ENICARTHAGE
المدرسة الوطنية للمهندسين بقرطاج
Ecole Nationale d'Ingénieurs de Carthage

Année Universitaire 2020-2021

Analyse de données

Analyse de données

Analyse Factorielle des Correspondances (AFC)

Analyse Factorielle des Correspondances (AFC)

L'analyse factorielle des correspondances (AFC) est une méthode exploratoire d'analyse des tableaux de contingences, c'est-à-dire aux tableaux de comptages obtenus par le croisement de deux variables nominales.

Le tableau de contingence suivant indique la répartition, en fonction de la marque et la finition de fabrication, des 1000 ordinateurs :

Variable en ligne : Marque de l'ordinateur
- HP : Hawlett-Packard
- ACER
- ASS : Assemblé

Variable en colonne : Finition de fabrication
- TB : Très Bien
- B : Bien
- M : Moyenne

Tableau de contingence en Effectif

Marque	Finition			
	TB	B	M	Total
HP	798	6	66	870
ACER	7	5	5	17
ASS	56	7	50	113
Total	861	18	121	1000

Analyse Factorielle des Correspondances (AFC)

Tableau de contingence en fréquences

Marque	Finition			
	TB	B	M	**Total**
HP	0,798	0,006	0,066	**0,87**
ACER	0,007	0,005	0,005	**0,017**
ASS	0,056	0,007	0,05	**0,113**
Total	**0,861**	**0,018**	**0,121**	**1**

Tableau de profils-lignes

Marque	Finition			
	TB	B	M	**Total**
HP	91,7	0,7	7,6	**100**
ACER	41,2	29,4	29,4	**100**
ASS	49,6	6,2	44,2	**100**
Profil-moyen	**86,1**	**1,8**	**12,1**	**100**

Tableau de contingence en pourcentages

Marque	Finition			
	TB	B	M	**Total**
HP	79,8%	0,6%	6,6%	87,0%
ACER	0,7%	0,5%	0,5%	1,7%
ASS	5,6%	0,7%	5,0%	11,3%
Total	**86,1%**	**1,8%**	**12,1%**	**100,0%**

Tableau de profils-colonnes

Marque	Finition			
	TB	B	M	**Profil-moyen**
HP	92,7	33,3	54,5	**87,0**
ACER	0,8	27,8	4,1	**1,7**
ASS	6,5	38,9	41,3	**11,3**
Total	**100**	**100**	**100**	**100**

Hypothèse d'indépendance :

Construisons le tableau de fréquences théoriques ($f_{i.} * f_{.j}$) sous l'hypothèse d'indépendance.

Tableau de fréquences empiriques

	TB	B	M	**Total**
HP	0,798	0,006	0,066	**0,87**
ACER	0,007	0,005	0,005	**0,017**
ASS	0,056	0,007	0,05	**0,113**
TOTAL	**0,861**	**0,018**	**0,121**	**1**

Tableau de fréquences théoriques

	TB	B	M	**Total**
HP	0,749	0,016	0,105	**0,870**
ACER	0,015	0,000	0,002	**0,017**
ASS	0,097	0,002	0,014	**0,113**
TOTAL	**0,861**	**0,018**	**0,121**	**1**

Naturellement, même sous l'hypothèse d'indépendance, une telle relation n'est qu'approximativement vraie. Le classique test deux χ^2 pour les tables de contingence permet précisément d'apprécier l'écart entre les lois empiriques f_{ij} et $f_{i.} * f_{.j}$

Hypothèse d'indépendance :

Le test de χ^2 est définit par :

H_0 : Les deux variables sont indépendantes

H_1 : Les deux variables sont dépendantes

La statistique du test est définie par :

$$d^2 = N.\sum_{i=1}^{n}\sum_{j=1}^{p}\frac{\left(f_{ij} - f_{i.}f_{.j}\right)^2}{f_{i.}f_{.j}} \qquad d^2 \rightarrow \chi^2(\nu)$$

En outre, le d^2 suit une loi du khi-2 de paramètre (s'appelle le nombre de degrés de liberté) avec :

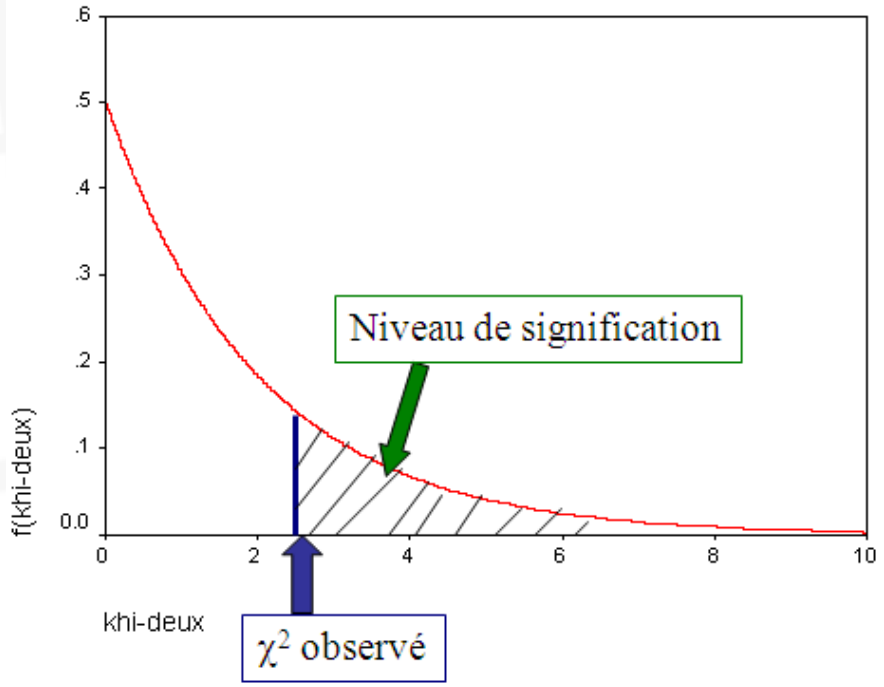
ν = (nombre de modalités de la première variable -1) x (nombre de modalités de la deuxième variable -1).

