الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

Université 20 Aout 1955 de Skikda

Faculté des Sciences

Département de Mathématiques

جامعة 20 أوت 1955 ، سكيكدة

كلـــــــية العلـــــوم

قسم الرياضيات

N^0 : U.S/F.S/D.M/……/2022.

Faculté des Sciences
Département de Mathématiques

Mémoire

Présenté en vue de l'obtention du diplôme de
Master en Mathématiques

Traitement des données avec Excel et Python et Implémentation de la régression logistique en utilisant Sklearn.

Option : ANEDP

Par : ***LEBDIOUI Khawla*** ***&*** ***BOUREKHOUM Ikram***

Encadré par : MALLEM Khadidja M.C.B U.SKIKDA

Devant le jury :

Président : KAREK Chafia M.C.B U. SKIKDA
Examinateur : LECHHEB Samira M.C.B U. SKIKDA

Année : 2021/2022

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

﴿ لِّيَعْلَمَ أَن قَدْ أَبْلَغُوا۟ رِسَٰلَٰتِ رَبِّهِمْ وَأَحَاطَ بِمَا
لَدَيْهِمْ وَأَحْصَىٰ كُلَّ شَىْءٍ عَدَدًا ﴿٢٨﴾ ﴾

الجن: ٢٨

Remerciements

Avent tout, nous remercions **Dieu** le tout-puissant pour nous avoir donné la santé, le courage et la volonté d'étudier et pour nous avoir permis de réaliser ce modeste travail dans les meilleures conditions.

D'abord nous exprimons notre profond remerciement à notre directrice de mémoire, **Khadidja Mallem**, pour sa patience, et surtout pour sa confiance, ses remarques et ses conseils, sa disponibilité et sa bienveillance.

Nous voudrions également remercier les membres du jury pour avoir accepté d'évaluer ce travail et pour toutes leurs remarques et critique.

Merci fortement tous les enseignants du département pour leur soutien inestimable.

Nous tient à tous nos collègues de la promo pour les sympathiques moments qu'on a passés ensemble.

Et enfin, nous remercions tous ceux qui ont contribué de loin ou de près à la réalisation de ce travail.

Merci à vous tous

Dédicases

Toutes les lettres ne sauraient trouver les mots qu'il faut… Tous les mots ne sauraient exprimer la gratitude, l'amour, le respect, la reconnaissance… Aussi, c'est tout simplement que

Je dédie cette mémoire…?

A l'homme, mon précieux offre du dieu, qui doit ma vie, ma réussite et tout mon respect : mon cher père ***Noureddine.***

A la femme qui a souffert sans me laisser souffrir, qui n'a jamais dit non à mais exigences et qui n'a épargné aucun effort pour me rendre heureuse : ma mère ***Yasmina.***

A mes frères ***Mohamed, Fatah*** *et mon adorable petit frère* ***Adam,*** *je vous dis merci beaucoup je vous souhaite la santé, le bonheur et sur toute la réussite, je vous aime.*

A mes chères tantes ***Zahia, fayza*** *et* ***Hayat*** *qui m'accompagnent depuis mon enfance et qui ont été mon soutien.*

A mes grands-pères ***Ahmed, Tahar****, ma grande mère* ***Zohra****, mes oncles, que Dieu leur donne une longue et joyeuse vie, et a l'âme de ma grand-mère, que Dieu ait pitié d'elle.*

A tous mes amies de l'université ou de la résidence universitaire qui ont laissé une empreinte dans mon cœur.

Sans oublier l'essence rare, ma compagne ***Khawla****, avec qui j'ai eu l'immense honneur de travailler avec elle .je remercie beaucoup, je lui souhaite tout le bonheur dans sa future vie personnelle et plus de succès.*

Ikram.

Dédicases

Toutes les lettres ne sauraient trouver les mots qu'il faut… Tous les mots ne sauraient exprimer la gratitude, l'amour, le respect, la reconnaissance… Aussi, c'est tout simplement que

Je dédie cette mémoire…?

A l'homme, mon précieux offre du Dieu, qui doit ma vie, ma réussite et tout mon respect : mon cher père ***Azzedine****.*

A la femme qui a souffert sans me laisser souffrir, qui n'a jamais dit non à mais exigences et qui n'a épargné aucun effort pour me rendre heureuse : ma mère ***Habiba****.*

A ma petite famille ***Djihane****,* ***Nazim****,* ***Alaedinne*** *et* ***Ghani****, je vous dis merci beaucoup je vous souhaite la santé, le bonheur et sur tout la réussite, je vous aime.*

A mes grande mères ***Massaouda****,* ***Zieneb****, mon grand-père* ***Hassen****, mes oncles et mes tontes, que Dieu leur donne une longue et joyeuse vie, sans oublier mon grand-père* ***Ismain*** *que Dieu lui fasse miséricorde et fasse de lui un des gens du paradis.*

A tous les cousines, les voisins, les amies et toute personne qui occupe une place dans mon cœur.

Sans oublier la meilleur et le plus merveilleux binôme du métal pur ***Ikram****, qui j'ai l'honneur de travailler avec elle et je lui souhaite encore plus de réussite et de succès.*

Khawla.

Table des matières

Résumé

En 2019, l'informatique et la programmation sont des domaines d'étude en pleine émergence. Avec l'informatisation des entreprises, les données récoltées sont de plus en plus nombreuses. C'est ce qui a fait naître le terme très généraliste de Big Data. Et c'est à ce niveau qu'intervient le machine learning. Nous nous intéressons ici à l'apprentissage supervisé en consacrant aux problèmes de classification en utilisant la régression logistique. Dans un premier temps nous aborderons les principaux outils de statistique descriptive indispensables à l'exploration des données, en mettant l'accent sur la visualisation de l'information par exemple dans un histogramme la surface du barre représente la fréquence qui est une notion très importante pour le traitement des données . Nous avons réalisé deux projets : dans le premier projet nous avons appliqué un modèle du régression linéaire avec Excel et dans le deuxième projet nous avons appliqué un modèle du corrélation avec python. L'objectif est d'approfondir nos connaissances et nos outils de calculs dans le domaine de la statistique descriptive, apprendre à manier un tableur Excel et à manipuler Python. Nous nous intéressons ensuite au modèle de Régression Logistique, qui permet de résoudre des problèmes de classification binaires. qui consistent à prédire ou classer la valeur d'une variable discrète. Dans ce cas le modèle linéaire ne convient pas, on développe alors une nouvelle fonction, c'est la fonction logistique (sigma) qui a la particularité d'être toujours comprise en 0 et 1. A partir de cette fonction, il est possible de définir une frontière de décision. Typiquement, on définit un seuil à 0.5. Lorsqu'on teste notre modèle sur le Dataset, celui-ci nous donne des erreurs. L'ensemble de ces erreurs, c'est ce qu'on appelle la Fonction Coût. Pour la régression linéaire, la Fonction Coût donnait une courbe convexe (qui présente un unique minima). C'est ce qui fait que l'algorithme de Gradient Descent fonctionne. En revanche, utiliser cette fonction pour le modèle Logistique ne donnera pas de courbe convexe (dû à la non-linéarité) et l'algorithme de Gradient Descent se bloquera au premier minima rencontré, sans trouver le minimum global. Il faut donc développer une nouvelle Fonction Coût spécialement pour la régression logistique. On utilise alors la fonction logarithme pour transformer la fonction sigma en fonction convexe . L'algorithme de Gradient Descent s'applique exactement de la même manière que pour la régression linéaire.L'idée centrale du Machine Learning, c'est de laisser la machine trouver quels sont les paramètres de notre modèle qui minimisent la Fonction Coût. Enfin, nous avons étudié l'algorithme de Nearest Neighbour (le voisin le plus proche) qui permet de résoudre des problèmes de classification à plusieurs classes de façon simple et très efficace. et nous avons réalisé le projet de prédiction de survie du Titanic en utilisant cet algorithme.

MOTS CLES : **Apprentissage automatique, Régression logistique, Excel, Python.**

Abstract

The rapid increase of information and accessibility in recent years has activated a paradigm shift in algorithm design for artificial intelligence. Machine learning is the latest in a long line of attempts to distill human knowledge and reasoning into a form that is suitable for constructing machines and engineering automated systems.

Machine learning is an application of artificial intelligence that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.

As machine learning becomes more ubiquitous and its software packages become easier to use, it is natural and desirable that the low-level technical details are abstracted away and hidden from the practitioner. However, this brings with it the danger that a practitioner becomes unaware of the design decisions and, hence, the limits of machine learning algorithms.

The enthusiastic practitioner who is interested to learn more about the magic behind successful machine learning algorithms currently faces a daunting set of pre-requisite knowledge of Mathematics and statistics and how machine learning builds on it

This master's thesis is a study of data processing with Excel and Python and implementation of logistic regression using Sklearn., which help to create strong decision criteria for users to make the training dataset based on which machine can predict the proper output.

Key words : **Machine learning, Logistic regression, Excel, Python.**

الملخص

لقد كان لتعلم الآلة نصيب كبير من الإهتمام في السّنوات الأخيرة، فلقد أحدث ثورةً كبيرة في مختلف المجالات، ولذلك ظهرت العديد من الخوارزميات، أي برامج يمكنها التّعلم والتّحسين دون تدخلات بشريّة، يعتمد على الرّسومات البيانيّة المتعلّقة بالإحصاء الوصفي ومدى إرتباط البيانات ببعضها.

لذلك قمنا في هذه المذكّرة بشرح أهم خوارزميّات تعلّم الآلة الخاضع للإشراف، وخصّصنا بالذّكر الإنحدار اللّوجستيكي الذّي هو أحد أهم خوارزميّات التّصنيف، نظراً لخطواته البسيطة التّي تستخدم لتصنيف البيانات إلى فئات منفصلة، ويتم ذلك من خلال الدّالة اللّوجستيكية التّي تفصل بين صنفين مختلفين، على غرار مذكّرة "سارة لعور" السّابقة التّي تناولت الإنحدار الخطّي، وذلك بإستخدام لغة البرمجة بايثن، كما تطرّقنا إلى كيفيّة معالجة البيانات بإستخدام الإكسِل الذّي بدوره يحلّل البيانات الصّغيرة.

الكلمات المفتاحية:

تعلّم الآلة، الإنحدار اللّوجستيكي، لغة البرمجة بايثن، تطبيق الإكسِل.

INTRODUCTION

En 2019, Le Machine Learning ou bien l'apprentissage automatique est tout autour de nous. Il intervient chaque fois que nous cherchons un mot dans Google, une série sur Netflix, une vidéo sur YouTube, un produit sur Amazon. Grâce au Machine Learning, des millions de cancers peuvent être diagnostiqués chaque année, des milliards de spams et de virus informatiques sont bloqués pour protéger nos ordinateurs.

Les fondations du Machine Learning

Comprendre pourquoi le Machine Learning est utilisé [7],[10] , [8], [1], [2], [12]
Nous, les êtres humains, sommes quotidiennement confronté à des problèmes que nous cherchons à résoudre. Par exemple : Comment construire un pont plus solide ? Comment augmenter nos bénéfices ? Comment éliminer le cancer ? Ou tout simplement quelle route emprunter pour aller au travail ? Pour nous aider dans nos recherches, nous avons inventé l'ordinateur, qui permet de résoudre en quelques minutes des calculs qui nous prendraient des millions d'années à effectuer. Mais il faut savoir qu'un ordinateur ne sait en réalité faire qu'une chose : résoudre les calculs qu'on lui donne.
À partir de là, 2 situations possibles :
1.On connait le calcul à effectuer pour résoudre notre problème.
Dans ce cas, facile ! On entre ce calcul dans l'ordinateur, c'est ce qu'on appelle la programmation, et l'ordinateur nous donne le résultat.
Exemple : Déterminer la structure d'un pon.
2.On ne connait pas le calcul qui résout notre problème
Dans ce cas... on est bloqué. Impossible de donner à un ordinateur un calcul que nous ne connaissons pas. C'est comme vouloir poster une lettre que nous n'aurions pas écrite.

Exemples : Reconnaître un visage sur une photo, prédire le cours de la Bourse, éliminer le cancer, composer de la musique, conduire une voiture ….

Le Machine Learning a justement été inventé pour venir débloquer la situation 2 (quand on ne connait pas le calcul) en utilisant une technique audacieuse,

Laisser la Machine apprendre à partir d'expériences

Le Machine Learning consiste à laisser l'ordinateur apprendre quel calcul effectuer, plutôt que de lui donner ce calcul (c'est-à-dire le programmer de façon explicite). On attribue généralement ses débuts à la création du test de Turing, en 1950. C'est le mathématicien britannique Alan Turing qui imagine cette épreuve, censée déterminer si une machine peut simuler la pensée humaine [15]. Dans ce contexte, de premiers programmes « intelligents » voient le jour. En 1959, c'est le mathématicien américain Arthur Samuel [11] qui utilise pour la première fois le terme « machine learning » qui a développé un programme pouvant apprendre tout seul comment jouer aux Dames.

Les deux méthodes d'apprentissage

Pour donner à un ordinateur la capacité d'apprendre, on utilise des **méthodes d'apprentissage** qui sont fortement inspirées de la façon dont nous, les êtres humains, apprenons è faire des choses. [4]. Parmi ces méthodes, on compte : l'apprentissage supervisé (Supervised Learning) et **l'apprentissage non supervisé** (Unsupervised Learning).

L'apprentissage supervisé

On parle de l'apprentissage supervisé lorsque l'on fournit à une machine beaucoup d'exemples qu'elle doit étudier par exemple : lorsqu'on à apprendre le chinois, il faudra soit acheter un livre de traduction chinois-arabe, ou bien trouver un professeur de chinois. Le rôle du professeur ou du livre de traduction sera de superviser votre apprentissage en vous fournissant des exemples de traductions arabe-chinois que vous devrez mémoriser.

Les quatre notions de l'apprentissage supervisé

Il y' a quatre notions importantes dans l'apprentissage supervisé : le Dataset, le Modèle et ses paramètres , la Fonction Coût et l'Algorithme d'apprentissage.

Notion 1 : Apprendre à partir d'exemples (Dataset)

Pour apprendre la langue chinoise, on parle d'apprentissage supervisé lorsque l'on fournit à une machine beaucoup d'exemples (x, y) dans le but de lui faire apprendre la relation qui relie x à y.

— La variable y porte le nom de **target** (la cible). C'est la valeur que l'on cherche à prédire.

— La variable x porte le nom de **feature** (facteur). Un facteur influence la valeur de y, et

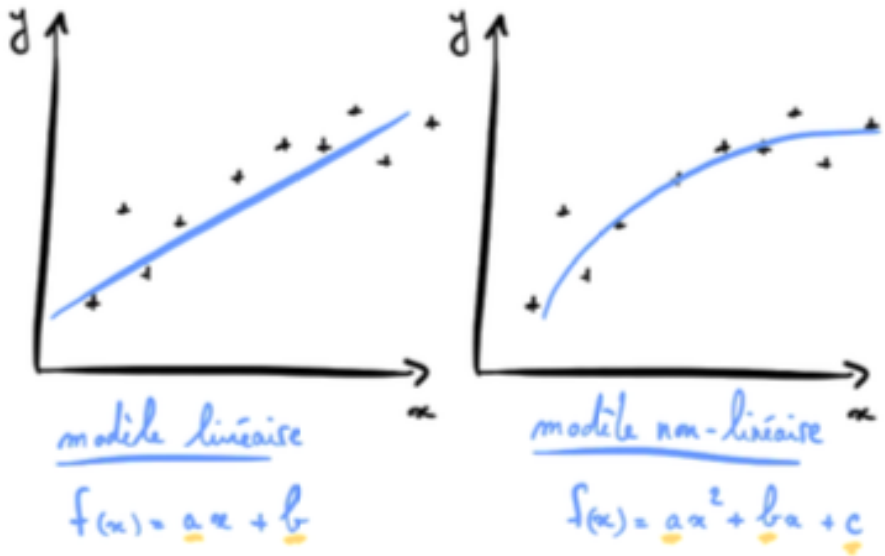
on a en général beaucoup de **features** (x_1,x_2,x_3 ...) dans notre Dataset que l'on regroupe dans une matrice X.

Ci-dessous, un Dataset qui regroupe des exemples d'appartements avec leur prix y ainsi que certaines de leurs caractéristiques (features).

Target	features		
y	x_1	x_2	x_3
Prix	Surface m2	N chambres	Qualité
€313,000.00	124	3	1.5
€2,384,000.00	339	5	2.5
€342,000.00	179	3	2
€420,000.00	186	3	2.25
€550,000.00	180	4	2.5
€490,000.00	82	2	1
€335,000.00	125	2	2

Notion 2 : Développer un modèle à partir du Dataset

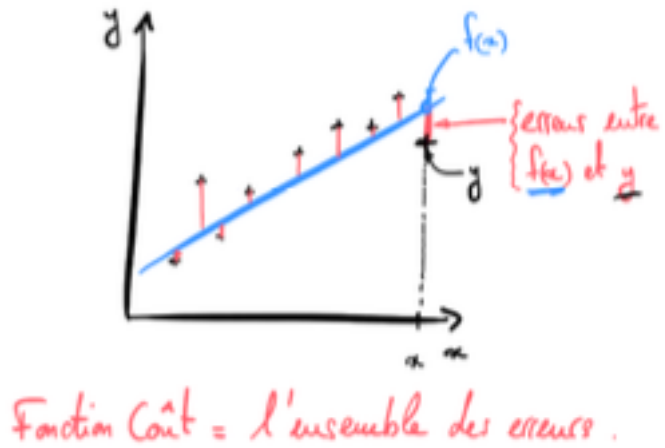
En Machine Learning, on développe un modèle à partir de ce Dataset. Il peut s'agir d'un modèle linéaire comme vous pouvez le voir à gauche, ou bien un modèle non-linéaire comme vous pouvez le voir à droite. Nous verrons dans ce livre comment choisir un modèle plutôt qu'un autre.



