

I.T.R.M.L.M.  
B.P. 30  
Papeete  
TAHITI

O.R.S.T.O.M.  
B.P. 529  
Papeete  
TAHITI

MODELE DE KIRITANI ET NAKASUJI (MODIFIE PAR MANLY) POUR  
L'ETUDE DES POPULATIONS ANIMALES SE DEVELOPPANT PAR  
STADES  
PROGRAMME BASIC D'APPLICATION

F. LARDEUX

Juin 1989

Ref. ITRMLM : n° 27 / 89 / ITRM / Doc-Ent.

## SOMMAIRE

### INTRODUCTION

#### 1 - LE MODELE

1.1 - Notations

1.2 - Théorie

1.3 - Estimations

1.3.1 - Estimation des paramètres

1.3.2 - Estimation des variances

#### 2 - LE PROGRAMME INFORMATIQUE

2.1 - Description générale

2.2 - Mise en route

2.3 - Saisie des données - création de fichier

2.4 - Corrections dans un fichier

2.5 - Listage d'un fichier

2.6 - Estimation des paramètres

### CONCLUSION

### BIBLIOGRAPHIE

Annexe : Listing du programme KINAMANI

## INTRODUCTION

Il est difficile d'estimer les paramètres démographiques des stades de développement d'une population animale lorsqu'ils sont mêlés. Les techniques de marquage-recapture ont résolu en partie ce problème. Souvent cependant, ces techniques sont difficiles à mettre en oeuvre ou inadaptées à l'espèce étudiée. Une alternative est de compter le nombre d'individus de chaque stade à des périodes données. Les paramètres démographiques peuvent alors être estimés selon plusieurs méthodes (Richards et Waloff 1954, Richards et al. 1960, Dempster 1956, 1961, Ashford et al. 1970, Lefkovitch 1965, Kiritani et Nakasuji 1967, Kobayashi 1968, Manly 1974 etc...).

La méthode qui fait l'objet du programme informatique décrit ici est celle de Kiritani et Nakasuji (1967), modifiée par Manly (1976, 1977). Selon ce dernier auteur, elle fournit, malgré sa simplicité, des résultats précis et fiables. De plus, son application ne requiert pas une maille d'échantillonnage temporelle régulière.

Le programme informatique a été écrit en Quick-Basic (Microsoft). Une version compilée est disponible sur demande pour ceux qui ne possèdent pas ce compilateur.

### 1 - LE MODELE

On reprend, dans ce paragraphe, les équations fournies par Manly.

#### 1.1 - Notations

Soit une population animale qui se développe selon  $q$  stades.

Pour un stade particulier, on note :

$M$  = le nombre d'animaux entrant dans le stade

$a$  = la durée du stade (supposée égale pour tous les animaux)

$e^{-\theta}$  =  $k$  = le taux de survie

$e^{-\theta a}$  = le taux de survie spécifique du stade

$\mu$  = la date moyenne d'entrée dans le stade

#### 1.2 - La théorie

A la suite d'échantillonnages, on peut se trouver dans le cas illustré par le tableau 1. Ce tableau donne les estimations du nombre d'animaux pour différents stades en fonction du temps. C'est un exemple typique, dont la situation est illustrée à la figure 1. Celle-ci représente les courbes de fréquence des différents stades de la population (de moustiques par exemple) en fonction du temps. On supposera aussi que les fluctuations

enregistrées d'un échantillon à l'autre sont dues en grande partie aux erreurs d'échantillonnage.

Date	t	Stade 1	Stade 2	Stade 3	Stade 4	Adultes
	-4	0				
Mai 20	1	3497				
25	6	4196				
29	10	7000				
Juin 3	15	5000	500			
10	22	3500	2500	500		
15	27	500	5000	0		
18	30	500	4000	500	500	
22	34	1500	1000	2000	1000	
25	37	3500	6000	3000	0	
29	41	0	3500	3000	3000	
Juil 2	44	500	500	3000	2000	500
9	51		500	1500	1000	500
13	55		2000	2000	2000	2500
.	.		.	.	.	.
.	.		0	.	.	.
.	.			0	.	.
Sep 23	127					500
	140					0

Tableau 1 : exemple de données stades-fréquences

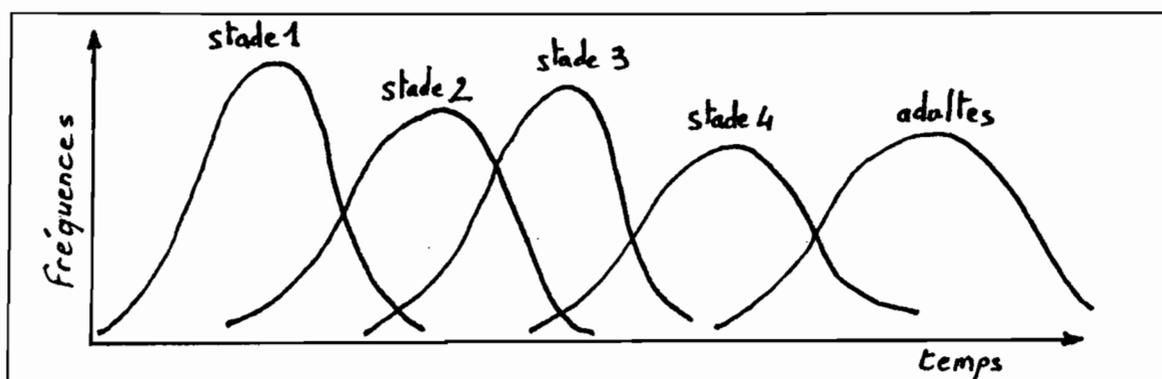


Figure 1 : exemple de courbes stades-fréquences

Soit  $A_i$  l'aire comprise sous la courbe de fréquence du stade  $i$ .

On a 
$$A_i = M_i \cdot (k^{ai} - 1) / \ln(k)$$

Si il y a  $q$  stades, la durée du dernier stade (le  $q$  ième) est alors infinie. Aussi, on a  $M_{i+1} = k^{ai} \cdot M_i$  d'où

$$A_j = -M_i / \ln(k) = M_i \cdot (1 - e^{-\theta a}) / \theta \quad (1)$$

d'où, le taux de survie spécifique du stade  $i$  :

$$s_i = k^{ai} = 1 - A_i / \sum_{j=i}^n A_j$$

Cette dernière équation est à la base de la méthode de Kiritani et Nakasuji, avec l'estimation de l'aire  $A_i$  faite par l'addition des fréquences pour le stade  $i$  et la multiplication par l'intervalle de temps entre deux échantillonnages. Cette façon de procéder n'est malheureusement valide que lorsque les échantillons sont pris à des intervalles de temps réguliers.

