ATELIER A4 DU DISCRET AU CONTINU DANS LE CAS PARTICULIER DU THÉORÈME DE MOIVRE-LAPLACE

MISE EN OEUVRE AVEC R
LOGICIEL PROFESSIONNEL LIBRE DE STATISTIQUE

Hubert Raymondaud[*]* ; Anne Perrut[**]**

L'atelier s'est déroulé en trois activités :

1° Exemple d'une séance de travaux pratiques d'algorithmique illustrant le théorème de Moivre-Laplace, en classe de terminale.
R est utilisé pour illustrer graphiquement la distribution d'une variable binomiale X, que l'on habille avec des rectangles (qu'on évite d'appeler histogramme, terme réservé à la représentation graphique d'une série "observée") puis pour illustrer la distribution de la variable centrée réduite correspondante et enfin pour observer ce qui se passe quand n augmente. Le passage à la variable centrée réduite entraine le passage à la densité pour la graduation des ordonnées, ce qui permet la superposition de la courbe de densité de la loi de Gauss centrée réduite.
On passe alors des lignes de commande **R** à une fonction **R**. **R** est utilisé sous RStudio qui est un environnement facilitant son utilisation (coloration syntaxique, complétion automatique, visualisation des objets créés en mémoire, rémanence des 40 derniers graphiques créés ...).

2° Illustration de la fluctuation d'échantillonnage et de la loi des grands nombres par la recherche d'un algorithme et sa mise en œuvre d'une simulation avec **R**.
On simule une série de n variables de Bernoulli dont on fait les sommes cumulées croissantes, puis les fréquences cumulées croissantes en divisant par les effectifs cumulés croissants. On obtient ainsi n simulations non indépendantes d'échantillons d'effectifs croissants de 1 à n. Les fréquences obtenues sont illustrées par un graphique en lignes brisées ou, mieux, en nuages de points, pour mettre l'accent sur l'aspect discret de la variable effectif.
Cette activité permet d'illustrer l'efficacité de l'utilisation des couleurs avec **R**.
Un prolongement possible consisterait à simuler des échantillons indépendants et à illustrer la suite des distributions ainsi obtenues. (cf. La Barrolière)

3° Exploration des intervalles de fluctuation de la seconde à la terminale.
L'activité consiste à analyser et à mettre en œuvre un algorithme permettant de calculer et d'illustrer graphiquement la probabilité binomiale "exacte" de séries d'intervalles de fluctuation asymptotiques de seconde et de terminale, en fonctions de la valeur de n.
Les représentations graphiques permettent de visualiser la complexité de la relation liant la probabilité binomiale à la valeur de n. C'est la conséquence de la "discrétisation" autrement dit le fait de passer d'une fréquence à une valeur entière de la variable binomiale correspondante.
L'activité se termine par la réalisation en **R**, des fameuses abaques, représentant les trois "types" d'IF (seconde, première et terminale) en fonction de p. Ces abaques électroniques permettent d'illustrer les problèmes d'approximation lorsque p se rapproche de 0 ou de 1.

Les pages suivantes présentent les feuilles de route de ces trois activités, comprenant le code des commandes utilisées ainsi que quelques exemples de résultats obtenus.

* LEGTA Louis Giraud (Ministère de l'Agriculture) à Serres, 84200 CARPENTRAS
** Institut Camille Jordan, Domaine de la Doua, 69366 Villeurbanne

DU DIAGRAMME EN BÂTON À L'"HISTOGRAMME" POUR ILLUSTRER LE THÉORÈME DE MOIVRE-LAPLACE

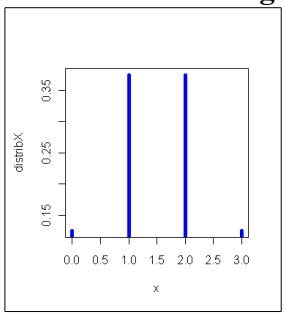
I - REPRÉSENTER GRAPHIQUEMENT LA DISTRIBUTION D'UNE LOI BINOMIALE (les code sont dans le fichier **"MoivreLaplace.R"**)

Un premier exemple d'**une variable aléatoire X** de loi binomiale de paramètres n = 3 et p = 0,5 :

On crée une liste contenant la distribution de probabilité de X :

```
(x <- 0:10) ; n <- 10 ; p <- 0.5
(distribX <- dbinom(0:n, n, p))
[1] 0.125 0.375 0.375 0.125
```

On réalise l'illustration graphique habituelle (comment appelle-t-on ce type de graphique ?) :




```
plot(x, distribX, type = "h", col = "blue", lwd = 5)
```

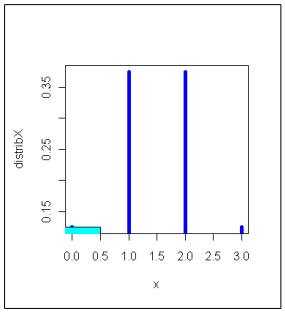
On va "habiller" avec des rectangles :

On prépare les bornes des intervalles :

```
(bornesX <- seq(-.5, n + .5, 1))
[1] -0.5  0.5  1.5  2.5  3.5
```

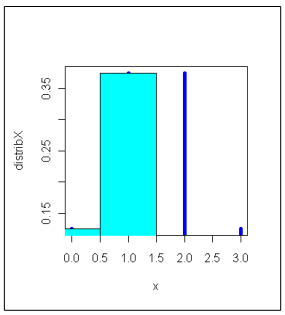
On superpose le premier rectangle :

