TP 10 - variables à densité

1 Manipulations simples de variables à densité

Exercice 1 (*Fonction de répartition de la loi normale*)

1. Que retourne la commande : `[p,q] = cdfnor("PQ",2,0,1)` ?
2. Que retourne la commande : `x = cdfnor("X",0,1,.975,.025)` ?
3. Représenter graphiquement et interpréter le résultat des instructions suivantes :

```
x = linspace(-5,5)
[p,q] = cdfnor("PQ",x,zeros(x),ones(x))
```

4. Vérifier, pour $N \hookrightarrow \mathcal{N}(0,1)$, et $x > 0$, l'inégalité : $\mathbb{P}(X \geqslant x) \leqslant \frac{1}{\sqrt{2\pi}\cdot x} \cdot \mathrm{e}^{-\frac{x^2}{2}}$.

Exercice 2 (*Le théorème central limite binomial*)

Vérifier empiriquement que pour n suffisamment grand, la loi binomiale $\mathcal{B}(n,p)$ est proche de la loi normale $\mathcal{N}(\mu,\sigma^2)$, avec :

- $\mu = n\cdot p$,
- $\sigma^2 = n\cdot p\cdot(1-p)$.

Exercice 3 (*Lois d'Erlang*)

1. Obtenir un échantillon suffisant de loi exponentielle $\mathcal{E}(1)$.
   Faire son histogramme, et le confronter avec la densité $x \mapsto \mathrm{e}^{-x}$.
2. Vérifier empiriquement les valeurs connues pour l'espérance et la variance.
   (Avec les commandes : `mean`, `variance` *et* `stdev`*).*
3. Vérifier empiriquement le résultat suivant :

   Proposition 1

   Si $X, Y \hookrightarrow \mathcal{E}(1)$, sont indépendantes, alors une densité de $X+Y$ sur $\mathbb{R}_+$ est : $x \mapsto x\cdot\mathrm{e}^{-x}$.
4. Par exemple en utilisant une boucle, on peut aussi vérifier empiriquement le résultat :

   Proposition 2

   Soient $X_0, \dots X_n$ telles que :
   - Les *v.a.* (X_k) sont mut$^{\text{nt}}$ indépendantes,
   - $\forall k \in [\![1,n]\!]$, on a : $X_k \hookrightarrow \mathcal{E}(1)$.

   Alors, la somme $S = \sum_{k=0}^{n} X_k$ admet sur $\mathbb{R}_+$ pour densité : $x \mapsto \frac{x^n}{n!}\cdot\mathrm{e}^{-x}$.

Exercice 4 (*Un transfert de loi*)

Qu'obtient-t-on à l'exécution du script suivant ? Interpréter.

```
N = 10^5
ech = grand(1,N,"exp",1)
histplot(100,exp(-ech))
```

2 Simulation de lois discrètes

Exercice 5 (*Simulation de la loi géométrique*)

1. Que retourne `rand()<0.5` ?
2. Dans une fonction `compteur = premierSucces(p)`, écrire une boucle pour compter le nombre d'essais nécessaires pour retourner le rang d'apparition du premier succès de probabilité `p`.
3. Dans une fonction `echGeom = echantillonGeometrique(n, p)`, écrire une deuxième boucle pour pour obtenir un échantillon de `n` valeurs de la loi géométrique.
4. Comparer les performances avec le programme `echantillonGeom.sce`

Exercice 6 (*Simulation de la loi binomiale*)

1. Écrire une fonction `nombreDeSucces(n,p)` qui simule la loi binomiale $\mathcal{B}(n,p)$.
2. Vérifier que pour n grand et $Np = \lambda$ pas trop grand, on a $\mathcal{B}(n,p) \sim \mathcal{P}(\lambda)$.

