

Rapport de stage

Vers la création d'un Package R pour l'estimation
du temps de séjour d'un individu dans un stade de
développement donné en présence d'une censure
post-évènement d'intérêt

Rédigé par: Sinclair Tsana
Sous l'encadrement de: Dr. Patrice Takam Soh

Stage de 3e Année

Septembre 2021

Contents

1	Introduction	7
2	Contexte	9
2.1	Cadre de travail	9
2.2	Objectifs du projet	9
3	Prérequis	11
3.1	Programmation R	11
3.1.1	Présentation du logiciel	11
3.1.2	Les bases du logiciel	11
3.1.3	Procédure de Création d'un package	12
3.2	Calcul Intégral avec le logiciel R	13
3.2.1	Intégrales simples	13
3.2.2	Intégrales doubles	13
4	Description des programmes	15
4.1	Importation des données	15
4.2	Fonction qui donne les numéros des individus par stade	15
4.3	Statistique du nombre d'individus par stade	16
4.4	Durée maximale discrète par stade	16
4.5	Vecteur des poids	16
4.6	Vecteurs des covariables	17
4.7	Fonction de Hazard Cumulée	18
4.8	Fonction de Survie	18
4.9	Fonction densité de la durée de vie au stade k	19
4.10	Statut sanitaire	19
4.11	Statut de tous les individus dans le stade k	20
5	Estimation des paramètres	21
5.1	Estimation des paramètres au stade 1	21
5.1.1	Calcul de la probabilité d'être observé sain au stade suivant . . .	21
5.1.2	Calcul de la probabilité d'être observé infecté au stade suivant . .	21
5.1.3	Calcul de la probabilité d'être observé infecté au dernier jour d'observation(avant la fin de l'étude)	22
5.1.4	Calcul de la probabilité d'être observé sain au dernier jour d'observation de l'étude	22

Contents

5.1.5	Calcul de la probabilité d'être observé attaqué au dernier jour d'observation de l'étude	22
5.1.6	Calcul de la vraisemblance au stade 1	23
6	Conclusion	25

Remerciements

Mes remerciements s'adressent en premier lieu à Monsieur le Directeur Général de l'ESSFAR, le Docteur **Patrick Seumen Tonou**, ainsi que le Directeur des Affaires Académiques, le Professeur **Etienne Tsamo** qui ont donné leurs accords pour tout mis en œuvre pour le bon déroulement de ce projet.

Je tiens à remercier mon maître de stage **M. Patrice Takam Soh** pour son accompagnement et ses enseignements; ainsi que tout le corps enseignant et administratif de **l'ESSFAR** pour leur encadrement.

Je remercie maintenant tous ceux ayant contribué de près ou de loin à ce projet.

1 Introduction

Le présent rapport de stage à été effectuée dans le cadre de ma 3e année de formation en Mathématique et Informatique Appliquées a l'Économie à l'École Supérieure des Sciences de la Finance, de l'Assurance et des Risques(**ESFFAR**). Ce projet vise l'application des notions mathématiques vue en cours aux problèmes du monde réel et l'approfondissement de mes compétences en programmation R tout en contribuant à l'amélioration de cet outil fabuleux qu'est le logiciel R.

En effet, logiciel R occupe une place de choix dans le domaine de la recherche en général et celui de la Science des données en particulier. Il est à la fois un langage de programmation et un logiciel destinés aux professionnels de la donnée. Dans le cadre de ma formation à l'ESSFAR la maîtrise de cet outil est très importante non seulement pour les simulations pratiques mais aussi pour l'insertion professionnelle.

R est un logiciel libre de droit qui propose des nombreuses fonctionnalités. Il est néanmoins possible d'installer des suppléments, appelés packages, afin de les étendre. Le but de ce projet est donc de créer un de ces packages pour l'estimation du temps de séjour d'un individu dans un stade de développement donné.

2 Contexte

2.1 Cadre de travail

Le stage s'est déroulé au sein du campus de l'ESSFAR dans la salle de cours des Licence 3 entre la période allant du **5 Juillet au 27 Septembre 2021**. Nous étions trois étudiants, chacun travaillant individuellement sur son projet. Nous n'hésitions pas à solliciter l'aide d'un camarade pour réfléchir ensemble en cas de blocage. Nous avions des rencontres avec l'enseignant deux ou trois fois par semaine pour faire le point et exposer nos difficultés à travers des présentations vidéo-projetés. Une connexion internet a été mise à notre disposition pour faciliter nos recherches.