On définit a, b , c etc. comme étant les paramètres d'un modèle.

Notion 3 : Les erreurs de notre modèle - la Fonction-Coût

Autre chose à noter est qu'un modèle nous retourne des erreurs par rapport à notre Dataset. On appelle **Fonction Coût** l'ensemble de ces erreurs (le plus souvent on prend la moyenne quadratique des erreurs comme dans le chapitre 2).



Allons droit au but : Avoir un bon modèle, c'est avoir un modèle qui nous donne de petites erreurs, donc une petite **Fonction Coût**.

Notion 4 : Minimiser la Fonction Coût

L'objectif central en Supervised Learning, c'est de trouver les paramètres du modèle qui minimisent la Fonction Coût. Pour cela, on utilise un algorithme d'apprentissage, l'exemple le plus courant étant **l'algorithme de Gradient Descent**, (dans le deuxième et le troisième chapitre).

Les applications du Supervised Learning

Avec le Supervised Learning on peut développer des modèles pour résoudre deux types de problèmes :

- Les problèmes de **Régression**
- Les problèmes de **Classification**[13]

Dans les problèmes de régression, on cherche à prédire la valeur d'une variable **continue**, c'est-à-dire une variable qui peut prendre une **infinité** de valeurs. Par exemple :

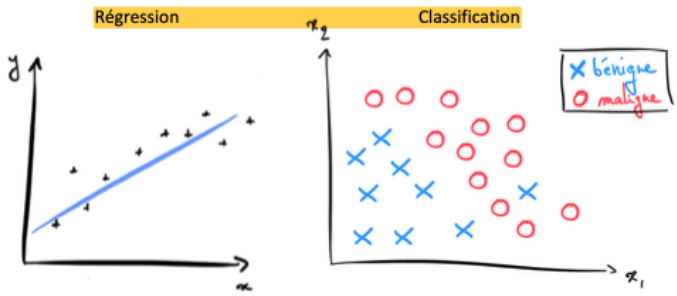
- Prédire le prix d'un appartement y selon sa surface habitable x
- Prédire la quantité d'essence consommée y selon la distance parcourue x

Dans un problème de classification, on cherche à classer un objet dans différentes classes, c'est-à-dire que l'on cherche à prédire la valeur d'une variable discrète (qui ne prend qu'un nombre fini de valeurs). Par exemple :

— Prédire si un email est un spam (classe $y = 1$) ou non (classe $y = 0$) selon le nombre de liens présent dans l'email

— Prédire si une tumeur est maligne ($y = 1$) ou bénigne ($y = 0$) selon la taille de la tumeur (x_1) et l'âge du patient (x_2)

Dans le cas d'un problème de classification, on représente souvent les classes par des symboles, plutôt que par leur valeur numérique $(0, 1, \dots)$



L'apprentissage non-supervisé

Supposons qu'on est , seul, en Chine, sans bouquin, sans traducteur, il existe tout de même une méthode pour apprendre le chinois. C'est l'apprentissage non-supervisé, dans ce mémoire on va pas introduire la notion de l'apprentissage non-supervisé .

Les quatre étapes essentielles pour l'apprentissage automatique ou bien en anglais " Machine Learning"

Le Dataset

En Machine Learning, tout démarre d'un Dataset qui contient nos données. Dans l'apprentissage supervisé, le Dataset contient les questions (x) et les réponses (y) au problème que la machine doit résoudre .

Le modèle et ses paramètres

A partir de ce Dataset, on crée un modèle, qui n'est autre qu'une fonction mathématique. Les coefficients de cette fonction sont les paramètres du modèle.

La Fonction Coût

Lorsqu'on teste notre modèle sur le Dataset, celui-ci nous donne des erreurs. L'ensemble de ces erreurs, c'est ce qu'on appelle la Fonction Coût.

L'Algorithme d'apprentissage

L'idée centrale du Machine Learning, c'est de laisser la machine trouver quels sont les paramètres de notre modèle qui minimisent la Fonction Coût.

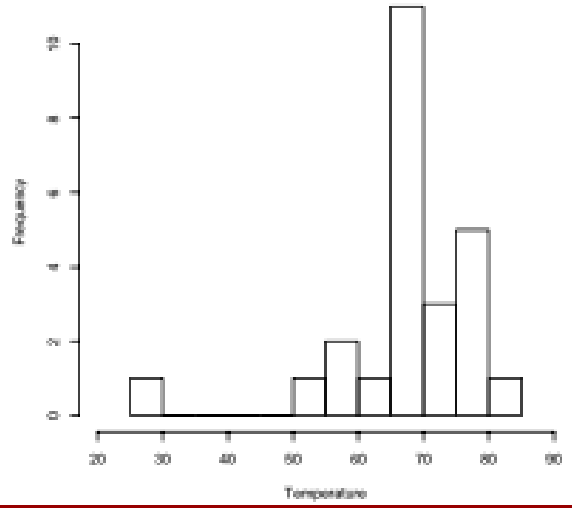
Plan du travaille

Tout d'abord nous avons commencé par une introduction générale sur l'apprentissage automatique en expliquant ces fondations et ces quatre étapes essentielles.
Les mathématiques se cache derrière l'apprentissage automatique pour cela nous avons importé dans le premier chapitre un résumé de statistique descriptive en abordons la notion du corrélation et en réalisant deux projets : dans le premier projet nous avons appliqué un modèle du régression linéaire avec Excel et dans le deuxième projet nous avons appliqué un modèle du corrélation avec python.
Dans le deuxième chapitre (resp. troisième chapitre) nous avons présenté le modèle du régression linéaire (resp. régression logistique).
Enfin le quatrième chapitre est consacré au programmation en effectuant le projet de prédiction de survie de Titanic.

CHAPITRE 1

RÉSUMÉ DE STATISTIQUE DESCRIPTIVE

Pourquoi la statistique descriptive est-elle importante ?
Parce qu' elle nous offre les moyens de résumer les données avec des chiffres et des graphiques. Voici un exemple : En janvier 1986, la navette spatiale Challenger s'est brisée peu après son décollage. L'accident a été causé par une pièce qui n'était pas conçue pour voler à la température exceptionnellement froide de 29◦ F au moment du lancement. Voici les températures de lancement des 25 premières missions de la navette (en degrés F) :
66,70,69,80,68,67,72,70,70,57,63,70,78,67,53,67,75,70,81,76,79,75,76,58,29 Et si on regarde cette longue liste de chiffres, il est assez difficile de voir ce qui se passe. Cependant, si on fait un graphique plus simple comme celui-ci, on voit immédiatement que la température de 29 degrés le jour du lancement est vraiment bien inférieure à toutes les autres températures.

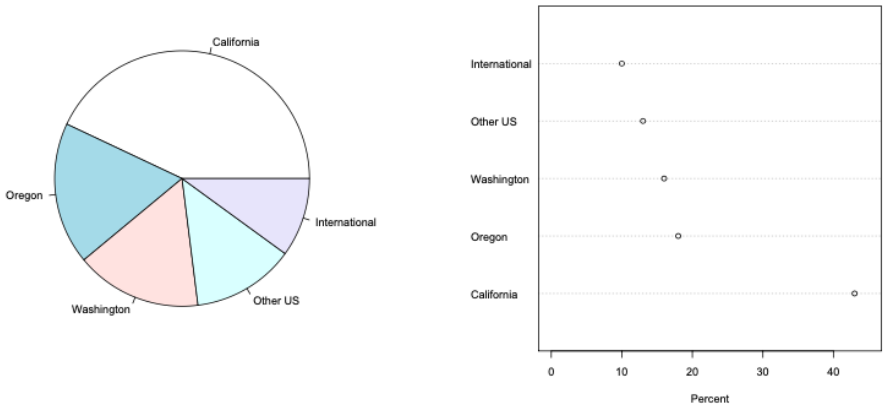


Il est préférable d'utiliser une représentation graphique pour communiquer des informations,

car les gens préfèrent regarder des images plutôt que des chiffres. Il existe de nombreuses façons de visualiser des données. La nature des données et l'objectif de la visualisation déterminent la méthode à choisir.

Le diagramme circulaire et le dot plot (regroupements de points)

Pour les données qualitatives on utilise le diagramme circulaire et le dot plot. Exemple : ces diagrammes représentent le pourcentage de l'origine géographique des étudiants d'une université de sud-ouest de l' USA.



Le dot plot facilite la comparaison des fréquences de différentes catégories, par exemple si on prend les portions des étudiants internationaux et les étudiants (other US) c'est difficile de les comparer en regardant le diagramme circulaire, par contre ceci est facile en utilisant le dot plot. Tandis que le diagramme circulaire permet plus facilement de présenter les pourcentages.

Le diagramme à barres

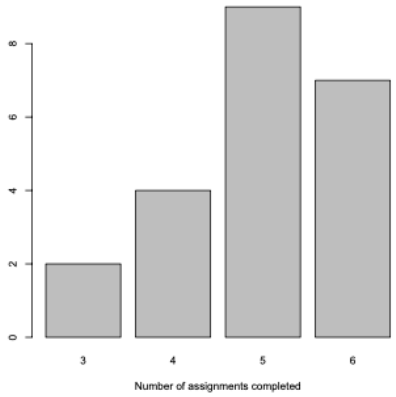


FIG. 1.1 – Diagramme à barres

Lorsque les données sont quantitatives (c'est-à-dire des nombres), elles doivent être placées sur une droite numérique.

Exemple : le diagramme dans la figure 1.1 représente le nombre de devoirs qui ont été effectués par 22 étudiants dans une classe.

L'histogramme

Un graphique à barres les barres ont toujours la même largeur mais l'histogramme permet d'utiliser des barres de différentes largeurs. L'histogramme dans la figure 1.2 représente l'âge

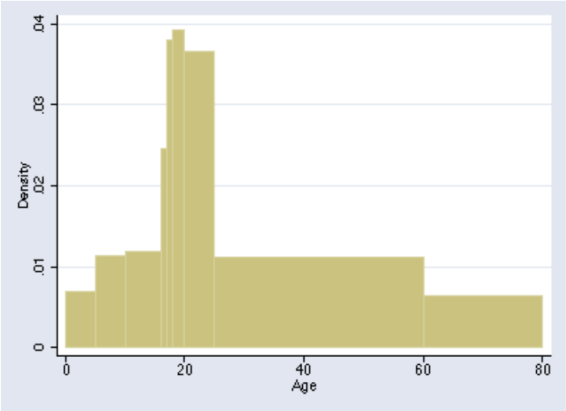


FIG. 1.2 – Histogramme

d'un certain nombre de personnes. Le point clé ici est que la surface du barre représente la fréquence c'est à dire : la surface = l'hauteur$\times$ la largeur

L'auteur représente la densité de fréquence divisée par la largeur de l'intervalle, par exemple, environ 14% de tous les sujets se situent dans la tranche d'âge 60-80 ans, car la surface correspondante est de (20 ans) x (0,7% par ans)=14%.

La boite à moustache ou le diagramme en boîte (the boxplot)

Un box-plot (figure 1.3) est un graphique simple composé d'un rectangle duquel deux droites sortent afin de représenter certains éléments des données.

Pour calculer la médiane d'une série statistique, il faut distinguer deux cas .

Premier cas :si l'effectif total N de la série est impair, La médiane M est la valeur située à la position $(N+1)/2$.

Deuxième cas :Si l'effectif total N de la série est pair, La médiane est égale à la demi somme des valeurs qui correspondent à $N/2$ et $(N/2)+1$.

Pour calculer des quartiles, on calcule $Q1$, ensuite $Q3$ et enfin $Q2 = M$.

Calcul de $Q1$: on divise l'effectif total par 4 (quartile) $N/4$ Le 1er quartile est la $(N/4)$ valeur

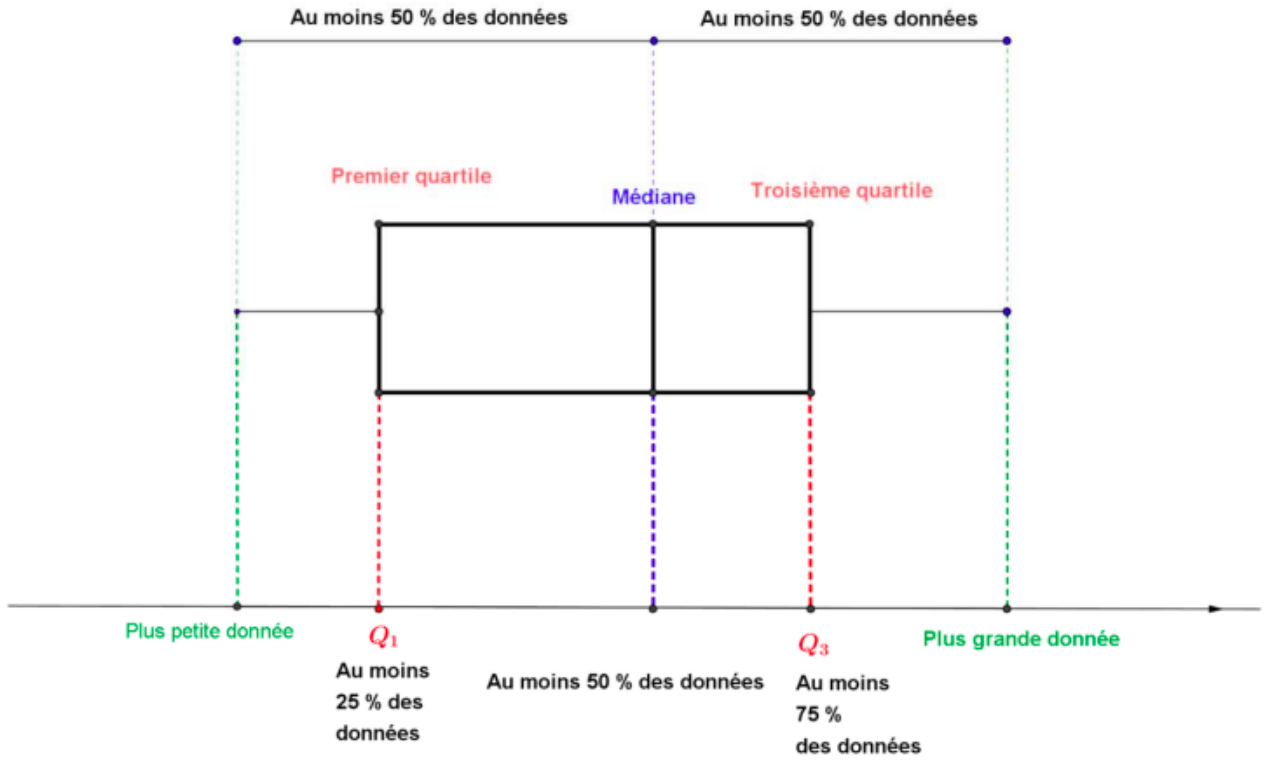


FIG. 1.3 – La boite à moustache

de la série statistique.

Calcul de $Q3$: pour $Q3$ on procède de la même façon mais à partir de la fin de la série statistique c'est -à -dire $(N - N/4)$ qui est égal à $3N/4$ Ainsi, la $(3N/4)e$ valeur est la valeur de $Q3$.

Exemple : on a relevé les notes de 24 élèves d'une classe lors d'un examen noté sur 100 points.

78	79	77	59	57	65	65	67
68	67	59	54	64	68	72	74
72	72	76	77	76	74	77	76

Comme il y a 24 valeurs la médiane est la moyenne entre la 12ème et la 13ème valeur soit $M = (72 + 72)/2 = 72$ le premier quartile est la 6ème valeur soit $Q1 = 65$ et le troisième quartile est la 8 ème valeur $Q3 = 76$.

