Applications de l’optimisation dans l’apprentissage automatique (machine learning) en économie cas des moindres carrés ordinaires (MCO) et des K-Means.

Axel-Cleris Gailloty

10/11/2020

Applications de l’optimisation dans le champ de l’économie : cas des moindres carrés ordinaires et des K-Moyennes.

# Introduction

L’économie est souvent définie comme la science de la rareté et des choix efficaces. Cette définition part du fait que les besoins des agents économiques sont illimités tandis que les ressources, elles sont limitées et rares. Il revient donc à ce que chaque agent économique définisse sa fonction objectif (fonction d’utilité pour les individus, fonction de production pour les entreprises) à optimiser afin de maximiser son gain compte tenu des moyens qu’il possède.  
Deux branches de la science économique emploient intensivement à l’optimisation : ce sont la microéconomie et l’économétrie.

La microéconomie emploie l’optimisation en vue de construire des théories qui expliquent le comportement humain en face de la rareté, de l’incertitude et dans le cadre des contrats. L’optimisation en microéconomie permet par exemple de trouver le panier optimal de biens à acheter compte tenu du revenu du consommateur. Cette approche permet par exemple de prédire le comportement du consommateur face aux variations de prix. Elle peut aussi aider à déterminer la combinaison optimale des facteurs de production (typiquement le capital et le travail) afin de maximiser le profit ou de réduire les coûts de production.

L’économétrie de son côté utilise tout un ensemble de techniques mathématiques, statistiques et informatiques en vue de tester les théories économiques à travers la modélisation mathématique, l’estimation des paramètres et la prévision.

L’économétrie en tant que branche des sciences économiques a importé dans la science économique un très grand nombre d’outils empruntés à d’autres disciplines comme la biométrie, les statistiques. La méthode des moindres carrés ordinaires qui est aujourd’hui largement utilisé avec des variations mineures dans plusieurs modèles économétriques illustre bien le recours de l’économétrie à d’autres disciplines.

Dans ce travail nous aimerions aborder l’aspect optimisation dans deux algorithmes souvent employés dans des problèmes économiques. Ce sont les moindres carrés ordinaires et les K-Moyennes.  
Nous commencerons par présente chaque algorithme puis nous cherchons deux applications pour chacun.

# Les moindres carrés ordinaires

## Historique de la méthode

La première publication de la méthode des moindres carrés (destinée à déterminer des quantités dans un système d’équations surdéterminé) est due à Legendre qui l’a donnée en annexe d’un ouvrage intitulé *Nouvelles méthodes pour la détermination des orbites des comètes (1805)*

## Les moindres carrés ordinaires simples

### Formalisation

La méthode des moindres carrés utilise de l’optimisation mathématique. Il s’agit en effet de trouver un vecteur de paramètres qui minimise la **fonction objective** qui est la somme des carrés des résidus.  
Certaines formalisations utilisent une autre fonction objective mais dans la plupart des cas nous cherchons à réduire la somme des carrés des résidus (SCR). L’intérêt de minimiser la SCR est qu’il s’agit d’une fonction convexe qui admet donc un minimum global.  
L’optimisation se fait en utilisant la méthode de Lagrange.

Dans les lignes suivantes nous exposons les étapes utilisées pour arriver au résultat optimal.

Nous savons que la forme finale de l’équation que nous voulons estimer est la suivante :

Ainsi en réécrivant l’équation initiale notre programme d’optimisation est le suivant :

Nous posons les conditions de premier ordre (CPO) du Lagrangien.

Etant donné que est une constante, la somme allant de 1 à N est égale à

Nous pouvons extraire

Maintenant nous cherchons à exprimer

Nous remplaçons par son expression trouvée plus haut.

Or empiriquement nous savons que

et

sont égales à , nous allons donc multiplier le numérateur et le dénominateur par

Nous trouvons ainsi l’expression finale de .

