Nous travaillons sous Python durant cette séance. Utilisez à votre convenance, soit Jupyter Notebook, soit VS Code (Visual Studio Code), soit Google Colab ; mais il faut bien créer un projet NOTEBOOK quoiqu'il en soit pour pouvoir alterner titres, codes, résultats, commentaires.

Supports de référence :

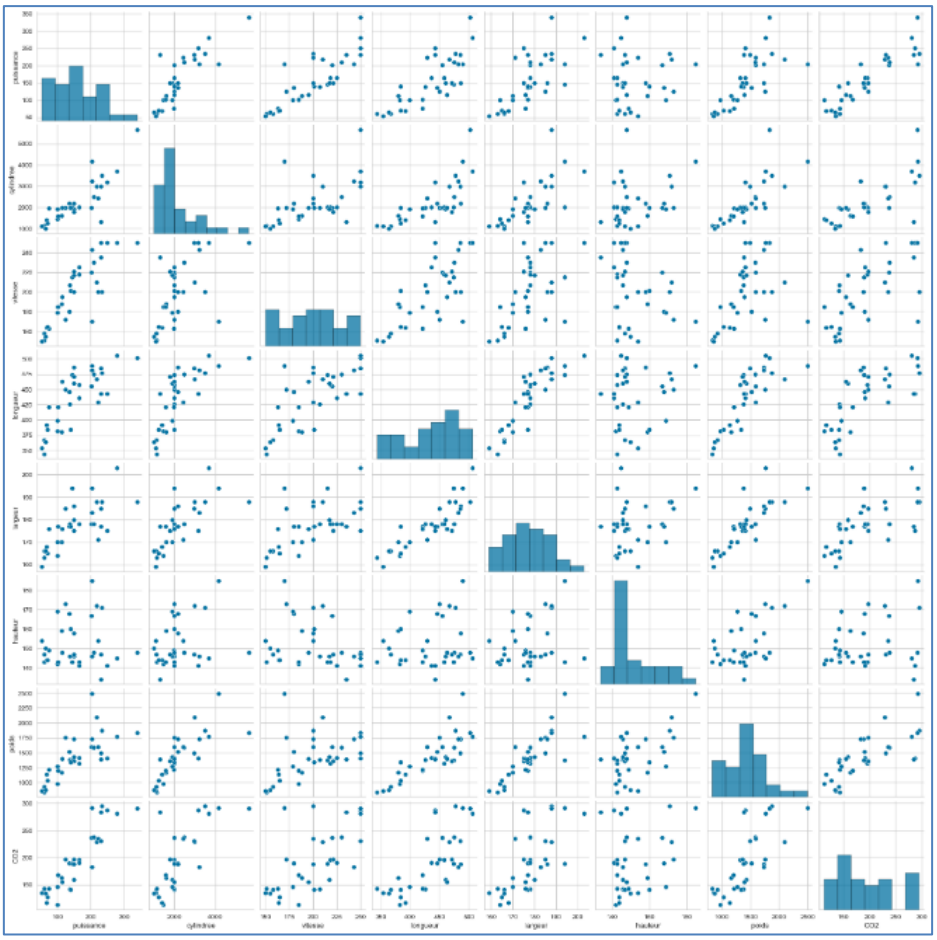
- **Vidéo 1** : https://www.youtube.com/watch?v=OKVmfEqa1jQ (VS Code, si vous l'utilisez)
- **Vidéo 2 :** https://www.youtube.com/watch?v=irjzix6HwOw [La méthode K-Means avec Python (scikit-learn)]
- **Vidéo 3 :** https://www.youtube.com/watch?v=5jggoPitXjw (La structure DataFrame de la librairie « pandas »)
- **Vidéo 4 :** https://www.youtube.com/watch?v=rawaCES1Qf8 (Standardisation des variables en analyse prédictive)
- **Tuto 1** : https://cours-machine-learning.blogspot.com/p/clustering.html, voir le tutoriel « **CAH et K-Means avec Python** ».
- **Support 1 :** https://cours-machine-learning.blogspot.com/p/clustering.html, voir les slides « **Méthode des centres mobiles – K-Means** ».
- **Support 2 :** https://cours-machine-learning.blogspot.com/p/clustering.html, voir les slides « **Caractérisation des classes** ».