2.2 Objectifs du projet

Le Projet vise:

- L'application des méthodes mathématiques vues en cours aux problèmes du monde réel.
- L'amélioration des compétences en programmation R
- L'initiation à la recherche reproductible en proposant un package R à la communauté scientifique.

Pour atteindre ces objectifs, nous avons organisé notre travail de la manière suivante: Le Chapitre 1 est une introduction générale du projet. Le chapitre 2 présente le contexte, le cadre de travail et les objectifs visés par le projet. Le chapitre 3 est une brève présentation du logiciel R et des outils de base nécessaires à la bonne marche du projet. Le chapitre 4 est une description détaillée du travail effectué notamment des programmes écrits pour la réalisation du package. Le chapitre 5 présente les différents algorithmes d'estimation des paramètres par la méthode du maximum de vraisemblance. Le chapitre 6 est une conclusion générale du travail réalisé.

3 Prérequis

3.1 Programmation R

3.1.1 Présentation du logiciel

Le logiciel **R** ¹ un outil très puissant de statistique crée par **Ross Ihaka** et **Robert Gentleman**. Il est à la fois un langage de programmation et un environnement de travail. Il fonctionne sous Linux, Windows et Mac. C'est aussi un logiciel gratuit et à code source ouvert (open-source). Tout le monde peut d'ailleurs contribuer à son amélioration en y intégrant de nouvelles fonctionnalités ou méthodes d'analyse non encore implémentées. C'est justement dans une logique de contribution que s'inscrit ce projet.

L'interface de développement **RStudio** est généralement associé à R pour augmenter la productivité des programmeurs. C'est un environnement de développement gratuit, libre et multi-plateforme pour R.

3.1.2 Les bases du logiciel

R est un langage interprété et non compilé, c'est-à-dire que les commandes tapées au clavier sont directement exécutées sans qu'il soit besoin de construire un programme complet. Ensuite, la syntaxe de R est très simple et intuitive. Voici par exemple quelques opérations basiques.

```
> x <- 10 # Affectation
> x
[1] 10
> vec <- 2:16 # Vecteur d'entiers entre 2 et 16
> vec
[1] 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
> length(x) # longueur du vecteur x
[1] 15
> seq(from=0, to=1, by=0.1)
> A <- matrix(1:9, nrow=3, ncol=3, byrow=TRUE)
> A
[,1] [,2] [,3]
[1,] 1 2 3
[2,] 4 5 6
[3,] 7 8 9
```

¹<http://cran.r-project.org/>

```
> maListe <- list(TRUE, 1:5, c(1+2i, 3), "chaîne de caractères")
> maListe
[[1]]
[1] TRUE

[[2]]
[1] 1 2 3 4 5

[[3]]
[1] 1+2i 3+0i

[[4]]
[1] "chaîne de caractères"
> maListe[3] # retrouver le 3e element de maListe
> install.packages("stats") # installer le package stats
> library(dplyr) # charger le package dplyr
```

3.1.3 Procédure de Création d'un package

Un package R est constitué d'un ensemble de fichiers et de répertoires dont la nomenclature est normalisée selon les directives publiées par le **CRAN**. Le répertoire racine doit porter le nom du package. Voici l'arborescence d'un package R:

- un répertoire **/R/**, qui contient les scripts R
- un répertoire **/man/**, qui contient les fichiers d'aide (chaque fonction publique doit être référencée dans un fichier d'aide).
- un répertoire **/data/**, qui contient les jeux de données
- un répertoire **/tests/**, qui contient des tests bloquants qui arrêtent la compilation du package en cas d'erreur lors de leur exécution.
- un répertoire **/demo/**, qui contient des démonstrations exécutées automatiquement lorsque l'utilisateur lance la commande `demo()`.
- un répertoire **/inst/doc/**, qui contient des documentations utilisateurs, aussi appelées vignettes : ces documentations ont été réalisées avec **Markdown** et compilé avec le package **knitr**
- un fichier **NAMESPACE**, qui précise la visibilité des fonctions,
- un fichier **DESCRIPTION**, qui décrit les différentes fonctions et leur utilisation
- un fichier **NEWS**, qui retrace l'historique des versions du package

