Feuille de TP n°1

A. Introduction au logiciel R

Le logiciel **R** est un logiciel libre permettant de réaliser une étude statistique de jeux de données. Le logiciel **R** est gratuit. La page officielle du logiciel est : www.r-project.org. Si vous avez un ordinateur à la maison, vous pouvez télécharger **R** à l'adresse :

http://cran.r-project.org/

Glossaire

R	Lance le logiciel R
q()	Quitte le logiciel
?commande	Demande la fiche de documentation de la ***commande***
s <- valeur	Initialise la variable ***s*** avec la ***valeur***. *Exemple :* `n <- 5`
n:m	Crée une suite de nombres entiers de ***n*** à ***m***. *Exemple :* `s <- 2:5` initialise s avec la suite 2, 3, 4, 5
seq(n,m,i)	Crée une suite de nombres de ***n*** à ***m*** en incrémentant par ***i*** *Exemple :* `seq(2,4,0.5)` produit la suite 2, 2.5, 3, 3.5, 4
c(s_1, s_2,..., s_k)	Crée une suite en collant les ***s_1, s_2,..., s_k*** dans l'ordre. *Exemple :* c(2:5, 7, seq(8, 9, 0.5)) produit 2,3,4,5,7,8,8.5,9
rep(s,n)	Crée une suite contenant ***n*** fois ***s***. *Exemple :* rep(c(1, 2, 3) , 2) donne 1,2,3,1,2,3
scan()	Saisir "au clavier" un jeu de données *numériques*
s[I]	Crée une suite composée des elements de la suite ***s*** indexées par ***I***. Ici ***I*** peut etre de la forme : • ***I*** est entier. *Exemple :* **s[3]** renvoie le 3ème element de **s** • ***I*** est une suite. *Exemple :* **s[3:5]** renvoie les 3e, 4e et 5e éléments de ***s*** • ***I*** est une condition. *Exemple :* **s[t>3]** renvoie les éléments de la suite ***s*** correspondants aux éléments de la suite ***t*** qui sont supérieurs à ***3***.
s[-I]	Crée une suite composée des éléments de la suite ***s*** qui sont complémentaires à ceux indexés par ***I***.
mode(s)	Affiche le mode (numérique, caractère, ...) de la variable ***s***
length(s)	Affiche le nombre d'éléments contenus dans la variable ***s***
names(s)	Renvoie les noms des éléments de ***s***. Si ***s*** est une table, renvoie les noms des colonnes. *Exemple :* `names(s) <- nom` renomme les éléments de s en utilisant les valeurs de la suite nom.
sort(s)	Trie les composantes d'un vecteur par ordre croissant
rev(sort(s))	Trie les composantes d'un vecteur par ordre décroissant
read.table("file")	Lit le fichier de données ***file*** ne contenant pas les noms des variables en première ligne. Rajouter « ,header=T » dans le cas contraire.
tab$col	Renvoie la suite composée des éléments de la colonne ***col*** de la table ***tab***. *Exemple :* `e <- amis$email` initialise la variable e avec les valeurs de la colonne `email` de la table `amis`.

La plupart des méthodes statistiques sont utilisables au travers de ce logiciel dont le développement se fait en collaboration avec les chercheurs du monde entier. En effet, la majorité des nouveaux outils statistiques sont rendus accessibles par les statisticiens au travers de procédures implémentées avec **R**, ce qui en fait un des logiciels les plus complets d'analyse statistique.

Le logiciel **R** fonctionne sous forme de lignes de commandes, c'est à dire que vous rentrez une ligne de commande et qu'il vous renvoie ensuite le résultat. Il n'y a pas d'étape de compilation.

1. Pour commencer avec R

Démarrer R :
Sous Unix :
Vous lancez le logiciel R en tapant R (puis valider par la touche entrée) dans un terminal.
Sous Windows :
Vous lancez le logiciel R en double-cliquant sur l'icône.