La boite à moustache de cette série est dans la figure 1.4 : On peut comparer les résultats de cette classe avec les résultats d'une autre classe dont on sait que la note minimale est 47, la note maximale est 85, la médiane est 70, $Q1$ est 67 et $Q3$ est 76. On trace sur le même graphique

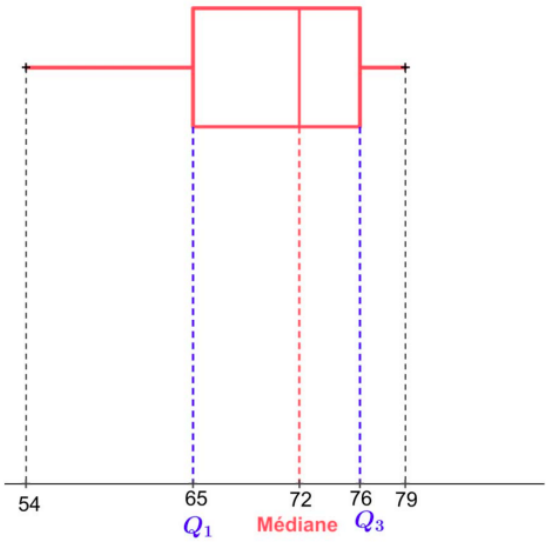


FIG. 1.4 – expemple1

la boîte à moustache de cette nouvelle série (figure :1.5). Cette deuxième classe semble un peu

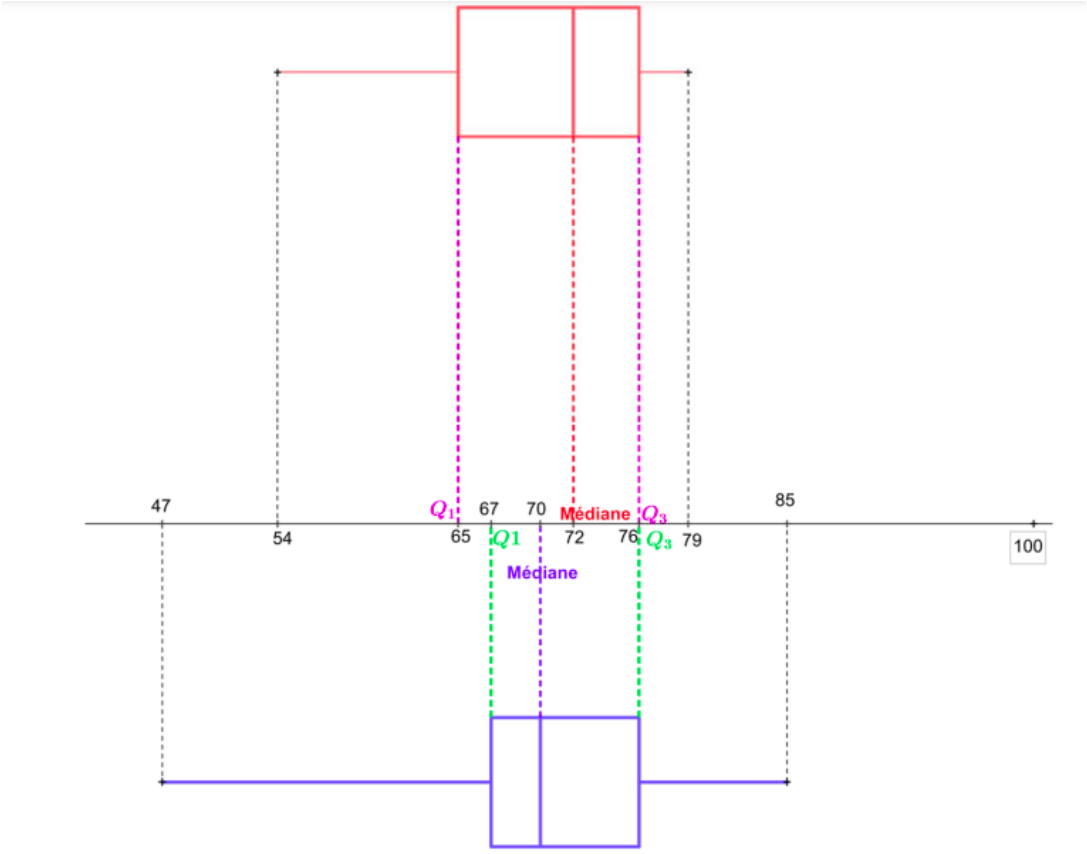


FIG. 1.5 – expemple2

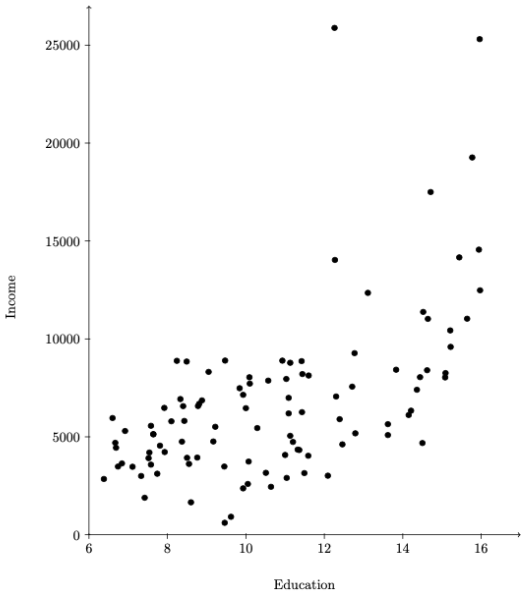
plus hétérogène (un minimum inférieur et un maximum supérieur) mais pour 50% des élèves (l'intérieur des boites) la deuxième classe est plus concentrée (boite moins large). Pour les deux classes 75% des élèves sont en dessous de 76 sur 100 .

Le nuage de points (The scatterplot)

Un nuage de points est un graphique qui représente chaque couple d'une distribution à deux types de variables strictement quantitatives. S'il existe un lien de dépendance entre les caractères étudiés, on place l'indépendant sur l'axe des abscisses et le dépendant sur l'axe des ordonnées.

L'exemple ici représente les années d'études et les salaires d'un certain nombre de personnes.

Par exemple, cette personne peut avoir environ 12 ans d'études et un revenu d'environ



15000$. Le nuage de points est très utile pour visualiser la relation entre deux variables. Par exemple, nous constatons que plus le niveau d'éducation augmente, plus le revenu a tendance à augmenter. De plus, nous constatons que pour de nombreuses années d'études, il semble y avoir une forte augmentation du revenu.

Fournir un contexte avec de petits multiples

Les analyses statistiques comparent généralement les données observées à une référence. Le contexte est donc essentiel pour l'intégrité du graphique [14]. Ce premier graphique (figure 1.6) représente une boîte à moustache pour chaque mois. Par exemple, la boîte à moustache du mois de janvier contient les données pour les 50 enregistrements de températures du mois de janvier. Cet affichage permet de comparer l'évolution des températures tout au long

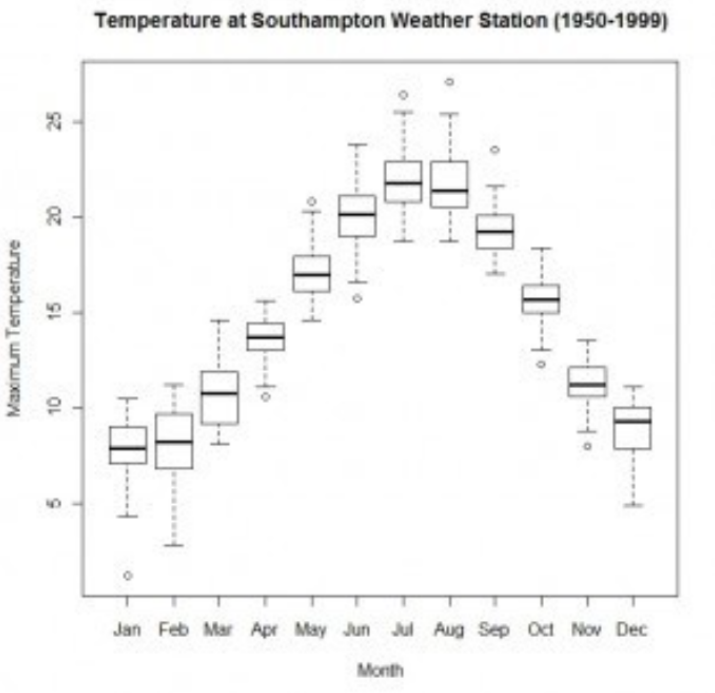


FIG. 1.6 – Des petits multiples de boite à moustache

de l'année. Ce deuxième graphique (figure 1.7) représente les taux de chômage mensuels

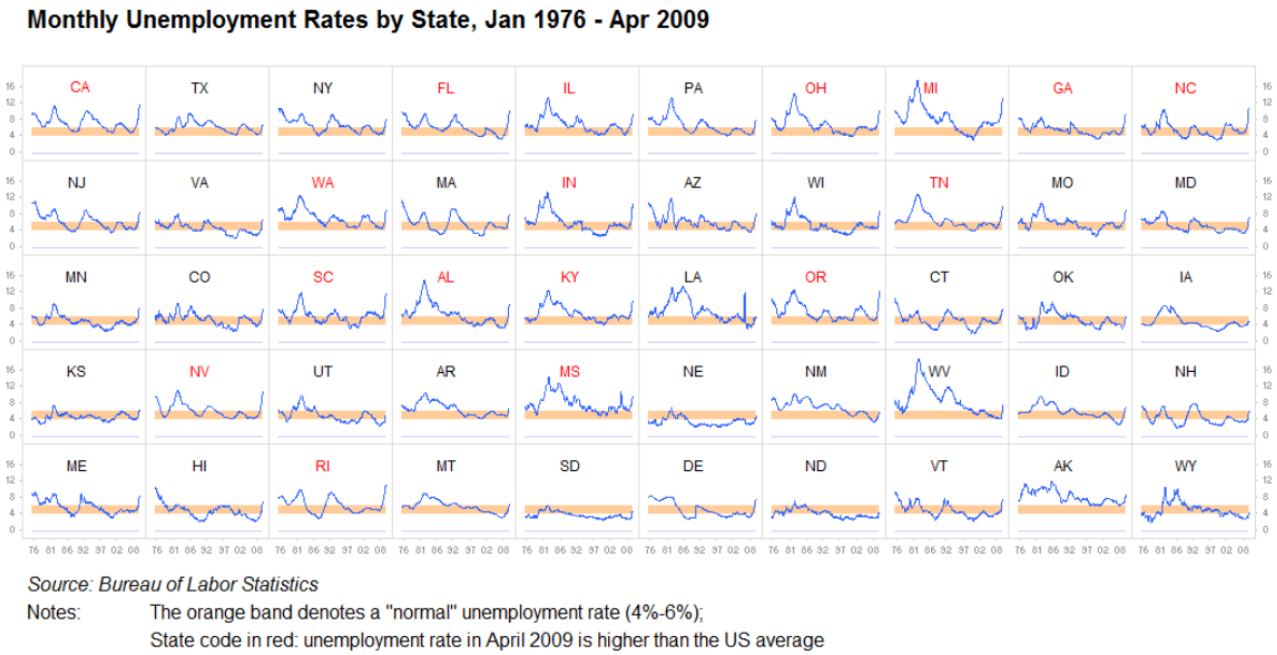


FIG. 1.7 – Des petits multiples représente le taux de chômage mensuels

pour chaque État de janvier 1976 à avril 2009. Chaque graphique est un graphique de série

chronologique. Cela signifie que sur l'axe horizontal, on a les années de 1976 à 2009, et sur l'axe vertical, on a le taux de chômage. La ligne qui est donnée dans chaque graphique montre comment le taux de chômage évolue au fil du temps. De plus, il y a une bande orange qui montre un taux de chômage normal entre 4 et 6%. En regardant ces petits multiples, on voit immédiatement qu'il y a une forte augmentation du taux de chômage pendant la crise financière dans des États comme la Californie et le Michigan, alors que certains États n'ont pas été touchés par la crise comme le Montana ou le Dakota du Sud.

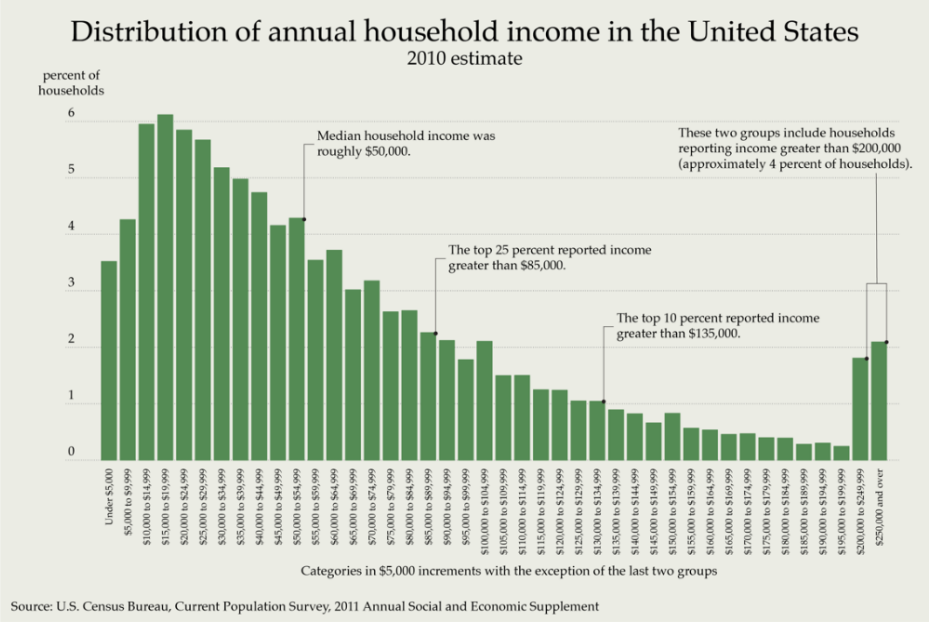
1.1 Résumé de mesures numérique

Moyenne vs. médiane

La Moyenne est la moyenne arithmétique d'une série de chiffres et la médiane est une valeur numérique qui sépare la moitié supérieure de la moitié inférieure d'un ensemble.

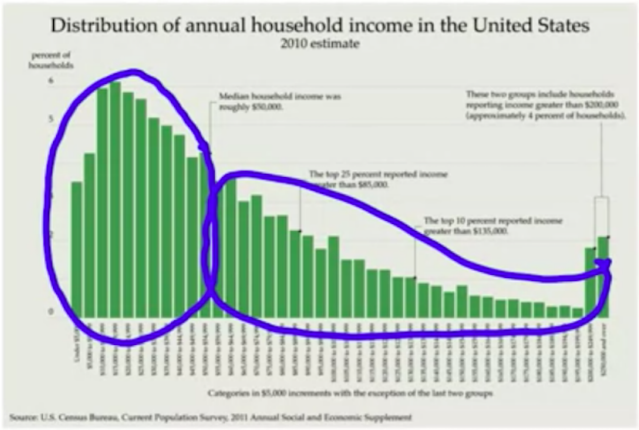
Exemple 1 : cet histogramme représente les revenus annuels des ménages aux États-Unis en 2010.

On rappelle que dans un histogramme la surface du barre représente la fréquence on voit ici

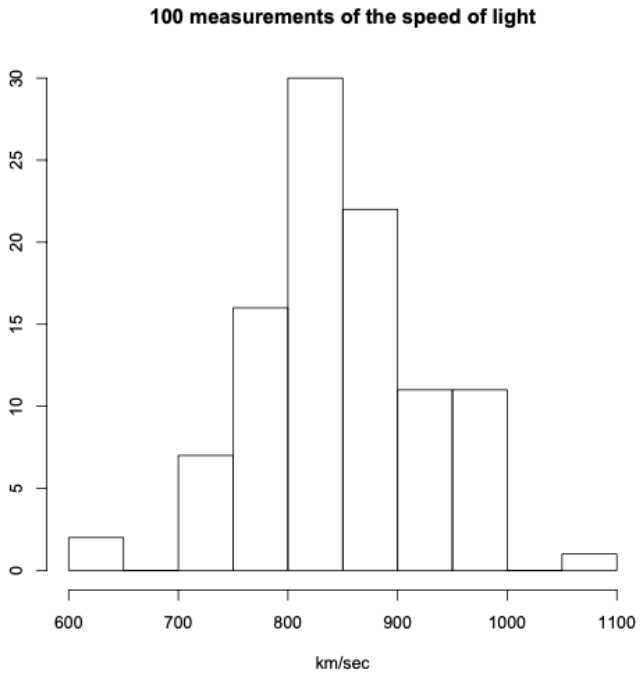


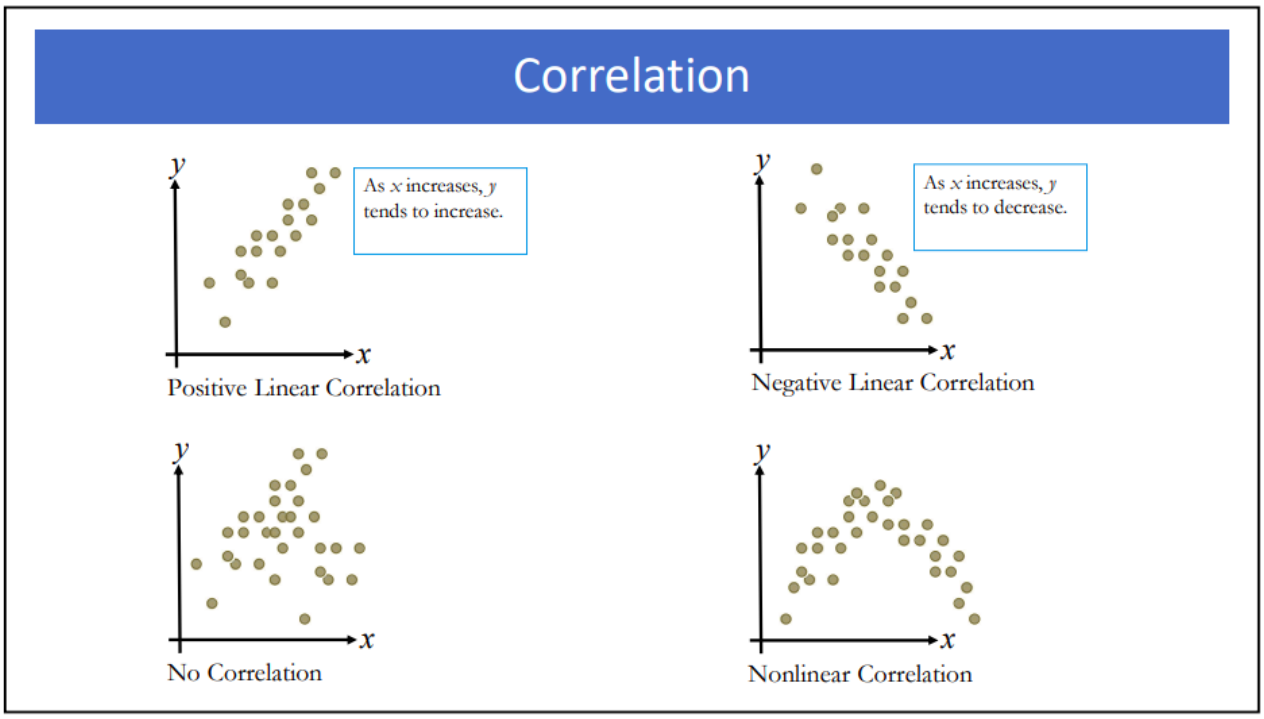
qu'environ la moitié de la surface de l'histogramme est inférieure à 50000$ et que la moitié de la surface de l'histogramme est supérieure à 50000$ donc la médiane vaut 50000$. Lorsque l'histogramme est incliné vers la droite (skewed to the right), la moyenne peut être beaucoup

plus grande que la médiane, dans ce cas on utilise la médiane.



