

Classification Hiérarchique sur Composantes Principales sous Python avec scientisttools

Duvérier DJIFACK ZEBAZE

Ce tutoriel a pour objectif de présenter rapidement les principales fonctionnalités offertes par le package « scientisttools » pour réaliser une classification hiérarchique combinée avec une analyse factorielle (HCPC, *Hierarchical Clustering on Principal Components*).

Présentation des données

On va réaliser une classification hiérarchique sur les composantes principales d’une analyse factorielle. Nous allons prendre un exemple sur les données météorologiques. Les données sur lesquelles nous allons travailler proviennent d’un jeu de données d’étudiants français qui avaient pris cela comme sujet d’examen (Tableau 1). En lignes, les individus statistiques sont représentés par les 15 villes de France sélectionnées et en colonnes les températures mensuelles moyennes. Ces températures mensuelles moyennes ont été calculées sur 30 ans. Donc, par exemple, à Bordeaux en Janvier, il fait en moyenne 5.6 degrés. Cette valeur de 5.6 degrés est la moyenne sur tous les jours de Janvier pendant 30 ans. On a ainsi 12 variables correspondants au 12 mois de l’année. On retrouve également en colonnes deux variables liées à la position géographique des villes (Latitude et Longitude).

```
# Chargement des données
import pandas as pd
Data = pd.read_excel("./donnee/temperature_acp.xlsx", sheet_name=0, index_col=0)
```

ACP

Le but général de l’étude est de comparer les températures mensuelles des différentes villes. D’une part du point de vue des villes, on pose les questions suivantes : Quelles sont les villes qui se ressemblent vis-à-vis de l’ensemble des variables (les mois). Quelles sont celles qui diffèrent. Plus généralement, peut-on faire une typologie des villes mettant en évidence l’ensemble des ressemblances ainsi définies ? D’autre part, du point de vue des mois : Quels mois sont corrélés entre eux ? Quels sont ceux qui le sont peu ? Plus généralement, peut-on faire un bilan de corrélation entre les 12 mois ? Les températures mensuelles sont-elles liées à la position géographique (variables supplémentaires) ?

Chargement de scientisttools

```
from scientisttools.decomposition import PCA
```

