

VAM un package R pour l'analyse de Modèles d'Age Virtuel

L. Doyen et R. Drouilhet

Deux grands types de maintenance (NF EN 13306)

- Maintenance Corrective (MC) : après détection d'une panne et destinée à remettre un bien dans un état dans lequel il peut accomplir une fonction requise.
- Maintenance Préventive (MP) : à des intervalles prédéterminés ou selon des critères prescrits et destinée à réduire la probabilité de défaillance ou la dégradation du fonctionnement d'un bien.
- MP peuvent être soit systématiques soit conditionnelles

Modèle d'âge virtuel (Virtual Age Model)

- Instants des MC/MP: $\{T_n\}_{n \geq 0}$ ($T_0 = 0$).
- Types des MC/MP: $U_n = -1$ si MC et $U_n = 1, 2, \dots$ si MP.
- $\forall t \geq 0$, N_t nombre de MC/MP entre 0 et t .
- $\{T_n\}_{n \geq 0}$ processus ponctuel avec intensité de défaillance:

$$\lambda_t = \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} \Pr(N_{t+\Delta t} - N_{t-} = 1 \mid U_{N_{t-}+1} < 0, \mathcal{H}_{t-})$$

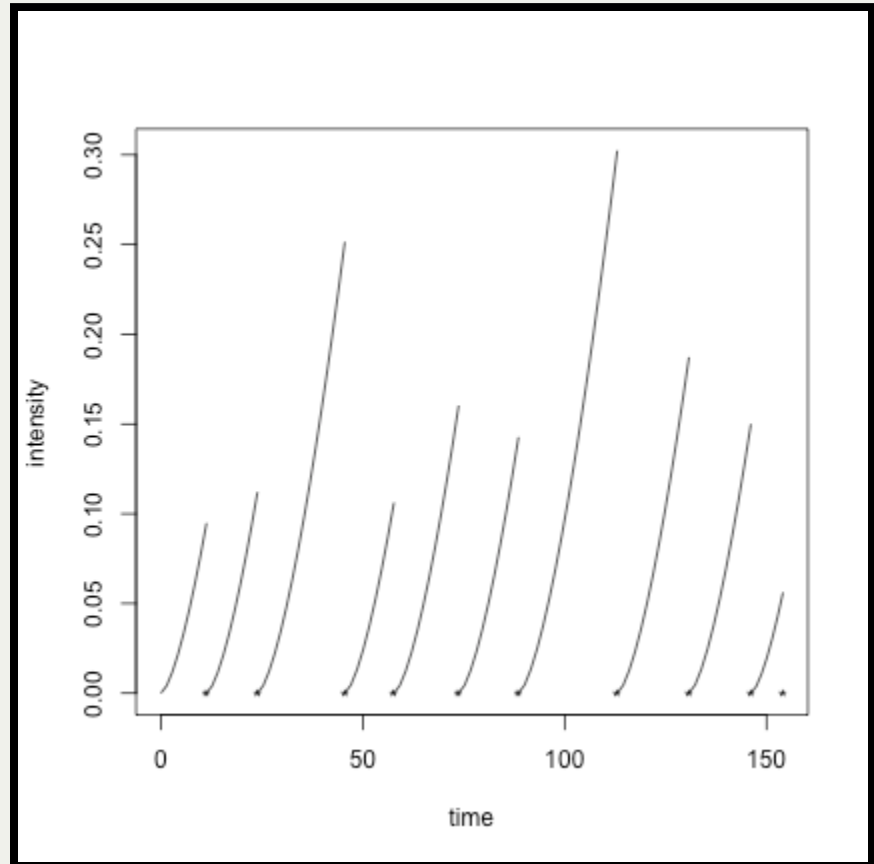
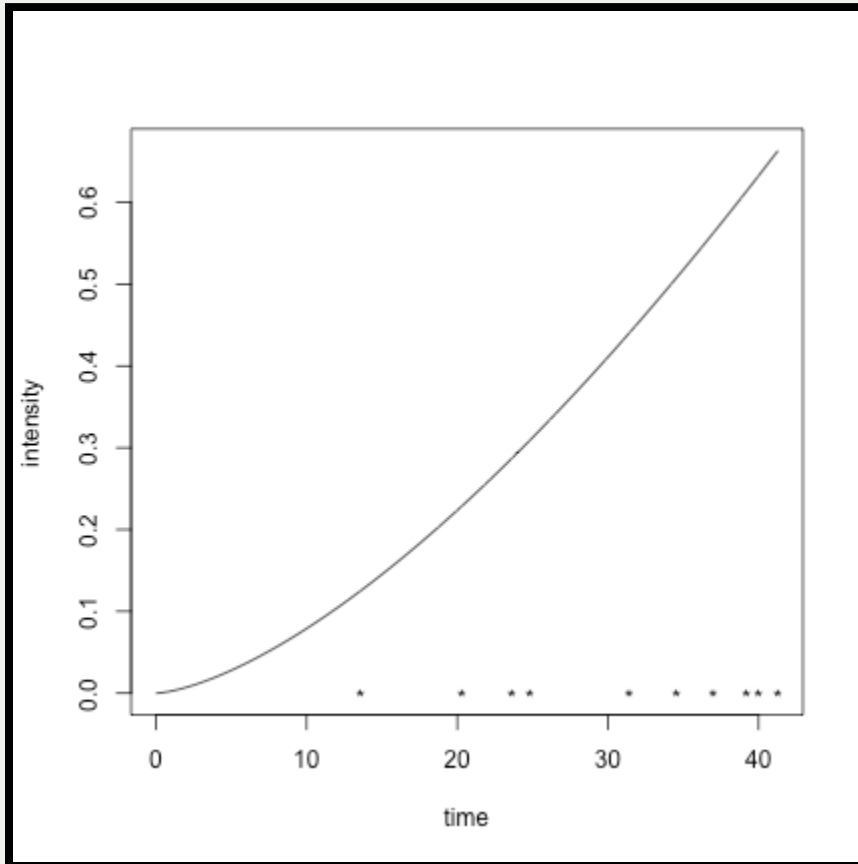
(où \mathcal{H}_{t-} est le passé du processus) de la forme

