DESU Data Science pour professionnels
Remise à Niveau en Statistiques

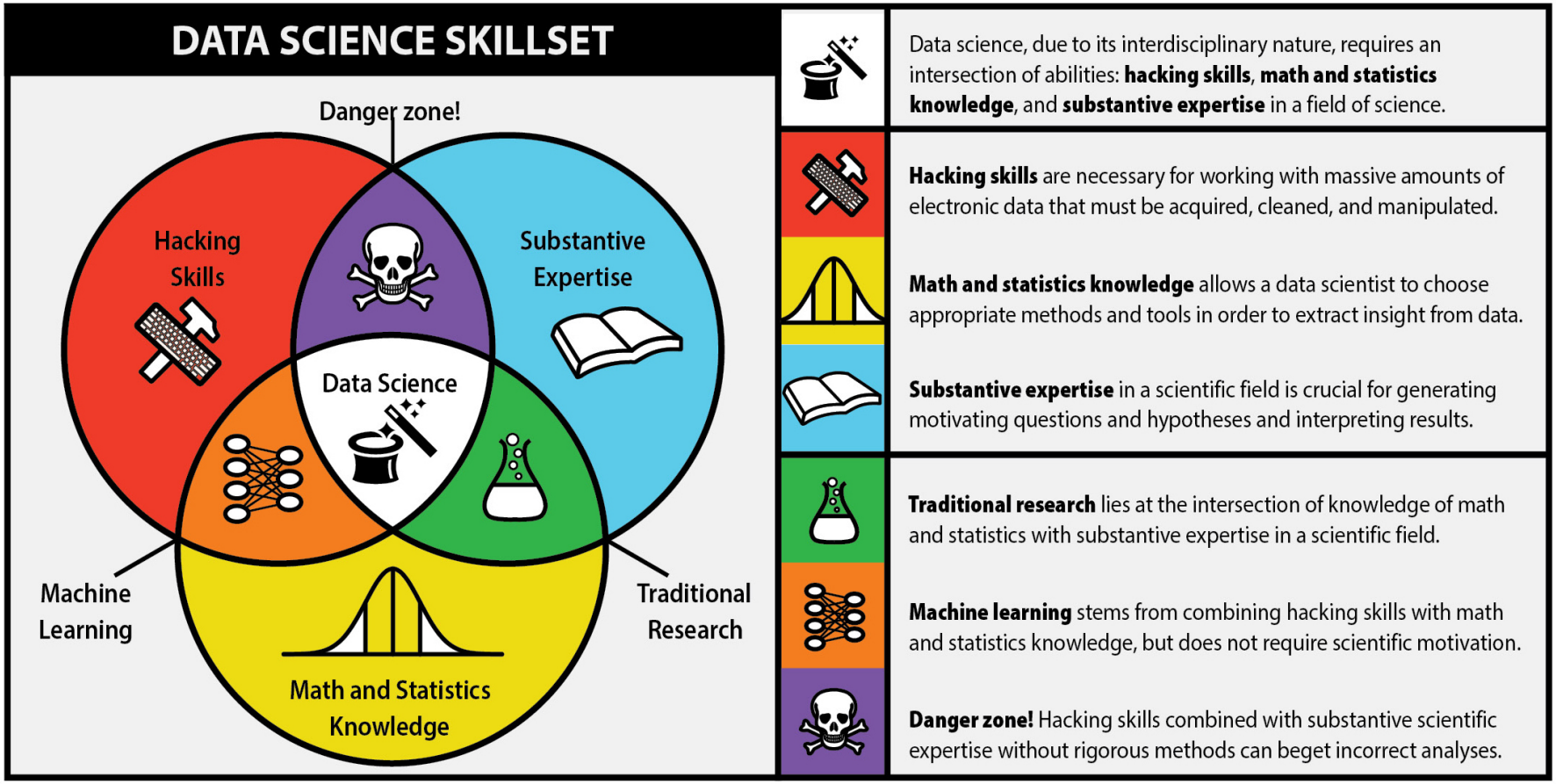
David Obst - david.obst@edf.fr

28 & 29 avril 2022

EDF R & D - Aix Marseille Université

Introduction

Pourquoi faire des statistiques ?



Phases de l'approche Statistique

1. Collecter les données, les labéliser et s'assurer de leur qualité.
2. Description des données et leur visualisation.
3. Modélisation, inférence, prévision (Quel modèle utiliser et pourquoi ?
   Que me permet d'apprendre le modèle sur le phénomène ?
   Comment l'exploiter pour des données futures ?).

Phases de l'approche Statistique

1. Collecter les données, les labéliser et s'assurer de leur qualité.
2. Description des données et leur visualisation.
3. Modélisation, inférence, prévision (Quel modèle utiliser et pourquoi ?
   Que me permet d'apprendre le modèle sur le phénomène ?
   Comment l'exploiter pour des données futures ?).

Phases de l'approche Statistique

1. Collecter les données, les labéliser et s'assurer de leur qualité.
2. Description des données et leur visualisation.
3. Modélisation, inférence, prévision (Quel modèle utiliser et pourquoi ?
   Que me permet d'apprendre le modèle sur le phénomène ?
   Comment l'exploiter pour des données futures ?).

Outils de Statistiques / Data Science

- Python avec **Numpy, ScikitLearn, Matplotlib**, Keras.

- Alternative populaire: `R`.

Table des matières

Statistiques descriptives et visualisation de base

Données

On considère un **échantillon / jeu de données** (dataset) $x = (x_1, x_2, \ldots, x_n)$ où chaque $x_i \in \mathbb{R}^d$.

- n est la **taille** du jeu de données.
- d est la **dimension** du problème, correspondant au nombre de variables disponibles.

En statistiques classique: $n \gg d$. Le cas où $d \approx n$ voire $d \geq n$ est celui du **Big Data**.

Sexe	Taille	Poids
F	166	64
H	176	80
H	181	82
F	170	63

Table 1: Exemple d'un dataset. Ici $n = 4$, $d = 3$.

Données

Chaque échantillon $x_i = (x_{i1}, x_{i2}, \ldots, x_{id})$. Les x_{ij} sont les **variables (aussi appelées features ou régresseurs)** du jeu de données. Elles peuvent être **continues ou catégorielles**.

Ex: Dans notre exemple précédent $x_1 = (F, 166, 64)$. Bien entendu, le catactère "F" devra être encodé numériquement (one-hot encoding par exemple).

Enfin, il ne faut pas oublier que les x_{ij} sont **aléatoires** ! L'hypothèse sous-jacente est donc qu'elles sont générées par une certaine loi de probabilités sous-jacente.
$\Rightarrow$ On va calculer des moyennes, écart-types, distribution, etc. dessus.

Description de données univariées ($d = 1$)

Objectif: Donner des indicateurs numériques sur les données. Les indicateurs suivants sont les plus utilisés:

1. Moyenne $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$.
2. Ecart-type/Variance: $\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2}$.
3. Médiane: $\text{Med}(x) = \text{plus petit } y \text{ tq } \hat{F}_n(y) \geq 0.5$
4. Quantiles d'ordre a: $\text{Quant}_a(x) = \text{plus petit } y \text{ tq } \hat{F}_n(y) \geq a$

Description de données univariées ($d = 1$)

Objectif: Donner des indicateurs numériques sur les données. Les indicateurs suivants sont les plus utilisés:

1. Moyenne $\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$.
2. Ecart-type/Variance: $\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \overline{x})^2}$.
3. Médiane: $\text{Med}(x) = \text{plus petit } y \text{ tq } \hat{F}_n(y) \geq 0.5$
4. Quantiles d'ordre a: $\text{Quant}_a(x) = \text{plus petit } y \text{ tq } \hat{F}_n(y) \geq a$

Description de données univariées ($d = 1$)

Objectif: Donner des indicateurs numériques sur les données. Les indicateurs suivants sont les plus utilisés:

1. Moyenne $\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$.
2. Ecart-type/Variance: $\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \overline{x})^2}$.
3. Médiane: $\text{Med}(x) = \text{plus petit } y \text{ tq } \hat{F}_n(y) \geq 0.5$
4. Quantiles d'ordre a: $\text{Quant}_a(x) = \text{plus petit } y \text{ tq } \hat{F}_n(y) \geq a$

Description de données univariées ($d = 1$)

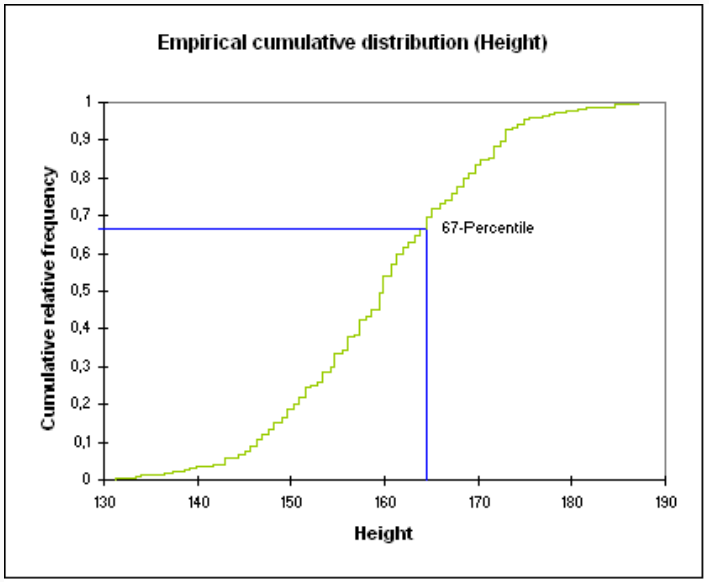
Objectif: Donner des indicateurs numériques sur les données. Les indicateurs suivants sont les plus utilisés:

1. Moyenne $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$.
2. Ecart-type/Variance: $\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2}$.
3. Médiane: $\text{Med}(x) = \text{plus petit } y \text{ tq } \hat{F}_n(y) \geq 0.5$
4. Quantiles d'ordre a: $\text{Quant}_a(x) = \text{plus petit } y \text{ tq } \hat{F}_n(y) \geq a$

Description de données univariées ($d = 1$)

Objectif: Donner des indicateurs numériques sur les données. Les indicateurs suivants sont les plus utilisés:

1. Moyenne $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$.
2. Ecart-type/Variance: $\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2}$.
3. Médiane: $\text{Med}(x) = \text{plus petit } y \text{ tq } \hat{F}_n(y) \geq 0.5$
4. Quantiles d'ordre a: $\text{Quant}_a(x) = \text{plus petit } y \text{ tq } \hat{F}_n(y) \geq a$

