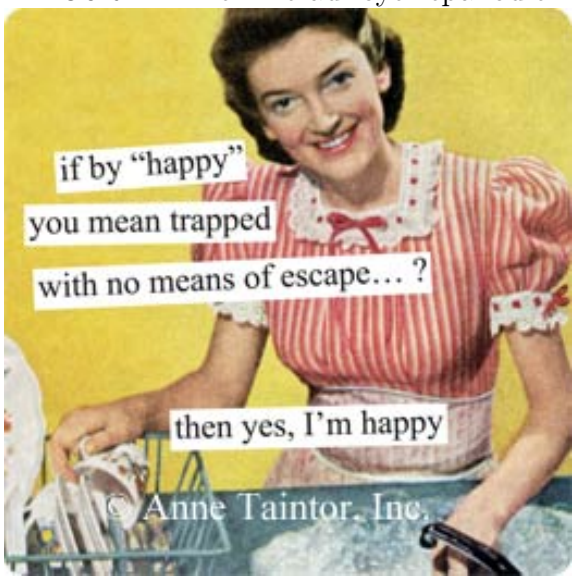
T. D. n° 2
Analyse Factorielle des Correspondances et Analyse des Correspondances Multiples

Résumé

Ce document est le TD n° 2 du module Analyse exploratoire. Il reprend rapidement des éléments du cours et propose une mise en pratique interactive de l'AFC et de l'ACM.

1 Tâches ménagères

FIGURE 1 – Femme au foyer épanouie



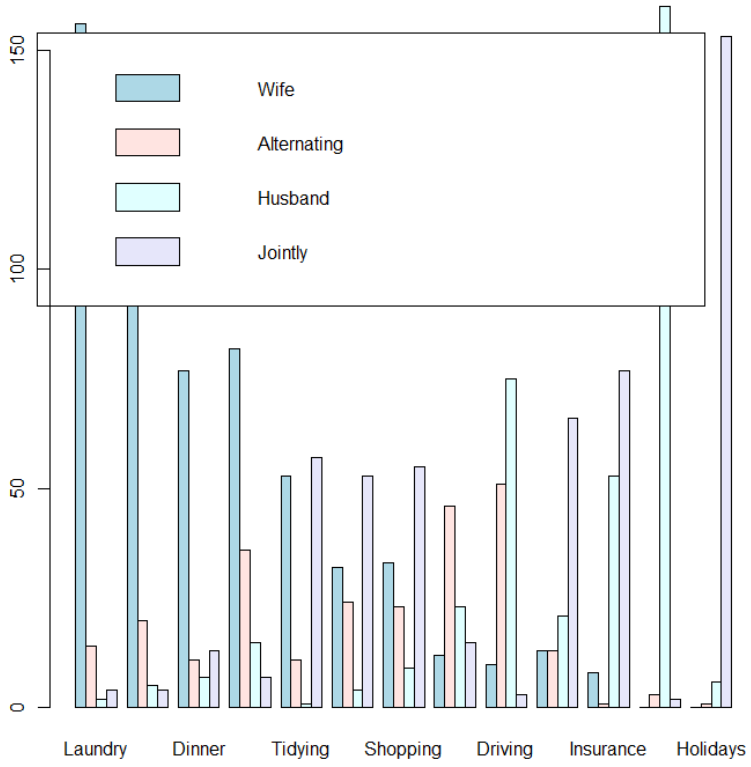
source :`http://www.annetaintor.com/`

1.1 Chargement des données

Après avoir téléchargé le fichier *afc_ tache_ menageres.csv* dans R. Associez ces données à un dataframe et appliquez la fonction *summary()*. Il est également possible de télécharger le fichier *afc_ taches_ menageres.csv* depuis le dataframe factoextra : `https://cran.r-project.org/web/packages/factoextra/factoextra.pdf`.
À partir de votre dataframe *DataP* représentez par un diagramme en bâtons les nombres de voix obtenues pour chaque partenaire (Épouse/Époux) selon les différentes tâches ménagères.

```
> dataP<- read.csv('C:/Users/claey/Documents/cour/My TD/TD2/
    afc_tache_menageres.csv', sep =';')
> barplot(t(dataP[,-1]),beside=T,names=dataP$Task, col = c("
    lightblue", "mistyrose","lightcyan", "lavender"),legend =
    colnames(dataP[,2:5]))
```

FIGURE 2 – Répartitions des tâches ménagères



À vous !

a) Commentez ces résultats.

b) Réalisez le test du chi2 permettant d'étudier le lien de dépendance entre les 13 tâches ménagères et les personnes affectées à ces tâches.

c) Concluez sur le résultat du test du chi2.

d) Quels sont les prérequis pour faire une analyse factorielle des correspondances ?