Hypothèse d'indépendance :

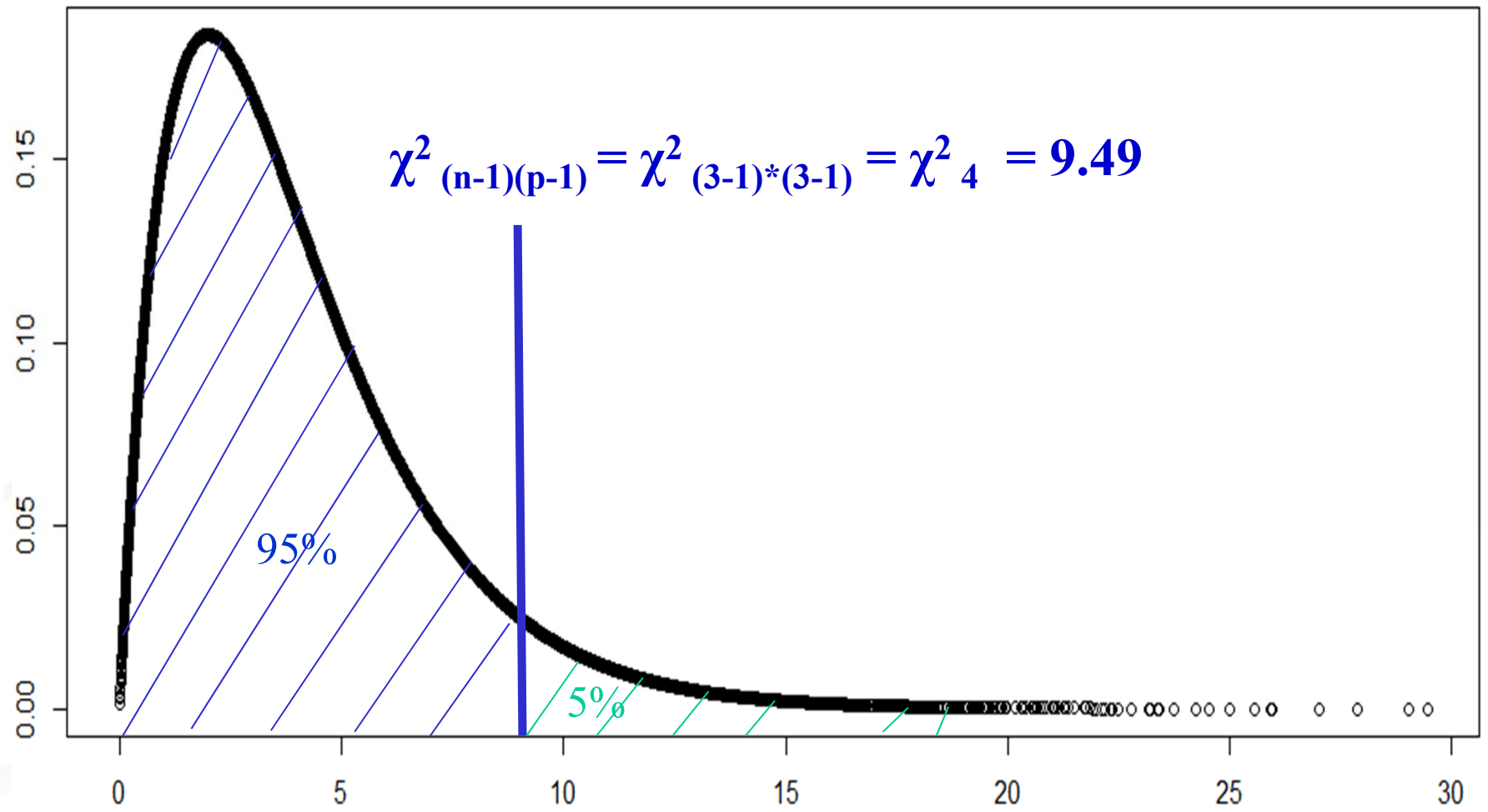
On rejettera donc l'hypothèse d'indépendance à un risque d'erreur α si $\mathbf{d^2}$ est supérieur à la valeur critique dans la table de χ^2 à (n-1)*(p-1) degré de liberté .