Le symbole > signifie que R est prêt à travailler. Il ne faut pas taper ce symbole au clavier car il est déjà présent en début de ligne sur la console. C'est à la suite de ce symbole > que vous pourrez taper les commandes R. Une fois la commande tapée, vous devez toujours la valider par la touche Entrée.

Quitter R :
Pour quitter R, on utilise la commande
>q()
ou les menus (sous windows).

La question "Save workspace image? [y/n/c]" est posée : R propose de sauvegarder le travail effectué. Trois réponses possibles : y (pour yes), n (pour no) ou c (pour cancel, annuler). En tapant "c", la procédure de fin de session sous R est annulée. Si vous tapez "y", cela permet que les commandes tapées pendant la session soient conservées en mémoire et soient donc "rappelables" (mais vous ne pouvez pas les imprimer).

Sauvegarder sous R :
Si vous quittez R en choisissant la sauvegarde de l'espace de travail, deux fichiers sont créés :

- le fichier .Rdata contient des informations sur les variables utilisées ,
- le fichier .Rhistory contient l'ensemble des commandes utilisées.

Attention, afin de garder et sauvegarder vos commandes de manière plus lisible et claire, il est impératif de les recopier dans un éditeur de texte en dehors de R.

Travailler avec R :
Par exemple, tapez la commande suivante et validez :
> 2 + 5
Le résultat s'affiche sous la forme :
[1] 7
Le chiffre 1 entre crochets indique l'indice du premier élément de la ligne[1], le second chiffre est le résultat de l'opération demandée.
Vous pouvez rappeler les commandes déjà exécutées (pendant cette séance) en utilisant la touche "Flèche vers le haut".

Consulter l'aide:
Pour toutes les commandes, vous pouvez consulter une fiche de documentation en tapant, par exemple pour la commande "read.table" :
> ?read.table
Faire défiler le texte avec la touche "Entrée" ou "Flèche vers la bas". Une fois arrivé à "END", taper "q".
Grâce à cette aide, il suffit de retenir le nom de la commande, mais pas toute la syntaxe.

2. Rentrer des données dans R

Différentes commandes sont disponibles pour saisir des données sous R.

2.1. Affectation
Un objet peut être créé avec l'opérateur "assigner" ou "affecter" qui s'écrit ← ou = :
> n<-15
> N=12
Pour vérifier le contenu d'un objet, taper son nom, par exemple pour n :
Par exemple dans le résultat suivant l'indice de l'élément 123 est 1 et celui de 142 est 20.
[1] 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141
[20] 142 143 144 145
> n
[1] 15

Notes :
· R différencie les lettres minuscules et les lettres majuscules.
· Quand on assigne un nom à un objet, l'affichage de cet objet n'est plus automatique, il faut le demander en tapant simplement le nom donné à l'objet.

2.2. Suite
Pour créer une suite d'entiers, par exemple de 1 à 12 :
> suite<-1:12
> suite
[1] 1 2 3 4 5 6 7 8 9 10 11 12
La fonction seq crée une suite (séquence) de nombres et possède trois arguments : from, to et by.
Par exemple, pour créer un vecteur formé par les éléments d'une suite arithmétique de premier terme 20, de dernier terme 40 et de raison 5, on utilise :
> seq(from=20,to=40,by=5)
[1] 20 25 30 35 40
On peut aussi écrire simplement :
> seq(20,40,5)

2.3. Combinaison ou vecteur
Il est possible de saisir une série de valeurs numériques, caractères ou logiques :
> serie1<-c(1.2,36,5.33,-26.5) *serie1 est un vecteur numérique*
> serie1
[1] 1.20 36.00 5.33 -26.50
> serie2<-c("bleu","vert","marron") *serie2 est un vecteur de chaînes de caractères*
> serie2
[1] "bleu" "vert" "marron"

Note : Si un vecteur est composé de caractères et de nombres, le vecteur sera un vecteur de chaînes de caractères.