1.2 AFC

FactoMineR est un package **R** dédié à l'analyse exploratoire multidimensionnelle de données (à la française). *FactoMineR* présente de nombreux avantages :

— il permet de réaliser des analyses classiques telles que l'analyse en composantes principales (ACP), l'analyse des correspondances (AC), l'analyse des correspondances multiples (ACM) ainsi que des analyses plus avancées.
— il permet l'ajout d'information supplémentaire telle que des individus et/ou des variables supplémentaires.
— il fournit un point de vue géométrique et de nombreuses sorties graphiques.
— il fournit de nombreuses aides à l'interprétation (description automatique des axes, nombreux indicateurs, ...).
— il peut prendre en compte diverses structures sur les données (structure sur les variables, hiérarchie sur les variables, structure sur les individus).
— une interface graphique est disponible.

Effectuez l'analyse factorielle des correspondances (AFC) du tableau à l'aide du package *FactoMineR* que vous aurez installé. Faites les choix suivants :

— lignes actives = les 13 tâches ménagères listées
— colonnes actives = les 4 possibilités

Créez un diagramme en bâtons pour étudier la décroissance de l'inertie des axes.

```
> install.packages("FactoMineR")
> library(FactoMineR)
> res_exo2 = CA(dataP[,2:5])
> ###voir Figure 3###
> barplot (res_exo2$eig[,2], names=paste("Dim",1:length(
   res_exo2$eig[,2])), main="Inertie
         (en %) des axes factoriels", col="orange", border="white")
> ###voir Figure 4###
```

FIGURE 3 – AFC des tâches ménagères

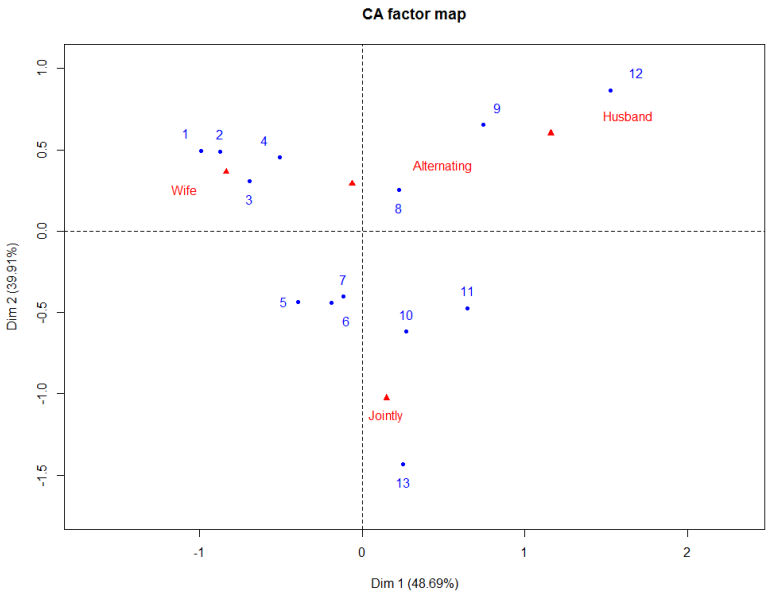
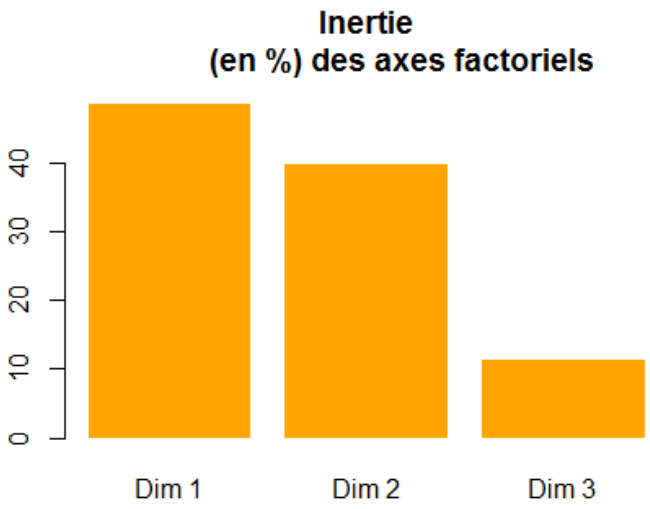


