UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

Apprentissage Statistique Avancé

Laurène DAVID – Safa HAMDAN – Allan PENDANT

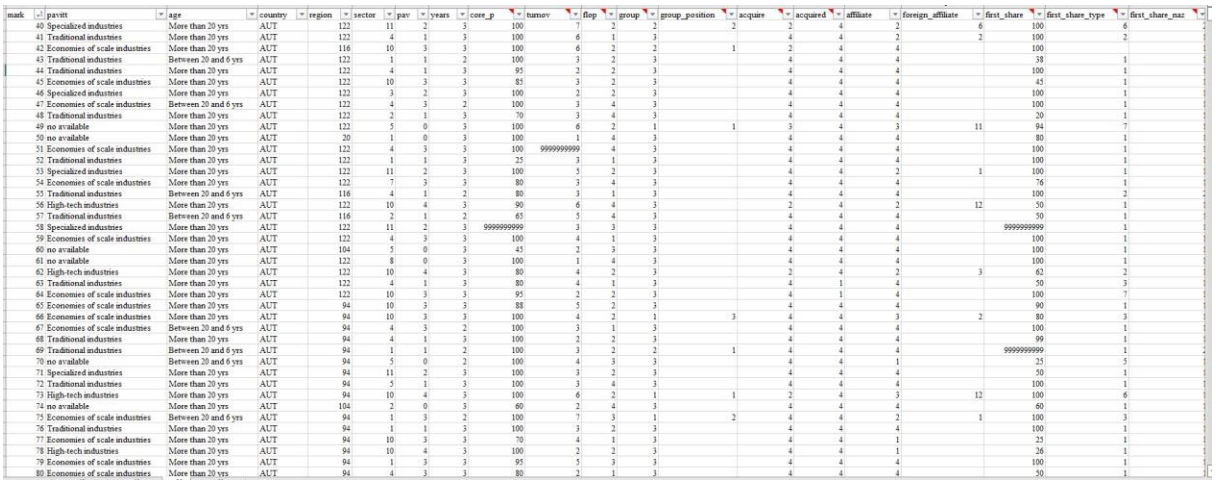
Juin 2022

Table des matières

I- Introduction

L'objectif ici est de faire apparaître les principales caractéristiques des entreprises européennes.

Ce jeu de données plus volumineux que le précédent comporte Initialement 14 759 observations et 489 variables.



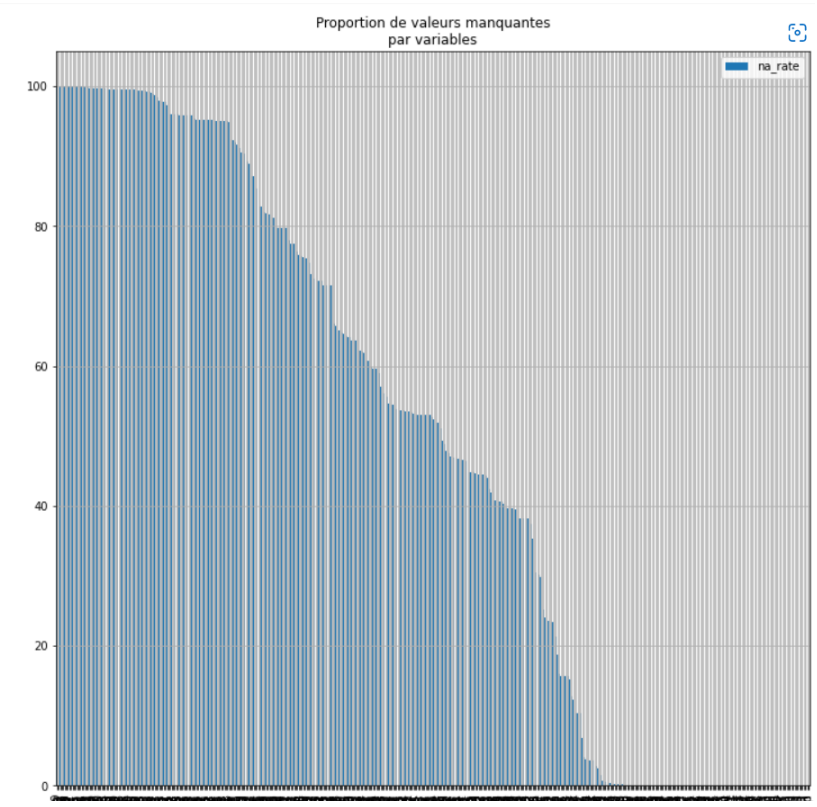
mark	pavitt	age	country	region	sector	pav	years	core_p	turnov	flop	group	group_position	acquire	acquired	affiliate	foreign_affiliate	first_share	first_share_type	first_share_naz
40	Specialized industries	More than 20 yrs	AUT	122	11	2	3	100	7	2	2	2	2	4	2	6	100	6	
41	Traditional industries	More than 20 yrs	AUT	122	4	1	3	100	6	1	3		4	4	2	2	100	2	
42	Economies of scale industries	More than 20 yrs	AUT	116	10	3	3	100	6	2	2	1	2	4	4		100		
43	Traditional industries	Between 20 and 6 yrs	AUT	122	1	1	2	100	3	2	3		4	4	4		38	1	
44	Traditional industries	More than 20 yrs	AUT	122	4	1	3	95	2	2	3		4	4	4		100	1	
45	Economies of scale industries	More than 20 yrs	AUT	122	10	3	3	85	3	2	3		4	4	4		45	1	
46	Specialized industries	More than 20 yrs	AUT	122	3	2	3	100	2	2	3		4	4	4		100	1	
47	Economies of scale industries	Between 20 and 6 yrs	AUT	122	4	3	2	100	3	4	3		4	4	4		100	1	
48	Traditional industries	More than 20 yrs	AUT	122	2	1	3	70	3	4	3		4	4	4		20	1	
49	no available	More than 20 yrs	AUT	122	5	0	3	100	6	2	1	1	3	4	3	11	94	7	
50	no available	More than 20 yrs	AUT	20	1	0	3	100	1	4	3		4	4	4		80	1	
51	Economies of scale industries	More than 20 yrs	AUT	122	4	3	3	100	9999999999	4	3		4	4	4		100	1	
52	Traditional industries	More than 20 yrs	AUT	122	1	1	3	25	3	1	3		4	4	4		100	1	
53	Specialized industries	More than 20 yrs	AUT	122	11	2	3	100	5	2	3		4	4	2	1	100	1	
54	Economies of scale industries	More than 20 yrs	AUT	122	7	3	3	80	3	4	3		4	4	4		76	1	
55	Traditional industries	Between 20 and 6 yrs	AUT	116	4	1	2	80	3	1	3		4	4	4		100	2	
56	High-tech industries	More than 20 yrs	AUT	122	10	4	3	90	6	4	3		2	4	2	12	50	1	
57	Traditional industries	Between 20 and 6 yrs	AUT	116	2	1	2	65	5	4	3		4	4	4		50	1	
58	Specialized industries	More than 20 yrs	AUT	122	11	2	3	9999999999	3	3	3		4	4	4		9999999999	1	
59	Economies of scale industries	More than 20 yrs	AUT	122	4	3	3	100	4	1	3		4	4	4		100	1	
60	no available	More than 20 yrs	AUT	104	5	0	3	45	2	3	3		4	4	4		100	1	
61	no available	More than 20 yrs	AUT	122	8	0	3	100	1	4	3		4	4	4		100	1	
62	High-tech industries	More than 20 yrs	AUT	122	10	4	3	80	4	2	3		2	4	2	3	62	2	
63	Traditional industries	More than 20 yrs	AUT	122	4	1	3	80	4	1	3		4	1	4		50	3	
64	Economies of scale industries	More than 20 yrs	AUT	122	10	3	3	95	2	2	3		4	1	4		100	7	
65	Economies of scale industries	More than 20 yrs	AUT	94	10	3	3	88	5	2	3		4	4	4		90	1	
66	Economies of scale industries	More than 20 yrs	AUT	94	10	3	3	100	4	2	1	3	4	4	3	2	80	3	
67	Economies of scale industries	Between 20 and 6 yrs	AUT	94	4	3	2	100	3	1	3		4	4	4		100	1	
68	Traditional industries	More than 20 yrs	AUT	94	4	1	3	100	2	2	3		4	4	4		99	1	
69	Traditional industries	Between 20 and 6 yrs	AUT	94	1	1	2	100	3	2	2	1	4	4	4		9999999999	1	
70	no available	Between 20 and 6 yrs	AUT	94	5	0	2	100	4	3	3		4	4	1		25	5	
71	Specialized industries	More than 20 yrs	AUT	94	11	2	3	100	3	2	3		4	4	4		50	1	
72	Traditional industries	More than 20 yrs	AUT	94	5	1	3	100	3	4	3		4	4	4		100	1	
73	High-tech industries	More than 20 yrs	AUT	94	10	4	3	100	6	2	1	1	2	4	3	12	100	6	
74	no available	More than 20 yrs	AUT	104	2	0	3	60	2	4	3		4	4	4		60	1	
75	Economies of scale industries	Between 20 and 6 yrs	AUT	94	1	3	2	100	7	3	1	2	4	4	2	1	100	3	
76	Traditional industries	More than 20 yrs	AUT	94	1	1	3	100	3	2	3		4	4	4		100	1	
77	Economies of scale industries	More than 20 yrs	AUT	94	10	3	3	70	4	1	3		4	4	1		25	1	
78	High-tech industries	More than 20 yrs	AUT	94	10	4	3	100	2	2	3		4	4	1		26	1	
79	Economies of scale industries	More than 20 yrs	AUT	94	1	3	3	95	5	3	3		4	4	4		100	1	
80	Economies of scale industries	More than 20 yrs	AUT	94	4	3	3	80	2	1	3		4	4	4		50	1	

