



Séance 9

20 octobre 2022

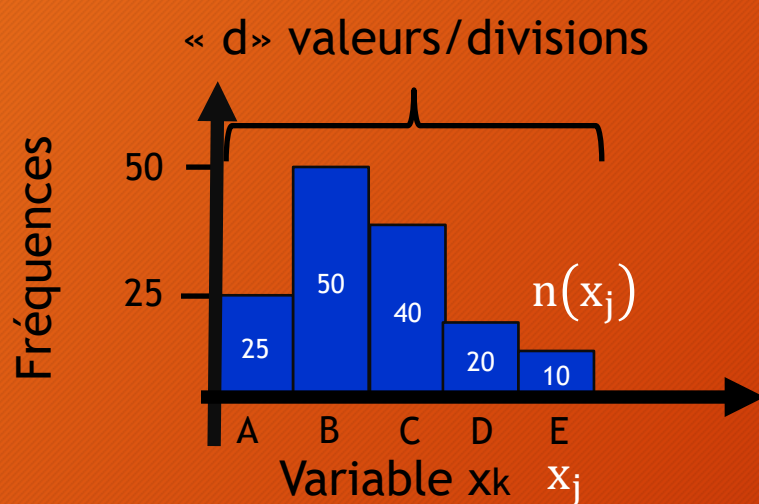
Pierre-Marc Juneau

Plan

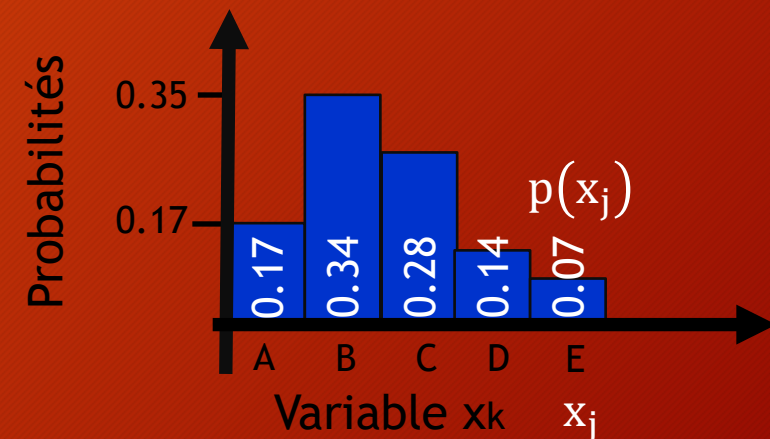
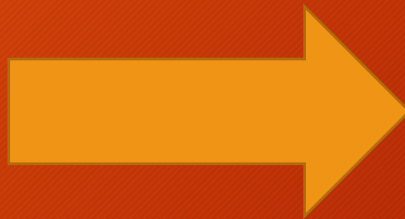
1. Distribution de fréquence versus distribution de probabilités
2. Théorème de Bayes
3. Loi normale
4. Théorème central limite

1- Distribution de fréquence versus probabilité

- La distribution de fréquence permet de voir le nombre d'instances pour les variables à valeurs discrètes ou pour chaque catégorie
- Cependant il est souvent plus utile, pour s'affranchir de la taille de l'échantillon, de travailler avec une fonction de probabilité
- Exemple: pour des variables catégoriques

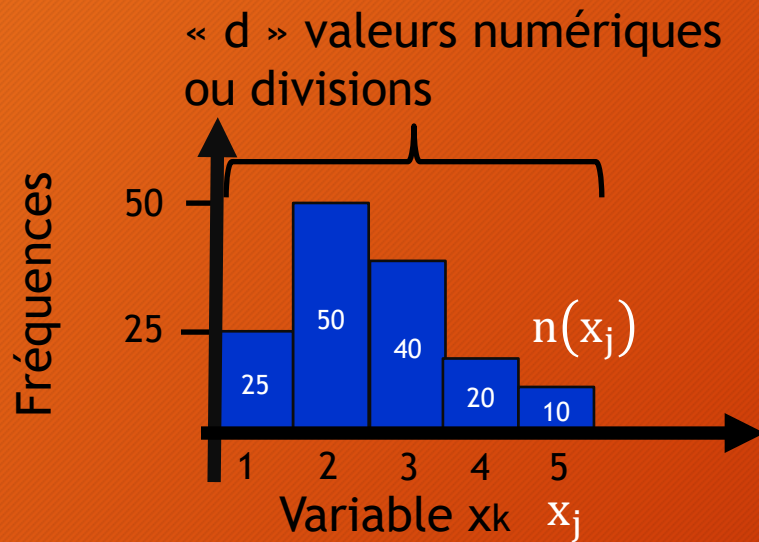


$$p(x_j) = \frac{n(x_j)}{\sum_{j=1}^d n(x_j)}$$

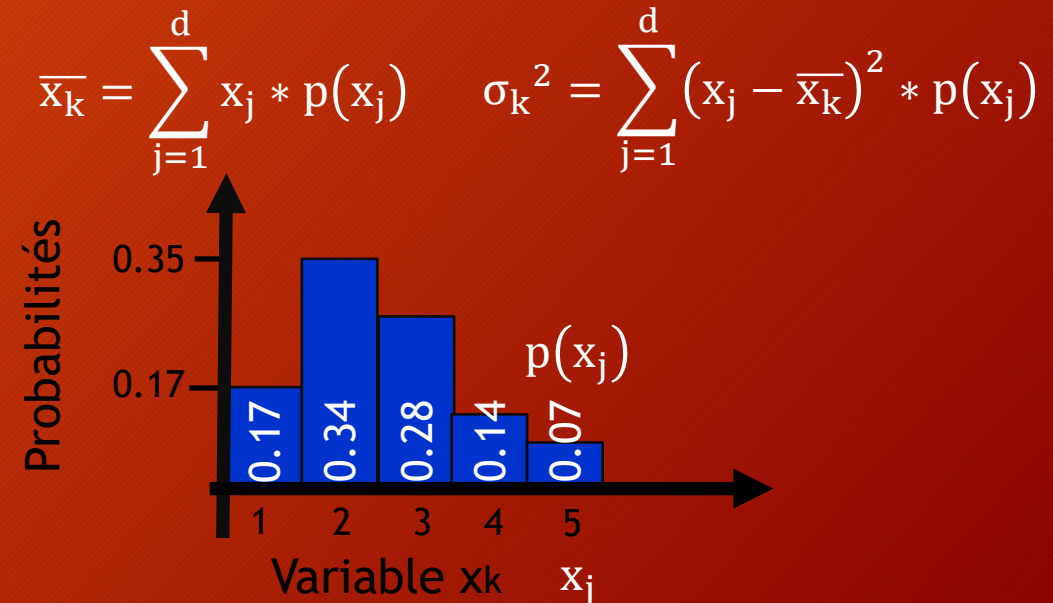
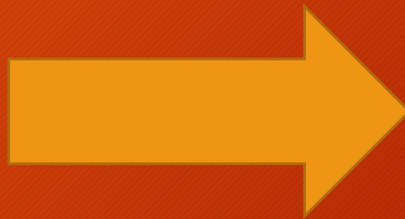


1- Distribution de fréquence versus probabilité

- Pour des variables numériques discrètes, c'est le même principe
- Cependant, avec des variables numériques discrètes, il est alors possible de calculer les moments des distributions (ex: la moyenne et la variance)



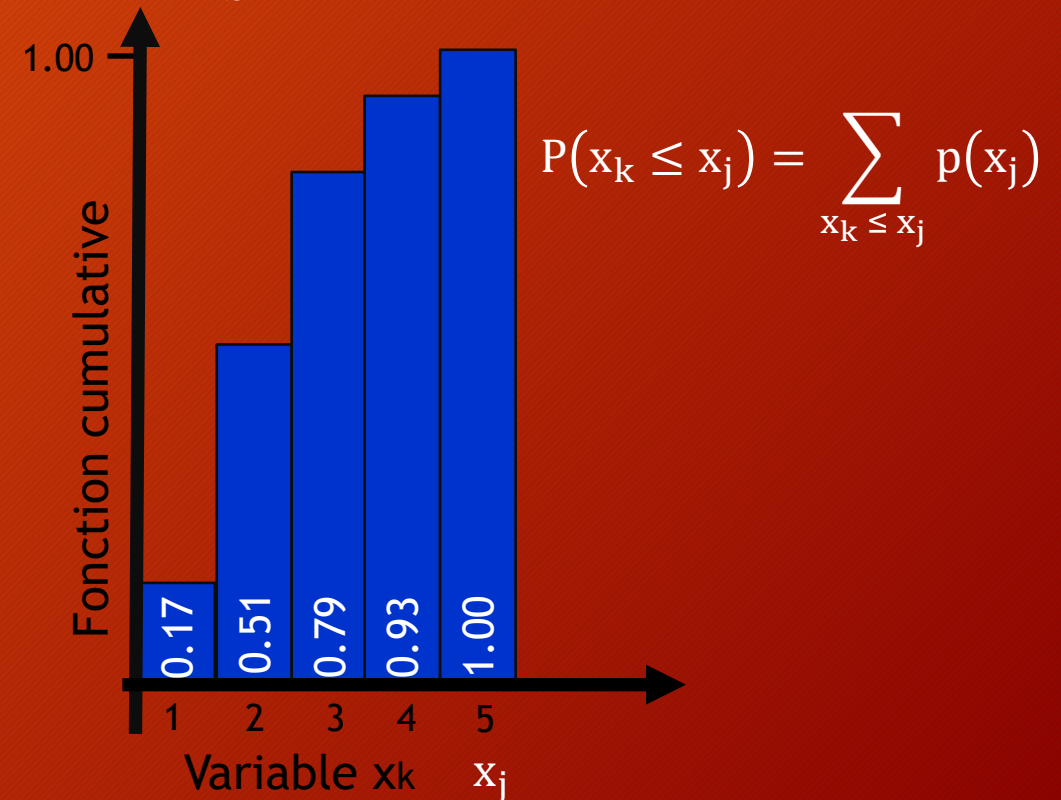
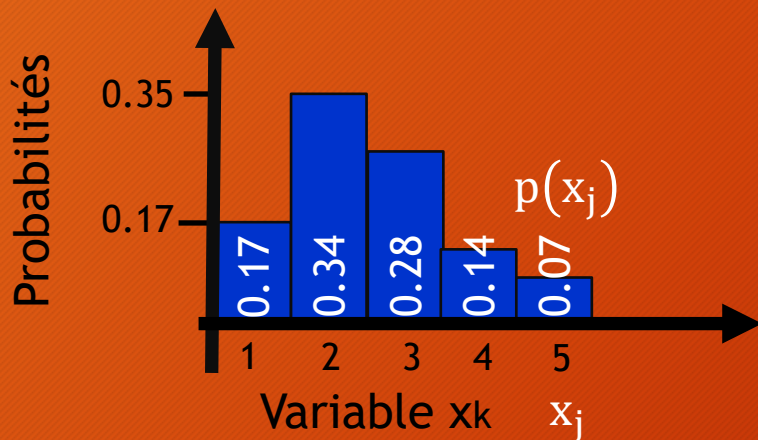
$$p(x_j) = \frac{n(x_j)}{\sum_{j=1}^d n(x_j)}$$



