

ASI (L2) : TP5 Programmation sous *Excel*

Objectifs du TP :

Savoir programmer des fonctions et des boucles sous *Excel* ; savoir bien gérer les relations de récurrence sous Excel.

1. Programmation avec *Excel*

1.1 Fonctions rapides

Montrer comment on écrit une fonction sous *Excel* à l'aide de

Outils / Macros / Visual Basic Editor

pour faire des ORF ("one-result functions") c'est à dire des fonctions qui renvoient une seule valeur comme pour

- *carre*(x) renvoie le carré de l'argument,
- *duree*(a) renvoie nb de secondes écoulées entre an 0 et an a ,

Remarque : A la fac, il faut sans doute faire

Insertion / Module

avant de pouvoir écrire le texte de la fonction.

Leur faire écrire les deux textes suivants :

```
Function carre(n)
' donne le carré d'un nombre, d'une case...
carre = n * n
End Function
```

```
Function duree(a)
' essaie de calculer le nombre de secondes écoulées
' entre la naissance du Christ et l'an fourni
' comme paramètre
duree = 365 * (a - 1) * 12 * 24 * 60 * 60
End Function
```

Tester ce que fait *carre* pour un nombre, une plage de valeurs. Leur faire écrire une fonction *cdv* qui renvoie ECARTYPEP/MOYENNE si l'argument est une plage de valeurs.

Rappeler et faire vérifier la distinction entre variance exacte (échantillon) donnée par VAR.P et variance estimée (VAR). Utiliser les valeurs 1, 2, 3, 4, 5, 6 c'est à dire la loi uniforme discrète $\mathcal{UD}(6)$ ou encore "loi de lancer d'un dé" pour voir le rapport $(n - 1)/n$ entre ces deux variances.

Ces fonctions ne font aucun test, aucune boucle. Elles se contentent de calculer ou d'appeler des fonctions d'*Excel*.

Voir l'onglet Fonctions de la feuille *Excel* nommée **funcs** sur K:\Stat_ad pour voir où mettre les valeurs...

1.2 Eviter les fonctions avec *Excel*

Comme la programmation n'est pas toujours simple avec *VBA*, on peut souvent se contenter de mettre des formules dans des cellules ce qui remplace la programmation.

Rappeler comment afficher la loi binomiale sous *Excel* selon le modèle de l'onglet Binom de la feuille *Excel* nommée **funcs** sur K:\Stat_ad à savoir : On met n en B1, p en D1 et on affiche les x_i en colonne 2, les p_i en colonne 3 via LOI.BINOMIALE et en colonne 4 à la main via $C_n^k p^k (1 - p)^{n-k}$; prendre $n = 5$, $p = 0.3$.

Là encore, on se servira de la page *Web*

<http://www.info.univ-angers.fr/pub/gh/vitrine/Democgi/loisStatp.htm>

pour vérifier les résultats.

Même chose pour la loi de *Poisson*.

On ajoutera en colonne 5 le calcul par la formule de récurrence d'un terme à l'autre à savoir : à l'ordre $k + 1$ on multiplie par λ et on divise par $k + 1$ (voir l'onglet *Poisson*).

Reprendre $\lambda = 1.5$ pour $\tau = 10$ termes (montrer comment on obtient λ et τ à partir de l et t via la police *Symboles*).

1.3 Fonctions avancées en *Excel*

Montrer comment écrire un programme de type *Sub* qui écrit 4 cellules avec un bouton pour lancer l'exécution du programme (qui au passage demande la plage de variation des cellules). Expliquer pourquoi on met *Application* dans *Averag* et *StDev*. Le fichier *VBALIST.XLS* contient la correspondance Anglais/Français et permet d'avoir une idée des fonctions à utiliser.