AN : $d^2 = 230.17$ $\chi^2_{(n-1)(p-1)} = \chi^2_{(3-1)*(3-1)} = \chi^2_4 = 9.49$

$d^2 >> \chi^2_4$ → on accepte H_1 → Les deux variables sont dépendantes



Hypothèse d'indépendance :



Distances entre profils. Métrique du χ^2

Pour remédier à cela, on pondère chaque écart par l'inverse de la masse de la colonne et l'on calcule une nouvelle distance appelée la distance du χ^2 :

$$d^2(i,i') = \sum_{j=1}^{p} \frac{1}{f_{.j}} \left(\frac{f_{ij}}{f_{i.}} - \frac{f_{i'j}}{f_{i'.}} \right)^2 = \sum_{j=1}^{p} \frac{1}{f_{.j}} \left(fl_{ij} - fl_{i'j} \right)^2$$

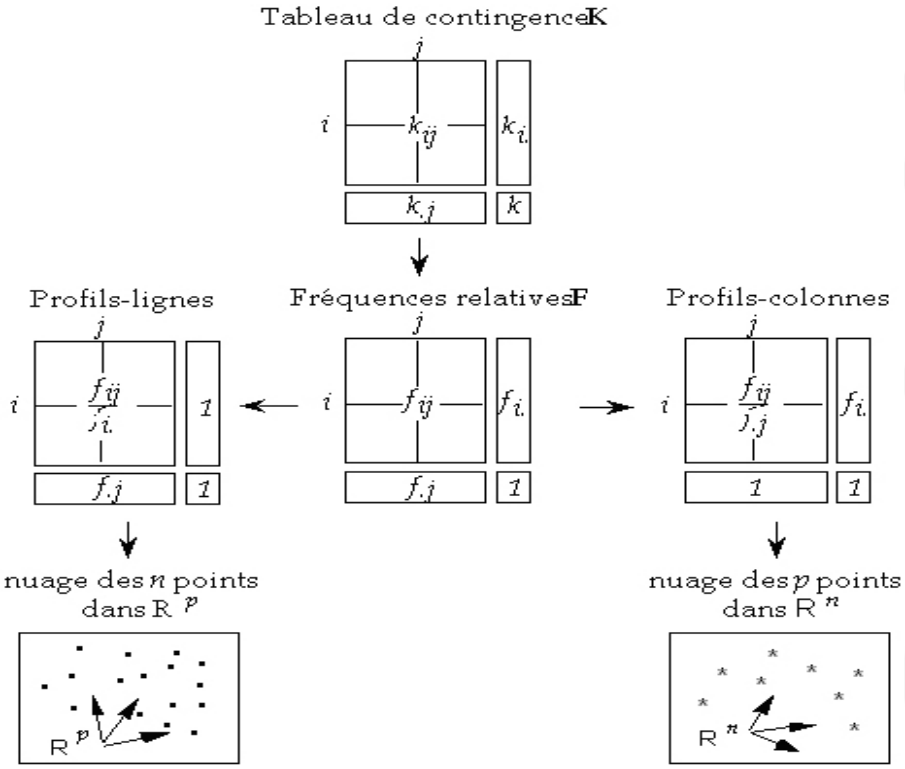
On définit de la même manière la distance entre les profils-colonnes par :

$$d^2(j,j') = \sum_{i=1}^{n} \frac{1}{f_{i.}} \left(\frac{f_{ij}}{f_{.j}} - \frac{f_{ij'}}{f_{.j'}} \right)^2 = \sum_{i=1}^{n} \frac{1}{f_{i.}} \left(fc_{ij} - fc_{ij'} \right)^2$$

AFC : Association entre les modalités

Construction des nuages

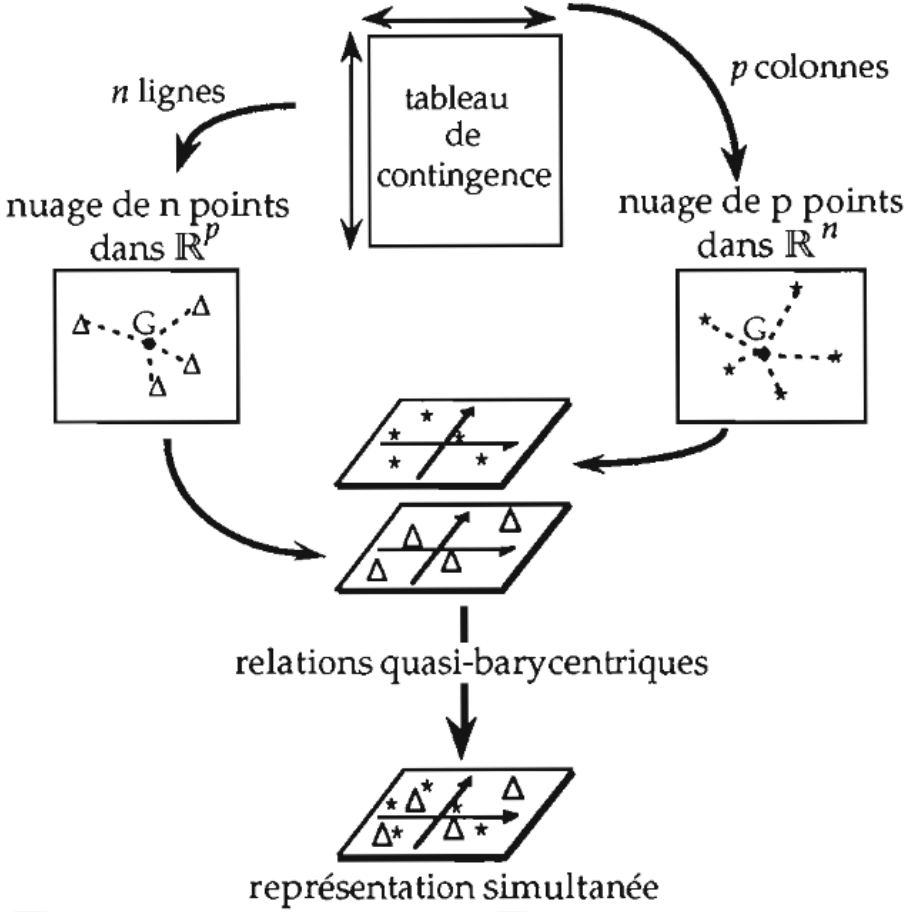
Contrairement à l'analyse en composantes principales, le tableau de données (tableau de contingence) subit deux transformations, l'une en profils-lignes, l'autre en profils-colonnes, à partir desquelles vont être construits les nuages de points dans IR^n et IR^p.