1. Importation et inspection des données

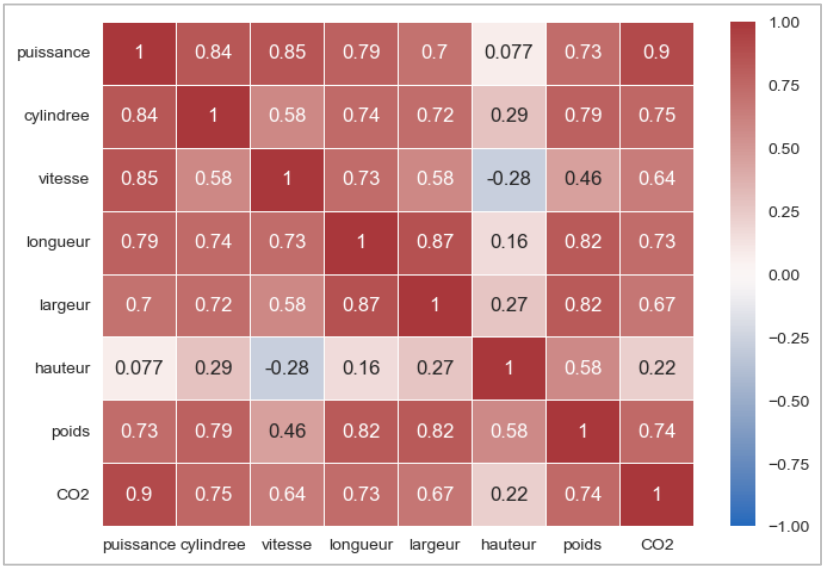
Nous travaillons avec le fichier « **autos2005.xlsx** ». La première feuille « actifs » décrit les propriétés les propriétés de n = 34 véhicules. Les p = 8 variables actives sont « puissance », ..., « CO2 » ; les 4 illustratives sont « prix », ..., « type4X4 ».

1. Ouvrez le fichier dans le tableur Excel (ou LibreOffice). Sélectionnez la première feuille « actifs » pour bien appréhender la structure des données. La colonne « Modèle » est un index qui permet de repérer les individus. Fermez le fichier après consultation parce qu'Excel a la fâcheuse habitude de verrouiller les classeurs ouverts et risque de faire échouer l'importation dans votre notebook.
2. Démarrez un nouveau notebook. Premières étapes :
    - Vérifiez la version de Python (https://www.geeksforgeeks.org/check-the-version-of-the-python-interpreter/ ; la « Method 1 » fait très bien l'affaire) (3.11.5 en ce qui me concerne ; ce n'est pas rédhibitoire pour autant si vous avez une version un peu différente, mais cela peut expliquer que, parfois, nous ayons une légère différence de résultats, ceci est également valable pour les packages...).
    - Vérifiez également les versions des packages « scikit-learn », « yellowbrick » et « seaborn » (Vidéo 1, 8:55) (resp. 1.3.0 ; 1.5 et 0.12.2). Peut-être que « yellowbrick » n'est pas présent dans votre environnement, il faut alors l'installer à la volée (pip install ...) via votre notebook même.
3. Importez la première feuille « actifs » du classeur « **autos2005.xlsx** ». Attention, la première colonne « Modèle » n'est pas une variable, elle correspond à un index (pandas.read_excel, option index_col) (Vidéo 3, 04:52). Vérifiez d'emblée le type de l'objet (type).

4. Affichez les dimensions de la matrice de données (shape) (34, 12)
5. Affichez les informations sur le dataset (info), vérifiez que la colonne « Modèle » a bien été reconnue comme « index ».
6. Affichez les premières lignes de données (head).
7. Affichez la liste des colonnes (columns).
8. Construisez un sous-ensemble de données ne regroupant que les variables actives, allant de « puissance » à « CO2 » (8 variables) (Vidéo 3, 11:50).
9. Calculez les statistiques descriptives (describe). Que constatez-vous ? A l'évidence, les variables ne sont pas exprimées dans les mêmes unités (puissance en chevaux, cylindrée en cm^3, vitesse en km/h, etc.), les plages de valeurs d'une variable à l'autre sont très différentes.
10. Affichez les nuages de points par paires de variables (cf. https://pandas.pydata.org/docs/reference/api/pandas.plotting.scatter_matrix.html ; ou encore, https://seaborn.pydata.org/generated/seaborn.pairplot.html, pour ce dernier il faut importer la librairie « seaborn » [qui a été installée avec l'environnement]). Qu'observe-t-on ? (pas de groupes « évidents », plusieurs variables assez corrélées, pas de points manifestement atypiques ...)



