Nous travaillons sous PYTHON notebook dans cette séance.

Supports de référence :

- [**Cours 1**] « Analyse factorielle de données mixtes » (Slides sur la page de cours : https://cours-machine-learning.blogspot.com/p/analyse-factorielle.html)
- [**Vidéo 1**] « Pipeline Python pour le déploiement », https://www.youtube.com/watch?v=-4tOZdlTT5U
- [**Vidéo 2**] « Classification automatique sur données mixtes », https://www.youtube.com/watch?v=Vhg0MKZ4t2E
- [**Vidéo 3**] « Clustering – Pipeline Python et déploiement » (scikit-learn + fanalysis) https://www.youtube.com/watch?v=H_A8-Dr8bIw

1 ACP + K-Means – Notion de Pipeline

Nous travaillons avec le fichier « **alcools.xlsx** ». Il recense les composés chimiques (teneur en méthanol, en butanol, etc.) de (n = 100) échantillons de liqueurs dans la feuille « **indiv actifs** ». Nous disposons de 3 individus supplémentaires dans la feuille « **indiv supplementaires** ». Nous avions déjà travaillé sur une variante de ces données dans un de nos précédents TD. J'ai légèrement modifié les valeurs.

1.1 Chargement et inspection des données

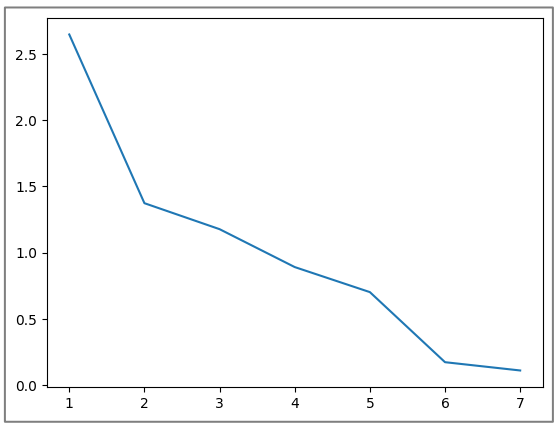
1. Chargez les individus actifs de notre base de données. Affichez-en les caractéristiques : nombre d'observations, de variables, leur type (on remarquera que toutes les variables sont quantitatives).
2. Calculez les statistiques descriptives. Que constatez-vous concernant les plages de valeurs, les moyennes, les écarts-type ?

1.2 Analyse en composantes principales (ACP)

3. Nous souhaitons réaliser une ACP normée sur nos données. Nous devons centrer et réduire nos variables tout d'abord. Nous utilisons la classe StandardScaler de la librairie « Scikit-Learn » (https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html). Vérifiez le type de la structure fournie par « fit_transform ». Calculez les moyennes et écarts-type des variables standardisées. Quelles doivent en être les valeurs ? (moyennes = 0, écarts-type = 1)
4. Dans un premier temps, nous souhaitons réaliser une ACP avec toutes les composantes possibles (https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html). Regardez

dans la documentation comment faire pour les produire, optez pour l'algorithme « `svd_solver = 'full'` ». A quoi correspond ce dernier paramètre ?

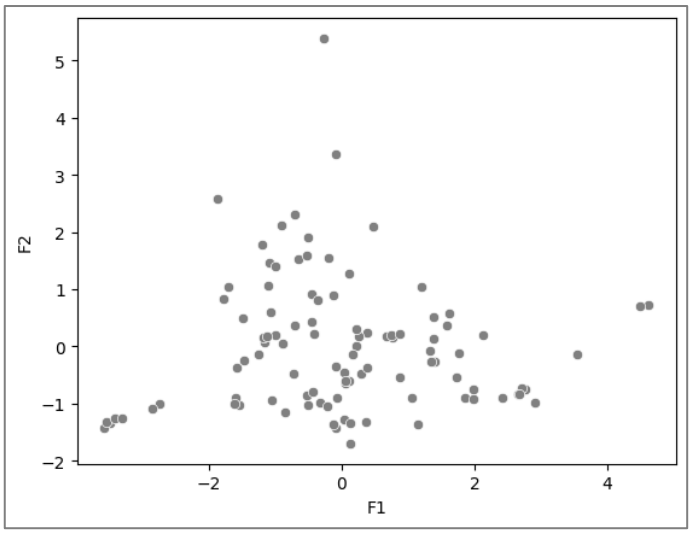
5. Créez un graphique avec en abscisse le nombre de composantes et en ordonnée la variance expliquée par chacune d'entre elles (avec la librairie « matplotlib » [https://matplotlib.org/] par exemple). Quel serait le bon nombre de composantes à retenir ?