1- Distribution de fréquence versus probabilité

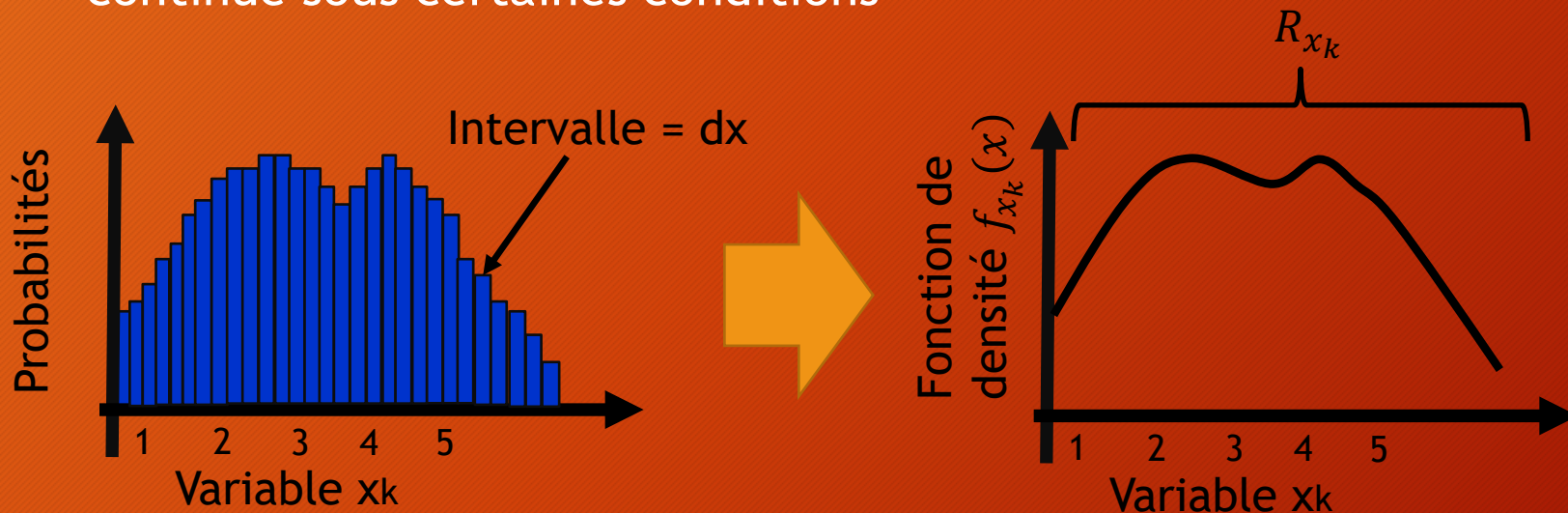
- Fonction cumulative de probabilité (cdf pour « cumulative density function »)
 - La fonction cumulative pour une fonction de probabilité (pour des variables numériques) peut-être définie comme suit:

$$\sum_{j=1}^d p(x_j) = 1$$



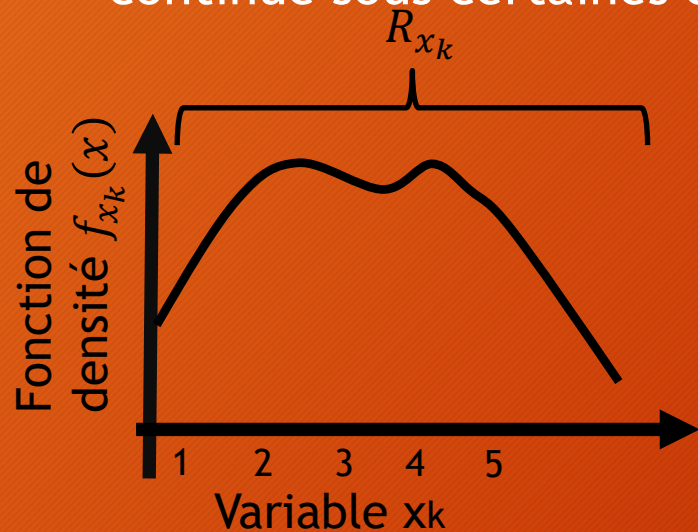
1- Distribution de fréquence versus probabilité

- Fonction de densité de probabilité (pdf pour « probability density function »)
 - Pour les variables numériques continues (avec des intervalles infinitésimaux), on parle de fonction de densité de probabilité, qui peut être considérée comme continue sous certaines conditions



1- Distribution de fréquence versus probabilité

- Fonction de densité de probabilité (pdf pour « probability density function »)
 - Pour les variables numériques continues (avec des intervalles infinitésimaux), on parle de fonction de densité de probabilité, qui peut être considérée comme continue sous certaines conditions



Conditions pour fonction de densité

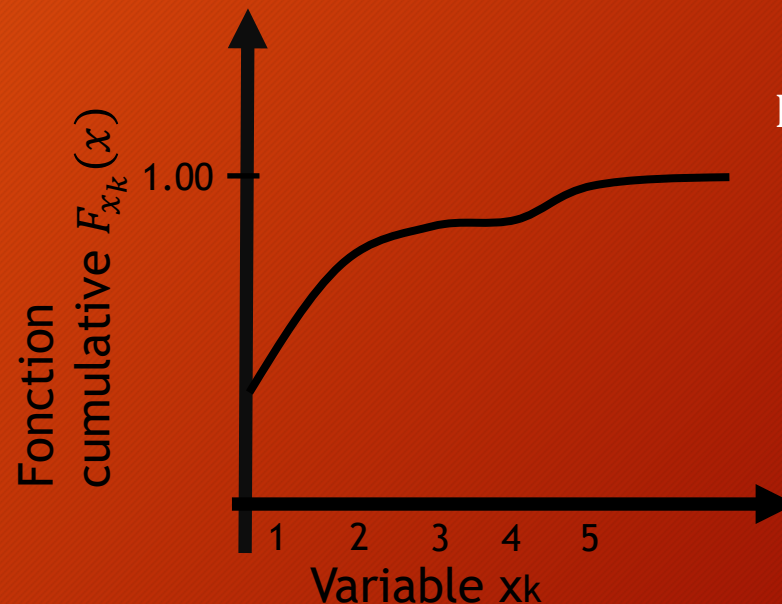
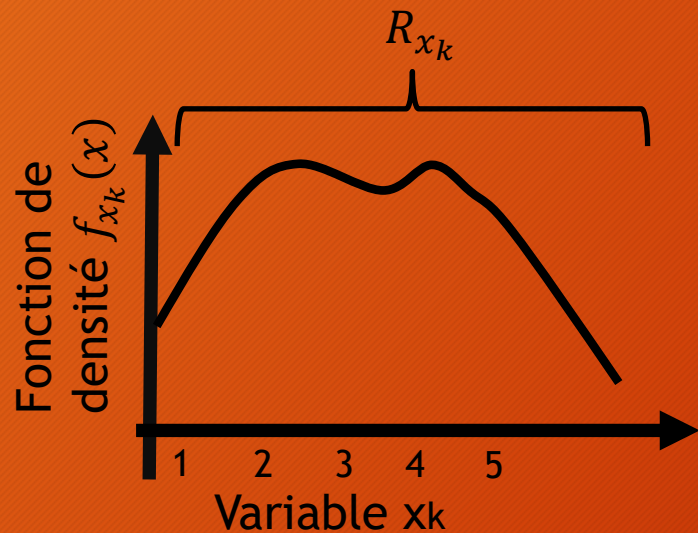
$$\int_{R_{x_k}} f_{x_k}(x) = 1 \quad f_{x_k}(x) \geq 0 \text{ pour tout } x \text{ dans } R_{x_k}$$

Fonction continue par morceaux

Fonction nulle pour les valeurs de x en – dehors de R_{x_k}

1- Distribution de fréquence versus probabilité

- Fonction cumulative de probabilité (cdf pour « cumulative density function »)
 - Cumule les probabilités pour une variable numérique continue



$$F_{x_k}(x) = \int_{-\infty}^x f_{x_k}(x) dx$$

1- Distribution de fréquence versus probabilité

- E09-1 - Jeu de données (adapté) sur les fumeurs dans un milieu de travail au Royaume-Unis (population adulte)

Genre	Âge	Statut	Revenus	Fumeur?	Cigarettes par jour (week-end)	Cigarettes par jour (semaine)	Type
Homme	38	Divorcé	2,600 à 5,200	Non	0	0	
Femme	42	Célibataire	Sous 2,600	Oui	12	12	Paquets
Homme	40	Marrié	28,600 à 36,400	Non	0	0	
Femme	40	Marrié	10,400 à 15,600	Non	0	0	
Femme	39	Marrié	2,600 à 5,200	Non	0	0	
Femme	37	Marrié	15,600 à 20,800	Non	0	0	
Homme	53	Marrié	Au-dessus 36,400	Oui	6	6	Paquets
Homme	44	Célibataire	10,400 à 15,600	Non	0	0	
Homme	40	Célibataire	2,600 à 5,200	Oui	8	8	Roulées à la n
Femme	41	Marrié	5,200 à 10,400	Oui	15	12	Paquets
Homme	72	Veuf	10,400 à 15,600	Non	0	0	



Cette photo par Auteur inconnu est soumise à la licence [CC BY](#)