11. On s'intéresse à la corrélation entre les variables. Calculez la matrice des corrélations (https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.corr.html). Qu'en est-il par exemple des corrélations entre puissance, cylindrée, vitesse ?
12. Nous souhaitons affichez les corrélations dans un graphique « heatmap » (https://seaborn.pydata.org/generated/seaborn.heatmap.html). Pour obtenir quelque chose qui ressemble au graphique ci-dessous, cf. les options vmin, vmax, center, annot, linewidth, et cmap (https://seaborn.pydata.org/tutorial/color_palettes.html). Qu'observe-t-on dans le graphique ? (les variables sont positivement corrélées entre elles pour la plupart) Quelle est la variable qui est la moins corrélée avec les autres ? (hauteur)



2. K-Means avec la librairie « scikit-learn »

On souhaite effectuer une classification automatique avec la méthode des K-Means. Pour l'heure, nous fixons à **K = 3** le nombre de classes, sachant qu'il faudra se pencher plus attentivement sur cette question un peu plus loin.

13. Dans le doute, les variables étant exprimées dans des unités différentes, nous décidons de les standardiser (Vidéo 4, 15:34).
14. Stockez le dataset transformé (qui est une structure numpy) dans un data frame pandas (https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html, attention aux options index et columns).
15. Calculez de nouveau les statistiques descriptives (describe). Que constatez-vous au niveau des moyennes, et des écarts-type ?
16. Ce dernier résultat (les écarts-type obtenus) est étonnant ? Pour vous assurer de la nature des calculs effectués, utilisez la fonction std() de la librairie numpy pour les obtenir

(https://numpy.org/doc/stable/reference/generated/numpy.std.html), lisez attentivement la documentation de l'option « ddof ». Faites calculer l'écart-type « population » par variable. Qu'observe-t-on maintenant ? Que faut-il comprendre ici ?

17. Nous souhaitons instancier un K-Means à (n_clusters = 3) groupes (Vidéo 2, 12:47). Nous procédons à (n_init = 10) itérations pour espérer aboutir à une bonne optimisation. Nous fixons (random_state = 0) pour que nous réalisons tous les mêmes calculs et obtenir des résultats identiques (lisez la doc, en particulier les options qui nous concernent : https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html)
18. Affichez l'inertie intra-classes (WSS) de la partition engendrée par K-Means (inertia_) (100.3...).
19. Affichez les groupes d'appartenance des individus (labels_) (Vidéo 2, 14:00).
20. Mettez en place un affichage plus avenant permettant d'associer les noms des véhicules à leur groupe d'appartenance (en passant par un data frame pandas peut-être ?)
21. Comptabilisez le nombre de véhicules par cluster (numpy.unique) (17, 12, 5). Récupérez ces effectifs dans un vecteur, nous en aurons l'usage plus loin.
22. Affichez les prototypes des classes (moyennes des variables conditionnellement aux classes, sachant qu'elles sont exprimées dans l'espace des variables centrées et réduites) (cluster_centers_). Arrangez-vous pour que la présentation soit avenante (Remarque : les numéros de groupe sont attribués arbitrairement, il se peut que vos résultats soient organisés différemment, mais les moyennes conditionnelles doivent être peu ou prou les mêmes).

	puissance	cylindree	vitesse	longueur	largeur	hauteur	poids	CO2
0	1.427925	1.834701	0.518390	1.165014	1.520195	1.028092	1.650028	1.516907
1	0.296261	0.002895	0.529867	0.489178	0.219435	-0.189080	0.211677	0.231718
2	-1.014671	-0.768560	-0.966641	-1.178425	-0.944280	-0.160508	-0.987388	-0.960311

23. Si l'on se réfère la puissance des véhicules, qu'est-ce que caractérise le premier groupe ? Le second ? Le troisième ?
24. En utilisant la formule vue en cours (Support 1, page 6). Calculez l'inertie inter-classes de la partition BSS (en décryptant un peu la formule : parce qu'on utilise la distance euclidienne, on se rend compte que $d^2(G_k,G)$ est égale à la somme des carrés des moyennes conditionnelles des variables puisqu'elles ont été centrées) (BSS = 171.6...)
25. Effectuez la somme BSS + WSS qui correspond à l'inertie totale (TSS = 272).
26. Sachant que par ailleurs, TSS = n (nombre total d'individus) * somme des variances, effectuez le calcul direct. Est-ce que la valeur obtenue coïncide avec la précédente ? (oui,