6. A priori 2 composantes serait pas mal ? Pour conforter cette idée, calculez la proportion de variance restituée par ces deux premiers facteurs (56.84%)
7. Nous partons sur l'idée de 2 facteurs malgré tout (pourquoi malgré tout ?). Relancez l'ACP en conséquence et calculez les coordonnées des individus dans le premier plan factoriel. Affichez les premières valeurs.

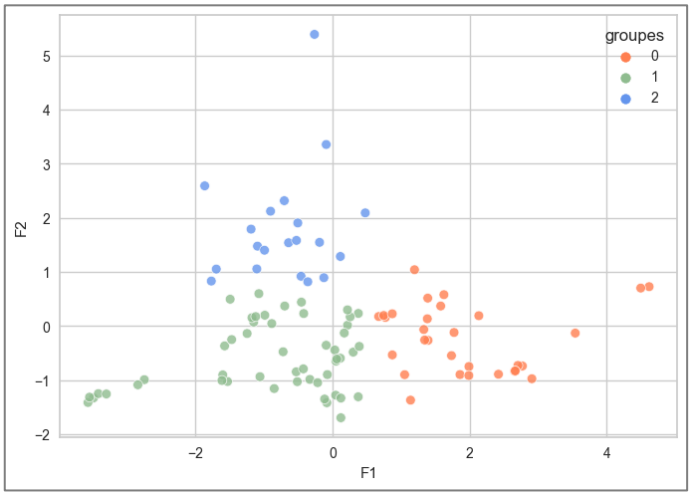
	F1	F2
0	-0.516114	-1.030670
1	0.031314	-0.444903
2	1.392829	-0.266363
3	-1.061051	-0.938873
4	1.382233	0.131638

8. Projetez les individus dans le premier plan factoriel (voir la librairie « Seaborn », https://seaborn.pydata.org/generated/seaborn.scatterplot.html ; vous trouverez sur un autre site des indications pour explorer les options sympathiques du package : https://python-charts.com/correlation/scatter-plot-seaborn/). A ce stade, est-ce qu'il y a des informations qui nous sauteraient aux yeux, notamment en ce qui concerne l'existence de groupes ?



1.3 K-Means sur les axes factoriels

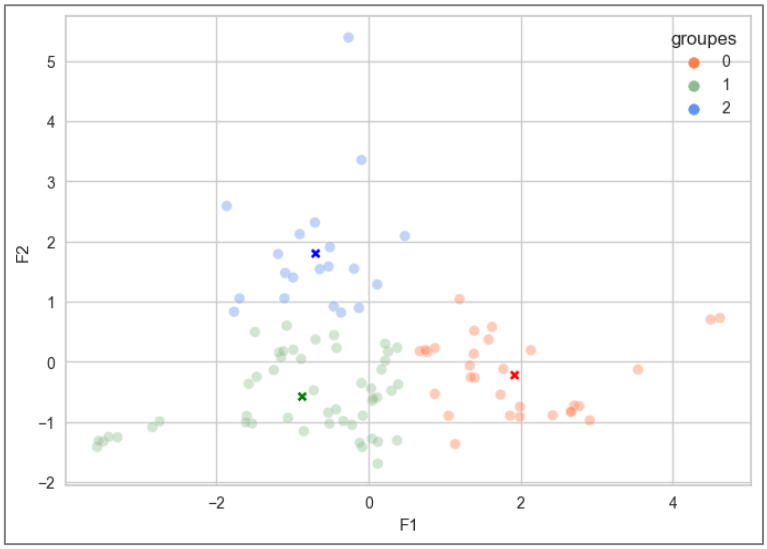
9. Outre le nombre de composantes à utiliser, l'autre inconnue est le nombre de groupes à produire avec l'algorithme de clustering. Mettons (parce que je connais la solution tout simplement pour le fichier des « Alcools », notre objectif est ailleurs dans ce TD) que nous partons sur (**K = 3**) classes. Instanciez la méthode des KMeans (https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html) avec (n_clusters = 3). Spécifiez (random_state = 0) pour que nous ayons tous le même résultat. Entraînez l'algorithme sur les 2 premiers facteurs issus de l'ACP.
10. Quels sont les effectifs des groupes ? (30, 50, 20)
11. Affichez de nouveau les points dans le premier plan factoriel en les illustrant selon leur classe d'appartenance.