$$\lambda_t = \epsilon'_{N_{t-}}(\mathbf{t}) \times \mathbf{h}(\epsilon_{N_{t-}}(\mathbf{t}))$$

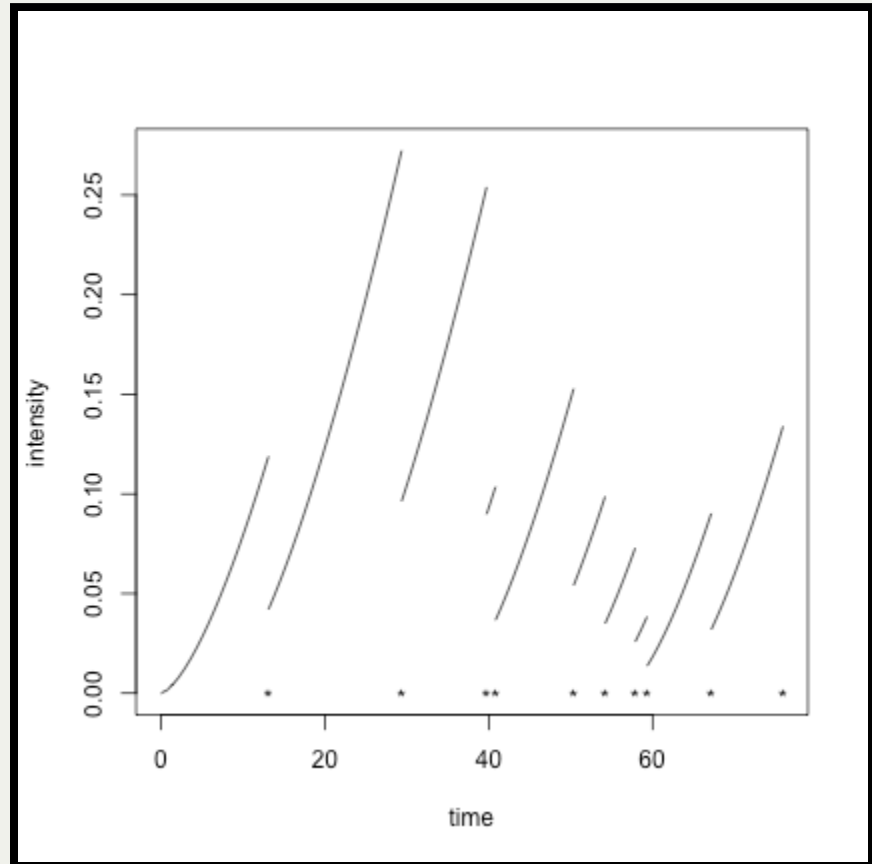
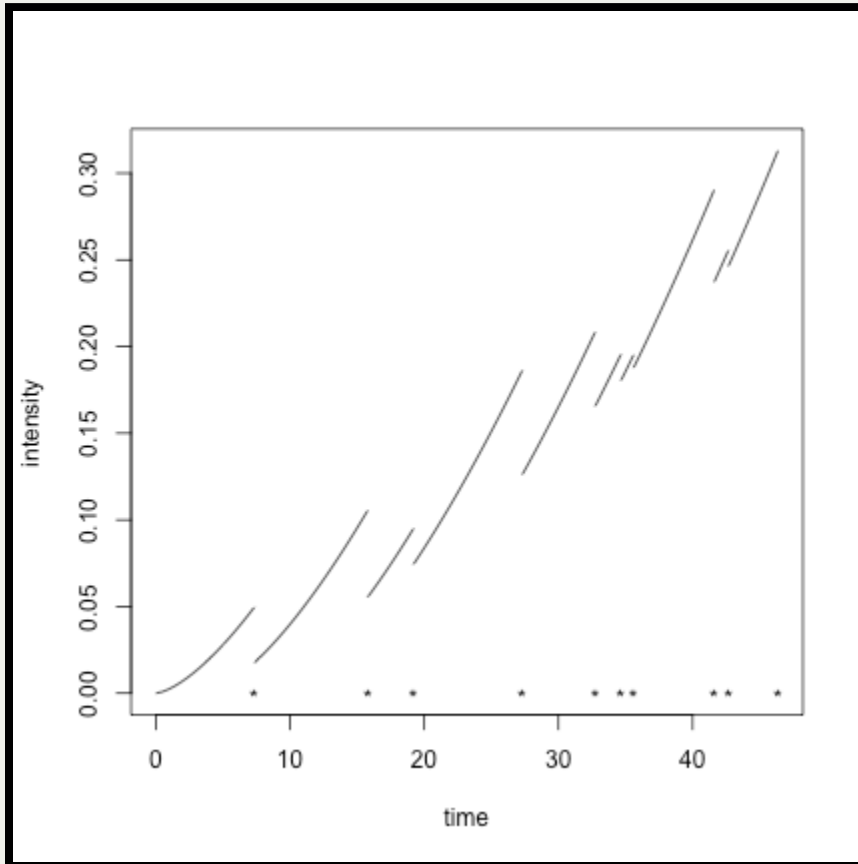
avec $\epsilon_n(\mathbf{t})$ l'âge virtuel du système pour $t \in]T_n, T_{n+1}]$.