Manly (1976, 1977) a proposé une alternative permettant d'échantillonner à des intervalles de temps irréguliers. En effet, si on suppose :

- que les insectes commencent à entrer dans le stade  $i$  juste après un temps  $t_1$  et qu'ils ont tous quitté ce stade juste après un temps  $t_2$
- que des estimations des fréquences  $f_1=0, f_2, \dots, f_{n-1}, f_n=0$  aux temps  $t_1, t_2, \dots, t_{n-1}, t_n$  sont disponibles (comme précédemment dans le tableau 1 par exemple), alors l'aire sous la courbe de fréquence devient :

$$A = 1/2 \cdot \sum_{j=1}^n (h_j + h_{j+1}) \cdot f_j \quad (2)$$

où  $h_j = t_j - t_{j-1}$

Manly (1977) propose une autre statistique, appelée  $B$ , qui estime l'aire sous la courbe [ $t$  multiplié par la courbe de fréquence]. Il montre que :

$$B = 1/\theta + \mu - a \cdot e^{-\theta a} / (1 - e^{-\theta a}) \quad (3)$$

ou encore 
$$B = 1/2 \cdot \sum_{j=1}^n (h_j + h_{j+1}) \cdot t_j \cdot f_j / A \quad (4)$$

En égalant les équations (1) et (2) d'une part et (3) et (4) d'autre part, il est possible d'obtenir les paramètres démographiques de la population.

### 1.3 - Estimations

#### 1.3.1 - Estimation des paramètres

Manly (1977) propose trois cas de figure :

- Estimation lorsque tous les stades peuvent être estimés
- Estimation lorsque le dernier stade est manquant
- Estimation lorsque le dernier stade est manquant mais qu'il n'y a pas de mortalité

On ne s'intéressera qu'au premier cas, qui seul a fait l'objet de la programmation informatique. Les deux derniers cas sont en cours de programmation.

Lorsque tous les stades sont estimables, et si on suppose que le taux de survie ( $e^{-\theta}$ ) est constant pour tous les stades, alors si les stades  $i, i+1, \dots, q$  sont sommés pour ne former qu'un seul stade (noté  $i$ ), les valeurs de A et B des équations (1) et (3) deviennent :

$$\begin{aligned} A_i &= M_i / \theta \\ B_i &= 1/\theta + \mu_i \end{aligned} \quad (5)$$

où  $M_i$  est le nombre d'animaux qui entrent dans le stade  $i$  et  $\mu_i$  la date moyenne d'entrée dans le stade.

Ce stade est de durée infinie si les animaux quittent le  $q$  ième stade par décès seulement.

Si maintenant on ne tient plus compte du stade  $i$  mais que l'on somme les stades  $i+1, \dots, q$  alors les valeurs de A et B deviennent :

$$\begin{aligned} A_{i+1} &= M_{i+1} / \theta \\ B_{i+1} &= 1/\theta + \mu_{i+1} = 1/\theta + \mu_i + a_i \end{aligned} \quad (6)$$

Des équations (5) et (6), les paramètres de la population sont estimés par :

$$\begin{aligned} w_i &= e^{-\theta a_i} = A_{i+1} / A_i \quad (\text{taux de survie spécifique du stade } i) \\ a_i &= B_{i+1} - B_i \quad (\text{durée du stade } i) \\ \theta &= - \text{Ln}(w_i) / a_i \quad (\text{paramètre de survie}) \\ \mu_i &= B_i - 1/\theta \quad (\text{date moyenne d'entrée dans le stade}) \\ M_i &= A_i \cdot \theta \quad (\text{Nombre d'animaux entrant dans le stade}) \end{aligned}$$

### 1.3.2 - Estimation des variances

Manly (1977) propose d'estimer les variances des paramètres par la méthode du jackknife (Quenouille 1949, 1956, Tukey 1958).

Dans cette méthode, l'idée de base est de diviser les données en  $r$  sous-échantillons de taille comparable, au hasard si nécessaire. Le premier sous échantillon est alors retiré de l'ensemble des données et le paramètre que l'on doit calculé est estimé avec les données restantes. Ceci donne un premier paramètre "partiel". Le sous-échantillon éliminé précédemment est alors remis dans l'ensemble des sous échantillons et le second sous échantillon est retiré. Un second paramètre "partiel" est alors calculé, comme précédemment. Cette procédure est répétée pour tous les sous-échantillons.

Si on appelle  $\theta$  le paramètre que l'on désire estimer, on a donc :

- .  $r$  paramètres "partiels", noté  $\theta_{-j}$  ( $j = 1, \dots, r$ )
- . un paramètre  $\theta^*$  estimé à partir de tout l'échantillon.

Avec ces valeurs, on peut former r pseudo-valeurs (notées  $\theta_j$ ) calculées comme suit :

$$\theta_j = r.\theta_* - (r-1).\theta_{-j} \quad (j = 1, \dots, r)$$

La moyenne de ces pseudo-valeurs donne une estimation du paramètre  $\theta$  à estimer :

$$\hat{\theta} = 1/r \cdot \sum_{j=1}^r \theta_j$$

La variance de cette moyenne (erreur standard) peut aussi être estimée à partir de ces pseudo-valeurs par :

$$\text{var} = 1/(r.(r-1)) \cdot \sum_{j=1}^r (\theta_j - \hat{\theta})^2$$

En pratique, on choisira r (le nombre de sous-échantillons) aussi grand que possible.

On notera que les valeurs trouvées par la méthode du jackknife peuvent ne pas correspondre à celles trouvées lorsque l'échantillon est pris dans son ensemble (c'est à dire, sans calculer de jackknife). Toutefois, si les données de fréquence des stades disponibles pour l'analyse sont correctes, ce problème est inexistant et la méthode du jackknife tout à fait juste et précise. Elle permet d'estimer alors facilement les variances liées aux paramètres.

## 2 - LE PROGRAMME INFORMATIQUE

### 2.1 - Description générale

Le programme proposé ici a été écrit en Quick-Basic (Microsoft) sur un compatible IBM PC/AT.

Il effectue les tâches suivantes :

- création de fichiers spécifiques pour les données à analyser.
- corrections d'erreurs dans un fichier de données.
- listage (écran et imprimante) des données.
- calcul des paramètres démographiques de la population animale étudiée, dans le cas où tous les stades de la population sont estimés, :
  - selon les formules du paragraphe 1.3.1 (sans la méthode du jackknife)
  - avec la méthode du jackknife (les variances sont alors estimées).

Le programme a été écrit dans l'optique d'une utilisation la plus conviviale possible. L'utilisateur n'a pas besoin de connaissances en programmation : il doit simplement répondre aux questions posées en clair par le programme.

## 2.2 - Mise en route

Le programme porte le nom de KINAMAN1. Il ne peut fonctionner qu'avec un fichier ASCII séquentiel qui contient les paramètres de dimensionnement pour les variables utilisées par le programme. Ce fichier doit s'appeler KINAMAN1.PAR et doit se trouver dans le même répertoire que le programme.

Le programme Basic doit être dans le même répertoire que Quick-Basic pour être exécuté. Il suffit alors de charger le Quick-Basic puis de charger le programme KINAMAN1 et de l'exécuter.