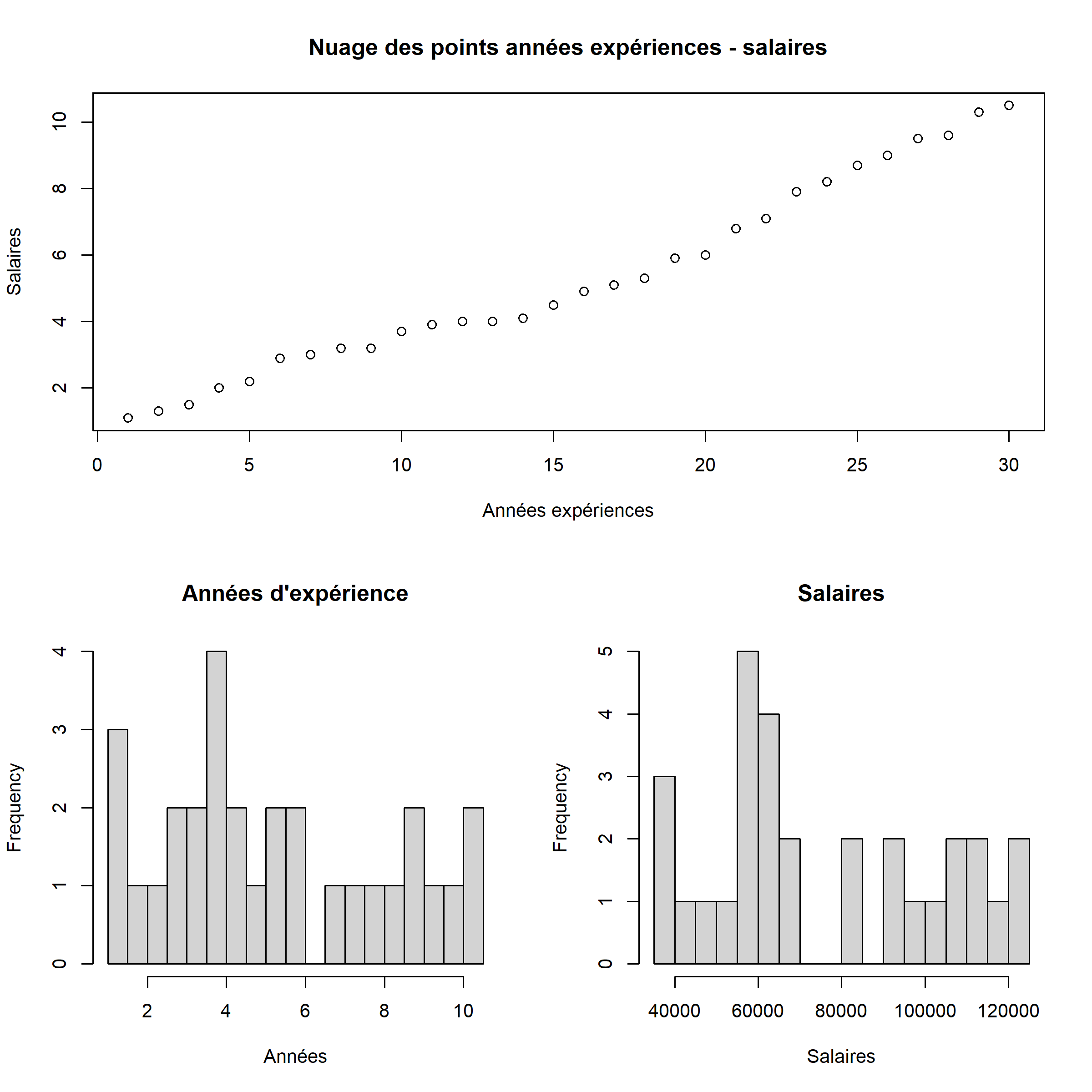
## Applications

### La relation entre le salaire et l’expérience professionnelle

Ayons un aperçu des 10 premières lignes des données sur lesquelles nous allons démontrer une application de la méthode d’estimation par les OLS simples.

