

# T. D. n° 3

## Analyse des Correspondances Multiples

### Résumé

Ce document est le T.D. n° 3 du module *Analyse exploratoire*. Il reprend rapidement des éléments du cours et propose une mise en pratique interactive de l'ACM.

Ce T.D. est issu de l'ouvrage « Modélisation statistique par la pratique avec R », F. Bertrand, E. Claeys, M. Maumy-Bertrand, Dunod, octobre 2019.

## 1 Pokemon

FIGURE 1 – Chasseur de pokemon douteux



source :<https://elliemaloney.wordpress.com>

### 1.1 Chargement des données et statistiques élémentaires

1. Téléchargez la bibliothèque `ade4` et la bibliothèque `adegraphics`.
2. Chargez les données du fichier `pokemon.csv` disponible à l'adresse <https://tinyurl.com/y4y6a86m>.<sup>1</sup>
3. Associez ce jeu de données à un jeu de données de type `data.frame` que vous appellerez `poke`.
4. Transformez la variable `Generation` en type `factor`.
5. Créez un sous jeu de données composé exclusivement des variables suivantes : `Type_1`, `Generation` et `Legendary`.

---

1. Il est également possible de télécharger le fichier `pokemon.csv` depuis la plate-forme Kaggle : <https://www.kaggle.com/secareanualin/football-events>.

6. Appliquez la fonction `summary()` au jeu de données `poke`, de type `data.frame`

Si vous n'arrivez pas à répondre à ces six questions, voici les lignes de commande qui vous permettraient d'y répondre :

```
> install.packages(c("ade4","adegraphics"))
> library(ade4)
> library(adegraphics)
> poke<-read.csv("https://tinyurl.com/y4y6a86m", na.strings=c("", "NA"))
> poke<-as.data.frame(poke)
> poke$Generation<-as.factor(poke$Generation)
> summary(poke)
> poke.x<-poke[,c(3,12,13)]
```

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	

## 1.2 ACM avec ade4

7. À l'aide de la bibliothèque `ade4` et de la bibliothèque `adegraphics`, appliquez la fonction `dudi.acm()` au jeu de données `poke.x`.
8. Sauriez-vous dire le nombre total de valeurs propres? Affichez les valeurs propres avec la fonction `fviz_screplot()` du package `ade4`.

Si vous n'arrivez pas à répondre à ces deux questions, voici les commandes qui vous permettraient d'y répondre :

```
> res.acm.poke<-dudi.acm(poke.x,scannf=FALSE)
> fviz_screplot(res.acm.poke)
> get_eig(res.acm.poke)
```

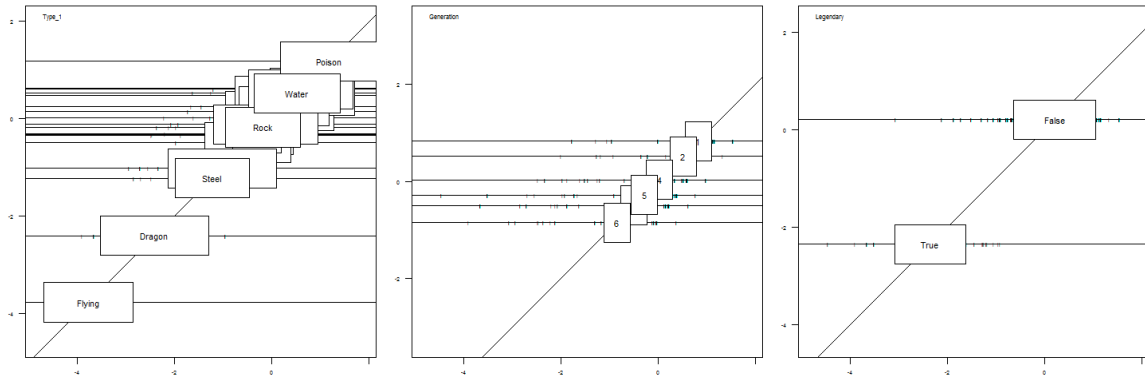
Nous remarquons qu'un nombre important de valeurs propres (lié, non pas aux facteurs mais aux modalités de ces facteurs) ne permet pas d'énoncer un critère de sélection du nombre d'axes à conserver.