1- Pré-analyses des valeurs manquantes

A première vue, cette base de donnée semble comporter un nombre conséquent de valeurs manquantes. Effectuons une analyse via Pandas pour voir ce qu'il en est.

	column_name	na_rate
346	Unnamed: 349	100.000000
294	d46b_spe_msf_18	99.952571
289	d46b_spe_msf_13	99.952571
282	d46b_spe_msf_6	99.952571
283	d46b_spe_msf_7	99.952571
...	...	...
113	bank_fin	0.000000
114	public_fin	0.000000
115	leasing_fin	0.000000
116	other_fin	0.000000
485	comp	0.000000

486 rows × 2 columns



En créant un tableau présentant le taux de valeurs manquantes par variables (rangé par ordre décroissant) et un graph explicitant ce tableau, nous nous apercevons qu'il sera nécessaire de choisir un seuil de valeurs manquantes.

Seulement 60 variables ne possèdent aucune valeurs manquantes. Sélectionner uniquement celles-ci reviendrait à mettre de côté 87% des variables disponibles. Pour réduire ce pourcentage, nous décidons de fixer un seuil à 5%. Ainsi nous obtenons 146 variables sur les 489 disponibles ainsi 70% des variables disponibles ne seront pas étudiées.

	column_name	na_rate
14	first_share	4.092418
347	external_financing	3.787519
27	strategy_evolution	3.577478
422	e3_m_c9	3.543601
26	strategy	3.536825
...	...	...
113	bank_fin	0.000000
114	public_fin	0.000000
115	leasing_fin	0.000000
116	other_fin	0.000000
485	comp	0.000000

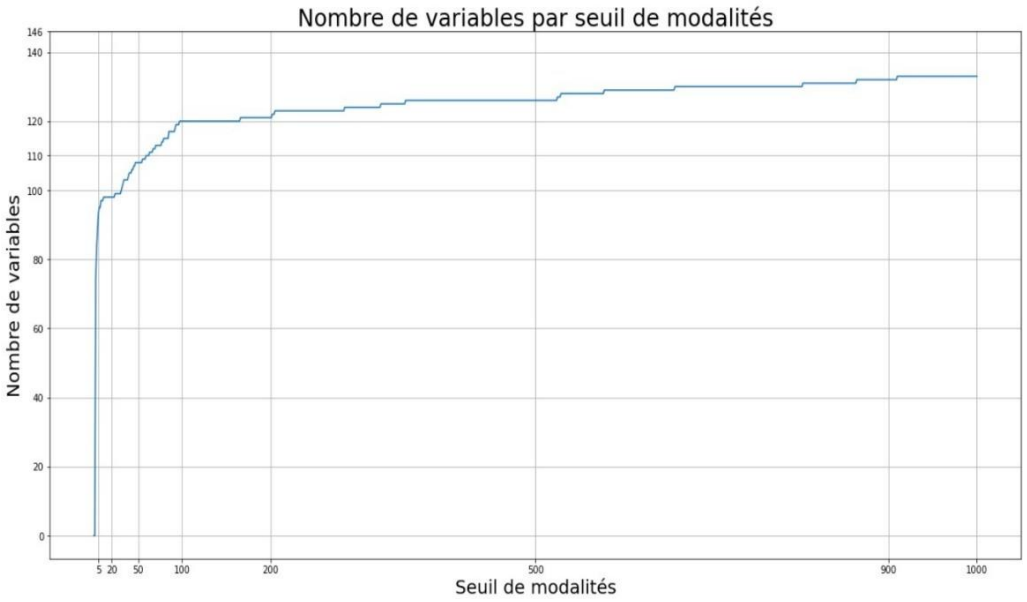
146 rows × 2 columns

Parmi toutes les variables numériques présentes, il doit y en avoir qui sont qualitatives (qualitatives numériques) et d'autres qui sont quantitatives

Pour pouvoir les détecter, étudions le nombre de modalités par variables en dressant un tableau.

	var_name	n_uniques
0	mark	13750
113	no_family_p	891
128	grad_p	838
115	involv_p	775
129	grad_n	637
...	...	...
66	d37_m_c2	2
65	c18_m_c2	2
64	d37_m_c3	2
63	exp_from_home	2
145	comp	2

146 rows × 2 columns



Avec ce graph affichant le nombre de variables en fonction d'un seuil de modalités fixé nous identifions ainsi 7 tranches.

Tranche 1 : variables à moins de 20 modalités

Tranche 2 : variables de 21 à 50 modalités

Tranche 3 : variables de 51 à 100 modalités

Tranche 4 : variables de 101 à 200 modalités

Tranche 5 : variables de 201 à 500 modalités

Tranche 6 : variables de 501 à 900 modalités

Tranche 7 : variables à plus de 900 modalités

En étudiant les variables de chaque tranches muni de leur descriptif, nous remarquons que les variables de la tranche 1 (moins de 20 modalités) semblent correspondre aux variables numériques qualitatives)

	var_name	n_uniques
127	sector	11
8	first_share_type	8
23	turnov	8
21	ceo_age	8
22	e10	6
12	f3	6
25	e8	6
126	pav	5
39	acquire	5
37	affiliate	5
30	e15	5
28	flop	5
3	strategy_evolution	4
17	d33	4
15	d27	4
29	exp_story_dummy	4

Extrait des variables de la Tranche 1

2- Analyse descriptive

Maintenant que nous avons fini notre préanalyse sur notre dataset, nous allons décider des variables qualitatives et quantitatives grâce aux fonctions qualitatives_numeriques() et quantitatives() que nous avons implémenté, qui traitent les variables par types et nombre de modalités.