Quand les composantes du vecteur sont des chaînes de caractères, il est obligatoire de les déclarer entre guillemets, sinon R ne reconnaît pas les composantes du vecteur :
> serie2<-c(bleu,vert,marron)
Error : Object "bleu" not found
> serie3<-c(T,T,F,F,T) *serie3 est un vecteur logique*
> serie3
[1] TRUE TRUE FALSE FALSE TRUE
Lors d'une étude statistique, il peut arriver que certaines données ne soient pas disponibles : on dit que la donnée est manquante. Pour saisir une donnée manquante, on utilise le symbole NA (*Not Available*) que l'objet soit numérique, caractère ou logique :
> serie4<-c(1.2,36,NA,-26.5) *la troisième valeur est laissée en valeur manquante*
> serie4
[1] 1.20 36.00 NA -26.50

2.4. Mode et longueur

Les objets sont caractérisés par deux attributs : le mode et la longueur. Le mode est le type des éléments d'un objet.
Comme nous venons de le voir, un objet peut être numérique, caractère ou logique. La longueur est le nombre d'éléments de l'objet. Par exemple, si vous saisissez une série d'observations obtenues sur un échantillon sous la forme d'un vecteur, la longueur de ce vecteur correspondra à la taille de l'échantillon. Pour connaître le mode et la longueur d'un objet, on utilise les fonctions *mode* et *length* :

> mode(serie1)
[1] "numeric"
2
> mode(serie2)
[1] "character"
> mode(serie3)
[1] "logical"
> length(serie1)
[1] 4
> length(serie2)
[1] 3
> length(serie3)
[1] 5

2.5. Saisie "au clavier" d'un jeu de données

En utilisant la fonction *scan*, la saisie d'une série de données peut paraître moins fastidieuse.

> jeu1<-scan()
R vous redonne la main et vous pouvez taper les valeurs du jeu de données :
1: 1.2
2: 36
3: 5.33
4: -26.5
5:

Le premier retour-chariot après une chaîne vide met fin à la saisie
> jeu1
[1] 1.20 36.00 5.33 -26.50

2.6. Eléments d'un vecteur

Il est possible de demander l'affichage d'un (ou de plusieurs) élément(s) d'un vecteur en spécifiant entre crochets, en plus du nom du vecteur, l'indice de l'élément du vecteur. Par exemple, pour afficher le troisième élément

```
> serie1[3]
[1] 5.33
> serie1[3:4]
[1] 5.33 –26.50
```

3. Manipuler des vecteurs

Plusieurs opérations sont possibles sur les vecteurs : concaténation, extraction, calculs, répétition, légende et tri.

3.1. Concaténer deux vecteurs

Il est possible de concaténer deux vecteurs (formés de variables de même type) pour en former un nouveau :

```
> x <- c(2.3,3.5,6,14,12)
> y <- c(3.2,5,7,10,13.5)
> z <- c(x,y)
> z
[1] 2.3 3.5 6.0 14.0 12.0 3.2 5.0 7.0 10.0 13.5
```

3.2. Extraire des données d'un vecteur

Il est possible d'extraire des données à partir d'un vecteur selon trois façons :

i. utiliser un vecteur pour préciser le numéro d'ordre des composantes à extraire. Ainsi pour extraire les 2ème et 5ème composantes du vecteur x :

```
> x[c(2,5)]
```

ii. L'utilisation du signe "-" permet de supprimer des composantes, par exemple pour supprimer les 2ème et 3ème composantes du vecteur :

```
> x[-c(2,3)]
```

iii. Utiliser un vecteur formé de valeurs logiques. Par exemple, pour obtenir un vecteur ne contenant que les composantes supérieures à 4, on peut utiliser la commande :

```
> x[x>4]
```

Si on dispose de deux vecteurs ayant le même nombre de composantes, on peut demander à afficher les valeurs de l'un pour lesquelles les valeurs de l'autre sont supérieures (ou inférieures) à une certaine valeur. Par exemple, les vecteurs x et y sont composés de 5 valeurs. On peut demander d'extraire de y les valeurs de y pour lesquels x est supérieur à 4 :

```
> y[x>4]
```