Voici le programme :

```
Sub decrit()  
  
    ' demande de la plage  
  
    Set r = Application.InputBox(  
        Prompt:="Sélectionner la plage de recherche",  
        Type:=8  
    )  
  
    ' calculs  
  
    lamoy = Application.Average(r)  
    sonect = Application.StDev(r)
```

```

' stockage dans les cellules

Worksheets(1).Cells(14, 2) = "moyenne"
Worksheets(1).Cells(14, 3) = lamoy
Worksheets(1).Cells(14, 4) = "écart-type"
Worksheets(1).Cells(14, 5) = sonect

End Sub

```

Une fois le programme saisi (et enregistré, ce qui montre les erreurs éventuelles), il faut faire apparaître la barre d'outils *Formulaires*, cliquer sur le symbole du bouton et le dessiner (aux environs de B4). Lorsque le panneau "affecter" apparaît, lui dire que c'est la macro *decrit* qui est associée au bouton. C'est pourquoi il vaut mieux écrire la fonction (quitte à n'écrire que la ligne de renvoi de la valeur) et ensuite le bouton.

Changer le texte du bouton puis vérifier que le programme marche en cliquant sur le bouton. Avec l'onglet **Sub** de la feuille proposée, c'est la plage **\$B\$1:\$K\$1** qu'il faut fournir comme zone d'entrée.

Leur faire chercher pourquoi on ne voit rien... si on l'exécute dans un onglet autre que l'onglet 1 et leur faire rajouter `MsgBox ("et voila ")` pour bien voir que le programme s'exécute.

Leur faire modifier le programme pour qu'il affiche au bon endroit, leur faire rajouter le nombre de termes et le *cdv* (coefficient de variation, déjà programmé). Détailler ce qui se passe si on clique sur Ok sans donner de plage de variation. Ne pas détailler le débogage.

Ne pas oublier d'indiquer que `&` permet de concaténer les chaînes,

1.4 Pour ceux qui voudraient approfondir :

Rappeler ce qu'est un test en **si** en algorithmique et donner la traduction en *Excel*. Faire un programme (sub) *testevaleurs* qui refuse de calculer si n n'est pas un entier, si p n'est pas un réel compris entre 0 et 1 ; n étant par exemple en B1 et p en D1.

Rappeler ce qu'est une boucle **pour** en algorithmique et donner la traduction en *Excel*. Ecrire un programme (sub) *sdpe* qui calcule la somme des n premiers entiers. Ne pas insister sur les variables locales, donner un algorithme rapide.

Rappeler ce qu'est une boucle **tant que** en algorithmique et donner la traduction en *Excel*. Ecrire un programme (sub) *dif falphan* qui donne affiche les valeurs de n et α_n tant que $|\alpha_n - 1/3| > 10^{-5}$ où α_n correspond à la position du sprite.

On rappelle que

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

On trouvera une solution dans la feuille `subs.xls` du disque K: à l'emplacement habituel.

`Dim` permet de déclarer des variables, dont les tableaux, exemple :

```
Dim Armoire(3)
  Armoire(1) = "Assiette"
  Armoire(2) = "Verre"
Dim Tableau(1 To 10) As Integer
  ' attention : Tableau(3) = "test" G\{e}n\{e}re une erreur
  Tableau(4) = 775
```

Ecrire un programme qui affiche la valeur du maximum et sa position, la plage de variation étant transférée dans un tableau (si on ne veut pas de question utilisateur, mettre 5 valeurs dans un tableau).

Pour ceux qui vont vite, leur faire lire `essais_excel.txt` et les laisser essayer les fonctions proposées.