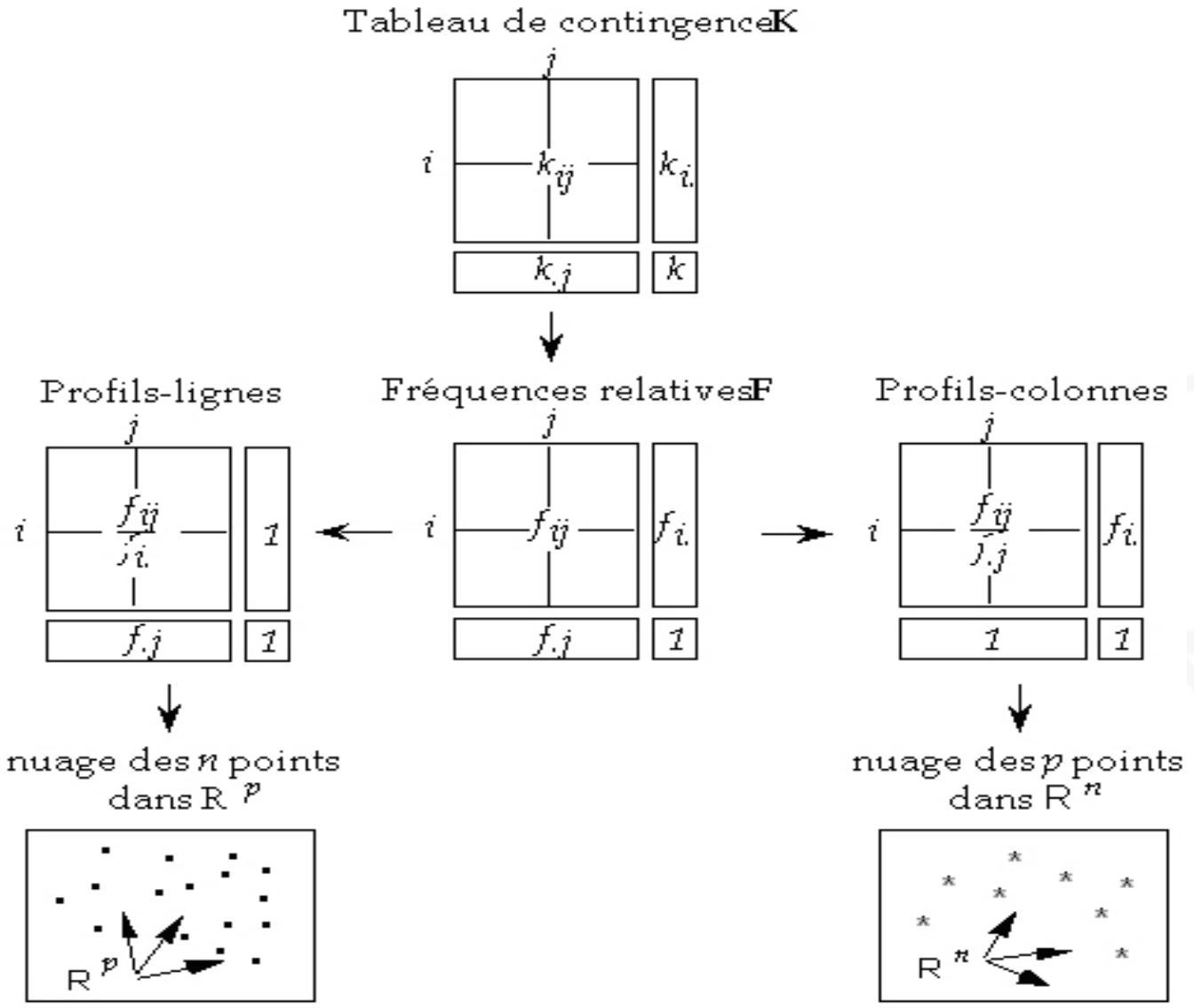
Représentation Simultanée

Construction des nuages

Les deux nuages de points (dans l'espace des colonnes et dans l'espace des lignes) sont construits de manière analogue.