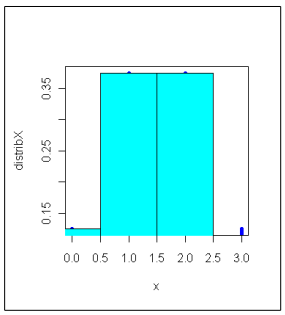
```
polygon(c(-0.5, -0.5, 0.5, 0.5), c(0, distribX[1], distribX[1], 0), col = "cyan")
```

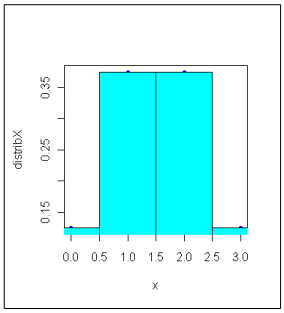


On superpose le deuxième rectangle puis le troisième et le quatrième :

```
polygon(c(0.5, 0.5, 1.5, 1.5), c(0, distribX[2], distribX[2], 0),
col = "cyan")
polygon(c(1.5, 1.5, 2.5, 2.5), c(0, distribX[3], distribX[3], 0),
col = "cyan")
polygon(c(2.5, 2.5, 3.5, 3.5), c(0, distribX[4], distribX[4], 0),
col = "cyan")
```

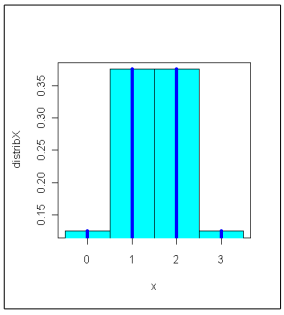






Il y a un petit aménagement à faire sur l'axe des abscisses, lequel ? Il faut reprendre la construction du graphique depuis la commande plot, mais avec le type "n". Placer ensuite les rectangles. Enfin on peut superposer les barres ...(voir le code dans le fichier **"MoivreLaplace.R"**)

Que vaut la surface totale des rectangles ?



II - REPRÉSENTATION GRAPHIQUE DE LA DISTRIBUTION DE PROBABILITÉ DE LA VARIABLE CENTRÉE RÉDUITE (X – E(X)) / racine(V(X))

(les lignes de code sont dans le fichier **"MoivreLaplace.R",** à partir de la ligne 108)

On crée une liste contenant les valeurs de la variable centrée réduite :

```
EspX <- n * p ; etypX <- sqrt(n * p * (1 - p))
(xCR <- (x - EspX) / etypX)
[1] -1.7320508 -0.5773503  0.5773503  1.7320508
```

On calcule les bornes centrées réduites des intervalles utilisés pour le graphique précédent.

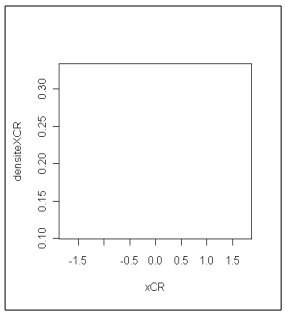
```
(BornesXCR <- (bornesX - EspX) / etypX)
[1] -2.309401 -1.154701  0.000000  1.154701  2.309401
```

On calcule les densités

```
(densiteXCR <- distribX * etypX)
[1] 0.1082532 0.3247595 0.3247595 0.1082532
```

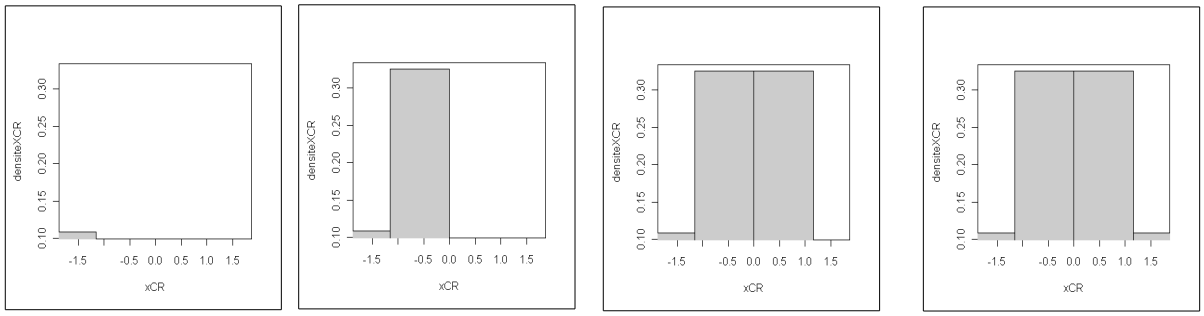
On ouvre une fenêtre graphique de tracé, vide

```
plot(xCR, densiteXCR, type = "n")
```



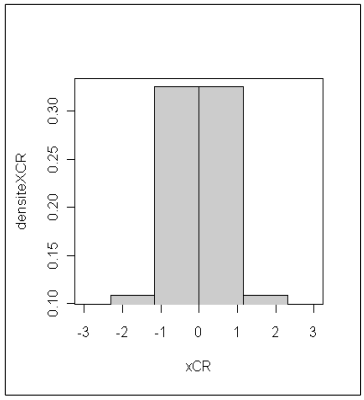
On trace les rectangles un par un :

```
polygon(c(BornesXCR[1], BornesXCR[1], BornesXCR[2], BornesXCR[2]),
        c(0, densiteXCR[1], densiteXCR[1], 0), col = "grey80")
polygon(c(BornesXCR[2], BornesXCR[2], BornesXCR[3], BornesXCR[3]),
        c(0, densiteXCR[2], densiteXCR[2], 0), col = "grey80")
polygon(c(BornesXCR[3], BornesXCR[3], BornesXCR[4], BornesXCR[4]),
        c(0, densiteXCR[3], densiteXCR[3], 0), col = "grey80")
polygon(c(BornesXCR[4], BornesXCR[4], BornesXCR[5], BornesXCR[5]),
        c(0, densiteXCR[4], densiteXCR[4], 0), col = "grey80")
```



