ASI (L2) : TP2 Calculs techniques sous *Excel* et *Rstat*

Objectifs du TP :

Savoir utiliser Excel et Rstat pour faire des calculs mathématiques et statistiques, mettre en forme les résultats.

1. Fonctions mathématiques avec Excel

Puissances et factorielles

On renconcontre dans les formules statistiques des puissances et des factorielles. Comment calcule-t-on 3^2 avec *Excel* ? Même question pour $5\,!$.

Mathématiquement, il n'y a pas de limite de calcul pour $n!$. Par contre les logiciels coincent souvent pour les très grands nombres.

Quel est le plus grand entier n pour lequel *Excel* sait calculer $n!$? Pourquoi cette limite ? (pour n^n le plus grand nombre est 144).

On veut comparer n^n et $n!$ ainsi que 10^n pour n de 1 à 30 puis pour n=100,200,300.

Utiliser un joli tableau avec 4 colonnes pour comparer ces valeurs. On utilisera des formats entiers puis des formats en puissance de 10.

Conclusion : quelle expression est la plus grande ?

Coefficients du binome

Comment calcule-t-on C_n^k ? Donner le tableau des coefficients du binome C_{15}^k pour k de 0 à 15. Est-ce que les coefficients croissent, décroissent ? Où sont les plus gros C_{15}^k ? Peut-on calculer C_{15}^{17} sous Excel ?

Calculer $C_n^{n/2}$ pour n= 10, 20, 30, 50, 100, 200, 300, 400, 500, 1000, 1100.

Que peut-on en conclure ?

Afficher pour n de 3 à 20 le nombre de droites qu'on peut tracer avec n points (formule C_n^2) et la probabilité qu'une de ces droites soit une diagonale (formule $(n-3)/(n-1)$).

Ne pas donner trop vite les formules, elles ont été vues au T.D. 1.

Quelle est la probabilité de gagner au loto ? On tire 6 boules sans répétition ni remise parmi 49. Formule : C_{49}^6. Et pour le "ghLoto" avec 8 boules parmi 60 ? Est-ce mieux en terme de chances ? Le corrigé du TD 1 disait 1/183. Est-ce correct ?

Comptages de fichiers

En "ghFs" il y a $I = N \times T$ identificateurs de fichiers avec $N = \sum_{i=0}^{4} 20.30^i$ et $T = \sum_{j=0}^{2} 20.30^j$.

Donner N, T et I. Retrouve-t-on 312 milliards ?

Comparer avec le Fs de l'ancien $MsDos$ sachant que pour N on a de 1 à 8 caractères et pour T on a de 0 à 3 caractères. On peut simplifier en prenant comme caractères les lettres, les chiffres et seulement l'espace souligné.

Suites par récurrence du *sprite*

Dans le TD 2, on avait trouvé 3 suites α_n, γ_n et δ_n telles que :

$\alpha_0 = 1$, $\gamma_0 = 0$ et $\delta_0 = 0$ avec les relations $\alpha_n = (\gamma_n + \delta_n)/2$, $\gamma_n = (\alpha_n + \delta_n)/2$ et $\delta_n = (\alpha_n + \gamma_n)/2$.

Donner avec *Excel* les valeurs de α_n, γ_n et δ_n pour n de 1 à 10. Vérifier que la somme $\alpha_n + \gamma_n + \delta_n$ vaut 1 puis programmer les relations de récurrence données en TD.

Rappeler éventuellement les formules

$$\alpha_n = \frac{1}{3} + \frac{2}{3}\left(\frac{-1}{2}\right)^n$$

$$\gamma_n = \frac{1}{3} - \frac{2}{3}\left(\frac{-1}{2}\right)^n$$

$$\delta_n = \frac{1}{3} - \frac{2}{3}\left(\frac{-1}{2}\right)^n$$

Trouve-t-on les mêmes valeurs ?