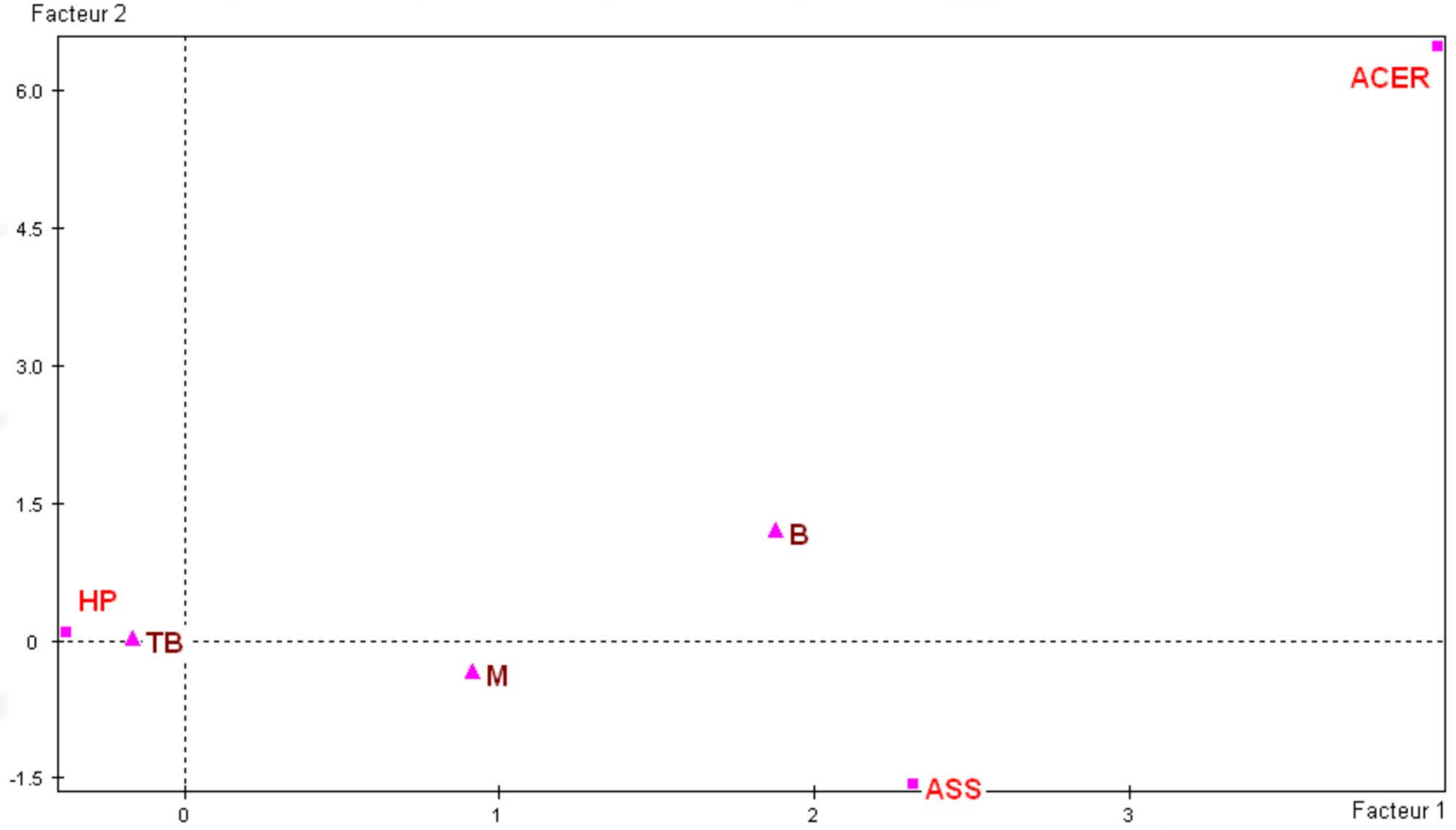


Représentation Simultanée



Représentation Simultanée

La représentation simultanée des différentes modalités de deux variables qualitatives est la suivante :



Exemple pratique de AFC sous Python

```
#changement de dossier
import os
os.chdir("C:/AFC")
```

```
#chargement des données - index_col = 0 pour indiquer que la colonne n°0 est un label
import pandas
D = pandas.read_excel("Data_Methodes_Factorielles.xlsx",sheet_name="AFC_ETUDES",index_col=0)

#affichage des données
print(D)
```