3 Rappels sur les lois usuelles

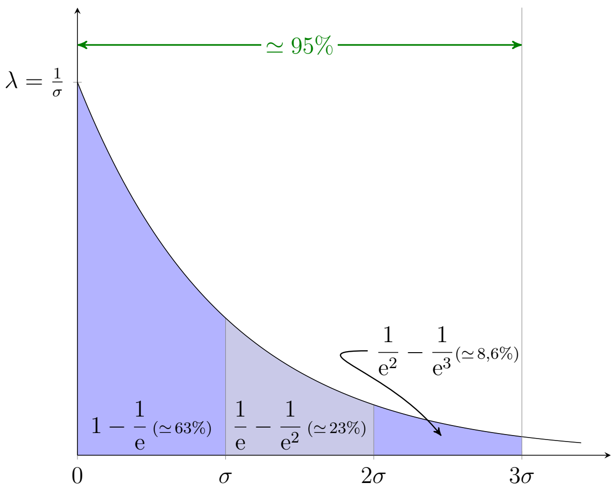
3.1 Rappels sur la loi exponentielle

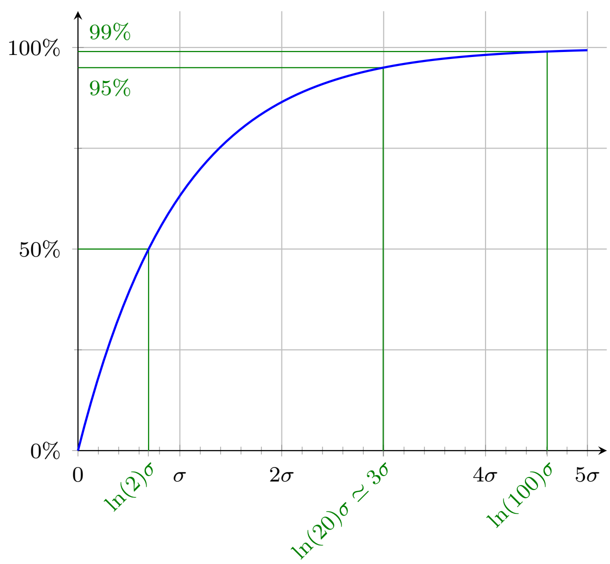
- **Densité**

$$f(x) = \begin{cases} \lambda\,\mathrm{e}^{-\lambda x}, & x \geqslant 0 \\ 0, & x < 0 \end{cases}$$

- **Fonction de répartition**

$$F(x) = \begin{cases} 1-\mathrm{e}^{-\lambda x}, & x \geqslant 0 \\ 0, & x < 0 \end{cases}$$





- **Fonction d'anti-répartition**

C'est la fonction $A(x) = \mathbb{P}(X > x)$.
Elle vérifie :

$$A(x) = 1 - F(x) = \begin{cases} \mathrm{e}^{-\lambda x}, & x \geqslant 0 \\ 1, & x < 0 \end{cases}$$

- **Moments**

(espérance)	*(variance)*	*(écart-type)*
$\mathbb{E}[X]$	$\mathrm{Var}(X)$	$\sigma(X)$
$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$	$\frac{1}{\lambda}$

3.2 Rappels sur la loi normale

- **Densité**

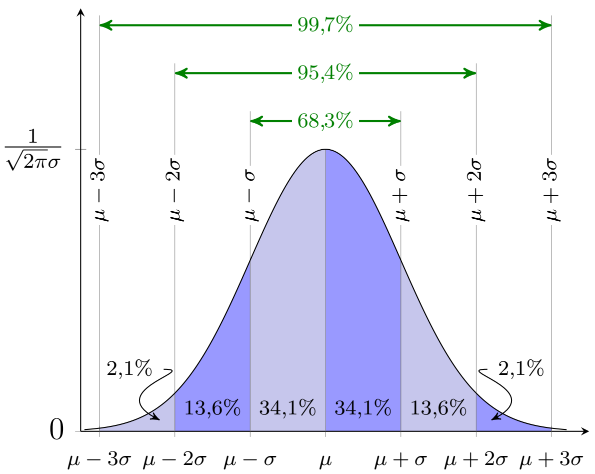
$$\forall x \in \mathbb{R},\ f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \qquad \text{pour } \mathcal{N}(0,1)$$
$$\forall x \in \mathbb{R},\ f(x) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad \text{pour } \mathcal{N}(\mu,\sigma^2)$$