<https://www.openintro.org/data/index.php?data=smoking>

1- Distribution de fréquence versus probabilité

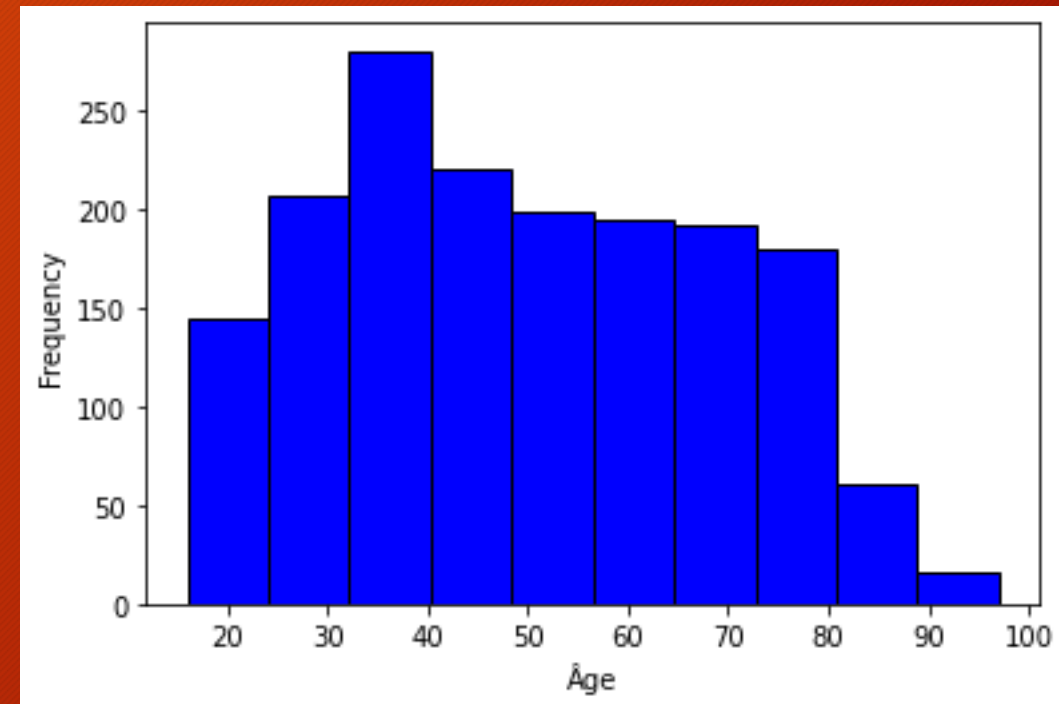
- E09-1 - Jeu de données (adapté) sur les fumeurs dans un milieu de travail au Royaume-Unis (population adulte)
 - Téléchargement des librairies et des données
 - *import numpy as np*
 - *import pandas as pd*
 - *import matplotlib.pyplot as plt*
 - *from sklearn.neighbors import KernelDensity*
 - *donnee = pd.read_csv('DonneesFumeursv0r2.csv')*
 - *stats=donnee.describe()*
 - *dimensions=donnee.shape*
 - *nomsvariables = pd.DataFrame(donnee.columns)*

donnee - DataFrame								
Index	Genre	Âge	Statut	Revenus	Fumeur?	par jour (par jour	Type
0	Homme	38	Divorcé	2,600 à 5,200	Non	0	0	nan
1	Femme	42	Célibataire	Sous 2,600	Oui	12	12	Paquets
2	Homme	40	Marrié	28,600 à 36,400	Non	0	0	nan
3	Femme	40	Marrié	10,400 à 15,600	Non	0	0	nan
4	Femme	39	Marrié	2,600 à 5,200	Non	0	0	nan
5	Femme	37	Marrié	15,600 à 20,800	Non	0	0	nan
6	Homme	53	Marrié	Au-dessus 36,400	Oui	6	6	Paquets
7	Homme	44	Célibataire	10,400 à 15,600	Non	0	0	nan
8	Homme	40	Célibataire	2,600 à 5,200	Oui	8	8	Roulées à la main
9	Femme	41	Marrié	5,200 à 10,400	Oui	15	12	Paquets
10	Homme	72	Veuf	10,400 à 15,600	Non	0	0	nan
11	Homme	49	Marrié	nan	Non	0	0	nan

1- Distribution de fréquence versus probabilité

- E09-1 - Jeu de données (adapté) sur les fumeurs dans un milieu de travail au Royaume-Unis (population adulte)
 - On s'intéresse particulièrement à la distribution d'âges
 - Histogramme des fréquences
 - *ax=donnee["Âge"].plot.hist(density=False, bins = 10, color = 'blue', edgecolor = 'black')*
 - *ax.set_xlabel("Âge")*

Distribution des fréquences

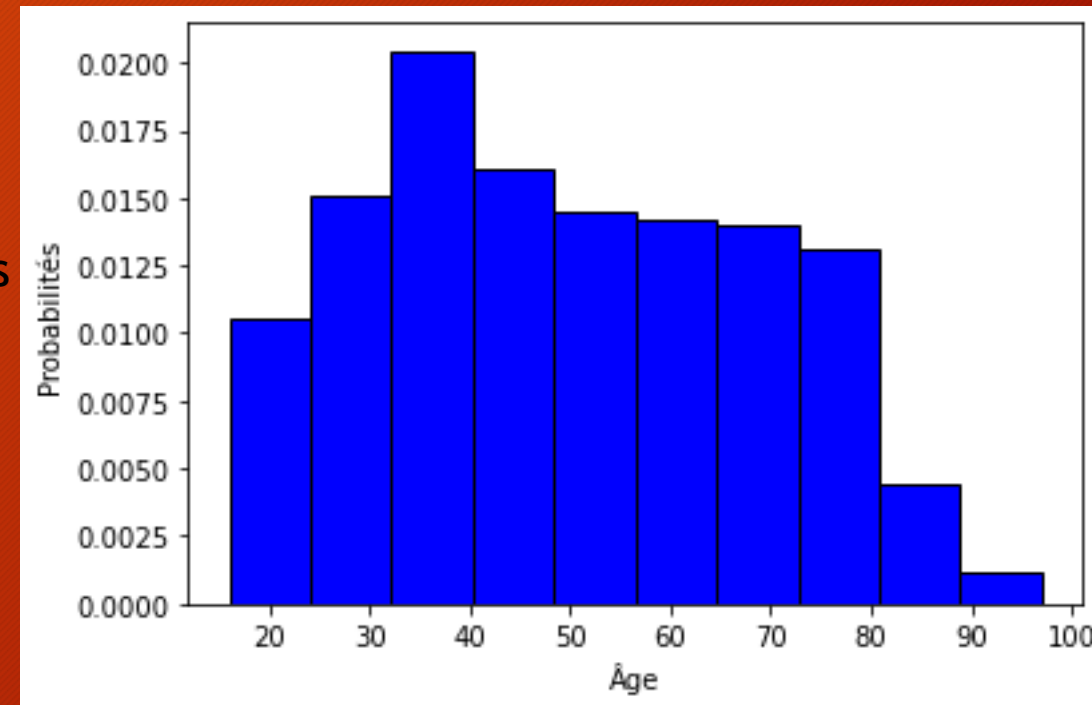


1- Distribution de fréquence versus probabilité

- E09-1 - Jeu de données (adapté) sur les fumeurs dans un milieu de travail au Royaume-Unis (population adulte)
 - On s'intéresse particulièrement à la distribution d'âges
 - Histogramme des fréquences

Distribution des probabilités

- `ax=donnee["Âge"].plot.hist(density=True, bins = 10, color = 'blue', edgecolor = 'black')`
- `ax.set_xlabel("Âge")`
- `ax.set_ylabel("Probabilités")`



1- Distribution de fréquence versus probabilité

- E09-1 - Jeu de données (adapté) sur les fumeurs dans un milieu de travail au Royaume-Unis (population adulte)
 - Pour une distribution de probabilités quelconque, la librairie `sklearn.neighbors` (KernelDensity) permet de « fitter » une fonction afin d'approximer le plus possible une la fonction de densité $f(x)$
 - Il faut d'abord déterminer une grille avec « d » intervalles qui vont capturer l'ensemble des valeurs de la variable (ici 1000 intervalles « dx »)
 - $d=1000$
 - $x_grid = np.linspace(donnee["\hat{Age}"].min()-10, donnee["\hat{Age}"].max()+10, d)$
 - $dx=(donnee["\hat{Age}"].max()+10-(donnee["\hat{Age}"].min()-10))/(d-1)$

stats - DataFrame	
Index	Âge
count	1691
mean	49.8362
std	18.7369
min	16
25%	34
50%	48
75%	65.5
max	97

$$dx = 0.1011$$

grille_x - NumPy object arr...	
	0
0	6
1	6.1011
2	6.2022
3	6.3033
4	6.4044
5	6.50551
6	6.60661
7	6.70771
8	6.80881
9	6.90991
10	7.01101

1- Distribution de fréquence versus probabilité

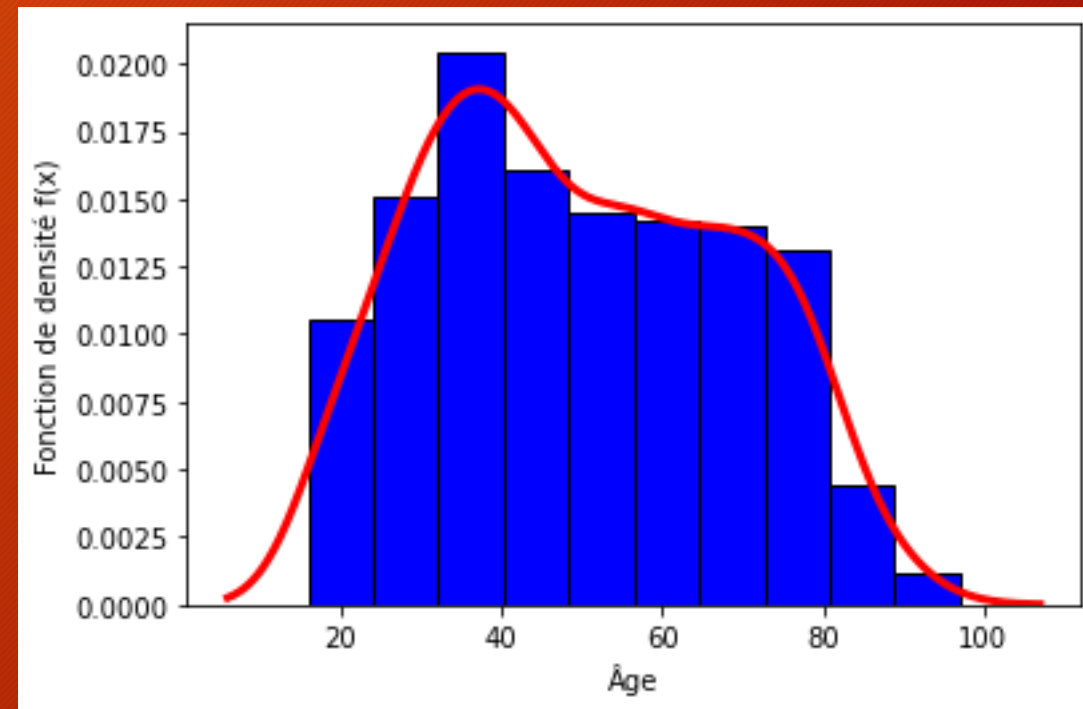
- E09-1 - Jeu de données (adapté) sur les fumeurs dans un milieu de travail au Royaume-Unis (population adulte)
 - Pour une distribution de probabilités quelconque, la librairie `sklearn.neighbors` (KernelDensity) permet de « `fit` » une fonction afin d'approximer le plus possible une la fonction de densité $f(x)$
 - Par la suite, il est possible de « `fit` » une fonction (ex: de type gaussienne), selon différentes résolutions (« `bandwidth` »)
 - `Age_valeurs=donnee["Âge"].values.reshape(-1, 1)`
 - `kde = KernelDensity(kernel='gaussian', bandwidth=5).fit(Age_valeurs)`
 - `pdf = np.exp(kde.score_samples(grille_x.reshape(-1, 1)))`

pdf = estimés de la fonction de densité pour chaque point dans grille_x

grille_x - NumPy object arr...			pdf - NumPy object arra	
	0			0
0	6	→	0	0.000254651
1	6.1011	→	1	0.000266699
2	6.2022	→	2	0.000279217
3	6.3033	→	3	0.000292217
4	6.4044	→	4	0.000305713
5	6.50551	→	5	0.000319719