Pour la version compilée, se mettre dans le répertoire où se trouve le programme et tapez KINAMAN1 puis faites un "retour-chariot".

Le fichier des paramètres (KINAMAN1.PAR) contient les trois valeurs entières suivantes :

- st = nombre de stades dans la population.
- jo = nombre d'occasions d'échantillonnages.
- gr = nombre de groupes à faire pour le calcul impliquant la méthode du jackknife.

Il permet de moduler la place mémoire disponible sur votre machine en donnant des valeurs par défaut aux tailles des variables utilisées par le programme.

Une fois le programme lancé, il suffit de suivre les indications portées à l'écran.

Un premier écran apparait (fig.2), indiquant le titre du programme, la publication source, le nom et l'adresse du programmeur et la date de création. Il suffit de faire un "Retour-chariot" pour passer à l'écran suivant.

Ce second écran permet justement de modifier le fichier KINAMAN1.PAR si les valeurs par défaut sont trop grandes (taille mémoire insuffisante) ou trop petites (votre problème précis nécessite des variables plus grandes).

Le programme affiche d'abord les valeurs contenues dans le fichier des paramètres et vous demande si vous désirez les modifier (fig. 3). Si la réponse est n (= non), le programme passe directement au "MENU GENERAL" (créations de fichiers, corrections, listages, calculs). Dans le cas contraire, il vous est possible de modifier ces paramètres. Il suffit de répondre simplement aux questions du programme. La modification des valeurs par défaut peut être conservée dans le fichier des paramètres ou être simplement effective pour la durée de la session (il suffit alors de répondre n (=non) à la question " Voulez vous les stocker dans le fichier des paramètres ? (o/n) : " (fig. 4)

Le programme passe ensuite au "MENU GENERAL" (fig. 5). Les options proposées sont choisies en indiquant simplement le numéro voulu et en faisant un "Retour-chariot".

Les paragraphes suivants analysent en détails ces différentes options.

Remarque 1 : A la fin d'exécution de chaque option, le programme revient automatiquement à ce menu (sauf si l'option "fin

```
-----  
ANALYSE DE DONNEES STADES - FREQUENCES  
-----  
  
Source : MANLY B.J.F. - 1977 - A further note on Kiritani and Nakasuji's  
model for stage-frequency data including  
comments on the use of Tukey's jackknife  
technique for estimating variances  
  
Res. Popul. Ecol. 10 : 177 - 186  
  
Programme F. LARDEUX - Centre ORSTOM de Tahiti - B.P. 529 Papeete  
version mai 1989  
  
Faites un 'Retour-chariot' pour continuer ....
```

**Figure 2 : Ecran de présentation du programme KINAMAN1.**

```
Les variables définissant la place mémoire nécessaire sont :  
  
Le nombre de stades dans la population étudiée  
Ce nombre est actuellement initialisé à : 24  
  
Le nombre total d'occasions d'échantillonnage.  
Ce nombre est actuellement initialisé à : 199  
  
Le nombre de groupe demandés pour le 'jackknife'  
Ce nombre est actuellement initialisé à : 39  
  
Si ces valeurs sont trop grandes pour la taille mémoire de votre machine,  
vous pouvez les modifier  
  
Voulez vous les modifier ? (o/n) :
```

**Figure 3 : Valeurs stockées dans le fichier des paramètres.**

```

Le nombre de stades dans la population étudiée
Ce nombre est actuellement initialisé à : 25

Le nombre total d'occasions d'échantillonnage.
Ce nombre est actuellement initialisé à : 200

Le nombre de groupe demandés pour le 'jackknife'
Ce nombre est actuellement initialisé à : 40

Si ces valeurs sont trop grandes pour la taille mémoire de votre machine,
vous pouvez les modifier

          Voulez vous les modifier ? (o/n) : 0

Nombre de stades                : 30
Nombre d'occasions d'échantillonnage : 150
Nombre de groupes pour le jackknife  : 45

Voulez vous les stocker dans le fichier paramètres ? (o/n) :

```

**Figure 4 : Modifications du fichier des paramètres.**

```

                                MENU GENERAL

Saisie des données ..... 1
Correction dans un fichier ..... 2
Listage d'un fichier ..... 3
Calculs ..... 4

Fin de la session ..... 5

          Votre choix (numéro) :

```

**Figure 5 : Menu général d'utilisation du programme KINAMAN1**

de session" est choisie, bien évidemment). Lorsqu'une nouvelle option est choisie, le programme vous demande de confirmer que vous travaillez bien avec le même fichier (fig. 6). Si vous désirez changer de fichier, il suffit de l'indiquer au programme. Lors du premier choix, la question est, bien sur, "Nom du fichier des données " (fig .7). Il faut, dans chaque cas (premier "run" ou "run" ultérieurs), préciser le nom du fichier avec son chemin d'accès complet. Par exemple, si votre fichier est sur une disquette dans le lecteur A, il faut écrire A:\ Nom-du-fichier. Si votre fichier est sur le disque dur C, dans un répertoire DATA par exemple, il faut écrire C:\DATA\Nom-du-fichier.

Remarque 2 : Lors de la création de fichier, le programme ajoute automatiquement l'extension .K11 au nom indiqué. Ceci permet de repérer plus facilement les fichiers créés et utilisables par ce programme. Le programme n'accepte pas des fichiers sans l'extension .K11. Par contre, lorsque le programme demande le nom d'un fichier, il ne faut pas donner l'extension .K11.

### 2.3 - Saisie des données - création de fichier

Cette option permet de créer des fichiers spécifiques à l'utilisation du programme KINAMAN1. Ces fichiers sont de format ASCII, séquentiels, dans lesquels les données sont stockées de la façon suivante :

nombre de stades,

pour i = 1 à stade

  pour j = 1 à nombre d'échantillons dans le stade i  
    écrire le couple (t, f)

où t = temps où l'échantillon a été pris

  f = fréquence des animaux de stade i au j<sup>ème</sup> échantillon

Par exemple, si les données du tableau 1 sont saisies, le fichier aura l'allure suivante :

```

nombre de stades
5, -4, 0, 1, 3497, 6, 4196, 10, 7000, ... 44, 500, 51, 0,
      (t, f)           (t, f)
      ▼
      stade 1

10, 0, 15, 500, 22, 2500, ....
      ▼
      stade 2

.....

41, 0, 44, 500, ....140, 0
      ▼
      stade 5 = adultes
  
```

Les données sont saisies stade par stade.

```

Le fichier de travail actuel est : A:\FLMANLY
voulez vous le changer ? (o/n) :

```

Figure 6 : Confirmation ou non du fichier de travail.

```

Nom du fichier des données (avec son chemin d'accès complet) :
A:\FLMANLY

```

Figure 7 : Demande de nom de fichier. Ici, le fichier FLMANLY se trouve sur une disquette dans le lecteur A.

stade n° 1			
Quantième	: -4	Nombre	: 0
Quantième	: 1	Nombre	: 20
Quantième	: 3	Nombre	: 25
Quantième	: 4	Nombre	: 30
Quantième	: 6	Nombre	: 15
Quantième	: 7	Nombre	: 5
Quantième	: 9	Nombre	: 0
Quantième	: 0	Nombre	: 0
stade n° 2			
Quantième	: 3	Nombre	: 1
Quantième	: 4	Nombre	: 5
Quantième	: 6	Nombre	: 20
Quantième	: 7	Nombre	: 45
Quantième	: 9	Nombre	: 60
Quantième	: 10	Nombre	: 26
Quantième	: 11	Nombre	: 5
Quantième	: 15	Nombre	: 0
Quantième	: 0	Nombre	: 0
stade n° 3			
Quantième	:	etc ...	

