HAI718 Probabilité et statistiques

Introduction à R et lois usuelles
Correction

1 Commencer avec R

1.1 Créér, lister et effacer des données en mémoire

L'assignation d'une valeur à une variable se réalise grâce à la flèche : `->`, dans un sens ou dans l'autre.

Exercice 1 *Observer ce que produisent les commandes suivantes tapées successivement :*

```
> n <- 15
# Assigne la valeur 15 à la variable n
> 5 -> n
# Assigne la valeur 5 à la même variable n
> x <- 1
# Assigne la valeur 1 à la variable x
> X <- 10
# Assigne la valeur 10 à la variable X différente de x
# (R est dépendant de la casse)
> n
[1] 5
# Évalue la variable n
> x
[1] 1
# Évalue la variable x
> X
[1] 10
# Évalue la variable X
> n <- 10 + 2
# Assigne la valeur du calcul 10 + 2 à la variable n
> n
[1] 12
# Évalue la variable n
> n <- 3 + rnorm(1)
# Assigne à n la valeur 3 + une valeur aléatoire suivant la loi N(0,1)
> (10 + 2)*5
[1] 60
# Évalue le calcul
> name <- "Carmen"; n1 <- 10; n2 <- 100; m<- 0.5
# Séquence d'instructions séparées par des ;
> ls()
[1] "m"    "n"    "n1"   "n2"   "name" "x"    "X"
# Indique quels sont les objets présents dans la mémoire
```

La fonction `ls.str()` fournit des informations complémentaires sur les objets, et la fonction `ls(pat = "m")` va donner uniquement les objets dont les noms comportent un "m".

Exercice 2

1. *Testez les fonctions* `ls` *et* `ls.str` *avec différents patterns.*

```
> ls(pat="name")
[1] "name"
> ls.str()
m :  num 0.5
n :  num 2.19
n1 :  num 10
n2 :  num 100
name :  chr "Carmen"
x :  num 1
X :  num 10
> ls.str(pat="n")
n :  num 2.19
n1 :  num 10
n2 :  num 100
name :  chr "Carmen"
> ls.str(pat="m")
m :  num 0.5
name :  chr "Carmen"
```

2. *Que fait la sequence d'instructions suivante :*

```
> M <- data.frame(n1, n2, m)
> ls.str(pat = "M")
M : 'data.frame':       1 obs. of  3 variables:
 $ n1: num 10
 $ n2: num 100
 $ m : num 0.5
> ls.str(pat = "M" , max.level = -1)
# produit un message d'erreur...
```

3. *Effacez tous les objets en mémoire à l'aide de la fonction* `rm`

```
> ls()
[1] "m"    "M"    "n"    "n1"   "n2"   "name" "x"    "X"
> rm("m","M","n","n1","n2","name","x","X")
> ls()
character(0)
```

1.2 L'aide en ligne

La fonction `help()` permet d'obtenir de l'aide sur la façon d'obtenir de l'aide... Pour résumer, il y a deux façons d'avoir une aide sur une fonction donnée, en tapant la commande précédée d'un point d'interrogation, ou en donnant la commande comme paramètre de l'aide.

Exercice 3 *Recherchez de l'aide sur la fonction* `rm`*. Comment détruit-on en une seule commande tous les objets de la session ? Comment détruit-on seulement les objets qui ont un "m" dans leur nom ? Testez l'aide en ligne* `help.start()`*.*

```
> ls()
[1] "m"    "M"    "n"    "n1"   "n2"   "name" "x"    "X"
```

```
> rm(list=ls(pat="m"))
> ls()
[1] "M"  "n"  "n1" "n2" "x"  "X"
> rm(list=ls())
> ls()
character(0)
```

Dans ce qui suit, quand on spécifiera une fonction à utiliser pour résoudre un exercice, prenez l'habitude d'invoquer l'aide sur cette fonction pour comprendre son utilisation.

1.3 Tracer des graphiques

Lorsqu'une fonction graphique est utilisée en R, si aucune fenêtre graphique n'a déjà été créée, R en ouvre une automatiquement. Le dernier périphérique ouvert devient le prériphérique actif dans lequel s'affichent les graphes suivants. La fonction `dev.list()` affiche la liste des périphériques ouverts. La liste des types de périphériques graphiques disponibles est accessible par `?device`.