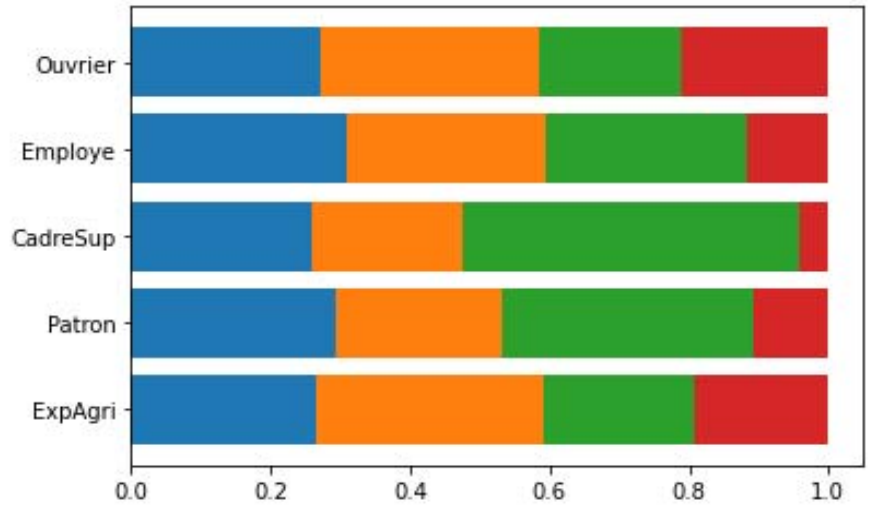
```
                Droit  Sciences  Medecine  IUT
CSP_vs_Filiere
ExpAgri            80        99        65   58
Patron            168       137       208   62
CadreSup          470       400       876   79
Employe           145       133       135   54
Ouvrier           166       193       127  129
```

```
#librairie
import numpy

#calcul des totaux en ligne
tot_lig = numpy.sum(D.values,axis=1)
print(tot_lig)

#calcul des totaux en colonne
tot_col = numpy.sum(D.values,axis=0)
print(tot_col)
```

```
[ 302  575 1825  467  615]
[1029  962 1411  382]
```

```python
#profils lignes
prof_lig = numpy.apply_along_axis(arr=D.values,axis=1,func1d=lambda x:x/numpy.sum(x))
print(prof_lig)

#représentation graphique
import matplotlib.pyplot as plt
somme = numpy.zeros(shape=(prof_lig.shape[0]))
for i in range(prof_lig.shape[1]):
    plt.barh(range(prof_lig.shape[0]),prof_lig[:,i],left=somme)
    somme = somme + prof_lig[:,i]

plt.yticks(range(prof_lig.shape[0]),D.index)
plt.show()
```

```
[[0.26490066 0.32781457 0.21523179 0.19205298]
 [0.29217391 0.23826087 0.36173913 0.10782609]
 [0.25753425 0.21917808 0.48       0.04328767]
 [0.31049251 0.28479657 0.28907923 0.11563169]
 [0.2699187  0.31382114 0.20650407 0.2097561 ]]
```



```
#profil marginal corresp.
prof_marg_lig = tot_col/numpy.sum(tot_col)
print(prof_marg_lig)
```

```
[0.27193446 0.25422833 0.37288584 0.10095137]
```

```
#effectifs totaux
n = numpy.sum(D.values)

#tableau sous indépendance
E = numpy.dot(numpy.reshape(tot_lig,(5,1)),numpy.reshape(tot_col,(1,4)))/n
print(E)
```

```
[[ 82.12420719  76.7769556  112.6115222   30.48731501]
 [156.36231501 146.18128964 214.40935518  58.04704017]
 [496.28039112 463.9667019  680.51664905 184.23625793]
 [126.99339323 118.72463002 174.13768499  47.14429175]
 [167.23969345 156.35042283 229.32478858  62.08509514]]
```

```
#statistique du KHI-2
KHI2 = numpy.sum(((D.values-E)**2)/E)
print(KHI2)

#degré de liberté
ddl = (E.shape[0]-1)*(E.shape[1]-1)
print(ddl)

#librairie scipy pour calcul des CDF
import scipy

#p-value du test
print(1-scipy.stats.chi2.cdf(KHI2,ddl))
```

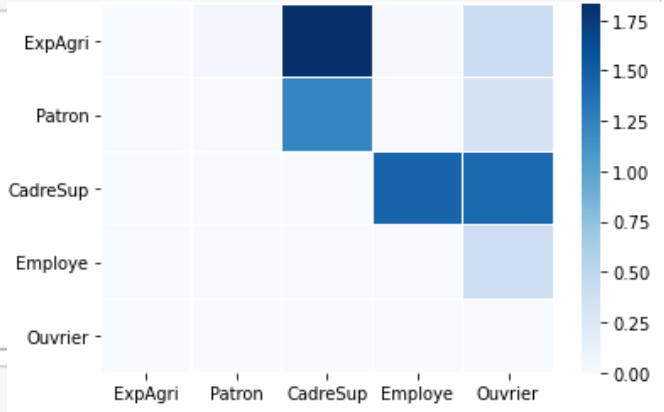
```
320.2658717522244
12
0.0
```

```
#distance du KHI-2 entre cadre(2) et ouvrier(4)
print(numpy.sum((prof_lig[2,:]-prof_lig[4,:])**2/prof_marg_lig))

#distance du KHI-2 entre cadre(2) et patron(1)
print(numpy.sum((prof_lig[2,:]-prof_lig[1,:])**2/prof_marg_lig))
```

```
1.4284648739611923
1.2314302986917416
```

```
#distance entre paires de modalités lignes
distPairesLig = numpy.zeros(shape=(prof_lig.shape[0],prof_lig.shape[0]))

#double boucle
for i in range(prof_lig.shape[0]-1):
    for j in range(i+1,prof_lig.shape[0]):
        distPairesLig[i,j] = numpy.sum((prof_lig[i,:]-prof_lig[j,:])**2/prof_marg_lig)

#affichage
print(pandas.DataFrame(distPairesLig,index=D.index,columns=D.index))
```