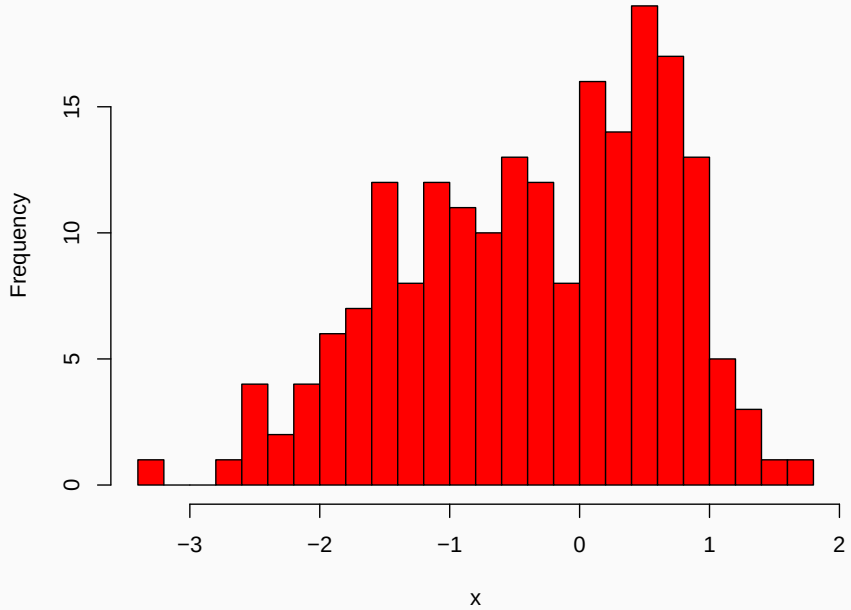


Visualisation élémentaire de données

Chaque visualisation a ses spécificités, ses forces et ses faiblesses. Il faut donc bien la choisir en fonction de ce que l'on souhaite représenter et identifier. Voici quelques exemples usuels.

Histogramme

Il permet de représenter x sous forme d'une densité de probabilité (discrétisée). Néanmoins il nécessite la disponibilité d'un grand nombre d'échantillons n.



Avantages:

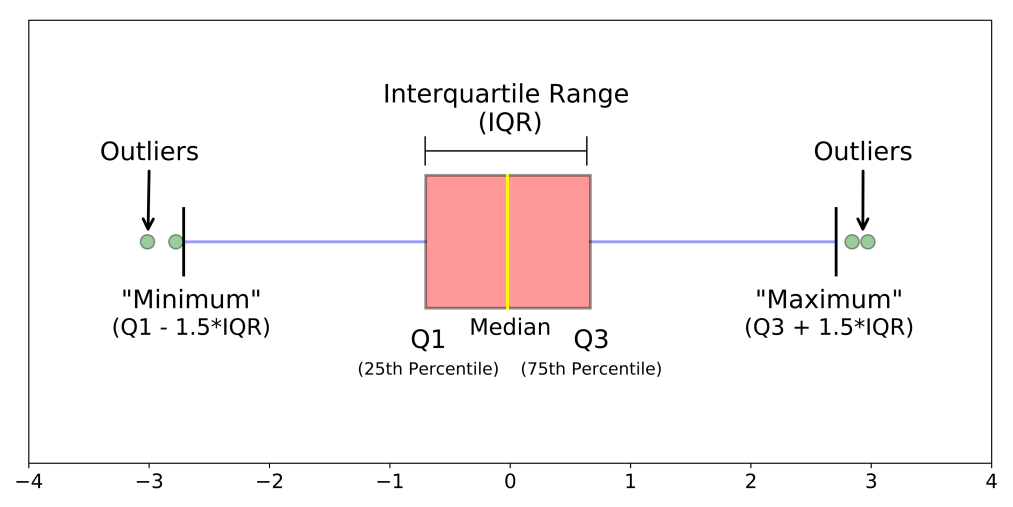
- Permet de voir rapidement la distribution des échantillons.

Inconvénients:

- Ne permet pas de voir clairement la moyenne, les quantiles et nécessite un nombre de données important.
- Nécessite de régler le nombre/la taille des rectangles de l'histogramme.

Boxplot (Boîte à Moustaches)

Il résume la répartition du jeu de données via les quantiles et la médiane.



Avantages:

- Permet d'avoir les valeurs précises de la médiane, des quartiles Q_1 et Q_3 (quantiles d'ordre 25 et 75%).
- Permet de visualiser les *outliers* et leur nombre.

Inconvénients:

- Peu d'informations sur la distribution entre les traits.

Comparer des données entre elles ($d \geq 2$)

Souvent un jeu de données sera composé de multiples variables (e.g. taille, âge, sexe, ...).

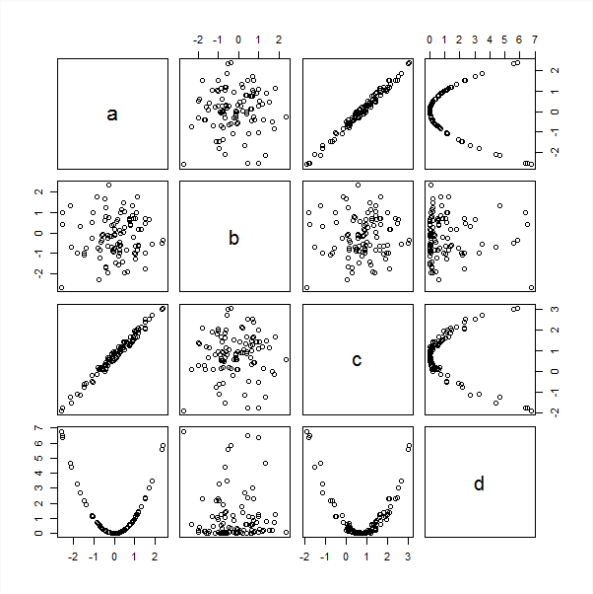
On va alors essayer de trouver des liens de causalité entre elles. Typiquement cela est fait pour construire un modèle d'inférence / de prévision.

Exemples :

- Pairs (scatter)plots.
- QQplots.
- Corrélation (linéaire) de Pearson.
- Information Mutuelle.

Pairs plot

Lors de la conception d'un modèle de classification ou de régression, il est souvent indispensable de déterminer les liens de causalité entre variables.



Avantages:

- Permet de voir directement comment les variables dépendent les unes des autres, guidant donc le choix des features à inclure dans le modèle.

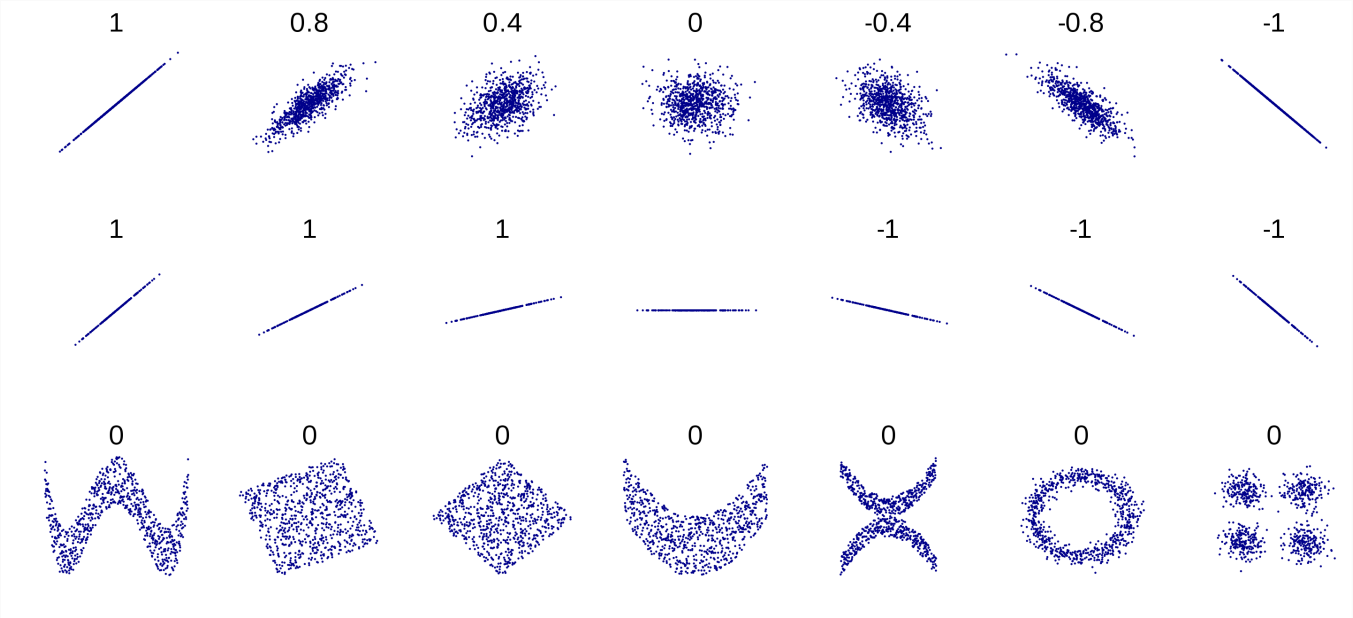
Inconvénients:

- Devient illisible et long à calculer quand d est grand (typiquement $d \geq 10$).