- Dans package VAM, $\epsilon_n(t) = A_n \times (t - T_n) + B_n$.

Exemples ABAO et AGAN



Exemples ARA1(.5) et ARAInf(.5)



Simulation

```
R> sim <- sim.vam( ~ (MC|h) & (MP1+MP2+...|Policy1*Policy2*...) )  
R> # ou  
R> sim <- sim.vam( Time & Type ~ (MC|h) & (MP1+MP2+...|Policy1*Policy2*...) )
```

avec

MC = Maintenance Corrective
MP? = Maintenance(s) Préventive(s)
h = Loi nominale du taux de défaillance
Policy? = Politique de Maintenance(s) Préventive(s)

```
R> sim <- sim.vam( ~ (ARA1(.5) | Weibull(0.01,2.5))  
+               & (ARAInf(.7)+ARAInf(.3)|Periodic(12,prob=c(0.6,0.4))))  
R> # Format des variable de sortie par défaut : (System,) Temps et Type  
R> # En français, on précise les variables de sortie:  
R> sim <- sim.vam( Systeme & Temps & Type ~ (ARA1(.5) | Weibull(0.01,2.5))  
+               & (ARAInf(.7)+ARAInf(.3)|Periodic(12,prob=c(0.6,0.4))))
```

Exemples simples

```
R> (simDF <- simulate(sim,30))
      Temps Type
1    6.984441  -1
2   11.298073  -1
3   12.000000   1
...
29  72.000000   1
30  76.701856  -1
R> (simDF <- simulate(sim,Time > 30))
      Temps Type
1    9.222083  -1
2   11.125386  -1
3   12.000000   2
...
12  27.558678  -1
13  30.975850  -1
R> (simDF <- simulate(sim,30,nb.system=3))
      Systeme  Temps Type
1           1  4.674090  -1
2           1 10.092217  -1
3           1 12.000000   1
...
89          3 63.792902  -1
90          3 64.795138  -1
```

Exemples avec "stop policy"

```
R> (simDF <- simulate(sim,Size==3))
      Temps Type
1 4.124839    -1
2 4.293673    -1
3 8.093694    -1
...
2 4.293673    -1
3 8.093694    -1
R> table(simDF$Type)

-1
 3
R> (simDF <- simulate(sim,Size[2] == 4))
      Temps Type
1      8.546295    -1
2     12.000000     1
3     15.554466    -1
...
226 383.716056    -1
227 384.000000     2
R> table(simDF$Type)

-1  1  2
195 28  4
```