3.3. Faire des calculs sur les composantes d'un vecteur

R peut faire des calculs sur l'ensemble des composantes d'un vecteur :

```
> 20 + x*5
[1] 31.5 37.5 50.0 90.0 80.0
> (x+y)/2
[1] 2.75 4.25 6.50 12.00 12.75
```

3.4. Remplacer des données dans un vecteur

Il est possible de remplacer certaines composantes d'un vecteur par de nouvelles valeurs. Par exemple, à partir d'une suite de valeurs :

```
> x<- 1:10
```

Si on veut remplacer la 3ème valeur de x par 35 :

```
> x[3] <- 35
> x
[1] 1 2 35 4 5 6 7 8 9 10
```
On veut remplacer la valeur 1 par la valeur 25 :
```
> x[x==1] <- 25
> x
[1] 25 2 35 4 5 6 7 8 9 10
```
Si on veut remplacer toutes les valeurs supérieures ou égales à 5 par 20, on utilise la commande :
```
> x[x>=5] <- 20
> x
[1] 25 2 35 4 20 20 20 20 20 20
```

3.5. Répéter les données d'un vecteur

La fonction *rep* a deux arguments x et times et crée un vecteur où x est répété times fois.
Par exemple, si on crée un variable donnees par :
```
> donnees <- c(1,2,3)
```
et si on veut qu'un nouveau vecteur contienne deux fois le vecteur donnees, on écrit :
```
> rep(x=donnees,times=2)
```
On peut également demander qu'un vecteur contienne 50 fois la valeur 1 :
```
> rep(1,50)
```
ou 4 fois la chaîne de caractères "chien" :
```
> rep("chien",4)
```

3.6. Nommer les composantes d'un vecteur

Il est possible de donner un nom à chaque composante d'un vecteur. Par exemple, si le vecteur notes.Jean contient les notes obtenues par Jean en Anglais, Informatique et Biologie, on peut utiliser la commande :
```
> notes.Jean <- c(Anglais=12,Informatique=19.5,Biologie=14)
```
Afficher le vecteur notes.Jean.

Une autre façon de nommer les composantes d'un vecteur est de définir un vecteur formé de chaînes de caractères, puis
utiliser la fonction names :
```
> matiere <- c("Anglais","Informatique","Biologie")
> note <- c(12,19.5,14)
> note
[1] 12.0 19.5 14.0
> names(note) <- matiere
> note
Anglais Informatique Biologie
12.0 19.5 14.0
```

Pour supprimer les noms :
```
> names(note) <- NULL
```

3.7. Trier les composantes d'un vecteur

On peut trier les composantes d'un vecteur par ordre croissant en utilisant la fonction sort. Par exemple, pour trier les notes précédentes dans l'ordre croissant :
```
> sort(note)
```
ou dans l'ordre décroissant :
```
> rev(sort(note))
```

4. Lire des données dans un fichier

Quand les données sont plus volumineuses, il n'est pas très conseillé d'utiliser R comme outil de saisie. Dans ce cas, vous pouvez utiliser un éditeur de texte ou un tableur quelconque pour saisir vos données (excel par exemple) et le transférer ensuite sous R. Supposons que les données suivantes ont été saisies dans le fichier table1.dat :

53.5 160
74.4 172
52.6 151
88.6 163
49.2 169

```
> read.table("table1.dat")
```

R affiche le tableau de données en numérotant les lignes et les colonnes, les lignes correspondant aux individus et les colonnes aux variables. R affiche un message d'avertissement concernant le nom des variables.