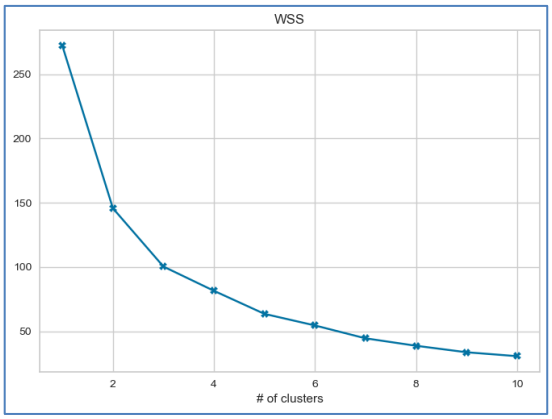
sinon gros problème ; attention : n'oublions pas que les variables ont été réduites, ce qui n'est pas sans impact sur leurs variances).

27. En déduire alors le R^2, l'inertie expliquée par la partition (BSS/TSS) (63.1 %).

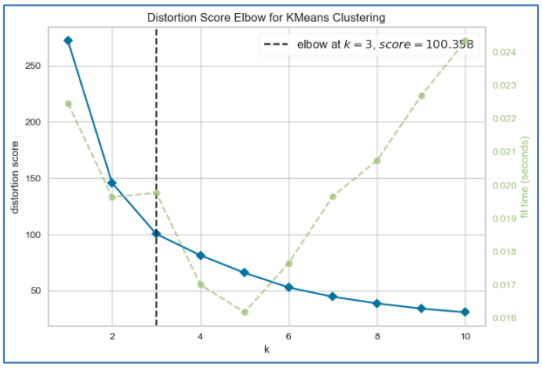
3. Détermination du nombre de classes

La détermination du nombre de clusters est un étape clé d'un processus de classification automatique. Avec un K-Means, une solution simple consiste à tester simplement différents scénarios (K = nombre de groupes) et à les comparer à travers les indicateurs de qualité des partitions.

28. On souhaite tracer la courbe d'évolution de l'inertie intra-classes WSS en fonction du nombre de classes (K). Essayez d'obtenir quelque chose qui ressemble à ci-dessous en faisant varier K = 1 à 10 (Tuto 1, page 10 ; attention : K = 1, WSS = TSS ; par rapport au tutoriel, je précise bien que l'on veut dessiner la courbe de WSS [inertia_] en ce qui nous concerne et non pas celui du critère silhouette !!!) (les options marker et linestyle de matplotlib / plot devraient vous aider pour obtenir une présentation similaire à ci-dessous)



29. Essayons de reproduire la même chose, mais en utilisant la librairie « yellowbrick » cette fois-ci avec l'outil KElbowVisualizer (Vidéo 2, 22:10).



30. Et si on utilise le critère silhouette, quel aspect aura la courbe de « yellowbrick » ?

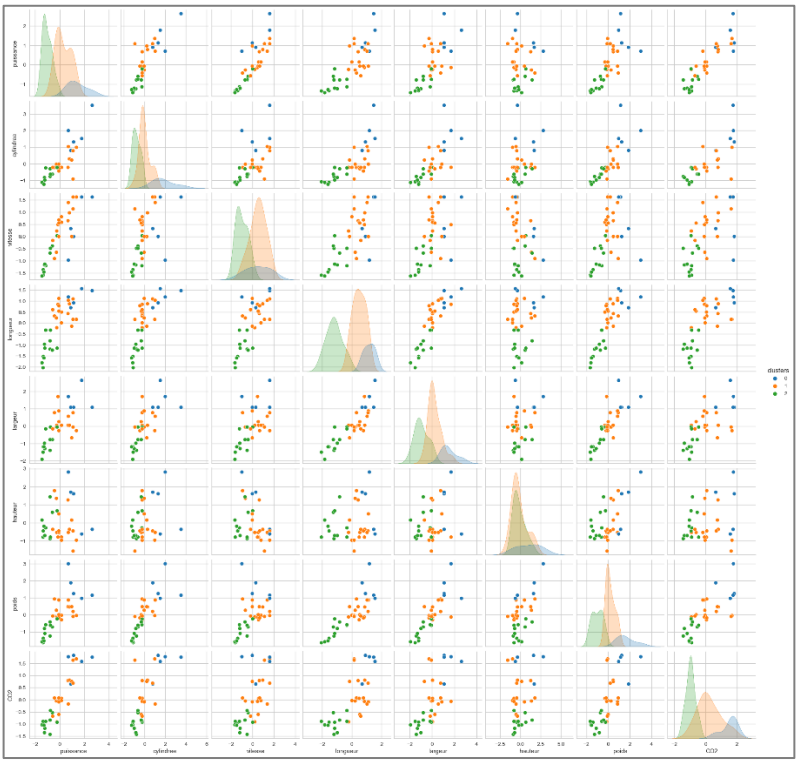
31. En conclusion, quelle est la valeur de K la plus crédible ? (K = 3 est pas si mal, mais K = 2 n'est pas à négliger non plus, tout dépend de l'interprétation finalement)