FIGURE 4 – Inertie des axes



La Figure 4 met en évidence deux axes prédominants, cumulant environ 85% de l'inertie. Comme en A.C.P, le premier axe est celui qui restitue la plus grande quantité d'inertie ; le deuxième est celui qui, tout en étant orthogonal au premier (au sens de la métrique du khi-deux), en restitue aussi le maximum, et ainsi de suite. Vous pouvez également retenir les axes cumulant au moins 80% de l'inertie. Vous observez souvent de fortes valeurs propres au départ puis ensuite de faibles valeurs avec un décrochage dans le diagramme. Retenez les axes avant le décrochage. (Cette technique s'appelle la règle du coude). Vos pouvez retrouver l'inertie totale de deux façons, la première se fait à partir du test du Chi2. L'inertie en AFC est égale au Phi2, c'est-à-dire la valeur statistique du test du Chi2 divisée par l'effectif total du tableau de contingence :

```
> resu.chi2$statistic/sum(dataP[1:13,2:5])
X-squared
  1.11494
```

À vous !

a) Affichez l'AFC avec les tâches à la place des numéros.

b) Commentez les résultats de ce nouveau graphique.

c) Retrouvez l'inertie totale en calculant la somme des inerties de tous les axes factoriels issus de l'AFC.

d) Calculez le V de Cramer.

e) Expliquez l'intérêt de cette valeur.

f) Vérifiez cette valeur à l'aide de la fonction *cramer.v()* du package *questionr* (que vous devrez installer).

1.3 ACM

Calculez la valeur propre moyenne. Déduisez-en le nombre d'axes dont l'inertie est supérieure à l'inertie moyenne par axe. Le nombre d'axes étant donné par min(ligne, colonne)-1 =3, l'inertie moyenne est calculée par :

```
> sum(res_exo2$eig[,1])/3
[1] 0.3716468
```

Or, les inerties des axes sont données par :

```
res_exo2$eig[,1]
[1] 0.5428893 0.4450028 0.1270484
```

Seules les deux premières dimensions ont donc une inertie supérieure à la moyenne.

```
> res_exo2$eig
#         eigenvalue               percentage of variance
                percentage of variance cumulative
dim 1   0.5428893                48.69222
                                   48.69222
dim 2   0.4450028                39.91269
                                   88.60491
dim 3   0.1270484                11.39509
                                   100.00000
```

En observant les *eigen value*, seules les deux premières dimensions sont donc potentiellement intéressantes. Le pourcentage de la variance cumulée est en effet de 88% ce qui est suffisant. Notez que par défaut, *FactoMineR* sort les deux premiers axes.

À vous !

a) Représentez l'AFC avec la troisième dimension.

b) Affichez les coordonnées des modalités colonnes.

c) Affichez la qualité de projection des modalités colonnes.

d) Affichez la contribution des modalités colonnes aux axes factoriels.

e) Affichez les coordonnées des modalités lignes dans le plan factoriel.

f) Affichez la qualité de représentation des modalités lignes.

g) Affichez les contribution des modalités lignes aux axes factoriels.

