

STID 1 - Programmation Statistique
TD4
Probabilités ; simulations ; fonctions Apply

Antoine ROLLAND et Anthony SARDELLITTI

Mars 2023

Contents

1	Probabilités et simulations	2
1.1	sources	2
1.2	générateur de nombres aléatoires	2
2	Fonctions statistiques	2
2.1	Lois de probabilités discrètes	2
2.2	Lois de probabilités continues	6
2.3	Récapitulatif	9
2.4	Exercices d'entraînement	9
3	Fonctions Apply	12

1 Probabilités et simulations

1.1 sources

Certaines notions de ce TP sont extraites du cours de Christophe Chesneau “Introduction aux lois de probabilité avec R” (<https://cel.archives-ouvertes.fr/cel-01389942/document>)

1.2 générateur de nombres aléatoires

Il est possible dans R de générer automatiquement des nombres aléatoires, ou plutôt des nombres qui *ont l'air* aléatoire - ce qu'on appelle des nombres pseudo-aléatoires. Cela est possible grâce une fonction interne au comportement très chaotique : deux valeurs très proches vont donner des images très différentes, ce qui permet d'obtenir des séries de nombres qui semblent se comporter complètement au hasard. Cependant cette fonction a besoin d'être initialisée. En l'absence de paramètre, l'initialisation se fait généralement par référence à l'heure/minute/seconde où est lancé le calcul, ce qui permet de lancer deux fois la même fonction de génération au hasard sans trouver les mêmes nombres. Mais on peut aussi choisir la valeur d'initialisation (la “graine”) de l'algorithme à l'aide de la fonction `set.seed`, par exemple :

```
set.seed(1789)
```

Si dans la suite vous exécutez la fonction `set.seed` aux mêmes moments et avec la même valeur que dans l'énoncé, vous trouverez les mêmes résultats.

Pour en savoir plus sur les générateurs de nombres aléatoires (par exemple). <https://www.iro.umontreal.ca/~lecuyer/ift6561/hasard09.pdf>

2 Fonctions statistiques

On appelle fonction de répartition de X la fonction $F(x)$ donnant pour toute valeur x choisie la probabilité que la variable aléatoire X soit inférieure ou égale à x . Ce qui peut s'écrire : $F(x) = P(X \leq x)$ (où P signifie probabilité). La fonction densité $f(x)$ est la dérivée de la fonction de répartition $F(x)$.

La loi de probabilité d'une variable aléatoire est définie par : Sa fonction densité notée $f(x)$ où la surface sous la courbe est égale à 1 Sa fonction de répartition $F(x)$ qui est croissante de 0 à 1

Pour une variable aléatoire X suivant une loi notée `loi` dans R, la syntaxe générale est la suivante :

- pour obtenir la densité de X , la commande est : `dloi`
- pour obtenir la fonction de répartition de X , la commande est : `ploi`
- pour obtenir le quantile de X , la commande est : `qloi`
- pour simuler des réalisations de X suivant la même loi que X , la commande est : `rloi`

2.1 Lois de probabilités discrètes

Une variable qui ne prend qu'un nombre dénombrable de valeurs est dite discrète, sinon continue. La loi d'une variable aléatoire discrète X est la liste de toutes les valeurs différentes que peut prendre X avec les probabilités qui lui sont associées.