Conservez six valeurs propres mais détaillez dans le compte-rendu du T.D. que les deux premières. Vous pouvez cependant regarder les axes 3, 4, 5 et 6.

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(res.acm.poke,xax=1)
```

FIGURE 2 – Les représentations graphiques des modalités des trois facteurs sur l'un des axes

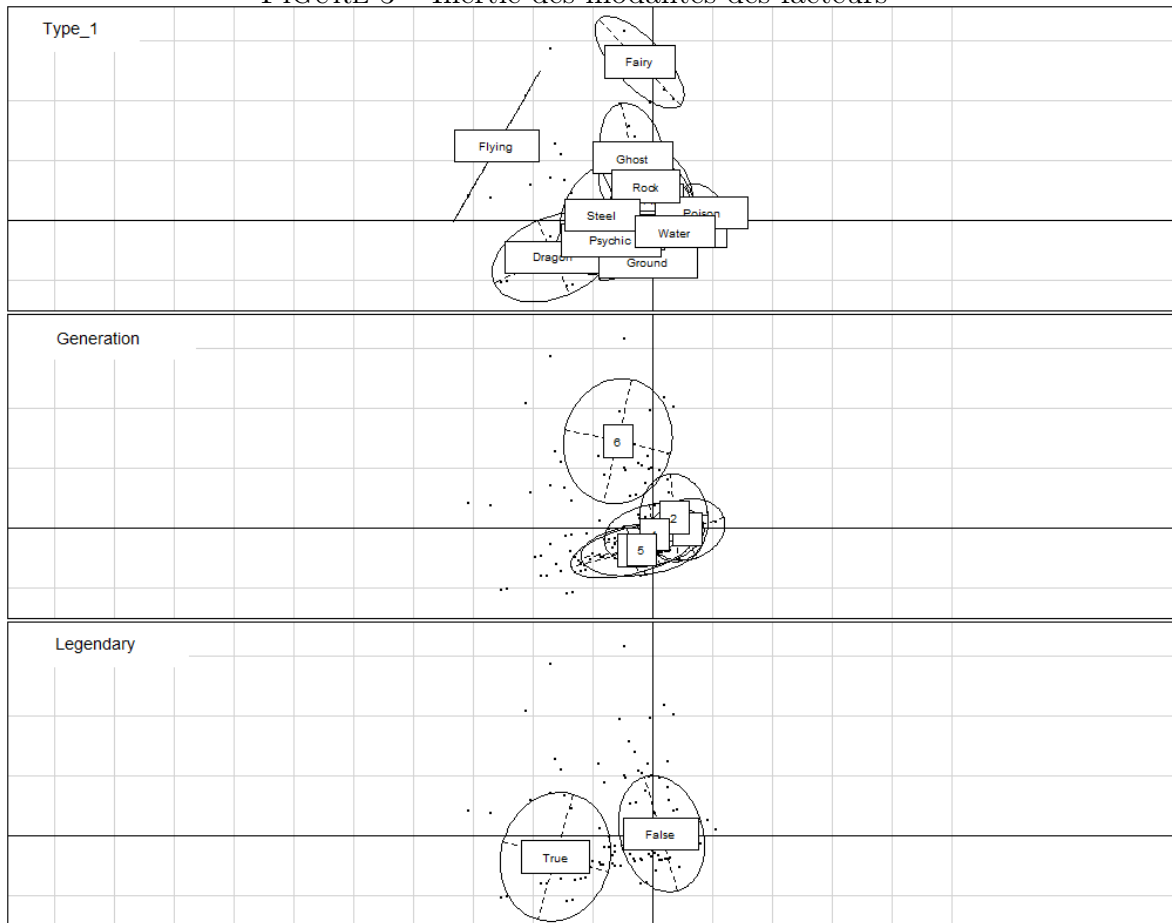


En gardant les six premiers facteurs, vous conservez moins de 50% de l'inertie totale. En pratique ce résultat est relativement courant. Vous pouvez représenter simultanément les individus et les modalités des facteurs sur un même graphique, démarche classique en analyse des données.

