Probabilités discrètes et Statistiques descriptives **L2 Universite Lyon 1 2018**

Correction du TP 1 d'Initiation à R

R est un logiciel libre qui peut être téléchargé gratuitement sur le site du CRAN (Comprehensive R archive project, https ://cran.r-project.org/), qui fonctionne sous Microsoft Windows, Mac OS et Linux.

1 Interface Rstudio

RAS

2 Prise en main

Ouvrir un nouveau script et l'enregistrer en utilisant un nom se terminant par .R . Taper une commande dans le script, par exemple 1+2. Pour la lancer dans la console, sélectionner la ligne correspondant et taper Ctrl + Entrée.

Remarque : si besoin plusieurs lignes peuvent être sélectionnées et lancées en même temps. Pour mettre une ligne en commentaire (cette ligne ne sera alors pas lancée dans la console, même si elle est sélectionnée), la commencer par #.

Pour obtenir de l'aide sur une fonction, par exemple plot, on peut utiliser directement le moteur de recherche situé dans la fenêtre d'aide, ou taper dans la console ?plot ou help(plot). Pour lancer une recherche plus générale, par exemple sur le mot 'matrix', taper help.search('matrix') dans la console. Écrire dans un script et lancer les commandes suivantes. Interpréter les résultats. PLus bas, nous mettons les réponses en commentaire, après #.

Commandes de base :

```
1+1 #somme

0.1*4 #produit

a <- exp(2)# affectation de la valeur

a #retourne valeur de a

b <- cos(1); b ;# pour suite instructions
```

Creation de vecteurs :

```
v <-c(1,9,-4,0.5); v# creation vecteur

v[3] # rep [1] -4

v[1] # rep [1] 1

1:10 # [1]  1  2  3  4  5  6  7  8  9 10

seq(from=0, to=1, by=0.1)# vecteur de subdivision def par pas

seq(from=0, to=2, length=11)# vecteur de subdivision def par nb points

rep(x=1, times=10)#[1] 1 1 1 1 1 1 1 1 1 1
```

```
rep(x=1:2, times=3)#[1] 1 2 1 2 1 2

rep(x=1:2, each=4)#[1] 1 1 1 1 2 2 2 2
```

Opérations sur les vecteurs :

```
A<-c(1,2,3,6); B<-c(0,-4,9,4);

exp(A)# exp, qu'on applique coord par coord
#[1]    2.718282    7.389056   20.085537 403.428793

A+B #somme usuelle [1]  1 -2 12 10

A*B #produit de Hadamard [1]  0 -8 27 24

A%*%B # produit scalaire 43 comme matrice $1\times 1$

A/B# division terme a terme[1]          Inf -0.50  0.33  1.50


2*A#usuel

A+1#usuel

A+c(1,2,3)#avertissement mais somme completee cycliquement

A==2#vect booleens [1] FALSE  TRUE FALSE FALSE

G<-A>1; A[G]# selectionne les coordonnees >1:[1] 2 3 6

u <- (1:10); sum(u)[1] 55

cumsum(u)#sommes partielles de serie [1]  1  3  6 10 15 21 28 36 45 55
```

Création et opération sur les matrices :

```
C<-matrix(A,ncol=2,nrow=2);
```

$$C = \begin{pmatrix} 1 & 3 \\ 2 & 6 \end{pmatrix}$$

```
D<-matrix(B,ncol=2,nrow=2,byrow=TRUE);
```

$$D = \begin{pmatrix} 0 & -4 \\ 9 & 4 \end{pmatrix}$$

```
t(C)# la transposee

class(C)# [1] "matrix"
C\%*\%D #produit matriciel:
```

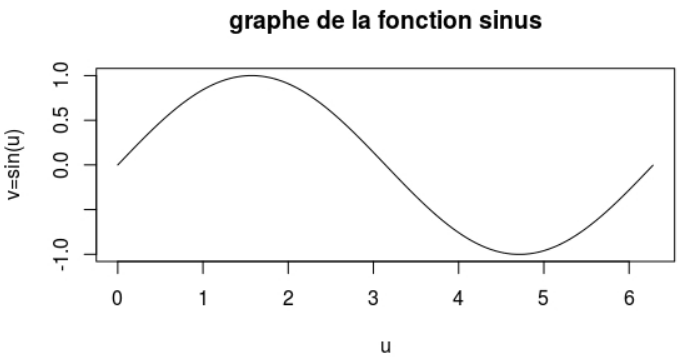
$$CD = \begin{pmatrix} 27 & 8 \\ 54 & 16 \end{pmatrix}$$