- un fichier **TODO**, qui répertorie les fonctionnalités à implémenter ou les problèmes à corriger

3.2 Calcul Intégral avec le logiciel R

L'idée fondamentale de l'intégration numérique est d'approcher $I = \int_a^b f(x)dx$ dans le cas où on ne connaît pas une primitive de f . Il y a deux façons de présenter le calcul:

- On approche I par une quadrature globale, c'est à dire qu'on remplace f par un polynôme d'interpolation.
- On approche f par des polynômes par morceaux. Encore appelée méthode composite, cette méthode est la plus utilisée notamment dans les logiciels comme **Matlab**. Le logiciel R propose plusieurs packages et fonctions pour calculer ou estimer la valeur d'une intégrale simple dans un domaine fini ou infini. Le package **stats** propose par exemple la commande **integrate()**.

3.2.1 Intégrales simples

Dans le cadre de ce travail, nous utiliserons les méthodes du triangle et des trapèzes pour approcher nos intégrales. Le programme s'écrit de la manière suivante:

```
integ1= function(f,a,b, ...,nrep=10){
  if(a>b) print('a doit etre inferieure a b')
  if(a == b) return(0)
  if(a<b){
    h = (b-a)/nrep; sub = seq(a,b,by=h); inter =
      sub[-c(1,length(sub))]
    return( h*(sum(f(inter, ...)) + (1/2)*(f(sub
      [1], ...) + f(sub[length(sub)], ...))))
  }
}
```

3.2.2 Intégrales doubles

Pour calculer une intégrale double, nous utilisons l'approximation d'une intégrale dans un triangle.

```
integrale.double.triangle = function(f,somet1,somet2,somet3
  ,...){
  g1=0.4459; g2=0.0916; w1=0.22338; w2=0.10995
  finter = function(g1,g2,g3){
    x = somet1[1]*g1 + somet2[1]*g2 + somet3[1]*g3
    y = somet1[2]*g1 + somet2[2]*g2 + somet3[2]*g3
```

3 Prérequis

```
res = f(x,y,...)
res}
res1 = w1*( finter(g1,g1,1-2*g1)+finter(g1,1-2*g1,g1)+finter
(1-2*g1,g1,g1) )
res2 = w2*( finter(g2,g2,1-2*g2)+finter(g2,1-2*g2,g2)+finter
(1-2*g2,g2,g2) )
aire.triangle = (somet2[1]-somet1[1])*(somet3[2]-somet1[2])/
2
resultat = aire.triangle*(res1+res2)
return(resultat)
}
```

R propose le package **cubature** et les fonctions **hcubature()** et **pcubature()** Ainsi que les packages **rmutil** et **pracma** pour approcher une intégrale. Cependant, pour de raison de vitesse, nous utiliserons l'algorithme du triangle suivant.

```
integ2 = function(f,lower,upper,...){
somet1 = c(lower[1],lower[2]); somet2 = c(upper[1],lower[2])
;
somet3 = c(upper[1],upper[2]);somet4 = c(lower[1],upper[2])
res1 = integrale.double.triangle(f,somet1,somet2,somet4
,...)
res2 = integrale.double.triangle(f,somet3,somet4,somet2
,...)
res = res1 + res2
return(res)
}
```