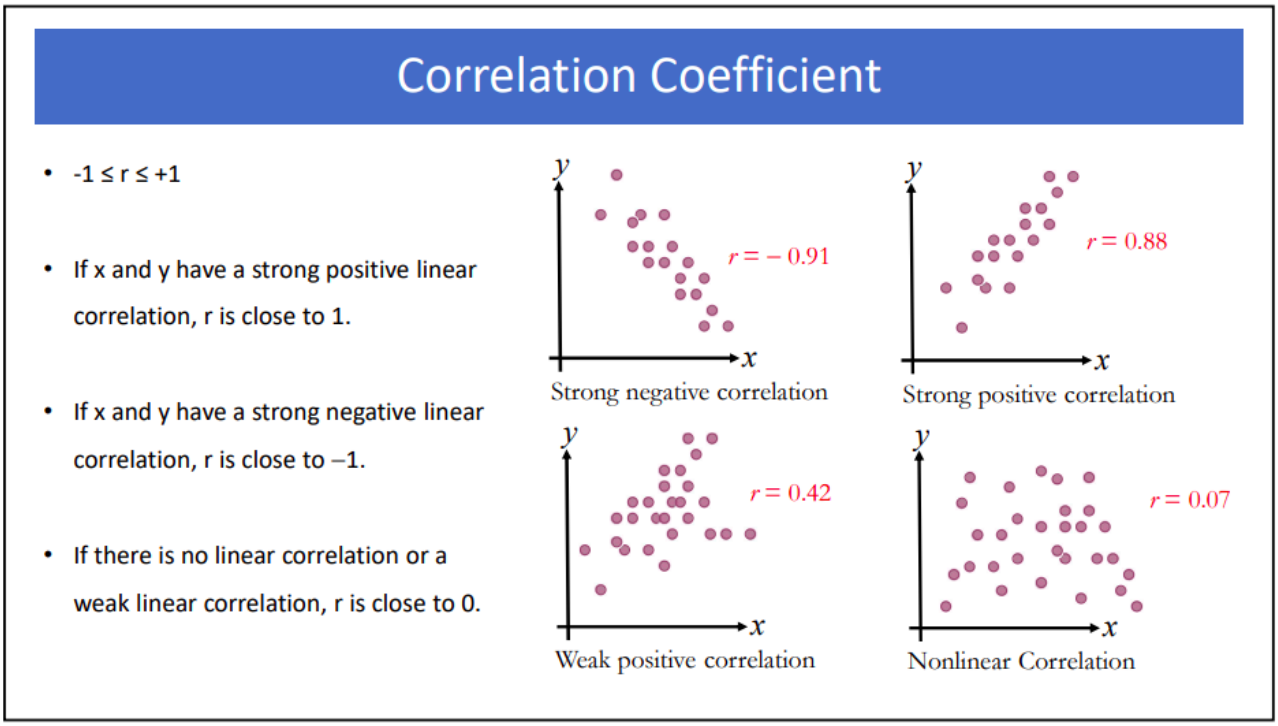
Exemple 2 :Cet histogramme représente les 100 premières mesures effectuées sur la vitesse de la lumière. La moyenne et la médiane sont identiques lorsque l'histogramme est symétrique.





1.2 Corrélation

Le coefficient de corrélation linéaire donne une mesure de l'intensité et du sens de la relation linéaire entre deux variables. Son calcul est assez complexe, c'est pourquoi on utilise souvent la calculatrice ou un logiciel. On s'intéresse ici à son interprétation.



Comment interpréter r

• Le coefficient de corrélation est compris entre -1 et 1.

• Plus le coefficient est proche de 1, plus la relation linéaire positive entre les variables est forte.

• Plus le coefficient est proche de -1, plus la relation linéaire négative entre les variables est forte.

• Plus le coefficient est proche de 0, plus la relation linéaire entre les variables est faible.

Exemple : les données suivantes représentent le nombre d'heures de télévision regardées par différents élèves pendant le week-end et le score de chaque élève ayant passé un test le lundi suivant. On calcule le coefficient de corrélation $r = -0.831$, Il existe une forte corrélation linéaire négative. Plus le nombre d'heures passées devant la télévision augmente, plus le test de score a tendance à diminuer.

Heurs, x	0	1	2	3	3	5	5	5	6	7	7	10
Score,y	96	85	82	74	95	68	76	84	58	65	75	50

Hours, x	0	1	2	3	3	5	5	5	6	7	7	10	54
Test score, y	96	85	82	74	95	68	76	84	58	65	75	50	908
xy	0	85	164	222	285	340	380	420	348	455	525	500	3724
x^2	0	1	4	9	9	25	25	25	36	49	49	100	332
y^2	9216	7225	6724	5476	9025	4624	5776	7056	3364	4225	5625	2500	70836

$$r = \frac{n\sum xy - (\sum x)(\sum y)}{\sqrt{n\sum x^2 - (\sum x)^2}\sqrt{n\sum y^2 - (\sum y)^2}} = \frac{12(3724) - (54)(908)}{\sqrt{12(332) - 54^2}\sqrt{12(70836) - (908)^2}} \approx -0.831$$

- **There is a strong negative linear correlation.**
- **As the number of hours spent watching TV increases, the test scores tend to decrease.**

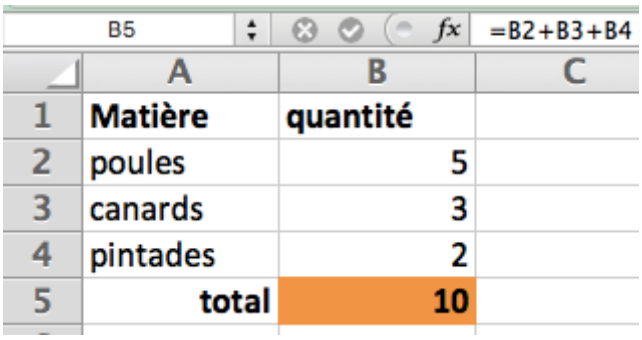
1.3 Indicateurs statistiques dans Excel

Une fonction est une formule prédéfinie qui vous permet de gagner du temps. Par exemple, utilisez la fonction SOMME pour additionner des nombres ou des cellules en grande quantité, et la fonction PRODUIT pour les multiplier. Les fonctions commencent par le signe "=" pour les distinguer des textes.

Les fonctions effectuent des opérations portant sur les cellules désignées par leur référence : par exemple $B6$ pour "Colonne B, ligne 6".

la fonction SOMME dans Excel : Syntaxe : SOMME(nombre1 ;nombre2 ;. . .)

Saisir dans la cellule $B6 := SOMME(B2 : B5)$, ou : SOMME(cellule départ :cellule arrivée) ici



on indique qu'on souhaite additionner la cellule B2 jusqu'à la cellule B5 ce qui peut également s'écrire =SOMME(B2 ;B3 ;B4 ;B5) ou encore =B2+B3+B4+B5.

la formule MOYENNE dans Excel : Syntaxe : MOYENNE(nombre1 ;nombre2 ;. . .)

saisir dans la cellule $B6 := MOYENNE(B2 : B5)$

B6 =MOYENNE(B2:B5)

	A	B	C
1	**Matière**	**note sur 20**	
2	français	12	
3	anglais	16	
4	maths	9	
5	géographie	13	
6	**total**	**12,5**	

Les 4 opérations arithmétiques de base dans Excel

l'addition avec Excel : Syntaxe : =nombre1 + nombre2 + nombre3 . . .

saisir dans la cellule B5 : =B2+B3+B4 Le résultat donne le nombre total d'animaux : 10

B5 =B2+B3+B4

	A	B	C
1	**Matière**	**quantité**	
2	poules	5	
3	canards	3	
4	pintades	2	
5	**total**	**10**	

la soustraction avec Excel : Syntaxe : =nombre1 – nombre2 – nombre3 . . .

saisir dans la cellule B6 : =B2-B3-B4-B5

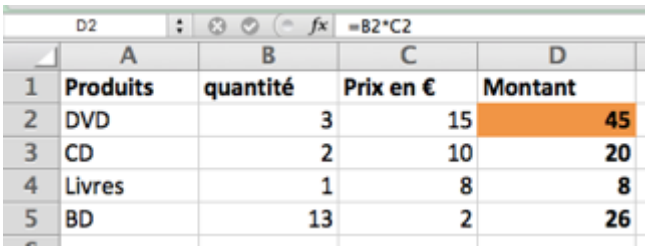
B6 =B2-B3-B4-B5

	A	B
1		**montant €**
2	**Argent donné**	20
3	achat journal	*3*
4	achat enveloppes	*8*
5	achat timbres poste	*2*
6	**monnaie reçue**	**7**

Ici on indique qu'on souhaite soustraire les achats effectués de l'argent donné au commerçant afin de calculer la monnaie que devra rendre celui-ci.

la multiplication avec Excel : Syntaxe : =nombre1 * nombre2 * nombre3 . . .

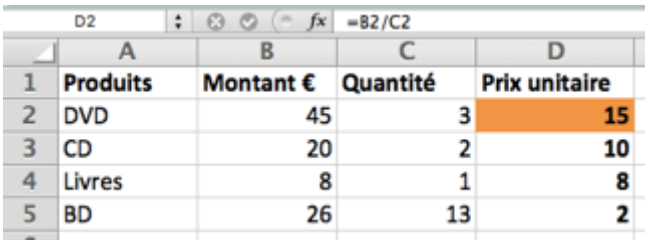
saisir dans la cellule D2 : =B2*C2 Ici on calcule le montant des DVD en multipliant la quantité par leur prix.



	A	B	C	D
1	**Produits**	**quantité**	**Prix en €**	**Montant**
2	DVD	3	15	**45**
3	CD	2	10	**20**
4	Livres	1	8	**8**
5	BD	13	2	**26**

la division avec Excel : Syntaxe : =nombre1 / nombre2 . . .

Saisir dans la cellule $D2 := B2/C2$

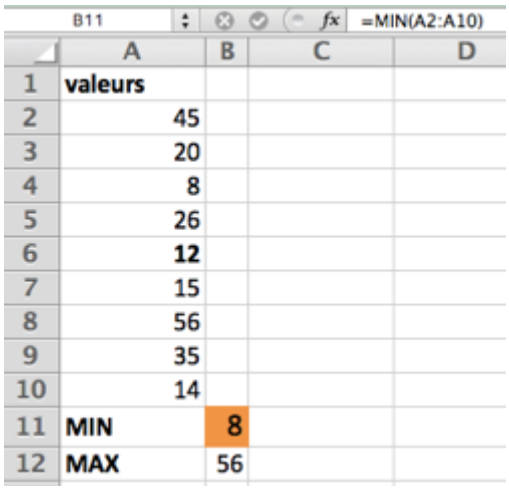


	A	B	C	D
1	**Produits**	**Montant €**	**Quantité**	**Prix unitaire**
2	DVD	45	3	**15**
3	CD	20	2	**10**
4	Livres	8	1	**8**
5	BD	26	13	**2**

ici on calcule le prix unitaire d'un DVD (15€) en divisant le montant DVD (45€) par la quantité (3) de DVD.

les fonctions MAX et MIN avec Excel : Syntaxe : =MIN(cellule départ :cellule arrivée) et =MAX(cellule départ :cellule arrivée)

Exemple, nous entrons 9 valeurs différentes de A2 à A10 et nous souhaitons connaître le plus



	A	B	C	D
1	**valeurs**			
2	45			
3	20			
4	8			
5	26			
6	**12**			
7	15			
8	56			
9	35			
10	14			
11	**MIN**	**8**		
12	**MAX**	56		

petit chiffre (ici 8) et aussi le plus grand nombre (ici 56). saisir dans $B11 = MIN(A2 : A10)$ et saisir dans $B12 = MAX(A2 : A10)$.

Médiane, mode, écart type et variance

— **La syntaxe du médiane** : MEDIANE(nombre1 ; nombre2 ;. . .)

— **La syntaxe de mode** MODE(nombre1 ; nombre2 ;. . .)

— **La syntaxe de l'écart type** STDEV.S(nombre1 ; nombre2 ;. . .)

— **La syntaxe du variance** VAR(nombre1 ; nombre2 ;. . .)

1.4 Projet1 : régression linéaire avec Excel

Les données du challenge se trouvent dans la feuille Excel suivante : `https://tinyurl.com/3w7nj6zt` qui contient un tableau de 228 lignes et 20 colonnes

L'objectif est de répondre aux questions ci-dessous avec Excel. Nous pouvons utiliser le compte

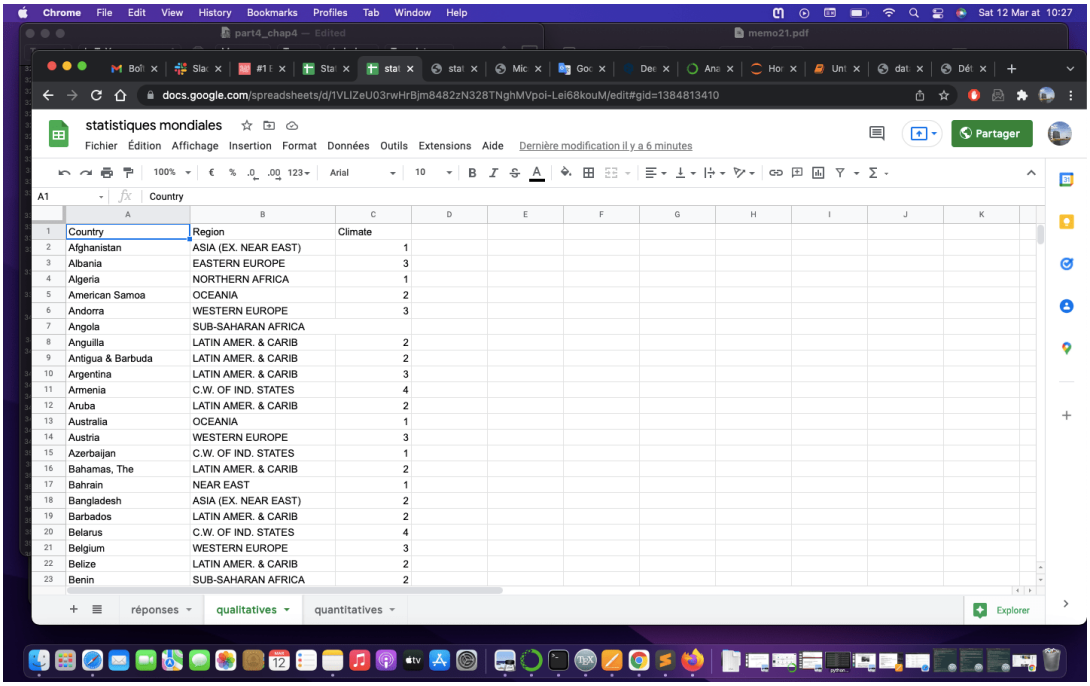


FIG. 1.8 – Capture de notre travail sur Google sheets

Google (Google Sheets) ou le compte Outlook (Excel).

1. Combien y a-t-il de variables quantitatives (numériques) ? De variables qualitatives ? Attention, parfois un groupe/une catégorie peut être matérialisé par un chiffre. Cependant, il s'agit toujours d'une variable qualitative. Scinder les données en mettant les variables qualitatives et quantitatives dans 2 feuilles à part. nommer les feuilles : "qualitatives" et "quantitatives".
2. Pour les variables quantitatives, calculer les indicateurs statistiques suivant : moyenne, médiane, maximum, minimum, mode, écart type, variance. Pour ce faire, utiliser des fonctions statistiques.
3. Effectuer une représentation en nuage de points (scatterplot) en croisant la colonne "Area" en axe X avec une autre colonne pertinente. Qu'en concluons nous ?
4. Choisir une variable quantitative et calculer ses quartiles.

	16025	274	58,5	
	2460492	5860	419,9	
	273008	266000	1	
	21456188	527970	40,6	
	11502010	752614	15,3	
	12236805	390580	31,3	
Moyenne	28740284,37	598226,9559	379,0471366	
Médiane	4786994	86600	78,8	
Somme	6524044551	135797519	86043,7	
Maximum	1313973713	17075200	16271,5	
minimum	7026	2	0	
mode	#N/A	102	13,8	
écart type	117891326,5	1790282,244	1660,185825	7
variance,	1,38984E+16	3205110512228	2756216,972	5

FIG. 1.9 – Calcul des indicateurs statistiques

5. Calculer le coefficient de corrélation pour les colonnes "Population" et "Area". Qu'en conclus-tu ? Idem pour les colonnes "Infant mortality" et "GDP", et enfin pour les colonnes "Phones" et "Arable".

Réponses

Combien y a-t-il de variables dans le fichier : 20 variables.

Combien y a-t-il d'observations ? 227 observations.

Combien y a-t-il de variables quantitatives ? 17 , De variables qualitatives ? 3

Pour les variables quantitatives, nous avons calculé les indicateurs statistiques suivant : moyenne, médiane, maximum, minimum, mode, écart type, variance ; en utilisant des fonctions statistiques (figure 1.9)

La figure 1.10 est une représentation en nuage de points (scatterplot) en croisant la colonne "Area" en axe X avec la colonne "Population" . Qu'en concluons nous ?il y a une corrélation faible entre la population et la surface et nous concluons que la population est située entre 10000 km carré et 1000000 km carré. nous allons bien vérifié ce résultat en calculant le coefficient de corrélation $r = 0.47$.