Table 1 – Données - Température des villes françaises

	Jan	Fev	Mars	Avril	Mai	Juin	Juil	Août	Sept	Oct	Nov	Dec	moy	amp	Lati	Long	groupe
Bordeaux	5.6	6.6	10.3	12.8	15.8	19.3	20.9	21.0	18.6	13.8	9.1	6.2	13.333333	15.4	44.50	-0.34	C
Brest	6.1	5.8	7.8	9.2	11.6	14.4	15.6	16.0	14.7	12.0	9.0	7.0	10.766667	10.2	48.24	-4.29	A
Clermont	2.6	3.7	7.5	10.3	13.8	17.3	19.4	19.1	16.2	11.2	6.6	3.6	10.941667	16.8	45.47	3.05	B
Grenoble	1.5	3.2	7.7	10.6	14.5	17.8	20.1	19.5	16.7	11.4	6.5	2.3	10.983333	18.6	45.10	5.43	B
Lille	2.4	2.9	6.0	8.9	12.4	15.3	17.1	17.1	14.7	10.4	6.1	3.5	9.733333	14.7	50.38	3.04	B
Lyon	2.1	3.3	7.7	10.9	14.9	18.5	20.7	20.1	16.9	11.4	6.7	3.1	11.358333	18.6	45.45	4.51	B
Marseille	5.5	6.6	10.0	13.0	16.8	20.8	23.3	22.8	19.9	15.0	10.2	6.9	14.233333	17.8	43.18	5.24	C
Montpellier	5.6	6.7	9.9	12.8	16.2	20.1	22.7	22.3	19.3	14.6	10.0	6.5	13.891667	17.1	43.36	3.53	C
Nantes	5.0	5.3	8.4	10.8	13.9	17.2	18.8	18.6	16.4	12.2	8.2	5.5	11.691667	13.8	47.13	-1.33	A
Nice	7.5	8.5	10.8	13.3	16.7	20.1	22.7	22.5	20.3	16.0	11.5	8.2	14.841667	15.2	43.42	7.15	C
Paris	3.4	4.1	7.6	10.7	14.3	17.5	19.1	18.7	16.0	11.4	7.1	4.3	11.183333	15.7	48.52	2.20	B
Rennes	4.8	5.3	7.9	10.1	13.1	16.2	17.9	17.8	15.7	11.6	7.8	5.4	11.133333	13.1	48.05	-1.41	A
Strasbourg	0.4	1.5	5.6	9.8	14.0	17.2	19.0	18.3	15.1	9.5	4.9	1.3	9.716667	18.6	48.35	7.45	B
Toulouse	4.7	5.6	9.2	11.6	14.9	18.7	20.9	20.9	18.3	13.3	8.6	5.5	12.683333	16.2	43.36	1.26	C
Vichy	2.4	3.4	7.1	9.9	13.6	17.1	19.3	18.8	16.0	11.0	6.6	3.4	10.716667	16.9	46.08	3.26	B
Amsterdam	2.9	2.5	5.7	8.2	12.5	14.8	17.1	17.1	14.5	11.4	7.0	4.4	NaN	NaN	NaN	NaN	NA
Anvers	3.1	2.9	6.2	8.9	12.9	15.5	17.9	17.6	14.7	11.5	6.8	4.7	NaN	NaN	NaN	NaN	NA
Athènes	9.1	9.7	11.7	15.4	20.1	24.5	27.4	27.2	23.8	19.2	14.6	11.0	NaN	NaN	NaN	NaN	NA
Barcelone	9.1	10.3	11.8	14.1	17.4	21.2	24.2	24.1	21.7	17.5	13.1	10.0	NaN	NaN	NaN	NaN	NA
Berlin	-0.2	0.1	4.4	8.2	13.8	16.0	18.3	18.0	14.4	10.0	4.2	1.2	NaN	NaN	NaN	NaN	NA
Bruxelles	3.3	3.3	6.7	8.9	12.8	15.6	17.8	17.8	15.0	11.1	6.7	4.4	NaN	NaN	NaN	NaN	NA
Budapest	-1.1	0.8	5.5	11.6	17.0	20.2	22.0	21.3	16.9	11.3	5.1	0.7	NaN	NaN	NaN	NaN	NA
Copenhague	-0.4	-0.4	1.3	5.8	11.1	15.4	17.1	16.6	13.3	8.8	4.1	1.3	NaN	NaN	NaN	NaN	NA
Cracovie	-3.7	-2.0	1.9	7.9	13.2	16.9	18.4	17.6	13.7	8.6	2.6	-1.7	NaN	NaN	NaN	NaN	NA
Dublin	4.8	5.0	5.9	7.8	10.4	13.3	15.0	14.6	12.7	9.7	6.7	5.4	NaN	NaN	NaN	NaN	NA

On crée une instance de la classe PCA, en lui passant ici des étiquettes pour les lignes et les variables. Ces paramètres sont facultatifs ; en leur absence, le programme détermine automatiquement des étiquettes.

Le constructeur de la classe PCA possède un paramètre `normalize` qui indique si l'ACP est réalisée :

- à partir de données centrées et réduites -> `PCA(normalize=True)`
- à partir de données centrées mais non réduites -> `PCA(normalize=False)`

Par défaut, la valeur du paramètre `normalize` est fixée à `True`, car c'est le cas le plus courant.

Réalisez l'ACP sur tous les individus (actifs et supplémentaires) et les variables (actives et supplémentaires) en tapant la ligne de code suivante :

```
# ACP - Instanciation
res_pca = PCA(normalize=True,
               n_components = None,
               row_labels=Data.index[:15],
               col_labels=Data.columns[:12],
               row_sup_labels=Data.index[15:],
               quanti_sup_labels=["moy", "amp", "Lati", "Long"],
               quali_sup_labels=["groupe"],
               parallelize=True).fit(Data)
```

HCPC

La première étape consistait à réaliser une ACP du tableau de données. On réalise ensuite la classification hiérarchique. Nous demandons une partition en 3 classes.

```
from scientisttools.clustering import HCPC
res_hcpc = HCPC(n_clusters=3)
```