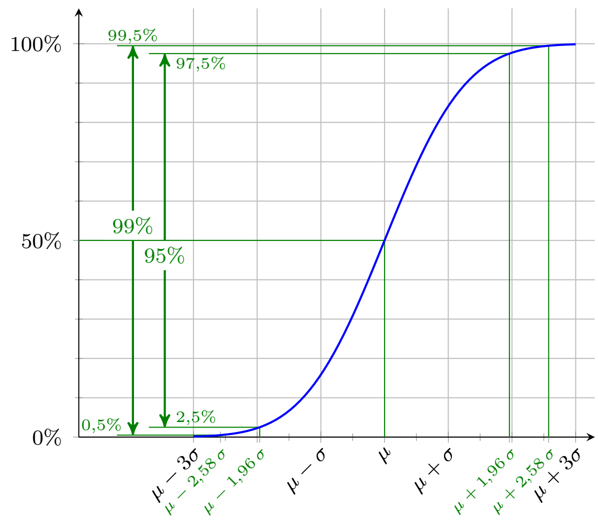
- **Fonction de répartition**

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2}}\, dt \quad \text{pour } \mathcal{N}(0,1),$$
$$F(x) = \Phi\left(\frac{x-\mu}{\sigma}\right) \qquad \text{pour } \mathcal{N}(\mu,\sigma^2).$$

- **Fonction d'anti-répartition**
  Par symétrie : $1 - \Phi(x) = \Phi(-x)$.

- **Repères sur les probabilités associées**





- **Moments**

(*espérance*) $\mathbb{E}[X]$	(*variance*) $\mathrm{Var}(X)$	(*écart-type*) $\sigma(X)$
μ	σ^2	σ

4 Quantiles d'une loi continue

4.1 Fonction de répartition

La fonction de répartition d'une distribution X est définie par : $x \mapsto F(x) = \mathbb{P}(X \leqslant x)$.
Il s'agit donc d'une représentation : valeur $\longmapsto$ probabilité
Pour X *va* continue *(p. ex : X à densité)* la fonc^{n} de répartit^{n} est « typiquement » **bijective**.
À chaque niveau de probabilité $p \in [0;1]$ est associée valeur x telle que $F(x) = p$.
Cette valeur x est appelée le **quantile** de la distribution **associé au niveau** de probabilité p.
La **bijection réciproque** de la fonct^{n} de répartit^{n} est donc **« l'application quantile »** de la distribution.

4.2 Vocabulaire des quantiles

- **Quartiles**
Il y a 5 quartiles Q_k avec $k \in [\![0,4]\!]$, correspondant aux niveaux $\frac{k}{4}$.
Pour $k \in [\![0,3]\!]$, un quart de la distribution est située entre les valeurs Q_k et Q_{k+1}.
Les quartiles extrémaux Q_0 et Q_4 correspondent respectivement au **minimum** et au **maximum** de la distribution. Le quartile Q_2 est la **médiane** de la distribution : elle sépare la distribution en « deux parties de masse égale ».

p	0%	25%	50%	75%	100%
quartile	min	Q_1	M	Q_3	max

- **Déciles**
C'est le même principe que pour les quartiles, mais en décomposant la distribution en 10 parties contenant chacune 10% de la masse.
Il y a donc 11 déciles D_k avec $k \in [\![0,10]\!]$, correspondant aux niveaux $\frac{k}{10}$.
Les déciles extrémaux D_0 et D_{10} correspondent respectivement au minimum et au maximum, et le décile D_5 à la **médiane** de la distribution.

p	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
décile	min	D_1	D_2	D_3	D_4	M	D_6	D_7	D_8	D_9	max

- **Centiles** *(ou percentiles)*
Encore le même principe, mais on subdivise cette fois la distribution en 100 parties $[C_k; C_{k+1}]$ où $k \in [\![0,99]\!]$

- **Implémentation Scilab**
Les conventions implémentées pour les commandes `quart` et `perctl` sont contradictoires :

  ⋆) *quartiles :* la distribution est subdivisée en 4 parties *(comme ci-dessus)*

  ⋆) *centiles :* la distribution est subdivisée en 99 parties *(au lieu de 100…!)*

et l'on obtient donc des aberrations liées à cette incompatibilité :