12. Affichez les coordonnées des barycentres conditionnels (elles sont fournies par la classe de calcul, nous pouvons également les calculer ex-post).

groupes	F1	F2
0	1.920305	-0.226437
1	-0.873434	-0.582737
2	-0.696873	1.796498

13. Faites figurer ces barycentres dans le graphique précédent (on peut enchaîner des scatterplots « seaborn » dans la même cellule d'un notebook pour empiler les graphiques).

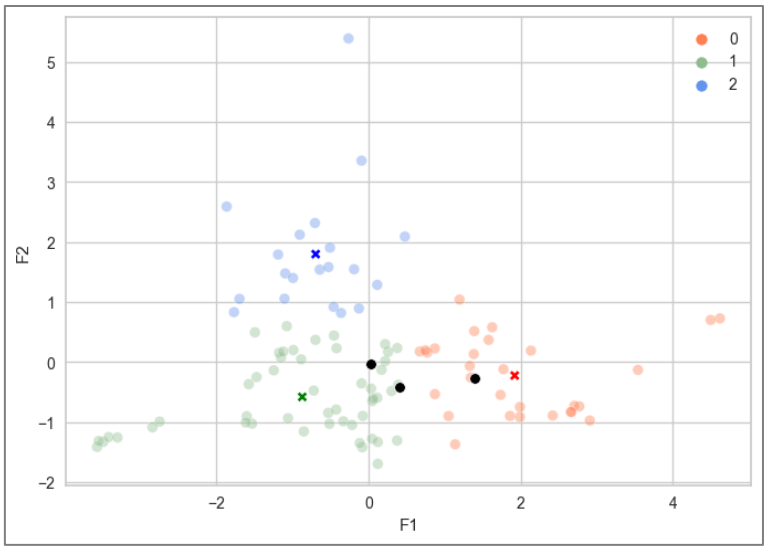


1.4 Traitement des individus supplémentaires

14. Chargez la feuille « indiv supplementaires » dans un nouveau data frame. Affichez les valeurs.
15. **Appliquez** le centrage réduction en utilisant les moyennes et écarts-type calculés sur les individus actifs (voir la fonction appropriée dans la doc de StandardScaler, https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html).
16. Calculez les coordonnées de ces individus dans le premier plan factoriel (même commentaire ici, il suffit d'appliquer la transformation).

	F1	F2
0	1.392829	-0.266363
1	0.409176	-0.412988
2	0.028218	-0.037222

17. Adjoignez ces individus dans le graphique ci-dessus pour les situer par rapport aux groupes et aux centres de classes (cf. les points en noir dans le plan factoriel)



18. A quelles classes pourrait-on rattacher chacun de ces individus supplémentaires ? (voir la doc pour identifier la fonction à utiliser : https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html ; pour situer les affectations, aidez-vous des coordonnées factorielles des points supplémentaires que nous avions calculés ci-dessus). Les résultats sont-ils cohérents ? (je veux mon neveu).
19. Est-ce qu'il serait possible de calculer les distances aux centres de classes de ces individus supplémentaires ? (cf. la doc encore une fois). Le résultat est-il aussi cohérent ici ? (mon neveu confirme derechef)

```
array([[0.5289847 , 2.28823919, 2.93636652],
       [1.52260051, 1.29379341, 2.47086447],
       [1.90152393, 1.05383247, 1.97187368]])
```

1.5 Création d'un Pipeline « Scikit-learn »

Le mécanisme des « pipeline » permet d'enchaîner des traitements « Scikit-Learn » en les empilant dans une structure ad hoc (voir **Vidéo 1**). Ce mécanisme rend transparent les différentes étapes du processus de machine learning. Les appels aux fonctions sont condensés aux méthodes de la classe Pipeline (voir « Methods » sur le site https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html).