Le corrigé disait que la limite de α_n était 1/3. Le voit-on ? Pour quelle valeur de n a-t-on $|\alpha_n - 1/3| \leqslant 10^{-5}$? Sauriez-vous le démontrer ?

Fichier pour enseignant

`K:\Stat_ad\comp.xls` ; il y a plusieurs onglets à regarder...

2. Fonctions mathématiques avec Rstat

Puissances et factorielles

On renconcontre dans les formules statistiques des puissances et des factorielles. Comment calcule-t-on 3^2 avec *Rstat* ? Même question pour 5 !. Quel est le plus grand entier n pour lequel *Rstat* sait calculer $n!$? Pourquoi cette limite ? On veut comparer n^n et $n!$ ainsi que 10^n pour n de 1 à 30 puis pour n=100,200,300.

Après avoir lu l'aide sur `array` et `matrix`, utiliser un joli tableau avec 4 colonnes pour comparer ces valeurs. Penser à utiliser des fonctions qui remplissent les colonnes... Rappeler comment on sélectionne les lignes et les colonnes, montrer `colnames`.

Conclusion : quelle expression est la plus grande ?

Coefficients du binome

Comment calcule-t-on C_n^k ? Donner le tableau des coefficients du binome C_{15}^k pour k de 0 à 15. Est-ce que les coefficients croissent, décroissent ? Où sont les plus gros C_{15}^k ? Peut-on calculer C_{15}^{17} sous Rstat ?

Calculer $C_n^{n/2}$ pour $n=$ 10, 20, 30, 50, 100, 200, 300, 400, 500, 1000, 1100.

Que peut-on en conclure ?

Afficher pour n de 3 à 20 le nombre de droites qu'on peut tracer avec n points (formule C_n^2) et la probabilité qu'une de ces droites soit une diagonale (formule $(n-3)/(n-1)$).

Ne pas donner trop vite les formules, elles ont été vues au T.D. 1.

Quelle est la probabilité de gagner au loto ? On tire 6 boules sans répétition ni remise parmi 49. Formule : C_{49}^6. Et pour le "ghLoto" avec 8 boules parmi 60 ? Est-ce mieux en terme de chances ? Le corrigé du TD 1 disait 1/183. Est-ce correct ?

Comptages de fichiers

En "ghFs" il y a $I = N \times T$ identificateurs de fichiers avec $N = \sum_{i=0}^{4} 20.30^i$ et $T = \sum_{j=0}^{2} 20.30^j$.

Donner N, T et I. Retrouve-t-on 312 milliards ?

On commencera par créer une fonction de i qui renvoie 20.30^i et on utilisera `sum(...)` pour additionner les valeurs.

Comparer avec le Fs de l'ancien $MsDos$ sachant que pour N on a de 1 à 8 caractères et pour T on a de 0 à 3 caractères. On peut simplifier en prenant comme caractères les lettres, les chiffres et seulement l'espace souligné soit 37 caractères possibles.

Suites par récurrence du *sprite*

Dans le TD 2, on avait trouvé 3 suites α_n, γ_n et δ_n telles que :

$\alpha_0 = 1$, $\gamma_0 = 0$ et $\delta_0 = 0$ avec les relations $\alpha_n = (\gamma_n + \delta_n)/2$, $\gamma_n = (\alpha_n + \delta_n)/2$ et 5

Donner avec *Excel* les valeurs de α_n, γ_n et δ_n pour n de 1 à 10. Vérifier que la somme $\alpha_n + \gamma_n + \delta_n$ vaut 1 puis programmer les relations de récurrence données en TD.

Rappeler éventuellement les formules

$$\alpha_n = \frac{1}{3} + \frac{2}{3}\left(\frac{-1}{2}\right)^n$$

$$\gamma_n = \frac{1}{3} - \frac{2}{3}\left(\frac{-1}{2}\right)^n$$

$$\delta_n = \frac{1}{3} - \frac{2}{3}\left(\frac{-1}{2}\right)^n$$

Trouve-t-on les mêmes valeurs ?