2 Pokemon

FIGURE 5 – Chasseur de pokemon douteux

source :`https://elliemaloney.wordpress.com`

2.1 Chargement des données

Commencez par charger les données du fichier *pokemon.csv*. Associez ces données à un dataframe, créez un sous ensemble composé exclusivement des informations suivantes : Type_1, génération et légendaire. Transformez la colonne légendaire en type facteur, appliquez la fonction *summary()*. Il est également possible de télécharger le fichier *pokemon.csv* depuis la plate-forme Kaggle : `https://www.kaggle.com/secareanualin/football-events`.

```
> library(ade4)
> library(adegraphics)
> poke <- read.csv('C:/Users/claey/Documents/cour/My TD/TD2/pokemon
   .csv',  na.strings=c("","NA"), sep =',')
> poke <- as.data.frame(poke)
> poke$Generation <- as.factor(poke$Generation)
> poke.x <- poke[,c(3,12,13)]
> summary(poke)
       X.                                  Name
 Min.   :  1.0   Abomasnow                  :  1
 1st Qu.:184.8   AbomasnowMega Abomasnow:  1
 Median :364.5   Abra                       :  1
 Mean   :362.8   Absol                      :  1
 3rd Qu.:539.2   AbsolMega Absol            :  1
 Max.   :721.0   Accelgor                   :  1
                 (Other)                    :794
    Type_1          Type_2        Total
 Water  :112   Flying  : 97   Min.   :180.0
 Normal : 98   Ground  : 35   1st Qu.:330.0
 Grass  : 70   Poison  : 34   Median :450.0
 Bug    : 69   Psychic : 33   Mean   :435.1
 Psychic: 57   Fighting: 26   3rd Qu.:515.0
 Fire   : 52   (Other) :189   Max.   :780.0
 (Other):342   NA's    :386
       HP              Attack       Defense
```

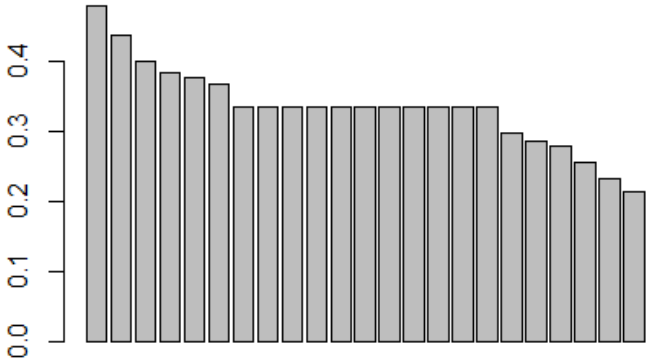
```
Min.   :  1.00   Min.   :  5   Min.   :  5.00
1st Qu.: 50.00   1st Qu.: 55   1st Qu.: 50.00
Median : 65.00   Median : 75   Median : 70.00
Mean   : 69.26   Mean   : 79   Mean   : 73.84
3rd Qu.: 80.00   3rd Qu.:100   3rd Qu.: 90.00
Max.   :255.00   Max.   :190   Max.   :230.00

   Sp..Atk          Sp..Def          Speed
Min.   : 10.00   Min.   : 20.0   Min.   :  5.00
1st Qu.: 49.75   1st Qu.: 50.0   1st Qu.: 45.00
Median : 65.00   Median : 70.0   Median : 65.00
Mean   : 72.82   Mean   : 71.9   Mean   : 68.28
3rd Qu.: 95.00   3rd Qu.: 90.0   3rd Qu.: 90.00
Max.   :194.00   Max.   :230.0   Max.   :180.00

  Generation    Legendary
Min.   :1.000   False:735
1st Qu.:2.000   True : 65
Median :3.000
Mean   :3.324
3rd Qu.:5.000
Max.   :6.000
```

2.2 ACM avec ade4

A l'aide de la librairie *ade4* et *adegraphics*, appliquez la fonction *dudi.acm()* à votre sous jeu de données. Affichez les valeurs propres.

```
> install.packages("ade4")
> library(ade4)
> acmtot <- dudi.acm(poke.x,scannf=FALSE)
> barplot(acmtot$eig)
```