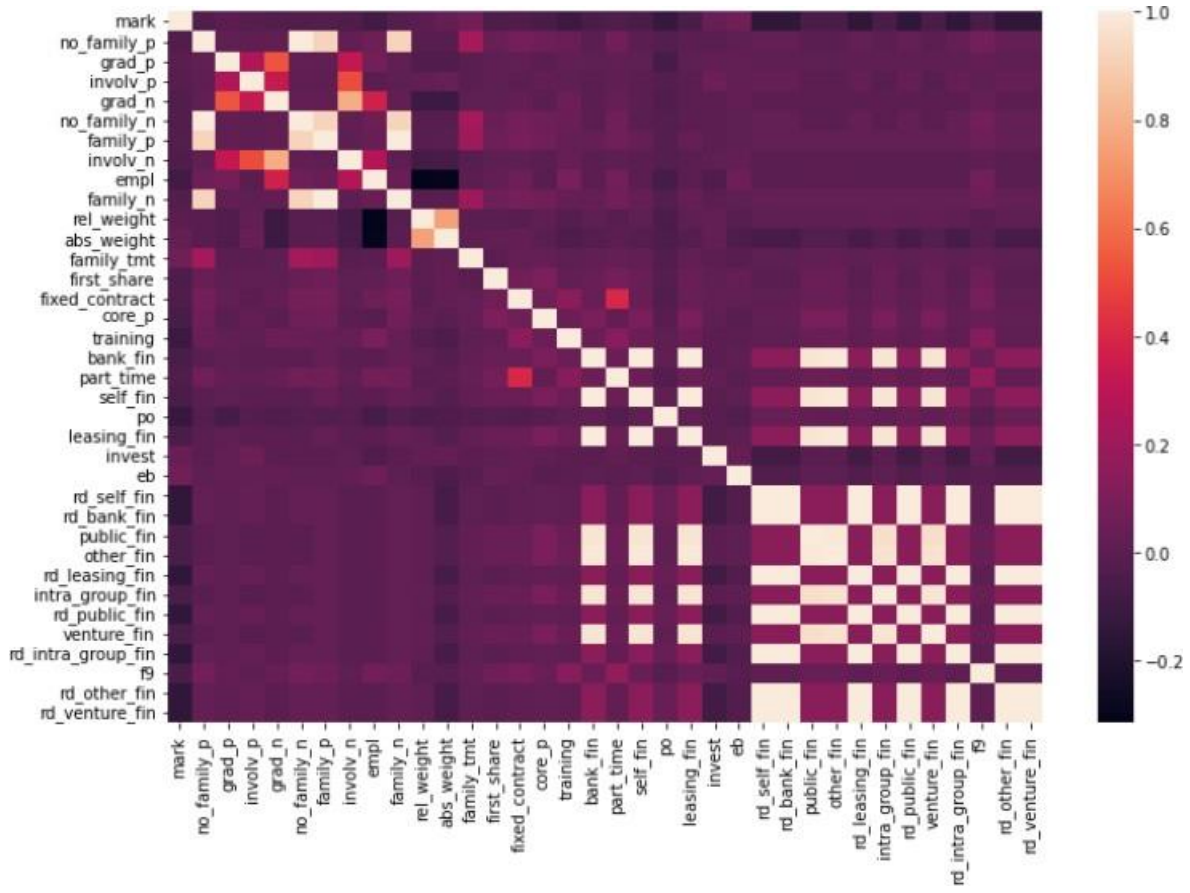
Les variables quantitatives sont choisies par type 'int' et 'float' et les variables qualitatives sont choisis par type 'objet' et 'str' ainsi que les variables numériques contenant moins de 20 modalités de type 'int' et 'float'.

Après avois classé chaque variables comme qualitatives ou quantitatives, nous avons décidé de les nettoyer.

a- Variables quantitative

Après avoir appliqué la fonction quantitatives() sur notre jeu de données (on appellera ce nouveau dataframe df_quant), on se retrouve avec 36 colonnes et 13 750 observations.

Nous avons d'abord voulu regarder les corrélations entre les variables :

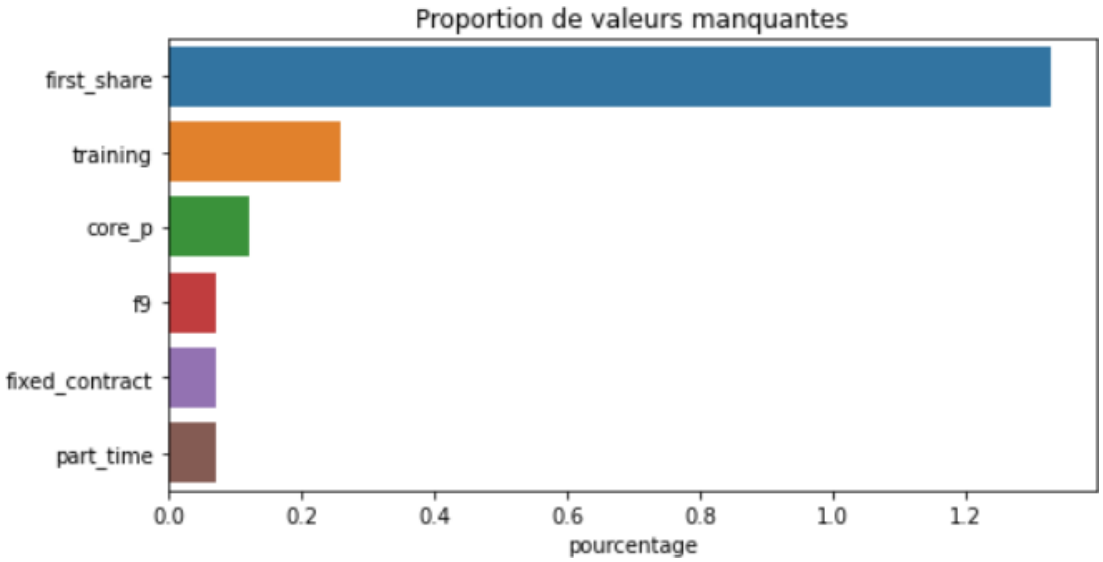


On observe une corrélation plus qu'évidente entre certaines variables, nous avons donc décidé de retirer des variables qui étaient très corrélées aux autres. Nous nous retrouvons maintenant avec 18 colonnes.

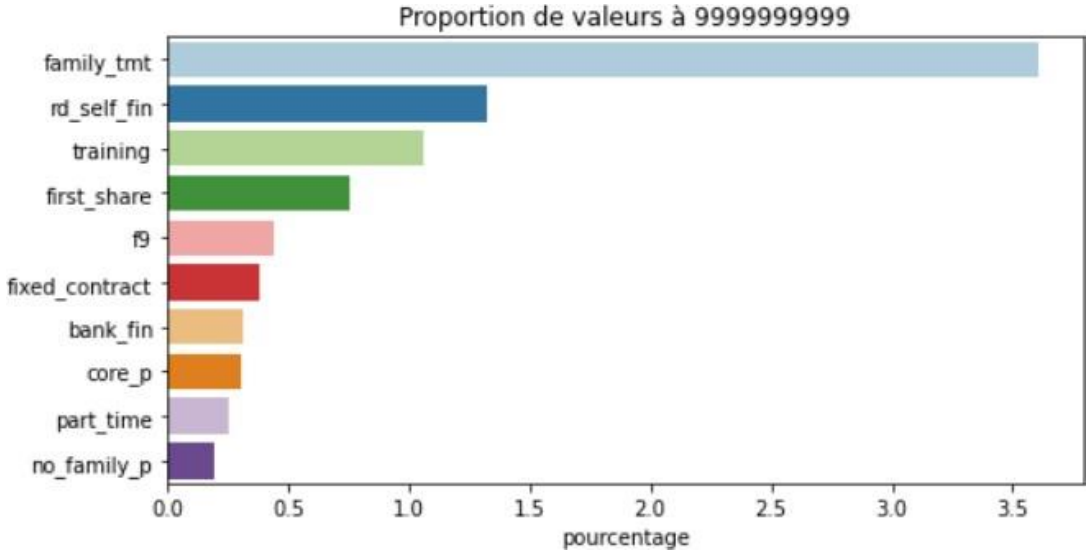
Nous avons maintenant voulu nous pencher sur les valeurs manquantes/ incohérentes avec leurs descriptions.

En effet dans notre jeu de données, nous avons observé des valeurs égales à '99999999' qui signifie 'Don't Know/Didn't Answer' en plus des valeurs vides/NaN. Parmi les valeurs incohérentes, nous avons aussi compté, pour les variables renvoyant des pourcentages, la valeurs au-dessus de 100% car leurs descriptions spécifiaient que celles-ci devaient aller de 0 à 100, donc toutes valeurs différentes ont été interprétées comme incohérentes.

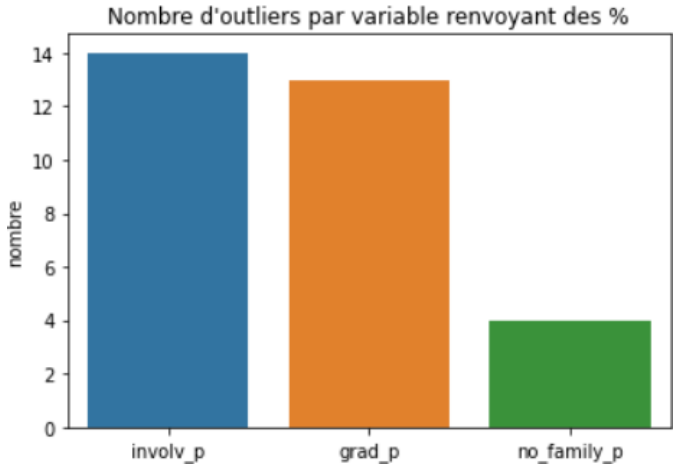
Nous avons d'abord calculé la proportion des valeurs NaN pour chacune des variables de df_quant :



Ensuite, la proportion de valeurs '99999999' :



Et pour finir, le nombre de valeurs, pour les variables renvoyant des pourcentages, au-dessus de 100% :