Exemples avec "stop policy"

```
R> (simDF <- simulate(sim, S[1]>=50 | S[2] >= 4))
```

```
      Temps Type
```

```
1      3.827365  -1
```

```
2      4.758171  -1
```

```
3      8.954676  -1
```

```
...
```

```
357 623.605117  -1
```

```
358 624.000000   1
```

```
R> table(simDF$Type)
```

```
 -1    1    2
```

```
306   50    2
```

```
R> (simDF <- simulate(sim, S[1]>=50 & S[2] >= 4))
```

```
      Temps Type
```

```
1      5.443606  -1
```

```
2      9.360594  -1
```

```
3     10.547067  -1
```

```
...
```

```
392 670.311120  -1
```

```
393 672.000000   1
```

```
R> table(simDF$Type)
```

```
 -1    1    2
```

```
337   50    6
```

Estimation MLE

```
R> mle <- mle.vam( Time & Type ~ (MC|h) & (MP1+MP2+...), data=... )
```

MC = Maintenance Corrective
MP? = Maintenance(s) Préventive(s)
h = Loi nominale du taux de défaillance
data = Matrice de données

```
R> mle <- mle.vam( Temps & Type ~ (ARA1(.5) | Weibull(0.01,2.5))  
+ & (ARAIInf(.7)+ARAIInf(.3) ) ,data=simDF)  
R> coef(mle)  
[1] 0.007437948 2.448710551 0.322577867 0.684648684 0.171559355  
R> formula(mle)  
Temps & Type ~ (ARA1(0.322577867144387) | Weibull(0.00743794799301195,  
2.44871055138657)) & (ARAIInf(0.684648683988457) + ARAInf(0.171559354920098))
```

Exemple multi-systèmes

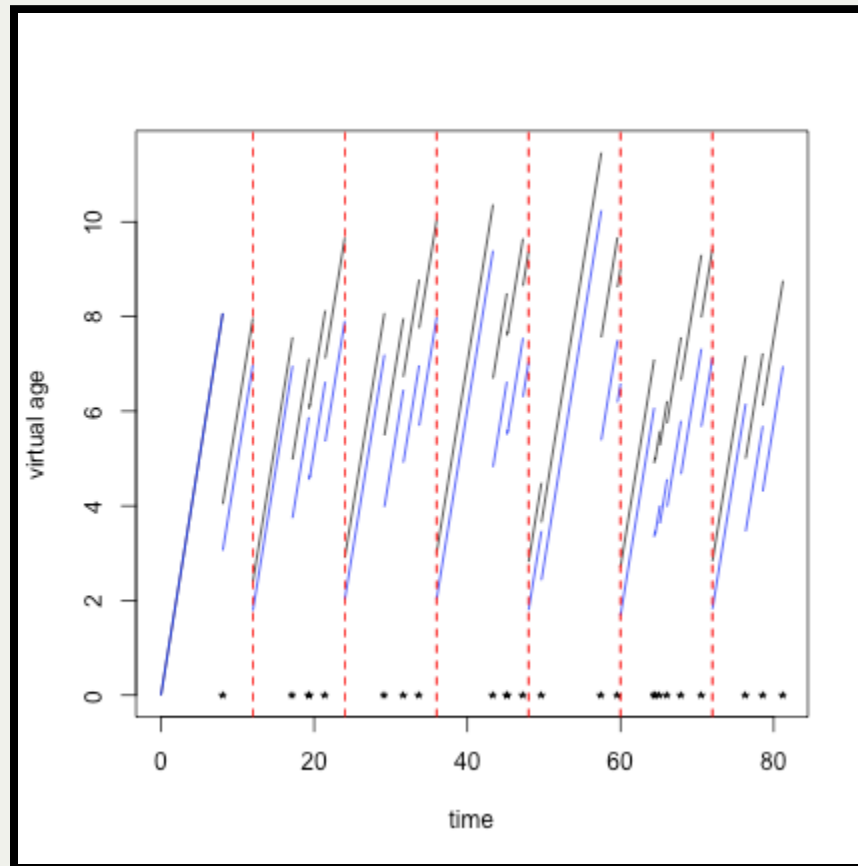
```
R> (simDF <- simulate(sim,30,nb.system=10))
      Systeme      Temps Type
1           1  8.070052   -1
2           1 12.000000    1
3           1 17.175419   -1
...
299         10 70.761027   -1
300         10 71.841015   -1
R> mle <- mle.vam( Temps & Type ~ (ARA1(.5) | Weibull(0.01,1))
+           & (ARAIInf(.5)+ARAIInf(.5) ) ,data=simDF)
R> coef(mle)
[1] 1.0037775 1.4268609 -0.2109483 0.5805128 1.0000000
R> # OUPS! Les résultats sont ... bizarres! Une erreur ???
```