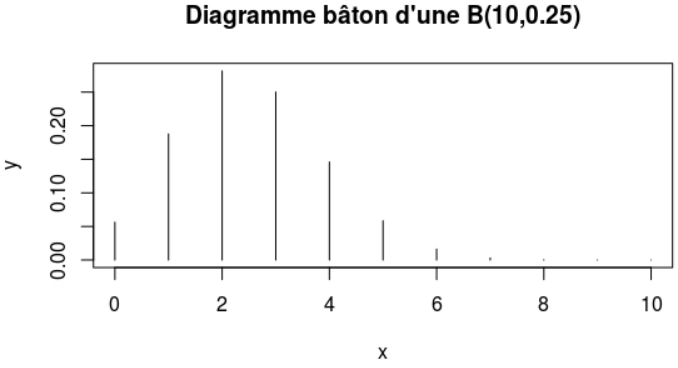
```
C[1,]# 1ere ligne de C comme vecteur colonne [1] 1 3

D/C[1,]# division colonne a colonne par 1ere ligne de C vue comme vecteur colonne

eigen(C) #valeures propres: 7 et 0
# vecteurs propres dans $vectors les donnent
#en colonne (-0.4472136,-0.8944272) et (-0.9486833,0.3162278)

# test qu'on a bien les vecteurs propres par:

E<-eigen(C)$vectors;C%*%E/E
```

eigen(**C**) renvoie une liste composée de 2 éléments le vecteur des valeurs propres et la matrice rassemblant les vecteurs propres en colonne. Donc cette matrice de passage s'obtient en prenant la deuxième composante de la liste appelé par son nom : **eigen**(**C**)**$**vectors. De même pour les valeurs propres, on tape : **eigen**(**C**)**$**values.

Tracés de graphiques :

```
u<- seq(from=0, to=2*pi, by=0.01)

v<-sin(u)

plot(x=u, y=v, type='l', ylim=c(-1,1), xlim=c(0,2*pi))


#ou mieux:

plot(x=u, y=v, type='l', ylim=c(-1,1),ylab="v=sin(u)", xlim=c(0,2*pi),
main="graphe de la fonction sinus")
```





3 Statistiques descriptives

3.1 Import de données

Pour illustrer les outils basiques de statistiques descriptives, on va charger un jeu de données, en utilisant la commande (attention au caractère ∼, obtenu avec AltGr+2)

Don<-read.delim("http ://math.univ-lyon1.fr/∼dabrowski/enseignement/ProbaL2/Donnees.csv")
attach(Don).

Les variables sexe, ronfle et tabac sont qualitatives nominales (les deux dernières sont même des variables logiques à valeur vrai ou faux), pas de variables ordinales, alcool est quantitative discrète, taille poids, age sont quantitatives continues.

Typez les variables avec les classes factor, ordered, integer ou double. Par exemple pour les variables qualitatives nominales qui étaient mal typées par R (on change aussi le nom des niveaux) :

```
sexe<-as.factor(sexe);ronfle<-as.factor(ronfle);tabac<-as.factor(tabac)
levels(ronfle)<-c("Non-ronfleur","Ronfleur")
```

3.2 Résumés numériques

En utilisant les fonctions mean, var, quantile, déterminer la moyenne empirique, la variance empirique, la variance empirique non-biaisée et l'écart type empirique pour les variables quantitatives. Déterminer la médiane pour chaque caractéristique ordinale ou quantitative de l'échantillon. On a crée une petite fonction pour calculer la variance empirique (biaisée) à partir de celle non biasée donnée par la fonction **var** de R :

```
varE<-function(x){l<-length(x);var(x)*(l-1)/l}

mean(age);varE(age);var(age);sqrt(var(age));quantile(age,type=1)
```

```
[1] 52.27
[1] 128.5971
[1] 129.8961
[1] 11.3972

   0\%   25\%   50\%   75\%  100\%
23    43    51    62    74
```

```
mean(poid);varE(poid);var(poid);sqrt(var(poid));quantile(poid,type=1)
```

```
[1] 90.41
[1] 347.5219
[1] 351.0322
[1] 18.73585

  0\%  25\%  50\%  75\% 100\%
  42    77    95   107   120
```

```
mean(taille);varE(taille);var(taille);sqrt(var(taille));quantile(taille,type=1)
```