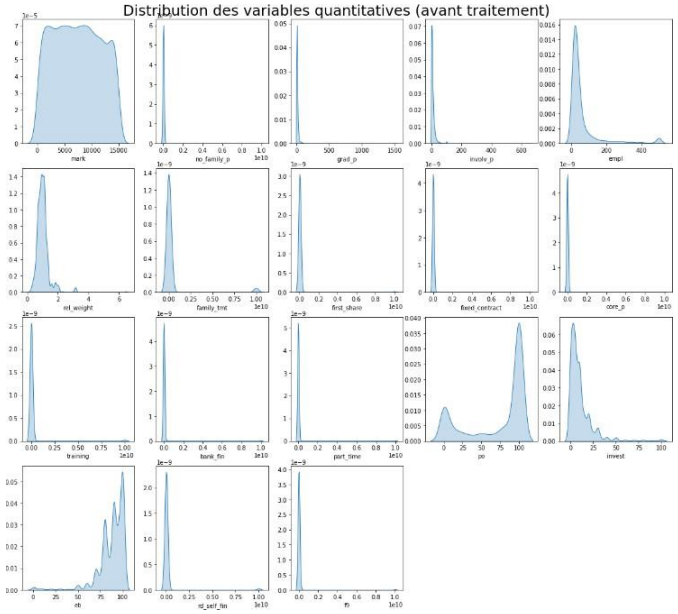


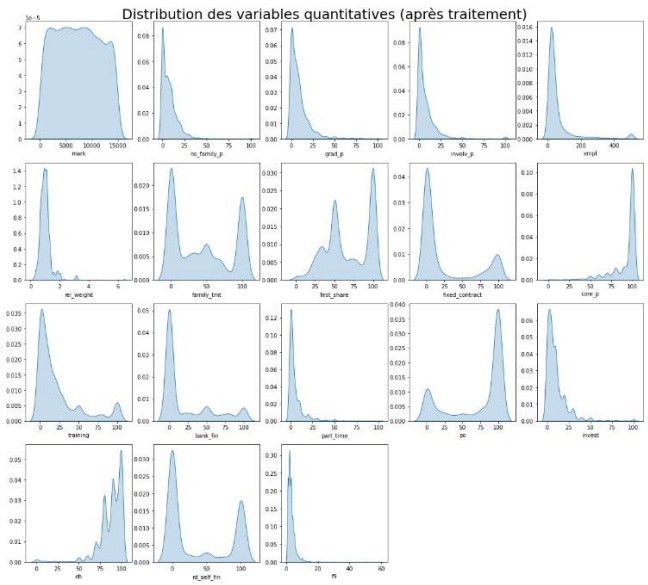
	involv_p	grad_p	no_family_p
mean	8.09	9.82	8.18
min	0.00	0.00	0.00
25%	0.00	0.00	0.00
50%	3.33	5.56	5.36
75%	10.00	11.76	11.11
max	666.67	1541.67	520.00

Les observations ayant le plus de valeurs incohérentes ont été supprimées du dataset, on se retrouve donc avec 13 724 observations.

Pour les valeurs manquantes, ainsi que les valeurs égales à '99999999', nous les avons remplacé en utilisant une méthode basée sur le k-NN. Nous avons utilisé la fonction 'KNNImputer' qui remplit les valeurs manquantes en regardant les observations similaires à celle manquante et qui la remplace par la valeur la plus récurrente parmi toutes les observations similaires.

Voici ce que donne leurs distributions après traitement des valeurs NaN/ incohérentes :



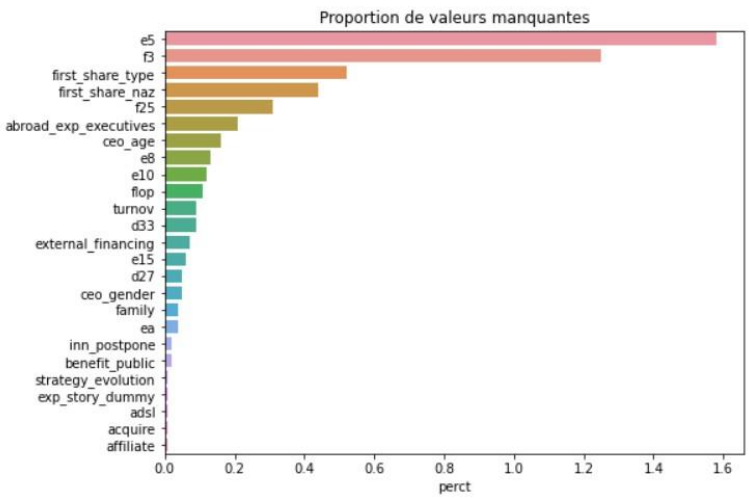


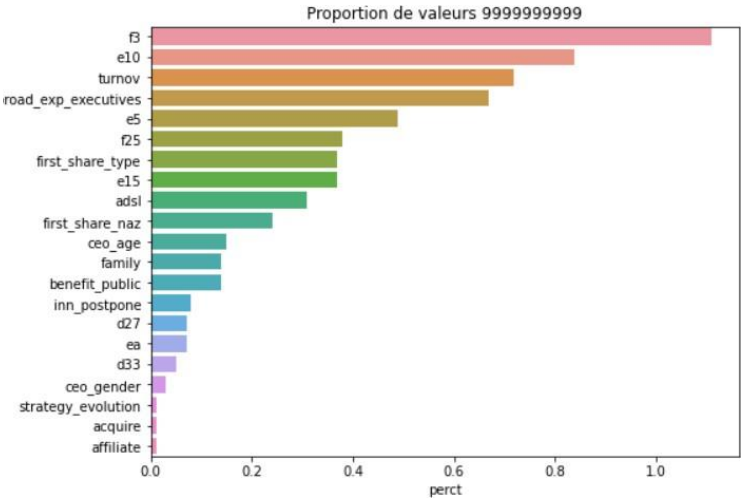
On voit que la distribution des valeurs est plus interprétable après le traitements des valeurs.

b- Variables qualitatives

Pour les variables qualitatives, en plus d'appliquer notre fonction qualitatives_numeriques(), nous avons rajouté les variables 'region','pavitt','age','country', quenous avons arbitrairement jugé comme qualitatives.

Ici aussi, nous avons voulu traiter les valeurs manquantes :





On voit que pour certaines variables, elles sont présentes en grand nombre. Dans ce cas, comme nous avons déjà un assez grand nombre de variables dans 'df_quali', nous avons décidé de totalement supprimer les variables.

La raison pour laquelle nous n'avons pas appliqué la méthode k-NN est que c'est une méthode qui s'applique strictement sur des datasets ne contenant que des variables numériques.

Si nous avions fait un One-HotEncoding sur ce dataset, les valeurs manquantes auraient été transformé en colonnes binaires et donc il n'existerait plus de valeurs manquantes à remplacer.

Il est donc préférable d'appliquer la méthode du k-NN uniquement sur les variables quantitatives et de supprimer les variables qualitatives contenant des valeurs manquantes.

II- Factor Analysis of Mixed Data (FAMD)

Dans cette prochaine partie, nous allons présenter la méthode de réduction de dimension que nous avons appliqué à notre jeu de données une fois nettoyé.

Contrairement à la première partie du projet sur la prédiction du défaut de paiement, nous n'avons pas pu exclure certaines variables par une méthode de sélection de modèles. Ces méthodes-là nécessitent de connaître la variable à prédire, ce qui n'est pas le cas ici.

Nous avons donc décidé de garder les variables qualitatives et quantitatives de notre jeu de données, ce qui nous a poussé à faire une FAMD.

1- Construction de la table de contingence

La FAMD ou Factor Analysis of Mixed Data est une méthode de réduction de dimension qui permet de traiter un jeu de données avec des variables qualitatives et quantitatives.

La première étape de cette méthode consiste à construire une table de contingence à partir des variables qualitatives. Nous avons remarqué que certaines variables étaient binaires et avaient déjà comme modalités 0/1. Nous avons décidé de les laisser dans la table de contingence final.

Pour les autres variables qualitatives, nous les avons transformé grâce à la fonction OneHotEncoder de sklearn.