Exercice 4 *Que font les commandes suivantes :*

```
> x11(); x11(); pdf();
# Ouvre deux fenêtres graphiques et une ressource pdf que l'on ne voit pas
> dev.list()
X11 X11 pdf
  2   3   4
# donne la liste des périphériques
> dev.cur()
pdf
  4
# donne le numéro du périphérique actif
> dev.set(3)
X11
  3
# rend le périphérique 3 actif
> dev.cur()
X11
  3
# on vérifie que c'est bien le périphérique 3 qui est actif
> dev.off(2)
X11
  3
# on ferme le périphérique 2, et affiche le périphérique actif
> dev.list()
X11 pdf
  3   4
# on vérifie que le périphérique 2 est bien fermé
> dev.off()
pdf
  4
# on ferme le périphérique actif (3), le 4 devient actif
> dev.off()
null device
          1
```

```
# on ferme le périphérique actif (4), il n'y en a plus
```

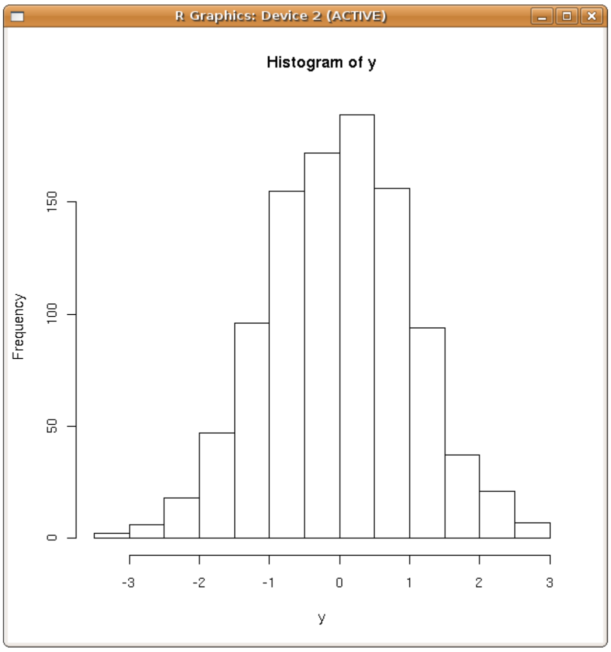
Les tracés de graphes se font à l'aide différentes fonctions. Nous utiliserons essentiellement `plot` et `hist`.

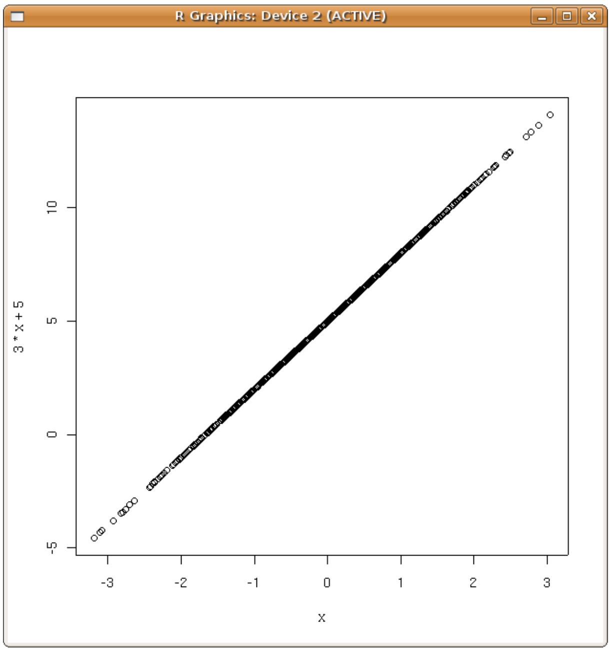
Exercice 5 *Après avoir lu l'aide relative à ces deux fonctions, définissez deux vecteurs de 1000 valeurs* `x <- rnorm(1000)` *et* `y <- rnorm(1000)`*, et tracez l'histogramme des fréquences de* `y` *puis les valeurs de* 3x+5 *en fonction des valeurs de* `x`*. Testez ensuite :*