Exemple multi-systèmes

```
R> mle <- mle.vam( Systeme & Temps & Type ~ (ARA1(.5) | Weibull(0.01,1))  
+           & (ARAIInf(.5)+ARAIInf(.5) ) ,data=simDF)  
R> coef(mle)  
[1] 0.009981148 2.607087775 0.621548967 0.744727857 0.376367300  
R> # Et là?  
R> formula(sim)  
Systeme & Temps & Type ~ (ARA1(0.5) | Weibull(0.01, 2.5)) & (ARAIInf(0.7) +  
  ARAInf(0.3))
```

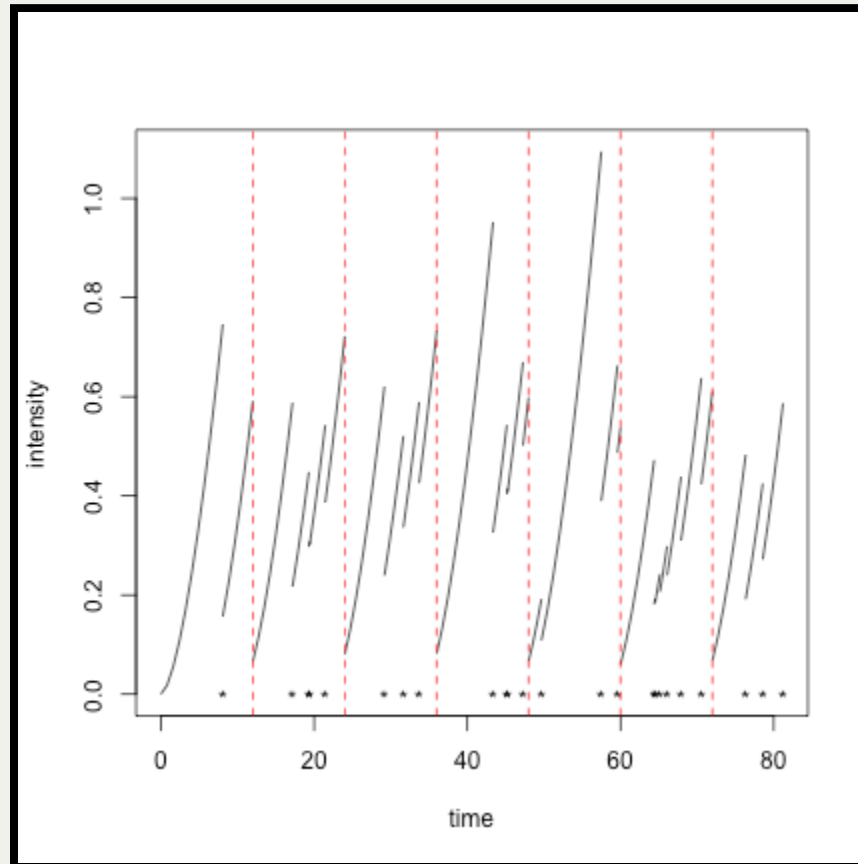
Graphique Age Virtuel

```
R> plot(sim)
R> plot(mle,col="blue",add=TRUE)
```