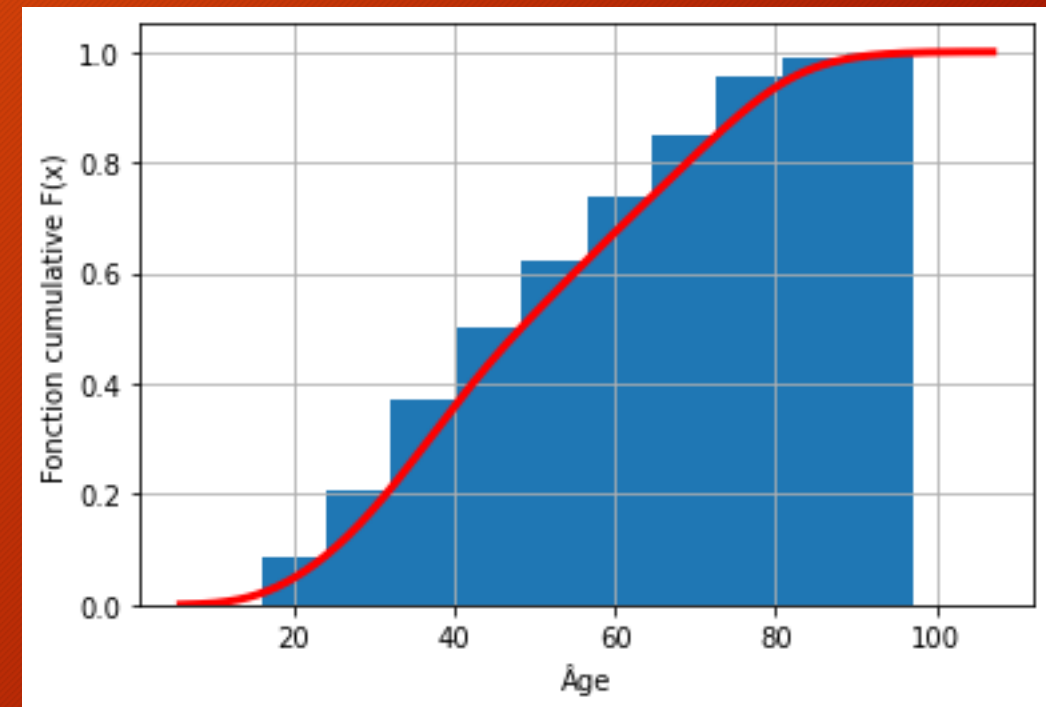
1- Distribution de fréquence versus probabilité

- E09-1 - Jeu de données (adapté) sur les fumeurs dans un milieu de travail au Royaume-Unis (population adulte)
 - L'histogramme des données et la fonction de densité estimée peuvent être comparées par graphique
 - `fig, ax = plt.subplots()`
 - `ax=donnee["Âge"].plot.hist(density=True, bins = 10, color = 'blue', edgecolor = 'black')`
 - `ax.set_xlabel("Âge")`
 - `ax.plot(grille_x, pdf, linewidth=3, color = 'red')`



1- Distribution de fréquence versus probabilité

- E09-1 - Jeu de données (adapté) sur les fumeurs dans un milieu de travail au Royaume-Unis (population adulte)
 - Il est possible de comparer les fonctions cumulatives
 - `fig, ax = plt.subplots()`
 - `donnee["Âge"].hist(cumulative=True, density=1, bins=10)`
 - `cdf = np.cumsum(pdf*dx)`
 - `ax.plot(grille_x, cdf, linewidth=3, color = 'red')`
 - `ax.set_xlabel("Âge")`
 - `ax.set_ylabel("Fonction cumulative F(x)")`



1- Distribution de fréquence versus probabilité

- E09-1 - Jeu de données (adapté) sur les fumeurs dans un milieu de travail au Royaume-Unis (population adulte)
 - Calculer la probabilité que les adultes aient 40 ans et moins:
 - `grille_x_pd=pd.DataFrame(grille_x)`
 - `grille_x_pd.columns=["Âges"]`
 - `pdf_pd=pd.DataFrame(pdf)`
 - `pdf_pd.columns=["PDF"]`
 - `cdf_pd=pd.DataFrame(cdf)`
 - `cdf_pd.columns=["CDF"]`
 - `DistAge=pd.concat([grille_x_pd,pdf_pd,cdf_pd],axis=1)`
 - `Prob_moins_40=DistAge["PDF"][DistAge["Âges"]<=40].sum()*dx`

DistAge - DataFrame			
Index	Âges	PDF	CDF
333	39.6667	0.0187735	0.351176
334	39.7678	0.0187503	0.353071
335	39.8689	0.0187263	0.354964
336	39.97	0.0187015	0.356855
337	40.0711	0.0186758	0.358743
338	40.1722	0.0186493	0.360629
339	40.2733	0.0186221	0.362512
340	40.3744	0.0185941	0.364391
341	40.4755	0.0185653	0.366268
342	40.5766	0.0185358	0.368142

2- Théorème de Bayes

- Permet de calculer des probabilités conditionnelles
- Le théorème de Bayes est utilisé en inférence statistique pour estimer par exemple une probabilité à partir des observations (distributions des variables) ou des lois de probabilité associées à ces observations

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

A étant donné B

B étant donné A

A: Évènement ou condition A
B: Évènement ou condition B

2- Théorème de Bayes

- E09-2 - Jeu de données (adapté) sur les fumeurs dans un milieu de travail au Royaume-Unis (population adulte)
- Selon les données, quelle est la probabilité qu'une personne adulte dans la quarantaine (de 40 à 49 ans inclusivement) fume?



[Cette photo](#) par Auteur inconnu est soumise à la licence [CC BY](#)

2- Théorème de Bayes

- E09-2 - Jeu de données (adapté) sur les fumeurs dans un milieu de travail au Royaume-Unis (population adulte)
- Vous voyez dans la salle de repos une personne dans la quarantaine, et vous vous demandez si elle fume
- Selon les données, quelle est la probabilité qu'une personne adulte dans la quarantaine (de 40 à 49 ans inclusivement) fume?
- Condition A: personne fumeur
- Condition B: $40 \leq \text{âge} \leq 49$
- On cherche: $P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$



Cette photo par Auteur inconnu est soumise à la licence [CC BY](#)

2- Théorème de Bayes

- E09-2 - Jeu de données (adapté) sur les fumeurs dans un milieu de travail au Royaume-Unis (population adulte)
- Les données sont traitées pour regrouper les observations selon certaines conditions
 - *Travailleurs_fumeurs=donnee[(donnee["Fumeur?"] == 'Oui')]*
 - *Travailleurs_dans_quarantaine = donnee[((donnee["Âge"] <= 49) & (donnee["Âge"] >= 40))]*
 - *Fumeurs_dans_quarantaine=donnee[((donnee["Fumeur?"] == 'Oui') & (donnee["Âge"] <= 49) & (donnee["Âge"] >= 40))]*

Données respectant la condition A

Données respectant la condition B

Données respectant les conditions A et B

2- Théorème de Bayes

- E09-2 - Jeu de données (adapté) sur les fumeurs dans un milieu de travail au Royaume-Unis (population adulte)
- Il est possible de calculer les proportions (probabilités) selon différentes conditions
 - $P_{\text{fumeurs}} = \text{Travailleurs_fumeurs.shape}[0] / \text{donnee.shape}[0]$
 - $P_{\text{quarantaine}} = \text{Travailleurs_dans_quarantaine.shape}[0] / \text{donnee.shape}[0]$
 - $\text{Prob_quarantaine_étant_fumeur} = \text{Fumeurs_dans_quarantaine.shape}[0] / \text{Travailleurs_fumeurs.shape}[0]$

$P(A)$: Proportion (probabilité) de fumeurs dans la population adulte de travailleurs

$P(B)$: Proportion (probabilité) de personnes dans la quarantaine dans la population adulte de travailleurs

$P(B|A)$: Proportion (probabilité) de personnes dans la quarantaine (B) chez les fumeurs (A)

2- Théorème de Bayes

- E09-2 - Jeu de données (adapté) sur les fumeurs dans un milieu de travail au Royaume-Unis (population adulte)
- Il est alors possible de calculer la probabilité qu'un adulte dans la quarantaine soit fumeur:
 - $Prob_fumeur_étant_quarantaine = Prob_quarantaine_étant_fumeur * P_fumeurs / P_quarantaine$
- Il est possible de le vérifier en le calculant directement:
 - $Prob_fumeur_étant_quarantaine_theo = Fumeurs_dans_quarantaine.shape[0] / Travailleurs_dans_quarantaine.shape[0]$

$$P(A) = 0.249$$

$$P(B) = 0.174$$

$$P(B|A) = 0.2209$$

$$P(A|B) = 0.3569$$

2- Théorème de Bayes

• Exercice L09 - #1

- Vous avez les données d'une étude réalisée sur les habitudes de consommation (marketing). Les données ont été adaptées de la source originale.