Le corrigé disait que la limite de α_n était 1/3. Le voit-on ? Pour quelle valeur de n a-t-on $|\alpha_n - 1/3| \leqslant 10^{-5}$? Sauriez-vous le démontrer ?

Commencer par α_n en formule exacte à l'aide d'une fonction *an*.

Montrer ensuite comment écrire α_n par récurence (la nommer *arn*). Les laisser écrire *brn* et *crn* pour remplir le tableau.

Pour n tel que $|\alpha_n - 1/3| \leqslant 10^{-5}$, faire des essais avec une fonction `difference(n)`.

Signaler que plus tard on utilisera une boucle tant que.

Remarques

Le prolongement continu de la fonction factorielle se nomme "la fonction GAMMA", notée Γ. Ainsi $\Gamma(x) = (x-1)!$ et $\Gamma(3.5) = 3.323$.

Le prolongement continu de la fonction $(n, k) \mapsto C_n^k$ se nomme "la fontion béta", notée β ; par exemple $\beta(5.3, 1.2) = 6.558$

Esquisse de solution

```
R : Copyright 2004, The R Foundation for Statistical Computing
Version 1.9.0  (2004-04-12), ISBN 3-900051-00-3

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for a HTML browser interface to help.
Type 'q()' to quit R.

> #  (gH)   --  asi_tp2.r  ;  TimeStamp (unix) : 02 Mars 2005 vers 13:26
> #
>
> ####################################################
> #
> #  factorielles et puissances
> #
> ####################################################
>
> # la puissance se calcule avec ^ ou avec **
>
> factorial(5)
[1] 120
>
> # factorial(n) renvoie  n! ;
> # la limite est 170 (il faut avoir Rstat version 1.9 ou supérieur)
>
> factorial(170)   # renvoie  7.257416e+306
[1] 7.257416e+306
> factorial(171)   # renvoie  Inf comme infini
[1] Inf
```

```
> # création de la matrice
>
> tdv <- matrix(nrow=33,ncol=4) ;
> tdv
      [,1] [,2] [,3] [,4]
 [1,]   NA   NA   NA   NA
 [2,]   NA   NA   NA   NA
 [3,]   NA   NA   NA   NA
 ...
[33,]   NA   NA   NA   NA
>
> # nom des colonnes
>
> colnames(tdv) <- c("n","n^n","n!","10^n")
> tdv # pour vérifier
       n n^n n! 10^n
 [1,] NA  NA NA   NA
 [2,] NA  NA NA   NA
...
[33,] NA  NA NA   NA
>
> # remplissage des colonnes
>
> lacol   <- c((1:30),(100*(1:3)))
> tdv[,1] <- lacol
> tdv[,2] <- lacol^lacol
> tdv[,3] <- factorial(lacol)
> tdv[,4] <- 10^lacol
>
> ####################################################
> #
> #  coefficients du binome
> #
> ####################################################
>
> # c_n_k se calcule par choose(n,k)
>
> # help(choose)
>
> cdb     <- matrix(nrow=16,ncol=2) ;
> cdb[,1] <- c(0:15)
```

```
> cdb[,2] <- choose(15,cdb[,1])
> cdb
      [,1] [,2]
 [1,]    0    1
 [2,]    1   15
 [3,]    2  105
 [4,]    3  455
 [5,]    4 1365
 [6,]    5 3003
 [7,]    6 5005
 [8,]    7 6435
 [9,]    8 6435
[10,]    9 5005
[11,]   10 3003
[12,]   11 1365
[13,]   12  455
[14,]   13  105
[15,]   14   15
[16,]   15    1
>
> # choose(15,17) renvoie 0 et non pas erreur.
>
> cnk     <- matrix(nrow=12,ncol=3) ;