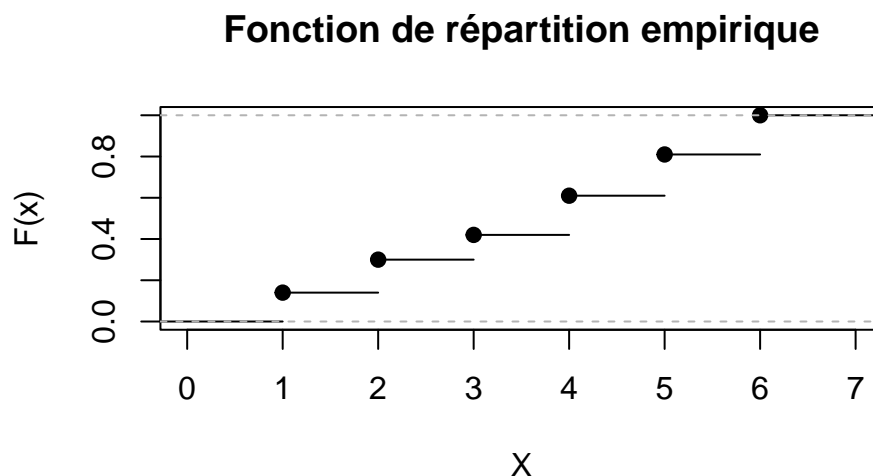
2.1.1 Loi uniforme discrète

Lorsqu'on ne connaît pas les probabilités de chaque issue d'une variable aléatoire, on associe à toutes les issues la même probabilité notée $1/n$. Les fonctions associées à la loi `runif` simulent une loi uniforme continue. On va donc utiliser la fonction `sample` de R pour générer un échantillon d'une loi uniforme discrète. La fonction `ecdf` permet de représenter graphiquement la fonction de répartition empirique.

Voici une simulation dans R de 100 réalisations d'un lancer de dé qui est une expérience aléatoire qui suit une loi uniforme :

```
set.seed(1789)
x <- sample(x = c(1,2,3,4,5,6), size = 100, replace = TRUE,
           prob = c(1/6,1/6,1/6,1/6,1/6,1/6))
```

```
Fx <- ecdf(x)
plot(Fx, xlab = "X",
     ylab = "F(x)", ylim = c(0,1),
     main = "Fonction de répartition empirique")
```



2.1.2 Loi binomiale

Une épreuve de Bernoulli est une expérience aléatoire qui n'a que deux issues possibles : succès ou échec, prenant la valeur 1 en cas de succès et 0 en cas d'échec. Une loi binomiale correspond au nombre de succès obtenus lors de n épreuves de Bernoulli. Les fonctions `binom` permettent de simuler une loi binomiale dans R.

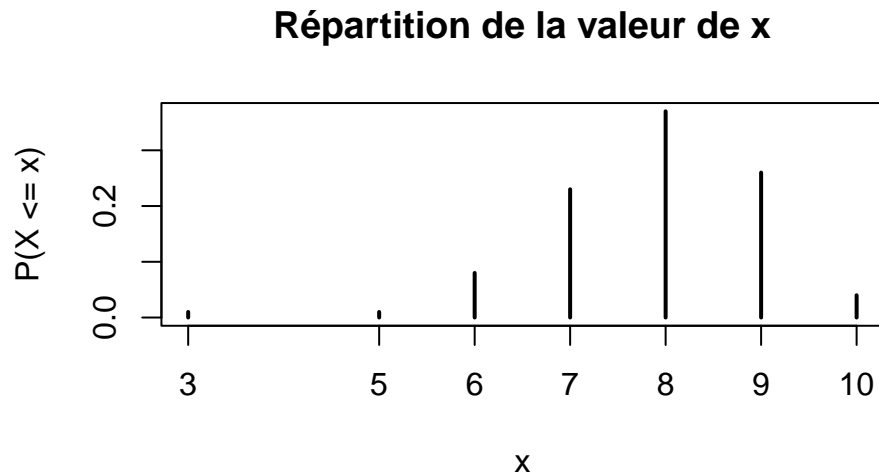
Voici une simulation dans R de 100 réalisations d'une variable aléatoire suivant une loi Binomiale de 10 épreuves de Bernoulli avec comme probabilité du succès 0.8 :

```
#Simulation de 100 réalisations d'une loi Binomiale de 10 épreuves de Bernoulli
set.seed(1789)
x <- rbinom(100, size=10, prob=0.8)

#On effectue un tri à plat en pourcentage des valeurs de X
```

```
freq_X <- prop.table(table(x))

#On construit un diagramme en bâton
plot(freq_X, xlab = "x", ylab = "P(X <= x)",
      main = "Répartition de la valeur de x")
```



On compare les statistiques de la simulation aux valeurs de la loi théorique. D'après la loi des grands nombres, plus le nombre de réalisation est grand, plus les caractéristiques de l'échantillon se rapprocheront de celles de la loi théorique.