ID	Âge	Statut Marital	Revenus	Enfants	Adolescents	Date	Temps depuis dernier achat	Vins (\$/2sem)	Fruits (\$/2sem)
5524	63	Célibataire	58138	0	0	2012-09-04	58	635	88
2174	66	Célibataire	46344	1	1	2014-03-08	38	11	1
4141	55	Conjoint de fait	71613	0	0	2013-08-21	26	426	49
6182	36	Conjoint de fait	26646	1	0	2014-02-10	26	11	4
5324	39	Marié	58293	1	0	2014-01-19	94	173	43
7446	53	Conjoint de fait	62513	0	1	2013-09-09	16	520	42
965	49	Divorcé	55635	0	1	2012-11-13	34	235	65
6177	35	Marié	33454	1	0	2013-05-08	32	76	10
4855	46	Conjoint de fait	30351	1	0	2013-06-06	19	14	0
5899	70	Conjoint de fait	5648	1	1	2014-03-13	68	28	0



https://www.kaggle.com/rodsaldanha/arketing-campaign?select=marketing_campaign.xlsx

<https://pxhere.com/fr/photo/1440159>

2- Théorème de Bayes

- Exercice L09 - #1

- Vous avez les données d'une étude réalisée sur les habitudes de consommation (marketing)
 - Explorez la distribution de la variable « Âge »
 - Trouvez (calibrer) une fonction de densité pour cette variable
 - Calculez la proportion de gens qui ont moins de 33 ans (avec la fonction de densité).



<https://pxhere.com/fr/photo/1440159>

2- Théorème de Bayes

- Exercice L09 - #1

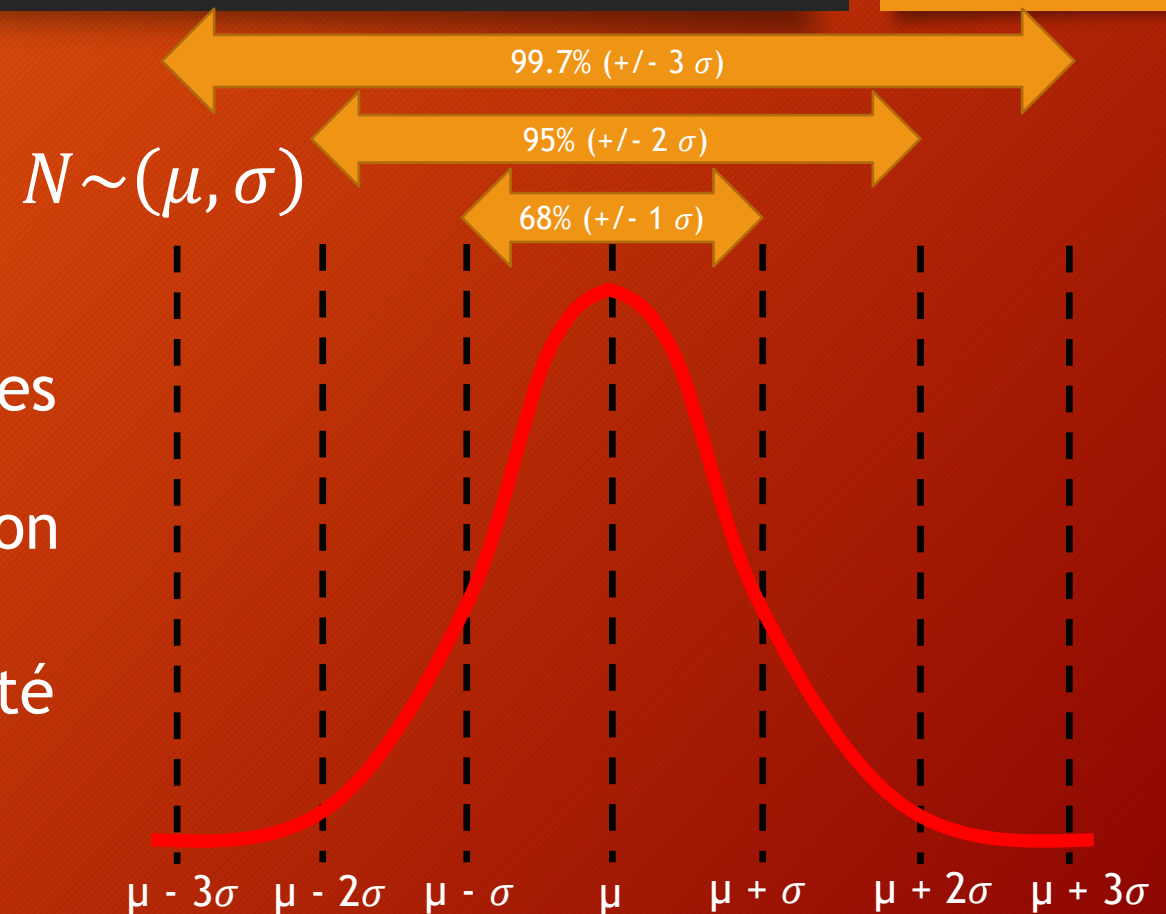
- Vous avez les données d'une étude réalisée sur les habitudes de consommation (marketing)
 - Quelle est la probabilité qu'une personne gagnant plus de 60000\$ (Condition B) achète pour plus de 100\$ en joaillerie en 2 semaines (A)?
 - Utilisez le théorème de Bayes pour calculer $P(A|B)$, puis confirmez le résultat:
 - Condition A: achète pour plus de 100\$ en joaillerie en 2 semaines
 - Condition B: gagne plus de 60000\$



<https://pxhere.com/fr/photo/1440159>

3- Loi normale

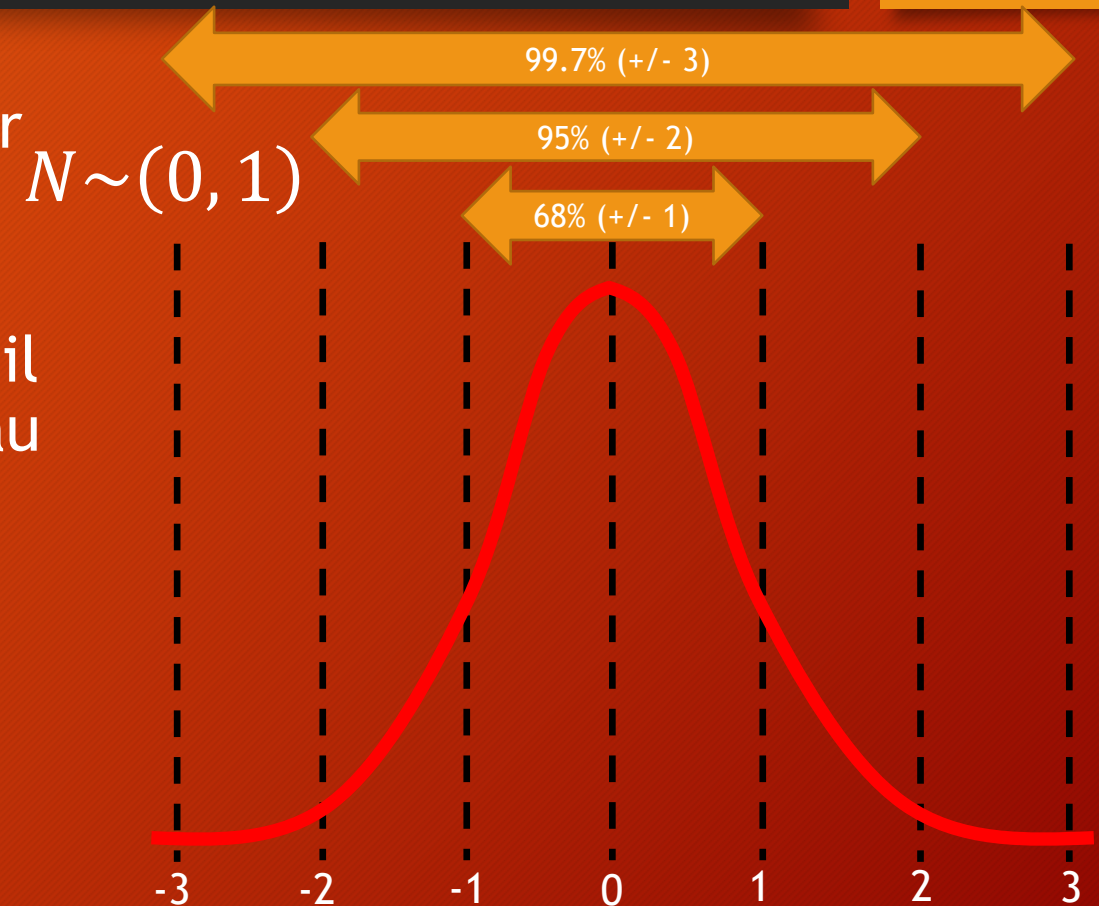
- La loi normale (ou gaussienne) est fondamentale en statistique
- Elle est en forme de cloche symétrique
- La distribution des métriques statistiques (ex: les moyennes pour différents échantillons) est souvent une distribution normale
- Plusieurs outils en mathématique ont été développés basés sur la distribution normale



3- Loi normale

- Il est souvent plus pratique de travailler avec des valeurs centrées-réduites (mean-centered).
- Dans le cas des distributions normales, il s'agit de travailler avec la variable Z (au lieu des valeurs x originales):

$$Z_{kj} = \frac{x_{kj} - \mu_k}{\sigma_k}$$



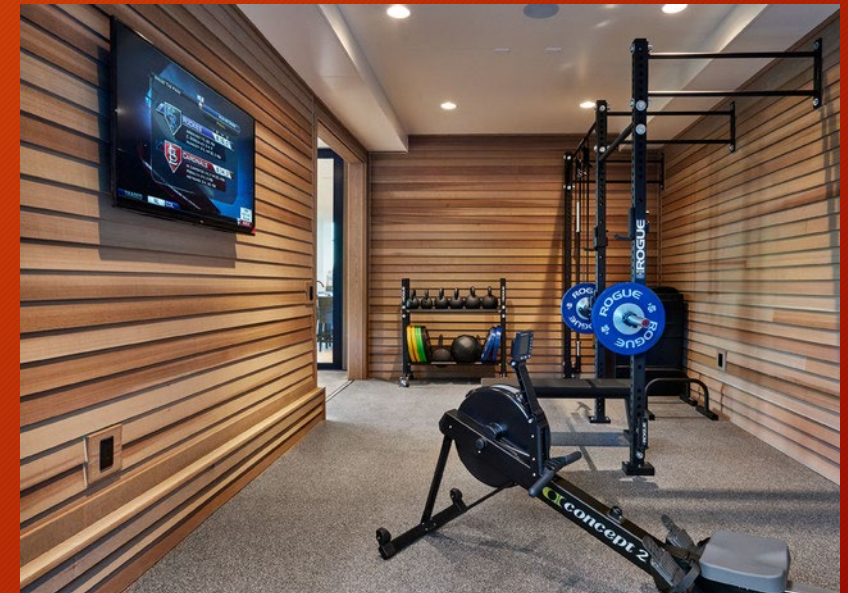
3- Loi normale

- E09-3 - Jeu de données (étude) sur les indices biométriques des personnes actives (données partielles)