Il y a un petit problème sur l'axe des abscisses qu'il faut s'empresser de résoudre : lequel ?

```
minX <- floor(min(BornesXCR)) ; maxX <- ceiling(max(BornesXCR))
plot(xCR, densiteXCR, type = "n", xlim = c(minX, maxX))
# IL SUFFIT DE REFAIRE LES POLYGONES (sélectionner, exécuter)
```



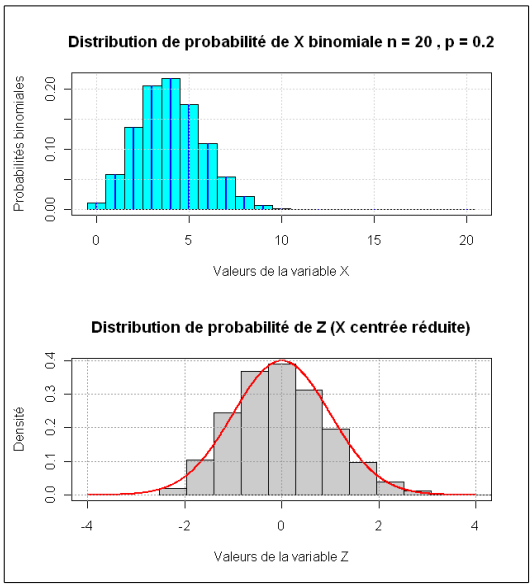
Un autre algorithme pour tracer les rectangles avec une boucle "pour".

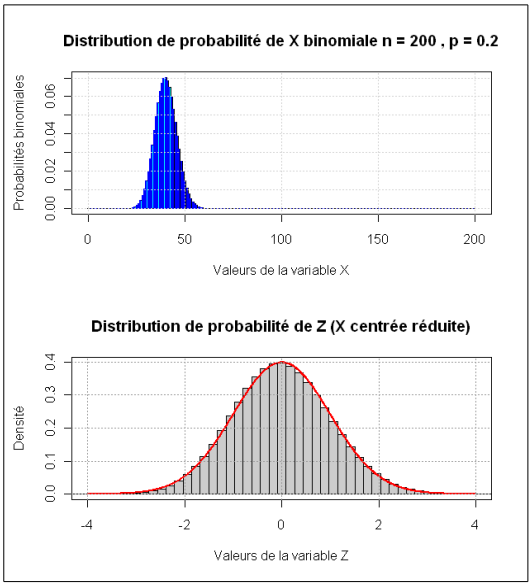
```
plot(xCR, densiteXCR, type = "n", xlim = c(minX, maxX))
for (k in 1:(n + 1)) {
   polygon(c(BornesXCR[k], BornesXCR[k], BornesXCR[k + 1], BornesXCR[k + 1]),
           c(0, densiteXCR[k], densiteXCR[k], 0), col = "grey80")
}
```

III – POUR POUVOIR FACILEMENT UTILISER D'AUTRES VALEURS DE n ET p ON FABRIQUE UNE FONCTION R

On rassemble les lignes de commande précédentes pour en faire une fonction **R**. Il faut juste leur donner la bonne structure, à l'aide de la fonction **function(...){...}**. On ajoute quelques légendes ...

```
MoivreLap0 <- function(n = 10, p = .5){
   x <- 0:n
   distribX <- dbinom(x, n, p)
   EspX <- n * p ; etypX <- sqrt(n * p * (1 - p))
   densiteXCR <- distribX * etypX
   subdivX <- seq(-.5, n + .5, 1)
   xCR <- (x - EspX) / etypX
   subdivXCR <- (subdivX - EspX) / etypX
   # "HISTOGRAMME" DE LA DISTRIBUTION BINOMIALE DE LA VARIABLE X
   par(mfrow = c(2, 1)) # Partage la fenêtre graphique en 2 lignes et 1 colonne.
   plot(x, distribX, type = "n", xlim = c(-.5, n + .5),
          xlab = "Valeurs de la variable X",
          ylab = "Probabilités binomiales",
          main = paste("Distribution de probabilité de X binomiale n =", n, ", p =", p))
      for (k in 1:(n + 1)) {
         polygon(c(subdivX[k], subdivX[k], subdivX[k + 1], subdivX[k + 1]),
                 c(0, distribX[k], distribX[k], 0), col = "cyan")
      }
      points(x, distribX, type = "h", col = "blue", lwd = 2)
      grid()
   # "HISTOGRAMME" DE LA VARIABLE CENTRÉE RÉDUITE Z = (X - E(X)) / écartype(X)
   plot(xCR, densiteXCR, type = "n", xlim = c(-4, 4),
          xlab = "Valeurs de la variable Z (X centrée réduite)",
          ylab = "Densité",
          main = "Distribution de probabilité de Z",
      for (k in 1:(n + 1)) {
         polygon(c(subdivXCR[k], subdivXCR[k], subdivXCR[k + 1], subdivXCR[k + 1]),
                 c(0, densiteXCR[k], densiteXCR[k], 0), col = "grey80")
      }
      xGauss <- seq(-4, 4, length = 1000) ; yGauss <- dnorm(xGauss)
      lines(xGauss, yGauss, col = "red", lwd = 2)
      grid()
}
```





DE LA FLUCTUATION D'ÉCHANTILLONNAGE
À LA LOI DES GRANDS NOMBRES

SIMULER ET REPRÉSENTER GRAPHIQUEMENT LA FLUCTUATION D'ÉCHANTILLONNAGE D'UNE VARIABLE FRÉQUENCE

Simulation de 18 (nbn) variables de Bernoulli de paramètre p, les valeurs sont stockées dans le “vecteur” `px` .