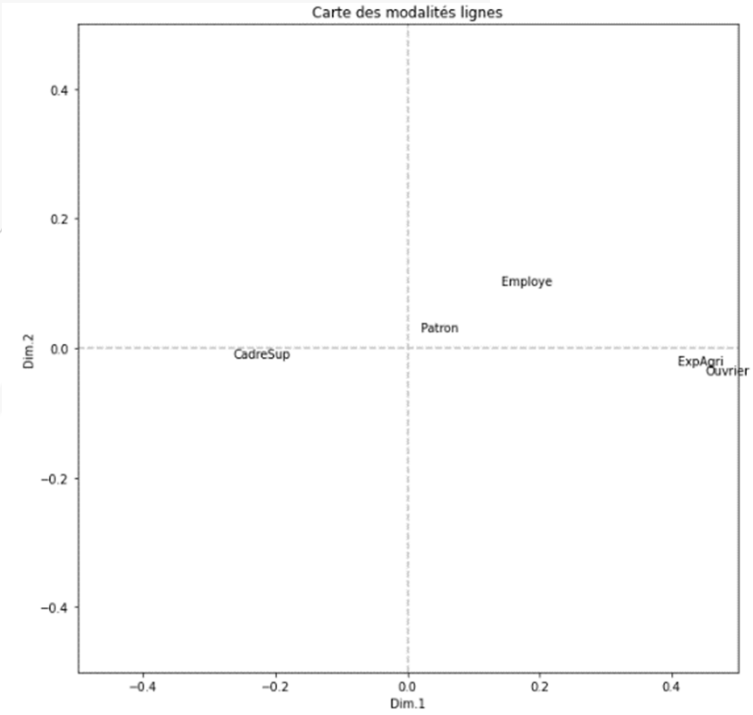
CSP_vs_Filiere	ExpAgri	Patron	CadreSup	Employe	Ouvrier
CSP_vs_Filiere					
ExpAgri	0.0	0.061664	1.829222	0.027273	0.410619
Patron	0.0	0.000000	1.231430	0.013428	0.326908
CadreSup	0.0	0.000000	0.000000	1.451883	1.428465
Employe	0.0	0.000000	0.000000	0.000000	0.398761
Ouvrier	0.0	0.000000	0.000000	0.000000	0.000000



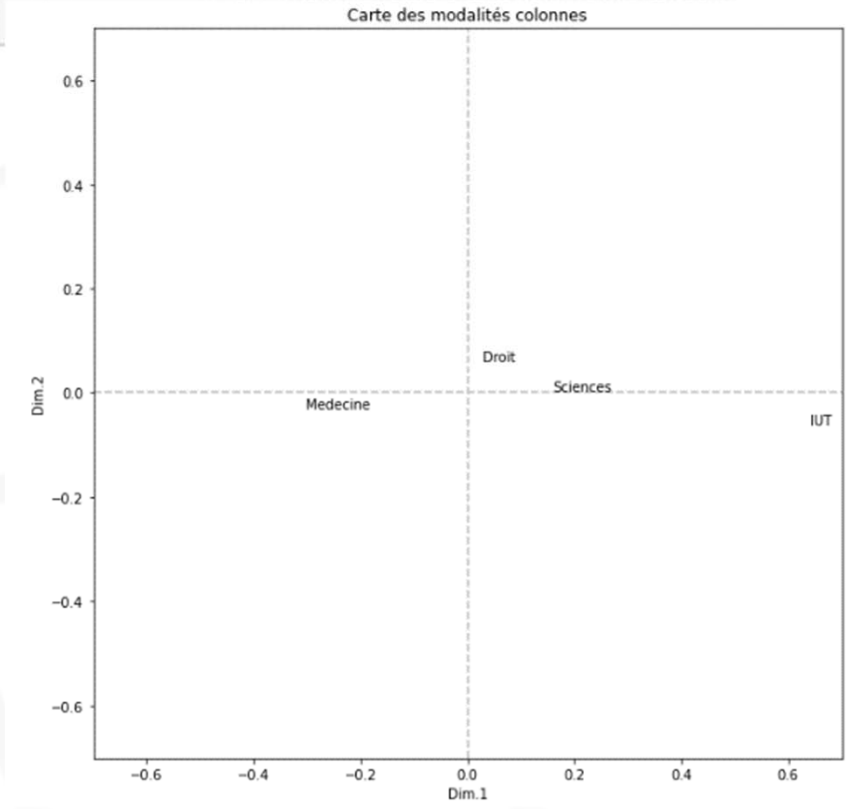

```
#affichage sous forme de heatmap
import seaborn as sns
sns.heatmap(distPairesLig,vmin=0,vmax=numpy.max(distPairesLig),linewidth=0.1,cmap='Blues',xticklabels=D.index,yticklabels=D.index
```

```python
#importation de la librairie
from fanalysis.ca import CA

#lancer les calculs
afc = CA(row_labels=D.index,col_labels=D.columns)
afc.fit(D.values)
```