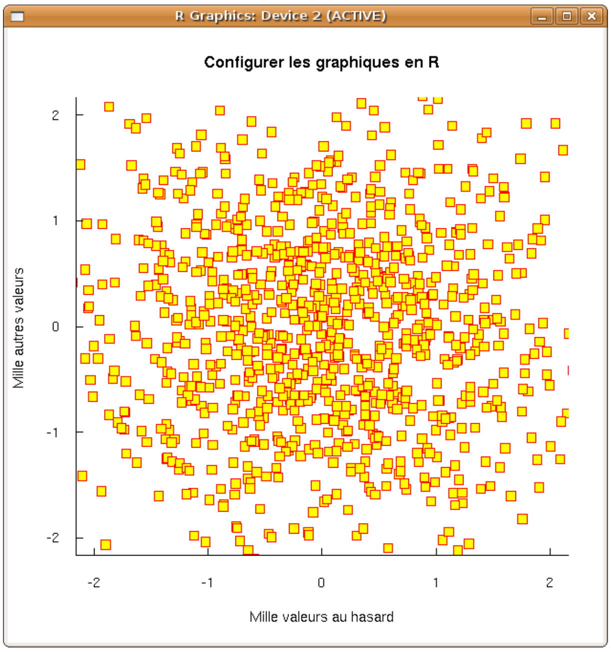
```
plot(x,y,xlab="Mille valeurs au hasard", ylab="Mille autres valeurs",
     xlim=c(-2,2), ylim=c(-2,2), pch=22, col="red", bg="yellow", bty="l",
     tcl=0.4, main="Configurer les graphiques en R", las=1, cex=1.5)
```

```
> x <- rnorm(1000)
> y <- rnorm(1000)
> hist(y)
> plot(x,3*x+5)
> plot(x,y,xlab="Mille valeurs au hasard", ylab="Mille autres valeurs",
+      xlim=c(-2,2), ylim=c(-2,2), pch=22, col="red", bg="yellow", bty="l",
+      tcl=0.4, main="Configurer les graphiques en R", las=1, cex=1.5)
```

Voici les trois graphiques qui apparaissent successivement :



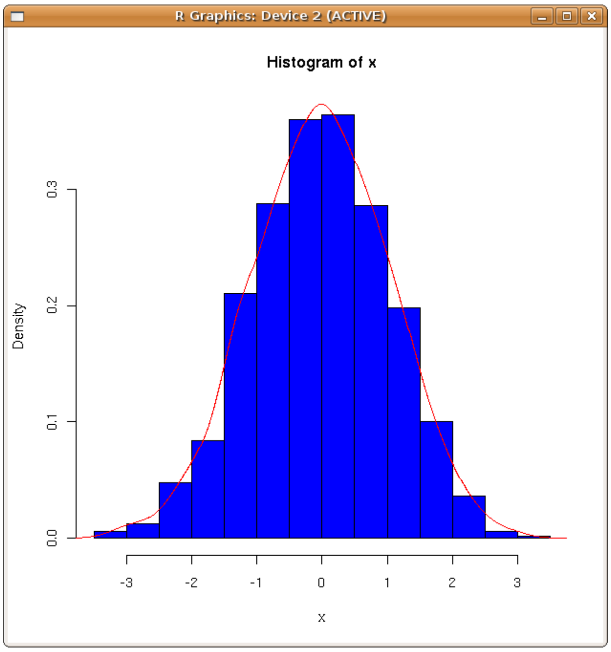




Pour mettre deux graphiques sur la même courbe, on peut utiliser `points` et `lines`

Exercice 6 *Mettre sur un même graphique en bleu l'histogramme des propbabilités de* `x` *(paramètre* `probability=T` *de* `hist`*) et en rouge sa densité (fonction* `density`*).*

```
> hist(x, probability=T, col="blue")
> lines(density(x), col="red")
```