Voici un petit échantillon des variables transformées par le OneHotEncoder.

	sector_1	sector_2	sector_3	sector_4	sector_5	sector_6	sector_7	sector_8	sector_9	sector_10	sector_11	pav_0
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
1	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00

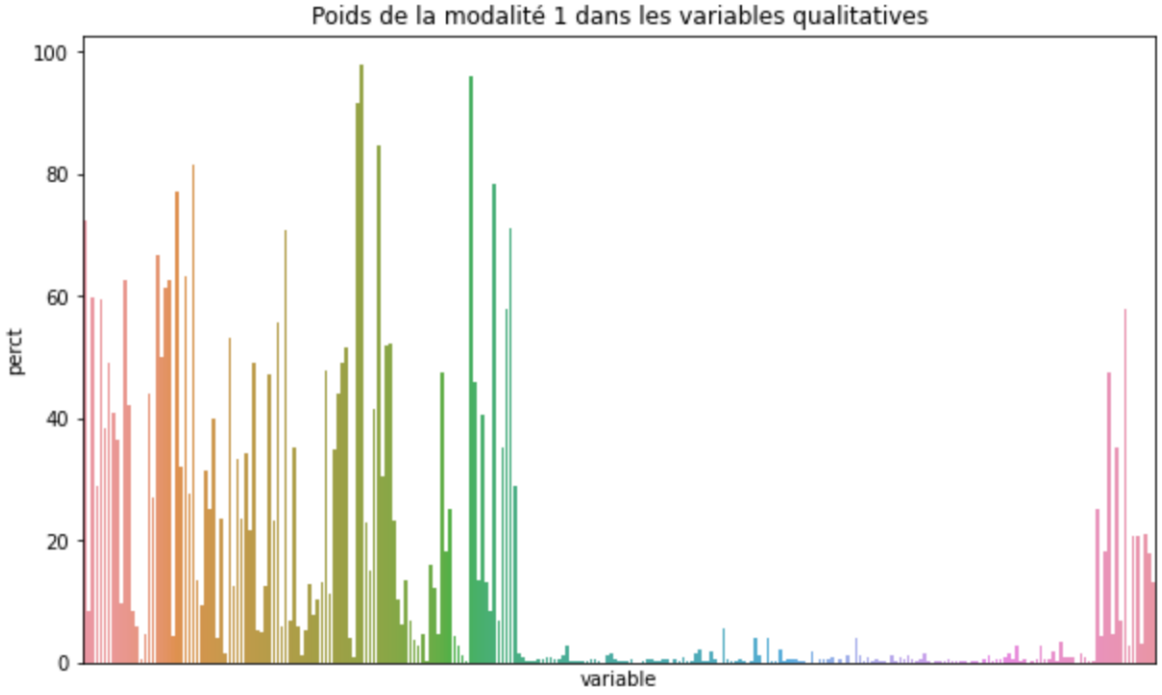
En regroupant le jeu de données construit à partir du OneHotEncoder avec les variables qualitatives que nous n'avons pas transformées, nous nous retrouvons avec 267 variables qualitatives.

```
df_quali_onehot.shape
```

```
(13724, 267)
```

Afin de réduire le nombre de variables à traiter par la FAMD, nous avons décidé d'étudier le poids de chaque variable dans le jeu de données totales.

Nous avons regardé le poids des modalités 1 dans chaque variable qualitative, ce qui nous donne le graph suivant.



Nous pouvons voir qu'une bonne partie des variables du tableau de contingence possèdent un pourcentage très faible de modalité 1.

Le maintien de ces variables peut potentiellement ajouté du bruit dans notre modèle, nous avons donc décidé de supprimer les variables avec un pourcentage inférieur à 2%. Cela nous permet aussi de réduire le nombre de variables qualitatives, et de limiter le problème de la grande dimension pour notre modèle.

Nous obtenons une table de contingence avec 125 variables qualitatives.

```
df_onehot_final.shape
```

```
(13724, 125)
```

2- Application de la FAMD

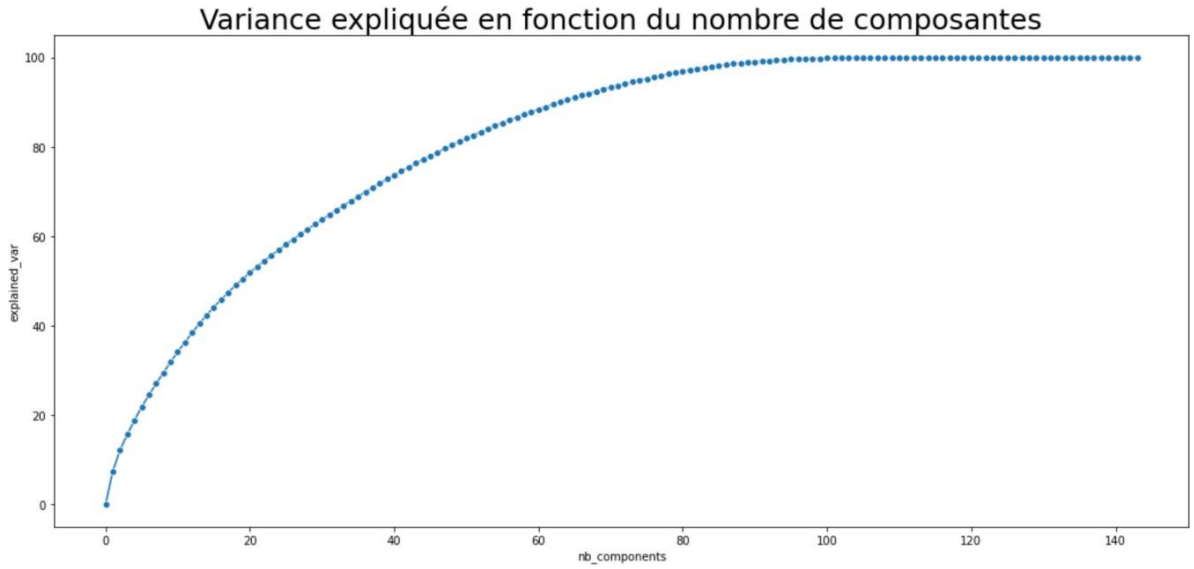
Nous avons regroupé dans un même jeu de données la table de contingence finale et les variables quantitatives restantes.

Nous pouvons maintenant normaliser les variables du jeu de données obtenues.

Pour les variables qualitatives, nous avons construit un vecteur avec la fréquence des variables du tableau de contingence. Chaque variable qualitative est ensuite divisée par la racine de la fréquence qui lui correspond. En ce qui concerne les variables quantitatives, nous les avons centrés puis réduites.

Nous avons fini par appliquer une Analyse en Composantes Principales sur le jeu de données normalisée. Les données ont projeté sur un espace construit à partir des axes principaux de l'ACP. La dimension du jeu de données final est de 143 variables.

En traçant la part de la variance expliquée en fonction du nombre de composantes, nous obtenons le graph suivant.



Pour atteindre 90% de variance expliquée, nous devons garder au moins 64 des 143 composantes de la FAMD. Nous pouvons le voir dans le dataset suivant.

	nb_components	explained_var
64	64	90.51
65	65	91.00
66	66	91.50
67	67	91.97
68	68	92.43

Comme dans le cas de notre MCA et du modèle de classification, nous allons choisir le nombre de composantes en fonction de la performance du clustering.

III- Clustering des données

1- Comparaison des méthodes

Nous allons maintenant comparer le clustering par K-means et le Clustering Hierarchique avec les linkages « average », « ward » et « complete ».