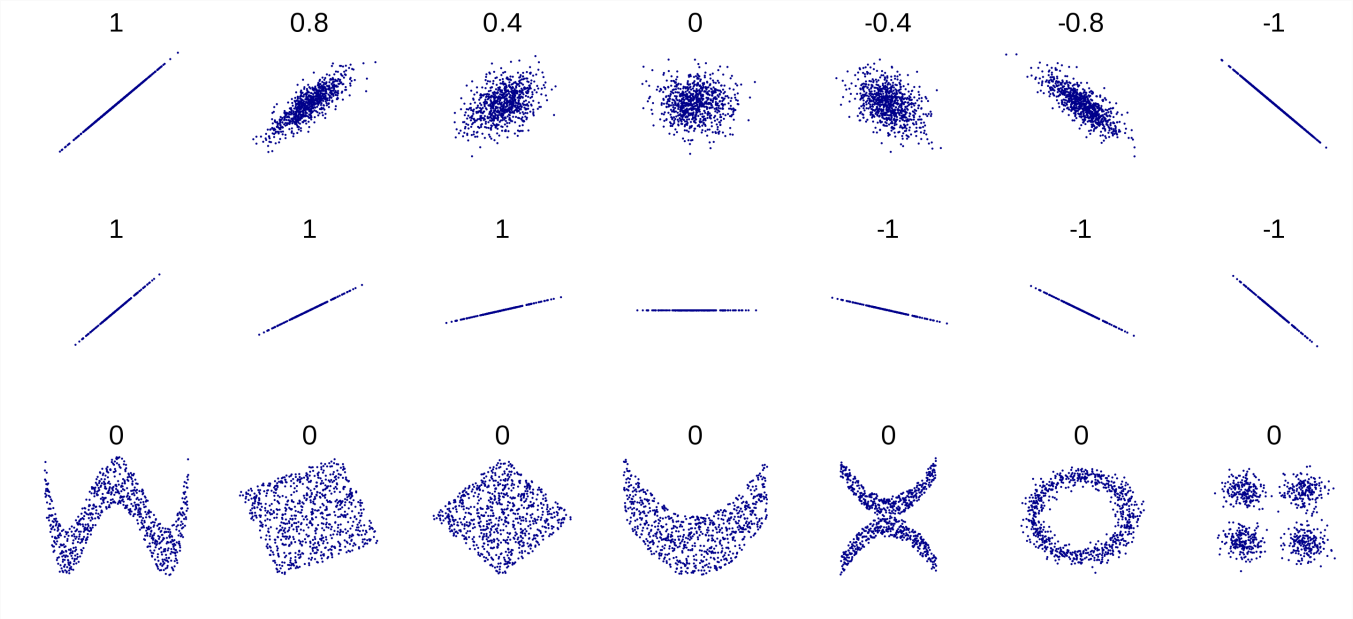
Corrélation de Pearson

La corrélation linéaire de Pearson entre deux séries de données $x = (x_1, \ldots, x_n)$ et $y = (y_1, \ldots, y_n)$ est définie par:

$$\rho(x, y) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sigma_x \sigma_y} \tag{1}$$



Corrélation de Pearson (II)



Avantages:

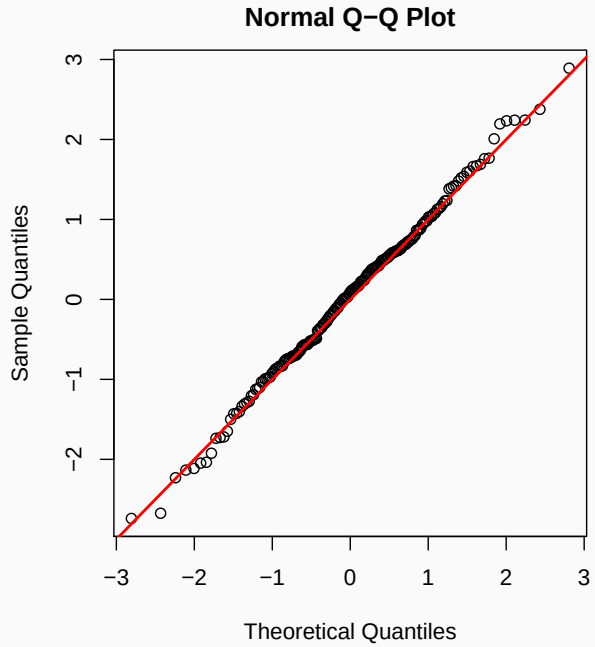
- Permet d'établir un lien de causalité entre variables.

Inconvénients:

- Ne détecte que des liens linéaires.

QQplots

Permet de comparer les quantiles empiriques d'un échantillon x avec ceux d'un autre échantillon x' ou avec les quantiles théoriques d'une loi.



Avantages:

- Permet de vérifier visuellement l'adéquation d'un échantillon avec une loi connue (e.g. la loi normale).

Inconvénients:

- En soi ne donne aucune information sur le type de distribution que les échantillons pourraient suivre.

Apprentissage non-supervisé - Réduction de dimension, visualisation et clustering

Visualisation en dimension $d > 1$?

Pour le moment nous nous sommes concentré sur la visualisation de données *univariées*, càd que $d = 1$ (ex: nous avons juste des tailles, des poids, ...). Nous aimerions visualiser un JDD $x \in \mathbb{R}^{n \times d}$ quelconque.

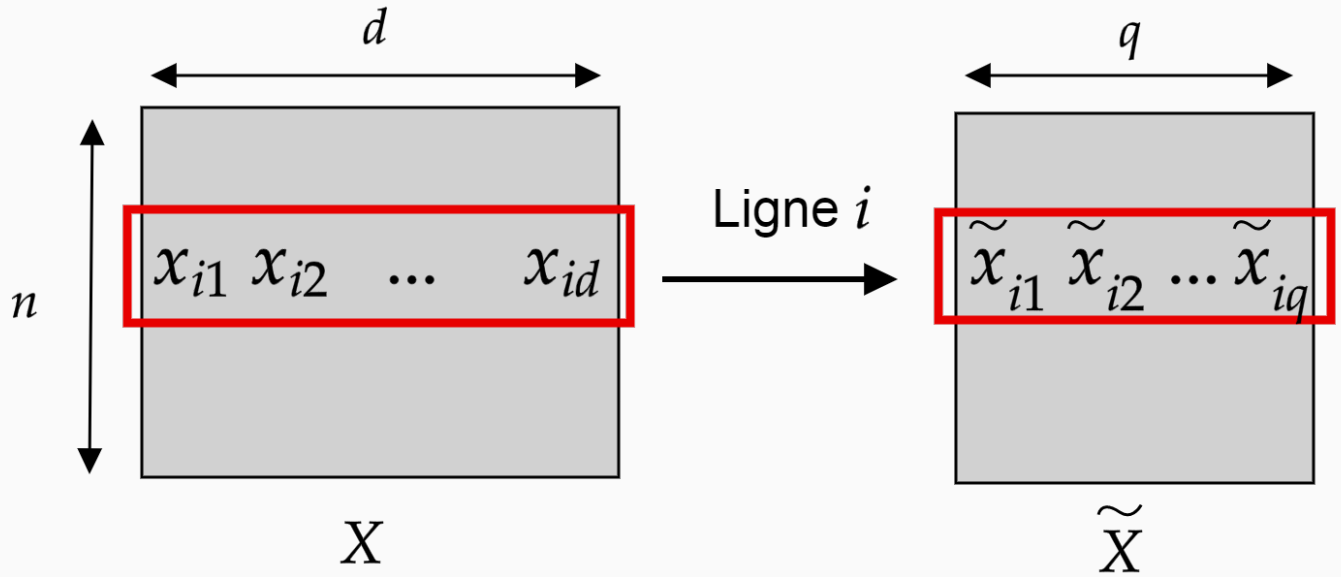
Problèmes:

- Quand $d \geq 4$, que faire ? (on voudra se ramener à la dimension 2, au 3 au plus).
- Dans le cadre du Big Data, d sera souvent immense. Comment le réduire pour les modèles en aval ?
- Comment tenir compte de l'hétérogénéité des variables x_j ?
- Comment inférer des choses sur une représentation forcément réductrice ?

L'Analyse en Composantes Principales (ACP)

On note X la matrice où les $x_i \in \mathbb{R}^d$ sont rangés par ligne.

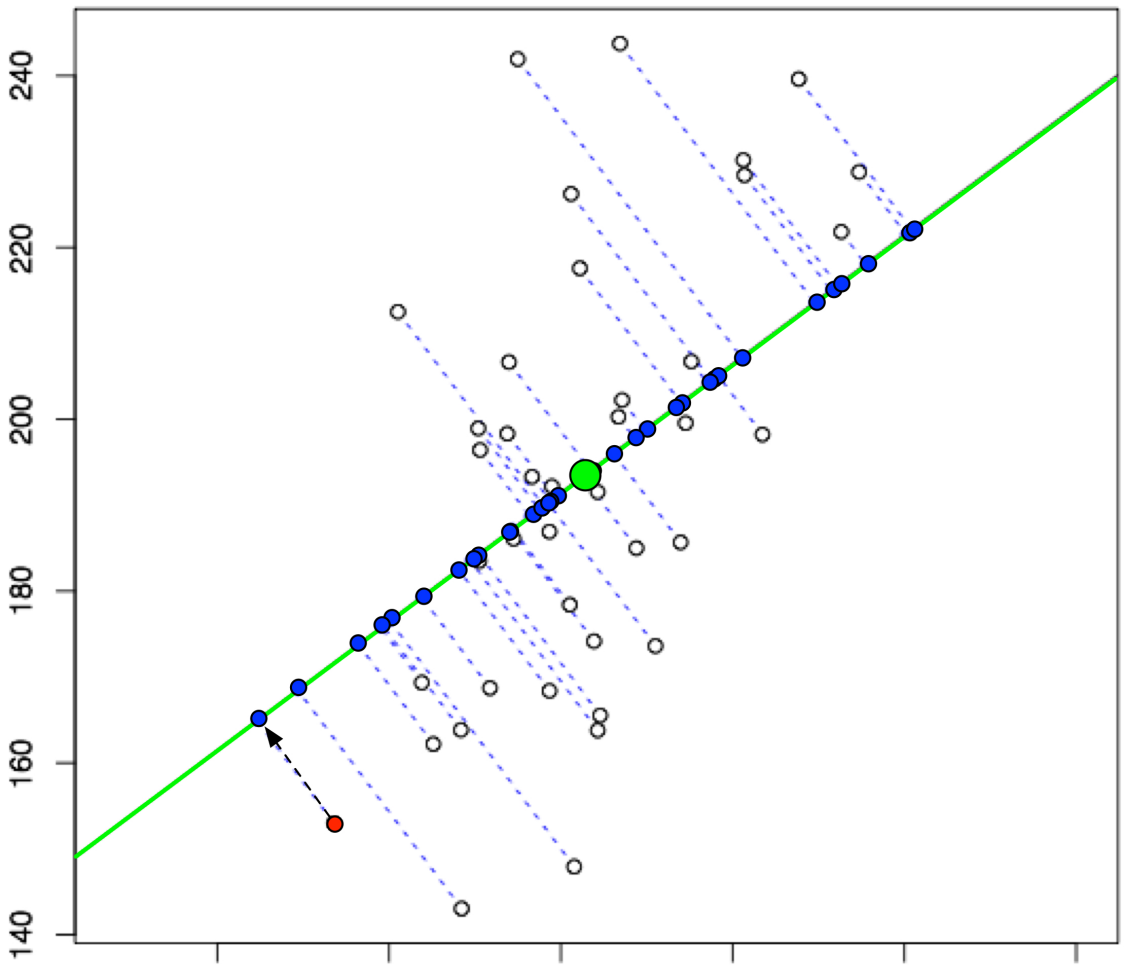
L'Analyse en Composantes Principales (ACP) est une méthode qui va permettre de réduire X de dimension $n \times d$ en $\tilde{X}$ de dimension $n \times q$ avec $q < d$ tout en gardant le maximum d'information de X. Le plus souvent on prendra $q = 2$.