2 Lois binomiale, normale, de Student

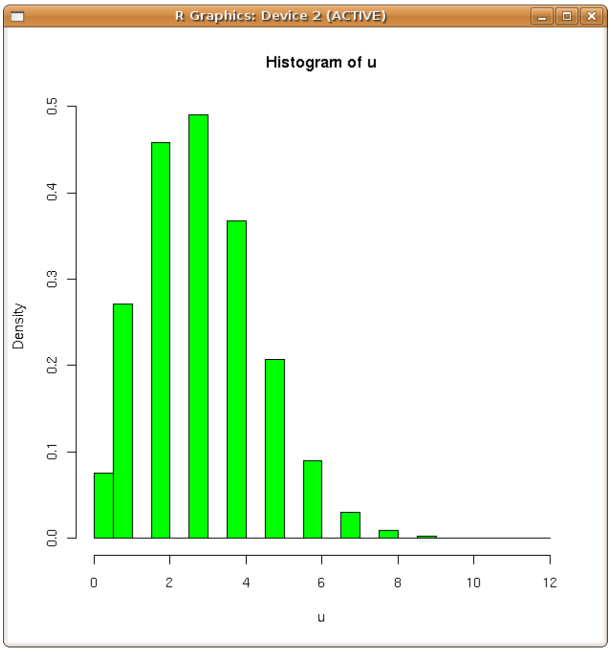
2.1 La loi binomiale $\mathcal{B}(n, p)$

Soit $X \sim \mathcal{B}(n, p)$ avec $n = 18$ et $p = 1/6$.

Exercice 7 *Calculer $P(X = 3)$, $P(X \leq 3)$, $P(X \geq 3)$, $P(X \leq 16)$ à l'aide de la fonction pbinom().*

```
> # P(X = 3)
> pbinom(3,18,1/6) - pbinom(2,18,1/6)
[1] 0.2451984
> # P(X <= 3)
> pbinom(3,18,1/6)
[1] 0.6478528
> # P(X >= 3)
> 1 - pbinom(2,18,1/6)
[1] 0.5973457
> # P(X <= 16)
> pbinom(16,18,1/6)
[1] 1
```

Remarque : Cette dernière valeur est FAUSSE... mais elle est si proche de 1 (cf. TD) que le système l'arrondit à 1. Pour information, voici à quoi ressemble la loi de cet exercice :



2.2 La loi normale $\mathcal{N}(m, \sigma)$

Exercice 8 1. *Soit $U \sim \mathcal{N}(0, 1)$ la loi normale centrée réduite.*

(a) *À l'aide de la fonction pnorm(), calculer $P(U < 1.41)$, $P(U < -2.07)$, $P(U > -1.26)$.*

```
> # P(U < 1.41)
> pnorm(1.41)
[1] 0.9207302
> # P(U < -2.07)
> pnorm(-2.07)
[1] 0.01922617
> # P(U > -1.26)
> pnorm(-1.26, lower.tail=FALSE)
[1] 0.8961653
> # ou...
> 1 - pnorm(-1.26)
[1] 0.8961653
```

(b) *À l'aide de la fonction qnorm(), trouver la valeur de u telle que $P(U < u) = 0.95$, $P(U < u) = 0.1$, $P(U > u) = 0.99$.*

```
> # P(U < u) = 0.95
> qnorm(0.95)
[1] 1.644854
> # P(U < u) = 0.1
> qnorm(0.1)
[1] -1.281552
> # P(U > u) = 0.99 => P(U < u) = 1 - 0.99 = 0.01
> qnorm(0.01)
[1] -2.326348
```

2. *Soit* $X \sim \mathcal{N}(m, \sigma^2)$ *avec* $m = -5$ *et* $\sigma = 4$.

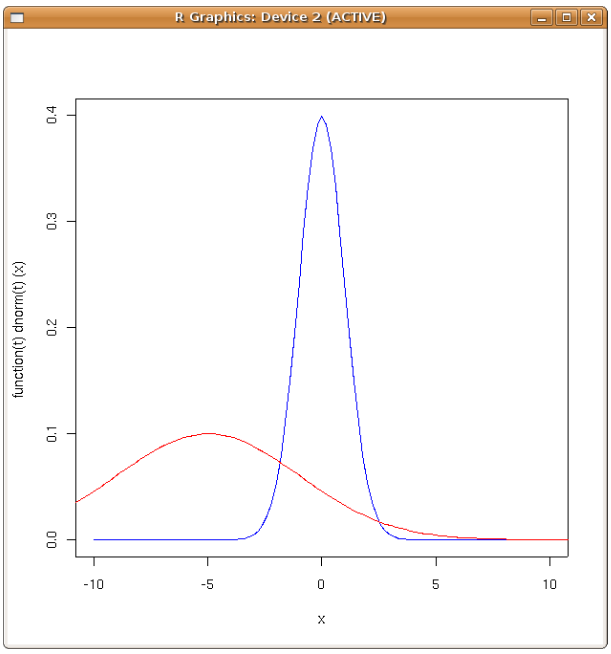
   (a) *Calculer* $P(X < -5)$, $P(X \leq 0)$, $P(X \geq 5)$.

```
> # P(X < -5)
> pnorm(-5,-5,4)
[1] 0.5
> # P(X <= 0)
> pnorm(0,-5,4)
[1] 0.8943502
> # P(X >= 5)
> pnorm(5,-5,4,lower.tail=FALSE)
[1] 0.006209665
```