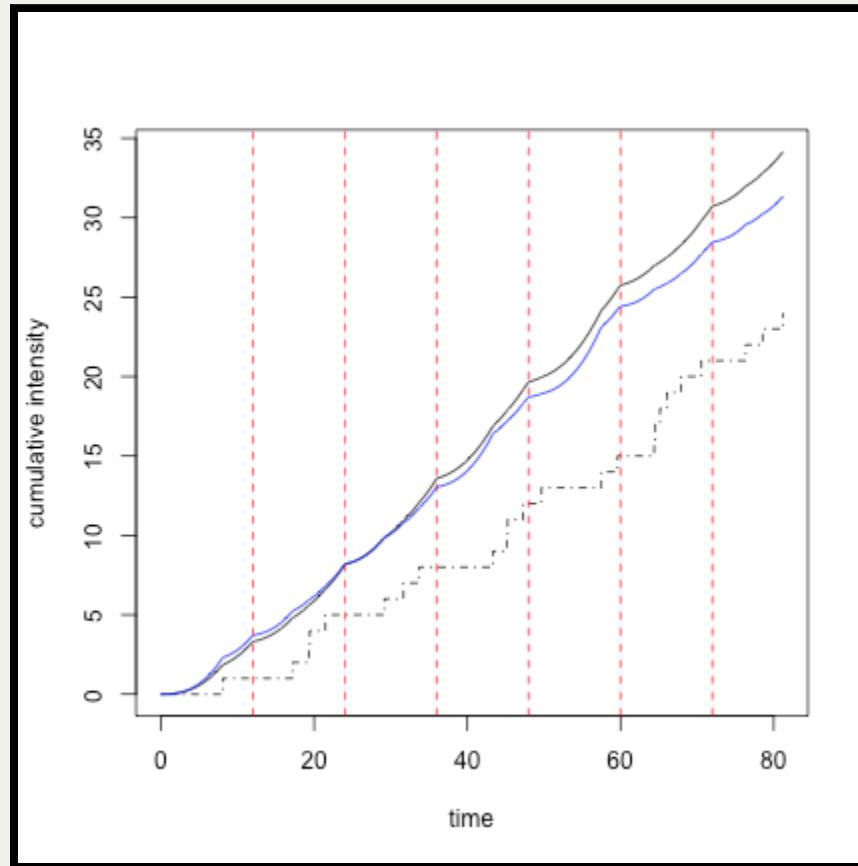
Graphique Intensité

```
R> plot(mle,type="i")
```



Graphique Intensité Cumulée

```
R> plot(sim, 'I')  