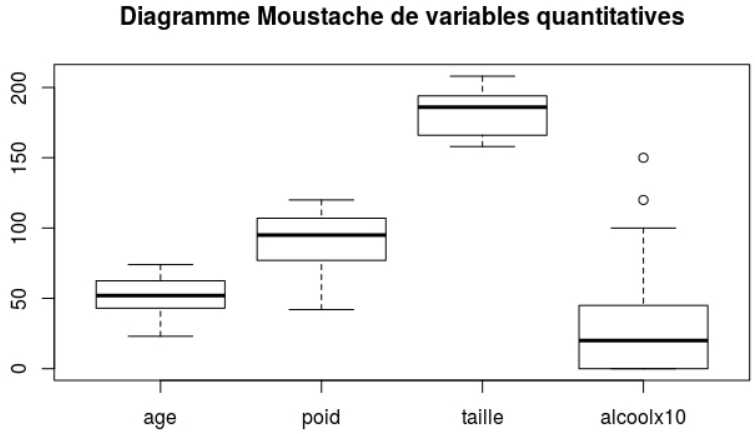
```
[1] 181.1
[1] 176.87
[1] 178.6566
[1] 13.36625

  0\%  25\%  50\%  75\% 100\%
 158   166   186   194   208
```

```
mean(alcool);varE(alcool);var(alcool);sqrt(var(alcool));quantile(alcool,type=1)
```

```
[1] 2.95
[1] 11.2075
[1] 11.32071
[1] 3.364626

   0\%   25\%   50\%   75\%  100\%
  0     0     2     4    15
```

Représenter également les données à l'aide d'un diagramme en boîte (boxplot). Rappeler les éléments représentés dans ce diagramme.

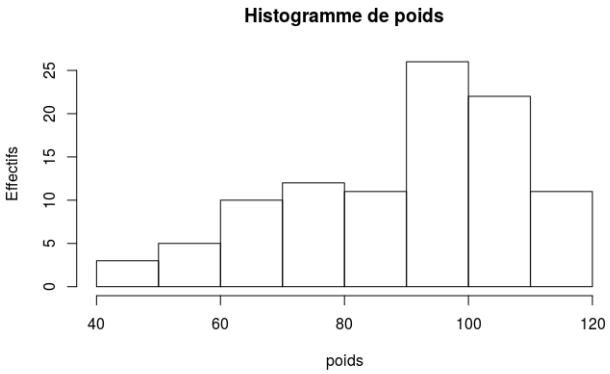
```
data<-data.frame(age,poid,taille,alcool*10)
boxplot(data,main="Diagramme Moustache de variables quantitatives")
```

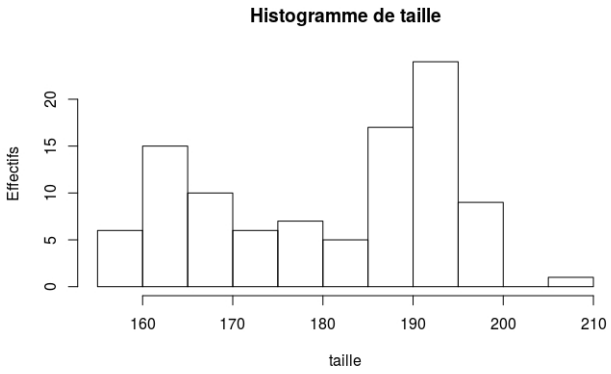


3.3 Résumés graphiques

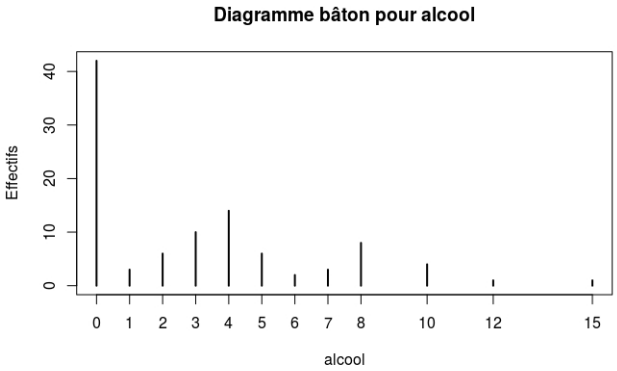
Représenter les variables quantitatives discrètes et continues à l'aide respectivement d'un diagramme en bâton et d'un histogramme (fonction hist).

```
hist(poids,main="Histogramme de poids",ylab="Effectifs")

hist(taille,main="Histogramme de taille",ylab="Effectifs")
```