On peut également conserver la table comme un objet pour pouvoir la réutiliser directement :

```
> tab<-read.table("table1.dat")
```

et demander l'affichage de cet objet :

```
> tab
```

ou seulement d'une colonne de cet objet :

```
> tab$V1
```

ou seulement de l'élément de la première ligne et de la première colonne :

```
> tab[1,c(1)]
```

ou `> tab[1,1]`

ou les éléments des deux premières lignes et de la première colonne :

```
> tab[1:2,1]
```

ou les éléments des deux premières lignes et des deux premières colonnes :

```
> tab[1:2,1:2]
```

Pour travailler ensuite sur les variables de la table, vous pouvez leur attribuer un nom (plus simple que la syntaxe utilisée) :

```
> V1 <- tab$V1
> V2 <- tab$V2
```

Si vous avez spécifié le nom des variables dans la première ligne de votre fichier de données (table2.dat), vous devez l'indiquer par l'option "header=TRUE" ou "header=T" :

```
> read.table("table2.dat",header=T)
```

Par défaut, R lit la première ligne comme une ligne de données et nomment les colonnes sous la forme V1, V2 … (comme pour table1.dat).

Par défaut, on utilise un point (.) pour les décimales. Mais si les décimales sont notées par une virgule dans votre fichier de données (comme dans table3.dat), il faut le spécifier par :

```
> read.table("table3.dat",dec=",")
```

Certains jeux de données sont fournis avec le logiciel R.

On peut les charger en tapant :

```
>data(faithful)
>attach(faithful)
```

Obtenir une description en tapant :

```
>?faithful
>head(faithful)
```

Note: Il arrive que les données soient incluses dans une librairie. Pour y accéder il suffit d'ouvrir la librairie:

```
>library(lycee)
```

puis d'appeler les données correspondant au fichier entreprise:

>data(entreprise)
Pour visualiser les données on tape le nom du fichier:
>entreprise

5. Modélisation et Probabilités

5.1 Quelques rappels de Probabilités:

On considère un espace de probabilité (O,A,P). Une variable aléatoire X est une fonction définie sur l'espace O et à valeurs dans un ensemble U. Vous avez vu en probabilité différents exemples de variables aléatoires à valeur dans R. On fait généralement la différence entre des variables dites discrètes qui ne prennent au plus qu'un nombre dénombrable de valeurs distinctes (par exemple des valeurs entières) et des variables continues qui peuvent prend leurs valeurs sur un intervalle de R.

La loi de la variable aléatoire correspond à l'ensemble des probabilités P(X(o) $\in$ u) pour tout u inclus dans U.
Elle est caractérisée par la fonction de répartition F(x)=F(X<=x) qui représente la probabilité que la variables soit inférieure ou égale à un seuil x.
On note E l'espérance (valeur moyenne) et Var la variance (écart carré moyen à l'espérance).
La fonction quantile Q est la réciproque de la fonction F. Elle est définie par Q(s)=inf(t,F(t)>s). Elle permet d'obtenir la plus petite valeur seuil t_0 pour laquelle X soit inférieure à t_0 avec une probabilité supérieure à un seuil s.

Remarque : si a<b, P(a<X<=b)=F(b)-F(a)
Attention !: si a<b, P(a<=X<=b)=F(b)-P(X<a), P(a<=X<b)=P(X<b)-P(X<a), et P(a<X<b)=P(X<b)-P(X<=a)

5.2 Commandes R

Les commandes détaillées dans le glossaire suivant sont données pour la loi gaussienne mais sont également proposées pour d'autres lois.

Glossaire

rnorm(n,m,s)	Simule n données provenant de la loi normale de moyenne m et de variance s^2.
dnorm(x,m,s)	Renvoie la valeur de la densité de la loi normale de moyenne m et de variance s^2 au point x.
pnorm(x,m,s)	Renvoie la valeur de la fonction de répartition de la loi normale de moyenne m et de variance s^2 au point x.
qnorm(x,m,s)	Renvoie la valeur v telle que P(X<=s)=x.

5.3 Variables discrètes:

Pour une variable discrète elle est également caractérisée par l'ensemble des probabilités P(X=m_i) ou les m_i sont les modalités (i.e. différentes valeurs possibles) de la variable.