Intuitivement, l'ACP combine les d variables du JDD en variables résumant de façon décroissante l'information.

Principe de l'ACP

Il s'agit de déterminer des directions orthogonales appelées **axes principaux** conservant au mieux les propriétés du nuages de points.

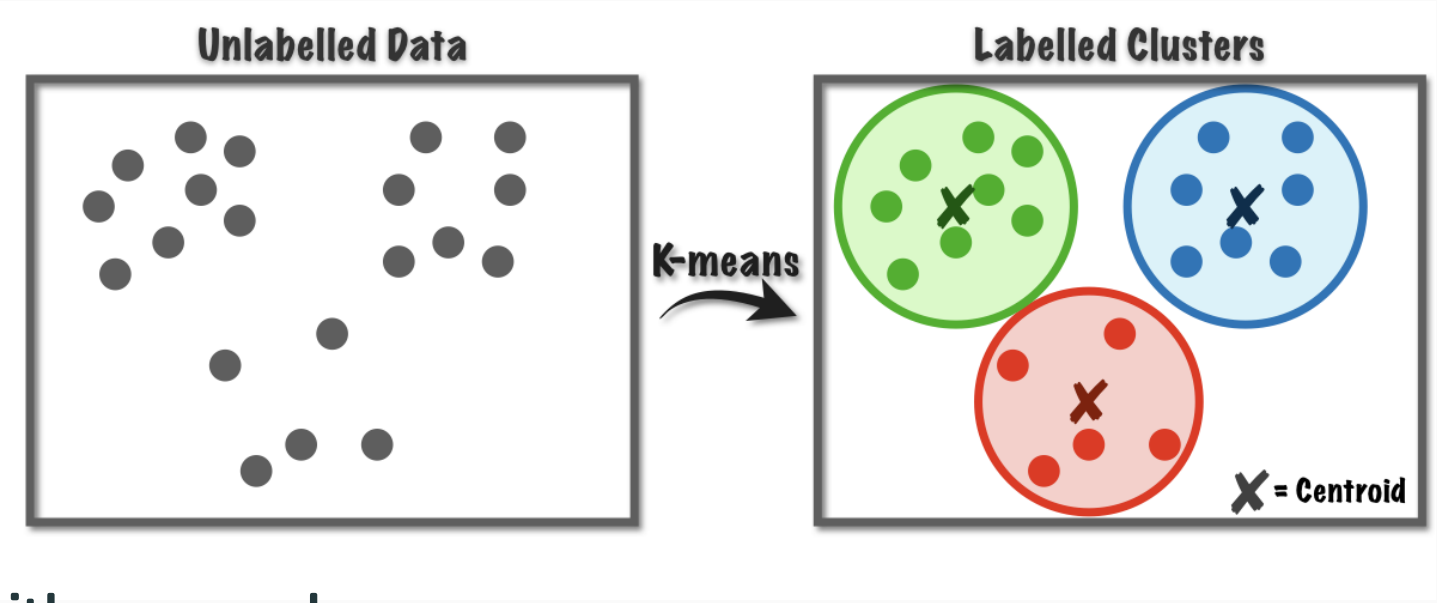


Méthodologie de l'ACP

1. Il faut **centrer et réduire** X, i.e. soustraire à chaque colonne sa moyenne (empirique) et la diviser par son écart-type.
2. Utiliser la fonction `PCA` de ScikitLearn (ou de `R`) qui calcule les w_k et λ_k.
3. On obtient ainsi la matrice $\tilde{X} = X[\frac{w_1}{\sqrt{\lambda_1}}, \frac{w_2}{\sqrt{\lambda_2}}]$ (par ex). qui nous intéressent pour la représentation (souvent les 1 et 2 ou 1 et 3). On peut vérifier la quantité de variance préservée par l'ACP. $\Rightarrow$ Typiquement une ACP est bonne quand au moins de 75% de la variance est conservée dans les dimensions 1 et 2.
4. On peut également tracer le cercle des corrélations.

Le clustering

Le **clustering** permet de partitionner un jeu de données en K groupes ayant des propriétés semblables. $\Rightarrow$ on regroupe les individus "proches"



Algorithmes usuels:

1. K-means
2. CAH (classification Ascendante Hiérarchique)
3. Clustering spectral

On généralement combiner K-means et ACP pour visualiser les groupes en 2D.

L'algorithme K-Means

Pour un entier K donné (nombre de clusters) on cherche $\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^d$ les **centroïdes** et une partition $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K$ minimisant:

$$\sum_{j=1}^{K} \sum_{x \in \mathcal{C}_k} \|x - \mu_i\|^2$$

Algorithme K-means:

Entrées: data $x_1, x_2, \ldots, x_n$, nb de clusters K. Initialisation des μ_j.

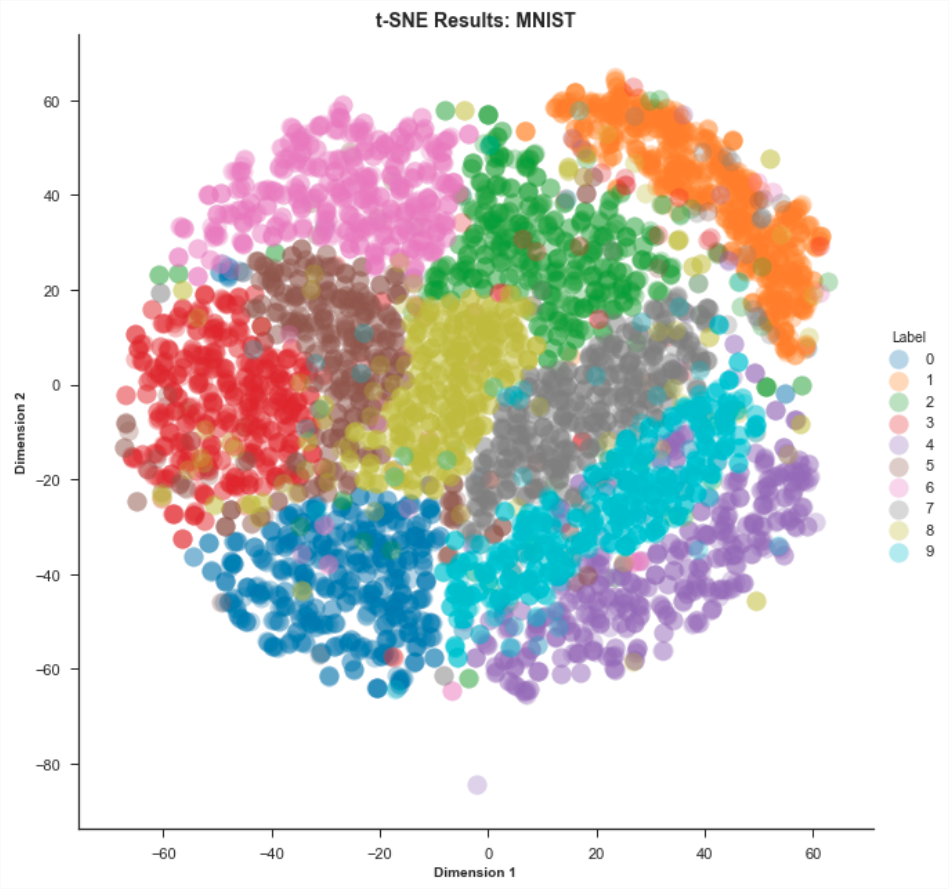
Pour $t = 1..N_{iter}$:

- Associer chaque x_i au $\mu_j^{(t-1)}$ le plus proche.
- Mettre à jour $\mu_j^{(t)} = \frac{1}{|\mathcal{C}_j|} \sum_{x_i \in \mathcal{C}_j} x_i$

1. **Avantage:** Facile à utiliser, converge toujours.
2. **Inconvénients:** Solution approchée, convient aux clusters "patatoïdes".

La projection t-SNE

Méthode relativement récente qui a été popularisée dans le cadre du Big Data, elle ressemble à certains égards à l'ACP mais avec moins d'inconvénients. Elle permet ainsi de projeter des nuages de points en très grande dimension en 2 dimensions.



Bases de la modélisation statistique & rappels de probabilités

Bases de la modélisation statistique

Nous avons un échantillon $(x_1, \ldots, x_n)$ avec $x_i \in \mathbb{R}^d$. On suppose que les échantillons sont **i.i.d. (indépendant et identiquement distribués)**.

Ex: vous interrogez au hasard des personnes dans la rue: elles ne *s'influencent pas* et auront toutes *les mêmes probabilités de voter pour les candidats*.

Souvent on supposera que x_i est issu d'une loi de probabilités sous-jacente dont il faudra estimer les paramètres.

Ex: Modèle de Bernoulli pour le vote au second tour.

Rappels de probabilités

On appelle **distribution** de probabilités une fonction $p(\cdot)$ définie sur $\mathbb{R}$ telle que:

1. $p(x)$ est toujours positif.
2. $p(\cdot)$ n'admet qu'un nombre fini de discontinuités.
3. L'aire sous la courbe vaut 1 (cas continu) ou $\sum_x p(x) = 1$ (cas discret).

Dans le cas discret, la probabilité d'une valeur est directement donnée par l'ordonnée de la distribution. Dans le cas continu, c'est **l'aire sous la courbe** entre deux valeurs.

La loi uniforme $\mathcal{U}([a, b])$

Deux paramètres: a et b avec $a < b$.