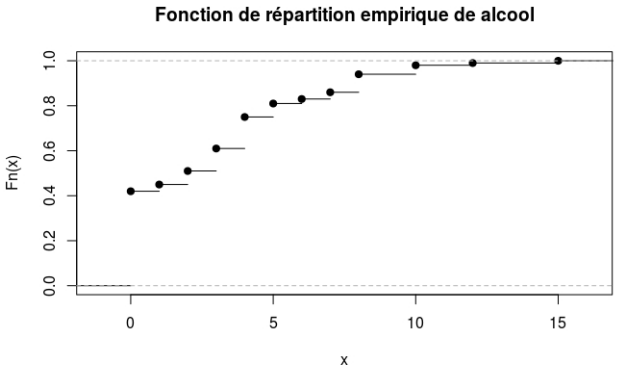


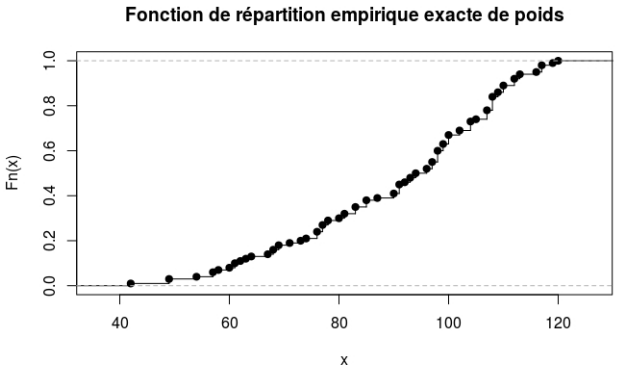

```
hist(age,main="Histogramme de age",ylab="Effectifs")

plot(table(alcool), type='h',main="Diagramme baton pour alcool",ylab="Effectifs")
```



Représentons leurs fonctions de répartitions empiriques (on peut utiliser les fonctions ecdf, plot et hist si on veut représenter la fonction de répartition de l'histogramme pour les variables continues).

```
plot(ecdf(alcool),verticals=FALSE,
main="Fonction de repartition empirique de alcool")

plot(ecdf(poids),verticals=TRUE,
main="Fonction de repartition empirique exacte de poids")
```



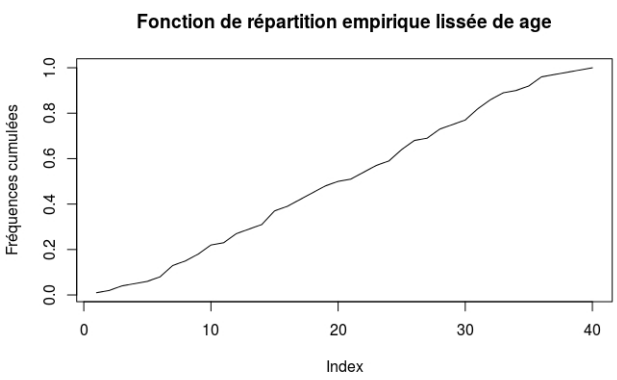


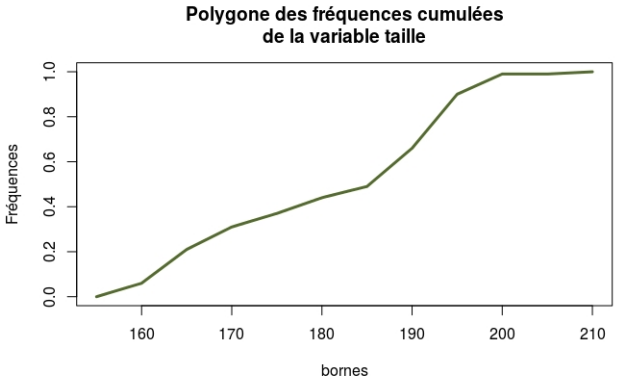
Au lieu de tracer les fonctions de répartitions exactes, ce qui revient à considérer les variables continues comme discrètes, on a aussi les options suivantes (qui n'ont pas forcément été vues en TP) :

```
TCage=table(age);
plot(cumsum(TCage)/sum(TCage),type='l',
main="Fonction de repartition empirique lissee de age",ylab="Frequences cumulees")

bornes <- hist(taille,right=T,plot=F)$breaks

plot (bornes,ecdf(taille) (bornes),type="l",
main=paste("Polygone des frequences cumulees", "de la variable taille",
sep="\textbackslash n"),ylab="Frequences",col="darkolivegreen",lwd=3)
```