> ind     <- c(10*1:5,100*1:5,1000,1100) ;
> cnk[,1] <- ind
> cnk[,2] <- ind/2
> cnk[,3] <- choose( cnk[,1] , cnk[,2] )
> cnk
      [,1] [,2]          [,3]
 [1,]   10    5  2.520000e+02
 [2,]   20   10  1.847560e+05
 [3,]   30   15  1.551175e+08
 [4,]   40   20  1.378465e+11
 [5,]   50   25  1.264106e+14
 [6,]  100   50  1.008913e+29
 [7,]  200  100  9.054851e+58
 [8,]  300  150  9.375970e+88
 [9,]  400  200 1.029525e+119
[10,]  500  250 1.167443e+149
[11,] 1000  500 2.702882e+299
[12,] 1100  550           Inf
```

```
> dro     <- matrix(nrow=18,ncol=3) ;
> ind     <- c(3:20)
> dro[,1] <- ind
> dro[,2] <- choose(ind,2)
> dro[,3] <- (ind-3)/(ind-1)

> dro

      [,1] [,2]      [,3]
 [1,]    3    3 0.0000000
 [2,]    4    6 0.3333333
 [3,]    5   10 0.5000000
 [4,]    6   15 0.6000000
 [5,]    7   21 0.6666667
 [6,]    8   28 0.7142857
 [7,]    9   36 0.7500000
 [8,]   10   45 0.7777778
 [9,]   11   55 0.8000000
[10,]   12   66 0.8181818
[11,]   13   78 0.8333333
[12,]   14   91 0.8461538
[13,]   15  105 0.8571429
[14,]   16  120 0.8666667
[15,]   17  136 0.8750000
[16,]   18  153 0.8823529
[17,]   19  171 0.8888889
[18,]   20  190 0.8947368


> # pour le loto
>
> c1 <- choose(49,6)
> c2 <- choose(60,8)
> c2/c1 ;

[1] 182.9701
```

```
>
> ####################################################
> #
> #  sommes de puissances
> #
> ####################################################
>
>
> nbk <- function(i) { 20*30^i }
> N   <- sum( nbk( 0:4) )
[1] 16758620
> T
[1] 18620
> N*T
[1] 312045504400

# autre solution : comp(4)*comp(2)
# si comp <- function(j) { sum(20*(30^(0:i))) }
>
> # pour Dos
>
> nbkD <- function(j) { 37^j }
> sum( nbkD(1:8) ) * sum( nbkD(0:3) )
[1] 1.879391e+17
>
> ####################################################
> #
> #  récurence et formules directes (sprite)
> #
> ####################################################
>
>
> an      <- function(n) { (1/3) + (2/3)*(-1/2)^n }
> spr     <- matrix(nrow=10,ncol=6)
> ind     <- 0:9
> spr[,1] <- ind
> spr[,6] <- an(ind)
> spr
      [,1] [,2] [,3] [,4] [,5]      [,6]
 [1,]    0   NA   NA   NA   NA 1.0000000
 [2,]    1   NA   NA   NA   NA 0.0000000
```

```
 [3,]    2   NA   NA   NA   NA 0.5000000
 [4,]    3   NA   NA   NA   NA 0.2500000
 [5,]    4   NA   NA   NA   NA 0.3750000
 [6,]    5   NA   NA   NA   NA 0.3125000
 [7,]    6   NA   NA   NA   NA 0.3437500
 [8,]    7   NA   NA   NA   NA 0.3281250
 [9,]    8   NA   NA   NA   NA 0.3359375
[10,]    9   NA   NA   NA   NA 0.3320312
>
> arn     <- function(n) {
+   if (n==0) {
+      return( 1 )
+   } else  {
+       calc1 <- brn(n-1)
+       calc2 <- crn(n-1)
+       calc3 <- (calc1 + calc2) / 2
+       return( calc3 )
+   } ; # fin de si
+ } ; # fin de fonction arn
>
> # plus aggressif et plus court à écrire
>
> arn <- function(n) {
+   if (n==0) { return( 1 ) }
+       else  { return( (brn(n-1)+crn(n-1))/2 ) }