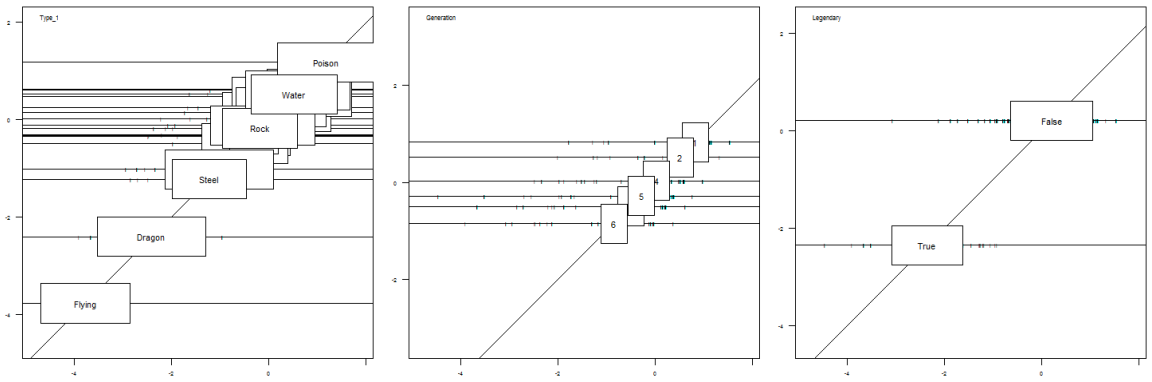
FIGURE 6 – Valeurs propres de l' ACM



Le nombre important des valeurs propres (liées, non aux variables mais aux modalités de ces variables) ne permet pas d'énoncer un critères de sélection du nombre de facteurs à conserver. Conservez 5 valeurs propres mais ne détaillez dans le TD que les deux premières. Vous pouvez cependant regarder les facteurs 3 et 4 et 5.
La fonction *score()* permet de visualiser les variables qualitatives avec un facteur. Pour chaque variable, les individus sont positionnés sur l'axe des abscisses par leur score sur l'axe factoriel considéré, et sur l'axe des ordonnées par le score de la modalité qu'ils portent. Le score d'une modalité est la moyenne des scores des individus portant cette modalité, ce qui est mis en évidence par la première bissectrice.

```
> score(acmtot,xax=1)
```