Quelques lois discrètes bien connues:
Loi de Bernoulli: P(X=0)=1-p, P(X=1)=p, lancé d'une pièce: pile ou face
Loi Binomiale: P(X=k)=n!/((n-k)!k!)p^k(1-p)^(n-k), nombre de piles sur n lancés indépendants
Loi de Poisson: P(X=k)=exp(-u)u^k/k!, modélisation de survenue d'événements.
Loi uniforme discrète: P(X=m_j)=1/N, 1<=j<=N; toutes les modalités ont la même probabilité.
Loi géométrique : P(X=k)=p(1-p)^k, nombre d'essais avant le premier succès.

Modélisation probabiliste et génération de données correspondant à l'observation d'une variable aléatoire avec le logiciel R :

Tracer un diagramme en bâtons représentant la loi Bin(n=6,p=0.4).
Quelle est la probabilité qu'une variable suivant cette loi soit inférieure à 4 ?
Donner la plus petite valeur de s pour laquelle X<=s avec une probabilité supérieure à 0.98.

Générer un échantillon de taille 50 d'observations correspondant à la loi Bin(n=8,p=0.2).
Calculer la moyenne, la variance et représenter la répartition des valeurs observées par un diagramme en bâtons.
Comparer avec la répartition théorique.

EXERCICE : Faire de même avec un échantillon de taille 5000.

EXERCICE: Générer un échantillon de 5000 observations de loi de poisson de paramètre 4.
Comparer les distributions empiriques et théoriques. Calculer la moyenne et la variance

Simuler un échantillon de taille 100 correspondant à des lancés de dés à 6 faces équilibrés.
Représenter la répartition des observations par un diagramme en bâtons.
Calculer la moyenne et l'écart-type des valeurs obtenues.
Quelle modélisation probabiliste devrait-on retrouver ? Est-ce le cas ?

EXERCICE : Faire de même avec des lancés de dés équilibrés à 20 faces.

5.4 Variables continues :
Pour les variables continues, les probabilités P(X=x) sont nulles. On caractérise souvent la loi de X par la densité f qui est la dérivée de F.

Remarque :
Dans le cas de variables continues on a P(a<X<=b)=P(a<=X<=b)=P(a<=X<b)=P(a<X<b)

Quelques lois continues bien connues:
Loi uniforme sur I: f(x)=1/|I| si x appartient à I, 0 sinon; répartition homogène.
Loi Normale: f(x)=(2 $\pi\sigma^2$)^(-1/2)exp(-(x-m)^2/2 σ^2), répartition symétrique, en forme de cloche.
Loi exponentielle: v>0, f(x)=v.exp(-vx) si x>0, 0 sinon; modélisation de durée de vie.
Loi Gamma: *f(x)= 1/(s^a Gamma(a)) x^(a-1) e^-(x/s), x >= 0, a > 0* et *s > 0.*
Loi du Chi 2 : f_n(x) = 1 / (2^(n/2) Gamma(n/2)) x^(n/2-1) e^(-x/2), pour x > 0.

Tracer la densité d'une variable gaussienne N(0,1) entre -10 et 10.
Changer les valeurs de la moyenne et de l'écart-type. Commenter.
Quelle est la probabilité qu'une variable aléatoire de loi N(0,1) soit négative ou nulle ? Soit inférieure à 1 ?

EXERCICE: Soit X une variable de loi N(1,2)
Calculer la probabilité que la variable soit comprise entre 1 et 2.
Calculer la probabilité que la variable soit supérieure à 2.
Quelle est la loi de Y=(X-1)/2 ?

Générer 20 observations provenant d'une loi gaussienne standard N(0,1). Calculer la moyenne et la variance des observations, tracer l'histogramme de ces données. Comparer avec les valeurs théoriques et la densité N(0,1).

EXERCICE:

1. Faire la même chose pour un échantillon de taille 5000. Commenter.
2. Faire de même pour un échantillon de taille 5000 distribué selon une loi uniforme entre 0 et 2.
3. Faire de même pour un échantillon de taille 5000 distribué selon une loi exponentielle exp(1). On constate qu'elle ne prend que des valeurs positives
4. Comparer sur un même graphique les densités des lois Exp(1), Gamma(1,1), Gamma(5,1) et Gamma(1,0.01). Commenter. On constate que le premier paramètre d'une loi gamma est un paramètre de forme, le second est un paramètre d'échelle.