Figure 8 : Exemple de saisie de données.

Il faut que chaque stade débute par une fréquence 0 et se termine par une fréquence 0.

Ainsi, dans la structure de fichier précédente, on remarque que le stade 1 débute par le couple  $(t, f) = (-4, 0)$  et se termine par  $(51, 0)$ . De même, le dernier stade commence par  $(41, 0)$  et se termine par  $(140, 0)$ .

Lorsque la saisie d'un stade est terminée, faire "quantième = 0" et "Nombre = 0" pour arrêter le processus. Le programme passe alors à la saisie du stade suivant.

Ceci apparait à la figure 8 (copie d'écran pour un petit fichier fictif), où on constate que chaque stade débute et se termine par 0 et que l'on a saisi deux fois 0 ("quantième" et "nombre") pour passer au stade suivant.

Le processus est répété jusqu'à épuisement des stades. A la fin de la saisie, le programme demande le nom du fichier où stocker ces données (voir remarque 1 et 2 au paragraphe précédent).

Dans cette rubrique, il n'y a pas de possibilités de corriger de saisie erronée. Au cas où une erreur dans la saisie se produirait, il faut tout de même finir de rentrer les autres données, comme si tout était correct et, lorsque le fichier est créé, passer à l'option de "CORRECTION" du MENU GENERAL (cf. paragraphe suivant).

#### 2.4 - Corrections dans un fichier

L'option de correction permet de modifier des données déjà stockées dans un fichier.

Après avoir répondu à la question "Nom du fichier", le programme liste celui-ci à l'écran. Pour chaque écran affiché (c'est à dire pour chaque partie de fichier listée), le programme vous demande si des corrections sont à faire. Si la réponse est n (=non), le programme passe à un écran suivant en affichant la suite des données. Dans le cas contraire, le programme demande à quelle ligne et colonne ("temps" ou "fréquence") se trouve l'erreur et qu'elle est la nouvelle valeur à saisir (fig.9). Une fois l'erreur corrigée, le programme réaffiche l'écran corrigé et redemande si d'autres corrections sont à faire. Le processus est répété jusqu'à ce qu'il n'y ait plus aucune correction à faire (répondre "n" à la question "Avez vous des données à corriger dans cet écran (o/n) : "). Le programme affiche alors la suite des données et réitère sa demande de correction. Ce processus est répété jusqu'à ce que tout le fichier ait été passé en revue.

#### 2.5 - Listage d'un fichier

Cette option permet de lister un fichier à l'écran ou sur l'imprimante. Après avoir donné le nom du fichier à lister, le programme liste celui-ci à l'écran. En fin de fichier, le programme demande si le listage doit être reproduit sur l'imprimante (fig.10). En cas de réponse affirmative (o = oui), la sortie sur imprimante se fait comme à la figure 11.

```

stade : 1

n° ligne      colonne 1      colonne 2
              (quantième)  (nombre)

1             -4             0
2             1             7
3             6             8
4            10            14
5            15            10
6            22             7
7            27             1
8            30             1
9            34             3
10           37             7
11           41             0
12           44             1
13           51             0

Avez vous des données à corriger dans cet écran ? (o/n) : 0
n° de ligne où se situe l'erreur : 7
n° de colonne où se situe l'erreur : 2
valeur à corriger : 1      nouvelle valeur : 3

```

Figure 9 : Exemple de corrections dans un fichier.

jour	stade 1	stade 2	stade 3	stade 4	stade 5
-4	0	0	0	0	0
1	7	0	0	0	0
6	8	0	0	0	0
10	14	0	0	0	0
15	10	1	0	0	0
22	7	5	1	0	0
27	1	10	0	0	0
30	1	8	1	1	0
34	3	8	1	2	0
37	7	12	6	0	0
41	0	7	6	6	0
44	1	1	6	1	1
51	0	1	3	2	1
55	0	4	4	4	5
58	0	0	1	3	2

appuyez sur un touche pour continuer .....

---

jour	stade 1	stade 2	stade 3	stade 4	stade 5
140	0	0	0	0	0

appuyez sur un touche pour continuer .....

Voulez vous l'impression sur imprimante (o/n) :

Figure 10 : Exemple de sortie d'un fichier à l'écran.

jour	stade 1	stade 2	stade 3	stade 4	stade 5
-4	0	0	0	0	0
1	7	0	0	0	0
6	8	0	0	0	0
10	14	0	0	0	0
15	10	1	0	0	0
22	7	5	1	0	0
27	1	10	0	0	0
30	1	8	1	1	0
34	3	8	1	2	0
37	7	12	6	0	0
41	0	7	6	6	0
44	1	1	6	1	1
51	0	1	3	2	1
55	0	4	4	4	5
58	0	0	1	3	2
jour	stade 1	stade 2	stade 3	stade 4	stade 5
62	0	1	1	5	6
66	0	1	1	2	5
69	0	0	0	0	6
73	0	0	0	0	6
78	0	0	0	1	6
84	0	0	0	1	1
87	0	0	0	0	3
92	0	0	0	0	3
97	0	0	0	0	5
101	0	0	0	0	3
106	0	0	0	0	4
112	0	0	0	0	2
115	0	0	0	0	2
121	0	0	0	0	2
127	0	0	0	0	1
jour	stade 1	stade 2	stade 3	stade 4	stade 5
140	0	0	0	0	0

Figure 11 : Exemple de sortie d'un fichier sur l'imprimante.

```

Le fichier de travail actuel est : A:\FLMANLY
voulez vous le changer ? (o/n) : N

Voulez vous aussi les calculs par la méthode du jacknife (o/n) : 0

```

Figure 12 : Début de l'option de calcul.

## 2.6 - Estimation des paramètres

Après avoir répondu à la question du "nom de fichier", le programme demande si les calculs doivent se faire selon la méthode du jacknife ou non (fig.12).

En cas de réponse négative, le programme débute les calculs et ceux-ci apparaissent à l'écran selon le schéma de la figure 13.