Nous avons choisi la variable quantitative de la population et nous avons calculé ses quartiles en utilisant la syntaxe : **=QUARTILE(B2 :B228 ;1)** ceci donne $Q1$ et **=QUARTILE(B2 :B228 ;3)** qui donne $Q3$

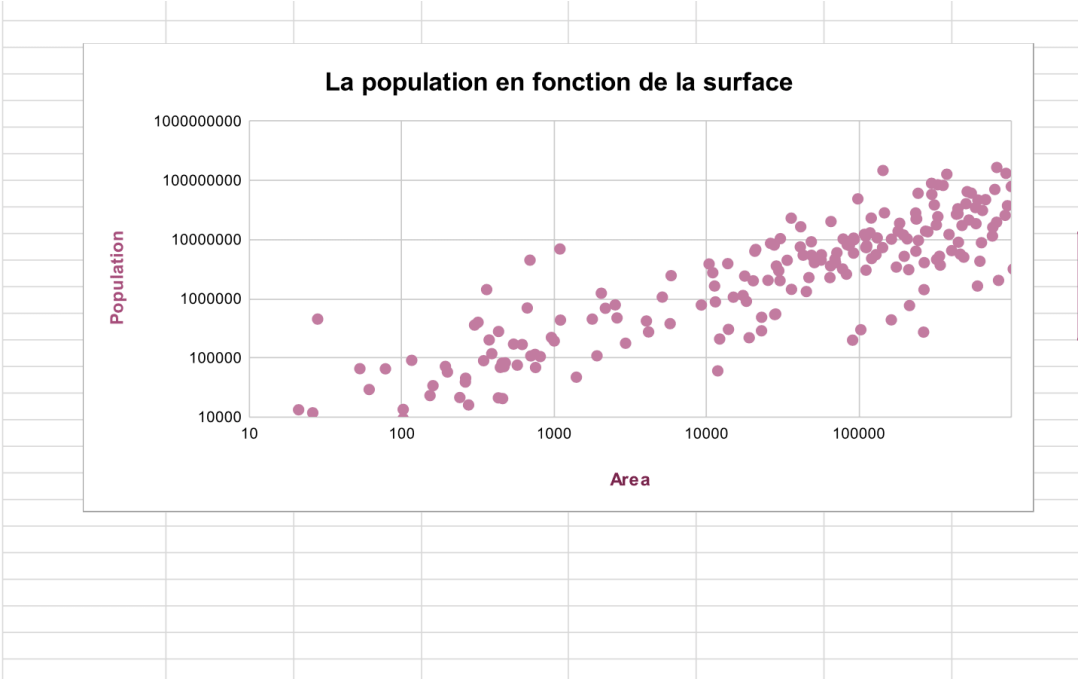


FIG. 1.10 – Le nuage de points de la population en fonction de la surface

quartiles1	437624	4647,5	29,15
quartiles3	17497772,5	441811	190,15
coefficient de corrélation Poulations/Area	0,4699850837	Le lien de corrélation est directe entre Poulation et area	
coefficient de corrélation Infant mortality /GDP	-0,6007739663	Le lien de corrélation est inversible entre infant mortality et GDP	
coefficient de corrélation Phones et Arable	0,06102155254	il y a pas un lien de corrélation entre phone et arable	

FIG. 1.11 – Calcul des quartiles et le coefficient de corrélation

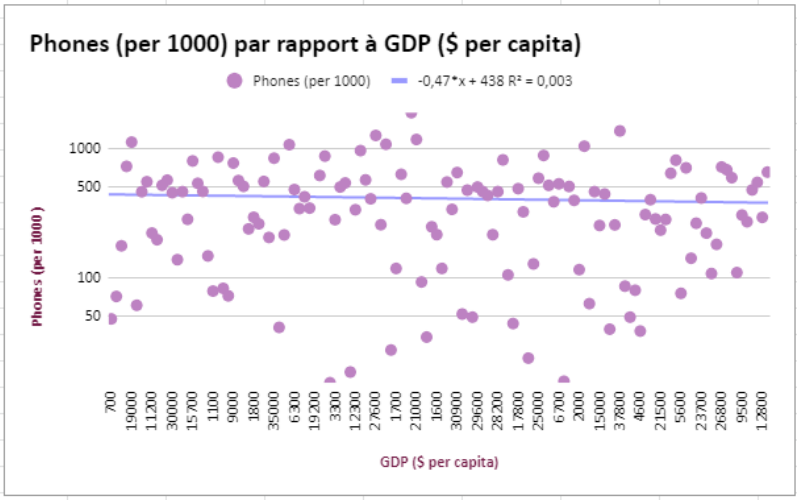
Nous avons calculé le coefficient de corrélation pour les colonnes "Population" et "Area" en utilisant la syntaxe : **=CORREL(B2 :B228 ;C2 :C228)** Idem pour les colonnes "Infant mortality" et "GDP", et enfin pour les colonnes "Phones" et "Arable" (figure 1.11).

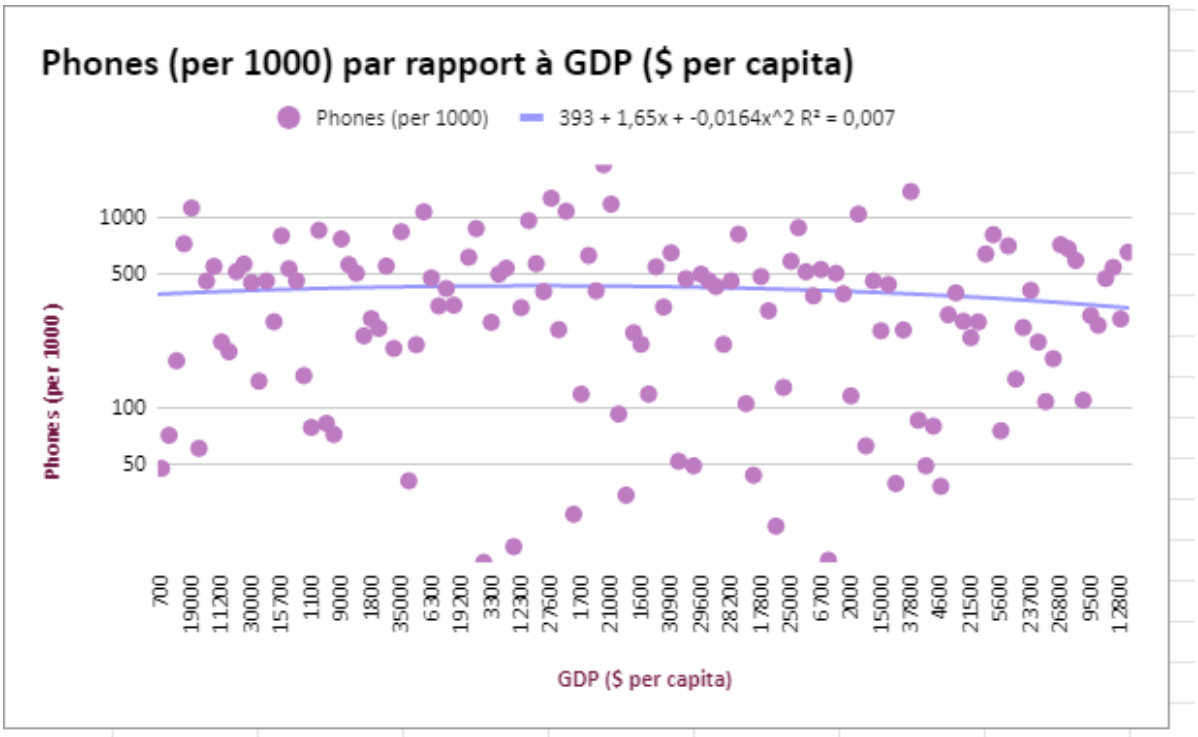
Dans ce deuxième challenge nous allons devoir tracer une régression linéaire entre une variable explicative X et une variable cible y de nos choix. Nous ferons de même avec d'autres variables pour tracer une régression polynomiale. Les données à utiliser seront celles du dataset de statistiques mondiales `https://tinyurl.com/3w7nj6zt`.

Réponses

Régression linéaire Régression poynomiale

GDP ($ per capita)	Phones (per 1000)
700	3,2
4500	71,2
6000	78,1
8000	259,5
19000	497,2
1900	7,8
8600	460
11000	549,9
11200	220,4
3500	195,7
28000	516,1
29000	565,5
30000	452,2
3400	137,1
16700	460,6
16900	281,3
1900	7,3
15700	481,9





1.5 Projet 2 : corrélation avec Python

Dans un notebook Jupyter nous avons réalisé ce travaille `https://tinyurl.com/3ubdtuzf`
Nous avons utilisé le Dataset 'vega-datasets' qui correspond aux véhicules.

1.5.1 Importer le Dateset

[1]:
```
import pandas as pd
import seaborn
import matplotlib.pyplot as plt
from vega_datasets import data
```

1.5.2 Data

[2]:
```
cars = data.cars()
cars.tail()


# sorted(cars['Cylinders'].unique())
```

[2]:

	Name	Miles_per_Gallon	Cylinders	Displacement	Horsepower	Weight_in_lbs	Acceleration	Year	Origin
401	ford mustang gl	27.0	4	140.0	86.0	2790	15.6	1982-01-01	USA
402	vw pickup	44.0	4	97.0	52.0	2130	24.6	1982-01-01	Europe
403	dodge rampage	32.0	4	135.0	84.0	2295	11.6	1982-01-01	USA
404	ford ranger	28.0	4	120.0	79.0	2625	18.6	1982-01-01	USA
405	chevy s-10	31.0	4	119.0	82.0	2720	19.4	1982-01-01	USA

[3]:
```
# view columns (copy and paste to make new dataframe)
cars.columns
```

[3]:
```
Index(['Name', 'Miles_per_Gallon', 'Cylinders', 'Displacement', 'Horsepower',
       'Weight_in_lbs', 'Acceleration', 'Year', 'Origin'],
      dtype='object')
```

[4]:
```
# new dataframe with only columns with numbers
cars = cars[['Acceleration', 'Cylinders', 'Displacement', 'Horsepower',␣
 ↪'Miles_per_Gallon', 'Weight_in_lbs']]
cars.head()
```

[4]:

	Acceleration	Cylinders	Displacement	Horsepower	Miles_per_Gallon	\
0	12.0	8	307.0	130.0	18.0	
1	11.5	8	350.0	165.0	15.0	
2	11.0	8	318.0	150.0	18.0	
3	12.0	8	304.0	150.0	16.0	
4	10.5	8	302.0	140.0	17.0	

	Weight_in_lbs
0	3504
1	3693
2	3436
3	3433
4	3449

1.5.3 Corrélation

[5]:
```
cars.corr()
```

[5]:

	Acceleration	Cylinders	Displacement	Horsepower	\
Acceleration	1.000000	-0.522452	-0.557984	-0.697124	
Cylinders	-0.522452	1.000000	0.951787	0.844158	
Displacement	-0.557984	0.951787	1.000000	0.898326	
Horsepower	-0.697124	0.844158	0.898326	1.000000	
Miles_per_Gallon	0.420289	-0.775396	-0.804203	-0.778427	
Weight_in_lbs	-0.430086	0.895220	0.932475	0.866586	

	Miles_per_Gallon	Weight_in_lbs
Acceleration	0.420289	-0.430086
Cylinders	-0.775396	0.895220
Displacement	-0.804203	0.932475
Horsepower	-0.778427	0.866586
Miles_per_Gallon	1.000000	-0.831741
Weight_in_lbs	-0.831741	1.000000

[6]:
```
# correlation coefficent using scipy stats
from scipy import stats


stats.pearsonr(cars.Cylinders, cars.Acceleration)
```

[6]:
```
(-0.5224515124210524, 8.183538342206756e-30)
```

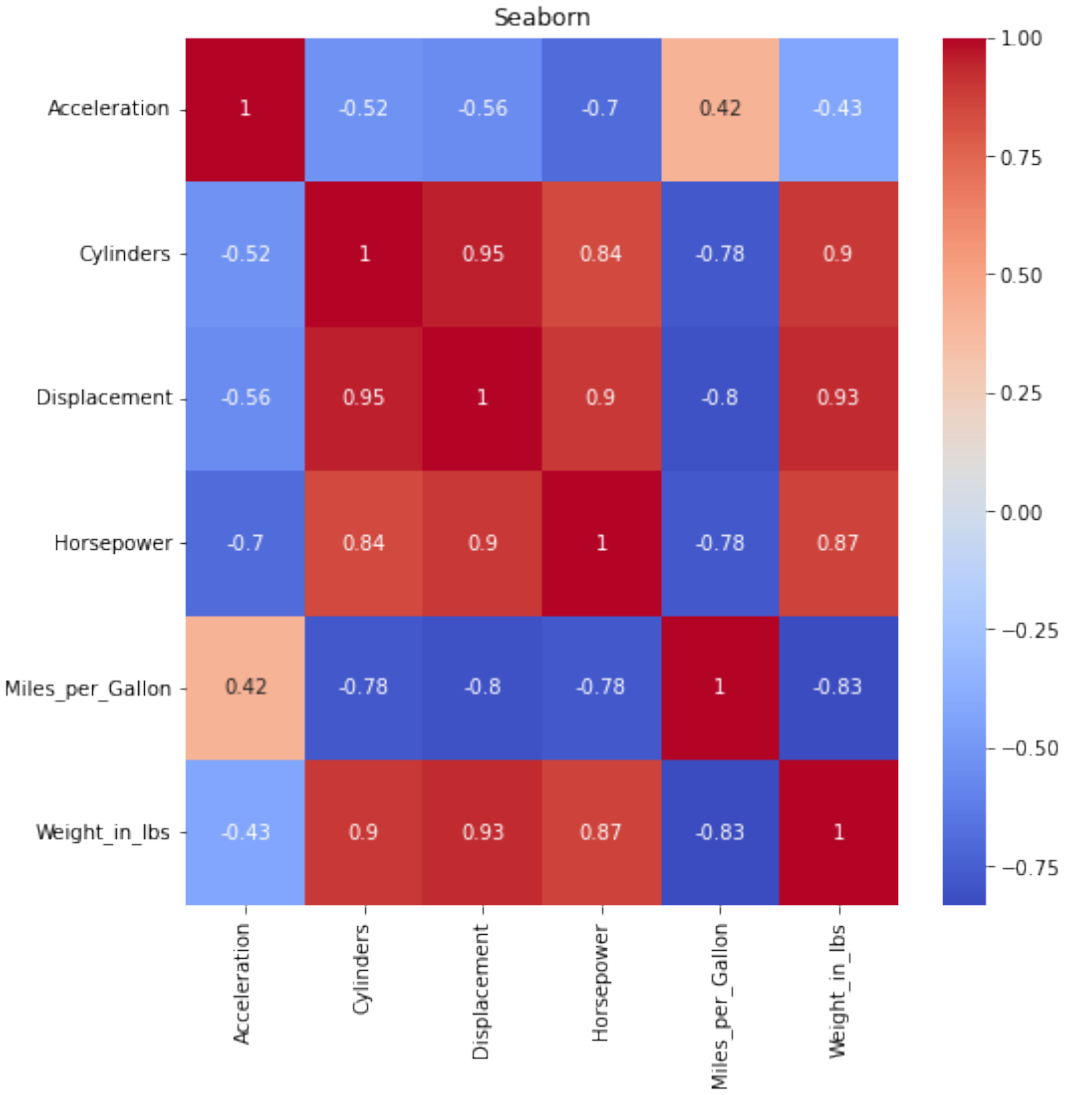
[7]:
```
from IPython.display import IFrame
wiki = IFrame(src='https://en.wikipedia.org/wiki/
 ↪Pearson_correlation_coefficient',
                      width=1000,
                      height=400)
display(wiki)
```

```
<IPython.lib.display.IFrame at 0x7f931ce975e0>
```

1.5.4 Heatmap corrélation avec Seborn

[8]:
```
plt.figure(figsize=(8,8))
seaborn.heatmap(cars.corr(), annot=True, cmap="coolwarm").set_title('Seaborn')
```

[8]:
```
Text(0.5, 1.0, 'Seaborn')
```



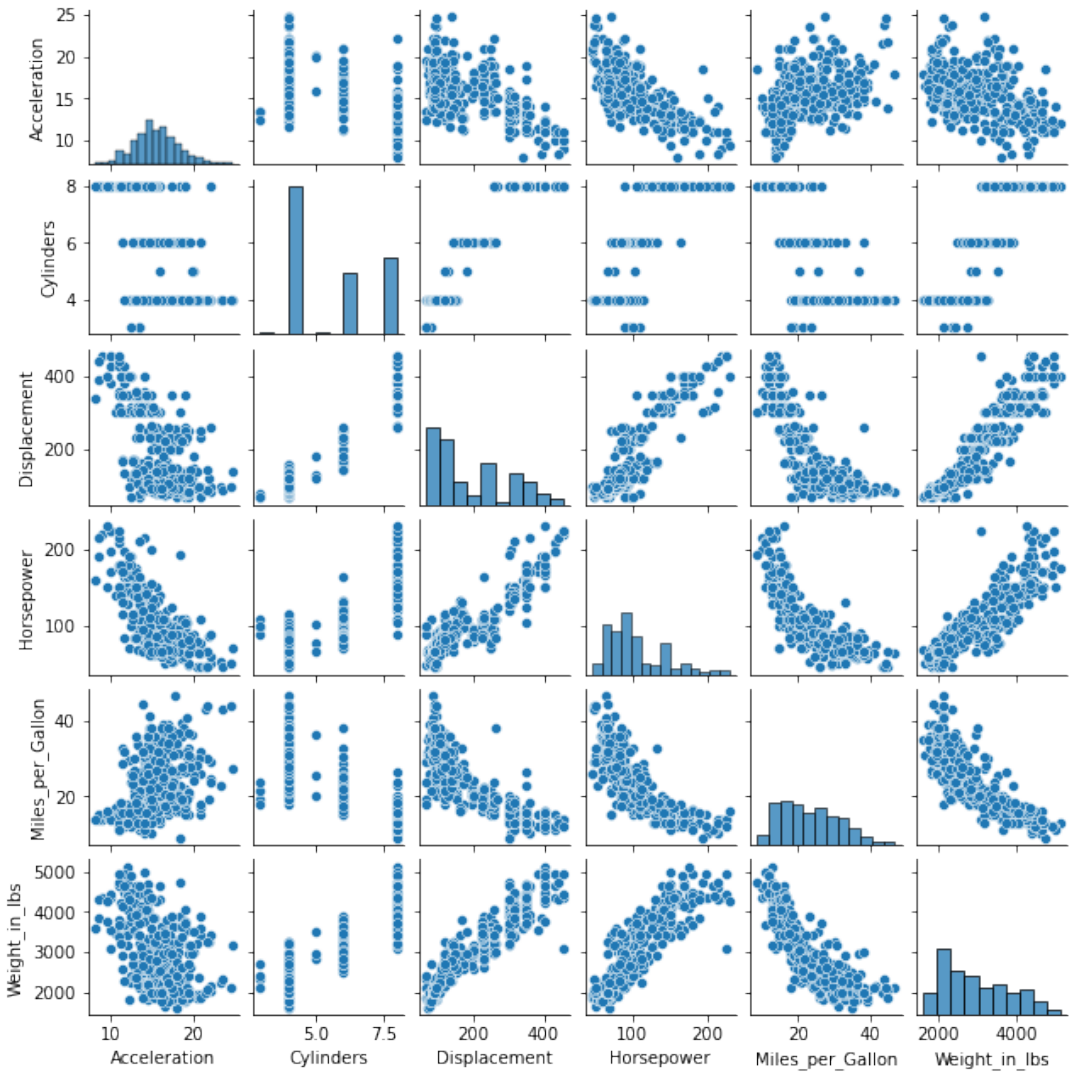
Cette carte montre les corrélations entre chaque variable en leur donnant un poids de -1 à +1. Les bleus signifient une corrélation négative, les rouges signifient une corrélation positive et se rapprocher de 1 ou -1 signifie que vous avez quelque chose de significatif,

