Université de Nantes
Département de mathématiques

Module X4M0020
Année 2016-2017

Devoir 1 de TP

Il faut remplacer, dans le code R, 1374 par les 4 derniers chiffres de votre code ETUDIANT, en enlevant éventuellement zéro quand il est en première position).
Tout le programme doit être dans un seul fichier NOM.R, où NOM est votre nom de famille (tronqué s'il est en plusieurs parties).
On fera une redaction dans un fichier PDF, NOM.pdf, pour commenter vos résultats et y inclure les figures).
Les 2 fichiers sont à envoyer en pièces jointes par Mail, avec comme objet "TP proba" à Abderemane.Morame@univ-nantes.fr

Exercice I

ON VERIFIE QUE LA PROBABILITE DE TIRER AVEC REMISE
6 NUMEROS DIFFERENTS SUR n EST EGALE A

$$p(n) = (n-1)(n-2)(n-3)(n-4)(n-5)/n^5$$

La fonction p(n) en R peut s'écrire comme ci-dessous.

```
######################
p = function(n)
 x = 1:5
  return( prod(n - x)/(n^5) )

#####################
```

ON SE LIMITERA A n dans 10 :99, (90 valeurs)
LA FONCTION g(n) CI-DESSOUS RENVOIE 1
QUAND 6 TIRAGES AVEC REMISE
dans 1 :n DONNE 6 VALEURS DISTINCTES
(ET RENVOIE 0 DANS LE CAS CONTRAIRE)

```
##################
g = function(n)
 x = 1:n
  z = 0
  v = sample(x, 6, replace=TRUE)
   if ( 6 == length(unique(v)) )  z = 1
  return( z )

#############
```

LA FONCTION DE SIMULATION CI-DESSOUS, simule(N, n),
APPELLE g(n) N-FOIS ET RENVOIE

LA FREQUENCE DES SUCCES :
LE NOMBRE DES g(n) EGAUX A 1 DIVISE PAR N

```
########################
simule = function(N, n)
 NV = rep(0, N)
  x = 0
  for (k in 1:N) NV[k] = g(n)
  x = sum(NV)/N
  return(x)

######################
```

DEBUT DE LA SIMULATION

SUIVRE LES INDICATIONS DONNEES APRES LE SIGNE # DU COMMENTAIRE.

```
########################
N = 1000 # nombre de testes
set.seed(1374) # remplacer 1374 par
# les 4 derniers chiffres de votre code ETUDIANT,
# en enlevant éventuellement zéro quand il est en première position)
nV = 10:99
L = length(nV)
xV =0*nV # Initialisation
yV = 0*nV # Initialisation
# A completer
```

- Changer le xV avec for (i in 1 :L) pour que xV[i] soit simule(N, nV[i])
- Changer yV avec for (i in 1 :L) pour que yV[i] soit la probabilité exacte p(nV[i])
- Faire un plot de (nV,xV) en bleu et de (nV,yV) en rouge dans la même figure.
- Calculer la moyenne m de xV - yV. Dans R fonction moyenne est **mean()**
- Calculer la moyenne de $(xV - yV - m)^2$, (c'est la variance de xV-yV).
- Répeter les opérations en doublant N

Exercice II

ON VERFIE QUE LA PROBABILITE DE TIRER m NUMEROS SUR n DANS UN BON ORDRE EST TOUJOURS EGALE A 1/m!.

La fonction R, f(m,n), ci-dessous fait un tirage sans remise de m valeurs dans 1 :n, et retourne 1 quand les entiers tirés sont ordonnés et 0 dans le cas contraire.

```
######################
f = function(m, n)
 x = 1:n
   v = sample(x, m, replace=FALSE)
   teste = ( v == sort(v)) # sort(v) ordonne v
   z = 0
```

```
   if ( m == sum(teste) )  z = 1
   return( z )

#############
```

La fonction R, simuleV(N,m,nV), N et m entiers et nV vecteur d'entiers ci-dessous, retourne un vecteur xV. Elle appelle la fonction f(m,nV[k]) N-fois et met dans xV[k] le nombre de 1 obtenu divisé par N.

```
######################
### FONCTION DE SIMULATION: n  est vectoriel
simuleV = function(N, m, nV)
 NV = rep(0, N)
   nL = length(nV)
   xV = rep(0, nL)
   for (i in 1:nL)
       for (k in 1:N) NV[k] = f(m, nV[i])
       xV[i] = sum(NV)/N
   return(xV)

#############
```

DEBUT DE LA SIMULATION

SUIVRE LES INDICATIONS DONNEES APRES LE SIGNE # DU COMMENTAIRE.

```
########################
mV = 2:8
nV = 10:30
N = 500 # nombre de testes
set.seed(1374) # remplacer 1374 par
# les 4 derniers chiffres de votre code ETUDIANT,
# en enlevant éventuellement zéro quand il est en première position).
# A completer
```

- Créer un vecteur pV de même taille que mV tel que **pV[i] = mean(simuleV(N, mV[i], nV))**.
- Creer un vecteur P de meme taille que que mV tel que P[i] = 1/mV[i] ! : c'est la probabilité exacte.
- Calculer le maximum des erreurs en valeurs absolues **max(abs(pV - P)))**
- Tracer en bleu les points(mV[i], pV[i]) et y ajouter en rouge la courbe (x,y)=(mV, P).
- Calculer la moyenne des erreurs |pV[i] -P[i]| .
- Calculer le maximum des erreurs |pV[i] -P[i]|.
- Recommencer en changeant N en 3*N.