4.3 Représentation des déciles, lois normale et exponentielle

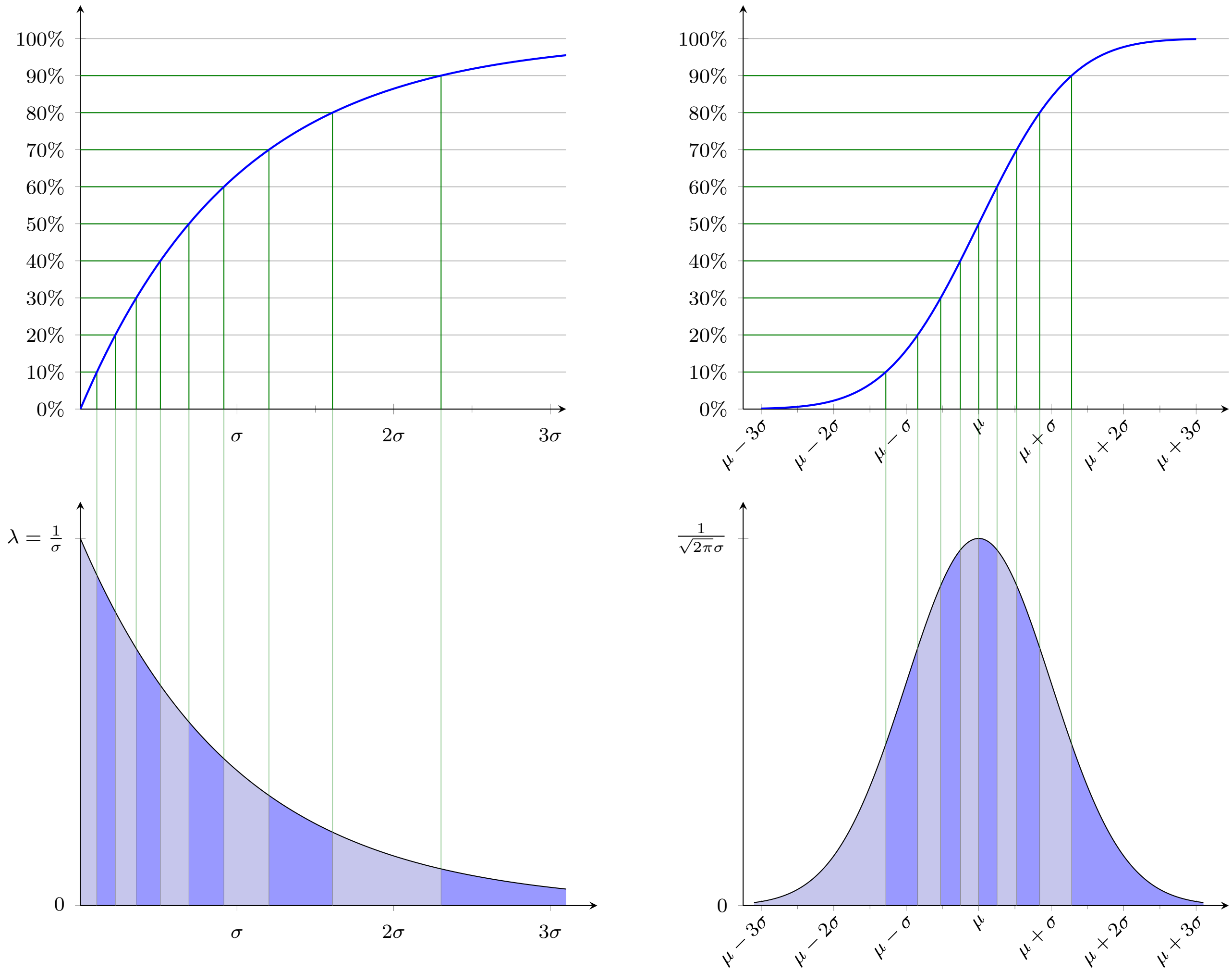
On calcule les déciles des lois exponentielles et normale par Scilab :

```
// p= 10% 20%... q = 1 - p
p = .1*(1:9)     , q = ones (p) - p
// La fonction quantiles
x = - log (q)    ,

disp (x)
```

```
// p= 10% 20%... q = 1 - p
p = .1*(1:9)     , q = ones (p) - p
// loi normale centrée réduite
Mean = zeros (p), Std = ones (p);
// On calcule les déciles
x = cdfnor("X",Mean,Std,p,q)
```

On représente les déciles sur les fonctions de répartition et sur les fonctions densité :