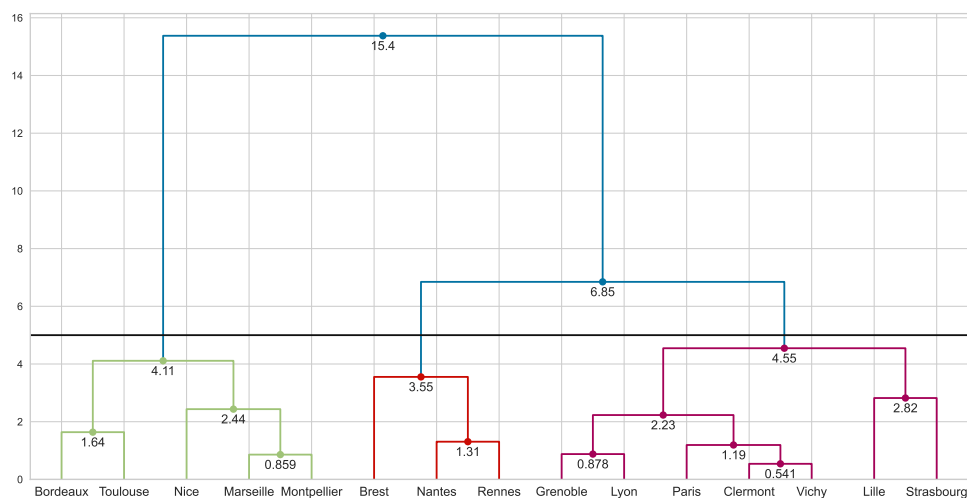
```
# Entraînement du modèle
res_hcpc.fit(res_pca)
```

```
## HCPC(n_clusters=3)
```

Dendrogram

L'arbre hiérarchique nous montre notre partition en trois classes.

```
# Plot dendrogram
from scientisttools.pyplot import plot_dendrogram
import matplotlib.pyplot as plt
fig,axe = plt.subplots(figsize=(16,8))
plot_dendrogram(res_hcpc,ax=axe,max_d=5)
plt.show()
```



Plan factoriel

Le plan factoriel où les individus sont coloriés en fonction de la classe à laquelle ils appartiennent est le suivant :

```
# Plan factoriel
from plotnine import *
from scientisttools.ggplot import fviz_hcpc_cluster
p = (fviz_hcpc_cluster(res_hcpc,add_ellipse=False,repel=False,
                        show_clust_cent=False,center_marker_size=5)+theme_gray()+
      theme(legend_direction="vertical",legend_position=(0.2,0.3)))
print(p)
```

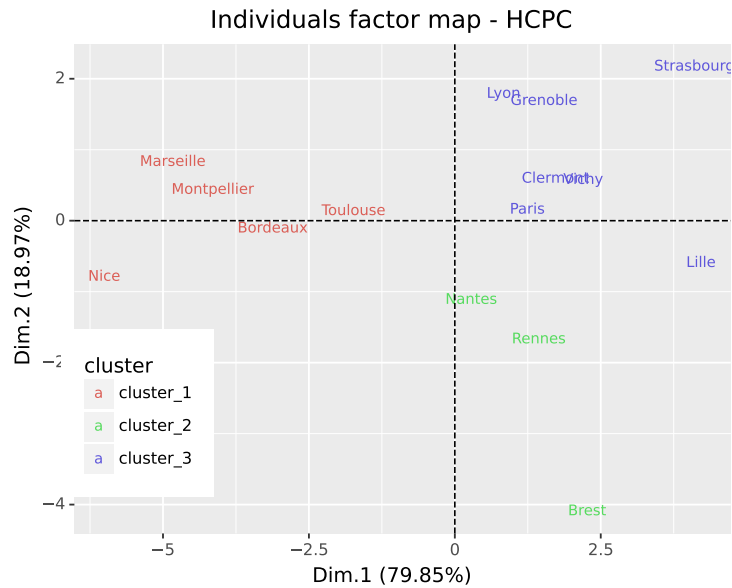


Figure 1 – Plan factoriel

```
# Statistiques sur les classes
cinfos = res_hcpc.cluster_infos_
```

Table 2 – Statistiques sur les classes

	n(k)	p(k)
cluster_1	5	0.3333333
cluster_2	3	0.2000000
cluster_3	7	0.4666667

En creusant plus en profondeur, on a la composition suivante :

- La classe 1 (Les 5 villes méridionales) : Bordeaux, Marseille, Montpellier, Nice et Toulouse.
- La classe 2 (les 3 villes les plus occidentales - à faible amplitude thermique) : Brest, Nantes et Rennes
- La classe 3 (les 7 villes à forte amplitude thermique) : Clermont, Grenoble, Lille, Lyon, Paris, Strasbourg et Vichy.