20. Empilez l'ensemble des traitements, en créant de nouvelles instances avec les mêmes caractéristiques, dans une structure pipeline (inspirez-vous de **Vidéo 1, 19:20**).
21. Entraînez le tout (en prenant en entrée les données originelles, sans aucune transformation) et effectuez la prédiction sur les observations actives (regardez dans la doc la méthode qui

permet d'enchaîner directement les deux opérations). Les groupes obtenus sont-ils identiques à ceux où nous effectuons les traitements pas-à-pas (oui)

22. Appliquez le pipeline sur les individus supplémentaires, là aussi sans aucune transformation préalable. Les classes prédites sont-elles les mêmes que précédemment (oui, oui, on a dit)
23. Faites de même concernant les distances des individus supplémentaires aux centres de classes. Retrouve-t-on les mêmes valeurs avec le Pipeline (oui définitivement)

1.6 Sauvegarde dans un fichier pickle

24. Il nous reste enfin à sauvegarder notre modèle dans un fichier « pickle » (**Vidéo 3**, 40:00).
25. Question subsidiaire (à faire chez vous si le cœur vous en dit) : comment pourrait-on exploiter ce modèle dans une application de déploiement ? (ex. avec une API REST - https://www.youtube.com/watch?v=0HaRdXCsOH4, ou encore avec une application Streamlit - https://www.youtube.com/watch?v=qCixiQgxSkI, …)

2 AFDM + CAH (variables actives hétérogènes)

Nous disposons de (n = 270) individus dans la feuille « **var actives** » du fichier « **heart_afdm_clustering.xlsx** ». Nous souhaitons réaliser une classification automatique sur ce dataset où certaines variables sont quantitatives, d'autres qualitatives. Le traiter directement avec une CAH n'est pas possible, nous passons par une étape intermédiaire, l'analyse factorielle des données mixtes (**AFDM**) (**Cours 1**), pour projeter les individus dans un espace continu, nous pourrons dès lors appliquer la CAH sur une sélection des composantes les plus informatives.

Nous considérons qu'il n'y a pas de package qui implémente directement l'AFDM sous Python (si en fait, mais nous faisons comme si nous ne connaissons pas). Il va falloir préparer nous-mêmes les données avant de les présenter à une librairie d'analyse en composantes principales (ACP) (voir **Cours 1,** pages 6 à 12, **page 8 en particulier** ; j'utilise un code R dans le support, nous l'effectuons en Python dans notre TD, ce serait trop facile sinon…).

2.1 Importation

26. Chargez la feuille « **var actives** » de notre fichier Excel. Enumérez les variables et leurs types : lesquelles sont quantitatives (numériques) ? (int ou float : age, pression, cholesterol, taux_max, depression, pic) Qualitatives ? (object : sexe, type_douleur, sucre, angine, vaisseau)

2.2 Préparation des données en vue de l'AFDM

2.2.1 Variables quantitatives

27. Isolez les variables quantitatives dans une structure spécifique (pour isoler des variables selon leur type, voir : https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.select_dtypes.html)
28. Calculez les moyennes des variables.
29. Calculez les écarts-type « population » c.-à-d. avec 1/n et non pas 1/(n-1) (voir https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.std.html, en particulier le paramètre ddof)

```
age           9.092182
pression     17.828501
cholester    51.590433
taux_max     23.122778
depression   11.430871
pic           0.613251
```

30. Avec l'outil StandardScaler de « scikit-learn », centrez et réduisez les variables quantitatives (https://www.youtube.com/watch?v=rawaCES1Qf8 ; **15:35**). Attention, l'outil renvoie nativement un objet de type Numpy, transformez-le en data frame avec pour index celui de la base originelle, et pour identifiant de colonnes les noms des variables quantitatives.
31. Recalculez de nouveau les moyennes et écarts-type pour vérifier que la standardisation a été opérée correctement (oui, ouf !)