Esquisse de solution en *Excel*

```
#####
```

```
Function carre(n)
```

```
#####
```

```
'donne le carré d'un nombre, d'une case...
```

```
carre = n * n
```

```
End Function
```

```
#####
```

```
Function duree(a)
```

```
#####
```

```
' essaie de calculer le nombre de secondes écoulées
```

```
' entre la naissance du Christ et l'an fourni
```

```
' comme paramètre
```

```
duree = 365 * (a - 1) * 12 * 24 * 60 * 60
```

```
End Function
```

```
#####
```

```
Function cdv(rge)
```

```
#####
```

```
' calcule le coefficient de variation
```

```
cdv = Application.StDev(rge) / Application.Average(rge)
```

```
End Function
```

```
#####
```

```
Sub decrit()
```

```
#####
```

```
Set r = Application.InputBox(  
    Prompt:="Sélectionner la plage de recherche", Type:=8)
```

```
If r.Count() < 2 Then  
    MsgBox "désolé, vous n'avez pas assez de valeurs !"  
End If
```

```
' calculs  
m = Application.Average(r)  
e = Application.StDev(r)  
' stockage  
Worksheets(1).Cells(14, 3) = "longueur"  
Worksheets(1).Cells(14, 4) = r.Count()  
Worksheets(1).Cells(14, 5) = "moyenne"  
Worksheets(1).Cells(14, 6) = m  
Worksheets(1).Cells(14, 7) = "écart-type"  
Worksheets(1).Cells(14, 8) = e  
Worksheets(1).Cells(14, 9) = "cdv"  
Worksheets(1).Cells(14, 10) = cdv(r)  
MsgBox ("et voilà")  
End Sub
```

```
#####
```

```
Sub Binomiale()
```

```
#####
```

```
' récupération des valeurs  
  
valn = Worksheets("testevaleurs").Cells(1, 2).Value  
valp = Worksheets("testevaleurs").Cells(1, 4).Value
```

```

' tests sur n

If IsEmpty(Worksheets("testevaleurs").Cells(1, 2)) Then
    MsgBox ("la case B1 est vide ")
    Exit Sub
End If

If Not IsNumeric(valn) Then
    MsgBox ("la valeur pour n, à savoir " & valn & " n'est pas numérique")
    Exit Sub
End If

If Not (valn = Int(valn)) Then
    MsgBox ("votre n " & " qui est " & valn & " ne doit pas être un entier...")
    Exit Sub
End If

' tests sur p

If IsEmpty(Worksheets("testevaleurs").Cells(1, 4)) Then
    MsgBox ("la case D1 est vide ")
    Exit Sub
End If

If Not IsNumeric(valp) Then
    MsgBox ("la valeur pour p, à savoir " & valp & " n'est pas numérique")
    Exit Sub
End If

If (valp < 0) Or (valp > 1) Then
    MsgBox ("la valeur de p soit " & valp & " n'est pas comprise entre 0 et 1")
    Exit Sub
End If

' si on arrive ici, c'est que tout va bien

MsgBox (" tout va bien, on pourrait calculer B(" & valn & " ; " & valp & ")")
End Sub

```



```
#####
```

```
Sub resetDecrit()
```

```
#####
```

```
Worksheets("bonne Sub").Cells(14, 4) = ""  
Worksheets("bonne Sub").Cells(14, 6) = ""  
Worksheets("bonne Sub").Cells(14, 8) = ""  
Worksheets("bonne Sub").Cells(14, 10) = ""  
End Sub
```

```
#####
```

```
Function sdpe(n)
```

```
#####
```

```
sdv = 0 ' somme des valeurs  
For indi = 1 To n  
    sdv = sdv + indi  
Next indi  
sdpe = sdv  
End Function
```

```
#####
```

```
Sub resetAlphan()
```

```
#####
```

```
nblig = 40  
nbcoll = 40
```

```

For indi = 1 To nblig
  For jndj = 1 To nbcol
    Worksheets("diffalphan").Cells(indi, jndj) = ""
  Next jndj
Next indi
End Sub

#####

Function Alphan(n)

#####

  Alphan = (1 / 3) + (2 / 3) * ((-1 / 2) ^ n)
End Function

#####

Sub DiffAlphan()

#####

Worksheets("diffalphan").Cells(10, 3) = "n"
Worksheets("diffalphan").Cells(10, 4) = "alphan"
Worksheets("diffalphan").Cells(10, 5) = "|alphan-1/3|"
n = 0
While Abs(Alphan(n) - 1 / 3) > 10 ^ (-5)
  n = n + 1
  Worksheets("diffalphan").Cells(10 + n, 3) = n
  Worksheets("diffalphan").Cells(10 + n, 4) = Alphan(n)
  Worksheets("diffalphan").Cells(10 + n, 5) = Abs(Alphan(n) - 1 / 3)
Wend
End Sub

```

REFERENCE WEB :