Description des classes

Les classes peuvent être décrites par :

- les variables et/ou des modalités
- les axes factoriels
- les individus

Moyenne par classe

```
# Moyennes des variables par classe
gmean=res_hcpc.gmean_
```

Table 3 – Moyennes des variables par classe

	cluster_1	cluster_2	cluster_3
Jan	5.78	5.300000	2.114286
Fev	6.80	5.466667	3.157143
Mars	10.04	8.033333	7.028571
Avril	12.70	10.033333	10.157143
Mai	16.08	12.866667	13.928571
Juin	19.80	15.933333	17.242857
Juil	22.10	17.433333	19.242857
Août	21.90	17.466667	18.800000
Sept	19.28	15.600000	15.942857
Oct	14.54	11.933333	10.900000
Nov	9.88	8.333333	6.357143
Dec	6.66	5.966667	3.071429

Coordonnées des classes

```
# Centre de gravité des classes
gclasse = res_hcpc.cluster_centers_
```

Table 4 – Centre de gravité

	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5	Dim.6	Dim.7	Dim.8	Dim.9	Dim.10	Dim.11	Dim.12
cluster_1	-3.968786	0.1003350	-0.0134384	0.0144852	-0.0095143	-0.0107136	0.0319173	0.0114914	-0.0012029	-0.0041956	0.0011478	0.0006105
cluster_2	1.329316	-2.2929976	0.0885413	-0.0346947	-0.0168230	0.0144214	-0.0559224	0.0008738	-0.0191192	-0.0002742	0.0048650	-0.0010853
cluster_3	2.265141	0.9110454	-0.0283474	0.0045226	0.0140058	0.0014720	0.0011687	-0.0085827	0.0090531	0.0031143	-0.0029048	0.0000291

Corrélation entre classe et variables

```
# Rapport de corrélation
eta2 = res_hcpc.correlation_ratio_
```

Table 5 – Rapport de corrélation

	Sum. Intra	Sum. Inter	correlation ratio	F-stats	pvalue
Oct	7.6787	39.2053	0.8362	30.6345	0e+00
Sept	8.1251	39.6922	0.8301	29.3106	0e+00
Fev	8.6638	40.2095	0.8227	27.8465	0e+00
Mars	6.1330	26.6004	0.8126	26.0237	0e+00
Jan	10.6166	45.7928	0.8118	25.8800	0e+00
Nov	8.7318	36.8175	0.8083	25.2989	0e+00
Avril	5.8038	22.2202	0.7929	22.9713	1e-04
Dec	11.4330	42.2644	0.7871	22.1803	1e-04
Août	12.1067	44.5667	0.7864	22.0870	1e-04
Juin	12.4238	32.6095	0.7241	15.7486	4e-04
Mai	8.9890	22.7044	0.7164	15.1549	5e-04
Juil	18.0438	45.4095	0.7156	15.0998	5e-04

Description par variables

```
# Description par les variables quantitatives
vardesc = res_hcpc.desc_var_quant_
# Pour un meilleur rendu
vardesc = (vardesc.reset_index()
            .rename(columns={"level_0":"cluster","level_1" : "variable"}))
```