1.5.5 Seaborn pairplot

[9]:
```
# compare pairplot and heatmap positive and negative correlations
cars_pairplot = cars.dropna()
# cars_pairplot.to_excel('path/cars.xlsx')
```

```
# seaborn.pairplot(cars_pairplot, height=1.5)
# ; gets rid of extra text that accompanies plot
seaborn.pairplot(cars_pairplot, size=1.5);
```

```
/Users/mallemkhadidja/opt/anaconda3/lib/python3.9/site-
packages/seaborn/axisgrid.py:2076: UserWarning: The `size` parameter has been
renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
```



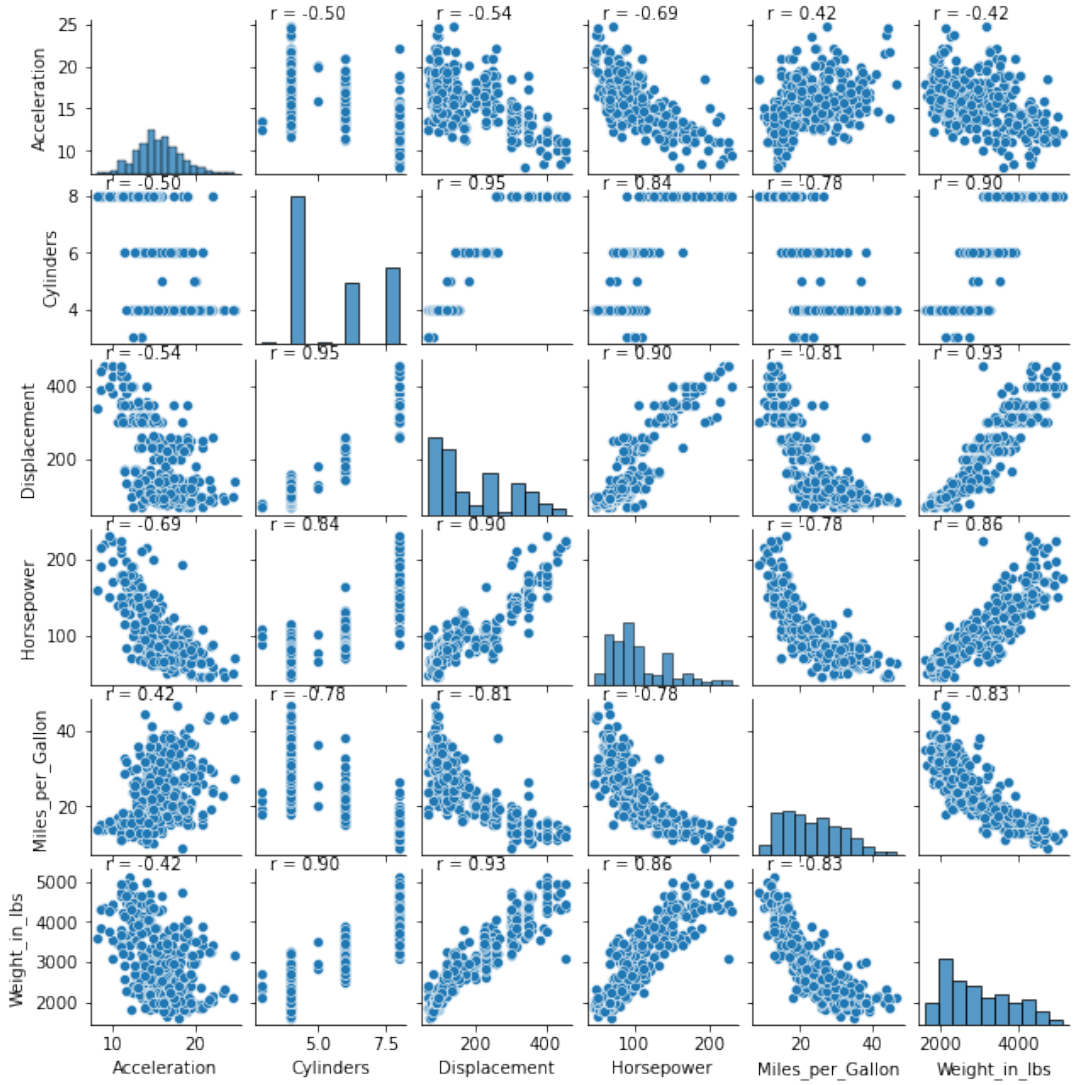
1.5.6 Ajouter le coefficient de corrélation au pairplot

[10]:
```
# add correlation coefficient to plot
from scipy import stats

def corrfunc(x, y, **kws):
    r, _ = stats.pearsonr(x, y)
    ax = plt.gca()
    ax.annotate('r = {:.2f}'.format(r), xy=(0.1, 1.0), xycoords=ax.transAxes)

# pair_plot = seaborn.pairplot(cars_pairplot, height=1.5)
pair_plot = seaborn.pairplot(cars_pairplot, size=1.5);
pair_plot.map_lower(corrfunc);
pair_plot.map_upper(corrfunc);
```

```
/Users/mallemkhadidja/opt/anaconda3/lib/python3.9/site-
packages/seaborn/axisgrid.py:2076: UserWarning: The `size` parameter has been
renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
```


r = -0.50
r = -0.54
r = -0.69
r = 0.42
r = -0.42
r = -0.50
r = 0.95
r = 0.84
r = -0.78
r = 0.90
r = -0.54
r = 0.95
r = 0.90
r = -0.81
r = 0.93
r = -0.69
r = 0.84
r = 0.90
r = -0.78
r = 0.86
r = 0.42
r = -0.78
r = -0.81
r = -0.78
r = -0.83
r = -0.42
r = 0.90
r = 0.93
r = 0.86
r = -0.83
Acceleration
Cylinders
Displacement
Horsepower
Miles_per_Gallon
Weight_in_lbs

CHAPITRE 2

LA RÉGRESSION LINÉAIRE

La régression linéaire est une modélisation linéaire qui permet d'établir des estimations dans le futur à partir d'informations provenant du passé, par exemple on cherche à prédire le cours de la bourse, le prix d'un appartement, ou bien l'évolution de la température sur Terre, donc on cherche en fait à résoudre un problème de régression.
On dispose d'un Dataset (x,y) donc on peut utiliser l'apprentissage supervisé pour développer un modèle de régression. Dans ce chapitre on va montrer comment développer notre premier modèle de Machine Learning ! [5], [3]

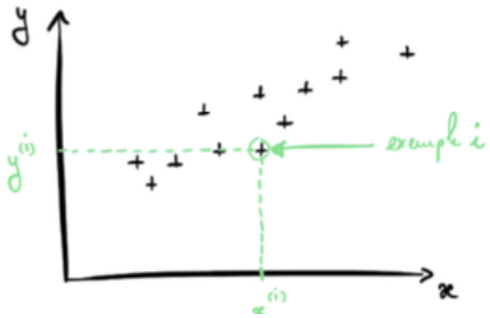
2.1 Récolter les données

Imaginons que plusieurs agences immobilières nous aient fourni des données sur des appartements à vendre, notamment le prix de l'appartement (y) et la surface habitable (x). En Machine Learning, on dit qu'on dispose de m exemples d'appartements.
On désigne :

— $x^{(i)}$ la surface habitable de l'exemple i

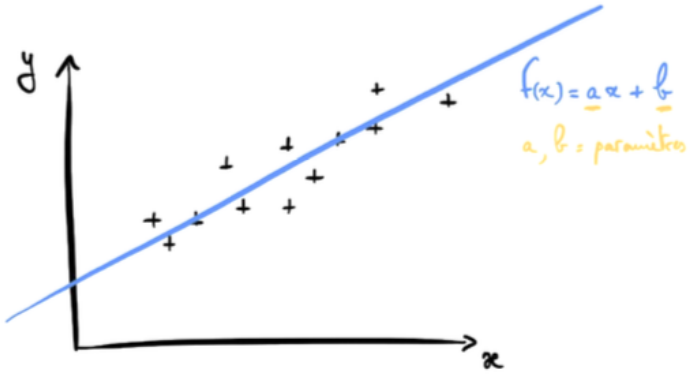
— $y^{(i)}$ le prix de l'exemple i

En visualisant notre Dataset, on obtient le nuage de points suivant :



2.2 Créer un modèle linéaire

A partir de ces données, on développe un modèle linéaire $f(x) = ax + b$ où a et b sont les paramètres du modèle.
Un bon modèle donne de petites erreurs entre ses prédictions $f(x)$ et les exemples (y) du Dataset. Nous ne connaissons pas les valeurs des paramètres a et b, ce sera le rôle de la machine de les trouver, de sorte à tracer un modèle qui s'insère bien dans notre nuage de point comme ci-dessous :



2.3 Définir La Fonction Coût

Pour la régression linéaire, on utilise le **squared error** pour mesurer les erreurs entre $f(x)$ et (y) . Concrètement, voici la formule pour exprimer l'erreur i entre le prix $y^{(i)}$ et la prédiction faites en utilisant la surface $x^{(i)}$

$$erreur^{(i)} = (f(x^{(i)}) - y^{(i)})^2$$

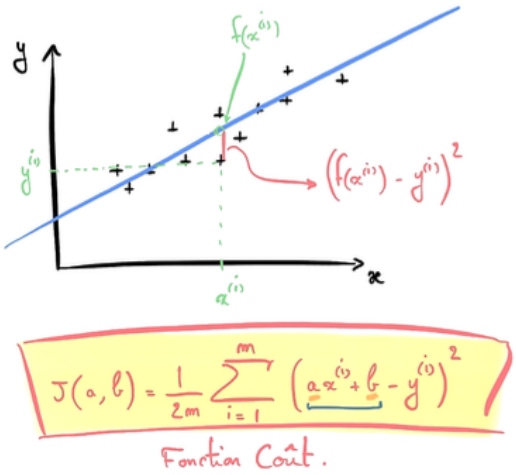
Par exemple, imaginons que le 10 ième exemple de notre Dataset soit un appartement de $x^{10} = 80\ m^2$ dont le prix s'élève à $y^{(10)} = 100,000\ €$ et que notre modèle prédise un prix de $f(x^{(10)}) = 100,002\ €$. L'erreur pour cette exemple est donc :

$$erreur^{(10)} = (f(x^{(10)}) - y^{(10)})^2 = (100,002 - 100,000)^2 = (2)^2 = 4$$

Chaque prédiction s'accompagne d'une erreur, on a donc m erreurs.
On définit la **Fonction Coût** $J(a,b)$ comme étant la moyenne de toutes les erreurs :

$$J(a,b) = \frac{1}{2m}\sum_{i=1}^{m} erreur^{(i)} = \frac{1}{2m}\sum_{i=1}^{m}(f(x^{(i)}) - y^{(i)})^2$$

Note : En français, cette fonction a un nom : c'est l'erreur **quadratique moyenne** (Mean Squared Error)



2.4 Trouver les paramètres qui minimisent la Fonction Coût "Gradient Descent"

La prochaine étape est l'étape la plus excitante, il s'agit de laisser la machine apprendre quels sont les paramètres qui **minimisent** la Fonction Coût, c'est-à-dire les paramètres qui nous donnent le meilleur modèle. Pour trouver le minimum, on utilise un algorithme d'optimisation qui s'appelle **Gradient Descent** (la descente de gradient).

Comprendre le Gradient Descent (la descente de gradient)

Imaginons nous, perdu en montagne. notre but est de rejoindre le refuge qui se trouve au point

le plus bas de la vallée. nous n'avons pas pris de carte avec nous donc nous ne connaissons pas les coordonnées de ce refuge, nous devons le trouver tout seul.
Pour nous en sortir, voici une stratégie à adopter :

1. Depuis notre position actuelle, nous partons en direction de là où la pente descend la plus forte.

2. nous avançons une certaine distance en suivant cette direction coûte que coûte (même si ça implique de remonter une pente)

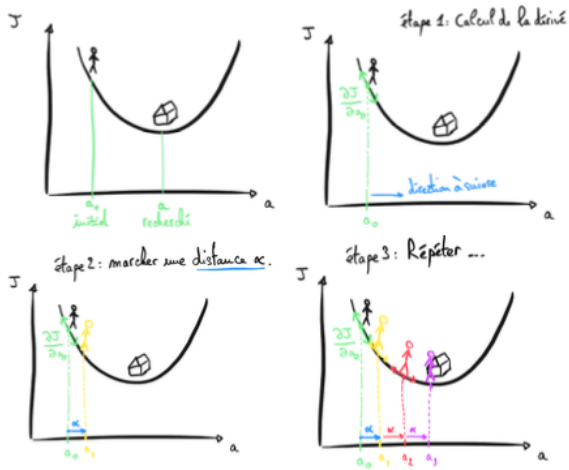
3. Une fois cette distance parcourue, nous répétons les 2 premières opérations en boucle, jusqu'à atteindre le point le plus bas de la vallée.

Les étapes 1, 2 et 3 forment ce qu'on appelle l'algorithme de **Gradient Descent**.
Cet algorithme vous permet de trouver le **minimum** de la Fonction Coût $J(a, b)$ (le point le plus bas de la montagne) en partant de coordonnées a et b **aléatoires** (votre position initiale dans la montagne) :

1. Calculer la **pente** de la Fonction Coût, c'est-à-dire la **dérivée** de $J(a, b)$.

2. **Evoluer** d'une certaine **distance** $\propto$ dans la direction de la pente la plus forte. Cela a pour résultat de modifier les paramètres a et b
   3. Recommencer les étapes 1 et 2 jusqu'à atteindre le minimum de $J(a, b)$.

Pour illustrer l'algorithme, voir le dessin ci-dessous, où on montre la recherche du paramètre idéal (la même chose s'applique au paramètre b)



Comment utiliser l'algorithme de Gradient Descent

Pour rappel, nous avons jusqu'à présent créé un Dataset, développé un modèle aux paramètres inconnus, et exprimé la **Fonction Coût** $J(a,b)$ associée à ce modèle.

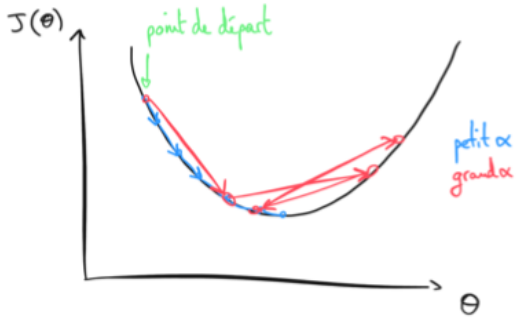
Notre objectif final : Trouver les paramètres a et b qui minimisent $J(a,b)$. Pour cela, nous allons choisir a et b au **hasard** (nous allons nous perdre en montagne) puis allons utiliser en **boucle** la descente de gradient pour mettre à jour nos paramètres dans la direction de la Fonction Coût **la plus faible**.

Répeter en boucle :

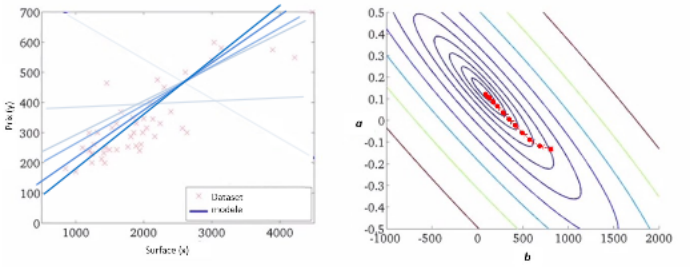
$$a = a - \propto \frac{\partial J(a,b)}{\partial a}$$
$$b = b - \propto \frac{\partial J(a,b)}{\partial b}$$

À chaque itération de cette boucle, les paramàtres a et b sont mis à jour en **soustrayant** leur propre valeur à la valeur de la **pente** $\frac{\partial J(a,b)}{\partial ...}$ multipliée par la distance à parcourir $\propto$. On appelle $\propto$ la vitesse d'apprentissage (**Learning rate)**.

Si la vitesse est trop lente, le modèle peut mettre longtemps à être entraîné, mais si la vitesse est trop grande, alors la distance parcourue est trop longue et le modèle peut ne jamais converger. Il est important de trouver un juste milieu. Voir le dessin ci-dessous .