   (b) *Trouver la valeur de* x *telle que* $P(X < x) = 0.95$, $P(X < x) = 0.05$, $P(X > x) = 0.01$.

```
> # P(X < x) = 0.95
> qnorm(0.95,-5,4)
[1] 1.579415
> # P(X < x) = 0.05
> qnorm(0.05,-5,4)
[1] -11.57941
> # P(X > x) = 0.01 => P(X < x) = 1 - 0.01 = 0.99
> qnorm(0.99,-5,4)
[1] 4.305391
```

Pour information, voici à quoi ressemblent les deux lois de cet exercice :



2.3 La loi du Chi-deux χ^2 (ou loi de Pearson)

$U_1, \ldots, U_p$ étant p variables $\mathcal{N}(0,1)$ indépendantes, on appelle loi du chi-deux à p degrés de liberté (χ^2_p) la loi de la variable $\sum_{i=1}^{p} U_i^2$.

Exercice 9

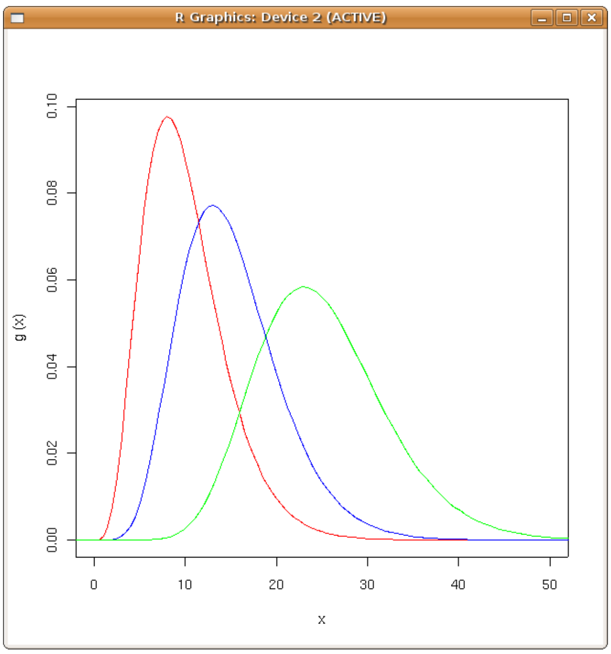
1. *Soit $X \sim \chi^2_{15}$ et $Y \sim \chi^2_{10}$. À l'aide de la fonction* `pchisq`*, calculer* $P(X < 6.26)$, $P(Y > 3.25)$, $P(X + Y > 11.52)$. On rappelle qu'une somme de variables aléatoire de lois chi-deux à p et q degrés de liberté respectivement est une variable aléatoire de loi chi-deux à $p+q$ degrés de liberté.

```
> # P(X < 6.26)
> pchisq(6.26,15)
[1] 0.02495843
> # P(Y > 3.25)
> pchisq(3.25,10,lower.tail=FALSE)
[1] 0.9749135
> # P(X+Y > 11.52)
> pchisq(11.52,25,lower.tail=FALSE)
[1] 0.9900255
```

2. *Soit $X \sim \chi^2_{15}$. À l'aide de la fonction* `qchisq`*, trouver x tel que $P(X < x) = 0.01$, $P(X < x) = 0.05$, $P(X < x) = 0.99$.*

```
> # P(X < x) = 0.01
> qchisq(0.01,15)
[1] 5.229349
> # P(X < x) = 0.05
> qchisq(0.05,15)
[1] 7.260944
> # P(X < x) = 0.99
> qchisq(0.99,15)
[1] 30.57791
```