Table 6 – Description par les variables

cluster	variable	vtest	pvalue	mean in category	overall mean	sd in categorie	overall sd	significant
cluster_1	Sept	3.3983582	0.0006779	19.280000	16.986667	0.7547185	1.785447	***
cluster_1	Avril	3.3293391	0.0008705	12.700000	10.980000	0.5796551	1.366845	***
cluster_1	Oct	3.3222728	0.0008929	14.540000	12.320000	0.9414882	1.767937	***
cluster_1	Mars	3.2357694	0.0012132	10.040000	8.233333	0.5238320	1.477235	**
cluster_1	Août	3.1760114	0.0014932	21.900000	19.566667	0.7924645	1.943765	**
cluster_1	Juin	3.0030209	0.0026731	19.800000	17.833333	0.7266361	1.732692	**
cluster_1	Mai	2.9972046	0.0027247	16.080000	14.433333	0.6910861	1.453578	**
cluster_1	Nov	2.9657190	0.0030198	9.880000	7.926667	0.9987993	1.742591	**
cluster_1	Juil	2.9157822	0.0035480	22.100000	19.833333	1.0039920	2.056750	**
cluster_1	Fev	2.8826335	0.0039437	6.800000	4.833333	0.9402127	1.805055	**
cluster_1	Dec	2.5356887	0.0112226	6.660000	4.846667	0.8957678	1.892042	**
cluster_1	Jan	2.4648880	0.0137056	5.780000	3.973333	0.9239048	1.939232	**
cluster_2	Jan	1.2798704	0.2005907	5.300000	3.973333	0.5715476	1.939232	
cluster_2	Dec	1.1074428	0.2681025	5.966667	4.846667	0.7318166	1.892042	
cluster_2	Fev	0.6564113	0.5115596	5.466667	4.833333	0.2357023	1.805055	
cluster_2	Nov	0.4365934	0.6624062	8.333333	7.926667	0.4988877	1.742591	
cluster_2	Mars	-0.2532880	0.8000457	8.033333	8.233333	0.2624669	1.477235	
cluster_2	Oct	-0.4091703	0.6824147	11.933333	12.320000	0.2494438	1.767937	
cluster_2	Avril	-1.2957215	0.1950714	10.033333	10.980000	0.6548961	1.366845	
cluster_2	Sept	-1.4529780	0.1462299	15.600000	16.986667	0.6976150	1.785447	
cluster_2	Mai	-2.0163796	0.0437603	12.866667	14.433333	0.9533566	1.453578	**
cluster_2	Août	-2.0212013	0.0432589	17.466667	19.566667	1.0873004	1.943765	**
cluster_2	Juin	-2.0514749	0.0402207	15.933333	17.833333	1.1585431	1.732692	**
cluster_2	Juil	-2.1830499	0.0290321	17.433333	19.833333	1.3474255	2.056750	**
cluster_3	Juil	-1.0048213	0.3149829	19.242857	19.833333	1.0390341	2.056750	
cluster_3	Juin	-1.1927489	0.2329677	17.242857	17.833333	0.9068897	1.732692	
cluster_3	Mai	-1.2153918	0.2242167	13.928571	14.433333	0.7439552	1.453578	
cluster_3	Août	-1.3804824	0.1674382	18.800000	19.566667	0.8799351	1.943765	
cluster_3	Sept	-2.0461725	0.0407394	15.942857	16.986667	0.7384485	1.785447	**
cluster_3	Avril	-2.1070413	0.0351140	10.157143	10.980000	0.6366366	1.366845	**
cluster_3	Oct	-2.8111867	0.0049359	10.900000	12.320000	0.6611678	1.767937	**
cluster_3	Mars	-2.8544325	0.0043114	7.028571	8.233333	0.8066054	1.477235	**
cluster_3	Nov	-3.1523946	0.0016194	6.357143	7.926667	0.6543419	1.742591	**
cluster_3	Fev	-3.2501325	0.0011535	3.157143	4.833333	0.7631835	1.805055	**
cluster_3	Dec	-3.2839302	0.0010237	3.071429	4.846667	0.9113793	1.892042	**
cluster_3	Jan	-3.3552795	0.0007928	2.114286	3.973333	0.8757504	1.939232	***

Le tableau 6 fournit directement les caractéristiques des classes. Nous les résumons en 3 points :

1. Les individus de la classe 1 sont caractérisés par une température élevée toute l'année, particulièrement en demi - saison. Ces villes sont méridionales (faible latitude).
2. « A l'opposé », les individus de la classe 3 sont caractérisés par une température faible toute l'année, particulièrement pendant les mois les plus froids.
3. La classe 2 comporte des villes présentant une faible amplitude thermique ; elles sont situées à l'ouest (faible longitude).

Description par les axes factoriels

```
# Description par les axes factoriels
axinfos = res_hcpc.desc_axes_infos_
axinfos= (axinfos.reset_index()
          .rename(columns={"level_0":"cluster","level_1" : "variable"}))
```