Quelle est la probabilité de réaliser 8 succès ou moins sur les 10 tentatives ?

```
# D'après la loi théorique
pbinom(q = 8, size=10, prob=0.8)
```

```
## [1] 0.6241904
```

```
#Dans notre simulation
cumsum(freq_X)
```

```
##      3      5      6      7      8      9     10
## 0.01 0.02 0.10 0.33 0.70 0.96 1.00
```

Quelle est la valeur de x telle que $P(X \leq x) = 0.62$?

```
# D'après la loi théorique
qbinom(p = 0.62, size=10, prob=0.8)
```

```
## [1] 8
```

```
#Dans notre simulation  
cumsum(freq_X)
```

```
##      3      5      6      7      8      9     10  
## 0.01 0.02 0.10 0.33 0.70 0.96 1.00
```

2.1.3 Loi de poisson

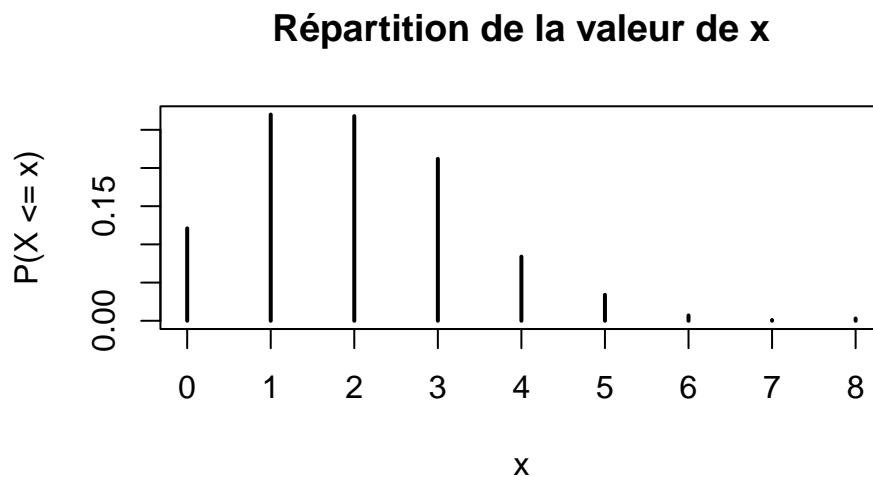
La loi de Poisson est une loi de probabilité discrète qui décrit le comportement du nombre d'événements se produisant dans un intervalle de temps fixé. Elle admet un paramètre appelé lambda. Par exemple, si un certain type d'événements se produit en moyenne 4 fois par minute, pour étudier le nombre d'événements se produisant dans un laps de temps de 10 minutes, on choisit un lambda de $10 \times 4 = 40$.

Dans cet exemple, on effectue 1000 réalisations d'une loi de Poisson avec un $\lambda = 2$

```
set.seed(1789)  
x <- rpois(n = 1000, lambda = 2)
```

```
#On effectue un tri à plat en pourcentage des valeurs de X  
freq_X <- prop.table(table(x))
```

```
#On construit un diagramme en bâton  
plot(freq_X, xlab = "x", ylab = "P(X <= x)",  
      main = "Répartition de la valeur de x")
```



On compare les statistiques de la simulation aux valeurs de la loi théorique. D'après la loi des grands nombres, plus le nombre de réalisation est grand, plus les caractéristiques de l'échantillon se rapprocheront de celles de la loi théorique.

Quelle est la probabilité que l'évènement se produit 2 fois ou moins ?

```
# D'après la loi théorique
ppois(q = 2, lambda = 2)
```

```
## [1] 0.6766764
```

```
#Dans notre simulation
cumsum(freq_X)
```

```
##      0      1      2      3      4      5      6      7      8
## 0.121 0.391 0.659 0.871 0.955 0.989 0.996 0.997 1.000
```

Quelle est le nombre d'évènements maximums qui peuvent se produire tel que $P(X \leq x) = 0.67$?

```
qpois(p = 0.67, lambda = 2)
```

```
## [1] 2
```