En cas de réponse positive, le programme demande le nombre de groupes de données que l'on veut faire pour utiliser la méthode du jacknife (voir paragraphe 1.3.1). Le programme donne les limites possibles dans le choix du nombre de groupes (fig.14). Une fois ce nombre choisi, le programme affiche le type de partition obtenue dans les données (fig.14). Si l'utilisateur désire modifier son choix à la vue de cette partition, il en a la possibilité : dans ce cas, le programme lui redemande le nombre de groupes qu'il désire. Lorsque la partition obtenue est acceptée par l'utilisateur, le programme débute les calculs et les résultats sont affichés à l'écran selon le schéma de la figure 15.

Une possibilité de sortie sur imprimante est donnée. Répondre "o" à la question "Voulez vous l'impression sur l'imprimante (o/n) : ". Les résultats sont imprimés de la même façon qu'ils apparaissent à l'écran.

## CONCLUSION

Le programme présenté ici ne permet pas de traiter les deux autres cas de figure donnés par Manly (1977), à savoir, celui où le stade final est manquant dans les échantillons et celui où ce stade est manquant mais où il n'y a pas de mortalité. Toutefois, le programme permet d'estimer un certain nombre de paramètres démographiques dans le cas où tous les stades sont échantillonnés. La possibilité d'utiliser la méthode du jacknife permet l'estimation de variances. On aura cependant intérêt à comparer les résultats obtenus avec et sans jacknife, afin de tester la solidité du jacknife (donc de l'estimation des variances), sur les données qu'on analyse.

Le programme n'est certes pas optimal du point de vue du programmeur, mais il a l'avantage d'être convivial pour l'utilisateur et facilement modifiable en fonction des besoins. Le programme peut être obtenu sur disquette, ainsi que le jeu d'essai qui a servi dans l'illustration des options du programme. Une version compilée est aussi disponible, pour ceux qui ne possèdent pas le Quick-Basic.

---

RESULTATS pour le fichier a:\flmanly				
	stade 1	stade 2	stade 3	stade 4
Taux de survie du stade :	0.7264	0.6937	0.7577	0.7244
Erreur standart :	Non calculée car pas de jacknife			
Durée :	11.4969	10.2159	7.4930	8.1292
Erreur standart :	Non calculée car pas de jacknife			
Paramètre de survie :	0.0278	0.0358	0.0370	0.0397
Erreur standart :	Non calculée car pas de jacknife			
Date moyenne d'entrée :	10.0829	29.6062	40.7546	50.0342
Erreur standart :	Non calculée car pas de jacknife			
Nombre entrant dans le stade :	28.8105	26.9398	19.3332	15.6860
Erreur standart :	Non calculée car pas de jacknife			
Taux de survie journalier :	0.9726	0.9648	0.9636	0.9611
Erreur standart :	Non calculée car pas de jacknife			

**Figure 13** : Résultats des calculs sans l'utilisation de la méthode du jacknife

```

Calculs par la méthode du jacknife

Combien de groupes voulez vous faire ?

Ce nombre est compris entre 2 et 15 ou bien vaut : 31
Ma suggestion est 31 (mais les calculs sont plus longs)

Votre choix : 5
-----
Nombre d'éléments dans les 4 premiers groupes : 6
Nombre d'éléments dans le dernier groupe      : 7

D'accord pour cette partition ? (o/n) : o

```

**Figure 14** : Choix du nombre de groupes dans le calcul du jacknife

RESULTATS pour le fichier a:\fimanly				
	stade 1	stade 2	stade 3	stade 4
Taux de survie du stade :	0.7284	0.7028	0.7664	0.7252
Erreur standart :	0.1085	0.0800	0.0542	0.0123
Durée :	11.8386	9.6754	6.8380	7.8418
Erreur standart :	3.8907	3.7278	2.7580	1.0406
Paramètre de survie :	0.0267	0.0353	0.0370	0.0404
Erreur standart :	0.0045	0.0016	0.0047	0.0047
Date moyenne d'entrée :	9.6684	29.7849	41.1317	50.1789
Erreur standart :	3.3576	4.6009	3.3123	2.3121
Nombre entrant dans le stade :	50.9846	48.0223	34.4155	28.0665
Erreur standart :	4.4956	5.3332	5.5223	5.1105
Taux de survie journalier :	0.9737	0.9653	0.9637	0.9604
Erreur standart :	0.0043	0.0016	0.0045	0.0046
Appuyez sur une touche pour continuer ...				
Voulez vous l'impression sur imprimante (o/n) :				

Figure 15 : Résultats des calculs avec l'utilisation de la méthode du jackknife.

## BIBLIOGRAPHIE

- Ashford, J. R., Read, K. L. Q. & G. G. Vickers. 1970. A system of stochastic models applicable to studies of animal population dynamics. *J. Anim. Ecol.* 39 : 29-50.
- Dempster, J. P. 1956. The estimation of the number of individuals entering each stage during the development of one generation of an insect population. *J. Anim. Ecol.* 25 : 1-5.
- Dempster, J. P. 1961. The analysis of data obtained by regular sampling of an insect population. *J. Anim. Ecol.* 30 : 429-432.
- Kiritani, K. & F. Nakasuji. 1967. Estimation of the stage-specific survival rates in an insect population with overlapping stages. *Res. Popul. Ecol.* 9 : 143-152.
- Kobayashi, S. 1968. Estimation of the individual number entering each development stage in an insect population. *Res. Popul. Ecol.* X : 40-44.
- Lefkovitch, L.P. 1965. The study of population growth in organisms grouped by stages. *Biometrics* 21: 1-18.
- Manly, B. F. J. 1974. Estimation of the stage-specific survival rates and other parameters for insect populations developing through several stages. *Oecologia (Berl.)* 15 : 277-285.
- Manly, B. F. J. 1976. Extensions to Kiritani and Nakasuji's method for analysing insect stage-frequency data. *Res. Popul. Ecol.* 17 : 191-199.
- Manly, B. F. J. 1977. A further note on Kiritani and Nakasuji's model for stage-frequency data including comments on the use of Tukey's jackknife technique for estimating variances. *Res. Popul. Ecol.* 18 : 177-186.
- Quenouille, M. H. 1949. Approximate tests of correlation in time series. *J. Roy. Statist. Soc. B11* : 68-84.
- Quenouille, M. H. 1956. Notes on biases in estimation. *Biometrika* 43 : 353-360.
- Richards, O. W. & N. Waloff. 1954. Studies on the biology and population dynamics of British grasshoppers. *Anti-Locust Bull.* 17 : 1-182.
- Richards, O. W., Waloff, N. & J. P. Spradberry. 1960. The measurement of mortality in an insect population in which recruitment and mortality widely overlapp. *Oikos* 11 : 306-310.
- Tukey, J. W. 1958. Bias and confidence in not quite large samples (Abstract). *Ann. Math. Statist.* 29 : 614.