Table 7 – Description par les axes

cluster	variable	vtest	pvalue	mean in category	overall mean	sd in categorie	overall sd	significant
cluster_1	Dim.7	1.0857353	0.2775961	0.0319173	0	0.0787699	0.0777770	
cluster_1	Dim.8	0.7274288	0.4669633	0.0114914	0	0.0577573	0.0417958	
cluster_1	Dim.12	0.3633195	0.7163663	0.0006105	0	0.0046557	0.0044455	
cluster_1	Dim.4	0.1924051	0.8474249	0.0144852	0	0.1525765	0.1991852	
cluster_1	Dim.11	0.1796193	0.8574515	0.0011478	0	0.0127439	0.0169067	
cluster_1	Dim.2	0.1759443	0.8603377	0.1003350	0	0.5439841	1.5087804	
cluster_1	Dim.9	-0.0823863	0.9343395	-0.0012029	0	0.0398512	0.0386287	
cluster_1	Dim.3	-0.1343700	0.8931100	-0.0134384	0	0.3803327	0.2646024	
cluster_1	Dim.5	-0.2124036	0.8317922	-0.0095143	0	0.1111880	0.1185128	
cluster_1	Dim.6	-0.3172798	0.7510313	-0.0107136	0	0.1012969	0.0893394	
cluster_1	Dim.10	-0.5003787	0.6168085	-0.0041956	0	0.0120035	0.0221841	
cluster_1	Dim.1	-3.3922172	0.0006933	-3.9687864	0	1.4585295	3.0954450	***
cluster_2	Dim.1	0.8034135	0.4217358	1.3293158	0	0.8147498	3.0954450	
cluster_2	Dim.3	0.6260174	0.5313035	0.0885413	0	0.1471386	0.2646024	
cluster_2	Dim.11	0.5383415	0.5903413	0.0048650	0	0.0113500	0.0169067	
cluster_2	Dim.6	0.3019941	0.7626566	0.0144214	0	0.1147072	0.0893394	
cluster_2	Dim.8	0.0391125	0.9688007	0.0008738	0	0.0263787	0.0417958	
cluster_2	Dim.10	-0.0231223	0.9815527	-0.0002742	0	0.0329162	0.0221841	
cluster_2	Dim.5	-0.2655659	0.7905735	-0.0168230	0	0.0843860	0.1185128	
cluster_2	Dim.4	-0.3258671	0.7445249	-0.0346947	0	0.0655059	0.1991852	
cluster_2	Dim.12	-0.4567303	0.6478649	-0.0010853	0	0.0032601	0.0044455	
cluster_2	Dim.9	-0.9259628	0.3544653	-0.0191192	0	0.0408106	0.0386287	
cluster_2	Dim.7	-1.3451437	0.1785788	-0.0559224	0	0.0520960	0.0777770	
cluster_2	Dim.2	-2.8432273	0.0044659	-2.2929976	0	1.2931294	1.5087804	**
cluster_3	Dim.1	2.5611802	0.0104317	2.2651406	0	1.2606742	3.0954450	**
cluster_3	Dim.2	2.1134016	0.0345664	0.9110454	0	0.9270734	1.5087804	**
cluster_3	Dim.9	0.8202697	0.4120624	0.0090531	0	0.0333123	0.0386287	
cluster_3	Dim.10	0.4913525	0.6231772	0.0031143	0	0.0215639	0.0221841	
cluster_3	Dim.5	0.4136290	0.6791458	0.0140058	0	0.1335385	0.1185128	
cluster_3	Dim.4	0.0794692	0.9366594	0.0045226	0	0.2566394	0.1991852	
cluster_3	Dim.6	0.0576673	0.9540136	0.0014720	0	0.0629416	0.0893394	
cluster_3	Dim.7	0.0525909	0.9580578	0.0011687	0	0.0727897	0.0777770	
cluster_3	Dim.12	0.0228943	0.9817346	0.0000291	0	0.0046409	0.0044455	
cluster_3	Dim.3	-0.3749629	0.7076881	-0.0283474	0	0.1820343	0.2646024	
cluster_3	Dim.11	-0.6013577	0.5476018	-0.0029048	0	0.0205370	0.0169067	
cluster_3	Dim.8	-0.7187154	0.4723163	-0.0085827	0	0.0299003	0.0417958	

Les individus de la classe 1 possèdent de faibles coordonnées sur le premier axe. Ceux de la classe 2 possèdent des coordonnées faibles sur le deuxième axe et les individus de la classe 2 possèdent des coordonnées élevées sur les deux premiers axes.

Description par les individus

Il existe deux types d'individus spécifiques pour décrire les classes :

- Les individus les plus proches du centre de classe (le parangons)
- Les individus les plus éloignés des centres des autres classes.

```
# Individu proches
near = res_hcpc.disto_near_
```

```
near = (near.reset_index()
        .drop(columns=["level_1"])
        .rename(columns={"level_0": "cluster"}))
```

Table 8 – Description par les individus proches

cluster	Villes	distance
cluster_1	Montpellier	0.2034325
cluster_1	Bordeaux	1.3121075
cluster_1	Marseille	1.4533573
cluster_1	Nice	5.0584238
cluster_1	Toulouse	5.0988801
cluster_2	Rennes	0.4379303
cluster_2	Nantes	2.5201571
cluster_2	Brest	4.2074689
cluster_3	Vichy	0.1900087
cluster_3	Clermont	0.4515212
cluster_3	Grenoble	1.4065429
cluster_3	Paris	1.7989452
cluster_3	Lyon	2.8368509
cluster_3	Strasbourg	5.1366175
cluster_3	Lille	6.2239000

```
### Individus loins
far = res_hcpc.disto_far_
far = (far.reset_index()
        .drop(columns=["level_1"])
        .rename(columns={"level_0": "cluster"}))
```