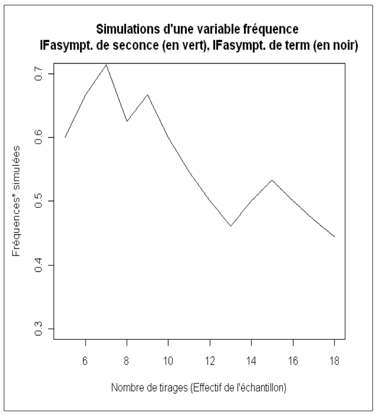
```
p <- .5 ; nbn <- 18
    px <- rbinom(nbn, 1, p)
  [1] 1 1 1 0 0 1 1 0 1 0 0 0 0 1 1 0 0 0
```

Calcul des 18 (nbn) fréquences cumulées correspondantes dont les valeurs sont stockées dans le vecteur `repartfreqsim`. On fait commencer le calcul des fréquences à partir de l'échantillon de taille 5.

```
(repartfreqsim <- cumsum(px)[5:nbn] / (5:nbn))
[1] 0.6000000 0.6666667 0.7142857 0.6250000 0.6666667 0.6000000 0.5454545
[8] 0.5000000 0.4615385 0.5000000 0.5333333 0.5000000 0.4705882 0.4444444
```

Illustration graphique en ligne brisée (`type = "l"`) de la première série de 18 simulations.

```
plot((5:nbn), repartfreqsim, type = "l", ylim = c(.3, .7),
      main = paste("Simulations d'une variable fréquence",
         "\nIFasympt. de seconce (en vert), IFasympt. de term (en noir)"),
      ylab = "Fréquences* simulées",
      xlab = "Nombre de tirages (Effectif de l'échantillon)")
```

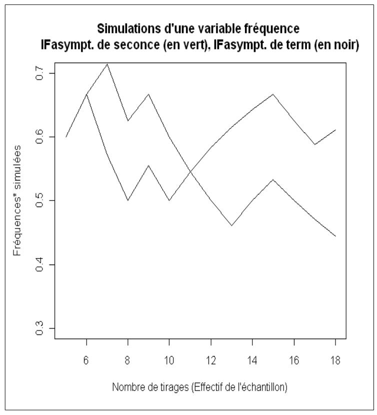


Simulation de la deuxième série de 18 simulations.

```
(rfs <- cumsum(rbinom(nbn, 1, p))[5:nbn] / (5:nbn))
 [1] 0.6000000 0.6666667 0.5714286 0.5000000 0.5555556 0.5000000 0.5454545
 [8] 0.5833333 0.6153846 0.6428571 0.6666667 0.6250000 0.5882353 0.6111111
```

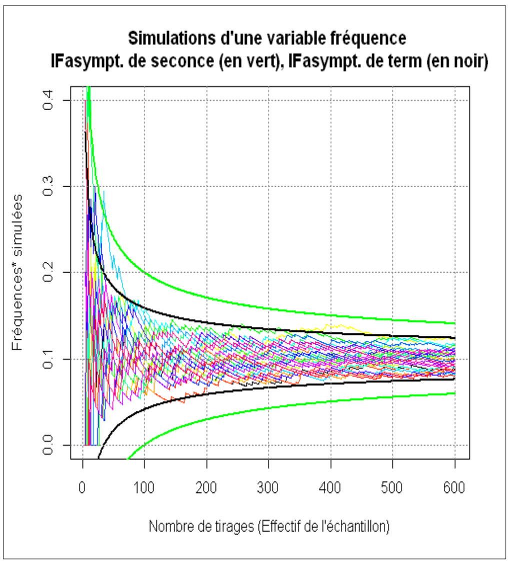
Illustration graphique, une deuxième ligne brisée se superpose à la première :

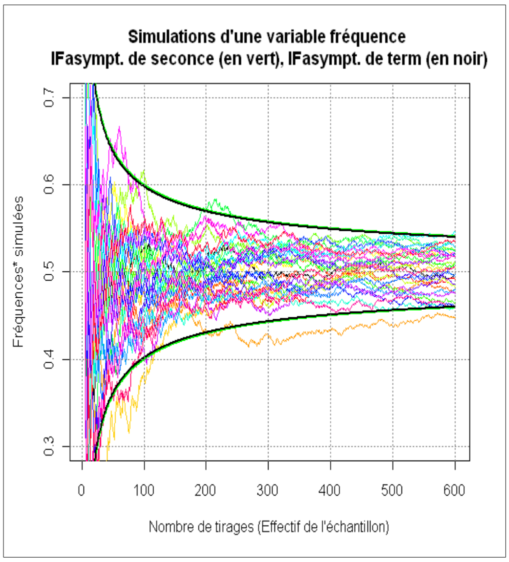
```
lines(5:nbn, rfs)
```



On continue à simuler des séries d'échantillons de tailles croissantes de 1 à 18, à l'aide d'une boucle for. Pour différencier les 18 séries, on fait intervenir la gestion de la couleur avec **R** et sa fonction rainbow(...)

```
for (i in 1:nbsim) {
    (rfs <- cumsum(rbinom(nbn, 1, p))[5:nbn] / (5:nbn))
    lines(5:nbn, rfs), col = rainbow(nbsim)[i])
}
```