4 Description des programmes

4.1 Importation des données

159 cherelles ont été tirées au hasard sur 15 arbres dans une parcelle paysanne de Mbankomo (à Yaoundé, Cameroun) avec récolte sanitaire toutes les semaines, en 1999. Ces fruits ont été choisis sur l'ensemble des niveaux des arbres (premier niveau de 0m à 0.5m, deuxième niveau de 0.5m à 1.5m, troisième niveau > 1.5m). On a noté chaque lundi matin, durant 20 semaines à partir du 26 avril 1999, le stade de développement de chacun de ces 159 fruits (1 = chérelle, 2 = jeune cabosse, 3 = cabosse adulte, 4 = cabosse mûre saine), ainsi que leur état sanitaire (0 = fruit sain, 1 = fruit présentant au moins un symptôme de la pourriture brune, 2 = fruit présentant au moins un symptôme de wilt, 3 = fruit rongé). Toutes ces informations ont été regroupées en 03 jeux de données (resp. data1=Etats sanitaire, data2=stade de développement et data3=données climatiques) qui nous ont permis de construire et de tester nos différents modèles. le fichier

```
data1 <- read.table(dataEtats, header = TRUE, sep = "")
data2 <- read.table(DataStadeDevelopement, header = TRUE,
  sep = "")
data3 <- read.table(Donnee_Climat_V1, header = TRUE, sep = "
")
data.e <- data.frame(data1)[,-1] # Suppression de la
  premiÃ¨re colonne
data.s <- data.frame(data2)[,-1]
data.cl <- data.frame(data3)[,-1]
plv <- data.cl[,4] # pluviomÃ©trie
Ta <- (data.cl[,5]+data.cl[,6])/2 # tempÃ©rature ambiante
Tc <- (data.cl[,7]+data.cl[,8])/2 # tempÃ©rature sous
  cacaoyÃ¨re
plv[plv==0]=0.01
cl = data.frame(plv,Ta,Tc)
```

4.2 Fonction qui donne les numéros des individus par stade

Avant toute chose, nous devons identifier nos individus selon leurs caractéristiques dans chaque stade à travers leurs indices.

```
Ik = function(k, datas){  
  res=c()  
  for(i in 1:nrow(datas)){  
    y = datas[i,]  
    if((sum(y==k)!=0) & (sum(y==k)!=1)) res[i]=i  
    else res[i]=0  
  }  
  res  
}
```

4.3 Statistique du nombre d'individus par stade

Cette fonction permet de déterminer les indices de tous les individus ayant effectivement atteint chaque stade.

```
nb1 = length(Ik(1,data.s)[Ik(1,data.s) !=0])  
nb2 = length(Ik(2,data.s)[Ik(2,data.s) !=0])  
nb3 = length(Ik(3,data.s)[Ik(3,data.s) !=0])
```

4.4 Durée maximale discrète par stade

La Durée maximale discrète par stade permet d'estimer le temps de séjour de chaque individu dans un stade k.

```
duree.disc = function(k,datas,L=7){  
  ik = Ik(k, datas); ik1=ik[ik!=0];res=c()  
  for(i in 1:length(ik1)){  
    y=datas[ik1[i],]  
    res[i] = (sum(y==k))*L + 1  
  }  
  res  
}
```

4.5 Vecteur des poids

Afin de rendre le concept plus réaliste nous avons défini un vecteur `w0_k` qui détermine le poids de chaque facteur. `k` étant le stade courant, **datas** le jeu de données sur le statut sanitaire de l'ensemble des individus. Cette fonction renvoi un vecteur comportant les poids.


```
w0k<- function(datas){
  w01 = 1/max(duree.disc(1,datas))
  w02 = 1/max(duree.disc(2,datas))
  w03 = 1/max(duree.disc(3,datas))
  res = c(w01,w02,w03)
  res
}
```

4.6 Vecteurs des covariables

Considérons un individu qui entre au stade k à une date donnée u , connaissant le vecteurs des variables qui agissent sur cet individu dans ce stade jusqu'à une date t ($t > u$), nous pouvons déterminer l'influence de chacun de ces facteurs à travers la fonction **x.tild** dont le code R est le suivant:

```
x_tild <- function(data.cl.cur,datas=data.s,u,v,k){#data.cl.
  cur=Ta,Tc ou plv
u <- ceiling(u)
v <- floor(v)
n = v-u+1
if(u>v){stop("not defined for u>v")}
if(v==u)
  return(0)
else
  res <- sum(data.cl.cur[1:(n-1)]) + (1/2)*(data.cl.
    cur[u]+data.cl.cur[v])
return(res*w0k(datas)[k])
}
```

Description

La fonction **x.tild()** détermine la part de chaque facteur qui influence la croissance d'un individu i dans un stade donné k . La fonction **x_tild** renvoie une valeur entière qui estime l'influence du climat courant sur le développement de l'individu.