Table 9 – Description par les individus éloignés

cluster	Villes	distance
cluster_1	Toulouse	5.0988801
cluster_1	Nice	5.0584238
cluster_1	Marseille	1.4533573
cluster_1	Bordeaux	1.3121075
cluster_1	Montpellier	0.2034325
cluster_2	Brest	4.2074689
cluster_2	Nantes	2.5201571
cluster_2	Rennes	0.4379303
cluster_3	Lille	6.2239000
cluster_3	Strasbourg	5.1366175
cluster_3	Lyon	2.8368509
cluster_3	Paris	1.7989452
cluster_3	Grenoble	1.4065429
cluster_3	Clermont	0.4515212
cluster_3	Vichy	0.1900087

Montpellier appartient à la classe 1 et est le plus proche du centre de cette classe. De même, Toulouse appartient à la même classe et est le plus éloigné du centre de cette classe.

L'attribut `.parangons_` permet avoir les individus les plus proches du centre de gravité de chaque classe.

```
# Parangons
para = res_hcpc.parangons_
```


Table 10 – Individus parangons

	parangons	distance
cluster_1	Montpellier	0.2034325
cluster_2	Rennes	0.4379303
cluster_3	Vichy	0.1900087

Autres attributs

```
# Correlation des variables supplémentaires et le cluster
vsupeta2 = res_hcpc.correlation_ratio_quanti_sup_
```

Table 11 – Rapport de corrélation variables supplémentaires quantitatives avec les axes factoriels

	Sum. Intra	Sum. Inter	correlation ratio	F-stats	pvalue
moy	5.8771	30.0873	0.8366	30.7168	0.0000
amp	26.7930	48.9844	0.6464	10.9695	0.0020
Lati	26.5720	47.1569	0.6396	10.6481	0.0022
Long	61.4075	92.7328	0.6016	9.0607	0.0040

```
# Rapport de corrélation avec les axes factoriels
axeta2 = res_hcpc.desc_axes_correlation_ratio_
```

Table 12 – Rapport de corrélation avec les axes factoriels

	Sum. Intra	Sum. Inter	correlation ratio	F-stats	pvalue
Dim.1	23.7531	119.9736	0.8347	30.3052	0.0000
Dim.2	12.5124	21.6339	0.6336	10.3740	0.0024
Dim.7	0.0763	0.0145	0.1596	1.1398	0.3522
Dim.9	0.0207	0.0017	0.0750	0.4861	0.6266
Dim.8	0.0250	0.0012	0.0450	0.2825	0.7588
Dim.11	0.0042	0.0001	0.0319	0.1975	0.8234
Dim.3	1.0202	0.0300	0.0286	0.1767	0.8402
Dim.10	0.0072	0.0002	0.0212	0.1296	0.8796
Dim.12	0.0003	0.0000	0.0182	0.1114	0.8955
Dim.5	0.2080	0.0027	0.0127	0.0772	0.9262
Dim.6	0.1185	0.0012	0.0101	0.0614	0.9407
Dim.4	0.5903	0.0048	0.0081	0.0488	0.9525

```
# Moyennes des axes par cluster
axgmean = res_hcpc.desc_axes_gmean_.T
```

Table 13 – Centre de gravité

	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5	Dim.6	Dim.7	Dim.8	Dim.9	Dim.10	Dim.11	Dim.12
cluster_1	-3.968786	0.1003350	-0.0134384	0.0144852	-0.0095143	-0.0107136	0.0319173	0.0114914	-0.0012029	-0.0041956	0.0011478	0.0006105
cluster_2	1.329316	-2.2929976	0.0885413	-0.0346947	-0.0168230	0.0144214	-0.0559224	0.0008738	-0.0191192	-0.0002742	0.0048650	-0.0010853
cluster_3	2.265141	0.9110454	-0.0283474	0.0045226	0.0140058	0.0014720	0.0011687	-0.0085827	0.0090531	0.0031143	-0.0029048	0.0000291

```
vsupdesc = res_hcpc.desc_var_quanti_sup_
vsupdesc = (vsupdesc.reset_index()
             .rename(columns={"level_0":"cluster","level_1" : "variable"}))
```