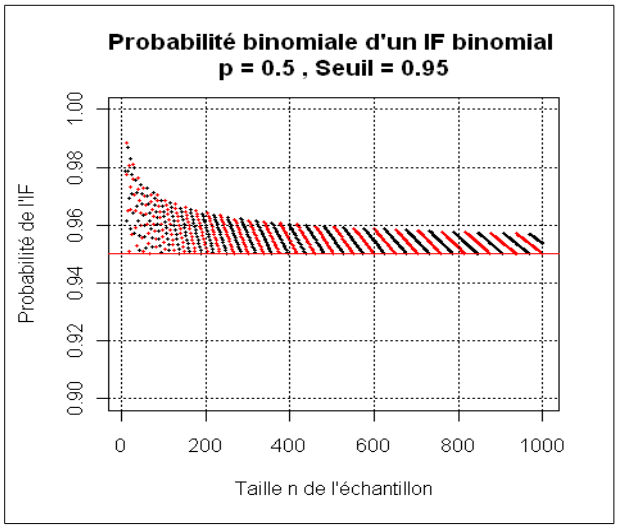
LES INTERVALES DE FLUCTUATION DE LA SECONDE À LA TERMINALE DES PROBABILITES BINOMIALES AUX ABAQUES

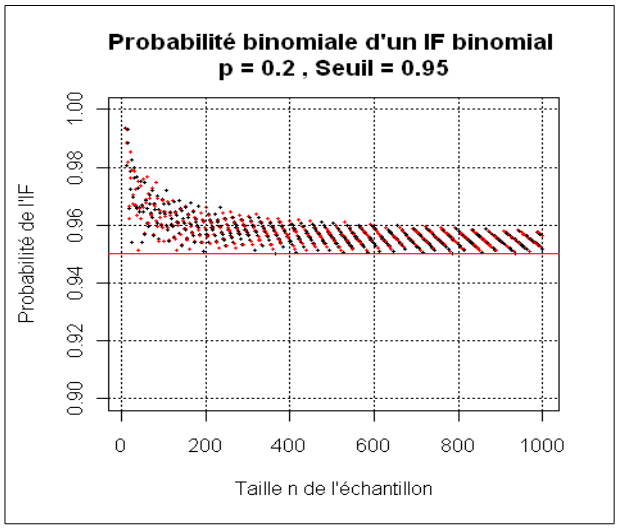
I – PROBABILITÉ BINOMIALE D'UN INTERVALLE DE FLUCTUATION D'UNE VARIABLE BINOMIALE CALCULÉ SELON LA MÉTHODE DU PROGRAMME DE PREMIÈRE (on le notera IF1)

```
PIF_Bino <- function(n, p, s){
   repartX <- pbinom(0:n, n, p)
   rang_a <- min(which(repartX > (1 - s) / 2))
   a <- rang_a - 1
   rang_b <- min(which(repartX >= (1 + s) / 2))
   b <- rang_b - 1
   pif <- sum(dbinom(a:b, n, p))
   return(pif)
}
```

```
# Le langage R permet de mettre directement en
# œuvre la méthode de calcul, sans passer par
# les boucles tant que qui constituent
# un obstacle supplémentaire
# Intervalle de fluctuation bilatéral au seuil
# minimal de probabilité s
# d'une variable binomiale X de paramètres n et p
# et de la variable fréquence X/n correspondante
# a est la plus petite valeur de X telle que
# P(X <= a) > (1 - s)/2)
# b est la plus petite valeur de X telle que
# P(X <= b) >= (1 - (1 - s)/2)
# [a ; b] est l'intervalle de fluctuation
# de X de proba. mini. s
```

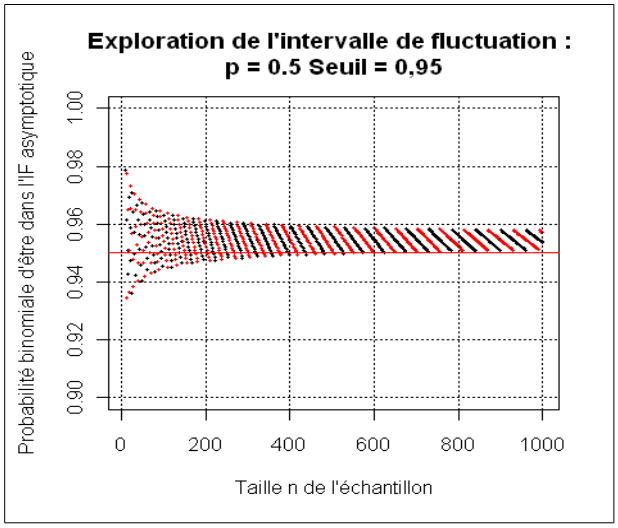
```
GrafPifBino <- function(N = 5:1000, p = .5, s = .95){
   SeriePif <- NULL
   for (i in N) {SeriePif <- c(SeriePif, PIF_Bino(i, p, s))}
   plot(N, SeriePif, pch = 19, col = c("black", "red"), cex = .4,
           ylim = c(.9, 1),
           main = paste("Probabilité binomiale d'un IF binomial",
                        "\np =", p, ", Seuil =", s),
           xlab = "Taille n de l'échantillon",
           ylab = "Probabilité de l'IF")
      abline(h = p, col = "red")
      grid(col = "black")
}
```