**Listing du programme**  
**(Quick-Basic)**

```

' programme kinaman1
' estimation de parametres de population se developpant par stades
' estimation des parametres dans le cas où tous les stades sont connus
' F.LARDEUX mai 1989
'----- programme principal -----
clear
cls
print
print tab(15) "-----"
print tab(15) " ANALYSE DE DONNEES STADES -- FREQUENCES "
print tab(15) "-----"
print : print : print
print "Source : MANLY B.J.F. - 1977 - A further note on Kiritani and Nakasuji's
print " model for stage-frequency data including
print " comments on the use of Tukey's jackknife
print " technique for estimating variances"
print
print " Res. Popul. Ecol. 18 : 177 - 186"
print : print
print "Programme F. LARDEUX - Centre OKSTOM de Tahiti - B.P. 529 Papeete"
print : print "version mai 1987"
print : print : print : print
input "Faites un 'Retour-chariot' pour continuer ....", r$

cls
open "i",#1,"kinaman1.par"
input #1,st,jo,gr
close #1

print "Les variables définissant la place mémoire nécessaire sont : "
print
print "Le nombre de stades dans la population étudiée"
print "Ce nombre est actuellement initialisé à : ";st
print
print "Le nombre total d'occasions d'échantillonnage."
print "Ce nombre est actuellement initialisé à : ";jo
print
print "Le nombre de groupe demandés pour le 'jackknife' "
print "Ce nombre est actuellement initialisé à : ";gr
print : print : print
print "Si ces valeurs sont trop grandes pour la taille mémoire de votre machine,
print "vous pouvez les modifier"
print : print
y$ = ""
while (y$ <>"o" and y$ <>"0" and y$ <>"n" and y$ <>"N")
input " Voulez vous les modifier ? (o/n) : ",y$
wend
print: print
if y$ = "o" or y$ = "0" then
input "Nombre de stades : ",st
input "Nombre d'occasions d'échantillonnage : ",jo
input "Nombre de groupes pour le jackknife : ", gr
print
k$ = ""
while (k$ <>"o" and k$ <>"0" and k$ <>"n" and k$ <>"N")
input "Voulez vous les stocker dans le fichier paramètres ? (o/n) : ",k$
wend
if k$ = "o" or k$ = "0" then
open "o",#1,"kinaman1.par"
print #1,st,jo,gr
close #1
end if
end if
end if

```

```

dim t(st,jo), f(st,jo)
dim a(gr,st), b(gr,st), c(gr,st)
dim nb(15)
dim jour(jo)
dim valeur(15,50)
dim z(jo), ndg(jo)
dim nba(gr,jo)
dim freq(st,jo)
dim aa(gr,st), w(gr,st), m(gr,st), mu(gr,st), teta(gr,st),d(gr,st)
dim moyw(st),moyaa(st),moyteta(st),moymu(st), moym(st), moyd(st)
dim varw(st),varaa(st),varteta(st),varmu(st), varm(st), vard(st)

arreter = 1
while arreter
  cls
  print : print : print
  print tab(35) "MENU GENERAL"
  print : print
  print tab(15) "Saisie des données ..... 1"
  print tab(15) "Correction dans un fichier ..... 2"
  print tab(15) "Listage d'un fichier ..... 3"
  print tab(15) "Calculs ..... 4"
  print
  print tab(15) "Fin de la session ..... 5"
  print : print : print : print
  while r < 1 or r > 5
    print tab(25) : input "Votre choix (numéro) : ",r
  wend
  if r = 1 then gosub saisie
  if r = 2 then gosub correc
  if r = 3 then gosub liste
  if r = 4 then gosub calcul
  if r = 5 then end
  r = 0
wend
end
'----- fin du programme principal -----

'----- fonction tirage d'un nombre aléatoire ----
def FNaleatoire
  randomize timer
  FNaleatoire = int(rnd * 100)
end def
'----- fin de la fonction aleatoire -----

'----- sous programme de saisie des données -----
saisie:
cls
print:print
input " Nombre de stades : ",stade
if stade <= 0 then return
for i = 1 to stade
  print
  print "stade n° ";i
  j = 1
  arret = 1
  while arret
    print tab(10);
    input; "Quantième : ", t(i,j)
    input " Nombre : ", f(i,j)
    if t(i,j) = 0 then arret = 0
    j = j + 1
  wend
  nb(i) = j - 2
next i
print : print
y$ = ""
while (y$ <>"o" and y$ <>"0" and y$ <>"n" and y$ <>"N")
  input "Voulez vous stocker ces données sur fichier (o/n) : ",y$
wend

```

```

if y$ = "N" or y$ = "n" then return
print
print "Nom du fichier des données (avec son chemin d'accès complet) : "
input "", fichier$
gosub ecriture
return
'----- fin du sous programme de saisie -----

'----- sous programme d'écriture sur fichier -----
ecriture:
open "o", #1, fichier$ + ".ki1"
print #1, stade
for i = 1 to stade
print #1, nb(i)
for j = 1 to nb(i)
print #1, t(i, j), f(i, j)
next j
next i
close #1

return
'----- fin du sous programme d'écriture -----

'----- sous programme de listage -----
liste:

close #2
open "scrn:" for output as #2
im = 0

call nomfich(fichier$) : '---- demande le nom du fichier à lister
gosub lecture : '----- lecture des données du fichier
gosub calcjour : '----- calcul le nombre de jours d'échantillonnage

for i = 1 to stade
for j = 1 to nb(i)
for k = 1 to njour
if t(i, j) = jour(k) then valeur(i, k) = f(i, j)
next k
next j
next i

impress:
debut = 1
fin = 15

arret = 1
while arret
if njour < fin then fin = njour : arret = 0

debutstade = 1
finstade = 7
arr = 1
while arr
cls
if finstade >= stade then finstade = stade : arr = 0
print #2, " jour ";
for i = debutstade to finstade
print #2, " stade"; i;
next i
print #2, "": print #2, ""
for j = debut to fin
print #2, using "#####"; jour(j);
if finstade >= stade then finstade = stade : arr = 0
for i = debutstade to finstade
print #2, using "#####"; valeur(i, j);
next i
print #2, ""
next j

```

```

print : print : print
if im = 0 then input "appuyez sur un touche pour continuer ..... ",a$
debutstade = finstade + 1
finstade = finstade + 7
wend

debut = fin + 1
fin = fin + 15

wend
close #2
print : print
y$ = ""
while (y$ <>"o" and y$ <>"O" and y$ <>"n" and y$ <>"N")
input "Voulez vous l'impression sur imprimante (o/n) : ",y$
wend
if y$ = "N" or y$ = "n" then im = 0 : return
open "lpt1:" for output as #2
im = 1
goto impress:
return
'----- fin du sous programme de listage -----

'----- sous programme de calcul des jours -----
calcjour:
for i = 1 to nb(1)
jour(i) = t(1,i)
next i

k = 0
for i = 2 to stade
for j = 1 to nb(i)
if t(i,j) > t(i-1,nb(i-1)) then
k = k+1
jour(nb(i)+k) = t(i,j)
end if
next j
next i
njour = nb(1) + k
return
'----- fin calcul du nbre de jours d'échantillonnage

'----- sous programme de correction -----
correc:
call nomfich (fichier$): '---- demande le nom du fichier à corriger
gosub lecture : '---- lecture des données sur le fichier

cls
for i = 1 to stade
print
print "stade : ";i
print
print "n° ligne      colonne 1      colonne 2"
print "              (quantième)   (nombre)"
print

debut = 1
fin = 15
if fin > nb(i) then fin = nb(i)

arret = 1
while arret
if fin = nb(i) then arret = 0
for j = debut to fin
print j,t(i,j),f(i,j)
next j
print : print
y$ = ""
while (y$ <>"o" and y$ <>"O" and y$ <>"N" and y$ <>"n")
input "Avez vous des données à corriger dans cet écran ? (o/n) : ",y$
wend