|  |  |
| --- | --- |
| annees | salaire |
| 1.1 | 39343 |
| 1.3 | 46205 |
| 1.5 | 37731 |
| 2.0 | 43525 |
| 2.2 | 39891 |
| 2.9 | 56642 |
| 3.0 | 60150 |
| 3.2 | 54445 |

Nous pouvons faire des graphiques pour afficher la distribution des colonnes (par des histogrammes) et la tendance avec un graphique



Notre but est donc d’exprimer le salaire comme une fonction des années d’expérience. Nous cherchons une relation de la forme telle que les coefficients estimés et minimisent la somme du carrées des erreurs.

Nous appliquons les relations que nous avons trouvées dans lorsque nous avons résolu le programme d’optimisation du Lagrangien.

Ainsi nous calculons d’abord puis .

La formule nous indique que

|  |  |  |
| --- | --- | --- |
|  | Sommes | Moyennes |
| Années Expériences (X) | 159.4 | 5.313333 |
| Salaires (y) | 2280090.0 | 76003.000000 |

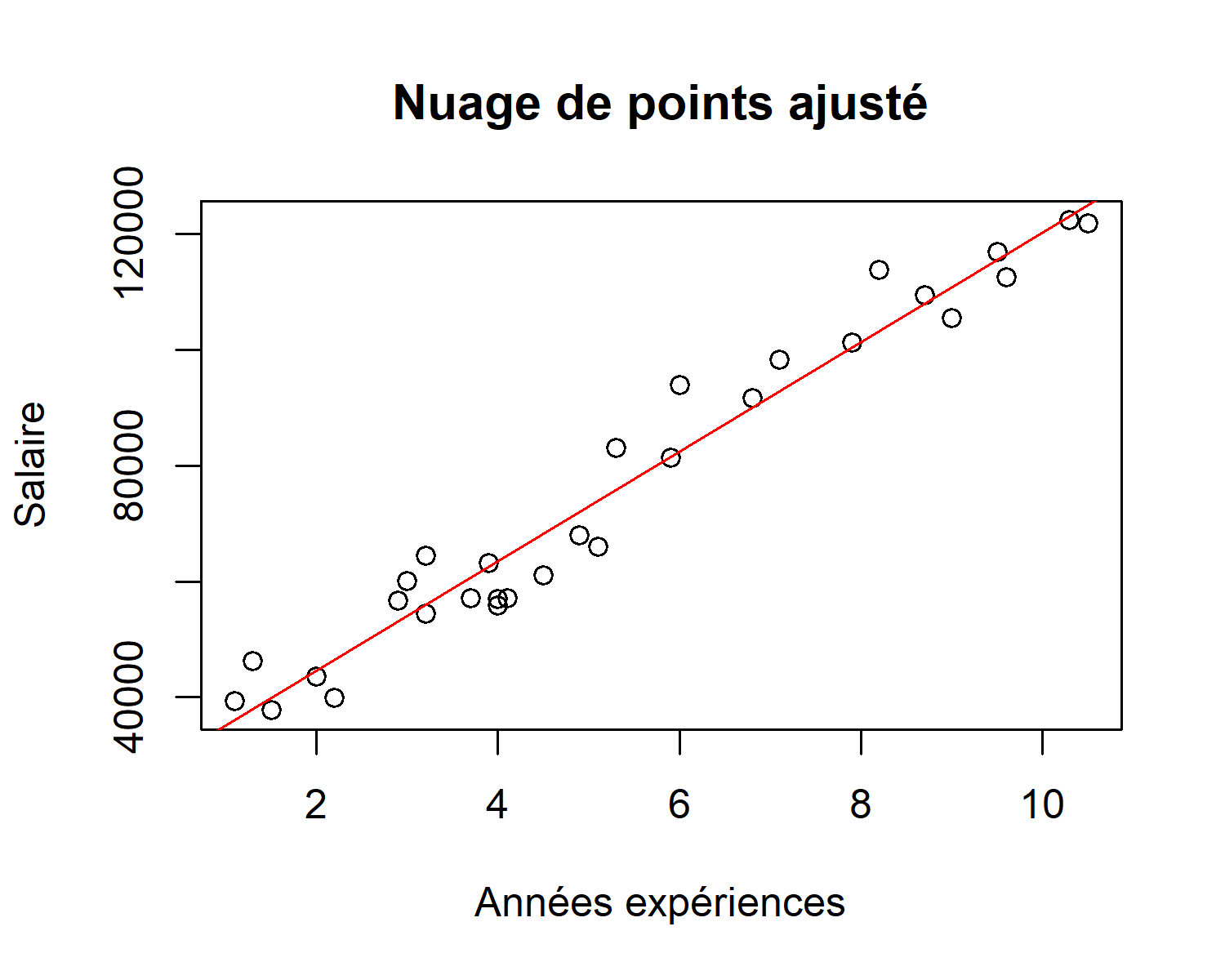
Nous allons donc calculer les statistiques intermédiaires en ajoutant des colonnes à notre tableau initial.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 1.1 | 39343 | -4.213333 | -36660 | 154460.80 | 17.752178 |
| 1.3 | 46205 | -4.013333 | -29798 | 119589.31 | 16.106844 |
| 1.5 | 37731 | -3.813333 | -38272 | 145943.89 | 14.541511 |
| 2.0 | 43525 | -3.313333 | -32478 | 107610.44 | 10.978178 |
| 2.2 | 39891 | -3.113333 | -36112 | 112428.69 | 9.692844 |
| 2.9 | 56642 | -2.413333 | -19361 | 46724.55 | 5.824178 |