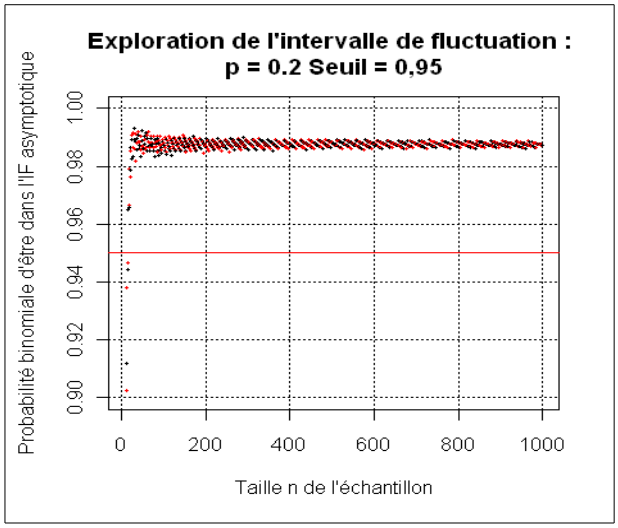




II – PROBABILITÉ BINOMIALE D'UN INTERVALLE DE FLUCTUATION ASYMPTOTIQUE D'UNE VARIABLE FRÉQUENCE, FORMULE DE SECONDE (on le notera IFasymp2)

```
# Probabilité binomiale de l'IF asymptotique Gaussien de seconde : p +- 1/racine(n)
# en fonction de n
pIFasy1_1 <- function(N = 5:1000, p = .5){
   P <- function(n, p) {
      binf <- max(floor(n * p - sqrt(n)), 0)
      bsup <- min(floor(n * p + sqrt(n)), n)
      sum(dbinom((binf + 1):bsup, n, p))
   }
   y <- NULL
   for (i in N) {y <- c(y, P(i, p))}
   #-----Affichage des graphiques-----
   plot(N, y, pch = 19, cex = .4, col = c("black", "red"),
         ylim = c(.9, 1),
         xlab = "Taille n de l'échantillon",
         ylab = "Probabilité binomiale d'être dans l'IF asymptotique",
         main = paste("Exploration de l'intervalle de fluctuation :",
                      "\np =", p, ", Seuil = 0,95"))
      abline(h = .95, col = "red")
      grid(col = "black")
}
```

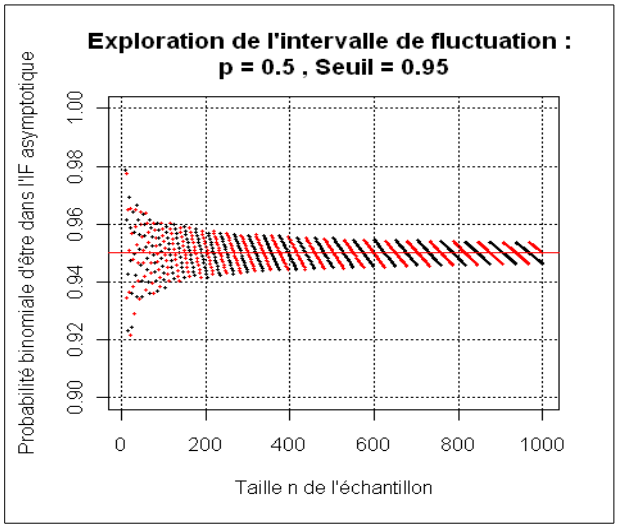


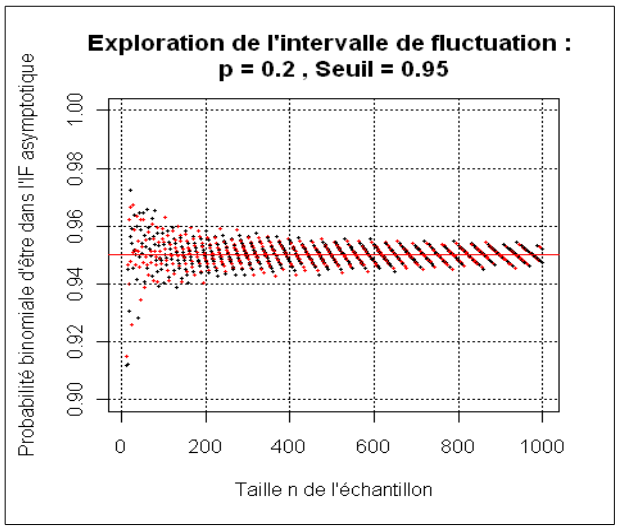


III – PROBABILITÉ BINOMIALE D'UN INTERVALLE DE FLUCTUATION ASYMPTOTIQUE D'UNE VARIABLE FRÉQUENCE, FORMULE DE TERMINALE (on notera IFasympT)

```
# Probabilité binomiale de l'IF asymptotique Gaussien de terminale
# en fonction de n, (p +- 1,96*racine(p(1-p)/n))
pIFasy2_1 <- function(N = 5:1000, p = .5, s = .95){
   P <- function(n, p, s) {
      g <- qnorm(1 - (1 - s) / 2)
      binf <- max(floor(n * p - g * sqrt(p * (1 - p) * n)), 0)
      bsup <- min(floor(n * p + g * sqrt(p * (1 - p) * n)), n)
      sum(dbinom((binf + 1):bsup, n, p))
   }
   y <- NULL
   for (i in N) {y <- c(y, P(i, p, s))}
   #-----Affichage des graphiques-----
   plot(N, y, pch = 19, cex = .4, col = c("black", "red"),
         ylim = c(.9, 1),
         xlab = "Taille n de l'échantillon",
         ylab = "Probabilité binomiale d'être dans l'IF asymptotique",
         main = paste("Exploration de l'intervalle de fluctuation :",
                      "\np =", p, ", Seuil =", s),
```

```
        abline(h = s, col = "red")
        grid(col = "black")
}
```





IV – CONSTRUCTION DE QUELQUES ABAQUES D'INTERVALLES DE FLUCTUATION BINOMIAUX D'UNE VARIABLE FRÉQUENCE ET LEUR UTILISATION POUR DÉTERMINER DES INTERVALLES DE CONFIANCE D'UNE PROPORTION

Cet algorithme tracera les abaques des intervalles de fluctuation suivants :

- asymptotique du programme de seconde, noté **IFasym2.** Cet intervalle n'apparaîtra que pour le seuil de 95 %.
- “exact” du programme de première, noté **IF1**
- asymptotique du programme de terminale, noté **IFasympT**

On a les paramètres suivants en entrée :

- n : la taille de l'échantillon (le nombre d'épreuves indépendantes)
- seuil : le seuil de probabilité de l'I.F.
- nbvalp : la subdivision de valeurs de p, proportion dans la population ou probabilité de succès
- NumIF : série des rang des valeurs de p pour lesquelles l'I.F. Est représenté graphiquement.