R> plot(mle, 'I-cm-pm', col='blue', add=TRUE)
```

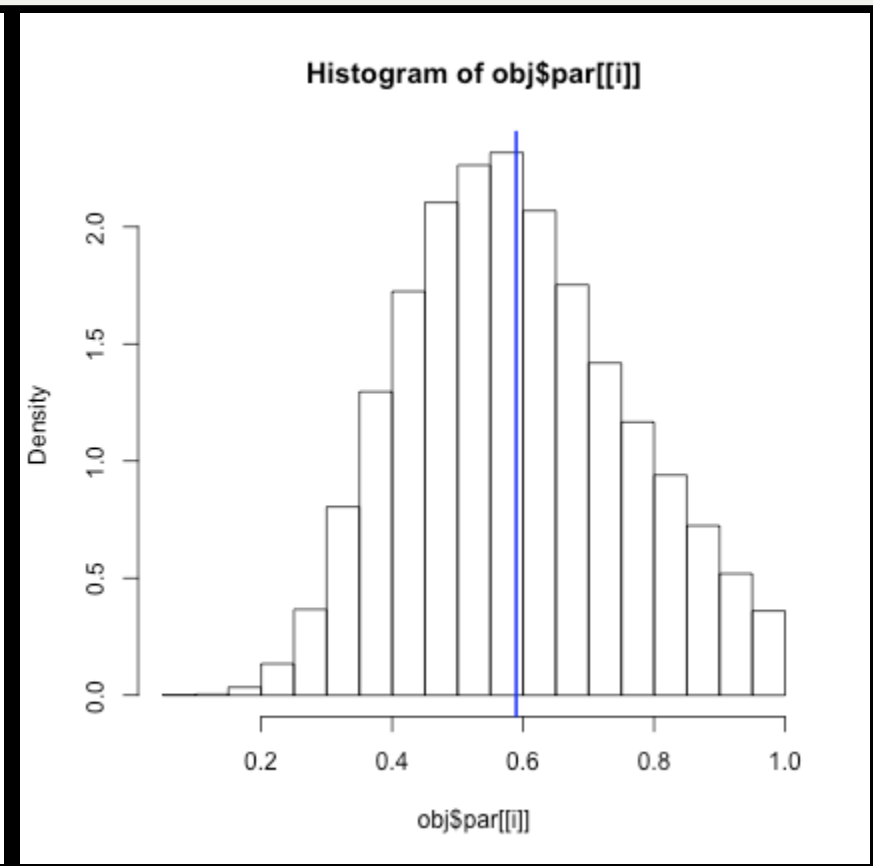
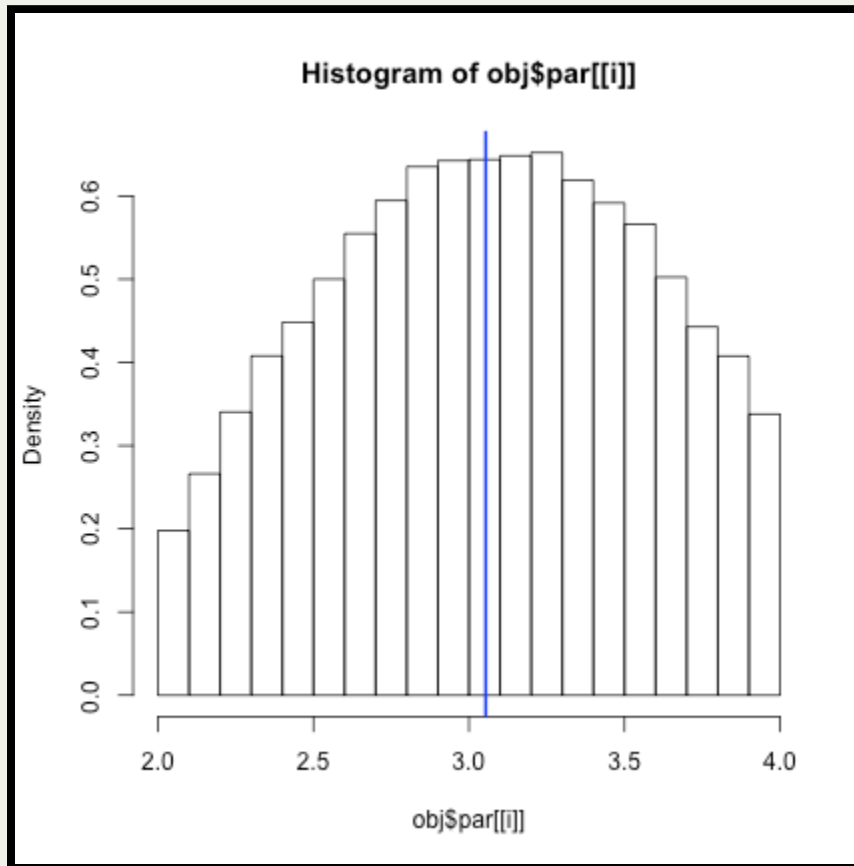