Âge	Poids (kg)	Grandeur (cm)	Genre
21	65.6	174	Homme
23	71.8	175.3	Homme
28	80.7	193.5	Homme
23	72.6	186.5	Homme
22	78.8	187.2	Homme
21	74.8	181.5	Homme
26	86.4	184	Homme
27	78.4	184.5	Homme
23	62	175	Homme
21	81.6	184	Homme

- Source des données: Heinz G, Peterson LJ, Johnson RW, Kerk CJ. 2003. Exploring Relationships in Body Dimensions. Journal of Statistics Education 11(2)

<https://www.openintro.org/data/index.php?data=bdims>



Cette photo par Auteur inconnu est soumise à la licence [CC BY-SA](#)

3- Loi normale

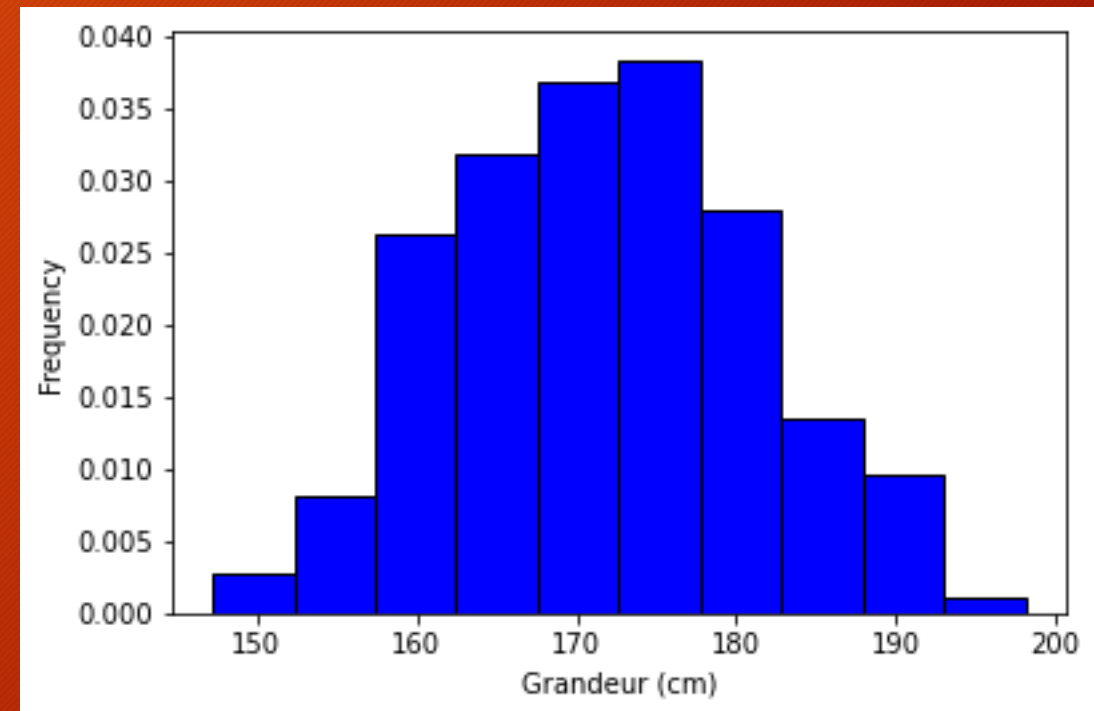
- E09-3 - Jeu de données (étude) sur les indices biométriques des personnes actives (données partielles)
 - Les données sont chargées dans Spyder avec les librairies pertinentes
 - *import numpy as np*
 - *import pandas as pd*
 - *import matplotlib.pyplot as plt*
 - *import scipy.stats as sts*
 - *import statsmodels.api as stm*
 - *donnee = pd.read_csv('PersonnesActivesv0r2.csv')*
 - *stats=donnee.describe()*
 - *dimensions=donnee.shape*
 - *nomsvariables = pd.DataFrame(donnee.columns)*

donnee - DataFrame				
Index	Âge	Poids (kg)	Grandeur (cm)	Genre
0	21	65.6	174	Homme
1	23	71.8	175.3	Homme
2	28	80.7	193.5	Homme
3	23	72.6	186.5	Homme
4	22	78.8	187.2	Homme
5	21	74.8	181.5	Homme
6	26	86.4	184	Homme
7	27	78.4	184.5	Homme
8	23	62	175	Homme
9	21	81.6	184	Homme
10	23	76.6	180	Homme
11	22	83.6	177.8	Homme

3- Loi normale

- E09-3 - Jeu de données (étude) sur les indices biométriques des personnes actives (données partielles)
 - Si on s'intéresse particulièrement à la distribution de grandeurs, la première étape est d'afficher la distribution
 - `ax=donnee["Grandeur (cm)"].plot.hist(density=True, bins = 10, color = 'blue', edgecolor = 'black')`
 - `ax.set_xlabel("Grandeur (cm)")`

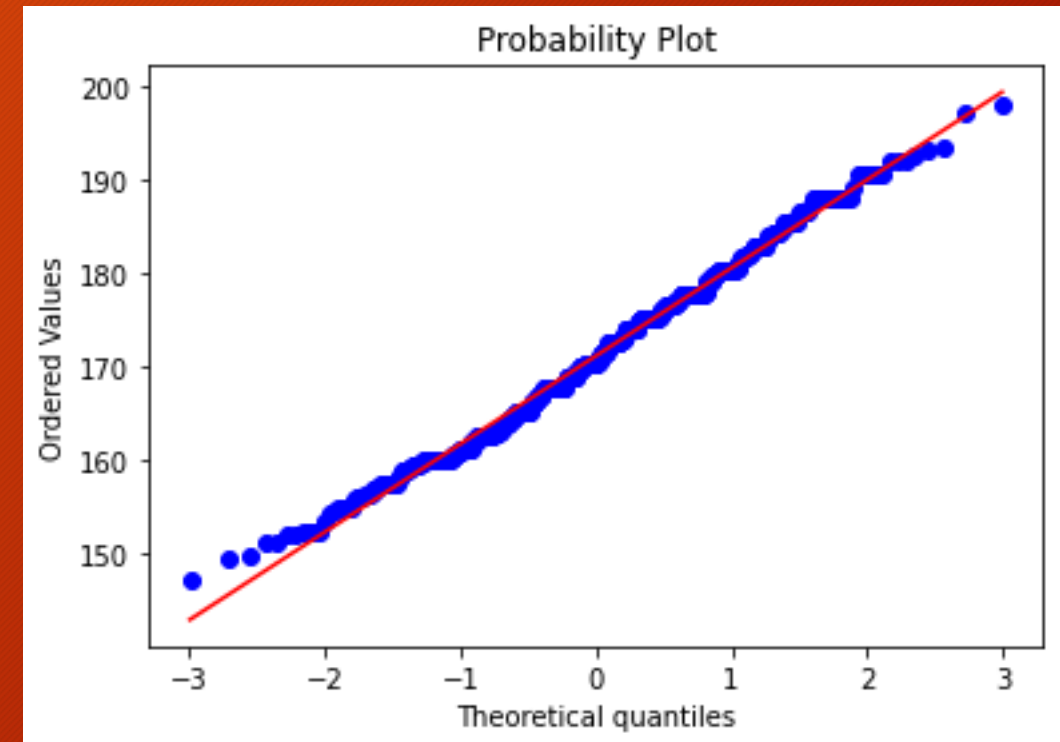
S'approche d'une distribution normale



3- Loi normale

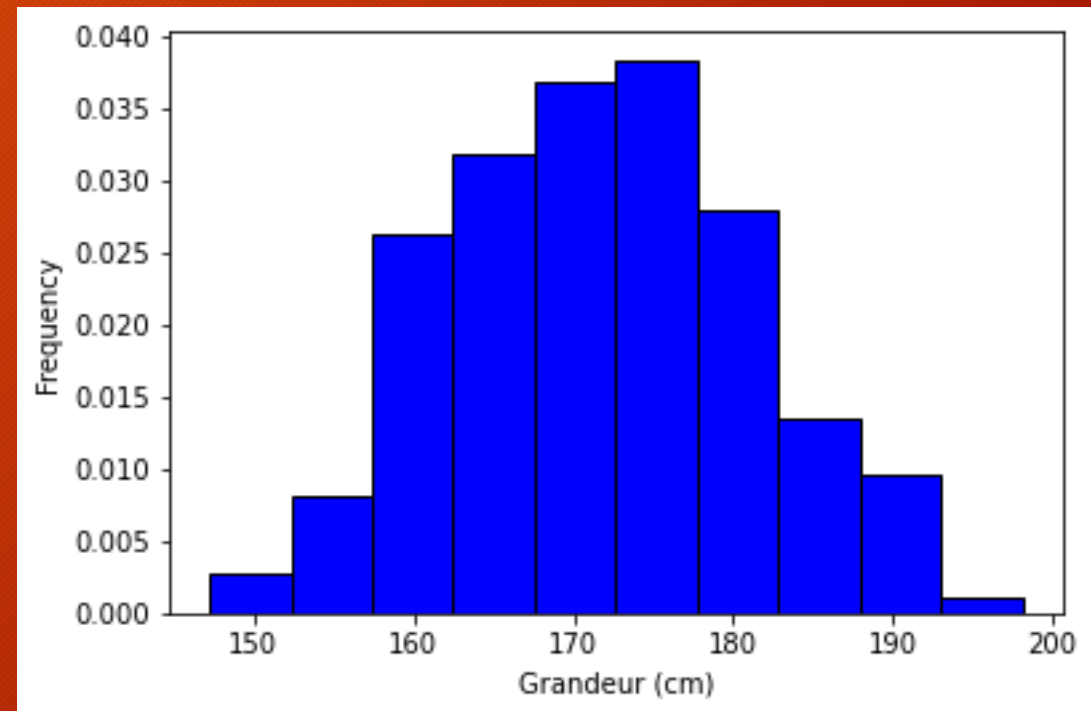
- E09-3 - Jeu de données (étude) sur les indices biométriques des personnes actives (données partielles)
 - Il est possible d'utiliser un graphique (QQ plot ou Quantile-Quantile plot) pour voir si les données sont distribuées selon une distribution précise
 - `sts.probplot(donnee["Grandeur (cm)"].values, dist=sts.norm, plot=plt.figure().add_subplot(111))`