Pour information, voici à quoi ressemblent les trois lois de cet exercice (10 en rouge, 15 en bleu et 25 en vert) :



2.4 La loi de Student T_n

Soit une variable aléatoire $U \sim \mathcal{N}(0,1)$ et X une variable aléatoire suivant indépendamment de U une loi χ^2_n. On définit alors la variable de Student T_n à n degrés de liberté comme étant :

$$T_n = \frac{U}{\sqrt{\frac{X}{n}}}.$$

On note que la loi de Student T_n est symétrique, cela signifie que :

$$\forall x \; P(T_n < -t) = P(T_n > t).$$

De cette propriété, il découle que pour tout $x > 0$:

$$\text{si } P(|T_n| < t) = p \text{ alors } P(T_n < -t) = p/2 \text{ et } P(T_n > t) = p/2.$$

Exercice 10 *Soit $T \sim T_5$ une loi de Student à 5 degrés de liberté.*

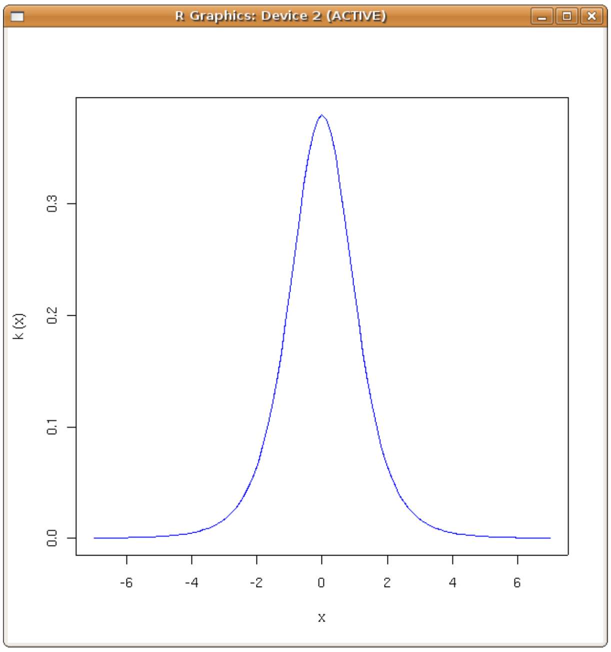
1. *À l'aide de la fonction pt(), calculer $P(T < 0.408)$, $P(T < -2.07)$, $P(T > 0.132)$.*

```
> # P(T < 0.408)
> pt(0.408,5)
[1] 0.6499213
> # P(T < -2.07)
> pt(-2.07,5)
[1] 0.04661982
> # P(T > 0.132)
> pt(0.132,5,lower.tail=FALSE)
[1] 0.4500658
```

2. *À l'aide de la fonction qt(), trouver la valeur de t telle que $P(T < t) = 0.05$, $P(T > t) = 0.9$, $P(T < t) = 0.5$.*

```
> # P(T < t) = 0.05
> qt(0.05,5)
[1] -2.015048
> # P(T > t) = 0.9 => P(T < t) = 1 - 0.9 = 0.1
> qt(0.1,5)
[1] -1.475884
> # P(T < t) = 0.5
> qt(0.5,5)
[1] 0
```

Pour information, voici à quoi ressemble la loi de cet exercice :



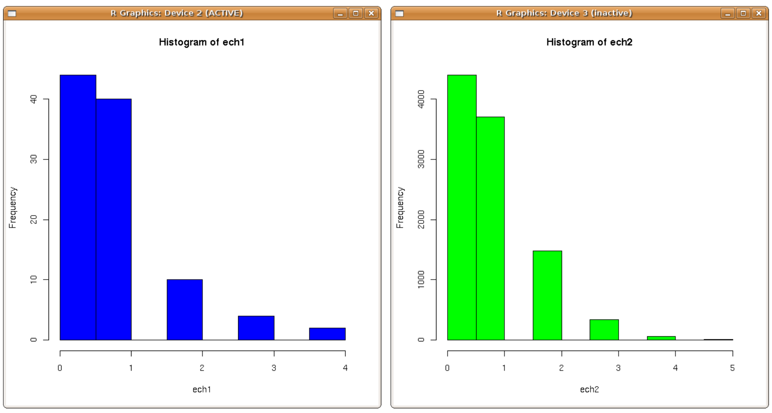
3 Simulation des lois binomiale, normale, de Student