Ici on a representé la fonction de répartition pour une densité donnée par l'histogramme, elle est linéaire par morceau, le nom de cette représentation est en titre. Pour cela on a retenu le vecteur de la subdivision (breaks) calculée par défaut par hist. on a ensuite tracée une courbes continue reliant les valeurs prises par la fonction de répartition à ces points (calculé par ecdf(taille) (bornes) qui applique

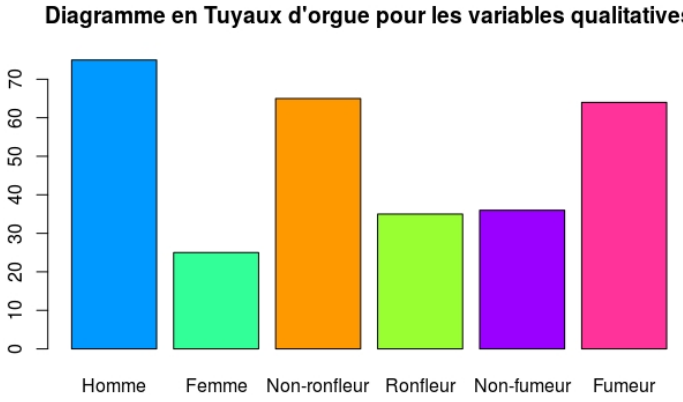
la fonction ecdf(taille) au vecteur des bornes).

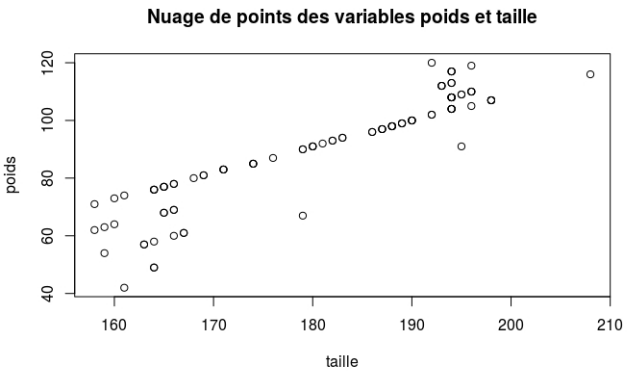




Représenter les variables qualitatives par un diagramme en tuyaux d'orgue (fonction barplot).

```
col <- c("#0099FF", "#33FF99","#FF9900",
 "#99FF33","#9900FF", "#FF3399","#FF0099", "#9933FF")

levels(sexe)<-c("Homme","Femme")

levels(ronfle)<-c("Non-ronfleur","Ronfleur")

levels(tabac)<-c("Non-fumeur","Fumeur")

barplot(c(table(sexe),table(ronfle),table(tabac)),
main="Diagramme en Tuyaux d'orgue pour les variables qualitatives",col=col)
```





Donner les tables de contingences pour les variables sexes et tabagisme.

```
> table(sexe,tabac)
        tabac
sexe  Non-fumeur  Fumeur
 Homme          21      54
 Femme          15      10
```

Représenter par un nuage de point les variables poids et taille. (image ci-dessus)

plot(poids~taille,main="Nuage de points des variables poids et taille")

4 Lois usuelles discrètes

4.1 Diagrammes en bâtons

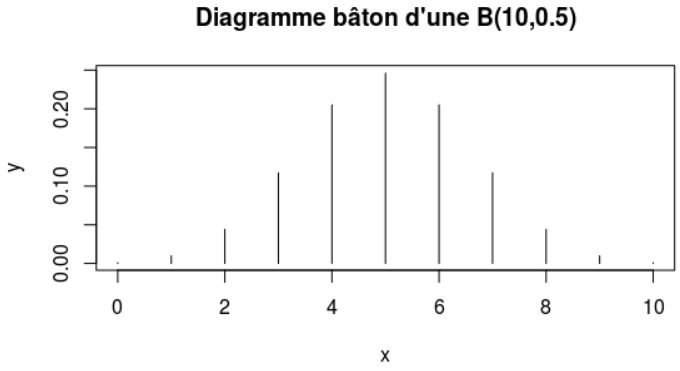
Pour tracer le diagramme en bâtons décrivant de la loi binomiale $\mathcal{B}(10, 0.25)$, on peut utiliser les commandes suivantes (résultat ci-dessus) :