Nous calculons donc à partir des valeurs générées dans le tableau.  
Le coefficient de estimé est égal à 9449.96.

Nous pouvons aussi déterminer la valeur de en utilisant sa formule qui est .  
Ainsi est égal à 25792.

Nous pouvons représenter la droite sur les données.



# La méthode des K-Moyennes

La méthode des K-Moyennes communément appelée l’algorithme K-Means est une méthode de partitionnement des données et un problème d’optimisation combinatoire. En Machine Learning nous considérons cette méthode comme une technique d’apprentissage non supervisé. Étant donnés des points et un entier , le problème est de diviser les points en groupes, souvent appelés clusters, de façon à minimiser une certaine fonction. On considère la distance d’un point à la moyenne des points de son cluster ; la fonction à minimiser est la somme des carrés de ces distances. Un cluster est un groupe de données qui partage des traits communs.

Beaucoup de problèmes économiques et sociaux utilisent l’algorithme K-Means pour segmenter, et classer des individus (personnes, collectivités, pays etc…) selon leurs similarités et tirer des conclusions et élaborer des théories.

## Généralités sur l’algorithme

La première étape dans l’implémentation de l’algorithme K-Means est de choisir une mesure de distance entre les points. Il est courant d’utiliser la distance euclidienne entre deux points. Cette distance se calcule comme suit :

et sont des vecteurs de taille

L’idée de base derrière le clustering k-means consiste à définir des clusters de sorte que la **variation intra-cluster totale (appelée variation totale intra-cluster) soit minimisée**. Il existe plusieurs algorithmes de k-moyennes disponibles. L’algorithme standard est l’algorithme de Hartigan-Wong (1979), qui définit la variation totale intra-cluster comme la somme des distances au carré: les distances euclidiennes entre les éléments et le centre de gravité correspondant.

où est un point qui appartient au cluster et est la valeur moyenne des points qui sont classés dans ce cluster.

Chaque observation est donc assignée à un cluster de telle manière que la somme des carrés de la distance de l’observation par rapport au centre du cluster est minimisée. Nous définissons la variation totale intra-cluster comme suit:

La variation totale intra-cluster msure la fiabilité du clustering et c’est elle que le programme d’optimisation cherche à minimiser. Plus elle est petite mieux est la fiabilité.