```
> head(inertia.dudi(res.acm.poke)$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(res.acm.poke)
```

FIGURE 3 – Inertie des modalités des facteurs



## C'est à vous !

9. Affichez et interprétez les rapports de corrélation pour le premier et le deuxième axe (en utilisant la liste de l'objet `res.acm.poke`).
10. Quelles modalités des trois facteurs décrivent le mieux le premier axe ?
11. À l'aide de la bibliothèque `vcd`, utilisez la fonction `assocstats()` sur votre sous jeu de données (attention il faut le transformer en tableau de contingence avant). Commentez.
12. Affichez la matrice de corrélation sur les variables quantitatives suivantes : Attack, Defense, Sp..Atk, Sp..Def et Speed.

## 1.3 ACP mixte

Grâce à la bibliothèque `PCAmixdata`, il est possible de réaliser une analyse en composantes principales sur un ensemble d'individus décrits par un mélange de variables quantitatives et qualitatives. `PCAmix()` effectue une analyse en composantes principales et y associe une analyse des correspondances multiples. `PCAmix()` utilise

les rapports de corrélation au carré entre la variable qualitative et les composantes principales.

### C'est à vous !

13. Téléchargez la bibliothèque `PCAmixdata` et chargez-là.
14. Appliquez la fonction `PCAmix()` sur les variables quantitatives suivante : `Attack, Defense, Sp..Atk, Sp..Def, Speed` et la variable qualitative : `Type_1`.
15. Affichez les valeurs propres
16. Affichez les corrélations des variables quantitatives suivantes : `Attack, Defense, Sp..Atk, Sp..Def, Speed`.  
Nous rappelons que pour une variable quantitative, les rapports de corrélation au carré sont la corrélation au carré entre la variable quantitative et les composantes principales.
17. Affichez les coordonnées des modalités de la variable qualitative : `Type_1`.

Si vous n'arrivez pas à répondre à ces cinq questions, voici les commandes qui vous permettraient d'y répondre.