**S'approche d'une distribution normale:
les points suivent la ligne rouge**



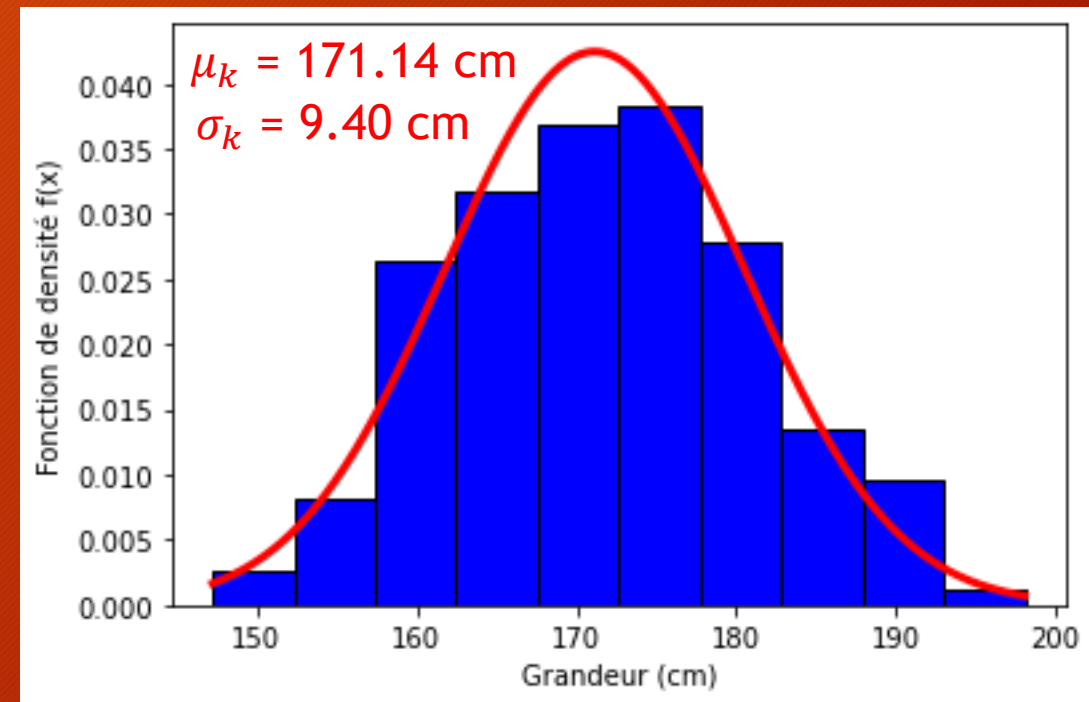
3- Loi normale

- E09-3 - Jeu de données (étude) sur les indices biométriques des personnes actives (données partielles)
 - Pour chacune des distributions (dont la loi normale), la librairie *scipy.stats* a une fonction « fit » pouvant être utilisée pour estimer les paramètres d'une distribution sur les données



3- Loi normale

- E09-3 - Jeu de données (étude) sur les indices biométriques des personnes actives (données partielles)
 - $d=1000$
 - `grille_x = np.linspace(donnee["Grandeur (cm)"].min(), donnee["Grandeur (cm)"].max(), d)`
 - `dx=(donnee["Grandeur (cm)"].max()-(donnee["Grandeur (cm)"].min()))/(d-1)`
 - `mu, sigma = sts.norm.fit(donnee["Grandeur (cm)"].values)`
 - `pdf = sts.norm.pdf(grille_x, mu, sigma)`
 - `ax=donnee["Grandeur (cm)"].plot.hist(density=True, bins = 10, color = 'blue', edgecolor = 'black')`
 - `ax.set_xlabel("Grandeur (cm)")`
 - `ax.plot(grille_x, pdf, linewidth=3, color = 'red')`
 - `ax.set_ylabel("Fonction de densité f(x)")`

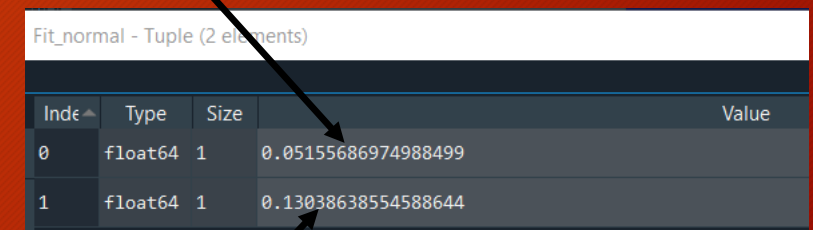


3- Loi normale

- E09-3 - Jeu de données (étude) sur les indices biométriques des personnes actives (données partielles)
 - Un test statistique peut-être utilisé pour évaluer le « fit » entre une distribution et la distribution réelle des données
 - Le test de Kolmogorov-Smirnov évalue la distance entre les deux fonctions cumulatives (« cdf »), et évalue la probabilité que 2 distributions soient les mêmes (selon la distance entre les 2 fonctions)
 - `Fit_normal = sts.kstest(donnee["Grandeur (cm)"], 'norm', [mu, sigma])`

Attention: utiliser ce test seulement pour comparer de façon relative des distributions (ex: pour comparer et voir quelle distribution a un meilleur « fit »). Avec une distribution calibrée, le test statistique en soi n'est plus valide.

Statistique de distance (D)



Index	Type	Size	Value
0	float64	1	0.05155686974988499
1	float64	1	0.13038638554588644

p-value: plus cette valeur est élevée, moins c'est probable qu'il y a une différence entre les 2 distributions (confirme l'hypothèse « nulle » que les deux distributions sont pareilles)

3- Loi normale

- E09-3 - Jeu de données (étude) sur les indices biométriques des personnes actives (données partielles)
 - Il est possible de calculer les Z-scores des variables (pour les rendre centrées-réduites)
 - `Z_scores=sts.zscore(donnee[["Âge", "Poids (kg)", "Grandeur (cm)"]].values,axis=0)`

donnee - DataFrame			
Index	Âge	Poids (kg)	Grandeur (cm)
0	21	65.6	174
1	23	71.8	175.3
2	28	80.7	193.5
3	23	72.6	186.5
4	22	78.8	187.2
5	21	74.8	181.5
6	26	86.4	184
7	27	78.4	184.5
8	23	62	175
9	21	81.6	184
10	23	76.6	180

Z_scores - NumPy object array			
	0	1	2
0	-0.956502	-0.26608	0.30392
1	-0.748147	0.198946	0.442248
2	-0.227259	0.866483	2.37885
3	-0.748147	0.258949	1.634
4	-0.852325	0.723975	1.70849
5	-0.956502	0.423958	1.10197
6	-0.435614	1.29401	1.36798
7	-0.331437	0.693973	1.42119
8	-0.748147	-0.536095	0.410326
9	-0.956502	0.933987	1.36798
10	-0.748147	0.558966	0.942359

3- Loi normale

- E09-3 - Jeu de données (étude) sur les indices biométriques des personnes actives (données partielles)

- Quelle est la probabilité qu'une personne mesure plus de 180 cm?

- `print(1-sts.norm(mu, sigma).cdf(180))`

$$P(x_k > 180) = 1 - P(x_k \leq 180) = 0.173 = 17.3\%$$

- Il est possible d'aller chercher également des valeurs précises de la fonction de densité (des probabilités à des valeurs de x précises).

- `print(sts.norm(mu, sigma).pdf(171))`

$$f_{x_k}(171) = 0.04246$$

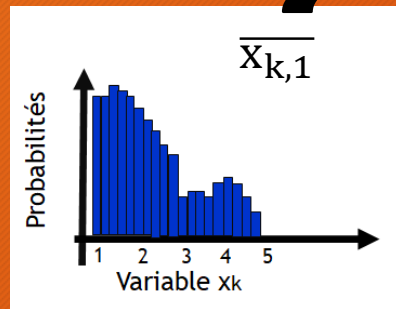
4- Théorème central limite

- Le théorème central-limite indique que les moyennes de multiples échantillons tirés d'une population seront normalement distribuées, et ce même si la distribution des données dans chaque échantillon n'est pas normale
- Ce théorème permet de faire des approximations intéressantes (ex: avec la loi de Student, vue plus tard), qui sont utilisées pour les tests d'hypothèses et le calcul d'intervalles de confiance.

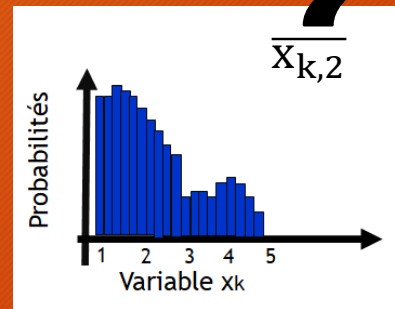


4- Théorème central limite

- Si le nombre « L » d'échantillons est suffisamment grand...

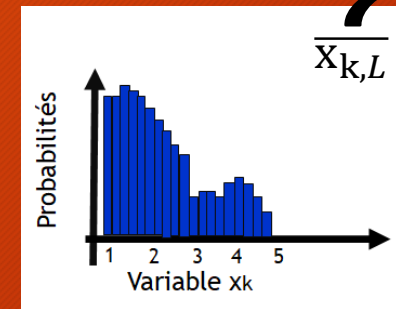


Échantillon 1

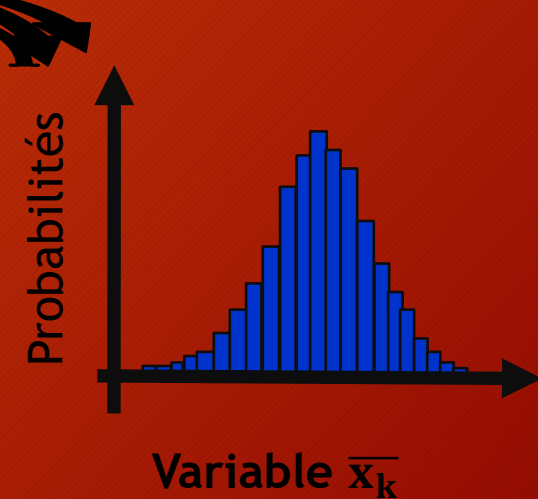


Échantillon 2

...



Échantillon L



4- Théorème central limite

- E09-4 - Jeu de données (adapté) sur les fumeurs dans un milieu de travail au Royaume-Unis (population adulte)
- Quelle est la distribution des moyennes des âges, si nous avons 1000 échantillons de 100 individus, tirés au hasard dans le jeu de données?