La résolution de ce programme se fait via l’algorithme suivant :  
L’algorithme K-means peut être résumé comme suit:

* Spécifiez le nombre de clusters (K) à créer (par l’analyste)
* Sélectionnez au hasard k objets de l’ensemble de données comme centres ou moyens de cluster initiaux
* Attribuer à chaque observation le centre de gravité le plus proche, en fonction de la distance euclidienne entre l’objet et le centre de gravité
* Pour chacun des k clusters, mettre à jour le centre de gravité du cluster en calculant les nouvelles valeurs moyennes de tous les points de données du cluster. Le centre de gravité d’un cluster est un vecteur de longueur contenant les moyennes de toutes les variables pour les observations du cluster; est le nombre de variables.
* Minimiser itérativement le total dans la somme des carrés. Autrement dit, répétez les étapes 3 et 4 jusqu’à ce que les attributions de cluster cessent de changer ou que le nombre maximal d’itérations soit atteint.

La résolution de cet algorithme pourrait être fastidieux manuellement donc nous allons recourir à une approche informatique en utilisant le langage de programmation R dans lequel cet algorithme est implémenté de manière efficiente.

## Applications

Nous voulons donc démontrer par un exemple l’utilité de cet algorithme pour des questions économiques. Nous allons nous intéresser à la macroéconomie mondiale. La Banque mondiale, dans son programme **World Development Indicators** a créé plus de 3000 indicateurs pour mesurer le niveau de développement des pays et territoires du monde. Ces données sont gratuitement disponible sur leur site [World Development Indicator | Databank](https://databank.worldbank.org/source/world-development-indicators).

Pour l’application des K-Means, j’ai sélectionné les variables suivantes :

* Le pourcentage de la population ayant accès aux technologies de communication (TELECOM)
* La part de la population active qui travaille dans le secteur agricole (EMPLOIAGRIC)
* La croissance du Produit Intérieur Brut (PIB) (PIBCROIS)
* Le PIB par habitant (PIBCAPITA)
* Le pourcentage du PIB consacré aux dépenses publiques de l’Etat (DEPENSESPUB)
* La superficie du pays en (SUPERFICIE)
* L’espérance de vie (ESPERANCE)
* Le taux de fertilité des femmes (nombre moyen d’enfants par femme en âge de procréer) (FERTILITE)
* La part du revenu national épargné (EPARGNE)

Il y a 142 observations dans la base de données, qui représentent des pays et territoires.

Avant d’appliquer l’algorithme nous avons choisi de normaliser chaque variable en soustrayant sa moyenne puis en la divisant par son écart-type. Le but de cette transformation est de donner à chaque variable un poids équivalent dans calcul des distance entre les points selon la distance euclidienne. Par exemple le PIB/habitant peut aller jusqu’à plus de 50000 dollars tandis que la croissance du PIB est souvent inférieur à 10%. Normaliser les variables revient donc à homogénéiser chaque variables et enlève l’unité de mesure.

Nous commençons par une première résolution de l’algorithme en choisissant , l’algorithme de Hartigan-Wong (que nous avons décrit plus haut) puis le nombre maximal d’itération à 15.

set.seed(123)  
res\_kmeans <- kmeans(matrix\_data\_scaled, centers = 4,   
 algorithm = "Hartigan-Wong",  
 iter.max = 15)

Voici donc les résultats :

str(res\_kmeans)

List of 9  
 $ cluster : int [1:142] 3 2 3 4 2 4 4 2 2 4 ...  
 $ centers : num [1:4, 1:9] -0.3817 -0.2067 -0.0283 0.3825 0.0255 ...  
 ..- attr(\*, "dimnames")=List of 2  
 .. ..$ : chr [1:4] "1" "2" "3" "4"  
 .. ..$ : chr [1:9] "TELECOM" "EMPLOIAGRIC" "PIBCROIS" "PIBCAPITA" ...  
 $ totss : num 1269  
 $ withinss : num [1:4] 92.2 178.5 130.1 302.5  
 $ tot.withinss: num 703  
 $ betweenss : num 566  
 $ size : int [1:4] 10 59 29 44  
 $ iter : int 4  
 $ ifault : int 0  
 - attr(\*, "class")= chr "kmeans"

L’algorithme a convergé au bout de 4 itérations, c’est ce qu’indique l’argument iter. La somme totale des carrés des distances par rapport au centre de chaque cluster est de 1269, ce qu’indique l’argument totss.