Soit la table de déciles suivante :

décile	D_1	D_2	D_3	D_4	M	D_6	D_7	D_8	D_9
$\mathcal{E}(\lambda)$	$0{,}10\sigma$	$0{,}22\sigma$	$0{,}35\sigma$	$0{,}51\sigma$	$0{,}69\sigma$	$0{,}91\sigma$	$1{,}20\sigma$	$1{,}60\sigma$	$2{,}30\sigma$
$\mathcal{N}(\mu,\sigma)\,(\mu\pm)$	$-1{,}28\sigma$	$-0{,}84\sigma$	$-0{,}52\sigma$	$-0{,}25\sigma$	$0\,(\mu)$	$0{,}25\sigma$	$0{,}52\sigma$	$0{,}84\sigma$	$1{,}28\sigma$

5 Simulations de lois à densité *(Tp)*

Exercice 7 (***Manipulations sur la loi exponentielle***)

1. Simuler un échantillon de 100 valeurs de loi $\mathcal{E}(1)$ en utilisant la commande `grand`
2. Obtenir sa moyenne et son écart-type avec les commandes `mean` et `stdev`.
3. Obtenir son histogramme. Est-il satisfaisant ?
4. Tracer la fonction densité au dessus de l'histogramme.
5. Vérifier empiriquement que le segment $[0;3\sigma]$ est un intervalle de fluctuation à $\simeq 95\%$.

Exercice 8 (***Avec des quantiles***)

1. Tester la commande `quart` *(les trois quartiles)* sur des progressions arithmétiques.
2. Que doit retourner la commande `quart(rand(1, 100))` ?
3. **a)** Que retourne `quart(grand(1, 100, "exp", 1))` ?
   b) Comment le résultat de `quart(grand(1, 100, "exp", sigma))` dépend-il de `sigma` ?
   c) Faire la confrontation graphique.

5.1 Simulation par la méthode d'inversion

Proposition 3 (***Méthode d'inversion pour la loi exponentielle***)

Pour une *v.a.* uniforme $U \hookrightarrow \mathcal{U}[0;1[$, la variable $X = -\ln(U)$ suit la loi exponentielle $\mathcal{E}(1)$.

Variantes

1. En général, pour simuler $\mathcal{E}(\lambda)$, on multiplie donc par $\sigma = \frac{1}{\lambda}$, soit : $-\frac{1}{\lambda}\ln(U) \hookrightarrow \mathcal{E}(\lambda)$
2. Cette formule vient de la **méthode d'inversion** pour la fonction d'**anti**-répartition.
   L'application à la fonction de répartition donne la formule : $-\frac{1}{\lambda}\ln(1-U) \hookrightarrow \mathcal{E}(\lambda)$
   (celle au programme!)

Exercice 9 (***Simulation par inversion***)

1. Faire un histogramme de la distribution `-log(rand(1, N))`.
2. Tracer la fonction densité de la loi $\mathcal{E}(1)$.
3. Confronter graphiquement les quantiles avec une loi exponentielle.

5.2 La loi normale et la commande `cdfnor`

La commande `cdfnor` implémente la fonction de répartition *(cumulated **d**istribution **f**unction)* de la loi **nor**male, soit pour la distribution $\mathcal{N}(\mu, \sigma^2)$ la formule

$$p = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{x} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] dx$$

Scilab pense à cette fonction de répartition comme à une relation entre tous les paramètres y intervenant ; soit p, x, μ, σ et la commande `cdfnor` se propose de calculer l'un quelconque de ces paramètres, les trois autres étant entrés par l'utilisateur.
On s'en servira sous les deux aspects suivants, où `Mean` *(=* μ*)* et `Std` *(=* σ*)* sont fixés.

- **Calcul de probabilités** avec la syntaxe

```
[P,Q] = cdfnor ( "PQ", X, Mean, Std )
```

  qui retourne deux réels `P,Q` représentant respectivement les valeurs respectives des fonction de répartition et d'anti-répartition *(on a donc* `Q = 1 - P`*)*.

- **Calcul de quantiles** avec la syntaxe

```
X = cdfnor ( "X", Mean, Std, P, Q )
```

  qui retourne le quantile `X` associé à la probabilité `P = 1 - Q`.