FIGURE 7 – Variables qualitatives avec un facteur sur deux axes

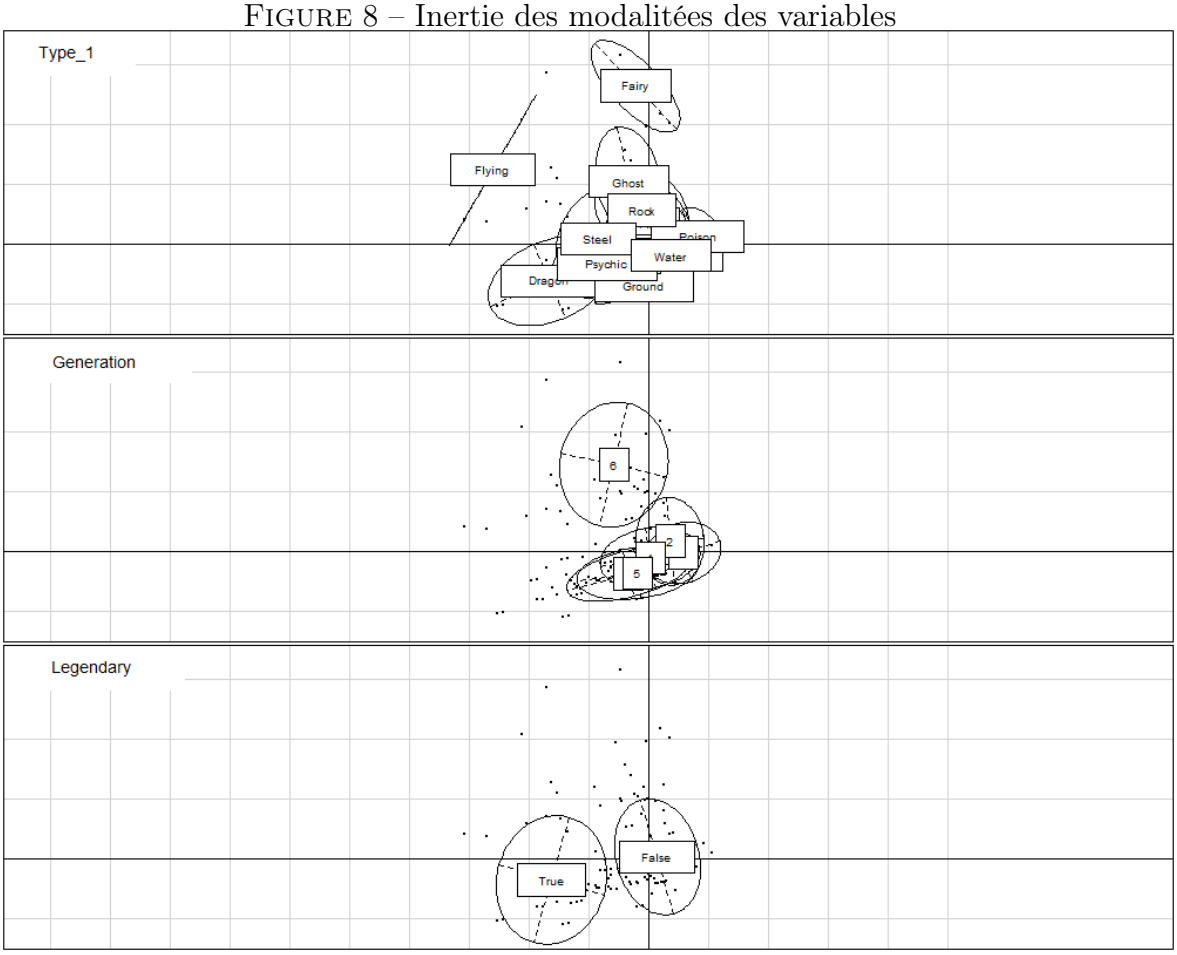


En gardant les quatre premiers facteurs, vous ne conservez que 16.06% de l'inertie

totale. Mais en pratique ce pourcentage est relativement courant. Vous pouvez représenter simultanément les individus et les modalités des variables sur un même graphique, démarche classique en analyse des données.

```
> head(inertia.dudi(acmtot)$TOT)
    inertia       cum       ratio
1 0.4780316 0.4780316 0.06235195
2 0.4365931 0.9146247 0.11929888
3 0.4003682 1.3149929 0.17152081
4 0.3831707 1.6981637 0.22149961
5 0.3750273 2.0731909 0.27041621
6 0.3670084 2.4401993 0.31828686

> scatter(acmtot)
```

FIGURE 8 – Inertie des modalitées des variables



À vous !

a) Affichez les rapports de corrélation pour le premier et deuxième facteur (en utilisant la liste de l'objet *acmtot*).

b) Quelles modalités décrivent le mieux chaque axe ?

c) À l'aide de la librarie *vcd*, utilisez la fonction *assocstats()* sur votre sous jeu de données (attention il faut le transformer en tableau de contingence avant). Commentez.

d) Affichez la matrice de corrélation sur les variables quantitatives suivantes : Attack, Defense, Sp..Atk, Sp..Def et Speed.