Table 14 – Description par les variables quantitatives supplémentaires

cluster	variable	vtest	pvalue	mean in category	overall mean	sd in categorie	overall sd	significant
cluster_1	moy	3.3879133	0.0007043	13.796667	11.813889	0.7410690	1.548427	***
cluster_1	Long	0.6476222	0.5172294	3.368000	2.583333	2.6842906	3.205624	
cluster_1	amp	0.5022428	0.6154968	16.340000	15.913333	0.9911609	2.247626	
cluster_1	Lati	-2.9539931	0.0031369	43.564000	46.039333	0.4748726	2.217039	**
cluster_2	Lati	1.4913489	0.1358699	47.806667	46.039333	0.4847221	2.217039	
cluster_2	moy	-0.7450643	0.4562329	11.197222	11.813889	0.3803223	1.548427	
cluster_2	Long	-2.8752439	0.0040372	-2.343333	2.583333	1.3768886	3.205624	**
cluster_2	amp	-2.9520952	0.0031563	12.366667	15.913333	1.5584893	2.247626	**
cluster_3	amp	1.8923671	0.0584421	17.128571	15.913333	1.4439176	2.247626	*
cluster_3	Long	1.6933783	0.0903835	4.134286	2.583333	1.6772840	3.205624	*
cluster_3	Lati	1.5955219	0.1105956	47.050000	46.039333	1.8799544	2.217039	
cluster_3	moy	-2.6038968	0.0092171	10.661905	11.813889	0.6207372	1.548427	**

```
vqsup = res_hcpc.desc_var_quali_sup_
vqsup.keys()
```

```
## dict_keys(['chi2', 'gtest', 'cramer', 'tschuprow', 'pearson'])
```

```
chi2 = vqsup["chi2"]
chi2
```

```
##          statistic    df    pvalue
## groupe  31.312711  4.0  0.000003
```

```
# Likelihood-test
gtest = vqsup["gtest"]
gtest
```

```
##          statistic    df    pvalue
## groupe  31.312711  4.0  0.000003
```

```
# Cramer's V statistic
vqsup["cramer"]
```

```
##          value
## groupe  0.816497
```

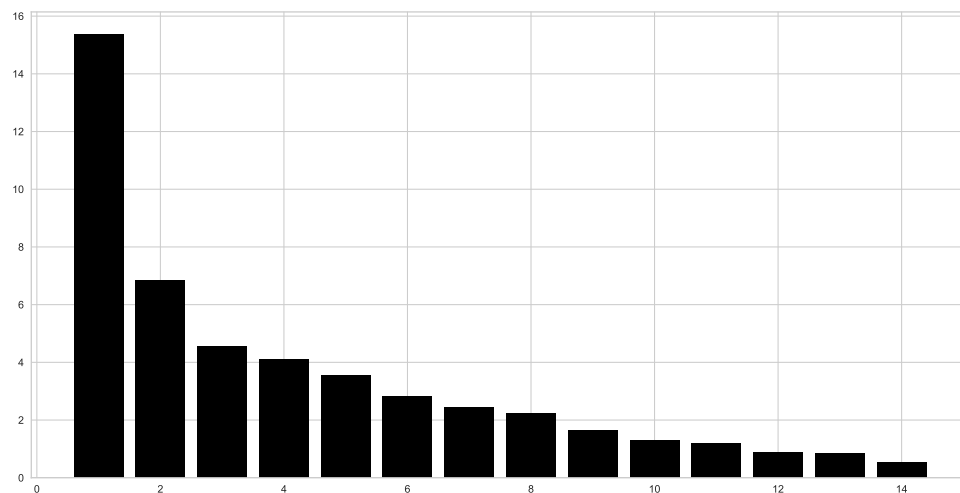
```
# Tschuprow's T statistic
vqsup["tschuprow"]
```

```
##          value
## groupe  0.816497
```

```
# Pearson statistic
vqsup["pearson"]
```

```
##          value
## groupe  0.816497
```

```
# Distances d'agrégation
dist = res_hcpc.distances_
# Histogram
import numpy as np
fig,axe = plt.subplots(figsize=(16,8))
axe.bar(np.arange(1,len(dist)+1),sorted(dist,reverse=True),color = "black")
plt.show()
```



Pour plus d'informations sur la classification hiérarchique combinée avec une analyse factorielle (HCPC) sous scientisttools, consulter le notebook

https://github.com/enfantbenidedieu/scientisttools/blob/master/notebooks/hcpc_pca_autos2005.ipynb.