Bayesien (Expérimental)

```
R> sim <- sim.vam( ~ (ARAIInf(.4) | Weibull(.001,2.5)))
R> simDF<-simulate(sim,30)
R> bayes <- bayesian.vam(
+ Time & Type ~ (ARAIInf(~Unif(0,1)) | Weibull(~Unif(1,1.5),~Unif(2,4))),
+ data=simDF)
R> coef(bayes)
[1] 0.0004413119 3.0539195534 0.5894580954
R> coef(bayes$mle)
[1] 0.0001014392 3.4494951639 0.5030418564
R> summary(bayes)
Initial parameters (by MLE): 0.000101439243466701, 3.44949516389172, 0.5030418563
(Mean) Bayesian estimates: 0.00044131187251543, 3.05391955344369, 0.5894580953577
(SD) Bayesian estimates: 0.504794965180067, 0.167928435921964
(Number) Bayesian estimates: 54885, 41546
```

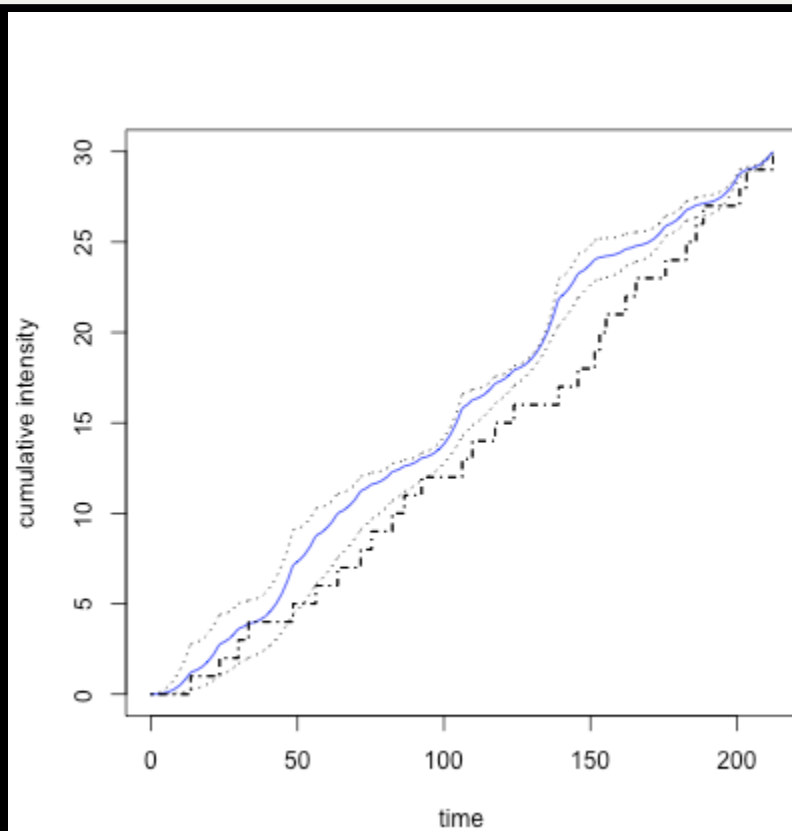
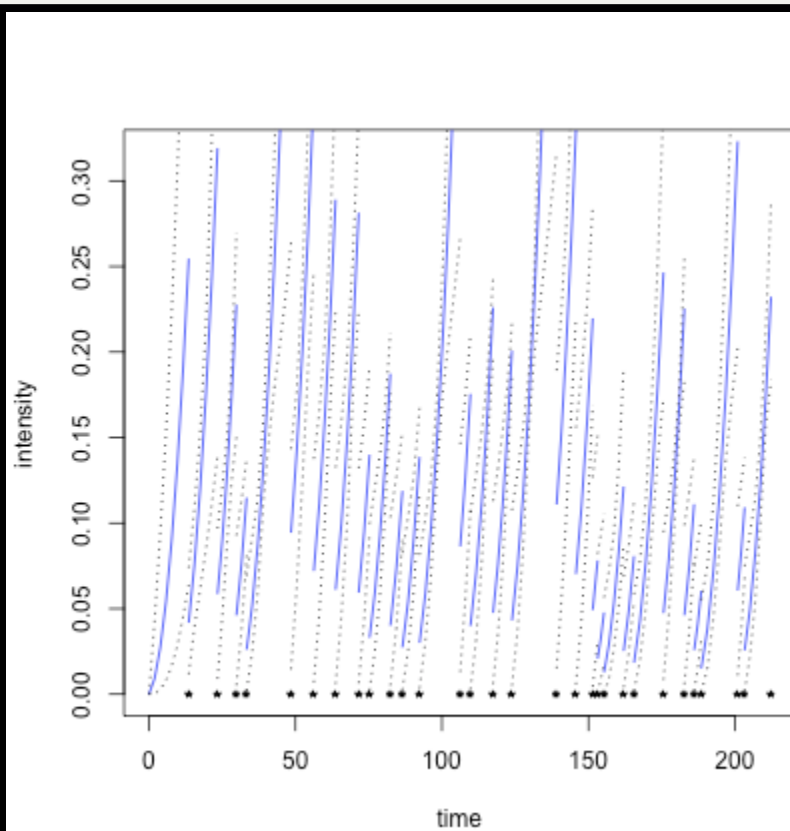

Bayesien: Répartitions

```
R> hist(bayes)
R> hist(bayes, 2)
```



Bayésien: Intensité et Cumulée

```
R> plot(bayes,type="i")  
R> plot(bayes,type="I")
```



Développement en quelques mots

- Usage de Rcpp obligatoire pour le calcul de la vraisemblance (coûteux en temps car dépendant de tout l'historique)
- Facilité et souplesse d'utilisation grâce à l'interfaçage des modèles via l'élégance des formules R
- Utilisation des objets `environment` en lieu et place des objets `list`
- Expérimental: tentative de simuler la persistance des objets Rcpp

.

Classes C++

- LA classe C++:

```
class VamModel {  
  ...  
  MaintenanceModelList* models;  
  FamilyModel* family;  
  MaintenancePolicy* maintenance_policy;  
  ...  
}
```

- Les classes C++ (SimVam, MLEVam, BayesianVam) sont accessibles dans R via le concept de module Rcpp

```
RCPP_MODULE(vam_module) {  
  ...  
  class_( "SimVam" )  
    .constructor<List>()  
    .method("simulate",&SimVam::simulate,"simulate")  
    ...  
  ;  
  class_( "MLEVam" )  
    .constructor<List,List>()  
    ...  
}
```

Extraction des infos de la formule R

```
R> (model <- parse.vam.formula(~ (ARA1(.5) | Weibull(.01,2))))  
$response  
NULL  
  
$models  
$models[[1]]  
$models[[1]]$name  
[1] "ARA1.va.model"  
  
$models[[1]]$params  
[1] 0.5  
...  
  
$pm.policy  
$pm.policy$name  
[1] "None"  
  
$max_memory  
[1] 1
```

Declaration de `sim.vam` en R (connecté à `SimVam` en C++)

```
sim.vam <- function(formula) {  
  self <- newEnv(sim.vam, formula=formula(formula))  
  ...  
  model <- parse.vam.formula(self$formula)  
  self$rcpp <- new(SimVam, model)  
  self  
}
```