La variation individuelle intra-classe est représentée par le résultat withinss. La variation totale intra-cluster est égale à 703, c’est la somme des variations individuelles. Nous ne pouvons pas dire si ce nombre est grand ou petit car il n’existe pas un indicateur pour exprimer la fiabilité en pourcentage. Mais ce premier résultat nous sert de référence. Les praticiens de cet algorithme utilisent souvent une approche itérative qui consiste à choisir un vecteur de paramètres qu’on donne à l’algorithme pour qu’il partitionne les données, puis le nombre qui donne la moindre variable totale intra-classe sera retenue.

Affichons comment l’algorithme a partitionné les pays selon les clusters.

|  |  |
| --- | --- |
| Clusters | Pays |
| 1 | Eswatini, Haiti, Iraq, Lebanon, Lesotho, Liberia, Namibia, Nicaragua, South Africa, Timor-Leste |
| 2 | Albania, Armenia, Azerbaijan, Bahamas, The, Bangladesh, Belarus, Belize, Bhutan, Bolivia, Bosnia and Herzegovina, Botswana, Bulgaria, Cabo Verde, Cambodia, Chile, Colombia, Costa Rica, Croatia, Dominican Republic, Ecuador, Egypt, Arab Rep., El Salvador, Georgia, Greece, Guatemala, Guyana, Honduras, India, Indonesia, Jamaica, Jordan, Kazakhstan, Kyrgyz Republic, Latvia, Lithuania, Malaysia, Mauritius, Mexico, Moldova, Mongolia, Montenegro, Morocco, Myanmar, Nepal, North Macedonia, Panama, Paraguay, Peru, Philippines, Poland, Romania, Serbia, Sri Lanka, Thailand, Tunisia, Turkey, Ukraine, Uruguay, Uzbekistan |
| 3 | Afghanistan, Angola, Benin, Burkina Faso, Burundi, Cameroon, Comoros, Congo, Dem. Rep., Cote d’Ivoire, Ethiopia, Gambia, The, Ghana, Guinea, Guinea-Bissau, Kenya, Madagascar, Malawi, Mali, Mauritania, Mozambique, Niger, Nigeria, Pakistan, Rwanda, Senegal, Sierra Leone, Togo, Uganda, Zambia |
| 4 | Argentina, Australia, Austria, Bahrain, Belgium, Brazil, Brunei Darussalam, Canada, China, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Hong Kong SAR, China, Hungary, Iceland, Ireland, Israel, Italy, Japan, Korea, Rep., Kuwait, Luxembourg, Macao SAR, China, Malta, Netherlands, New Zealand, Norway, Oman, Portugal, Qatar, Russian Federation, Saudi Arabia, Singapore, Slovak Republic, Slovenia, Spain, Sweden, Switzerland, United Kingdom, United States |

Nous ne pouvons pas dire si ces résultats sont les meilleurs mais nous voyons déjà que l’algorithme a reproduit la réalité. En effet nous voyons par exemple que presque tous les pays se trouvent dans le cluster 3, ce qui montre que ces pays sont plus homogènes au regard des variables que nous avons choisies.

# Conclusion

Au travers de ce travail nous avons réalisé la puissance et l’utilité des méthodes d’optimisation. Les domaines d’application de l’optimisation sont très variées

# Bibliographie

Ce travail a été réalisé en prenant en compte la reproductibilité des résultats. Le code source de ce travail peut être consulté à travers ce [lien](https://github.com/agailloty/Panel_Econometrics/blob/main/ROO/ROO.Rmd).

R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.