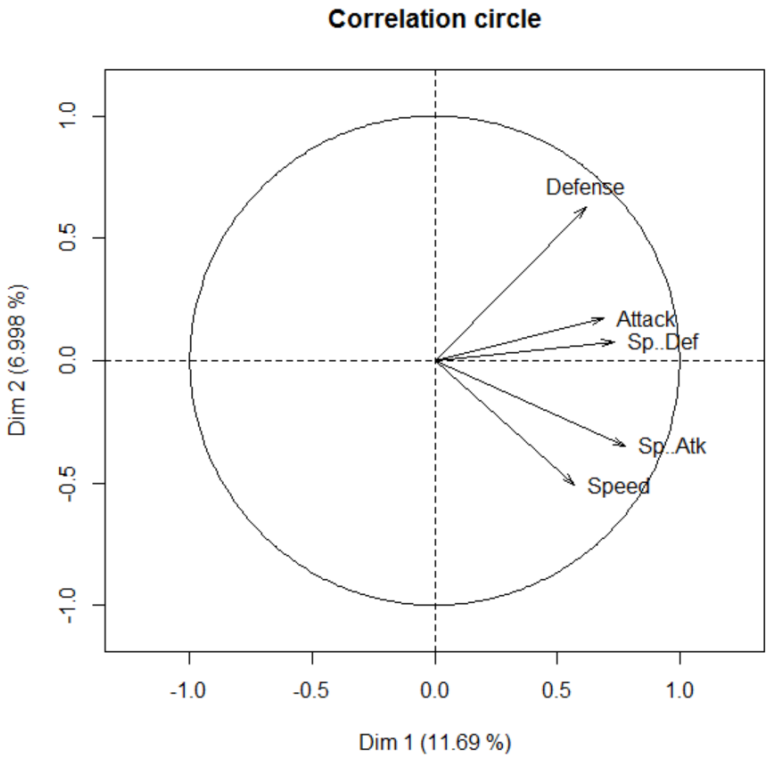
2.3 ACP mixte

Grâce à la librairie *PCAmixdata*, il est possible de réaliser une analyse en composantes principales sur un ensemble d'individus décris par un mélange de variables qualitatives et quantitatives. *PCAmix()* effectue une analyse ordinaire en composantes principales (ACP) et y associe une analyse de correspondance multiple (ACM). *PCAmix* utilise les rapports de corrélation au carré entre la variable qualitative et les composantes principales. Appliquez la fonction *PCAmix()* sur les variables quantitatives : Attack,Defense,Sp..Atk , Sp..Def et Speed ; et qualitatives : Type_1.

```
> install.packages("PCAmixdata")
> library(PCAmixdata)
> pcamix.temp<- PCAmix(subset(poke,select=c(7:11)) , subset(poke,
   select=c(3)))

#valeurs propres
> print(round(pcamix.temp$eig))
> print(round(pcamix.temp$eig))
       Eigenvalue Proportion Cumulative
dim 1           3         12         12
dim 2           2          7         19
dim 3           1          6         24
dim 4           1          5         29
dim 5           1          5         34
dim 6           1          5         38
dim 7           1          5         43
dim 8           1          5         47
dim 9           1          5         52
dim 10          1          5         56
dim 11          1          5         61
dim 12          1          5         65
dim 13          1          5         70
dim 14          1          5         75
dim 15          1          5         79
dim 16          1          5         84
dim 17          1          5         88
dim 18          1          4         92
dim 19          1          3         95
dim 20          0          2         97
dim 21          0          2         99
dim 22          0          1        100

> #correlations
> print(round(pcamix.temp$quanti.cor))
```

```
         dim1  dim2  dim3  dim4  dim5
Attack      1     0     0     0     0
Defense     1     1     0     0     0
Sp..Atk     1     0     0     0     0
Sp..Def     1     0     0     0     0
Speed       1    -1     0     0     0

#coord. des modalites dudi.mix de ADE4
> print(round(pcamix.temp$categ.coord))

> print(round(pcamix.temp$categ.coord))
          dim1  dim2  dim3  dim4  dim5
Bug         -1     0     0    -1     1
Dark         0     0     1     0     0
Dragon       1     0     1     1     1
Electric     0    -1     0    -2    -1
Fairy        0     0    -3     2     3
Fighting     0     0     2     2     1
Fire         0    -1     0     1    -1
Flying       1    -2     1    -3    -1
Ghost        0     0    -1     0     0
Grass        0     0    -1     1    -1
Ground       0     1     2     0    -1
Ice          0     0    -1     1     1
Normal      -1     0     1    -1     1
Poison       0     0     0     0     0
Psychic      1    -1    -1    -1     1
Rock         0     2     0     0     0
Steel        1     3    -1    -2    -1
Water        0     0     0     0    -1
```

FIGURE 9 – Cercle de corrélation de *pcamix.temp*



Pour une variable quantitative, les rapport de corrélation au carré sont la corrélation au carré entre la variable et les composantes principales.