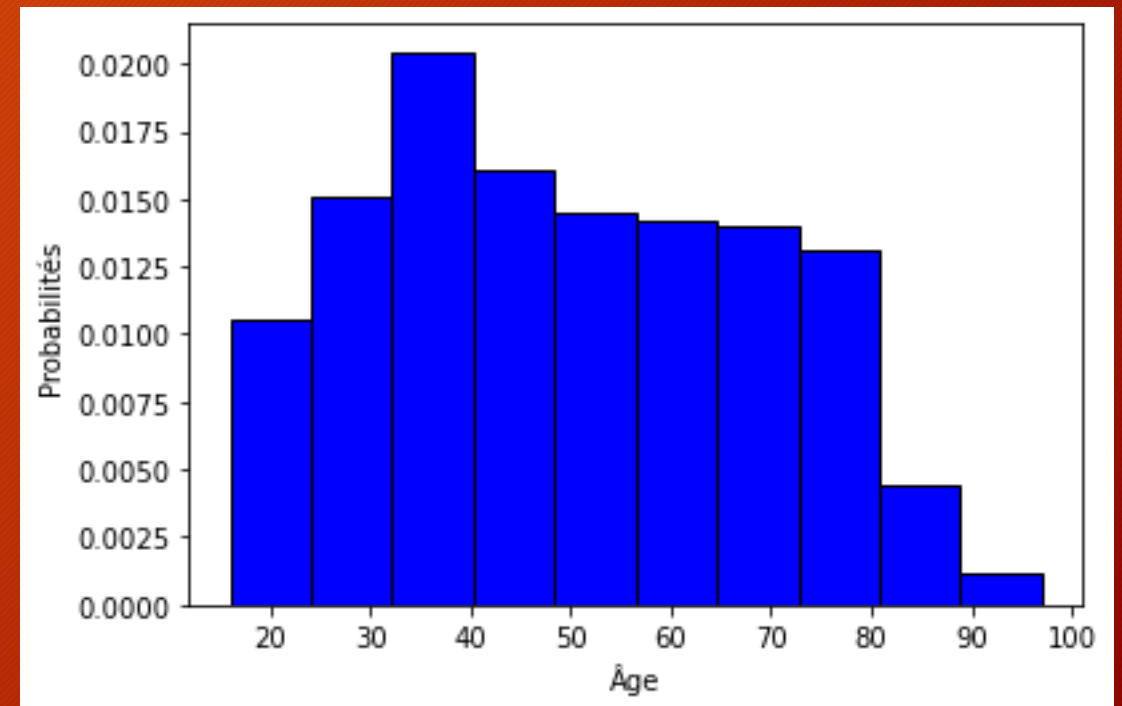


Cette photo par Auteur inconnu est soumise à la licence [CC BY](#)

4- Théorème central limite

- E09-4 - Jeu de données (adapté) sur les fumeurs dans un milieu de travail au Royaume-Unis (population adulte)
 - Histogramme de la distribution des âges de l'échantillon
 - `import numpy as np`
 - `import pandas as pd`
 - `import random`
 - `import scipy.stats as sts`
 - `donnee = pd.read_csv('DonneesFumeursv0r2.csv')`
 - `stats=donnee.describe()`
 - `dimensions=donnee.shape`
 - `nomsvariables = pd.DataFrame(donnee.columns)`
 - `ax=donnee["Âge"].plot.hist(density=True, bins = 10, color = 'blue', edgecolor = 'black')`
 - `ax.set_xlabel("Âge")`
 - `ax.set_ylabel("Probabilités")`

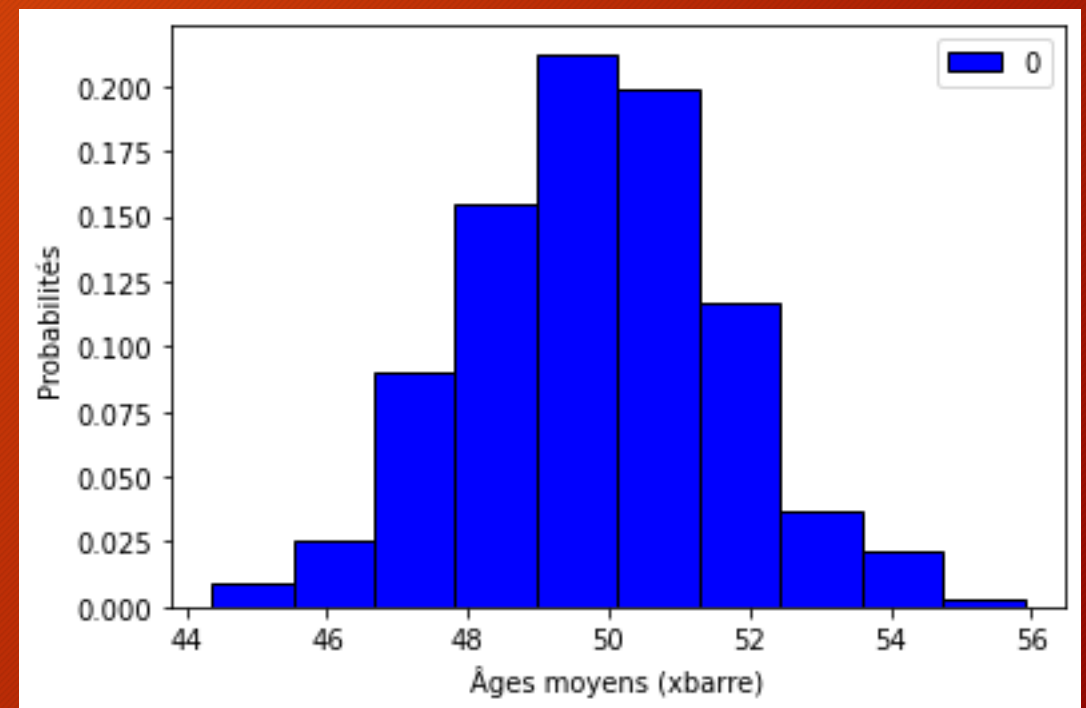
Pas très normal comme distribution



4- Théorème central limite

- E09-4 - Jeu de données (adapté) sur les fumeurs dans un milieu de travail au Royaume-Unis (population adulte)
 - Fait un tirage de 1000 échantillons contenant 100 instances (observations) chacune, et calcule la moyenne de chaque échantillon. On regarde ensuite la distribution des moyennes.
 - $L=1000$
 - $k=100$
 - `Moyennes = []`
 - `Agevar=donnee["Âge"]`
 - `for i in range(L):`
 - `Age_echantillon=random.choices(Agevar,weights=None,k=k)`
 - `Moyennes.append(np.mean(Age_echantillon))`
 - `Moyennes_Ages=pd.DataFrame(Moyennes)`
 - `ax=Moyennes_Ages.plot.hist(density=True, bins = 10, color = 'blue', edgecolor = 'black')`
 - `ax.set_xlabel("Âges moyens (xbarre)")`
 - `ax.set_ylabel("Probabilités")`

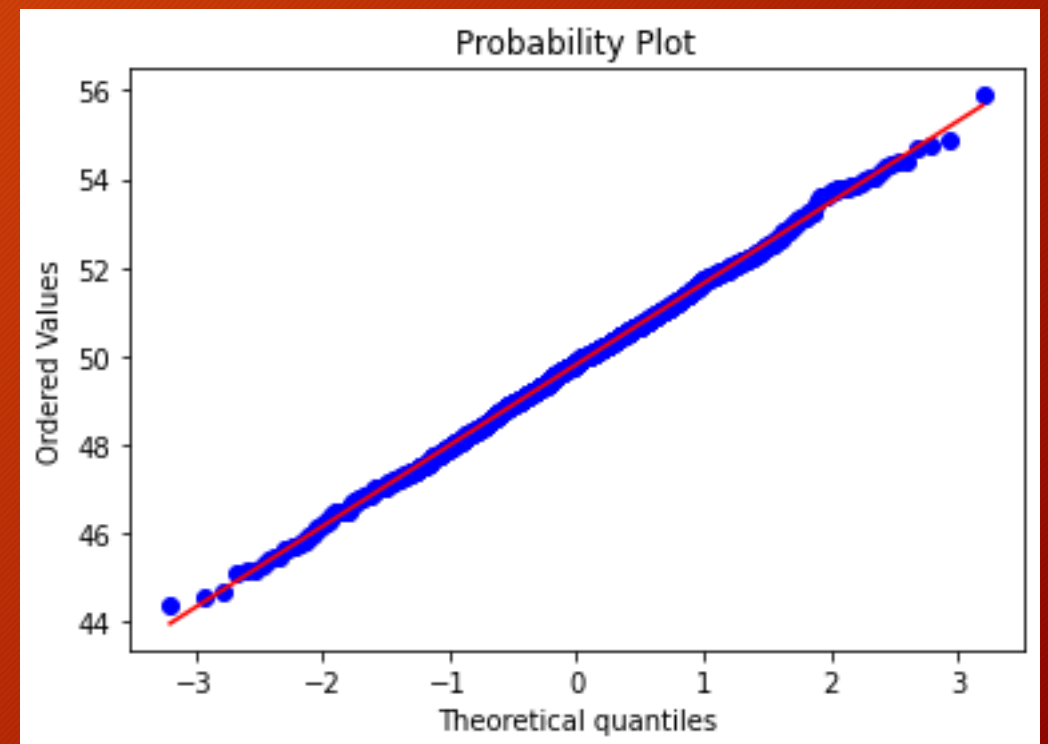
Les moyennes sont normalement distribuées



4- Théorème central limite

- E09-4 - Jeu de données (adapté) sur les fumeurs dans un milieu de travail au Royaume-Unis (population adulte)
 - Le QQ-plot confirme que les moyennes sont normalement distribuées
 - `sts.probplot(Moyennes_Ages[0].values, dist=sts.norm, plot=plt.figure().add_subplot(111))`

Les moyennes sont normalement distribuées



4- Loi normale et théorème central-limite

• Exercice L09 - #2

- Vous avez les données d'une étude réalisée sur les habitudes de consommation (marketing) (voir les données de l'exercice précédent)
- Vérifiez la distribution de la variable « Revenus ». Si jamais elle est normale, estimez les paramètres.
 - Conseil: pensez à un pré-traitement des données.



<https://pxhere.com/fr/photo/1440159>

Références

- Médiagraphie
 - Probabilités et statistiques pour ingénieurs (éd. française), Chenelière Éducation (2005), par Hines, Montgomery, Goldsman et Borror
 - Practical Statistics for Data Scientists: 50 Essential Concepts, May 28th 2017 by Peter Bruce (Author), Andrew Bruce (Author).
- Sites web
 - <https://jakevdp.github.io/blog/2013/12/01/kernel-density-estimation/>
 - <https://docs.scipy.org/doc/scipy/reference/stats.html>
 - <https://www.kite.com/python/answers/how-to-fit-a-distribution-to-a-histogram-in-python>