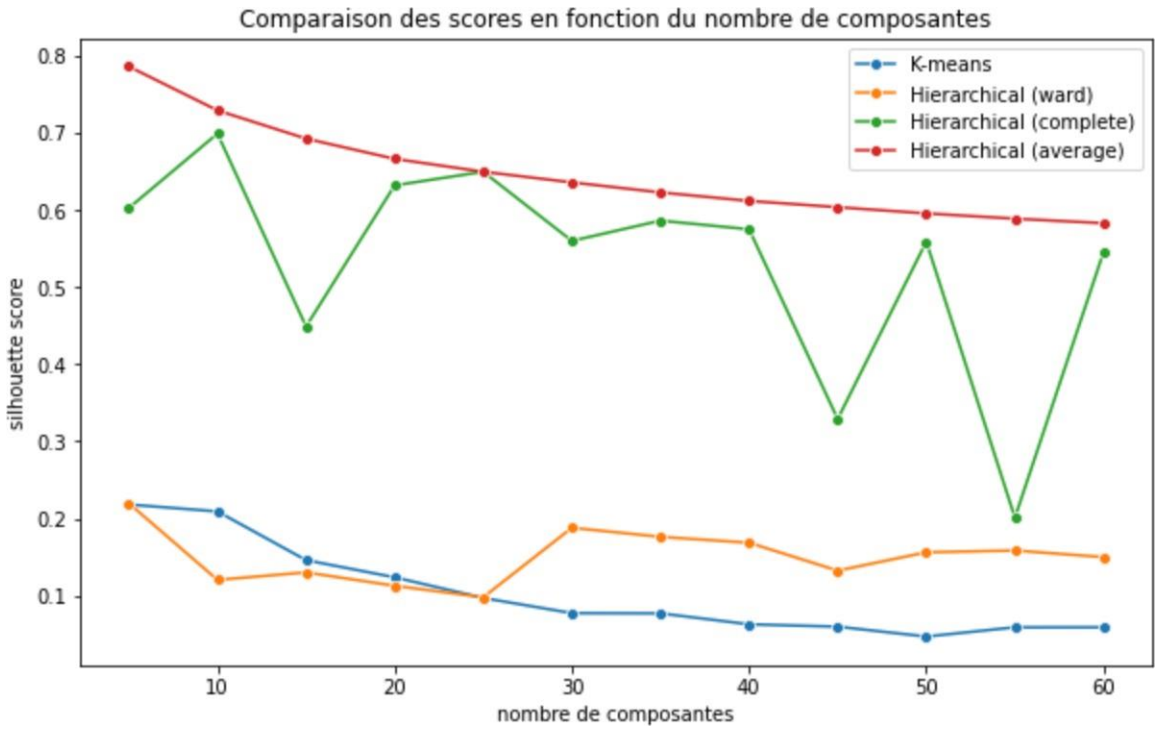
Les différents linkages du Hierarchical Clustering correspondent à des méthodes différentes pour définir la distance entre deux clusters. Nous expliquerons en détail plus tard la méthode choisit.

Pour comparer les méthodes de clustering, nous allons utiliser le silhouette score. Plus ce score est élevé, plus l'observation est proche des observations du cluster qu'on lui a donné. On peut aussi le voir comme un score de dissimilarité avec les autres clusters crées.

Pour chaque méthode testée et pour un nombre de composantes qui varie de 5 à 60, nous allons calculer le silhouette score moyen des observations.

Nous avons laissé les paramètres initiaux de chaque méthode, sauf le paramètre linkage du Clustering Hiérarchique qui nous a permis de tester les différentes méthodes de linkage.

Nous obtenons le graph suivant :



Nous constatons que les scores du Clustering Hierarchique avec un Average Linkage sont supérieur aux autres méthodes.

Le K-means et le Clustering Hiérarchique avec un ward linkage ont des scores très faibles. Les résultats du Clustering Hiérarchique avec complete linkage varie très fortement selon le nombre de composantes sélectionnée.

Nous avons donc décidé de sélectionner le Clustering Hiérarchique avec un Average Linkage.

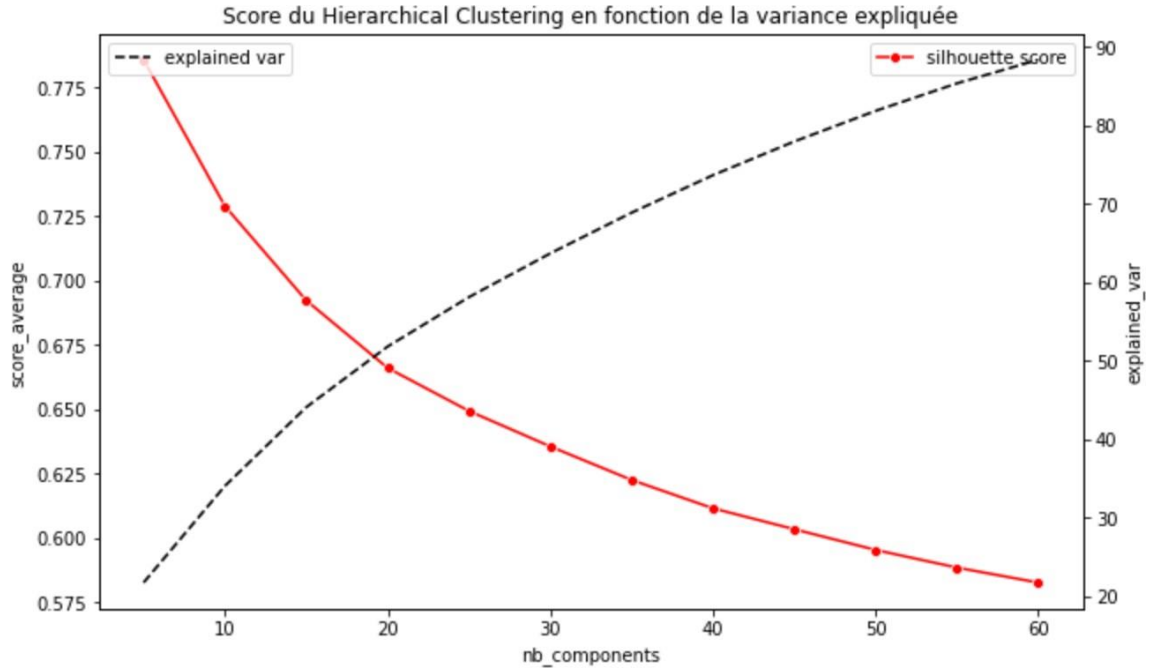
Le Average Linkage définit la distance entre deux clusters comme la distance moyenne entre les points du premier cluster et les points du second.

2- Choix du nombre de composantes

Nous avons remarqué dans le graph précédent que la performance du clustering semble être décroissante en fonction du nombre de composantes choisit.

Cependant, si nous décidons de ne sélectionner qu'un faible nombre de composantes de notre FAMD pour maximiser le silhouette score, nous risquons de ne garder qu'un faible pourcentage de la variance expliquée.

Pour choisir un nombre de composantes suffisant afin de maintenir la variance des données, tout en maximisant le score du clustering, nous avons tracé le graph suivant.



La courbe rouge correspond au silhouette score du Clustering Hiérarchique avec un average linkage. La courbe noire en pointillée correspond à la variance expliquée par le nombre de composantes choisis.

Les deux courbes se croisent autour de 20 composantes. En gardant ce nombre de composantes, nous obtenons un silhouette score de 0,67 et une part de variance expliquée à 51,85%.

	nb_components	score_average	explained_var
20	20	0.67	51.85

La part de la variance expliquée peut paraître assez faible. Nous avons fait le choix de privilégier un modèle de clustering avec un bon score, même si cela engendre une perte conséquente de la variance des données.

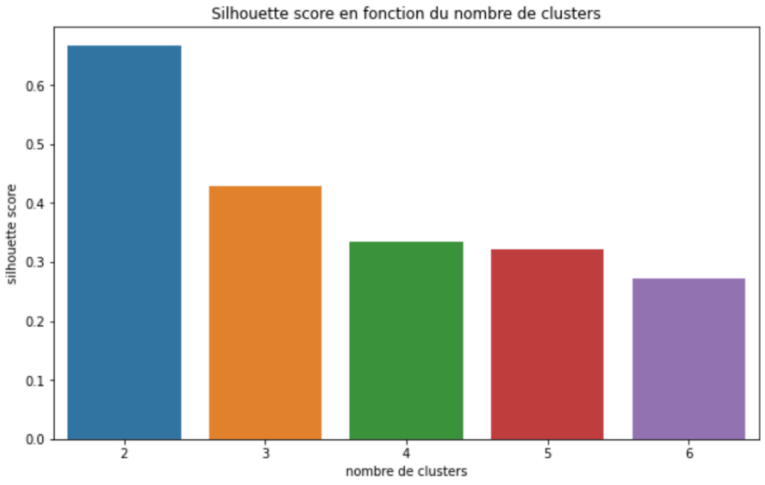
Nous avons aussi considéré que puisque le nombre d'observations du jeu de données est très important pour un clustering (plus de 10 000 observations), la perte d'information est moins problématique.

3- Choix du nombre de clusters

Maintenant que nous avons fixé le nombre de composantes à garder ainsi que la méthode de clustering que nous allons appliquer, il nous reste à choisir le nombre de clusters final.

Pour faire ce choix, nous allons aussi utiliser le silhouette score moyen que nous avons étudié en fonction du nombre de clusters.

Voici un graph et un dataframe qui représente les différents scores.