2.2.2 Variables qualitatives

32. Isolez les variables qualitatives dans une structure spécifique.
33. On souhaite effectuer un codage disjonctif complet (codage 0/1) où toutes les modalités doivent figurer parmi les indicatrices (cf. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html#sklearn.preprocessing.OneHotEncoder). Nous demandons une structure non-sparse pour que la matrice soit facilement lisible.
34. Nous effectuons la transformation « fit_transform ». Quel est le type de la structure obtenue ? (matrice numpy).
35. Affichez la liste des colonnes produites par le codage disjonctif (get_features_names_out).
36. Transformez la matrice numpy en data frame Pandas en précisant les index et noms de colonnes adéquats. Vérifiez l'intégrité de la nouvelle structure produite (info).

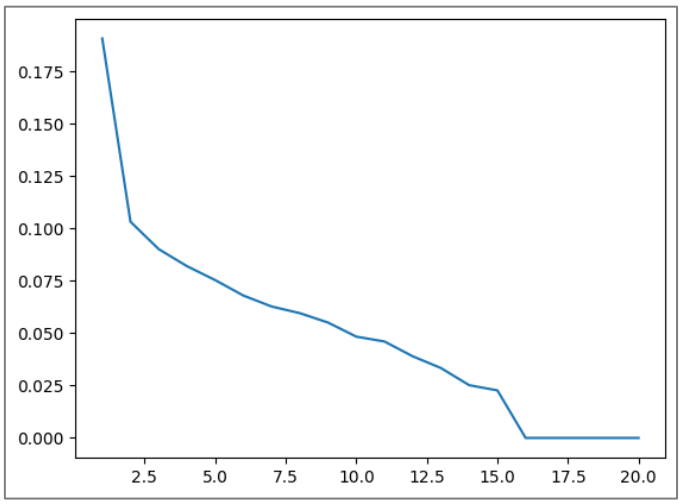
37. A partir de ce nouveau data frame, répondez aux questions suivantes : Quelle est la proportion des femmes dans la base ? (32.22%) Des personnes qui présentent une angine (de poitrine) (32.96%) Des douleurs de type « D » (47.77%)
38. Pour ces variables binaires, effectuez la transformation décrite dans notre support de cours (**Cours 1, page 8, la partie bleue**)
39. A titre de vérification, calculez les moyennes après transformation (sexe_feminin = 0.567, sexe_masculin = 0.823, type_douleur_1 = 0.272, etc.)

2.2.3 Concaténation des bases transformées

40. Réunissez les deux bases : les variables quantitatives standardisées d'une part, les indicatrices corrigées d'autre part (https://pandas.pydata.org/docs/reference/api/pandas.concat.html).
41. Quel est le nombre de lignes et de colonnes dans notre nouveau jeu de données (270, 20)