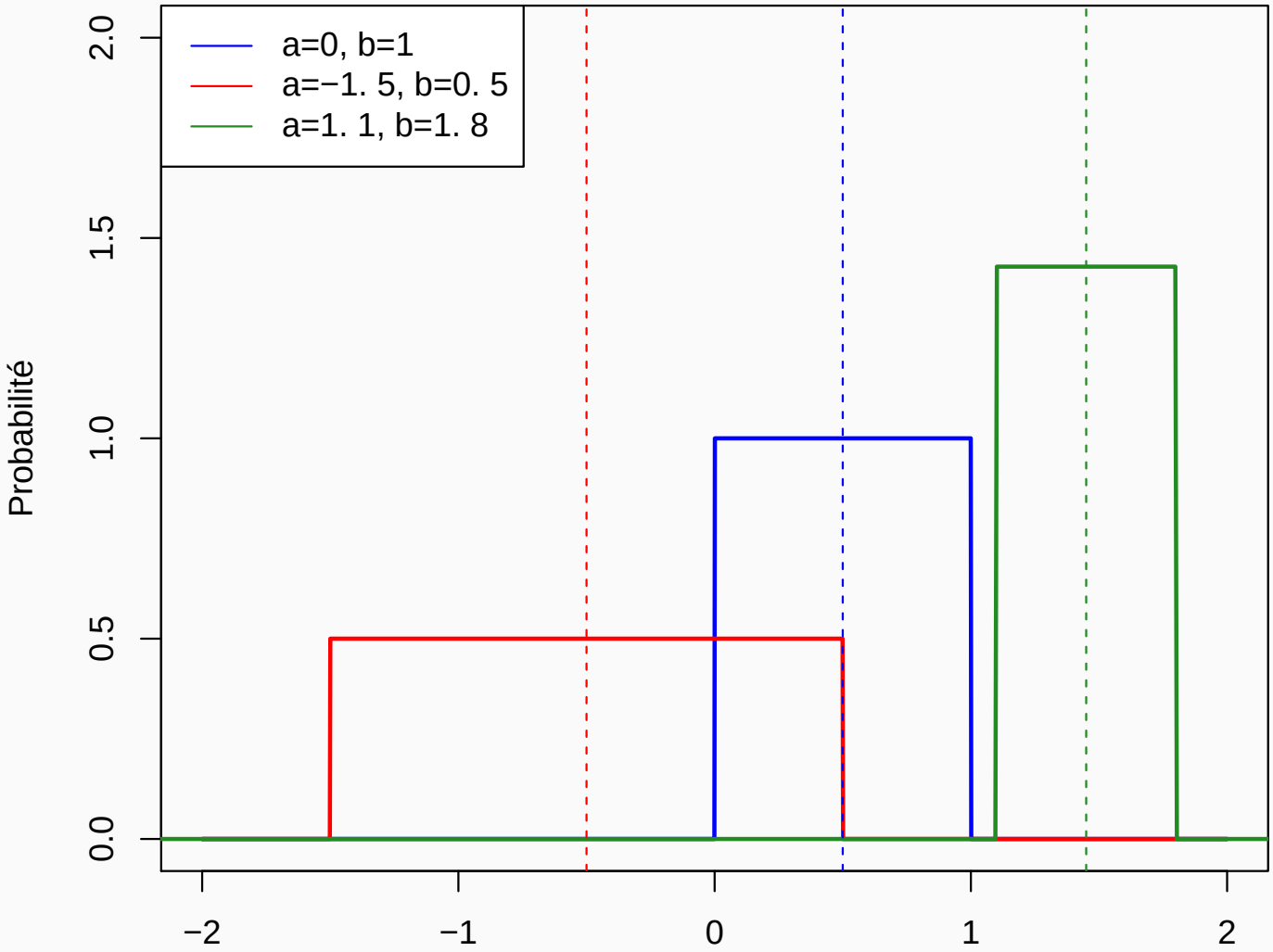


Figure 1: La loi uniforme pour différents paramètres a et b.

La loi normale $\mathcal{N}(\mu, \sigma^2)$

Deux paramètres μ, σ^2 avec $\sigma^2 > 0$.

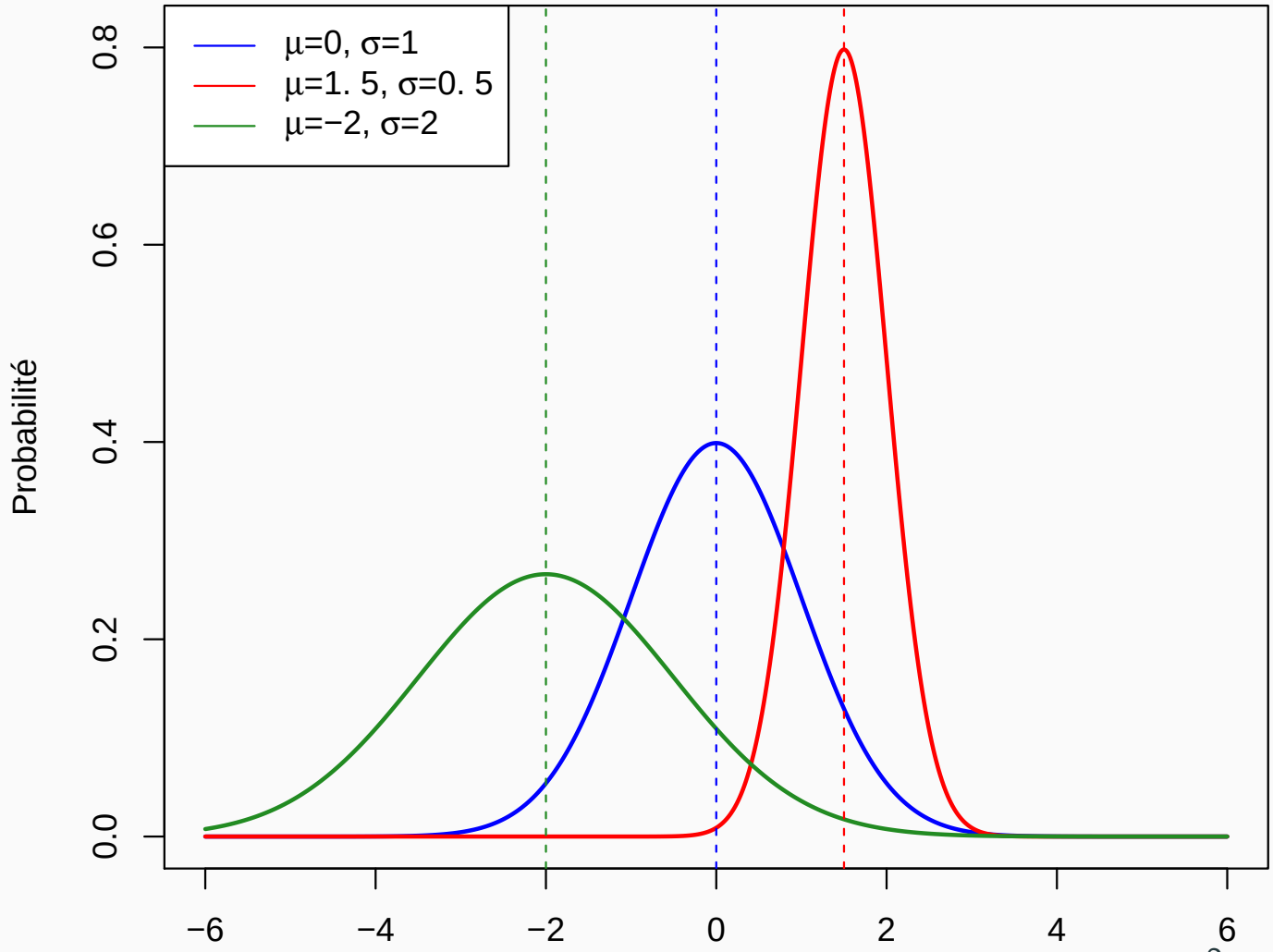
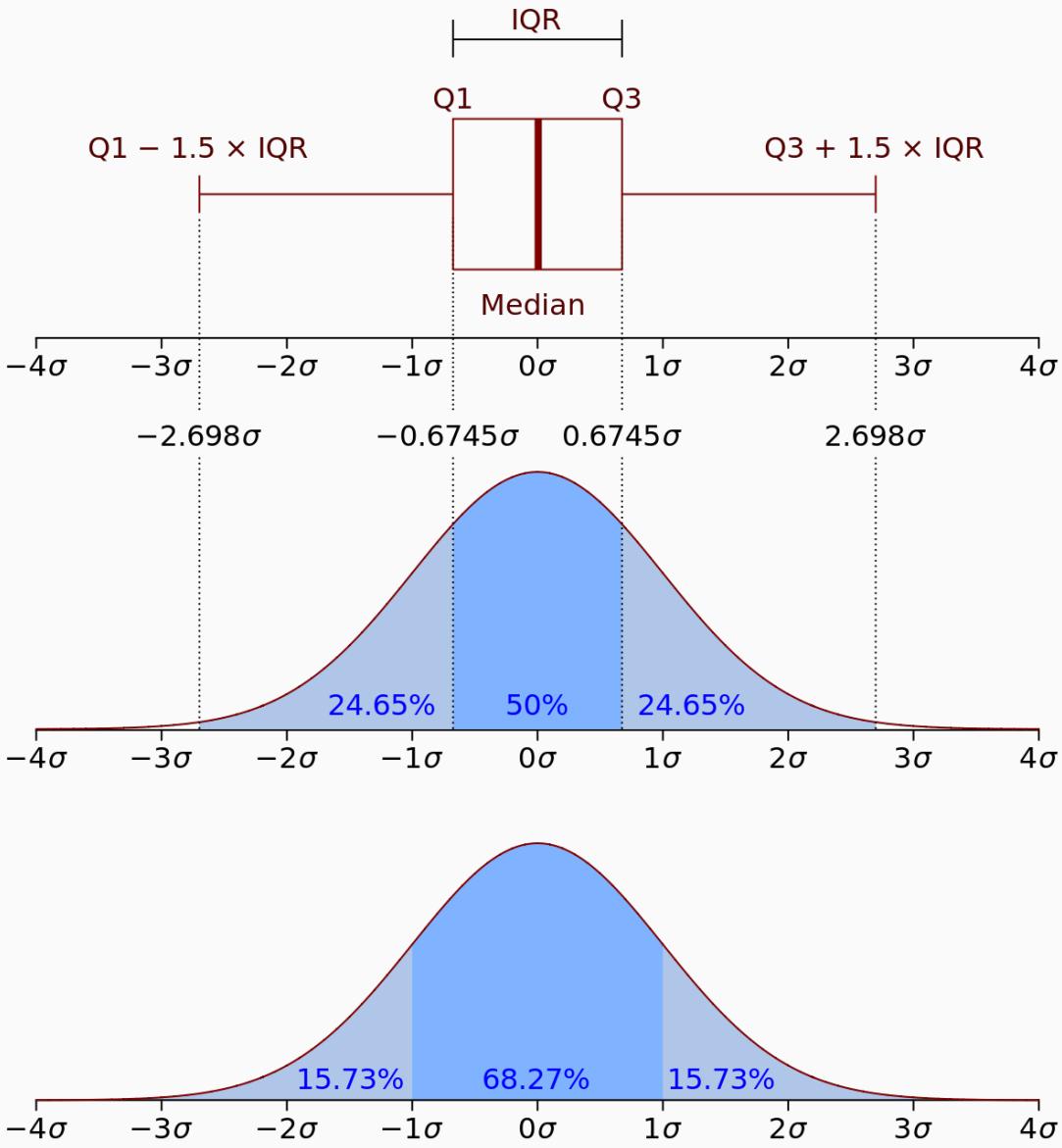


Figure 2: La loi normale pour différents paramètres (μ, σ^2).

