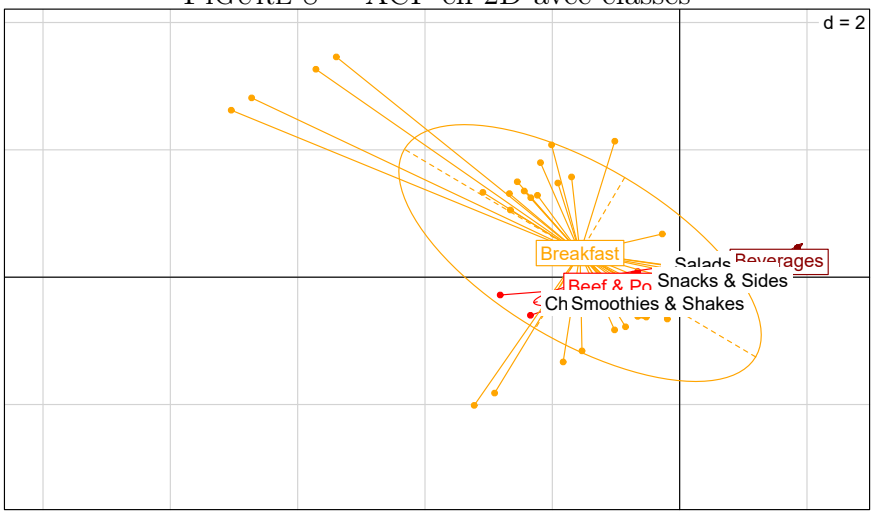
FIGURE 7 – Représentation labellisé des cidres avec l'ACP.



La fonction `s.class()` permet de porter en information supplémentaire une variable qualitative définissant des groupes d'individus, par exemple :

```
gcol <- c("red1", "red4","orange")
s.class(dfxy = acp$li, fac = data_macdo$Category, col = gcol, xax =
    1, yax = 2)
```

FIGURE 8 – ACP en 2D avec classes



À vous !

32. Utilisez la fonction `scatter(acp)`.
33. Reprenez l'analyse à partir de la section 1.4 mais en incluant les variables `Sodium`, `Sugars` et `Protein`.
34. Concluez sur le jeu de données. Iriez-vous prendre votre petit déjeuner chez MacDonald's ? Pourquoi ?