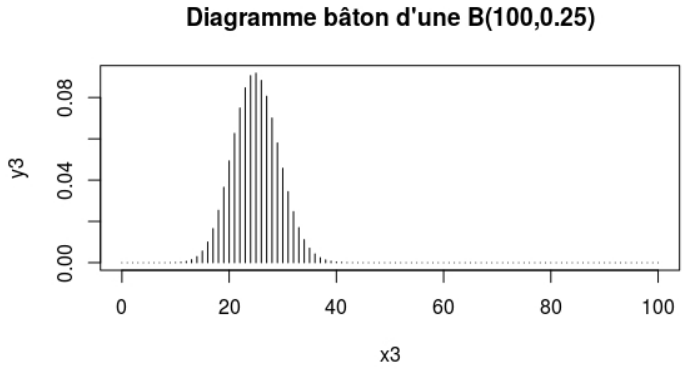
```
x <- 0:10

y <- dbinom(x, size=10, prob=0.25)

plot(x, y, type='h')
```

Tracer de même un diagramme en bâton des lois $\mathcal{B}(10, 0.5)$, $\mathcal{B}(100, 0.25)$, $\mathcal{P}(2)$, $\mathcal{P}(10)$, $\mathcal{G}(0.75)$, $\mathcal{G}(0.25)$ et de la loi uniforme sur $\{1, 2, 3, 4, 5, 6\}$.

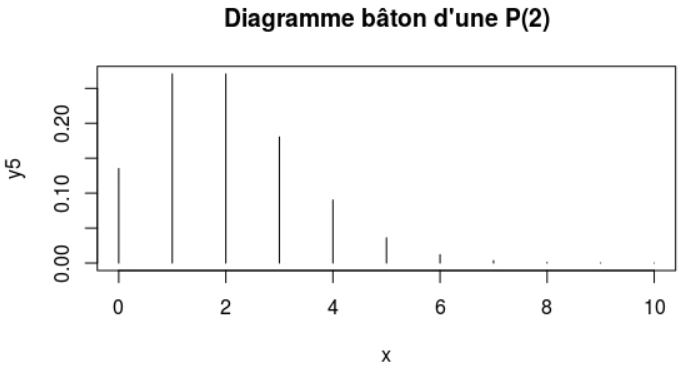
Réponse :

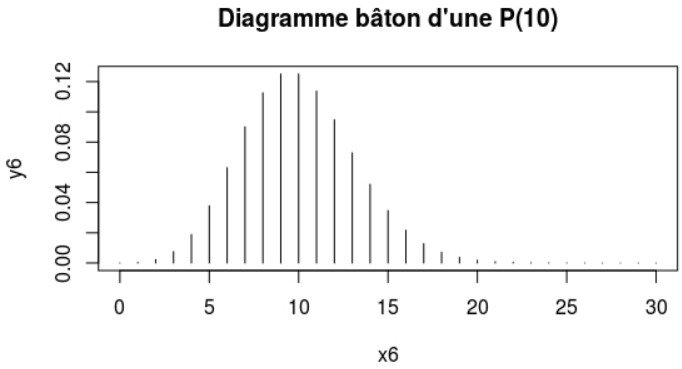
```
x <- 0:10

y <- dbinom(x, size=10, prob=0.5)

plot(x, y, type='h',main="Diagramme baton d'une B(10,0.5)")


x3 <- 0:100

y3 <- dbinom(x3, size=100, prob=0.25)

plot(x3, y3, type='h')
```