3.1 La loi binomiale $\mathcal{B}(n, p)$

Exercice 11 *Soit* $X \sim \mathcal{B}(n, p)$ *avec* $n = 20$ *et* $p = 0.04$.
À l'aide de la fonction rbinom(), simuler un échantillon de taille 100 *et* 100000 *de la loi de* X *et représenter les histogrammes pour chaque échantillon à l'aide de la fonction hist().*

```
> ech1 <- rbinom(100,20,0.04)
> ech2 <- rbinom(10000,20,0.04)
> hist(ech1,col="blue")
> x11()
> hist(ech2,col="green")
```

Voici les deux histogrammes obtenus :



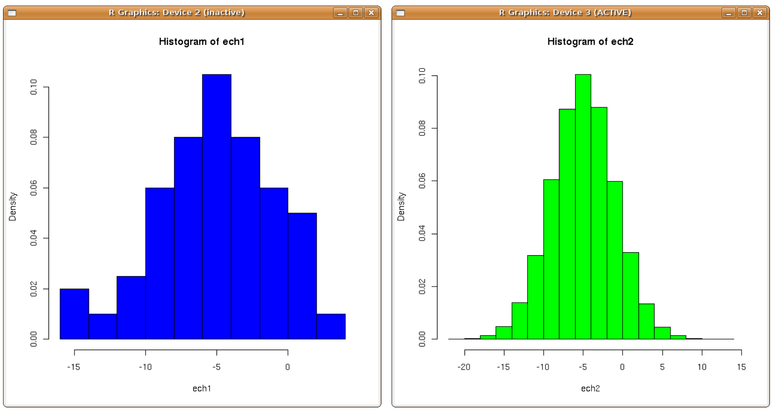
3.2 La loi normale $\mathcal{N}(m, \sigma)$

Exercice 12 *Soit $X \sim \mathcal{N}(m, \sigma)$ avec $m = -5$ et $\sigma = 4$.*

1. *À l'aide de la fonction rnorm(), simuler un échantillon de taille $n = 100$, et $n = 100000$ de la loi X et représenter les histogrammes pour chaque échantillon.*

```
> dev.set(2)
X11
  2
> ech1 <- rnorm(100,-5,4)
> ech2 <- rnorm(100000,-5,4)
> hist(ech1,probability=T,col="blue")
> dev.set(3)
X11
  3
> hist(ech2,probability=T,col="green")
```

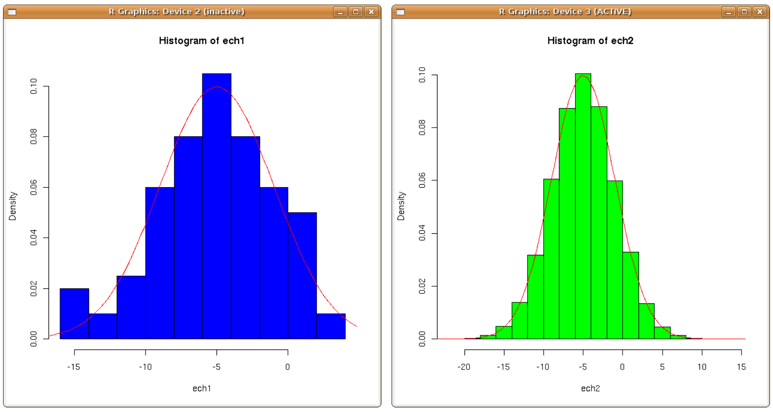
Voici les deux histogrammes obtenus :



2. *À l'aide des fonctions dnorm() et points(), représenter la fonction de densité de la variable X sur l'histogramme.*

```
> dev.set(2)
X11
  2
> f <- function (t) dnorm(t,-5,4)
> curve(f,add=T,col="red")
> dev.set(3)
X11
  3
> curve(f,add=T,col="red")
```

Voici les deux graphiques obtenus :



On remarque, comme on s'y attend, que plus l'échantillonage est important, meilleure est l'approximation que l'on a de la loi théorique.