```
> 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))
```

	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
>#coordonnees des modalites de la fonction dudi.mix() de ade4
> 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 4 – Cercle de corrélation de `pcamix.temp`

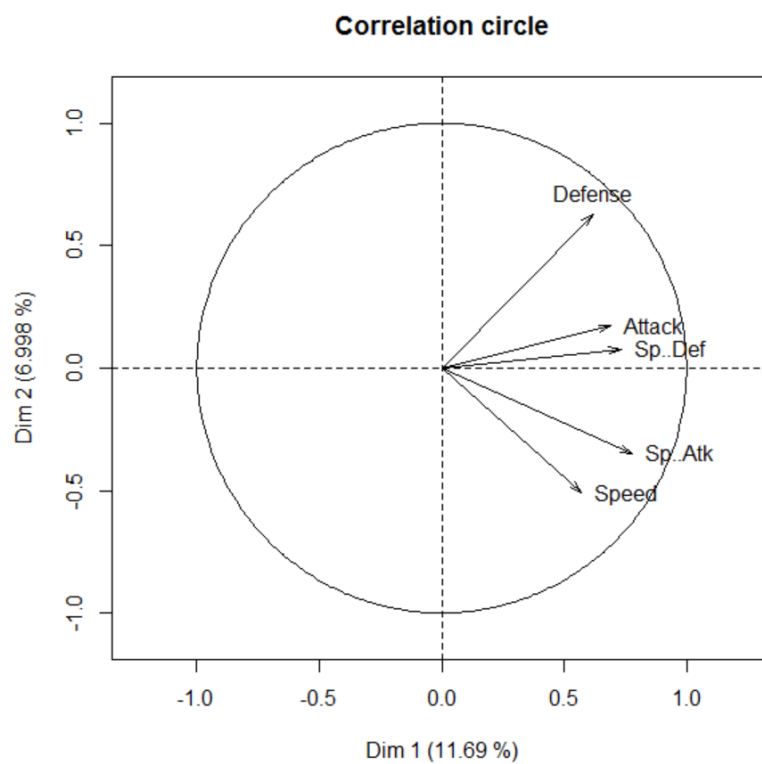




FIGURE 5 – Rapports de corrélation de pcamix.temp

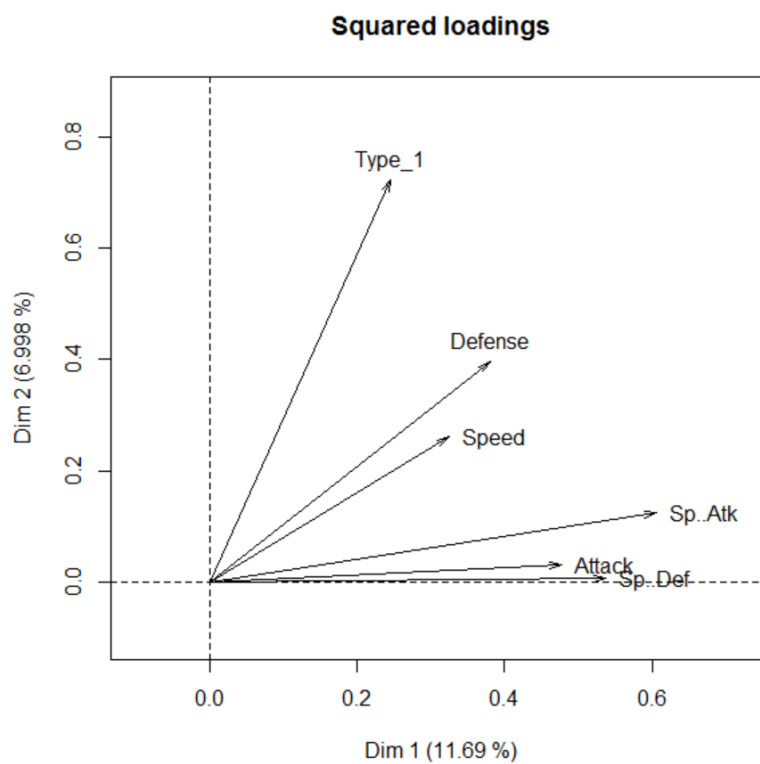
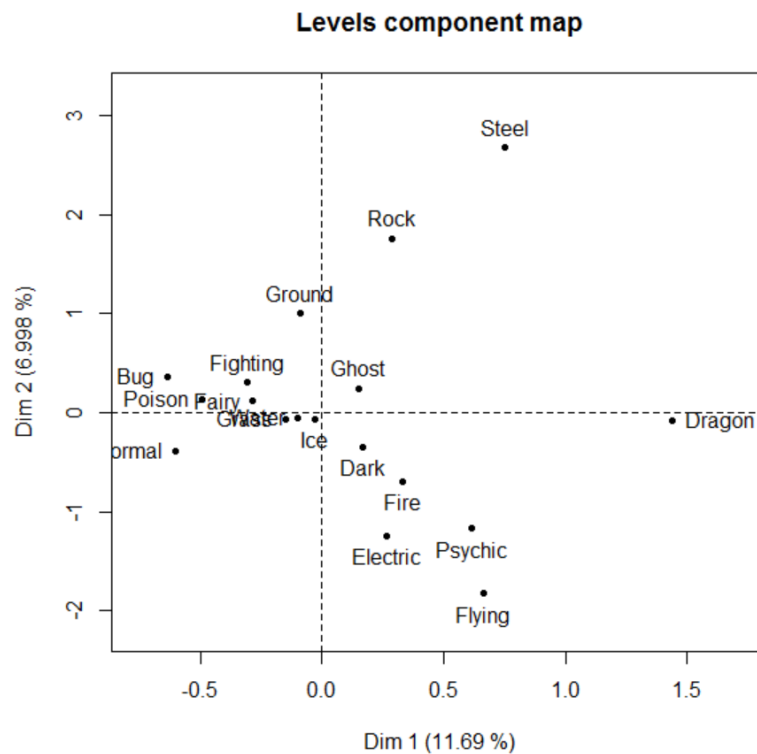


FIGURE 6 – Carte des composantes de `pcamix.temp`



### C'est à vous !

18. Réalisez cette A.C.P. mixte en remplaçant le facteur `Type_1` par le nom des *pokemons*.
19. Observez et commentez les coordonnées du *pokemon Pikachu* sur l'A.C.P. mixte.
20. Appliquez la fonction `FAMD()` au jeu de données `poke`. Observez le résultat avec la fonction `summary()`.
21. Expliquez ce qu'est la fonction `FAMD()` et son utilité.