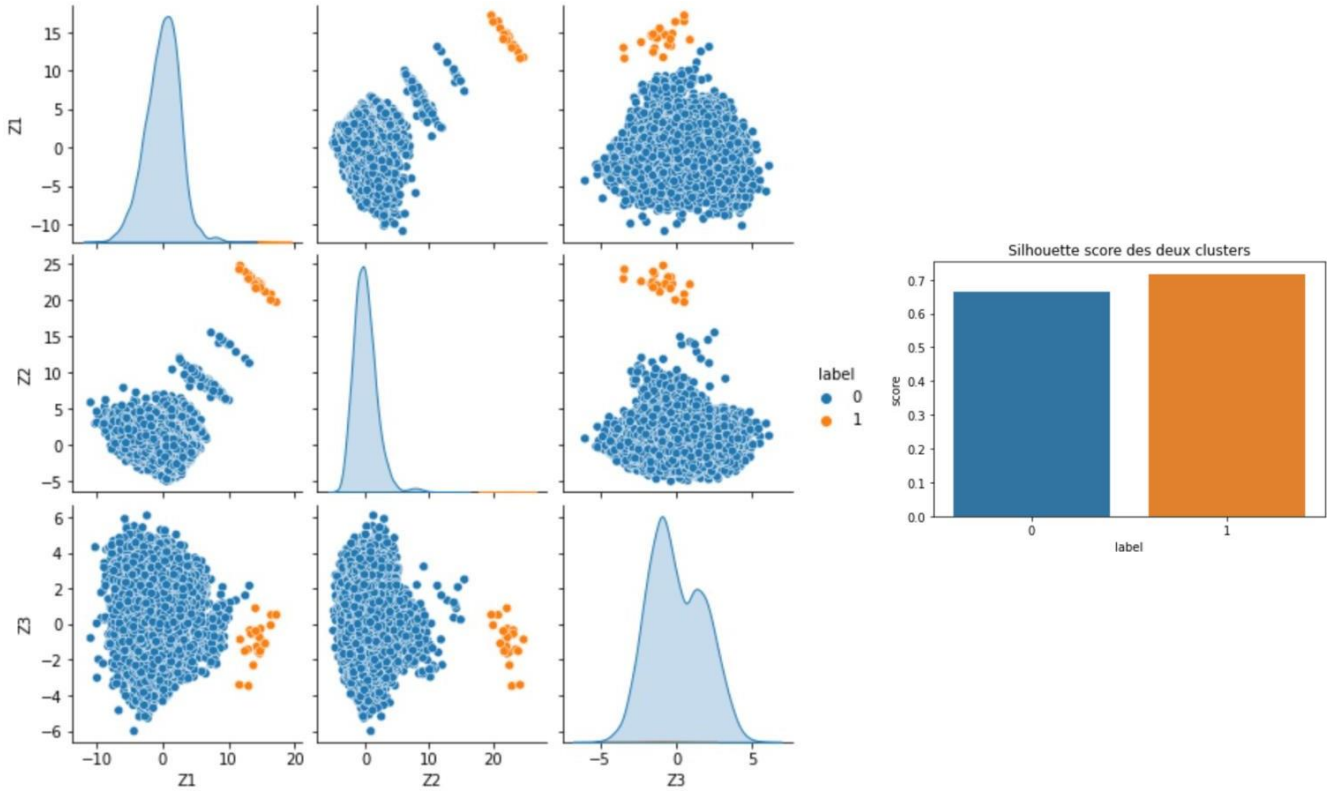
Nous avons constaté une chute assez importante du score du Hierarchical Clustering lorsqu'on passe de deux clusters à trois clusters. Ce score continue de décroitre lorsqu'on passe à quatre clusters.

	nb_clusters	silhouette score
0	2	0.67
1	3	0.43
2	4	0.33
3	5	0.32
4	6	0.27

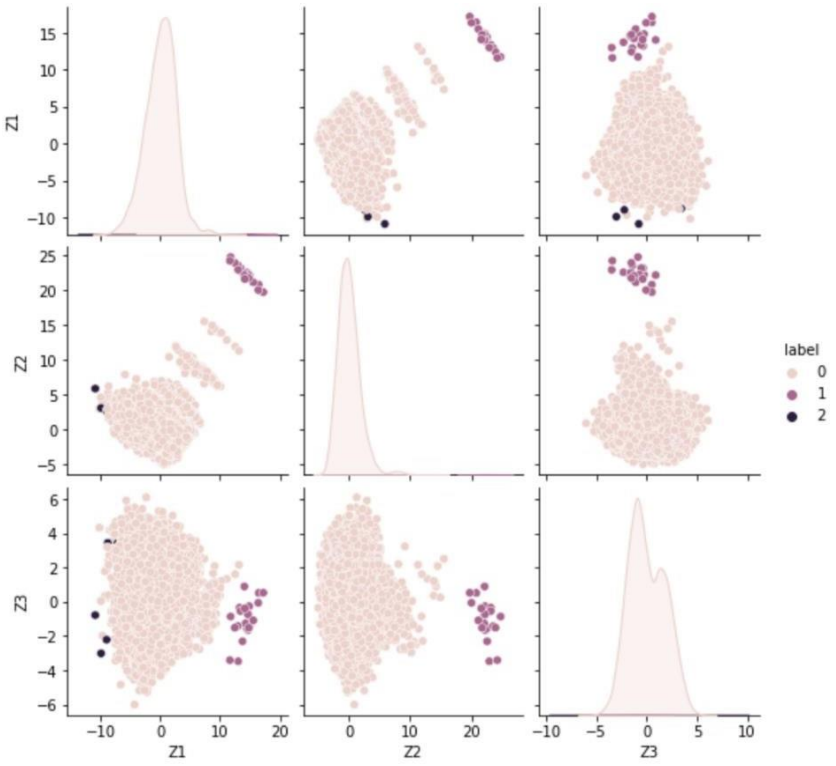
Nous avons tout de même décidé de comparer graphiquement le Clustering Hiérarchique avec average linkage avec 2,3 et 4 clusters construits.

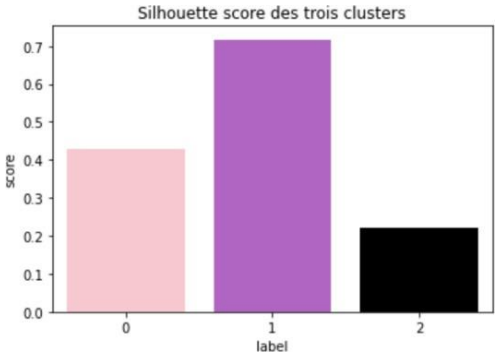
Nous avons représenté le nuage de points du clustering obtenu ainsi que le silhouette score moyen des clusters. Nous voulons noter que les nuages de points ont été construit qu'avec les trois premières composantes par souci de représentation

Nous obtenons les graphiques suivants en ayant choisi 2 clusters.

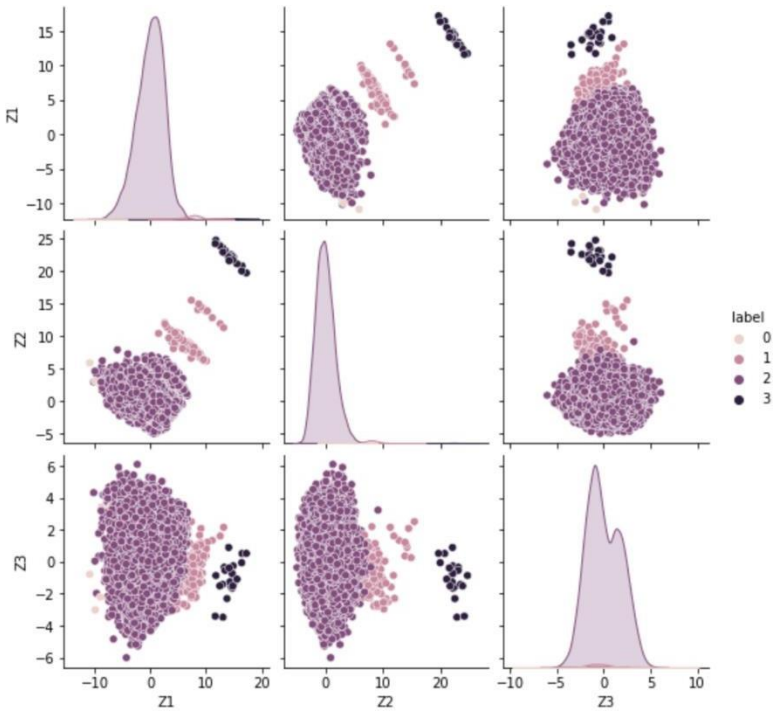


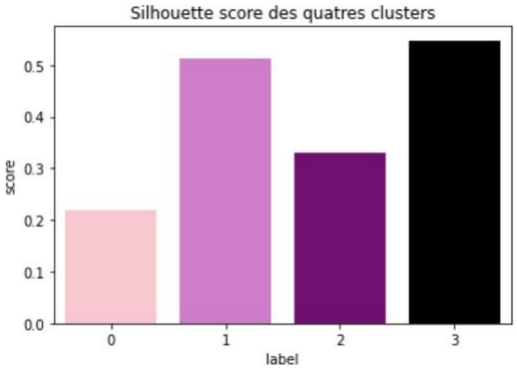
Pour trois clusters, nous obtenons :





Enfin, avec quatre clusters, nous obtenons :





Nous constatons que les deux clusters construits dans le premier ont des scores plus élevés que les autres et semblent plutôt équilibrés.