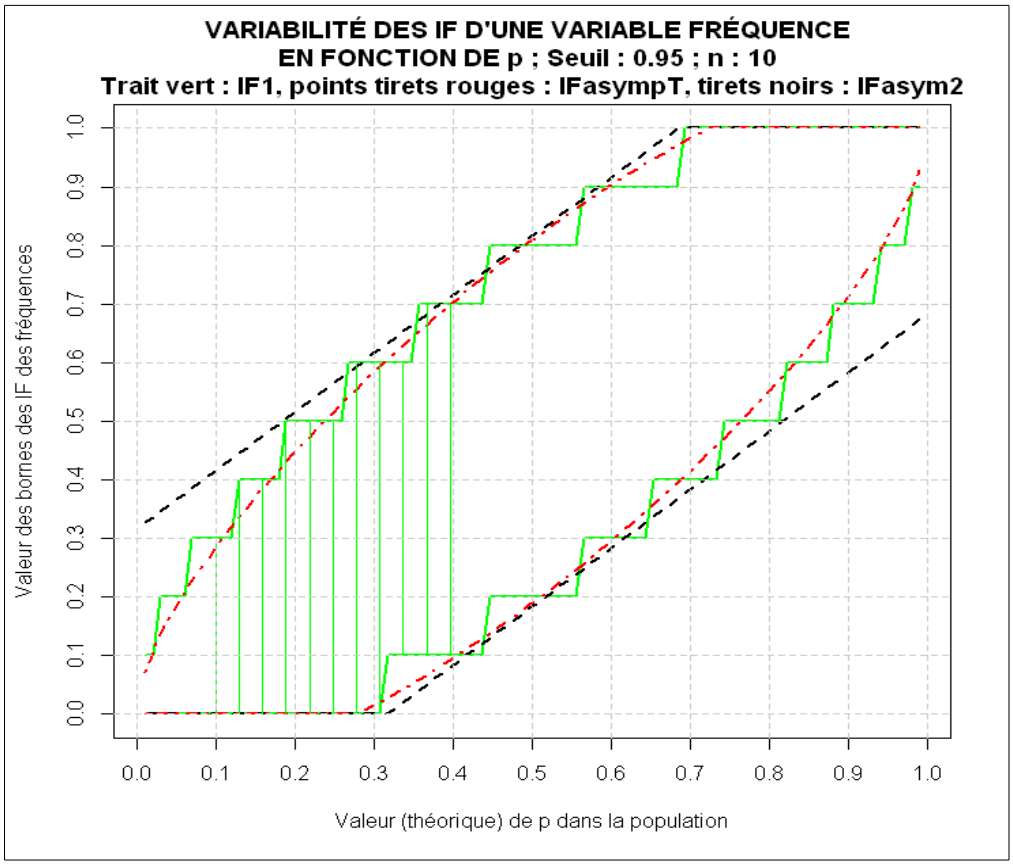
Détermination d'un intervalle de confiance d'une proportion à l'aide des abaques

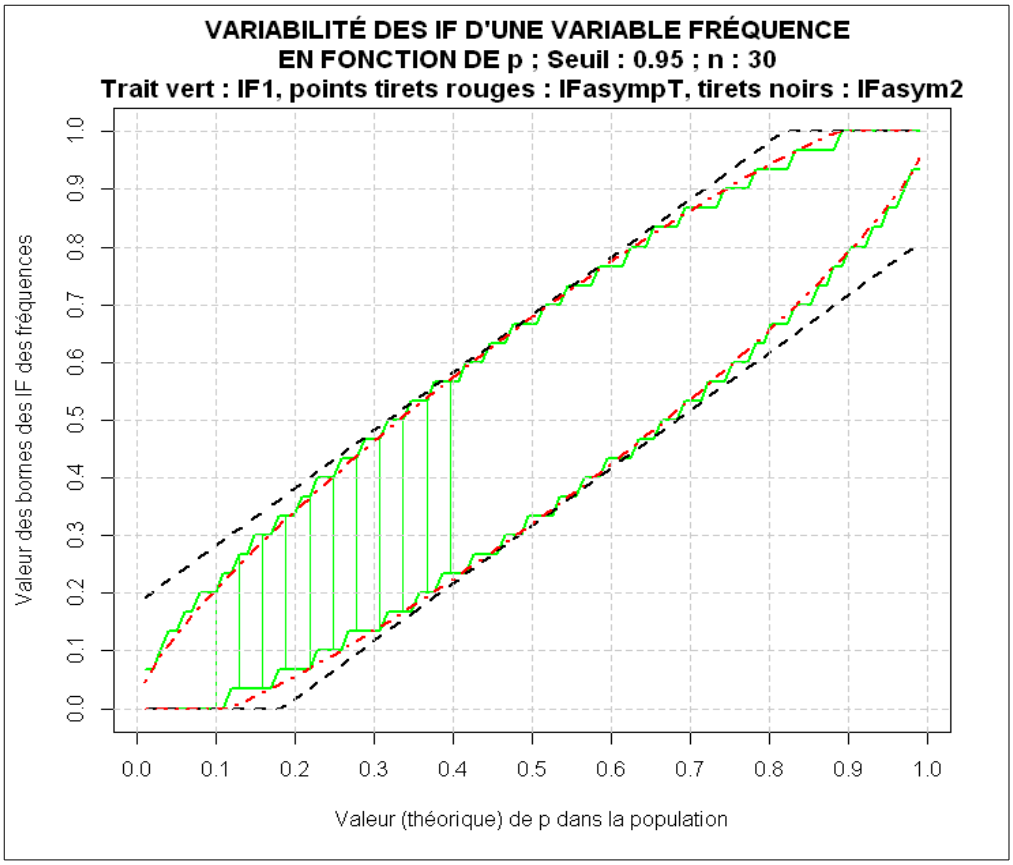
Il suffit de lire l'abaque en sens “inverse” : pour une valeur de la fréquence f lue sur l'axe des ordonnées, on détermine les deux valeurs de p sur l'axe des abscisses...