<http://www.info-3000.com/vbvba/tableau.php>

=====

Les tableaux, exemple basique

```
Sub Tableau()  
  Dim Armoire(3)  
  Armoire(1) = "Assiette"  
  Armoire(2) = "Verre"  
  Armoire(3) = "Serviette"  
  MsgBox Armoire(1)  
  MsgBox Armoire(2)  
  MsgBox Armoire(3)  
End Sub
```

----- Exemple de tableau typé

```
Sub tableauV2()  
  Dim Tableau(1 To 10) As Integer  
  ' Tableau(3) = "test" Génère une erreur  
  Tableau(4) = 775  
End Sub
```

```
Sub ChercheMaxDansTableau()  
  
  Dim monTableau(5)  
  Nbval = 5  
  leMax = 1  
  monTableau(1) = 3 ; monTableau(2) = 8 ; monTableau(3) = 8 etc.  
  leMax = monTableau(1)  
  For indval = 1 To Nbval  
    If monTableau(indval) > leMax Then  
      leMax = monTableau(indval)  
    End If  
  Next indval  
  MsgBox (" le max est " & leMax)  
End Sub
```

```

*****
*                                     *
* Fichier  essais_excel.txt         *
*                                     *
*****

```

```

Attribute VB_Name = "EnvironnementEtLoginName"
'renseignements sur l'environnement local et réseau

```

```

'Ron de Bruin, mpep

```

```

Sub PCInformation()

```

```

Dim msg

```

```

    msg = "UserName" & vbTab & Environ$("username") & vbNewLine _
          & "UserProfile" & vbTab & Environ("UserProfile") & vbNewLine _
          & "Computer #" & vbTab & Environ$("ComputerName") & vbNewLine _
          & "Logon Server" & vbTab & Environ$("Logonserver") & vbNewLine _
          & "UserDomain " & vbTab & Environ$("UserDomain")

```

```

    MsgBox msg, , "Environment Variables"

```

```

End Sub

```

```

'suite

```

```

Sub RetrieveLogonName()

```

```

Dim wshNetwork As Object

```

```

Dim LogonName As Variant

```

```

    Set wshNetwork = CreateObject("WScript.Network")
    LogonName = "Logon Name = " & wshNetwork.UserName
    MsgBox LogonName

```

```

End Sub

```

```

=====

```

```

Dim new_value As String

```

```

Dim txt As String

```

```

Dim i As Integer

```

```

Dim Buffer() As Byte

```

```

Dim Addr As Long

```

```

Dim sTemp As String

```

```

Dim i As Lon

```

```

Dim TLApp As TLI.TLIApplication
Dim TLInfo_XL As TLI.TypeLibInfo
Dim MemInfo As TLI.MemberInfo
Dim ConstInfo As TLI.ConstantInfo
Dim DestCell As Range
Dim hpl$
Dim MyData As DataObject

```

```
=====
```

```
Attribute VB_Name = "AvecDoublonsOuNon"
```

```
'savoir si une plage comporte ou non des doublons
```

```
Function HasDoublons(Plage As Range)
```

```
Dim Coll As New Collection, cell As Range
```

```
On Error Resume Next
```

```
For Each cell In Plage
```

```
    If cell.Text <> "" Then Coll.Add "zaza", cell.Text
```

```
Next
```

```
Err.Clear
```

```
HasDoublons = Not (Coll.Count = Plage.Count)
```

```
End Function
```

```
=====
```

```
Attribute VB_Name = "ChercherDansUneChaine"
```

```
'Comment je peux tester facilement la présence d'un caractère dans une chaîne?
```

```
'la petite fonction suivante qui renvoie VRAI si Caractere est
```

```
'trouvé dans Chaine (Pierre Fauconnier, mpfe)
```

```
Function RechercheCar(Chaine, Caractere) As Boolean
```

```
Dim i As Integer
```

```
For i = 1 To Len(Chaine)
```

```
    If Mid(Chaine, i, 1) = Caractere Then RechercheCar = True
```

```
Next i
```

```
End Function
```

```

=====
'Y a t'il une fonction VBA qui renvoie le nombre d'occurrences
'd'un caractère trouvées dans une chaîne

Function nbOcc(TexteCherché As String, Texte As String) As Integer

'Iznogood, mpfe
Dim i As Long
Do
    i = InStr(i + 1, Texte, TexteCherché, 1)
    If i <> 0 Then nbOcc = nbOcc + 1
Loop Until i = 0
End Function