4. Interprétation des groupes – Variables actives

32. **On conserve l'idée de K = 3**. On souhaite mettre en évidence les variables actives qui ont le plus fort impact dans la détermination des classes. Sachant qu'elles ont été centrées et réduites, calculez pour chacune d'entre elles : le SCT, le SCE et le SCR. En déduire alors leur rapport de corrélation (voir Support 2, page 8 ; Tuto 1, page 12) (puissance : 0.707, cylindrée : 0.703, ..., CO2 : 0.690). Triez les résultats de manière à ce que les variables les plus impactantes soient dans les premières positions. Quelles sont les variables qui influent le plus dans la constitution des groupes ? (longueur, poids, ...) Le moins ? (hauteur, vitesse)

longueur	0.809369
poids	0.766880
puissance	0.707107
cylindree	0.703500
CO2	0.690711
largeur	0.678633
vitesse	0.509685
hauteur	0.182405

33. Construisez de nouveau les nuages de points par paires de variables, mais en les illustrant selon leur cluster d'appartenance. Confortent-t-ils les résultats numériques de la question précédente ? (oui, si on regarde le couple longueur – poids par ex.) (Vidéo 2, 15:35)



34. Calculer les prototypes (les moyennes conditionnelles) mais dans l'espace originel cette fois-ci (avec les variables exprimées dans leurs unités d'origine). Quelle est la puissance moyenne des véhicules du cluster n°0, du n°1, du n°2 (Vidéo 2, 19:40). Que peut-on dire de la nature des groupes ? (voitures moyennes, petites, imposantes)

	puissance	cylindree	vitesse	longueur	largeur	hauteur	poids	CO2
0	255.200000	4001.200000	216.000000	488.200000	192.200000	164.200000	2013.000000	277.600000
1	175.882353	2214.764706	216.352941	457.764706	179.647059	150.000000	1500.176471	205.823529
2	84.000000	1462.416667	170.333333	382.666667	168.416667	150.333333	1072.666667	139.250000

5. Variables supplémentaires

Nous nous intéressons aux variables supplémentaires maintenant pour renforcer notre interprétation des groupes.

35. Isolez dans une structure spécifique (data frame) les variables « prix », « origine », « carburant » et « type4X4 ». Ajouter au data frame la colonne indicatrice des groupes d'appartenance issu des K-Means.
36. Calculez le prix moyen des véhicules pour chaque classe (plusieurs pistes possibles, avec pivot_table peut-être ? pandas.pivot_table — pandas 1.5.2 documentation (pydata.org)) (0 : 59268, 1 : 31430, 2 : 15306 ; encore une fois, le numéro de groupe est susceptible d'être différent chez vous). Conclusion ?
37. Calculez le V de Cramer de chaque variable illustrative qualitative (origine, carburant, type4X4) avec l'indicatrice des classes. Quelle est la variable la plus liée avec l'appartenance aux classes ? (cf. Cramer's V correlation matrix | Kaggle ; attention : la fonction décrite fournit le carré du v de Cramer, il faut corriger le résultat ; ou bien : https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.contingency.association.html) (variable la plus fortement liée « type4X4 » avec v = 0.373).
38. Quelle est la proportion des « type4X4 = oui » dans chaque cluster ? (0 : 0.4, 1 : 0.17, 2 : 0) (autre lecture P(type4X4 = oui / cluster = 1) = 0.176)
39. Quelle est la répartition des « type4X4 = oui » selon les clusters (60% des véhicules 4X4 sont dans le cluster 1, etc.) (autre lecture P(cluster = 1 / type4X4 = oui) = 60%)