Arguments

data.cl.cur	climat courant(plv, Ta ou Tc)
u	Date d'entrée au stade
v	Dernier jour au stade
tp	dernier jour de l'étude
k	stade courant

4.7 Fonction de Hazard Cumulée

Nous pouvons donc être intéressé par l'estimation de l'incidence cumulé de chacun des facteurs dans le stade k . Pour se faire, nous avons défini la fonction Cumulative Hazard (H_k).

```
alpha.1 = rep(1,3) #Initialisation des paramètres
mu.1 = rep(1,3)

Hk <- function(v,u,climat=cl,datas=data.s,alpha=alpha.1,mu=
mu.1,k=1){
  if(v<u){
    stop("not defined for v<u")}
  else if(v==u)
    return(0)
  else{
    plv = climat[,1];Ta=climat[,2];Tc=climat[,3]
    v.cum_clim <- c(x_tild(plv,datas,u,v,k),
    x_tild(Ta,datas,u,v,k),
    x_tild(Tc,datas,u,v,k))
    som <- sum(alpha* (v.cum_clim^mu))
    return(som)
  }
}
```

4.8 Fonction de Survie

Lorsqu'on se place dans un stade de développement donné k , l'évènement d'intérêt est le passage au stade suivant $k+1$. Nous pouvons estimer la probabilité que l'individu reste sain a la fin du stade k a l'aide de la fonction $S_k()$ définie comme suit:

```
Sk <- function(v,u,climat=cl,datas=data.s,alpha=alpha.1,mu=
mu.1,k=1){
  if(v>=u){
    return(exp(-Hk(v,u,climat,datas,alpha,mu,k))
  )
  }else{
    stop("Verifiez que v>=u")
  }
}
```

4.9 Fonction densité de la durée de vie au stade k

Ici, nous voulons estimer la loi d'une variable aléatoire T_k qui désigne le temps passé par un individu sain au stade k. La variable aléatoire T_k est donc non-négative et nous supposons qu'elle est indépendamment et identiquement distribuée de densité $\psi\theta_k$.

```
phi_k <- function(v,u,climat=cl,datas=data.s,alpha=alpha.1,
  mu=mu.1,k=1){
  Ta.tild <- x_tild(Ta,datas,u,v,k)
  Tc.tild <- x_tild(Tc,datas,u,v,k)
  plv.tild <- x_tild(plv,datas,u,v,k)
  v.cum_clim <- c(plv.tild,Ta.tild,Tc.tild)
  v.inter <- c(plv[v],Ta[v], Tc[v])
  betha <- alpha*mu*v.inter
  s <- sum(betha*v.cum_clim^(mu-1))
  return(Sk(v,u,climat,datas,alpha,mu,k)*s)
}
```

4.10 Statut sanitaire

Pour un individu i ayant été observé au moins une fois au stade k, on distingue 4 possibilités:

- Soit l'individu i reste sain jusqu'au dernier jour d'observation
- Soit il est sain jusqu'au premier jour au stade k+1
- Soit il est attaqué au stade k
- Soit il est attaqué pendant la transition du stade k au stade k+1.

```
statut <- function(i,k=1,tp=20,datas=data.s, datae=data.e){
  ik = Ik(k,datas)
  if(ik[i]==0){stop("L'individu n'est pas dans ce
    stade")}
  else{
    y = datas[i,]
    eik <- min((1:length(y))[y==k]) # Premier
      jour de i au stade k
    eikp1 <- max((1:length(y))[y==k]) # Dernier
      jour de i au stade k
    if(eikp1<tp){
      #Cas ou i a changÃ© de stade
    }
  }
}
```

```

        if((datae[i,eikp1]==0) & (datas[i,
            eikp1] != datas[i,eikp1+1]) & (
            datae[i,eikp1+1]==0))
        res=1
        if((datae[i,eikp1]==0) & (datas[i,
            eikp1] != datas[i,eikp1+1]) &(
            datae[i,eikp1+1]!=0))
        res=2
        # Cas ou i n'a pas changÃ© de stade
        if(datae[i,eikp1]!=0)
        res=3
    }
    if(eikp1==tp){
        #Ici l'individu n'a pas pu changÃ©
        car il est observÃ© au stade k Ã
        tp
        # Cas ou i est observÃ© sain a tp
        if(datae[i,eikp1]==0) res=4
        # Cas ou i est observÃ© attaquÃ© Ã
        tp
        if(datae[i,eikp1]!=0) res=5
    }
}
res
}