2.2 Lois de probabilités continues

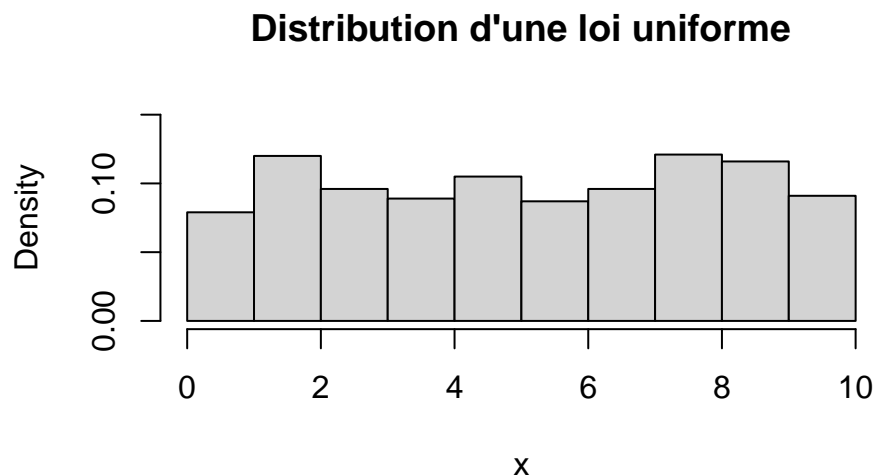
Une variable aléatoire qui peut prendre comme valeurs tous les nombres réels d'un certain intervalle est dite continue. On représente une variable aléatoire continue en regroupant ses valeurs dans un histogramme.

2.2.1 Loi uniforme continue

La loi uniforme sur un intervalle indique, intuitivement, que toutes les valeurs de l'intervalle ont les mêmes chances d'apparaître.

Voici un exemple dans R avec l'issue de 1000 réalisations d'une loi uniforme sur l'intervalle $[0 ; 10]$

```
set.seed(1789)
x <- runif(1000, min=0, max=10)
hist(x, main="Distribution d'une loi uniforme",
     probability = TRUE, ylim = c(0, 0.15))
```



Quelle est la probabilité que x soit inférieur ou égale à 8 ($P(X \leq 8)$) ?

```
# D'après la loi théorique
punif(q = 8, min=0, max=10)
```

```
## [1] 0.8
```

Quelle est la valeur de x pour que la probabilité $P(X \leq x) = 0.8$?

```
qunif(p = 0.8, min=0, max=10)
```

```
## [1] 8
```

La fonction `dunif` retourne la densité de probabilité de la variable aléatoire qui suit une loi uniforme. C'est cette valeur qu'on retrouve sur l'histogramme ci-dessus.

```
dunif(x = 0:10, min = 0, max = 10)
```

```
## [1] 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1
```

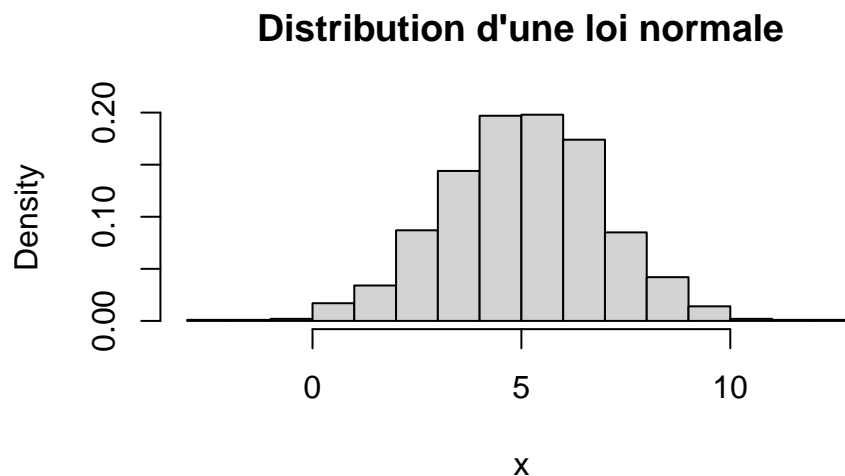
2.2.2 Loi normale

La loi normale est l'une des lois de probabilité les plus adaptées pour modéliser des phénomènes naturels issus de plusieurs événements aléatoires. Elle est également appelée loi gaussienne.