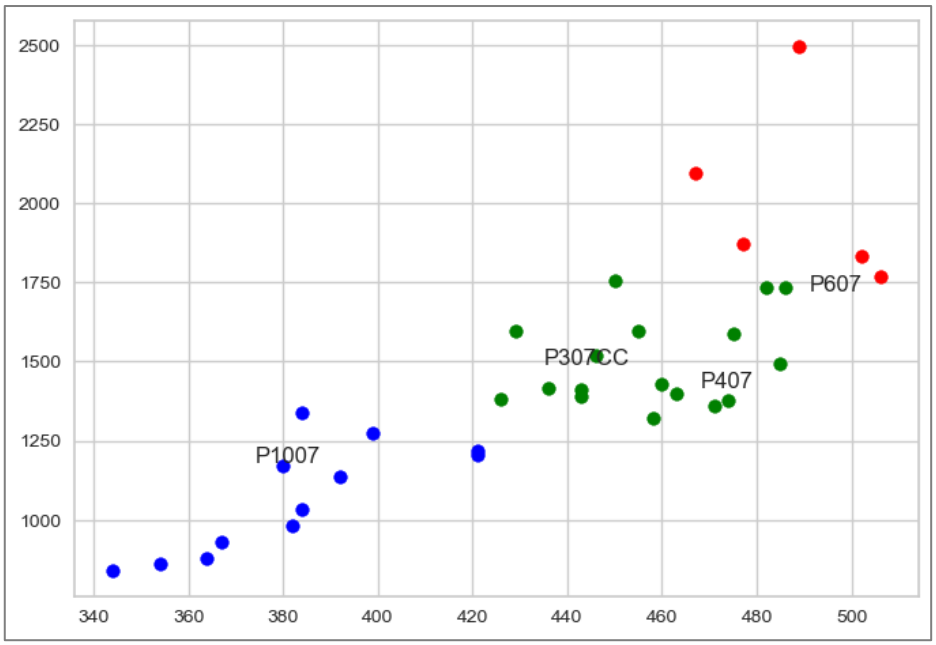
6. Traitement des individus illustratifs

Nous nous intéressons aux véhicules supplémentaires (feuille « illustratifs » du classeur « **autos2005.xlsx** »), qui sont tous de la marque Peugeot. Nous aimerions les situer par rapport aux groupes qui ont été mis en évidence par l'algorithme de clustering.

40. Chargez la feuille Excel dans un nouveau data frame (pandas.read_excel, encore une fois, « Modèle » est un index ici).
41. Récupérez dans une structure spécifique les variables actives.
42. Centrez et réduisez les variables en utilisant les moyennes et écarts-type calculés sur les individus actifs (Vidéo 4, 18:48). Affichez les résultats.

Modele	puissance	cylindree	vitesse	longueur	largeur	hauteur	poids	CO2
P407	-0.272761	-0.220400	0.388314	0.716460	0.463251	-0.617662	-0.027223	0.020012
P307CC	0.355009	-0.220400	0.811060	-0.016328	-0.158481	-0.789094	0.183135	0.306499
P1007	-1.143079	-0.873578	-1.140075	-1.370874	-0.883834	0.753799	-0.683538	-0.714111
P607	0.697429	0.521988	0.973655	1.227190	0.670495	-0.617662	0.836646	0.539270

43. Attribuez à chaque individu illustratif sa classe d'appartenance (predict du K-Means) (P407 : cluster 1, P307CC : 1, P1007 : 2, P607 : 1 ; tout dépend encore une fois des numéros de groupe attribués aux clusters). Est-ce que les résultats vous paraissent cohérents ? (oui, plus ou moins, doute quand-même pour la Peugeot 607 qui est plutôt une berline relativement imposante)
44. Pour essayer de mieux situer ces affectations : (1) placez les voitures actives dans le plan (longueur, poids) (c'étaient les variables à plus fort impact, rappelons-le) en coloriant les points selon leur cluster d'affectation ; (2) placez ensuite dans le même repère les véhicules illustratifs en indiquant le nom du modèle. Finalement la Peugeot 607 ? (limite des voitures imposantes en réalité, en tout les cas dans ce plan-là) (inspirez-vous – mais il n'est pas question de calcul factoriel pour nous à ce stade – de Tuto 1, page 16)



45. **A FAIRE, MAIS JE NE METTRAI PAS LA CORRECTION**. Pour disposer d'une vue plus fidèle de la réalité [les groupes ont été construits sur la base d'un calcul en p = 8 dimensions et non pas uniquement à partir du couple (longueur, poids)], on se propose de projeter les individus dans un espace factoriel pour mieux situer les positions relatives des groupes (avec un ACP par ex. ?), on pourra alors y placer avec une meilleure appréciation nos véhicules illustratifs « Peugeot ». Que constatez-vous ?

Remarque : Pour réaliser une ACP (analyse en composantes principales) sous Python, voir https://www.youtube.com/watch?v=LeAq3iS-bpY