```

```
    if y$ ="o" or y$ ="0" then arret = 1 : gosub corige : fin = debut-1

    debut = fin + 1
    fin = debut + 15
    if fin > nb(i) then fin = nb(i)
    cls
wend
next i
gosub ecriture
return
'----- fin du sous programme de correction -----

'----- sous programme de demande du nom de fichier -----
sub nomfich (f$) static
cls
for i = 1 to 10 : print : next i
if f$ <> "" then
    print "Le fichier de travail actuel est : ";f$
    print
    y$ = ""
    while (y$ <>"0" and y$ <>"o" and y$ <>"N" and y$ <>"n")
        input "voulez vous le changer ? (o/n)  " : ",y$
    wend
    if y$ = "o" or y$ = "0" then f$ = "" : call nomfich(f$)
    else
        while f$ = ""
            print "Nom du fichier des données (avec son chemin d'accès complet) : "
            input "",f$
        wend
    end if
end sub
'----- fin du sous programme de nom de fichier -----

'----- sous programme de correction : corige -----
corige:
nl = 0
while nl > fin or nl < debut
    input "n° de ligne où se situe l'erreur  " : ",nl
wend

nc = 0
while (nc <1 or nc >2)
    input "n° de colonne où se situe l'erreur : ",nc
wend
print "valeur à corriger  " ;
if nc = 1 then
    print t(i,nl);
    input " nouvelle valeur : ",t(i,nl)
else
    print f(i,nl) ;
    input " nouvelle valeur : ",f(i,nl)
end if
return
'----- fin du sous programme corige -----

'----- sous programme de lecture du fichier -----
lecture:
open "i",#1,fichier$ + ".KI1"
input #1,stade
for i = 1 to stade
    input #1,nb(i)
    for j = 1 to nb(i)
        input #1,t(i,j), f(i,j)
    next j
next i
close #1
return
'----- fin du sous programme de lecture -----
```

```

'----- sous programme de calcul -----
calcul:
call nomfich(fichier%)
gosub lecture
gosub calcjour : 'calcule le nombre de jours d'échantillonnage (njour)
print:print
jk$ = ""
while (jk$ <>"0" and jk$ <>"o" and jk$ <>"n" and jk$ <>"N")
input "Voulez vous aussi les calculs par la méthode du jacknife (o/n) : ",jk$
wend
if jk$ = "o" or jk$ = "0" then
gosub jacknife
else ngroupe = 1
ndg(1) = njour
end if
gosub option1
return
'----- fin du sous programme de calcul -----

'----- jacknife -----
jacknife:
'----- tirage des goupes pour le jacknife -----
a$ = ""
while (a$ <>"o" and a$ <>"0")
'--- nombre de groupes possibles
ngroupe = 0
while (ngroupe <> njour and ngroupe > int(njour/2)) or ngroupe < 2
cls
print : print : print tab(20) "Calculs par la méthode du jacknife"
print : print : print : print
print "Combien de groupes voulez vous faire ? "
print : print "Ce nombre est compris entre 2 et ";int(njour/2);
print "ou bien vaut : ";njour
print "Ma suggestion est ";njour; " (mais les calculs sont plus longs)"
print : print
print tab(20) :input "Votre choix : ",ngroupe
wend

'--- calcul du nombre d'éléments dans chaque groupe (ndg)
for i = 1 to ngroupe-1
ndg(i) = int(njour/ngroupe)
next i
ndg(ngroupe) = njour - (ngroupe-1) * int(njour/ngroupe)
'--- fin de ndg ---

print
print "Nombre d'éléments dans les ";
print using "##";ngroupe - 1;
print " premiers groupes : ";ndg(1)
print "Nombre d'éléments dans le dernier groupe : ";ndg(ngroupe)
a$ = ""
while (a$ <>"o" and a$ <>"0" and a$ <>"n" and a$ <>"N")
print : input "D'accord pour cette partition ? (o/n) : ",a$
wend
wend

```

```

'----- allocation des jours d'échantillonnage aux groupes -----
k = 1
for i = 1 to ngroupe
  for j = 1 to ndg(i)
    alea:
      z(k) = 0
      while z(k) > njour or z(k) < 1
        z(k) = FNaleatoire
      wend
      for l = 1 to k-1
        if z(k) = z(l) then goto alea
      next l
      nba(i,j) = z(k)
      k = k + 1
    next j
  next i
'----- fin de l'allocation -----
return
'----- fin de jackknife -----

'----- option 1 -----
'--- calculs lorsque tous les stades sont connus

option1:

gosub zero : '---mise à zero des tableaux a et b
'calcul pour les sous échantillons
for r = 0 to ngroupe
gosub transfo
  for j = 1 to njour
    for m = 1 to ndg(r)
      if j = nba(r,m) then
        for i = 1 to stade
          freq(i,j) = 0
        next i
      end if
    next m
  next j

  nstade = 1
  for i = nstade to stade
    for k = nstade to stade
      for j = 1 to njour
        a(r,i) = a(r,i) + (jour(j+1) - jour(j-1)) * freq(k,j) / 2
      next j
    next k

    for k = nstade to stade
      for j = 1 to njour
        b(r,i) = b(r,i) + (jour(j+1) - jour(j-1)) * jour(j) * freq(k,j) / a(r,i) / 2
      next j
    next k

    nstade = nstade + 1
  next i
next r

' calcul des valeurs sans jackknife et pour chaque sous échantillon
for r = 0 to ngroupe
  for i = 1 to stade - 1
    w(r,i) = a(r,i+1) / a(r,i)
    aa(r,i) = b(r,i+1) - b(r,i)
    teta(r,i) = - LOG(w(r,i)) / aa(r,i)
    mu(r,i) = b(r,i) - (1 / teta(r,i))
    m(r,i) = a(r,i) * teta(r,i)
    d(r,i) = EXP(-teta(r,i))
  next i
next r