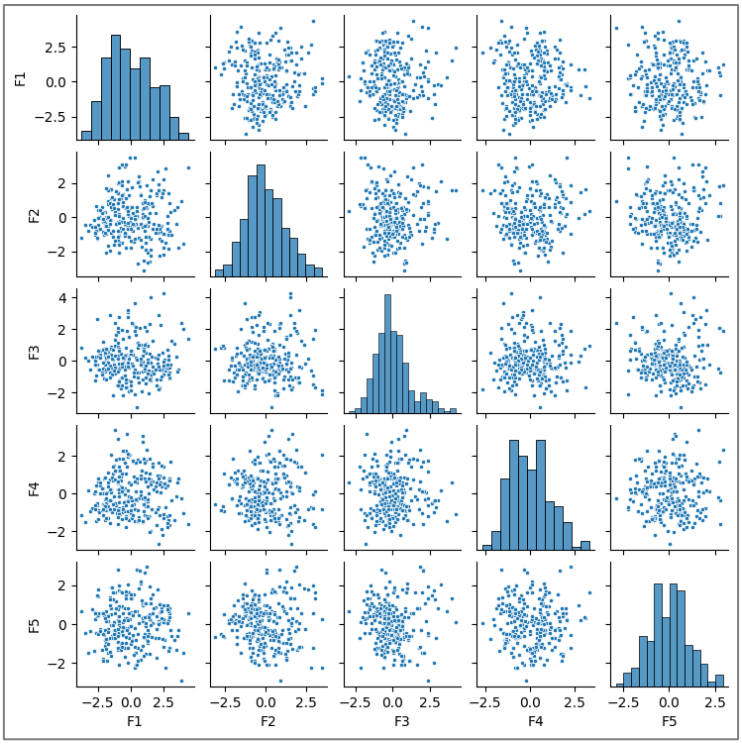
2.3 AFDM = ACP sur données transformées

42. Il ne sous reste plus qu'à appliquer ACP sur cette nouvelle version de notre jeu de données (**Cours 1, page 11** ; sauf que nous travaillons sous Python). Instanciez une ACP. Nous ne spécifions pas le nombre de facteurs, nous utilisons l'algorithme exact (https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html). Appliquez-le sur les données transformées ci-dessus.
43. Affichez la courbe des variances expliquées par les facteurs. Que constatez-vous ? (a priori 2 facteurs si on se réfère au coude, mais quand la courbe descend en pente douce comme ça, décider est difficile)



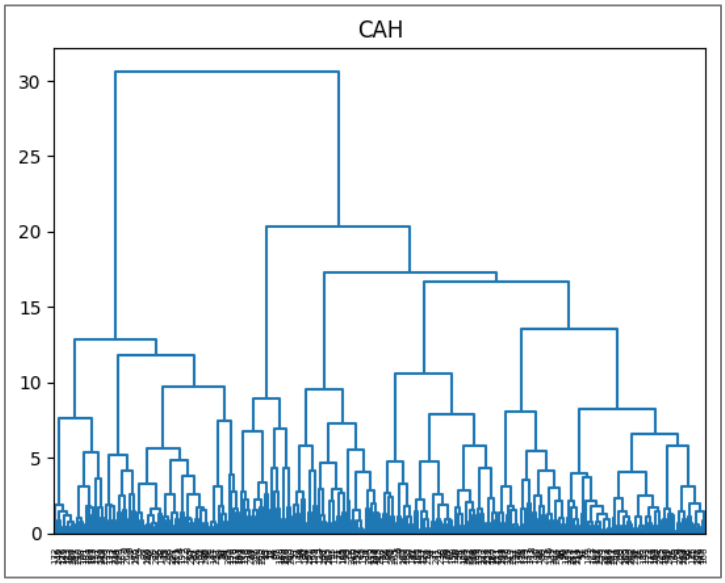
44. Pour aller plus loin, il faudrait combien de facteurs pour capter au moins la moitié de l'information disponible ? (au moins 5, avec une variance de 54.06%)
45. **Va pour 5 facteurs**. Affichez les coordonnées factorielles des 10 premiers individus.

46. Construisez un graphique nuage de points des 5 facteurs pris deux à deux pour identifier les « patterns » possibles (https://seaborn.pydata.org/generated/seaborn.pairplot.html).