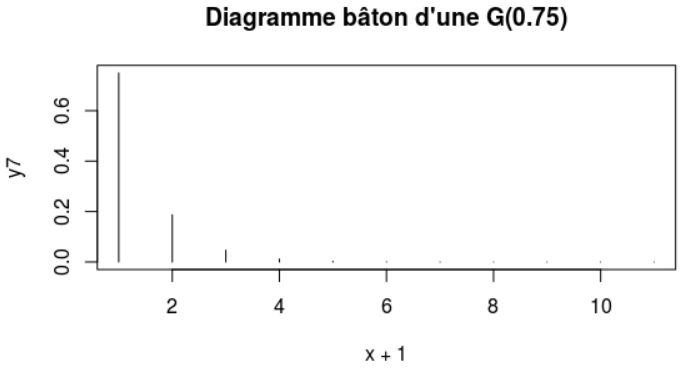


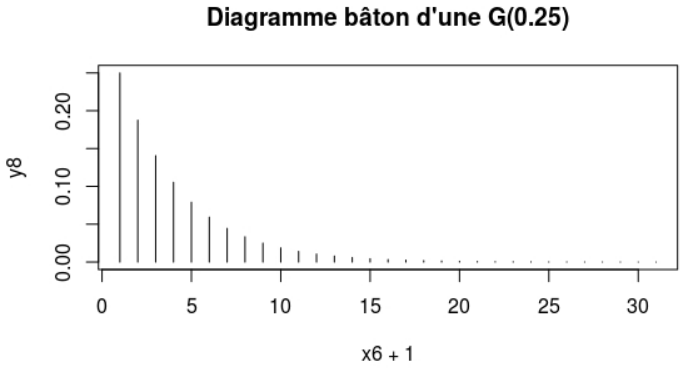

```
y5<-dpois(x, lambda=2, log = FALSE)

plot(x, y5, type='h')

x6 <- 0:30

y6<-dpois(x6, lambda=10, log = FALSE)

plot(x6, y6, type='h')
```



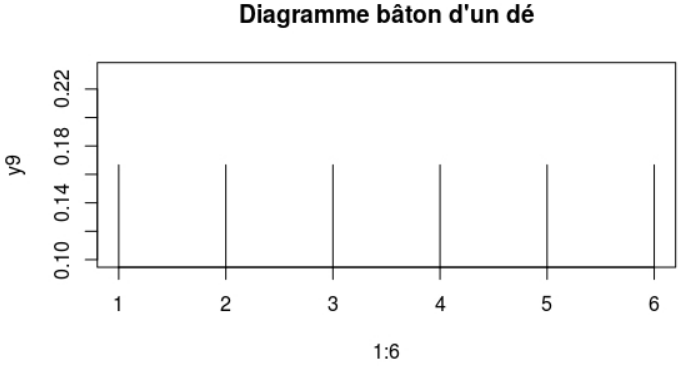


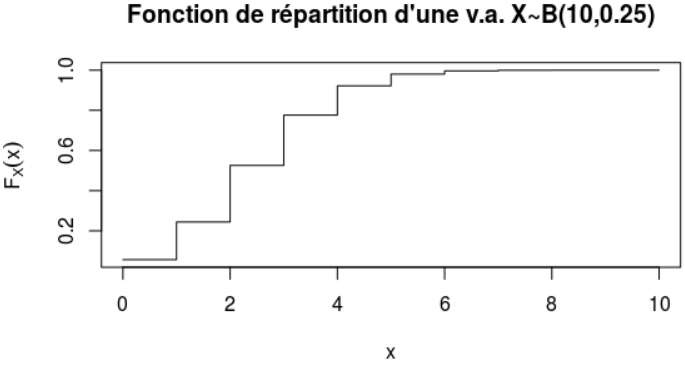

```
y7<-dgeom(x, prob=0.75, log = FALSE)
plot(x+1, y7, type='h')
y8<-dgeom(x6, prob=0.25, log = FALSE)
plot(x6+1, y8, type='h')
```






```
y9=rep(x=1/6, times=6)
plot(1:6, y9, type='h',main="Diagramme baton d'un de")
```