C'est une loi de probabilité continue qui dépend de deux paramètres : son espérance, son écart type.

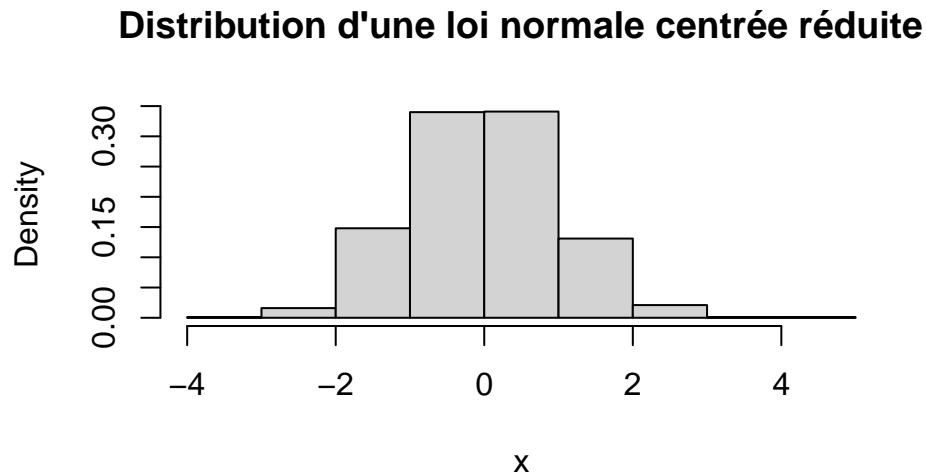
Voici un exemple dans R avec l'issue de 1000 réalisations d'une loi normale sur l'intervalle avec $\mu = 5$ et $\sigma = 2$

```
set.seed(1789)
x <- rnorm(n = 1000, mean = 5, sd = 2)
hist(x, main="Distribution d'une loi normale",
     probability = TRUE)
```



On parle de loi normale centrée réduite lorsque $\mu = 0$ et $\sigma = 1$

```
set.seed(2023)
x <- rnorm(n = 1000, mean = 0, sd = 1)
hist(x, main="Distribution d'une loi normale centrée réduite",
     probability = TRUE)
```



On compare les statistiques de la simulation aux valeurs de la loi théorique. D'après la loi des grands nombres, plus le nombre de réalisation est grand, plus les caractéristiques de l'échantillon se rapprocheront de celles de la loi théorique.

Quelle est la probabilité que x soit inférieur ou égale à 1.96 ($P(X \leq 1.69)$) ?

```
# D'après la loi théorique
pnorm(q = 1.96, mean = 0, sd = 1)
```

```
## [1] 0.9750021
```

Quelle est la valeur de x pour que la probabilité $P(X \leq x) = 0.975$?

```
# D'après la loi théorique
qnorm(p = 0.975, mean = 0, sd = 1)
```

```
## [1] 1.959964
```

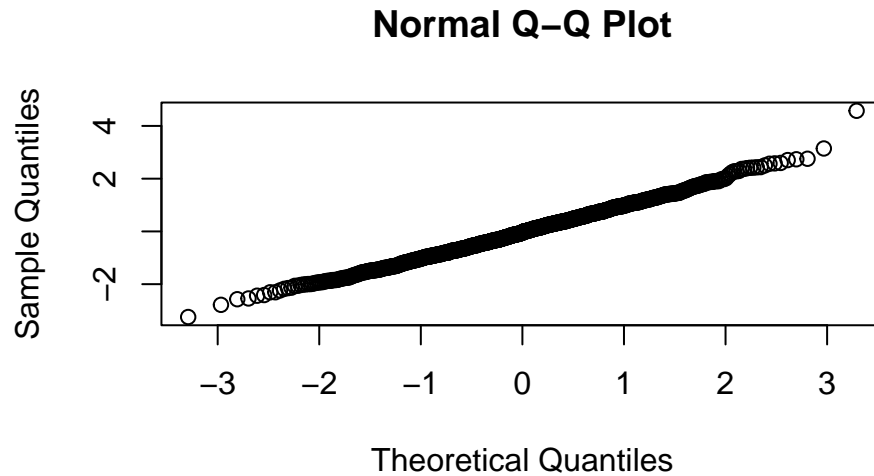
La fonction `dnorm` renvoie la valeur de la fonction de densité de probabilité pour $x = 0$ qu'on retrouve dans l'histogramme ci-dessus.

```
dnorm(x = 0, mean = 0, sd = 1)
```

```
## [1] 0.3989423
```