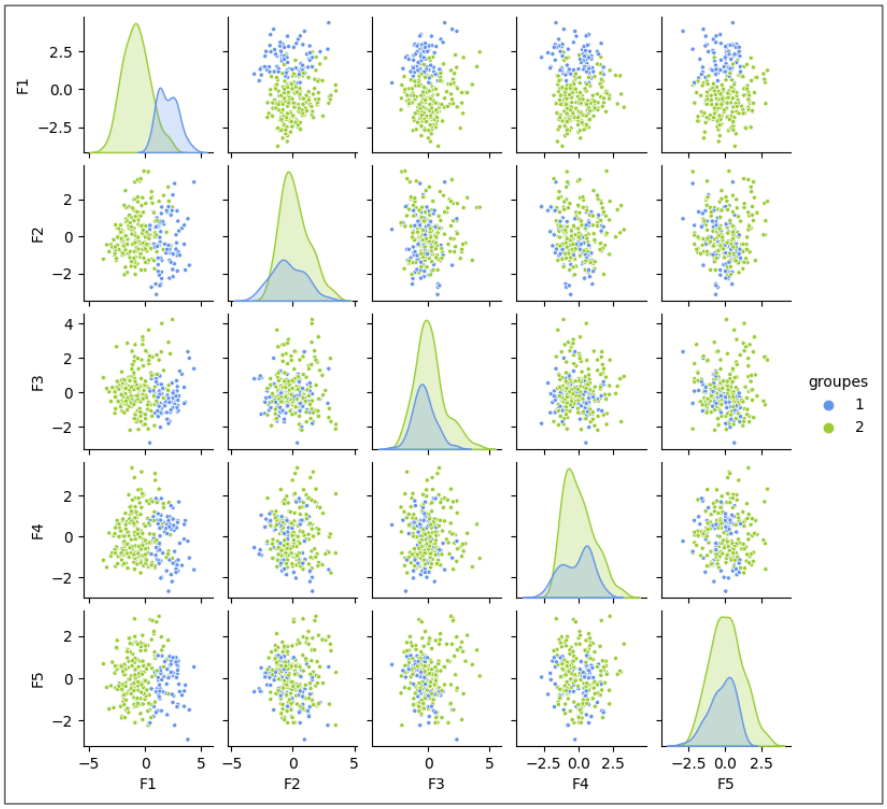
2.4 CAH sur les axes factoriels

47. Tout ça n'est pas vraiment probant à vrai dire, nous distinguons difficilement des groupes à ce stade. Nous comptons sur l'algorithme de classification automatique pour les identifier. Sur les 5 premiers facteurs, lancez une CAH avec une métrique euclidienne et la méthode de Ward. Affichez le dendrogramme.



48. Nous optons pour le découpage en 2 groupes. Quels en sont les effectifs ? (**1** : 77 ; **2** : 193)

49. Refaites le graphique des nuages de points en illustrant les points par leurs classes d'appartenance. Quelle est la composante qui contribue le mieux à la séparabilité des classes ? (plutôt la première, non ?)



2.5 Interprétation des groupes

Les groupes ont été établis à partir de la représentation factorielle, mais ils doivent être interprétés à partir des variables originelles pour que nous puissions comprendre leur nature.

50. Calculez les moyennes marginales des variables quantitatives. Quel est l'âge moyen sur l'ensemble de la base ? (54.43) Et pour les autres variables ?
51. Calculez les mêmes moyennes mais conditionnellement au groupe d'appartenance. Que peut-on dire alors du 1er groupe ? Pour l'âge ? Pour la variable dépression ? Pour le taux_max ? Qu'est ce qui caractérise le premier groupe par rapport au second ? [Remarque (vous pouvez travailler dessus chez vous) : les valeurs intra-variables sont comparables, mais est-ce qu'elles le sont d'une variable à l'autre ? Que faudrait-il calculer pour qu'elles le soient ?]
52. Quelles sont les fréquences relatives marginales des modalités ? Ex. la proportion de femmes (32.2%), des hommes (67.8%), des (angine = oui) (32.96%), etc.
53. Calculez les mêmes proportions au sein de chaque cluster. Ex. dans le « cluster 1 », quelle est la proportion des (angine = oui) ? (72.72%), et dans le cluster 2 ? (17.09%) Et si on compare ces

valeurs à la proportion marginale, que peut-on dire alors à ce sujet ? Regardez ce qu'il se passe pour les autres modalités. Même question ici : qu'est-ce qui caractérise le premier groupe par rapport au second ?

2.6 Variable illustrative

54. Dans la feuille « var illustrative » de notre fichier, récupérez la colonne « cœur ». Elle indique l'occurrence (présence) ou non (absence) d'une maladie cardiaque chez les personnes étudiées. Quelles sont les proportions des personnes non malades et malades ? (resp. 44.4% et 55.6%)
55. Croisez cette variable avec les groupes obtenus par classification automatique. Que peut-on dire à propos de ce résultat ?

col_0	1	2
coeur		
absence	9	141
presence	68	52

Est-ce qu'il y aurait moyen d'élaborer un pipeline que l'on peut sauvegarder et déployer ici ???