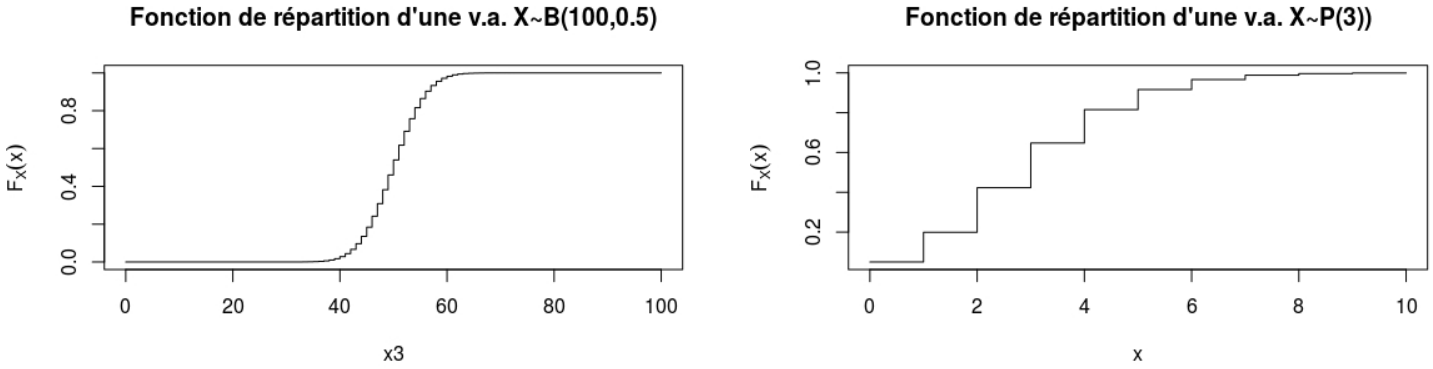




4.2 Fonctions de répartition

Tracer la fonction de répartition de la loi $\mathcal{B}(10, 0.25)$ (on pourra utiliser la fonction pbinom et l'option type='s' de la fonction plot). Tracer de même la fonction de répartition des lois $\mathcal{B}(100, 0.5)$ et $\mathcal{P}(3)$.

```
yA<-pbinom(x, size=10, prob=0.25)

plot(x,yA,main="Fonction de repartition d'une v.a. X~B(10,0.25)",
ylab=expression(F[X] (x)),type='s')

yB<-pbinom(x3, size=100, prob=0.5)

plot(x3,yB,main="Fonction de repartition d'une v.a. X~B(100,0.5)",
ylab=expression(F[X] (x)),type='s')

yC<-ppois(x, lambda=3)

plot(x,yC,main="Fonction de repartition d'une v.a. X~P(3))",
ylab=expression(F[X] (x)),type='s')
```



Que valent $P(X \leq 3)$ et $P(X > 30)$ si X suit la loi $\mathcal{B}(50, 0.2)$?

```
pbinom(3, size=50, prob=0.2)

[1] 0.005656361

1-pbinom(30, size=50, prob=0.2)

[1] 1.102433e-10
```

4.3 Lien binomiale/Poisson

Rappelons le théorème de convergence de la loi binomiale vers la loi de Poisson. Si $X_n \sim \mathcal{B}(n, \lambda/n)$ alors $P(X_n = k) \to_{n\to\infty} P(X = k)$ avec $X \sim P(\lambda)$

Illustrons cette convergence en représentant sur un même graphique les lois $\mathcal{B}(500, 0.02)$ et $\mathcal{P}(10)$.

```
yD <- dbinom(x6, size=500, prob=0.02)

plot(x6, yD, type='h',
main="Illustration thm limite de Poisson B(500,0.02) proche P(10)")

points(x6,dpois(x6, lambda=10))
```

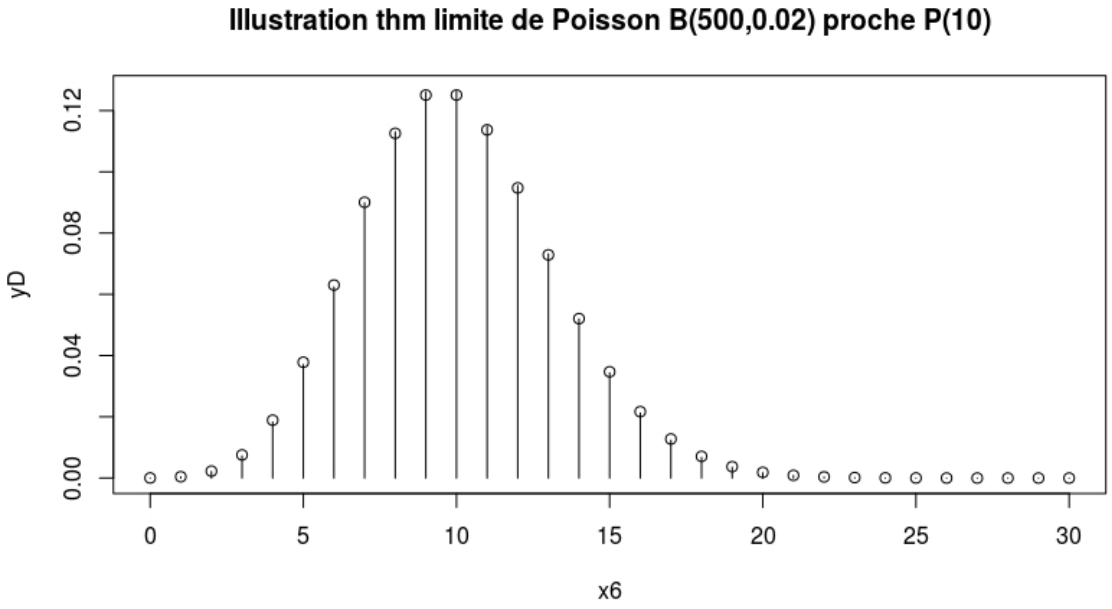


Illustration thm limite de Poisson B(500,0.02) proche P(10)
yD
0.00
0.04
0.08
0.12
0
5
10
15
20
25
30
x6