Visualisation sur un exemple



La loi exponentielle $\mathcal{E}(\lambda)$

Un seul paramètre $\lambda \in \mathbb{R}_+^*$.

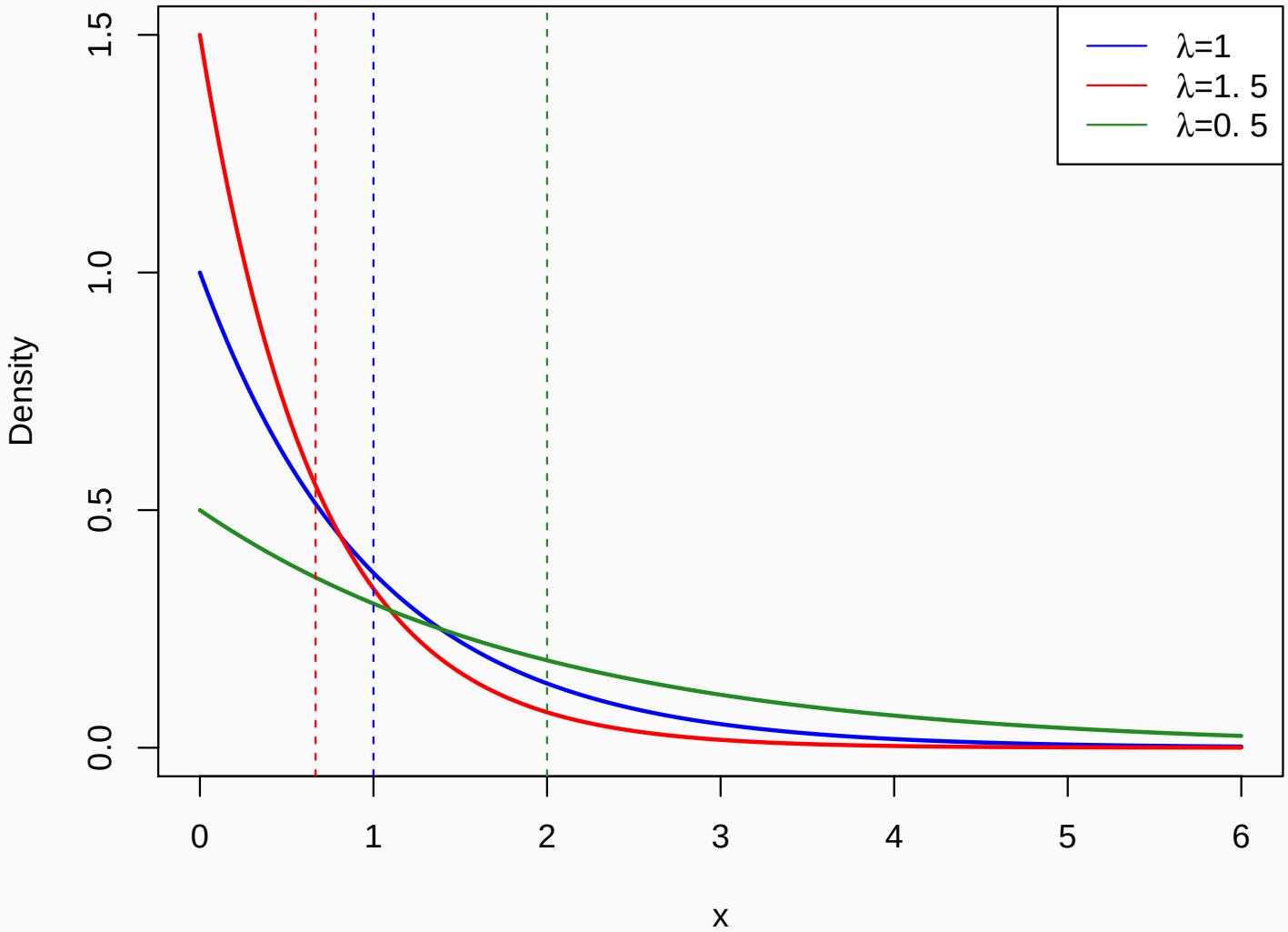


Figure 3: La loi exponentielle pour différents paramètres λ

Espérance et variance

L'espérance d'une variable aléatoire X, notée $\mathbb{E}[X]$, est la "moyenne" :

- Cas discret : $\mathbb{E}[X] = \sum_x x\, p(x)$.
- Cas continu : $\mathbb{E}[X] = \int_x x\, p(x)\, \mathrm{d}x$.

La **variance** d'une variable aléatoire X, notée $\mathrm{Var}(X)$, est la répartion autour de l'espérance:

$$\mathrm{Var}(X) = \mathbb{E}\Big[(X - \mathbb{E}[X])^2\Big] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$$

Estimation

On va supposer que les données $z_1, z_2, \ldots, z_n$ sont i.i.d. (indépendantes et identiquement distribuée) selon une certaine loi de probabilités p_θ. Cependant on ne connaît pas θ: on va chercher à **l'estimer** à partir des données.

Exemples :

- On sait que les z_i sont tirés selon une loi normale: on va estimer la moyenne et la variance avec $\overline{z} = \frac{1}{n}\sum_i z_i$ et $\hat{\sigma}^2 = \frac{1}{n}\sum_i (z_i - \overline{z})^2$ respectivement.
- On cherche à estimer le coefficient d'un modèle linéaire.

Intervalles de confiance et tests

Une fois que l'on a un estimateur, il semble naturel de se poser les questions suivantes:

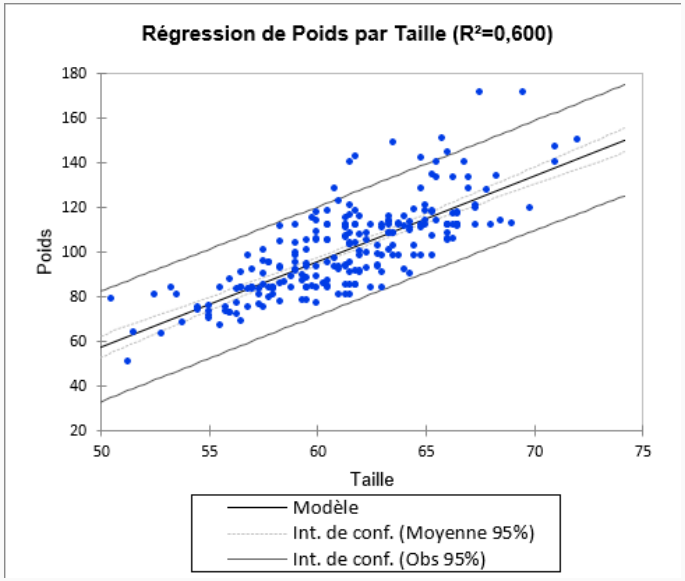
- Peut-on construire un intervalle à partir de $\hat{\theta}_n$ pour lequel on aura une certaine probabilité que θ s'y situe ?
- L'estimateur permet t-il de confirmer ou d'infirmer quelque chose sur la vraie valeur de θ ?

Un **intervalle de confiance de θ (IC)** de niveau $1-\alpha$ est un intervalle I_α construit à partir de $\hat{\theta}_n$ tel que $P(\theta \in I_\alpha) \geq 1-\alpha$.

Un **test d'hypothèses statistiques** va permettre de tester si θ vaut une certaine valeur ou non.

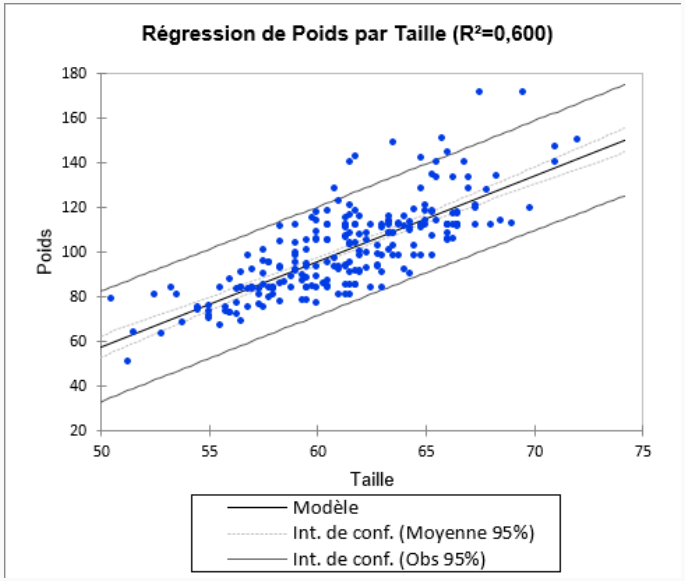
Apprentissage supervisé - Le modèle linéaire

Pourquoi étudier ce modèle ?



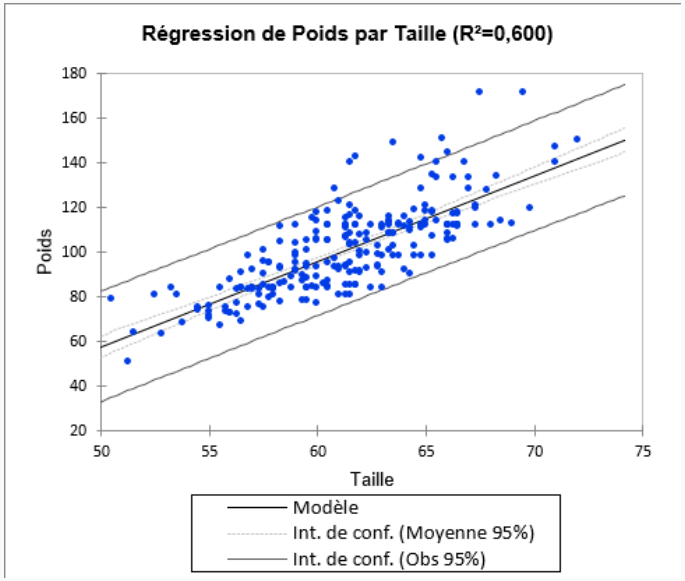
1. Il s'agit du modèle fondateur de la régression, et donc fondateur du Machine Learning moderne.
2. Le modèle est interprétable, ce qui est indispensable souvent pour des applications industrielles.
3. Le modèle reste très utilisé même de nos jours (GAM, interprétation de réseaux de neurones, etc...).