```

```

' calcul des pseudo-valeurs pour chaque sous échantillon
for r = 1 to ngroupe
  for i = 1 to stade - 1
    w(r,i) = ngroupe * w(0,i) - (ngroupe - 1) * w(r,i)
    aa(r,i) = ngroupe * aa(0,i) - (ngroupe - 1) * aa(r,i)
    teta(r,i) = ngroupe * teta(0,i) - (ngroupe - 1) * teta(r,i)
    mu(r,i) = ngroupe * mu(0,i) - (ngroupe - 1) * mu(r,i)
    m(r,i) = ngroupe * m(0,i) - (ngroupe - 1) * m(r,i)
    d(r,i) = ngroupe * d(0,i) - (ngroupe - 1) * d(r,i)
  next i
next r

' calcul des moyennes des pseudo - valeurs
for i = 1 to stade - 1
  for r = 1 to ngroupe
    moyw(i) = moyw(i) + w(r,i)
    moyaa(i) = moyaa(i) + aa(r,i)
    moyteta(i) = moyteta(i) + teta(r,i)
    moymu(i) = moymu(i) + mu(r,i)
    moym(i) = moym(i) + m(r,i)
    moyd(i) = moyd(i) + d(r,i)
  next r

  moyw(i) = moyw(i) / ngroupe
  moyaa(i) = moyaa(i) / ngroupe
  moyteta(i) = moyteta(i) / ngroupe
  moymu(i) = moymu(i) / ngroupe
  moym(i) = moym(i) / ngroupe
  moyd(i) = moyd(i) / ngroupe

next i

' variances
deno = (ngroupe - 1)
if ngroupe = 1 then goto ecriture1 : 'on ne fait pas de jackknife et on ne calcule par de variance
for i = 1 to stade -1
  for r = 1 to ngroupe
    varw(i) = varw(i) + (w(r,i) - moyw(i))^2
    varaa(i) = varaa(i) + (aa(r,i) - moyaa(i))^2
    varteta(i) = varteta(i) + (teta(r,i) - moyteta(i))^2
    varmu(i) = varmu(i) + (mu(r,i) - moymu(i))^2
    varm(i) = varm(i) + (m(r,i) - moym(i))^2
    vard(i) = vard(i) + (d(r,i) - moyd(i))^2
  next r
  varw(i) = varw(i) / deno
  varaa(i) = varaa(i) / deno
  varteta(i) = varteta(i) / deno
  varmu(i) = varmu(i) / deno
  varm(i) = varm(i) / deno
  vard(i) = vard(i) / deno

next i

'----- ecriture des résultats -----
ecriture1:
close #2
open "scrn:" for output as #2
im = 0

imprim:
debut = 1
fin = debut + 3

```

```

arret = 1
while arret
  if fin >= stade - 1 then fin = stade - 1 : arret = 0

  cls
  print #2, ""
  print #2, tab(20) "RESULTATS";
  print #2, " pour le fichier "; fichier$
  print #2, ""
  print #2, " ";
  for i = debut to fin
    print #2, "   stade";i;" ";
  next i
  print #2, "" : print #2, ""
  print #2, "Taux de survie du stade      : ";
  for i = debut to fin
    print #2, using "#####.###"; moyw(i);
  next i
  print #2, ""
  print #2, "Erreur standart                : ";
  if ngroupe = 1 then
    print #2, "          Non calculée car pas de jacknife "
  else
    for i = debut to fin
      print #2, using "#####.###"; sqr(varw(i)/ngroupe);
    next i
    print #2, ""
  end if
  print #2, ""
  print #2, "Durée                          : ";
  for i = debut to fin
    print #2, using "#####.###"; moyaa(i);
  next i
  print #2, ""
  print #2, "Erreur standart                : ";
  if ngroupe = 1 then
    print #2, "          Non calculée car pas de jacknife"
  else
    for i = debut to fin
      print #2, using "#####.###"; sqr(varaa(i)/ngroupe);
    next i
    print #2, ""
  end if
  print #2, ""
  print #2, "Paramètre de survie                : ";
  for i = debut to fin
    print #2, using "#####.###"; moyteta(i);
  next i
  print #2, ""
  print #2, "Erreur standart                : ";
  if ngroupe = 1 then
    print #2, "          Non calculée car pas de jacknife "
  else
    for i = debut to fin
      print #2, using "#####.###"; sqr(varteta(i)/ngroupe);
    next i
    print #2, ""
  end if
  print #2, ""
  print #2, "Date moyenne d'entrée              : ";
  for i = debut to fin
    print #2, using "#####.###"; moymu(i);
  next i
  print #2, ""
  print #2, "Erreur standart                : ";
  if ngroupe = 1 then
    print #2, "          Non calculée car pas de jacknife"
  else
    for i = debut to fin
      print #2, using "#####.###"; sqr(varmu(i)/ngroupe);
    next i
    print #2, ""
  end if
end if

```

```

print #2, ""
print #2, "Nombre entrant dans le stade : ";
for i = debut to fin
  print #2, using "#####.###"; moym(i);
next i
print #2, ""
print #2, "Erreur standart          : ";
if ngroupe = 1 then
  print #2, "          Non calculée car pas de jacknife"
else
  for i = debut to fin
    print #2, using "#####.###"; sqr(varm(i)/ngroupe);
  next i
  print #2, ""
end if
print #2, ""
print #2, "Taux de survie journalier   : ";
for i = debut to fin
  print #2, using "#####.###"; moyd(i);
next i
print #2, ""
print #2, "Erreur standart          : ";
if ngroupe = 1 then
  print #2, "          Non calculée car pas de jacknife "
else
  for i = debut to fin
    print #2, using "#####.###"; sqr(var(i) / ngroupe);
  next i
end if

debut = fin + 1
fin = debut + 3
print #2, ""
if im = 0 then input "Appuyez sur une touche pour continuer ... ", k$

wend

close #2
print
y$ = ""
while (y$ <> "o" and y$ <> "0" and y$ <> "n" and y$ <> "N")
  input "Voulez vous l'impression sur imprimante (o/n) : ", y$
wend
if y$ = "N" or y$ = "n" then im = 0 : return
open "lpt1:" for output as #2
im = 1
goto imprim:

return

'----- fin de l'option 1 -----

'----- transformation des données en tableau pour le calcul-----
transfo:
for i = 1 to nb(1)
  freq(1,i) = f(1,i)
next i

for i = 2 to stade
  for j = 1 to nb(i)
    for k = 1 to njour
      if t(i,j) = jour(k) then freq(i,k) = f(i,j)
    next k
  next j
next i
return
'----- fin de la transformation en tableau -----

```

```
'----- mise à zero des tableaux a et b -----  
zero:  
for r = 0 to ngroupe  
  for i = 1 to stade  
    a(r,i) = 0  
    b(r,i) = 0  
    w(r,i) = 0  
    aa(r,i) = 0  
    teta(r,i) = 0  
    mu(r,i) = 0  
    m(r,i) = 0  
    d(r,i) = 0  
  next i  
next r  
for i = 1 to stade  
  moyw(i) = 0  
  moyaa(i) = 0  
  moyteta(i) = 0  
  moymu(i) = 0  
  moym(i) = 0  
  moyd(i) = 0  
  varw(i) = 0  
  varaa(i) = 0  
  varteta(i) = 0  
  varmu(i) = 0  
  varm(i) = 0  
  vard(i) = 0  
next i  
return  
'----- fin de mise à zero des tableaux a et b -----
```