On a notre disposition diverses formes de lois théoriques pour modéliser les distributions qu'on observe.

5.4 Quelques propriétés

A. La loi Bin(n,p) peut être approchée par une loi de Poisson de paramètre np sous les conditions n>30, et np(1-p)>=12.

Tracer sur un même graphique les densités Bin(50,0.08) et P(4)
EXERCICE: Faire la même chose pour p=0.5 et commenter.

B. La loi Pois(u) peut être approchée par une loi N(u,u^{1/2}) sous les condition u>10 ou 20.

Tracer sur un même graphique les densités P(20) et N(20,20^{1/2}))
EXERCICE: Faire la même chose pour u=1 et commenter.

C. Loi des grands nombres.

Soient X_1,..., X_N des variables indépendantes, de même loi et intégrables (E[|X|] existe et est fini). Alors S_N/N= (X_1+...+X_N)/N converge (en probabilité ou presque surement) vers E[X].

Visualiser la loi des grands nombres avec les commandes:

```
>x=rpois(1000,2)
>plot(cumsum(x)/(1:1000),type="l",ylim=c(1,3))
>for (s in 1:100)
   {
    x=rpois(1000,2)
    lines(cumsum(x)/(1:1000),type="l")
   }
```

EXERCICE : Changer la loi de poisson de paramètre 2 en loi exponentielle de paramètre 0.5. Commenter.

D.Théorème de la limite centrale.

Soient X_1,..., X_N des variables indépendantes, de même loi et de carré intégrable (E[X] et V(X) existent et sont finis) avec V(X) ≠ 0. Posons S_N=X_1+...+X_N.
*On montre que (S_N-N*E[X])*(NV(X))^(-1/2) est asymptotiquement (lorsque N tend vers l'infini) de loi N(0,1). On peut donc approcher la loi de (S_N-N*E[X])*(NV(X))^(-1/2) par la loi N(0,1) pour N assez grand (N>30 suffit en général).*

```
>x=matrix(rpois(100000,2),ncol=1000,byrow=T)
>Moy=t((apply(x,1,cumsum))/(1:1000))
>plot(Moy[1,],ylim=c(1,3),type="l")
>for (j in 1:100) lines(Moy[j,])
>u=t((t(Moy)-2)/sqrt(2)*sqrt(1:1000))
>hist(u[,100],freq=F,nclass=10)
>z=seq(-5,5,0.01)
>lines(density(u[,100]),col="green")
>lines(z,dnorm(z),col="red")
```

EXERCICE : Modifier ces commandes pour considérer le cas de variables de loi exponentielle de paramètre 1.

EXERCICE SUPPLEMENTAIRE :

1. ***Créer un vecteur contenant les données recueillies au début du cours concernant les nombres de frères et sœurs.***
2. ***Représenter la répartition de ces données par un diagramme en bâtons.***
3. ***Calculer le nombre moyen de frères et sœurs ainsi que l'écart-type.***
4. ***Proposer une modélisation.***

EXERCICES du polycopié:

EXERCICES:

A. Pour essayer de prévoir la défaillance des entreprises, l'économiste W.BEAVER introduit le ratio défini pour chaque entreprise, par le quotient de la marge brute d'autofinancement (cash flow) par la dette totale. On dispose dans le fichier "entreprise" du ratio d'un échantillon de 1000 entreprises. On peut maintenant regarder des données réelles et discuter dessus. *(utilise les données contenues dans le fichier* ***entreprise*** *de la librairie* ***lycee****) :*

(a.1) Définir les variables ratio et etat contenant les ratios et les états des entreprises
(a.2) Tracer l'histogramme du ratio à l'aide de la commande **hist**.
(a.3) Commenter.
(b.1) Trouver quelles valeurs prend la variable etat à l'aide de la fonction table
(b.1) Définir deux variables ratioS et ratioD correspondant aux ratios des entre prises saines et défaillantes respectivement.
(b.3) Tracer les histogrammes de ces deux variables. Commenter.
Optionel: calculer les moyennes mean() et les écart-types sd().