L'utilisation de l'abaque de l'**IF1** nécessite des précautions particulières .... On expliquera pourquoi...

Il existe un algorithme spécifique qui permet de déterminer l'intervalle de confiance “exact” d'une proportion, c'est la méthode de CLOPPER-PEARSON qui peut donner lieu à bel exercice d'algorithmique, utilisant simplement des répartitions binomiales.

```r
# Fonction utilitaire
IF_Bino <- function(n, p, s){
   repartX <- pbinom(0:n, n, p)
   if (max(repartX) <= (1 - s) / 2) {
      a <- 0
   } else {
      rang_a <- min(which(repartX > (1 - s) / 2))
      a <- rang_a - 1
   }
   if (min(repartX) > (1 + s) / 2) {
       b <- n
   } else {
      rang_b <- min(which(repartX >= (1 + s) / 2))
      b <- rang_b - 1
   }
   IF <- c(a, b)
   return(IF)
}
#---------------------------
# Cette fonction utilise IF_Bino
abak_IF <- function(n = 10, seuil = .95, nbvalp = 100, NumIF = seq(10, 40, 3)){
   subdivp <- seq(.01, .99, length = nbvalp)
   bInfIF <- NULL ; bSupIF <- NULL
   bInfIFasyT <- NULL ; bSupIFasyT <- NULL
   if (seuil == .95) {bInfIFasy2 <- NULL ; bSupIFasy2 <- NULL}
   g <- qnorm(1 - (1 - seuil) / 2)
   for (p in subdivp) {
      bInfIF <- c(bInfIF, IF_Bino(n, p, seuil)[1] / n)
      bSupIF <- c(bSupIF, IF_Bino(n, p, seuil)[2] / n)
      if (seuil == .95){
         bInfIFasy2 <- c(bInfIFasy2, max(p - 1 / sqrt(n), 0))
         bSupIFasy2 <- c(bSupIFasy2, min(p + 1 / sqrt(n), 1))
      }
      bInfIFasyT <- c(bInfIFasyT, max(p - g * sqrt(p * (1 - p) / n), 0))
      bSupIFasyT <- c(bSupIFasyT, min(p + g * sqrt(p * (1 - p) / n), 1))
   }
   # GRAPHIQUES
      if (seuil == .95) {
         minY <- min(c(bInfIF, bInfIFasy2, bInfIFasyT))
         maxY <- max(c(bSupIF, bSupIFasy2, bSupIFasyT))
      } else {
         minY <- min(c(bInfIF, bInfIFasyT))
         maxY <- max(c(bSupIF, bSupIFasyT))
      }
   plot(subdivp, bInfIF, type = "l", col = "green", ylim = c(minY, maxY), lwd = 2,
         main = paste("VARIABILITÉ DES IF D'UNE VARIABLE FRÉQUENCE",
            "\nEN FONCTION DE p ; Seuil :", seuil, "; n :", n,
            "\nTrait vert : IF1, points tirets rouges : IFasympT, tirets noirs : IFasym2"),
         xlab = "Valeur (théorique) de p dans la population",
         ylab = "Valeur des bornes des IF des fréquences",
         xaxp = c(0, 1, 10), yaxp = c(0, 1, 10))
      lines(subdivp, bSupIF, col = "green", lwd = 2)
      if (seuil == .95) {
         lines(subdivp, bInfIFasy2, lty = 2, lwd = 2)
         lines(subdivp, bSupIFasy2, lty = 2, lwd = 2)
      }
      lines(subdivp, bInfIFasyT, lty = 4, col = "red", lwd = 2)
      lines(subdivp, bSupIFasyT, lty = 4, col = "red", lwd = 2)
      for (i in NumIF) {
         lines(c(subdivp[i], subdivp[i]), c(bInfIF[i], bSupIF[i]), col = "green")
      }
      abline(h = seq(0, 1, .1), v = seq(0, 1, .1), col = "grey80", lty = 2)
}
```


VARIABILITÉ DES IF D'UNE VARIABLE FRÉQUENCE
EN FONCTION DE p ; Seuil : 0.95 ; n : 10
Trait vert : IF1, points tirets rouges : IFasympT, tirets noirs : IFasym2
Valeur des bornes des IF des fréquences
Valeur (théorique) de p dans la population


VARIABILITÉ DES IF D'UNE VARIABLE FRÉQUENCE
EN FONCTION DE p ; Seuil : 0.95 ; n : 30
Trait vert : IF1, points tirets rouges : IFasympT, tirets noirs : IFasym2
Valeur des bornes des IF des fréquences
Valeur (théorique) de p dans la population