+ } ; # fin de fonction arn
>
>
> brn <- function(n) {
+   if (n==0) { return( 0 ) }
+       else  { return( (arn(n-1)+crn(n-1))/2 ) }
+ } ; # fin de fonction brn
>
> crn <- function(n) {
+   if (n==0) { return( 0 ) }
+       else  { return( (arn(n-1)+brn(n-1))/2 ) }
+ } ; # fin de fonction crn
>
> # pas de chance, car
>
> spr[,2] <- arn(ind)
```

```
Warning message:
the condition has length > 1
and only the first element will be used in:
   if (n == 0) {

>
> # n'est pas correct ; il faut écrire
>
> spr[,2] <-  tapply(as.vector(ind),FUN=arn,INDEX=ind)
>
> # complétons pour les autres
>
> spr[,3] <-  tapply(as.vector(ind),FUN=brn,INDEX=ind)
> spr[,4] <-  tapply(as.vector(ind),FUN=crn,INDEX=ind)
> spr
      [,1]      [,2]      [,3]      [,4] [,5]      [,6]
 [1,]    0 1.0000000 0.0000000 0.0000000   NA 1.0000000
 [2,]    1 0.0000000 0.5000000 0.5000000   NA 0.0000000
 [3,]    2 0.5000000 0.2500000 0.2500000   NA 0.5000000
 [4,]    3 0.2500000 0.3750000 0.3750000   NA 0.2500000
 [5,]    4 0.3750000 0.3125000 0.3125000   NA 0.3750000
 [6,]    5 0.3125000 0.3437500 0.3437500   NA 0.3125000
 [7,]    6 0.3437500 0.3281250 0.3281250   NA 0.3437500
 [8,]    7 0.3281250 0.3359375 0.3359375   NA 0.3281250
 [9,]    8 0.3359375 0.3320312 0.3320312   NA 0.3359375
[10,]    9 0.3320312 0.3339844 0.3339844   NA 0.3320312
>
> # sans oublier la somme pour vérification
>
> spr[,5] <- spr[,2] + spr[,3] + spr[,4]
>
> spr
      [,1]      [,2]      [,3]      [,4] [,5]      [,6]
 [1,]    0 1.0000000 0.0000000 0.0000000    1 1.0000000
 [2,]    1 0.0000000 0.5000000 0.5000000    1 0.0000000
 [3,]    2 0.5000000 0.2500000 0.2500000    1 0.5000000
 [4,]    3 0.2500000 0.3750000 0.3750000    1 0.2500000
 [5,]    4 0.3750000 0.3125000 0.3125000    1 0.3750000
 [6,]    5 0.3125000 0.3437500 0.3437500    1 0.3125000
 [7,]    6 0.3437500 0.3281250 0.3281250    1 0.3437500
```

```
 [8,]    7 0.3281250 0.3359375 0.3359375    1 0.3281250
 [9,]    8 0.3359375 0.3320312 0.3320312    1 0.3359375
[10,]    9 0.3320312 0.3339844 0.3339844    1 0.3320312

> # pour an proche de 1/3
>
> difference <- function(n) { abs(1/3-an(n)) }
>
> difference(10)
[1] 0.0006510417
> difference(20)
[1] 6.357829e-07
>
> difference(17)
[1] 5.086263e-06
>
> # solution avec boucle while
>
> cherche_n <- function(eps) {
+   n=0
+   while( abs(1/3-an(n))>eps ) {
+     n = n + 1
+   } ; # fin de tant que
+   c( n, an(n) , an(n)-1/3)
+ } ; # fin de fonction cherche_n
>
>
> cherche_n( 10^(-5) )
[1]  1.700000e+01  3.333282e-01 -5.086263e-06

# depuis la naissance du Christ, il y a eu
# 2005*365*24*60*60 secondes écoulée soit
# à peu près 63 milliards de secondes
# et sans doute 10^80 atomes dans l'univers

# remarque : lire les messages d'erreur de rstat
#            est important

# que signifient NA NaN et Inf ?

-- fin du TP2, ouf !
```