Pourquoi étudier ce modèle ?



1. Il s'agit du modèle fondateur de la régression, et donc fondateur du Machine Learning moderne.
2. Le modèle est interprétable, ce qui est indispensable souvent pour des applications industrielles.
3. Le modèle reste très utilisé même de nos jours (GAM, interprétation de réseaux de neurones, etc...).

Pourquoi étudier ce modèle ?



1. Il s'agit du modèle fondateur de la régression, et donc fondateur du Machine Learning moderne.
2. Le modèle est interprétable, ce qui est indispensable souvent pour des applications industrielles.
3. Le modèle reste très utilisé même de nos jours (GAM, interprétation de réseaux de neurones, etc...).

Le modèle

On a nos données $\{(x_i, y_i)\}_{i=1..n}$ où:

- y_i est la variable que l'on cherche à prédire / expliquer (*réponse*). Ici on suppose qu'elle est **quantitative** (e.g. les prix).
- x_i est un vecteur contenant les variables explicatives (régresseurs, e.g.les infos sur les salaires, la population, le nombre de chambres...).

L'hypothèse du modèle linéaire est de représenter y_i par la formule :

$$y_i = b_0 + \sum_{j=1}^{D-1} b_j \, x_{ij} + e_i \quad \text{où } e_i \sim \mathcal{N}(0, \sigma^2) \text{ est i.i.d.}$$

Néanmoins le modèle peut inclure des non-linéarités avec une écriture judicieuse :

$$y_i = b_0 + b_1 \, x_i + b_2 x_i^2 + e_i$$

Le modèle

On a nos données $\{(x_i, y_i)\}_{i=1..n}$ où:

- y_i est la variable que l'on cherche à prédire / expliquer (*réponse*). Ici on suppose qu'elle est **quantitative** (e.g. les prix).
- x_i est un vecteur contenant les variables explicatives (régresseurs, e.g.les infos sur les salaires, la population, le nombre de chambres...).

L'hypothèse du modèle linéaire est de représenter y_i par la formule :

$$y_i = b_0 + \sum_{j=1}^{D-1} b_j \, x_{ij} + e_i \quad \text{où } e_i \sim \mathcal{N}(0, \sigma^2) \text{ est i.i.d.}$$

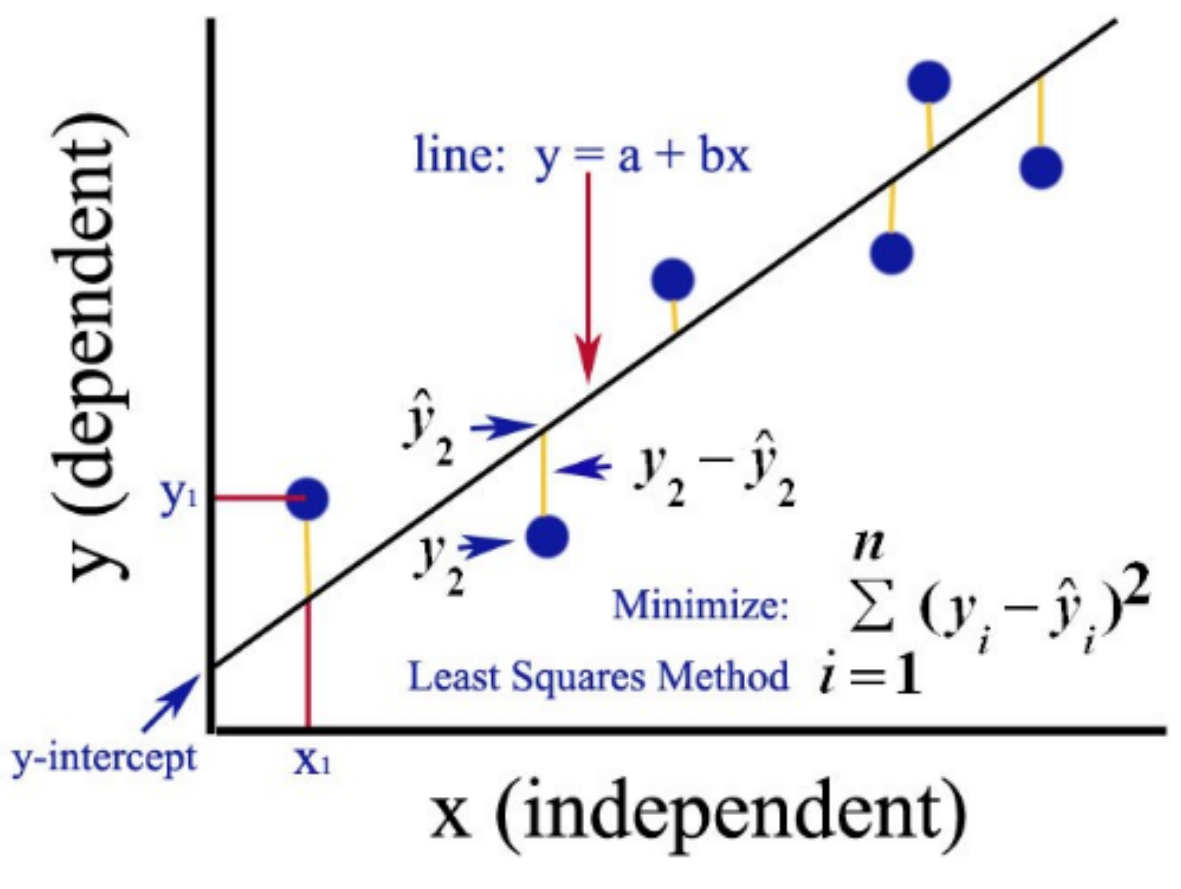
Néanmoins le modèle peut inclure des non-linéarités avec une écriture judicieuse :

$$y_i = b_0 + b_1 \, x_i + b_2 x_i^2 + e_i$$

Estimation des coefficients b_i

On va résoudre le problème des moindres carrés ordinaires (MCO - OLS en anglais) :

$$\hat{b}_{MC} = \text{argmin}_{b_0, b_1, \ldots, b_{D-1}} \sum_{i=1}^{n} \left(y_i - b_0 - \sum_{j=1}^{D-1} b_j \, x_{ij} \right)^2$$



Interprétation des résultats

```
In [10]: print(res.summary())
                            OLS Regression Results
==============================================================================
Dep. Variable:                Lottery   R-squared:                       0.338
Model:                            OLS   Adj. R-squared:                  0.287
Method:                 Least Squares   F-statistic:                     6.636
Date:                Sat, 28 Nov 2020   Prob (F-statistic):           1.07e-05
Time:                        14:39:43   Log-Likelihood:                -375.30
No. Observations:                  85   AIC:                             764.6
Df Residuals:                      78   BIC:                             781.7
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      38.6517      9.456      4.087      0.000      19.826      57.478
Region[T.E]   -15.4278      9.727     -1.586      0.117     -34.793       3.938
Region[T.N]   -10.0170      9.260     -1.082      0.283     -28.453       8.419
Region[T.S]    -4.5483      7.279     -0.625      0.534     -19.039       9.943
Region[T.W]   -10.0913      7.196     -1.402      0.165     -24.418       4.235
Literacy       -0.1858      0.210     -0.886      0.378      -0.603       0.232
Wealth          0.4515      0.103      4.390      0.000       0.247       0.656
==============================================================================
```

1. **Coeff. de détermination** $R^2 = \dfrac{\text{Variance expliquée par le modèle}}{\text{Variance totale}}$
2. `P>|t|` : p-value du test de Student. Plus elle est petite, plus cela signifie que le coefficient associé est **significatif** (c.f. slide suivante).

Test t de Student

Une variable intégrée au modèle est-elle réellement pertinente ? Pour vérifier cela, on va faire un **test d'hypothèses**:

$$H_0 : \{\text{La variable } i \text{ est inutile}\} \quad \text{contre} \quad H_1 : \{\text{La variable } i \text{ est utile}\}$$
$$\Leftrightarrow$$
$$H_0 : \{b_i = 0\} \quad \text{contre} \quad H_1 : \{b_i \neq 0\}$$

p-value: donne la probabilité de notre échantillon x si H_0 est vraie ($P_{H_0}(x)$).
$\Rightarrow$ **plus la p-value est petite, plus la preuve contre H_0 est forte**.

Typiquement on rejette H_0 quand $p_{val}(x) < 0.05$.

Vérifier notre modèle de régression

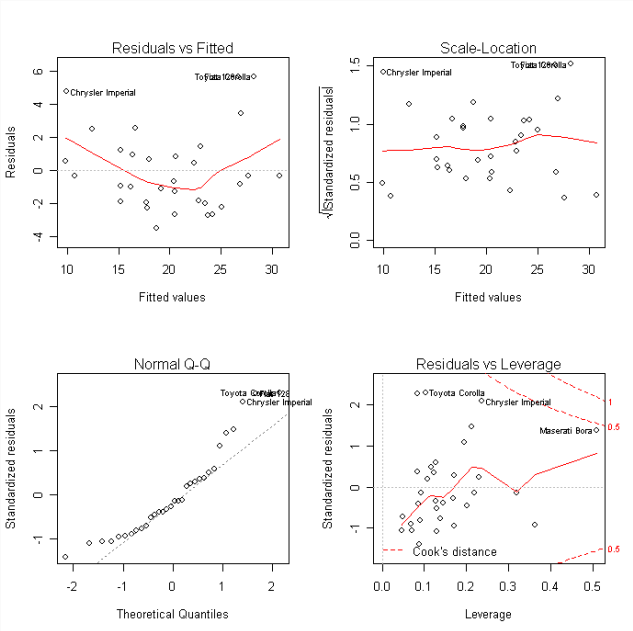
Après toute régression il faut vérifier que les résultats du modèle sont conformes et qu'ils ne violent pas (trop) les hypothèses fondamentales.

1. Adéquation des résidus $\hat{e}_i = \hat{y}_i - y_i$ avec une loi normale. (graphes histo, QQplot), homoscédasticité.
2. Indépendance statistique entre $\hat{e}_i$ et $\hat{y}_i$.
3. Performances du modèle.
4. Coefficients utiles à la régression.