Le qqplot ou tracé quantile-quantile, est une représentation graphique permettant d'évaluer si un ensemble de données suit une distribution théorique telle qu'une normale.

```
qqnorm(y = x)
```



2.3 Récapitulatif

Loi	Appellation R	Arguments
Uniforme	unif	<code>min, max</code>
Binomiale	binom	<code>size, prob</code>
Poisson	pois	<code>lambda</code>
Normale	norm	<code>mean, sd</code>
Student	t	<code>df</code> (degrés de liberté)
Fisher	f	<code>df1, df2</code>
Chi deux	chisq	<code>df</code> (degrés de liberté)
...

2.4 Exercices d'entraînement

```
set.seed(1515)
```

1. Simulez 100 expériences aléatoires où la variable aléatoire suit une loi normale avec $\mu = 7$ et $\sigma = 2$
2. Représentez la distribution de cette simulation dans un histogramme
3. Quelle est la probabilité que la valeur de x soit inférieure ou égale à 9 ($P(X) \leq 9$)
4. Quelle est la probabilité que la valeur de x soit strictement supérieure 9 ($P(X) > 9$)
5. Construisez un Q-Q plot

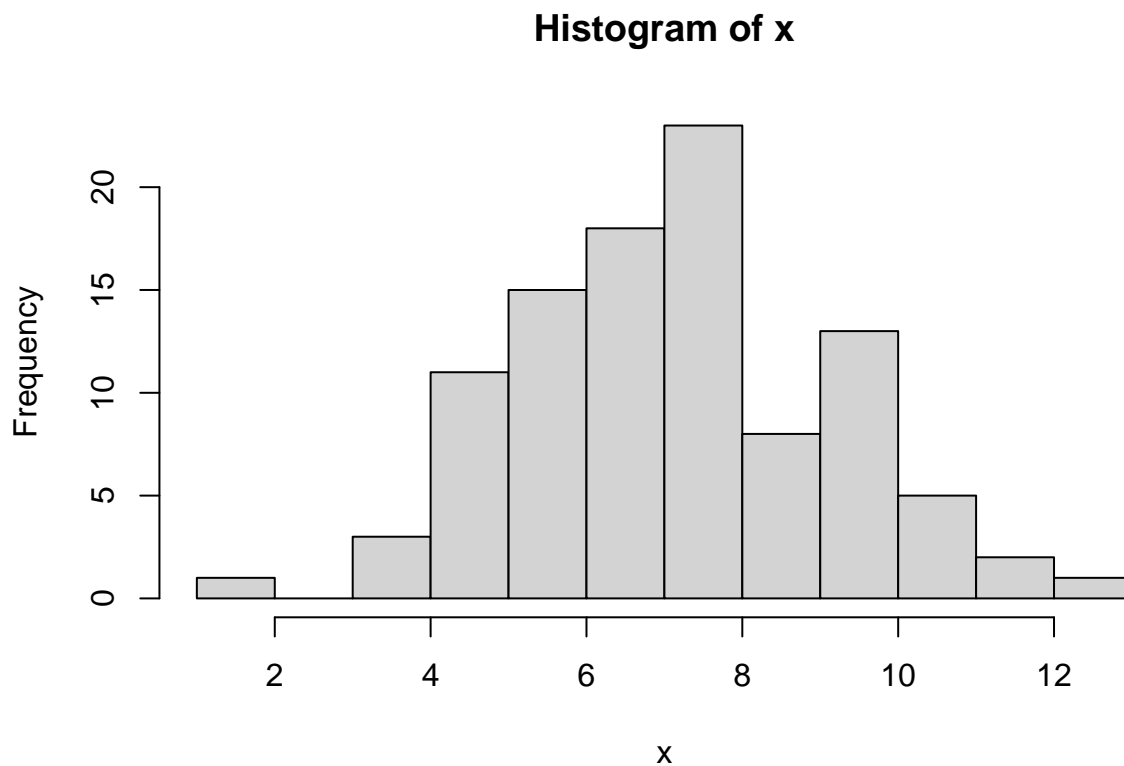
Corrigé :