Au contraire lorsqu'on fixe le nombre de clusters à 3 et 4, les scores varient plus significativement entre les clusters. Les scores de ces clusters sont aussi inférieurs au premier clustering.

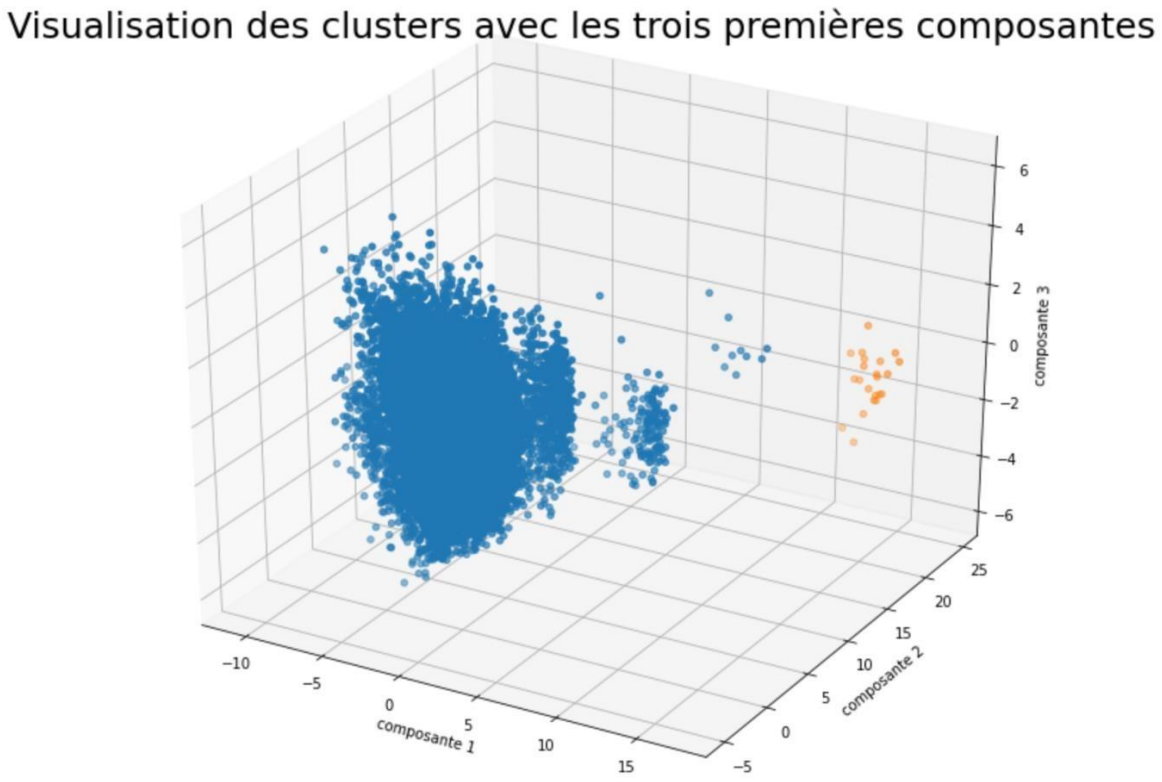
De plus, nous remarquons avec les nuages de points que seuls les premiers clusters construits sont séparables par une distance significative

Les trois clusters du deuxième cas sont très similaires à ceux construits précédemment, avec un très faible nombre d'observations dans le troisième cluster.

Enfin, pour le troisième cas avec quatre clusters, certains semblent très peu distancés. Par exemple, les observations avec les labels 2 et 3 sont pratiquement collées dans la majorité des nuages de points.

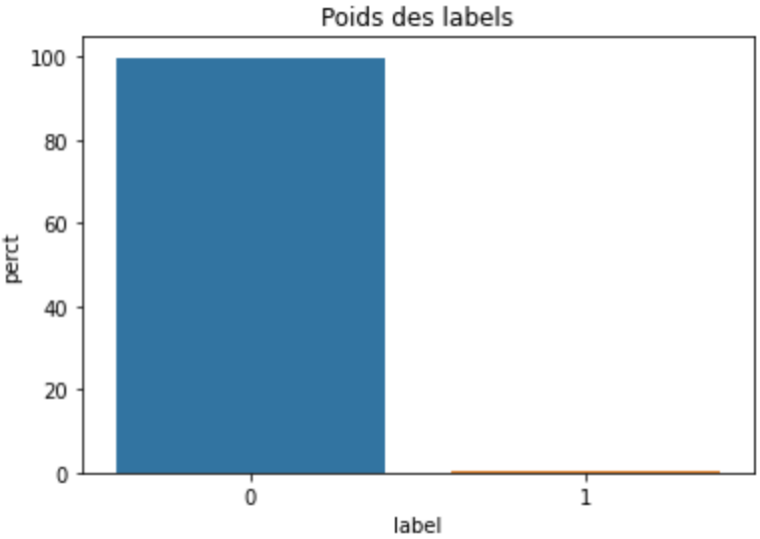
Pour conclure, nous avons décidé de fixer à deux le nombre de clusters du Clustering Hiérarchique.

Voici un graphique en trois dimensions de notre clustering final.



Les résultats de ce clustering sont relativement satisfaisant en regardant le silhouette score du modèle puisqu'il est à 0,67, ce qui est plutôt correct.

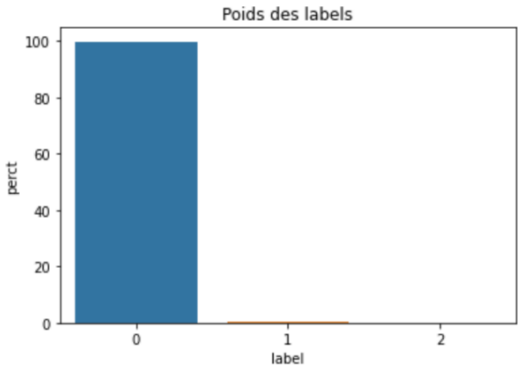
Cependant, nous pouvons noter que le poids des deux clusters n'est pas du tout équilibré, comme en témoigne ce graphique.

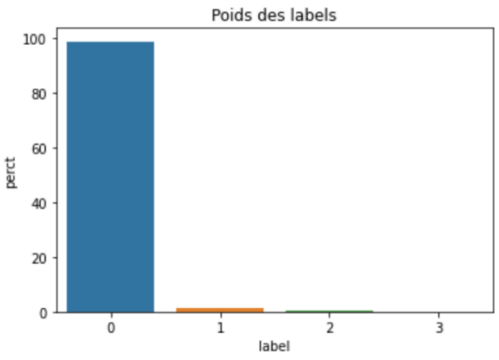


	index	perct
0	0	99.84
1	1	0.16

Le premier label représente plus de 99% des observations, contre moins de 1% pour le deuxième.

Ce problème ne s'améliore pas en augmentant le nombre de clusters, ce que montre les deux graphiques suivants. Ils correspondent au poids des labels lorsqu'on choisit de maintenir 3 et 4 clusters respectivement.





Conclusion

Nous pouvons conclure que notre modèle de clustering nous a permis de construire des clusters bien distinguables mais pas homogènes.

Nous pensons que cela est peut-être dû au modèle que nous avons choisi ou du fait que nous n'avons pas pu tester d'autres méthodes comme le Kernalized K-means et le K-memoirs, qui auraient peut-être mieux performé.

Pourtant, nous constatons un manque d'hétérogénéité dans nos données, qui viendrait potentiellement de notre FAMD.

Enfin, nous voulons ajouter que la transformation de nos variables en composantes par FAMD rend l'interprétation des clusters très difficile, voire impossible de par le nombre assez gros de n_composantes imposé par notre méthode de réduction de dimension.