Une fois cet algorithme programmé, on va voir notre première intelligence artificielle apprendre à prédire le prix d'un appartement selon sa surface habitable. on voit comme ci-dessous que l'algorithme arrive à minimiser la Fonction Coût avec le nombre d'itérations.



A partir de là, c'est la porte ouverte aux algorithmes qui automatisent les transactions immobiliàres, et le même concept que celui que vous venez d'apprendre sera appliqué pour apprendre à une machine comment reconnaitre un visage sur une photo, comment prédire le cours de la bourse, etc. Mais avant de voir la magie s'opérer, il faut avoir préalablement calculer les **dérivées partielles** de la Fonction Coût.

Calcul des dérivées partielles

Pour implémenter l'algorithme de Gradient Descent, il faut donc calculer les **dérivées partielles** de la Fonction Coût. On rappelle que , la dérivée d'une fonction en un point nous donne la valeur de sa pente en ce point.

Fonction Coût :

$$J(a,b) = \frac{1}{2m}\sum_{i=1}^{m}(ax^{(i)} + b - y^{(i)})^2$$

Dérivée selon le paramètre a :

$$\frac{\partial J(a,b)}{\partial a} = \frac{1}{m}\sum_{i=1}^{m}(ax^{(i)} + b - y^{(i)}) \times x^{(i)}$$

Dérivée selon le paramètre b :

$$\frac{\partial J(a,b)}{\partial b} = \frac{1}{m}\sum_{i=1}^{m}(ax^{(i)} + b - y^{(i)})$$

Note :
la dérivée d'une fonction composée :

$$(g \circ f)' = f' \times g' \circ f$$

Dans notre cas : $f = ax + b - y$ et $g = (f)^2$
En dérivant, le carré tombe et se simplifie avec la fraction $\frac{1}{2m}$ pour devenir $\frac{1}{m}$ et $x^{(i)}$ apparait en facteur pour la dérivée par rapport à a.

2.5 La régression linéaire à plusieurs variables-utilisation des matrices et des vecteurs

Algèbre linéaire [16] et [6].
Dans la pratique, on exprime notre Dataset et nos paramètres sous forme **matricielle**, ce qui simplifie beaucoup les calculs. On créer ainsi un **vecteur** $\theta = \begin{pmatrix} a \\ b \end{pmatrix} \in \mathbb{R}^{n+1}$ qui contient tous les paramètres pour notre modèle, un **vecteur** $y \in \mathbb{R}^m$ et une **matrice** $X \in \mathbb{R}^{m\times(n+1)}$ qui inclut toutes les features n. Dans la régression linéaire à une seule variable, $n = 1$.

2.6 Résumé des étapes pour développer un programme de Régression Linéaire

1. Récolter des données (X, y) avec $X \in \mathbb{R}^{m\times(n+1)}$, $y \in \mathbb{R}^m$
2. Donner à la machine un modèle linéaire $F(X) = X.\theta$ où $\theta = \begin{pmatrix} a \\ b \end{pmatrix} \in \mathbb{R}^{n+1}$
3. Créer la Fonction Coût $J(\theta) = \frac{1}{2m}\sum(F(X) - y)^2$
4. Calculer le gradient et utiliser l'algorithme de **Gradient Descent**.

Répeter en boucle :

$$\theta = \theta - a \times \frac{\partial J(\theta)}{\partial \theta}$$

Gradient $\frac{\partial J(\theta)}{\partial \theta} = \frac{1}{m} X^T.(F(X) - Y)$

Le learning rate a prend le nom **d'hyper-paramètre** de par son influence sur la performance finale du modèle (s'il est trop grand où trop petit, la fonction le Gradient Descent ne converge pas).

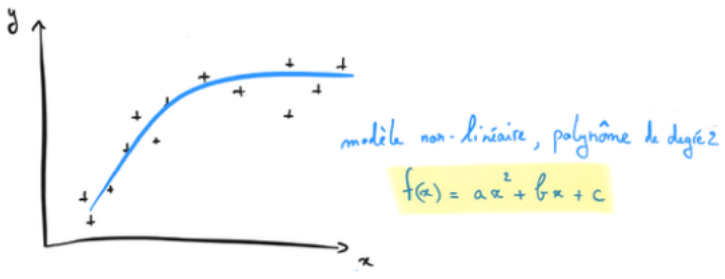
2.7 Régression Polynômiale à plusieurs variables

Si on achète un stylo à 10 (DA), combien on coûteront 100 stylos ? 1000 (DA) ? Faux !
Nous vivons dans un monde réagit par des lois souvent non-linéaires et une infinité de facteurs peuvent influencer nos résultats.
Par exemple, si on achète 100 stylos, on aura peut-être une réduction à 900 (DA). Si en revanche il y a une pénurie de stylos, ce même stylo qui coûtait 10(DA) pourrait valoir 15 (DA).
C'est là qu'Excel ne pourra plus rien pour nous et que le Machine Learning trouve son utilité dans le monde réel.

Problème non-linéaire : Un problème plus compliqué ?

Pour le nuage de point ci-dessous, il semblerait judicieux de développer un modèle polynômial de degré 2.

$$f(x) = ax^2 + bx + c$$



Nous pouvons améliorer nos features et la forme de notre fonction de prédiction $f(x)$ pour avoir un bon modèle c'est à dire avoir des petites erreurs. La fonction de prédiction $f(x)$ a besoin d'être non linéaire si elle ne correspond pas bien aux données en anglais on dit : it does not fit the data well.

CHAPITRE 3

LA RÉGRESSION LOGISTIQUE/ CLASSIFICATION

Dans l'apprentissage supervisé, il y a deux type de problèmes :

— Les régressions linéaire

— Les régressions logistique (classifications)

Dans ce chapitre, on va découvrir le modèle de Régression Logistique, qui permet de résoudre des problèmes de classification binaires.

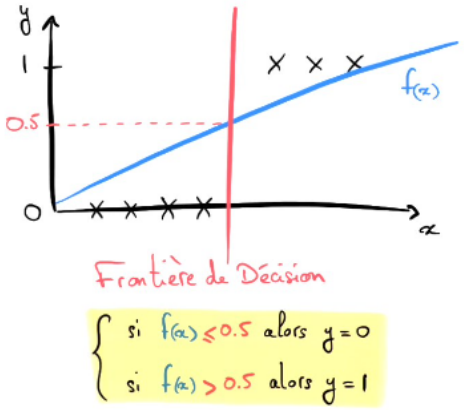
3.1 Les problèmes de Classification

Jusqu'à présent, nous avons appris comment résoudre des problèmes de régression. Au cours de l'introduction, on a parlé des problèmes de classification, qui consistent par exemple à classer un email en tant que 'spam' ou 'non spam'.
Dans ce genre de problème, on aura un Dataset contenant une variable target y pouvant prendre 2 valeurs seulement, par exemple 0 ou 1

— si $y = 0$, alors l'email n'est pas un spam

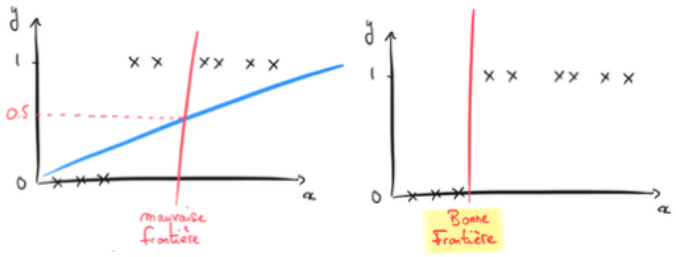
— si $y = 1$, alors l'email est un spam

On dit également que l'on a **2 classes**, c'est une classification **binaire**
Pour ces problèmes, on ajoute au modèle une frontière de décision qui permet de classer un email dans la classe 0 ou la classe 1.

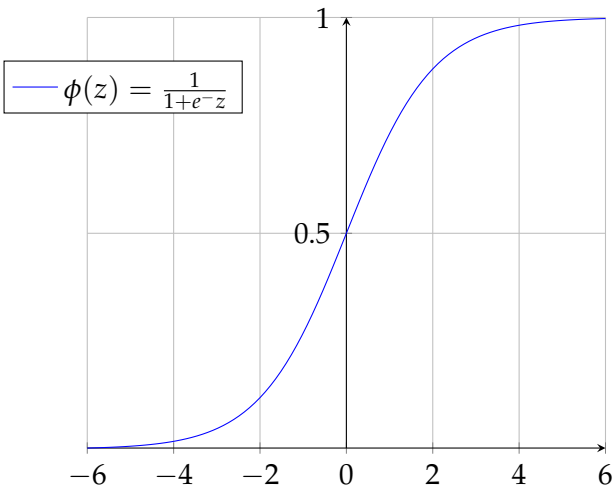


3.2 Le modèle de Régression logistique

Pour les problèmes de classification binaire, un modèle linéaire $F = X.\theta$, comme on l'a tracé sur la figure précédente, ne convient pas. on voit plutôt le résultat que l'on obtient avec un tel modèle pour le Dataset suivant :



On développe alors une nouvelle fonction pour les problèmes de classification binaire, c'est la fonction logistique (aussi appelé fonction sigmoïde ou tout simplement sigma σ). Cette fonction a la particularité d'être toujours comprise en 0 et 1.

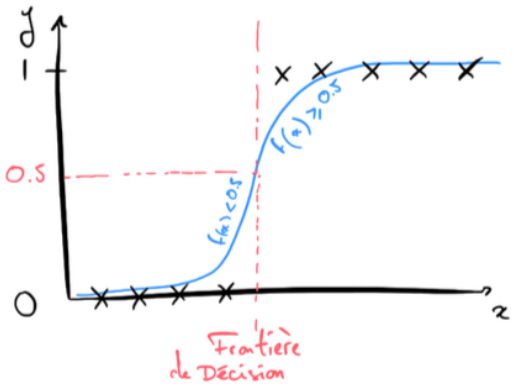


Pour coller la fonction logistique sur un Dataset X, y on y fait passer le produit matriciel $X.\theta$ ce qui nous donne le modèle de Logistic Regression :

$$\sigma(X.\theta) = \frac{1}{1 + e^{-X.\theta}}$$

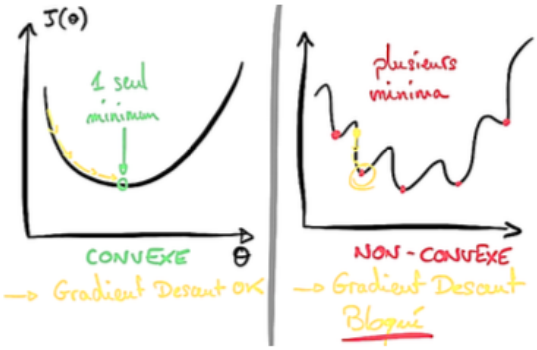
A partir de cette fonction, il est possible de définir une frontière de décision. Typiquement, on définit un seuil à 0.5 comme ceci :

$$y = 0 \quad \text{si} \quad \sigma(X.\theta) < 0.5$$
$$y = 1 \quad \text{si} \quad \sigma(X.\theta) \geq 0.5$$

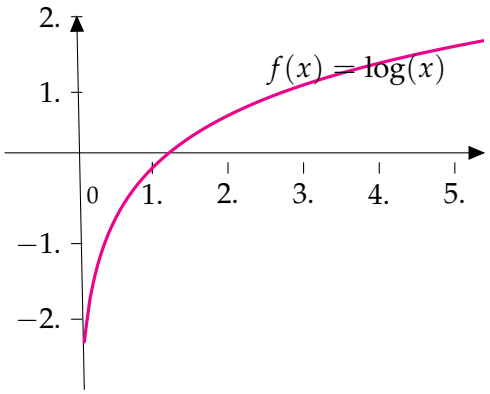


3.3 Fonction Coût associée à la Régression Logistique

Pour la régression linéaire, la Fonction Coût $J(\theta) = \frac{1}{2m}\sum(X.\theta - Y)^2$ donnait une courbe **convexe** (qui présente un unique minima). C'est ce qui fait que l'algorithme de Gradient Descent fonctionne.
En revanche, utiliser cette fonction pour le modèle Logistique ne donnera pas de courbe convexe (dû à la non-linéarité) et l'algorithme de Gradient Descent se **bloquera** au premier minima rencontré, sans trouver le minimum **global**.



Il faut donc développer une nouvelle Fonction Coût spécialement pour la régression logistique. On utilise alors la fonction **logarithme** pour transformer la fonction sigma en fonction convexe en séparant les cas où $y = 1$ des cas où $y = 0$

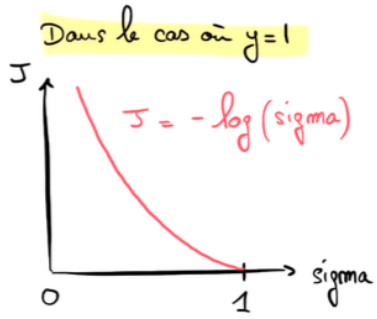


3.3.1 Fonction Coût dans les cas où $y = 1$

Voici la Fonction Coût que l'on utilise dans les cas où $y = 1$:

$$J(\theta) = -\log(\sigma(X.\theta))$$

Explications :
Si notre modèle prédit $\sigma(x) = 0$ alors que $y = 1$, on doit pénaliser la machine par une grande erreur (un grand coût). La fonction logarithme permet de tracer cette courbe avec une propriété convexe, ce qui poussera le Gradient Descent à trouver les paramètres θ pour un coût qui tend vers 0.

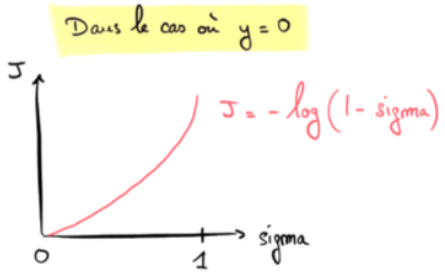


3.3.2 Fonction Coût dans les cas où $y = 0$

Cette fois la Fonction Coût devient :

$$J(\theta) = -\log(1 - \sigma(X.\theta))$$

Explications :
Si notre modèle prédit $\sigma(x) = 1$ alors que $y = 0$, on doit pénaliser la machine par une grande erreur (un grand coût). Cette fois $-log(1 - 0)$ donne la même courbe, inversée sur l'axe vertical.



3.3.3 Fonction Coût complète

Pour écrire la Fonction Coût en une seule équation, on utilise l'astuce de séparer les cas $y = 0$ et $y = 1$ avec une annulation :

$$J(\theta) = \frac{-1}{m} \sum y \times \log(\sigma(X.\theta)) + (1-y) \times \log(1 - \sigma(X.\theta))$$

Dans le cas où $y = 0$, il nous reste :

$$J(\theta) = \frac{-1}{m} \sum \cancel{0 \times \log(\sigma(X.\theta))} + 1 \times \log(1 - \sigma(X.\theta))$$

Dans le cas où $y = 1$:

$$J(\theta) = \frac{-1}{m} \sum 1 \times \log(\sigma(X.\theta)) + \cancel{0 \times \log(1 - \sigma(X.\theta))}$$

3.4 Gradient Descent pour la Régression Logistique

L'algorithme de Gradient Descent s'applique exactement de la même manière que pour la régression linéaire. En plus, la dérivée de la Fonction Coût est la même aussi ! On a :

Gradient : $\frac{\partial J(\theta)}{\partial \theta} = \frac{1}{m} \sum (\sigma(X.\theta) - y).X$

Gradient Descent : $\theta = \theta - \alpha \times \frac{\partial J(\theta)}{\partial \theta}$

3.5 Résumé de la Régression Logistique

Modèle	$\sigma(X.\theta) = \frac{1}{1+e^{-X.\theta}}$
Fonction Coût	$\frac{-1}{m} \sum y \times \log(\sigma(X.\theta)) + (1-y) \times \log(1 - \sigma(X.\theta))$
Gradient	$\frac{\partial J(\theta)}{\partial \theta} = \frac{1}{m} X^T.(\sigma(X.\theta) - y)$
Gradient Descent	$\theta = \theta - \alpha \times \frac{\partial J(\theta)}{\partial \theta}$

CHAPITRE 4

PROGRAMMATION

4.1 L'algorithme de Nearest Neighbour

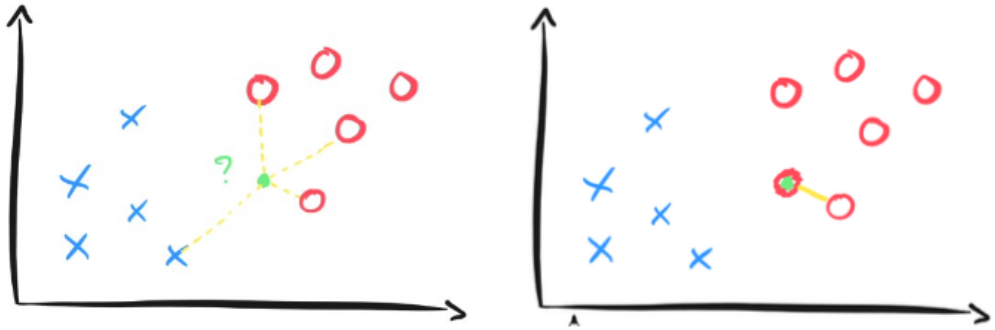
L'année dernière nous avons abordé la structure générale pour développer des modèle de régression Linéaire et Polynômiale avec SKlearn [9] . Cette année nous nous intéressons à l'algorithme de Nearest Neighbour (le voisin le plus proche) qui permet de résoudre des problèmes de classification à plusieurs classes de façon simple et très efficace. et à la fin de chapitre nous avons réalisé le projet de prédiction de survie du Titanic en utilisant cet algorithme.
Voici un exemple pour bien comprendre l'idée de L'algorithme de Nearest Neighbour.
Si Ikram part se promener en montagne avec Khawla. Avant de partir, il fait 30 °C et khawla dit à Ikram qu'elle a chaud. Arrivé en montagne, il fait désormais 10 °C et Khawla dit qu'elle a froid. En redescendant la vallée, il fait maintenant 15 °C, Ikram, pense-tu que Khawla aura froid ou bien chaud ? 15 °C étant plus proche de 10 °C (froid) que de 30 °C (chaud), il semble légitime de prédire que Khawla aura froid.
Voilà l'essentiel de ce qu'il y à savoir sur l'algorithme Nearest Neighbour. Quand on doit faire une nouvelle prédiction, on doit trouver dans le Dataset l'exemple le plus proche par rapport aux conditions dans lesquelles on est.