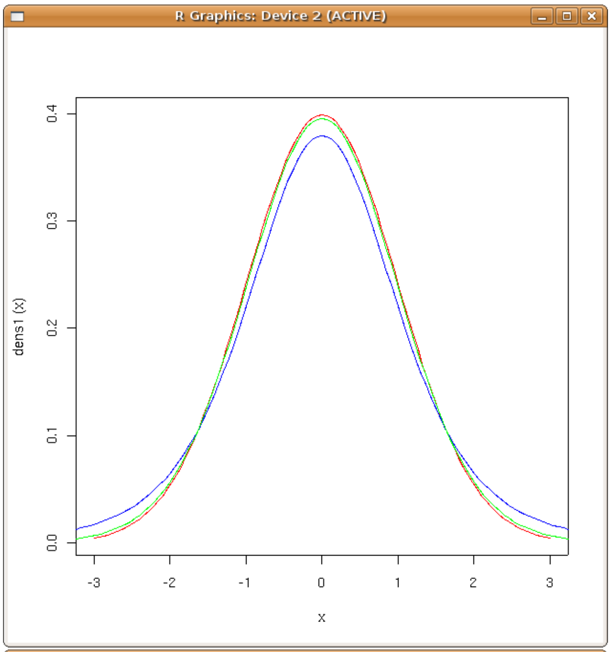
3.3 Comparaison de la $\mathcal{N}(0,1)$ et de la T_n

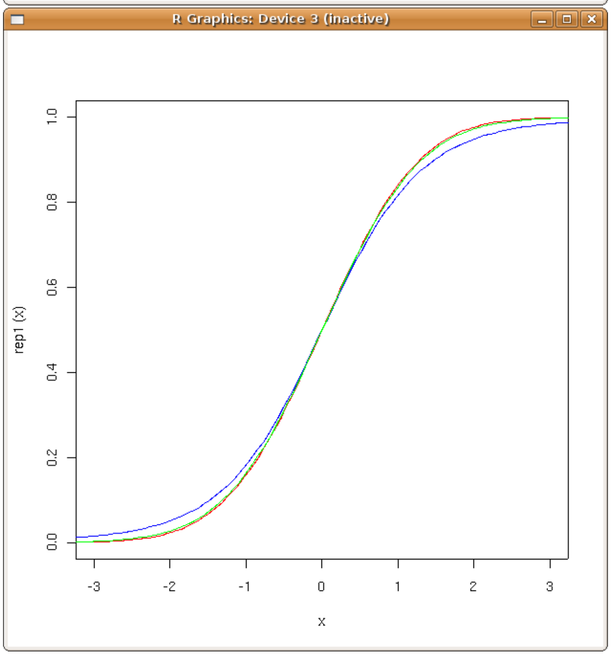
À l'aide de graphiques, on souhaite comparer la loi normale (centrée réduite) et la loi de Student à n degrés de liberté pour différentes valeurs de n ($n = 5$, $n = 30$). Pour cela, on utilisera les deux fonctions *dt()* pour calculer la densité d'une loi de Student et *pt()* pour calculer une probabilité.

Exercice 13 *On comparera sur un même graphique les fonctions de densité pour la loi normale et les lois de Student. Faire de même avec les fonctions de répartition (la fonction de répartition de la variable X est la fonction $F(x) = P(X \leq x)$). Qu'en déduisez-vous ?*

```
> dens1 <- function (t) dnorm(t)
> dens2 <- function (t) dt(t,5)
> dens3 <- function (t) dt(t,30)
> rep1 <- function (t) pnorm(t)
> rep2 <- function (t) pt(t,5)
> rep3 <- function (t) pt(t,30)
> dev.set(2)
X11
  2
> plot(dens1,-3,3,col="red")
> curve(dens2, add=T, col="blue")
> curve(dens3, add=T, col="green")
> dev.set(3)
X11
  3
> plot(rep1,-3,3,col="red")
> curve(rep2, add=T, col="blue")
> curve(rep3, add=T, col="green")
```

Voici les deux graphiques obtenus :





On en déduit, bien sûr, que les lois se ressemblent, et cela d'autant plus que le degré de liberté de la loi de Student est élevé. On va donc pouvoir se permettre d'approximer une loi de Student à grand degré de liberté par une loi normale centrée réduite.

4 Le théorème centrale limite

cf. TP suivant...