FIGURE 10 – Rapports de corrélation de *pcamix.temp*

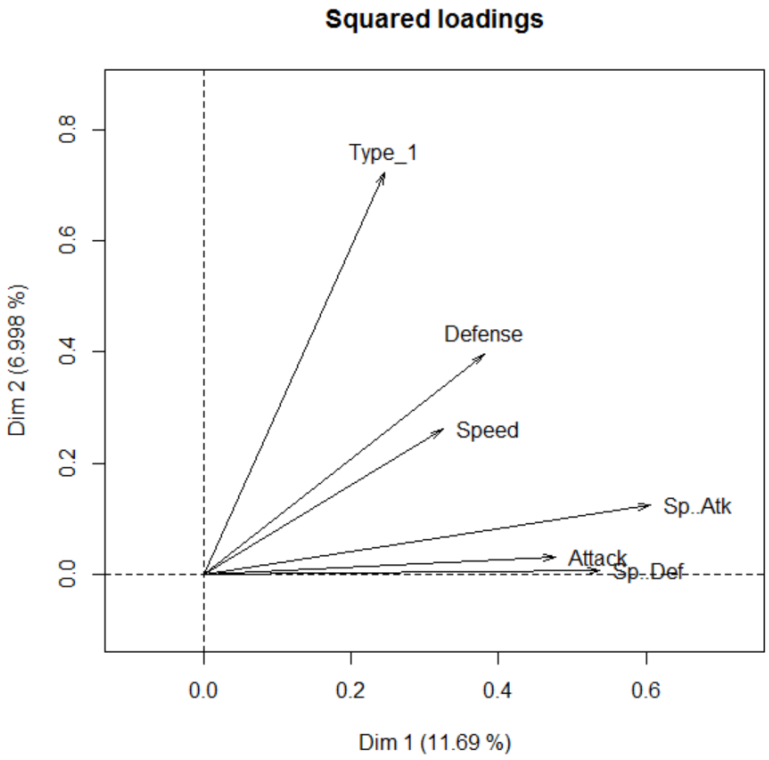
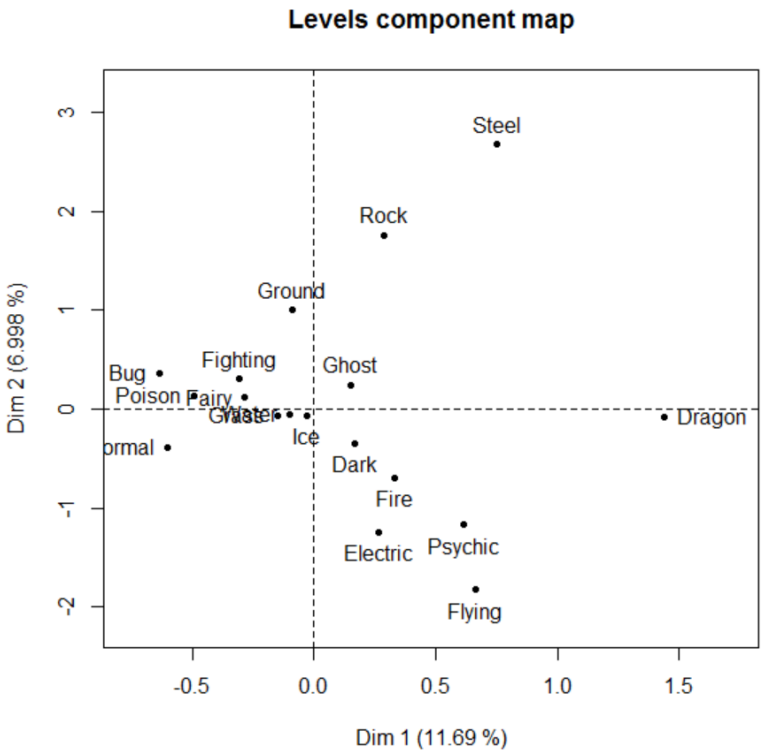


FIGURE 11 – Carte des composante de *pcamix.temp*



À vous !

a) Réalisez cette ACP mixe en remplaçant le *Type_1* par le nom des pokemons.

b) Observez et commentez les coordonnées du pokemon Pikachu sur l'ACP mixte.

c) Appliquez la fonction *FAMD()* à votre dataframe *poke*. Observez le résultat avec la fonction *smmary()*.

d) Expliquez ce qu'est la fonction *FAMD()* et son utilité.