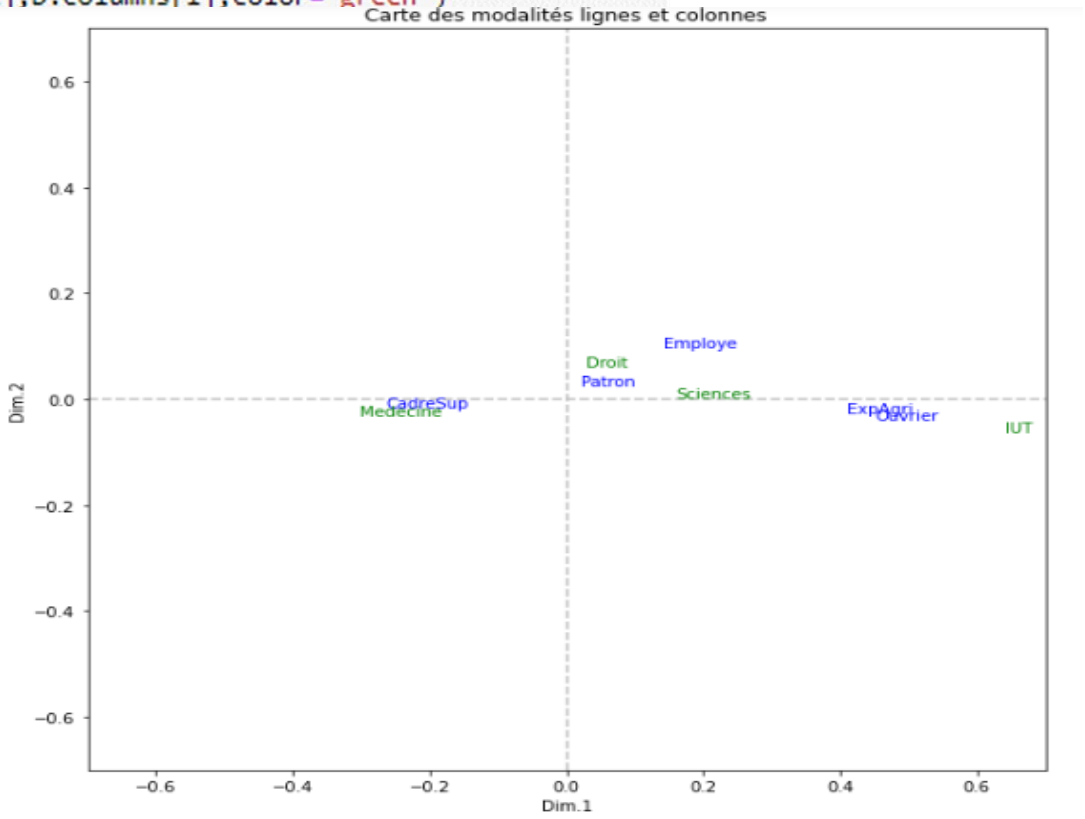
```python
#affichage dans le premier plan factoriel
fig, ax = plt.subplots(figsize=(10,10))
ax.axis([-0.5,+0.5,-0.5,+0.5])
ax.plot([-0.5,+0.5],[0,0],color='silver',linestyle='--')
ax.plot([0,0],[-0.5,+0.5],color='silver',linestyle='--')
ax.set_xlabel("Dim.1")
ax.set_ylabel("Dim.2")
plt.title("Carte des modalités lignes")

for i in range(D.shape[0]):
    ax.text(afc.row_coord_[i,0],afc.row_coord_[i,1],D.index[i])

plt.show()
```



```python
#affichage dans le premier plan factoriel
fig, ax = plt.subplots(figsize=(10,10))
ax.axis([-0.7,+0.7,-0.7,+0.7])
ax.plot([-0.7,+0.7],[0,0],color='silver',linestyle='--')
ax.plot([0,0],[-0.7,+0.7],color='silver',linestyle='--')
ax.set_xlabel("Dim.1")
ax.set_ylabel("Dim.2")
plt.title("Carte des modalités colonnes")

for i in range(D.shape[1]):
    ax.text(afc.col_coord_[i,0],afc.col_coord_[i,1],D.columns[i])

plt.show()
```



```python
#représentation simultanée
fig, ax = plt.subplots(figsize=(10,10))
ax.axis([-0.7,+0.7,-0.7,+0.7])
ax.plot([-0.7,+0.7],[0,0],color='silver',linestyle='--')
ax.plot([0,0],[-0.7,+0.7],color='silver',linestyle='--')
ax.set_xlabel("Dim.1")
ax.set_ylabel("Dim.2")
plt.title("Carte des modalités lignes et colonnes")

#modalités ligne
for i in range(D.shape[0]):
    ax.text(afc.row_coord_[i,0],afc.row_coord_[i,1],D.index[i],color='blue')

#modalités colonne
for i in range(D.shape[1]):
    ax.text(afc.col_coord_[i,0],afc.col_coord_[i,1],D.columns[i],color='green')

plt.show()
```



Exemple pratique de AFC sous R

```
# tableau de contingence entre deux variables qualitatives
data=read.table("ordinateurs.txt", header=T, sep="\t")
TC=table(data$Marque,data$Finition)
addmargins(TC)
TCp=prop.table(TC)
addmargins (TCp)
# tableau de profils-lignes entre deux variables qualitatives
PL=prop.table(TCp,1)
addmargins(PL,2)
# tableau de profils-colonnes entre deux variables qualitatives
PC=prop.table(TCp,2)
addmargins(PC,1)
# test de khi deux entre deux variables qualitatives et Analyse d'association
chisq.test(TC)
chisq.test(TC)$expected   # tableau de contingence théorique
chisq.test(TC)$observed   # tableau de contingence empirique
 etude=CA(TC, ncp=2)
```