```

```

=====
' Retourne le nombre d'occurrences du 1er caractère de Texte2 dans Texte1
' utilisation de references-cellule et/ou textes entre quotes doubles.
' ex: NbrOccurrences(A5;A3)
' ou NbrOccurrences("Le blaba abcdefgh";"e")

```

```

Function NbrOccurrences(Texte1, Texte2)

'GeeDee, mpfe
NbrOccurrences = 0
For i = 1 To Len(Texte1)
    If Mid(Texte1, i, 1) = Left(Texte2, 1) Then NbrOccurrences = NbrOccurrences + 1
Next
End Function

```

```

'Sans VBA (JièL, mpfe)
'=NBCAR(A1)-NBCAR(SUBSTITUE(MINUSCULE(A1);"e";""))

```

```

Const Symb = "IVXLCDM"
Dim I As Integer, J As Integer
Dim K As Integer, L As Integer, S As Integer
Dim C As String * 1, Prec As Boolean

```

```
=====
Attribute VB_Name = "ChiffresRomainsChiffresArabes1"
```

```
'-----
' Conversion d'un nombre < 4000 en chiffres romains (style "classique")
' vers un nombre en chiffres arabes
'-----
'Laurent Longre, mpfe
```

```
Function ROMINVERSE(Nombre As String)
```

```
    Const Symb = "IVXLCDM"
    Dim I As Integer, J As Integer
    Dim K As Integer, L As Integer, S As Integer
    Dim C As String * 1, Prec As Boolean

    On Error GoTo Erreur
    I = Len(Nombre)
    Do
        K = InStr(1, Symb, Mid$(Nombre, I, 1))
        If K = 0 Or K = J Then Err.Raise xlErrValue
        S = IIf(K Mod 2, 1, 5) * 10 ^ ((K - 1) \ 2)
        If K < J Then
            If Not Prec Then Err.Raise xlErrValue
            Select Case Mid$(Nombre, I, 2)
                Case Is = "ID", Is = "IM", Is = "VX", Is = "VD", _
                    Is = "VM", Is = "LC", Is = "DM"
                    Err.Raise xlErrValue
            End Select
            ROMINVERSE = ROMINVERSE - S
            I = I - 1
            Prec = False
        ElseIf K Mod 2 Then
            C = Mid$(Symb, K, 1)
            L = 0
            Do
                If Mid$(Nombre, I, 1) = C Then
                    If L = 3 Then Err.Raise xlErrValue
                    ROMINVERSE = ROMINVERSE + S
                    I = I - 1
                End If
            Loop
        End Do
    Loop
```

```

        L = L + 1
    Else
        Prec = L = 1
        Exit Do
    End If
    Loop While I
Else
    ROMINVERSE = ROMINVERSE + S
    I = I - 1
    Prec = True
End If
J = K
Loop While I
Exit Function

```

Erreur:

```

    ROMINVERSE = CVerErr(Err)

```

End Function

=====

Attribute VB_Name = "CompareChaines"

```

'Eric Jeanne, mpfe
'compare lettre à lettre 2 chaines de caractères
'renvoie -1 si elles sont identiques,
'ou le nombre de caractères qu'elles ont en commun,
'de la gauche vers la droite (sensible à la casse)

```

Function comparCh(ch1, ch2)

```

Dim lg1%, lg2%, i%

```

```

    Application.Volatile True

```

```

    If ch1 = ch2 Then
        comparCh = -1
    
```

```

    Else
        lg1 = Len(ch1)
    
```



```
lg2 = Len(ch2)

i = 1
  While Mid(ch1, i, 1) = Mid(ch2, i, 1)
    i = i + 1
  Wend

  comparCh = i - 1

End If

End Function

Sub test()
  MsgBox comparCh("paul za", "paul zaza")
  MsgBox comparCh("paul za", "paul Zaza")
End Sub

=====
```