1. Simulez 100 expériences aléatoires où la variable aléatoire suit une loi normale avec $\mu = 7$ et $\sigma = 2$

```
x <- rnorm(n = 100, mean = 7, sd = 2)
```

2. Représentez la distribution de cette simulation dans un histogramme

```
hist(x = x)
```



3. Quelle est la probabilité que la valeur de x soit inférieure ou égale à 9 ($P(X) \leq 9$)

```
pnorm(q = 9, mean = 7, sd = 2)
```

```
## [1] 0.8413447
```

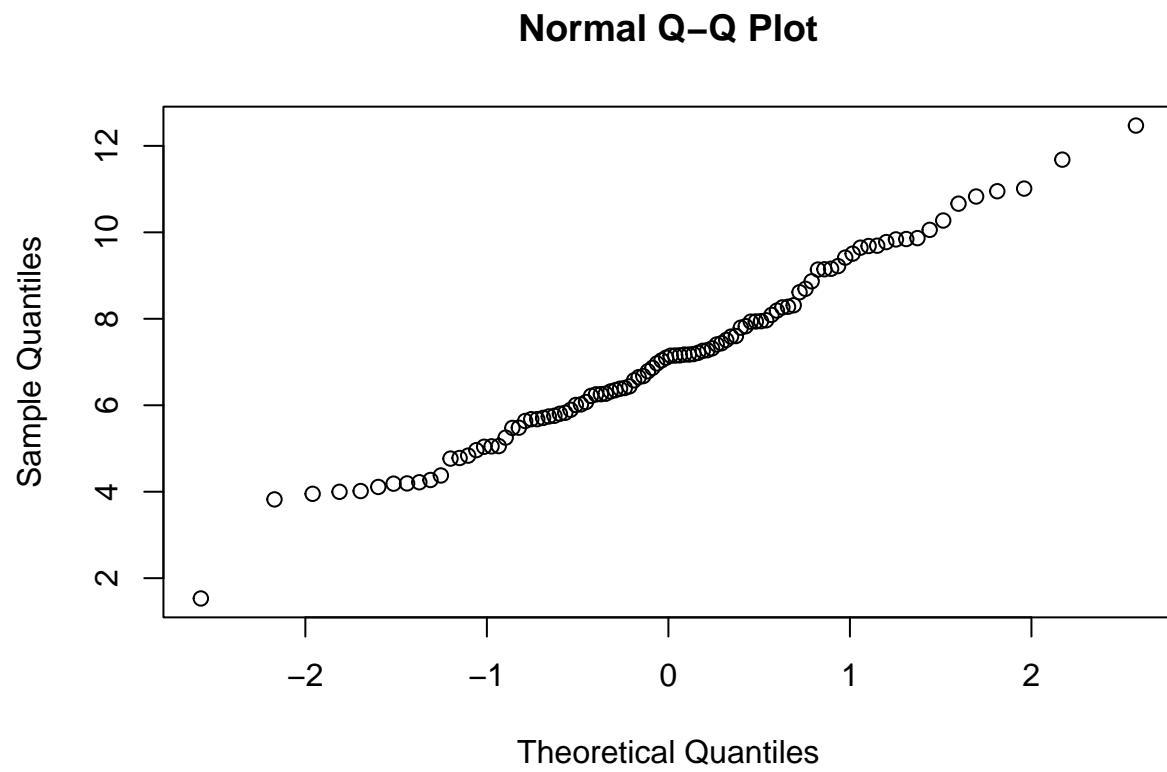
4. Quelle est la probabilité que la valeur de x soit strictement supérieure 9 ($P(X) > 9$)

```
1 - pnorm(q = 9, mean = 7, sd = 2)
```

```
## [1] 0.1586553
```

5. Construisez un Q-Q plot

```
qqnorm(y = x)
```



3 Fonctions Apply

Dans le cours, parcourir la partie Chapitre 8 - les fonctions Apply