```

4.11 Statut de tous les individus dans le stade k

Ici nous dÃ©finissons une fonction qui prends en entrÃ©e le numÃ©ro d'un stade et donne le statut de tous les individus dans le stade en question en vue de les classer et les diffÃ©rencier dans l'Ã©tude.

```

statut.data.k = function(k=1,tp=20,datas=data.s, datae=data.
e){
    res=c()
    ik = Ik(k,datas);ik=ik[ik!=0]
    for(i in ik)
        res[i] = statut(i,k,tp,datas, datae)
    res
}

```

5 Estimation des paramètres

5.1 Estimation des paramètres au stade 1

5.1.1 Calcul de la probabilité d'être observé sain au stade suivant

```
proba1 = function(i,climat=cl,datas=data.s,alpha=alpha.1,mu=
  mu.1,k=1,L=7,w0k=w0.1){
y = datas[i,]
eik <- min((1:length(y))[y==k])
eikp1 <- max((1:length(y))[y==k])
eik.c = eik*L + 1; eikp1.c = eikp1*L + 1;v0.1=eikp1.c-L;v0
  .2=eikp1.c
f = function(u) return( Sk(u,v=v0.1,climat=climat,alpha=
  alpha,mu=mu,k=k,w0k=w0k) )
g = function(u) return( Sk(u,v=v0.2,climat=climat,alpha=
  alpha,mu=mu,k=k,w0k=w0k) )
f = Vectorize(f); g= Vectorize(g)
res1 = integ1(f,eik.c-L,eik.c); res2 = integ1(g,eik.c-L,eik
  .c)
resf = (1/L)*(res1-res2)
resf
```

5.1.2 Calcul de la probabilité d'être observé infecté au stade suivant

```
proba2 = function(i,climat=cl,datas=data.s,alpha=alpha.1,mu=
  mu.1,k=1,L=7,w0k=w0.1){
tas[i,]
  min((1:length(y))[y==k])
<- max((1:length(y))[y==k])
= eik*L + 1; eikp1.c = eikp1*L + 1;v0.1=eikp1.c-L;v0.2=eikp1
  .c
nction(u) return( Sk(u,v=v0.1,climat=climat,alpha=alpha,mu=
  mu,k=k,w0k=w0k) ); f = Vectorize(f)
1/L)*integ1(f,eik.c-L,eik.c)
(1/L^2)*integ2(Sk,c(eik.c-L,eikp1.c-L),c(eik.c,eikp1.c),
  climat,alpha,mu,k,w0k)
```

```
res1-res2; res
```

5.1.3 Calcul de la probabilité d'être observé infecté au dernier jour d'observation(avant la fin de l'étude)

```
proba3 = function(i,climat=cl,datas=data.s,alpha=alpha.1,mu=
  mu.1,k=1,L=7,w0k=w0.1){
  y = datas[i,]
  eik <- min((1:length(y))[y==k])
  eikp1 <- max((1:length(y))[y==k])
  eik.c = eik*L + 1; eikp1.c = eikp1*L + 1;
  res = (1/L^2)*integ2(Sk,c(eik.c-L,eikp1.c-L),c(eik.c,eikp1
    .c),climat,alpha,mu,k,w0k)
  res
}
```

5.1.4 Calcul de la probabilité d'être observé sain au dernier jour d'observation de l'étude

```
proba4 = function(i,climat=cl,datas=data.s,alpha=alpha.1,mu=
  mu.1,k=1,L=7,tp=20,w0k=w0.1){
  y = datas[i,]
  eik <- min((1:length(y))[y==k])
  eikp1 <- tp
  eik.c = eik*L + 1; eikp1.c = eikp1*L + 1; v0.2=eikp1.c
  f = function(u) return( Sk(u,v=v0.2,climat=climat,alpha=
    alpha,mu=mu,k=k,w0k=w0k) ); f = Vectorize(f)
  res=(1/L)*integ1(f,eik.c-L,eik.c)
  res
}
```