En tenant compte des lois supposées pour les ratios dans chacun des deux groupes
(c.1) Déterminer la probabilité qu'une entreprise défaillante ait un ration a) inférieur à 0,94; b) supérieur à 0,475.
(c.2) Déterminer la probabilité qu'une entreprise saine ait un ration a) entre 0.2590 et 1.195; b) entre 0.2590 et 0.52; c) supérieur à 0.2590.
(c.3) Trouver l'intervalle [a;b] centré en m_1 dans lequel se trouve le ratio d'une entreprise saine avec une probabilité de 95%.
(c.4) On décide de juger saines les entreprises dont le ratio est supérieur ou égal à a et défaillantes celles dont le ratio est inférieur à a.
a) Quel est le risque de classer défaillante une entreprise saine?
b) Quel est le risque de classer saine une entreprise défaillante?
c) Optionnel: Quelles serait le nombre d'erreurs de classement parmi les entreprises étudiées?

B. D'après une étude statistique , les demandes de location des studios (appartement de une pièce) ont représenté 40% des demandes en 2001 dans la région Parisienne. La direction d'une agence de

location veut vérifier si elle peut considérer ce pourcentage comme pouvant s'appliquer à son portefeuille de clientèle. Pour ceci, elle prélève un échantillon aléatoire de 600 demandes sur l'ensemble des demandes de son portefeuille de l'année. Les données sont contenues dans le fichier "locappart".

0. Créer une séquence c contenant 263 le chiffre 1, 170 le chiffre 2, 73 le chiffre 3, 43 le chiffre 4, 28 fois le chiffre 5, 11 le chiffre 6 et 12 fois le chiffre 7.
a. Si la probabilité qu'un contrat concerne un studio est bien de 0.40,
(a.1) Dire quelle est la loi du nombre s de contrats portant sur des studios parmi les N contrats de l'agence.
(a.2) Combien peut-on espérer de demandes de studios sur les 600 demandes étudiées.
b. La séquence c est une version réarrangée des résultats obtenus lors de l'étude. Commenter.
Optionnel: Déterminer les effectifs avec la commande **table**, les fréquence en divisant les résultats obtenus par 600, puis tracer le diagramme des valeurs obtenues avec la commande **plot**.
c. Quelle est la probabilité, en supposant que 0.4 est la vraie valeur de la probabilité, d'avoir obtenu autant de demandes de studios?
d. Quelle conclusion peut-on en tirer?

C. Exercice sur le Surbooking.
Le grand stade de France a 80000 places. Pour la coupe du monde, vu les demandes,les organisateurs se sont demandé s'il était raisonnable de pratiquer ce que l'on appelle "le surbooking" en vendant 84000 places. Ils ont donc fait des études statistiques pour estimer que la probabilité notée p "qu'une personne ayant acheté un billet n'assiste pas au match" est 0,05. Ils peuvent donc répondre aux questions suivantes et décider du surbooking.

1. Si les organisateurs vendent 84000 places, quelle est la probabilité pour que toutes les personnes ayant acheté un billet pour un match donné et se présentant le soir du match puissent avoir une place ?

2. Quel est le nombre maximum N de billets que l'on peut vendre, pour que la probabilité que toutes les personnes, ayant acheté un billet et se présentant pour assister au match puissent avoir une place, soit supérieure à 99%?

Remarque: si on vend N billets, on dit qu'on fait du surbooking avec un risque inf?erieur ?a 0.01

3. On peut imaginer que l'étude puisse être poussée plus loin, comme dans les aéroports, comment peut-on dédommager les supporter qui ont un billet mais pas de place le soir du match, pour que en moyenne les organisateurs aient encore intérêt à pratiquer le surbooking?..... On note B le prix du billet et I l'indemnisation.