4.2 K-Nearest Neighbour (K-NN)

La distance la plus courte

En regardant le nuage de points qui suit. L'exemple le plus proche du point vert est un exemple de la classe rouge. L'algorithme de Nearest Neighbour calcule ainsi la distance entre le point vert et les autres points du Dataset et associe le point vert à la classe dont l'exemple est le plus

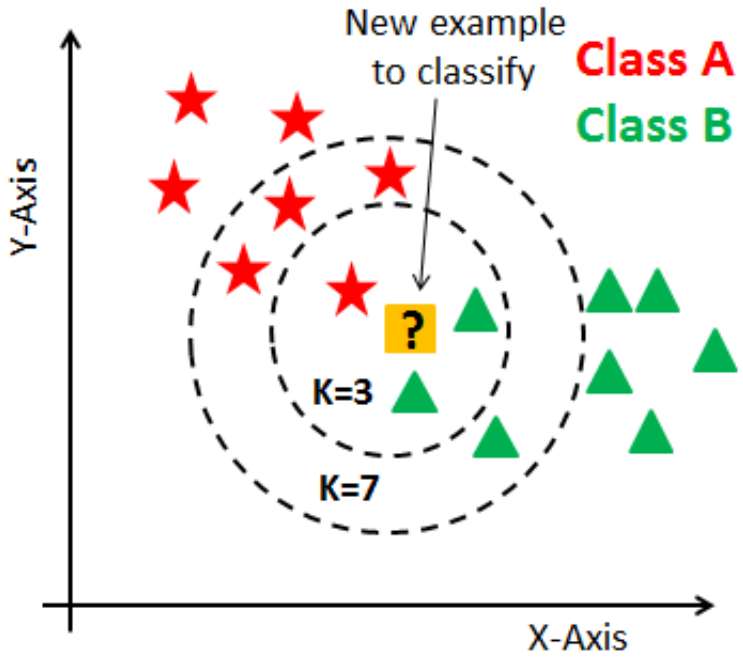
proche en terme de distance.



Typiquement, on utilise la distance euclidienne (c'est la droite direct entre deux points) mais d'autres métriques sont parfois plus utiles, comme la distance de Manhattan ou bien la distance cosinus.

Le nombre de voisin K

Pour prédire la classe d'une nouvelle donnée d'entrée, il va chercher ses K voisins les plus proches (en utilisant la distance euclidienne, ou autres) et choisira la classe des voisins majoritaires.



Pour appliquer cette méthode, les étapes à suivre sont les suivantes :

— on fixe le nombre de voisins k

— on détecte les k-voisins les plus proches des nouvelles données d'entrée que l'on veut classer

— on attribue les classes correspondantes par vote majoritaire

Mais, comment choisit-on ce paramètre k lors de l'implémentation de l'algorithme ?

— on fait varier k

— pour chaque valeur de k, on calcule le taux d'erreur de l'ensemble de test

— on garde le paramètre k qui minimise ce taux d'erreur test.

4.3 Rappel des étapes essentiels d'implémentation avec SKlearn

— model=KNeighborsClassifier() :pour sélectionner un estimateur et préciser ses hyperparamètres, cette année nous nous intéressons à la régression logistique et l'algorithme de K-Nearest Neighbour

— model.fit(x, y) : pour entraîner notre modèle

— model.score(x, y) : pour évaluer notre modèle

— model.predict(x) :pour générer des prédictions.

4.4 Projet 3 : Prédiction de survie du Titanic (régression logistique)

Dans ce projet, un dataset très connu dans la communauté des Data Scientists sera utilisé. Il s'agit d'un sous-ensemble des passagers du fameux navire Titanic.

L'objectif de ce projet est de construire un modèle de logistique régression qui sait prédire pour un passager particulier s'il a survécu ou pas à ce drame. Ce dataset indique pour chaque passager les informations suivantes :

pclass : variable indiquant la classe de la cabine. Elle prend ses valeurs parmi les valeurs 1, 2 et 3 qui correspondent respectivement à la première, seconde et troisième classe

survived : variable binaire indiquant si le passager a survécu au drame ou non.

name (resp. **sex** , **age**) : le nom (resp. le sex , l'âge) d'un passager

sibsp : variable indiquant si le passager a des frères, des sœurs, un époux ou une épouse à bord du bateau

parch : variable indiquant si le passager a des parents ou des enfants à bord du bateau.

ticket : numéro de ticket

fare : prix du ticket

cabin : cabine

embarked : port d'embarcation (C = Cherbourg, Q = Queenstown, S = Southampton).

boat : canot de sauvetage

body : numéro d'identification du corps

home.dest : Domicile/Destination.

En résumé, l'objectif est de prédire la variable survived à partir des valeurs des variables pclass, sex, age. Et voilà le projet réalisé sur Notebook Jupiter.

PRÉDICTION DE SURVIE DU TITANIC

May 21, 2022

1 Importer les bibliothèques et le Dataset.

[1]:
```
import numpy as np                # permet d'effectuer des calculs numériques.
import matplotlib.pyplot as plt   # Permet d'affichier des graphiques.
import pandas as pd               # permet de manipuler facilement des données.
import seaborn as sns             # permet de créer des graphiques statistiques.
```

2 Traitement de données (Data processing) .

- Lire le fichier titanic3 dans un DataFrame pandas.

[2]:
```
titanic=pd.read_excel('titanic3 (3).xls')
titanic
```

[2]:

	pclass	survived	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked	boat
0	1	1	Allen, Miss. Elisabeth Walton	female	29.0000	0	0	24160	211.3375	B5	S	2
1	1	1	Allison, Master. Hudson Trevor	male	0.9167	1	2	113781	151.5500	C22 C26	S	11
2	1	0	Allison, Miss. Helen Loraine	female	2.0000	1	2	113781	151.5500	C22 C26	S	NaN
3	1	0	Allison, Mr. Hudson Joshua Creighton	male	30.0000	1	2	113781	151.5500	C22 C26	S	NaN
4	1	0	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	female	25.0000	1	2	113781	151.5500	C22 C26	S	NaN
…	…	…	…	…	…	…	…	…	…	…	…	…
1304	3	0	Zabour, Miss. Hileni	female	14.5000	1	0	2665	14.4542	NaN	C	NaN
1305	3	0	Zabour, Miss. Thamine	female	NaN	1	0	2665	14.4542	NaN	C	NaN
1306	3	0	Zakarian, Mr. Mapriededer	male	26.5000	0	0	2656	7.2250	NaN	C	NaN
1307	3	0	Zakarian, Mr. Ortin	male	27.0000	0	0	2670	7.2250	NaN	C	NaN
1308	3	0	Zimmerman, Mr. Leo	male	29.0000	0	0	315082	7.8750	NaN	S	NaN

```
1307    male  27.0000      0      0    2670    7.2250      NaN        C  NaN
1308    male  29.0000      0      0  315082    7.8750      NaN        S  NaN

       body                        home.dest
0       NaN                     St Louis, MO
1       NaN  Montreal, PQ / Chesterville, ON
2       NaN  Montreal, PQ / Chesterville, ON
3     135.0  Montreal, PQ / Chesterville, ON
4       NaN  Montreal, PQ / Chesterville, ON
…     …                                 …
1304  328.0                             NaN
1305    NaN                             NaN
1306  304.0                             NaN
1307    NaN                             NaN
1308    NaN                             NaN

[1309 rows x 14 columns]
```

- Afficher la dimension du DataFrame.

[3]:
```
titanic.shape
```

[3]:
```
(1309, 14)
```

- Afficher les 5 premières lignes du DataFrame.

[4]:
```
titanic.head()
```

[4]:
```
   pclass  survived                                             name     sex  \
0       1         1                    Allen, Miss. Elisabeth Walton  female
1       1         1                   Allison, Master. Hudson Trevor    male
2       1         0                     Allison, Miss. Helen Loraine  female
3       1         0             Allison, Mr. Hudson Joshua Creighton    male
4       1         0  Allison, Mrs. Hudson J C (Bessie Waldo Daniels)  female

       age  sibsp  parch  ticket      fare    cabin embarked boat   body  \
0  29.0000      0      0   24160  211.3375       B5        S    2    NaN
1   0.9167      1      2  113781  151.5500  C22 C26        S   11    NaN
2   2.0000      1      2  113781  151.5500  C22 C26        S  NaN    NaN
3  30.0000      1      2  113781  151.5500  C22 C26        S  NaN  135.0
4  25.0000      1      2  113781  151.5500  C22 C26        S  NaN    NaN

                        home.dest
0                     St Louis, MO
1  Montreal, PQ / Chesterville, ON
2  Montreal, PQ / Chesterville, ON
3  Montreal, PQ / Chesterville, ON
4  Montreal, PQ / Chesterville, ON
```

- Supprimer des colonnes.

[5]:
```
titanic=titanic.
↪drop(['name','sibsp','parch','ticket','fare','cabin','embarked','boat','body','home.
↪dest'], axis=1)
```

- Afficher du nouveau le DataFrame.

[6]:
```
titanic.head()
```

[6]:

	pclass	survived	sex	age
0	1	1	female	29.0000
1	1	1	male	0.9167
2	1	0	female	2.0000
3	1	0	male	30.0000
4	1	0	female	25.0000

- Afficher certains détails statistiques de base (statistiques rapide).

[7]:
```
titanic.describe()
```

[7]:

	pclass	survived	age
count	1309.000000	1309.000000	1046.000000
mean	2.294882	0.381971	29.881135
std	0.837836	0.486055	14.413500
min	1.000000	0.000000	0.166700
25%	2.000000	0.000000	21.000000
50%	3.000000	0.000000	28.000000
75%	3.000000	1.000000	39.000000
max	3.000000	1.000000	80.000000

- Donner le nombre de personnes qui étaient en troisième ,première et deuxième classe.

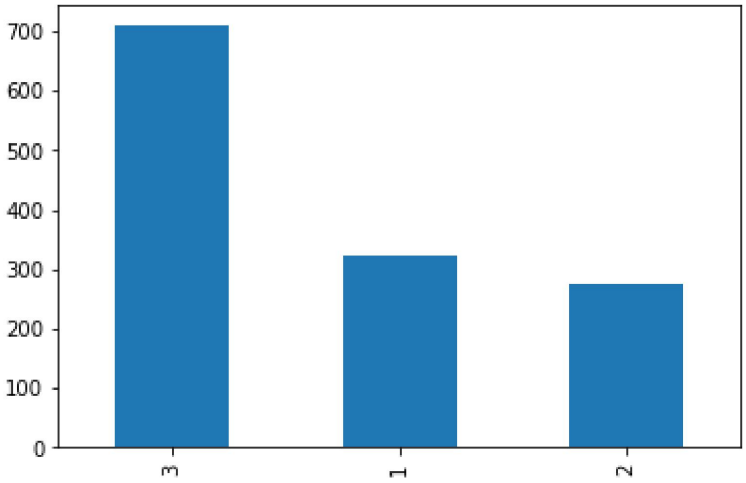
[8]:
```
titanic['pclass'].value_counts()
```

[8]:
```
3    709
1    323
2    277
Name: pclass, dtype: int64
```

- Afficher ces derniers résultats dans un diagramme à barres.

[9]:
```
titanic['pclass'].value_counts().plot.bar()
```

[9]:
```
<AxesSubplot:>
```



- Regrouper les passagers selon le sex et la classe de la cabine en utilisant la moyenne.

[10]:
```
titanic.groupby(['sex','pclass']).mean()
```

[10]:
```
               survived        age
sex    pclass
female 1       0.965278  37.037594
       2       0.886792  27.499191
       3       0.490741  22.185307
male   1       0.340782  41.029250
       2       0.146199  30.815401
       3       0.152130  25.962273
```

3 Classification survivants du Titanic .

- Filtrer les colonnes de dataset en sélectionnant les colonnes qui nous intéressent.

[11]:
```
titanic=titanic[['survived', 'pclass', 'sex', 'age']]
```

- Supprimer les valeurs manquantes de dataset.

[12]:
```
titanic.dropna(axis=0, inplace=True)
```

- Remplacer les mots"male"et"female"par des valeurs numérique à savoir 0 et 1.

[13]:
```
titanic['sex'].replace(['male', 'female'], [0, 1], inplace=True)
titanic.head()
```

[13]:

	survived	pclass	sex	age
0	1	1	1	29.0000
1	1	1	0	0.9167
2	0	1	1	2.0000
3	0	1	0	30.0000
4	0	1	1	25.0000

- Sélectionner l'estimateur K-Nearest Neighbour.

[14]:
```
from sklearn.neighbors import KNeighborsClassifier
```

- Initialiser le modele.

[15]:
```
model = KNeighborsClassifier()
```

- Prédire la variable survived (y) à partir de valeur des variable x=(pclass, sex, age).

[16]:
```
y = titanic['survived']
x = titanic.drop('survived', axis=1)
```

- Afficher y et Afficher x.

[17]:
```
y
```

[17]:
```
0       1
1       1
2       0
3       0
4       0
       ..
1301    0
1304    0
1306    0
1307    0
1308    0
Name: survived, Length: 1046, dtype: int64
```

[18]:
```
x
```

[18]:

	pclass	sex	age
0	1	1	29.0000
1	1	0	0.9167
2	1	1	2.0000
3	1	0	30.0000
4	1	1	25.0000

```
…       …  …       …
1301        3    0  45.5000
1304        3    1  14.5000
1306        3    0  26.5000
1307        3    0  27.0000
1308        3    0  29.0000

[1046 rows x 3 columns]
```

- Entraîner notre modèle.

[19]:
```
model.fit(x, y)
```

[19]:
```
KNeighborsClassifier()
```

- Evaluer notre modèle.

[20]:
```
model.score(x, y)
```

[20]:
```
0.8279158699808795
```

4 Test de survie du Titanic.

- Prédire si un passager a survécu ou non,0 pour ne pas avoir survécu, 1 pour survivre .
  - On va utiliser la fonction “survie”.

Test 01(ikram):

[21]:
```
def survie(model, pclass=2, sex=1, age=24):
    x=np.array([pclass, sex, age]).reshape(1, 3)
    print(model.predict(x))
```

[22]:
```
survie(model)
```

```
[1]
```

Test 02(khawla):

[23]:
```
def survie(model, pclass=2, sex=1, age=26):
    x=np.array([pclass, sex, age]).reshape(1, 3)
    print(model.predict(x))
```

[24]:
```
survie(model)
```

```
[1]
```

D’après le résultat Ikram et Khawla ont survécu du naufrage du Titanic.

Et Vous ?

Bibliographie

[1] S. V. ALEX SMOLA, *Introduction to Machine Learning*, Cambridge University Press (2008).

[2] D. BARBER, *Bayesian Reasoning and Machine Learning*, Cambridge University Press, 2011.

[3] C. M. BISHOP, *Pattern Recognition and Machine Learning*, Springer, 2006.

[4] L. BREIMAN, *Statistical Modeling : The Two Cultures (with comments and a rejoinder by the author)*, Statistical Science, 16 (2001), pp. 199 – 231.

[5] T. HASTIE, R. TIBSHIRANI, AND J. FRIEDMAN, *The Elements of Statistical Learning*, Springer Series in Statistics, Springer New York Inc., New York, NY, USA, 2001.

[6] J. HEFFERON, *Linear Algebra*, SBN 978-1-944325-11-4, OCLC 1178900366, OL 30872051M, 2020.

[7] A. NG, *Machine Learning Yearning*, Online Draft, 2017.

[8] N. J. NILSSON, *Introduction to machine learning : An early draft of a proposed textbook. pages 175-188. http ://robotics.stanford.edu/people/nilsson/mlbook.ht ml*, 1996.

[9] F. PEDREGOSA, G. VAROQUAUX, A. GRAMFORT, V. MICHEL, B. THIRION, O. GRISEL, M. BLONDEL, P. PRETTENHOFER, R. WEISS, V. DUBOURG, J. VANDERPLAS, A. PASSOS, D. COURNAPEAU, M. BRUCHER, M. PERROT, AND E. DUCHESNAY, *Scikit-learn : Machine learning in Python*, Journal of Machine Learning Research, 12 (2011), pp. 2825–2830.

[10] G. SAINT-CIRGUE, *Machine Learning*, machinelearnia, 2019.

[11] A. L. SAMUEL, *Some studies in machine learning using the game of checkers*, IBM J. Res. Dev., 3 (1959), pp. 210–229.

[12] S. SHALEV-SHWARTZ AND S. BEN-DAVID, *Understanding Machine Learning - From Theory to Algorithms.*, Cambridge University Press, 2014.

[13] O. SIMEONE, *A brief introduction to machine learning for engineers*, ArXiv, abs/1709.02840 (2018).

[14] E. R. TUFTE, *The Visual Display of Quantitative Information*, Graphics Press, USA, 1986.

[15] A. M. TURING, *I.—COMPUTING MACHINERY AND INTELLIGENCE*, Mind, LIX (1950), pp. 433–460.

[16] B. WISE AND N. GALLAGHER, *An introduction to linear algebra*, Critical Reviews in Analytical Chemistry, 28 (1998), pp. 1–19.