Vérifier notre modèle de régression

Après toute régression il faut vérifier que les résultats du modèle sont conformes et qu'ils ne violent pas (trop) les hypothèses fondamentales.

1. Adéquation des résidus $\hat{e}_i = \hat{y}_i - y_i$ avec une loi normale. (graphes histo, QQplot), homoscédasticité.

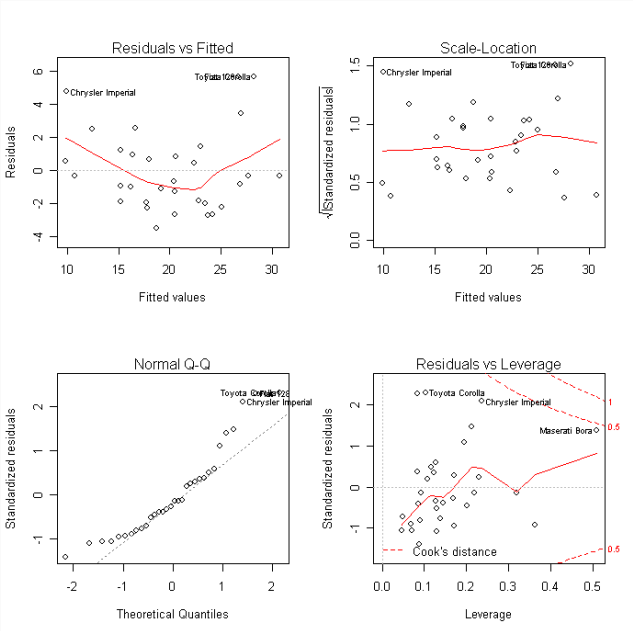


2. Indépendance statistique entre $\hat{e}_i$ et $\hat{y}_i$.
3. Performances du modèle.
4. Coefficients utiles à la régression.

Vérifier notre modèle de régression

Après toute régression il faut vérifier que les résultats du modèle sont conformes et qu'ils ne violent pas (trop) les hypothèses fondamentales.

1. Adéquation des résidus $\hat{e}_i = \hat{y}_i - y_i$ avec une loi normale. (graphes histo, QQplot), homoscédasticité.



2. Indépendance statistique entre $\hat{e}_i$ et $\hat{y}_i$.
3. Performances du modèle.
4. Coefficients utiles à la régression.

Vérifier notre modèle de régression

Après toute régression il faut vérifier que les résultats du modèle sont conformes et qu'ils ne violent pas (trop) les hypothèses fondamentales.

1. Adéquation des résidus $\hat{e}_i = \hat{y}_i - y_i$ avec une loi normale. (graphes histo, QQplot), homoscédasticité.
2. Indépendance statistique entre $\hat{e}_i$ et $\hat{y}_i$.
3. Performances du modèle.
4. Coefficients utiles à la régression.

Vérifier notre modèle de régression

Après toute régression il faut vérifier que les résultats du modèle sont conformes et qu'ils ne violent pas (trop) les hypothèses fondamentales.

1. Adéquation des résidus $\hat{e}_i = \hat{y}_i - y_i$ avec une loi normale. (graphes histo, QQplot), homoscédasticité.
2. Indépendance statistique entre $\hat{e}_i$ et $\hat{y}_i$.
3. Performances du modèle.
4. Coefficients utiles à la régression.

Pour résumer

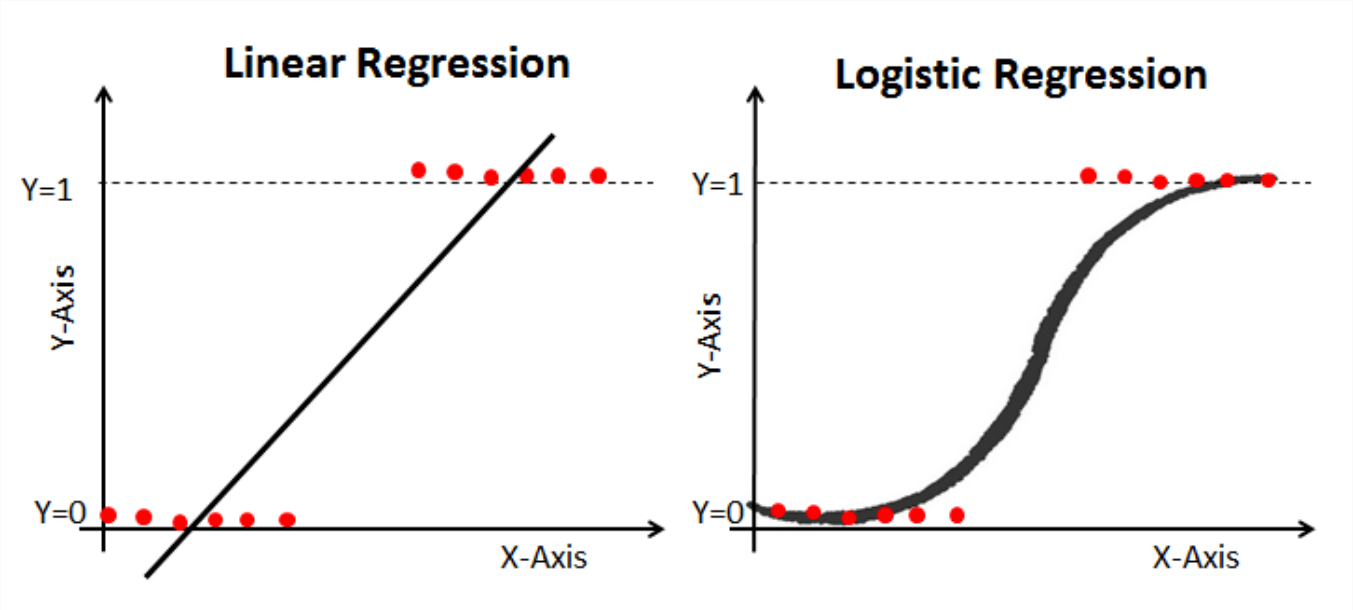
Lorsqu'on souhaite construire un modèle (pas exclusivement le linéaire):

1. On étudie le lien entre la variable qui nous intéresse (la réponse que l'on souhaite expliquer) et les autres variables (e.g. avec un pairs scatterplot).
2. On sépare notre jeu de données en un jeu d'apprentissage, et un de validation (typiquement avec le ratio 2/3 ; 1/3).
3. On apprend notre modèle de régression (e.g. `model = sm.OLS(Y,X)` puis `res = model.fit()`.
4. On effectue les étapes de vérification des résidus du modèle (caractère gaussien et homoscédastique avec histogramme, QQplot et scatterplot de $\hat{e}_i$ contre $\hat{y}_i$).
5. On évalue les performances du modèle sur le jeu de données de validation (avec RMSE, R^2, MAPE...).
6. On répète le processus lorsqu'on introduit / retire des variable et compare les modèles entre eux (e.g. AIC, BIC, RMSE, test des modèles emboîtés).

Petit bonus - La régression logistique

Différences avec la régression linéaire

En régression linéaire : on cherche à prévoir une variable **quantitative** ($y \in \mathbb{R}$). En régression logistique on va chercher à prévoir une variable catégorielle **binaire** ($y \in \{0, 1\}$).



Le modèle de régression logistique

Idée: On va modéliser la probabilité que pour x on ait $y(x) = 1$ via un modèle linéaire:

$$p(x_i) = \frac{1}{1 + \exp(-\beta_0 - \beta_1 x_{i,1} - \cdots - \beta_{D-1} x_{i,D-1})}$$

équivalent à :

$$\textbf{logit}(p(x_i)) = \beta_0 + \beta_1 i, 1 + \cdots + \beta_{D-1} x_{i,D-1}$$

Où le logit est la fonction définie par $\text{logit}(x) = \ln\left(\frac{x}{1-x}\right)$.

- Si $p(x_i) > 0.5$: on va prévoir $\hat{y}_i = 1$.
- Si $p(x_i) \leq 0.5$, on va donc prévoir $\hat{y}_i = 0$.

Conclusion

Conclusion

- La connaissance des outils de statistiques est indispensable au Data Scientist. Faire une description empirique des données ainsi que des visualisations adéquates est primordial avant de commencer à faire des modèles.
- Comparer des modèles et les comprendre permet de guider le meilleur choix en pratique.
- Les concepts de tests, pvalues sont d'une importante capitale même en machine learning pour vérifier la significativité des résultats (surtout à l'ère du Big Data).
- Le modèle de régression linéaire est le modèle à l'origine du Machine Learning. Le comprendre permet aussi de comprendre des méthodes plus complexes (les réseaux de neurones typiquement).

Conclusion

- La connaissance des outils de statistiques est indispensable au Data Scientist. Faire une description empirique des données ainsi que des visualisations adéquates est primordial avant de commencer à faire des modèles.
- Comparer des modèles et les comprendre permet de guider le meilleur choix en pratique.
- Les concepts de tests, pvalues sont d'une importante capitale même en machine learning pour vérifier la significativité des résultats (surtout à l'ère du Big Data).
- Le modèle de régression linéaire est le modèle à l'origine du Machine Learning. Le comprendre permet aussi de comprendre des méthodes plus complexes (les réseaux de neurones typiquement).

Conclusion

- La connaissance des outils de statistiques est indispensable au Data Scientist. Faire une description empirique des données ainsi que des visualisations adéquates est primordial avant de commencer à faire des modèles.
- Comparer des modèles et les comprendre permet de guider le meilleur choix en pratique.
- Les concepts de tests, pvalues sont d'une importante capitale même en machine learning pour vérifier la significativité des résultats (surtout à l'ère du Big Data).
- Le modèle de régression linéaire est le modèle à l'origine du Machine Learning. Le comprendre permet aussi de comprendre des méthodes plus complexes (les réseaux de neurones typiquement).

Conclusion

- La connaissance des outils de statistiques est indispensable au Data Scientist. Faire une description empirique des données ainsi que des visualisations adéquates est primordial avant de commencer à faire des modèles.
- Comparer des modèles et les comprendre permet de guider le meilleur choix en pratique.
- Les concepts de tests, pvalues sont d'une importante capitale même en machine learning pour vérifier la significativité des résultats (surtout à l'ère du Big Data).
- Le modèle de régression linéaire est le modèle à l'origine du Machine Learning. Le comprendre permet aussi de comprendre des méthodes plus complexes (les réseaux de neurones typiquement).

Merci pour votre attention !