5.1.5 Calcul de la probabilité d'être observé attaqué au dernier jour d'observation de l'étude

```
proba5 = function(i,climat=cl,datas=data.s,alpha=alpha.1,mu=
  mu.1,k=1,L=7,tp=20,w0k=w0.1){
  tas[i,]
  min((1:length(y))[y==k])
  <- max((1:length(y))[y==k])
```

```
= eik*L + 1; eikp1.c = eikp1*L + 1; v0.1=eikp1.c-L; v0.2=eikp1
.c
integ2(Sk, c(tp*L+1-L, eik.c-L), c(eik.c, eikp1.c), climat, alpha,
mu, k, w0k)
```

5.1.6 Calcul de la vraisemblance au stade 1

On rappelle qu'on ne considérera que les individus ayant été observé au moins deux fois dans le stade.

```
statut.stade1 = statut.data.k(1)
Lvrais1 = function(coefs, statut=statut.stade1, climat=cl,
  datas=data.s, k=1, L=7, tp=20, w0k=w0.1){
  alpha = coefs[1:3]; alpha = exp(alpha/100)
  mu = coefs[4:6]; mu=exp(-mu/100)
  ik=Ik(k, data.s); ik1=ik[ik!=0]; res=c()
  for(i in ik1){
    if(statut[i]==1) res[i]=proba1(i, climat, datas, alpha, mu, k
      ,L, w0k)
    if(statut[i]==2) res[i]=proba2(i, climat, datas, alpha, mu, k
      ,L, w0k)
    if(statut[i]==3) res[i]=proba3(i, climat, datas, alpha, mu, k
      ,L, w0k)
    if(statut[i]==4) res[i]=proba4(i, climat, datas, alpha, mu, k
      ,L, tp, w0k)
    if(statut[i]==5) res[i]=proba5(i, climat, datas, alpha, mu, k
      ,L, tp, w0k)
  }
  res=res[!is.na(res)]; res[res==0]=0.0001
  resf=-sum(log(res))
  resf
}
```


6 Conclusion

Je tire un bilan très positif de ce stage, qui fut une expérience très enrichissante tant sur le plan académique et personnel. Sur le plan académique d'abord, j'ai pu implémenter les notions vues au cours de l'année académique comme l'estimation Paramétrique et la théorie des tests. J'ai rempli les objectifs fixés, à savoir améliorer mes compétences en programmation R ainsi que l'application des notions mathématiques vue en cours aux problèmes du monde réel. Sur le plan personnel ensuite, j'ai compris l'importance de la patience dans les études scientifiques et surtout en programmation. Ainsi, j'ai appris la procédure à suivre lorsque je sèche devant un problème. Je me souviens encore lorsque mon encadreur me disait que: "Je veut que tu apprenne à réfléchir comme ça". Au cours de cette période, comme dans toute phase d'apprentissage, il m'est par ailleurs arrivé de faire quelques erreurs que j'ai pu rapidement corriger en me rapprochant de mon encadreur et en respectant scrupuleusement ses consignes. Grâce aux acquis d'une méthodologie de travail forte que mon encadreur m'a transmise, combinée à la formation théorique que j'ai reçu, je suis aujourd'hui apte à animer un projet de statistique sous le logiciel R. Finalement au sortir de cette expérience, j'ai renforcer ma passion pour l'analyse des données massives qui sera certainement l'objet de mes prochains travaux.

Bibliography

- [1] Patrice Takam Soh *MCEM-Algorithm for estimating lifetime distribution in presence of interval censored observations and a censoring variable. Application to cocoa data.*
- [2] <https://cran.r-project.org/web/views/Optimization/>: Stefan Theussl, Florian Schwendinger, Hans W. Borchers
- [3] CRAN Task View: *Optimization and Mathematical Programming*
- [4] Paulo Cortez *Modern Optimization with R*
- [5] <https://rdrr.io>: *Practical Numerical Math Functions*
- [6] <https://stackoverflow.com/>
- [7] Donald E. Knuth (1986) *The T_EX Book*, Addison-Wesley Professional.