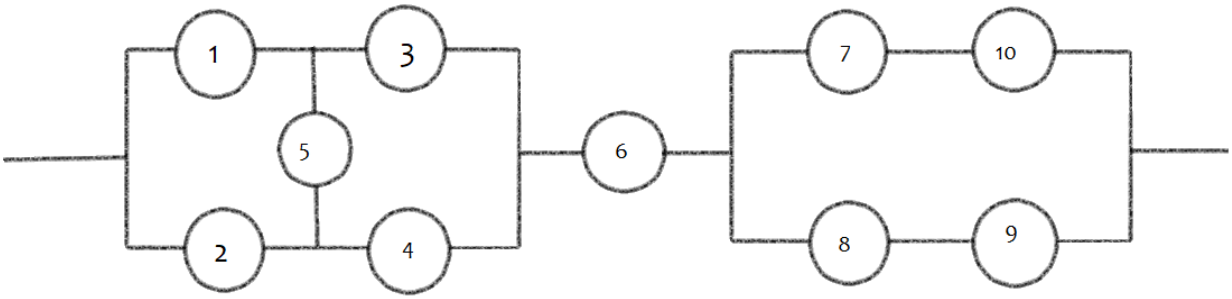
rapport

Étude d'un système complexe avec possibilité de réparations de composants

Dans cette seconde partie, nous étusierons un autre système complexe, composé de 10 composants. Le but de cette partie sera de faire de nombreuse simulation et voir ce qui se passe dans le cas où un composant est réparable. Nous étudierons aussi l'impacte financier que des pannes pour une entreprise. Dans cette partie, nous éviterons de mettre les morceaux de code car nous en avons déja beaucoup mis dans la partie précédente et dans cette partie, les code sont plus ou moin simalaire

Fonction de structure

Le système que nous allons simuler a 10 composants. Voici le schéma du système :



Il a pour fonction de structure :

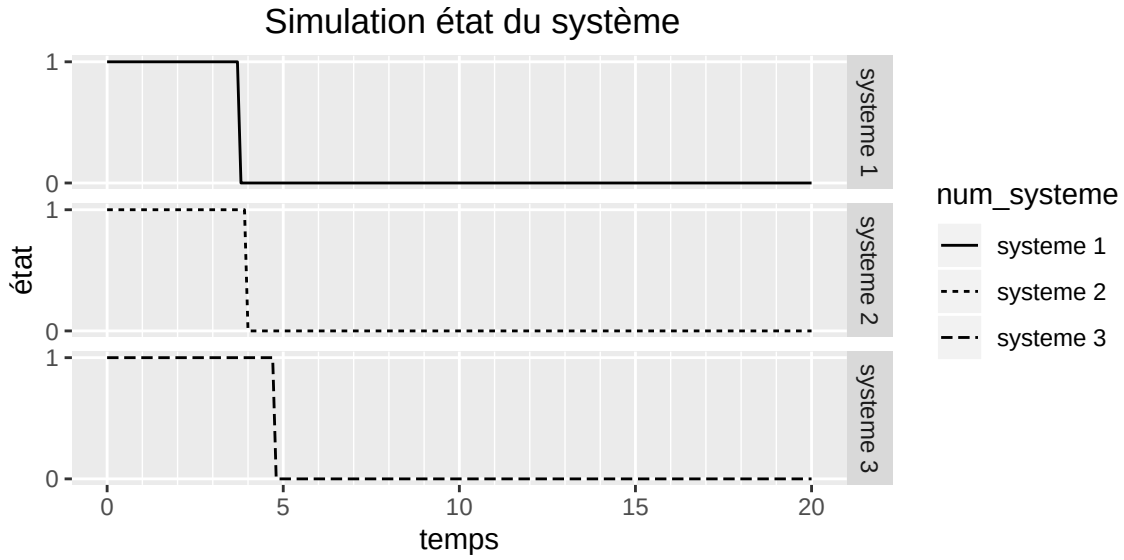
$$\phi(x) = [x_5(1-(1-x_1)(1-x_2))(1-(1-x_3)(1-x_4))+(1-x_5)(1-(1-x_1x_3)(1-x_2x_4))]x_6[1-(1-x_7x_{10})(1-x_8x_9)]$$

Trajectoire de l'état du système et espérence de la durée de vie

Trajectoire de l'état du système

Nous allons maintenant simuler l'état du système. On considérera que les composantes du système suivent des lois de Weibull de paramètre $\lambda = 5$ et $\beta = 5$.

La trajectoire du système est :



En fixant les paramètres à $\lambda = 5$ et $\beta = 5$, le premier système a une durée de vie de t = 3,8, le deuxième de t = 4 et le troisième de t = 4.8.

On va maintenant utiliser la loi des grands nombres pour estimer l'espérance de la durée de vie de notre système. On simule le temps de vie de moyen sur 10000 systèmes.

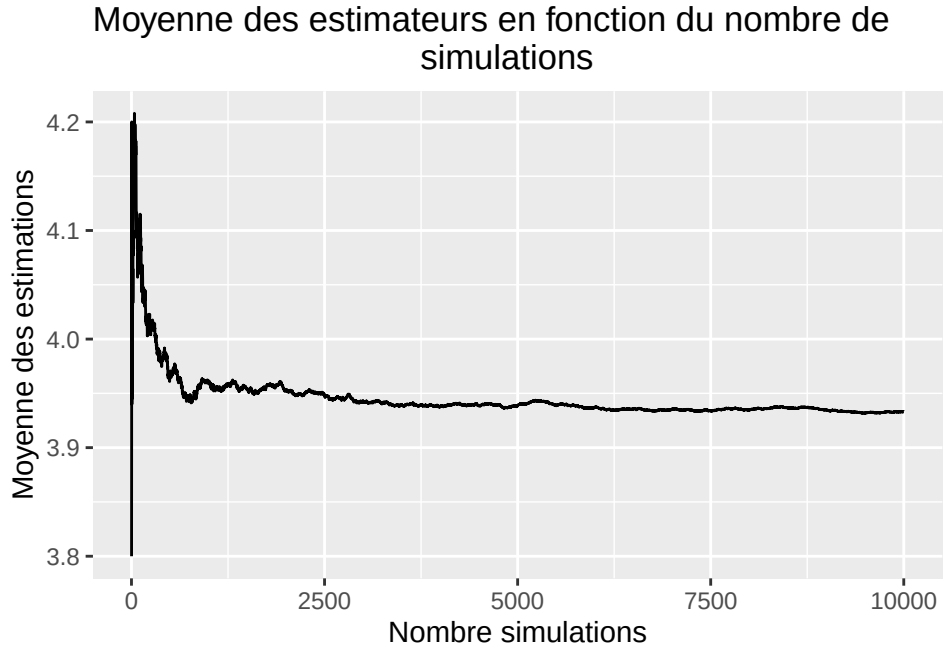
Estimation de l'espérance de la durée de vie et intervalle de confiance

```
## [1] 3.93306
```

```
## [1] 3.919898 3.946222
```

On estime le temps de vie moyen de notre système à 3,933. Son intervalle de confiance à 95% est $[3,920; 3.946]$.

On va maintenant représenter notre estimation de l'espérance de la durée de vie de notre système en fonction du nombre de simulations choisies.



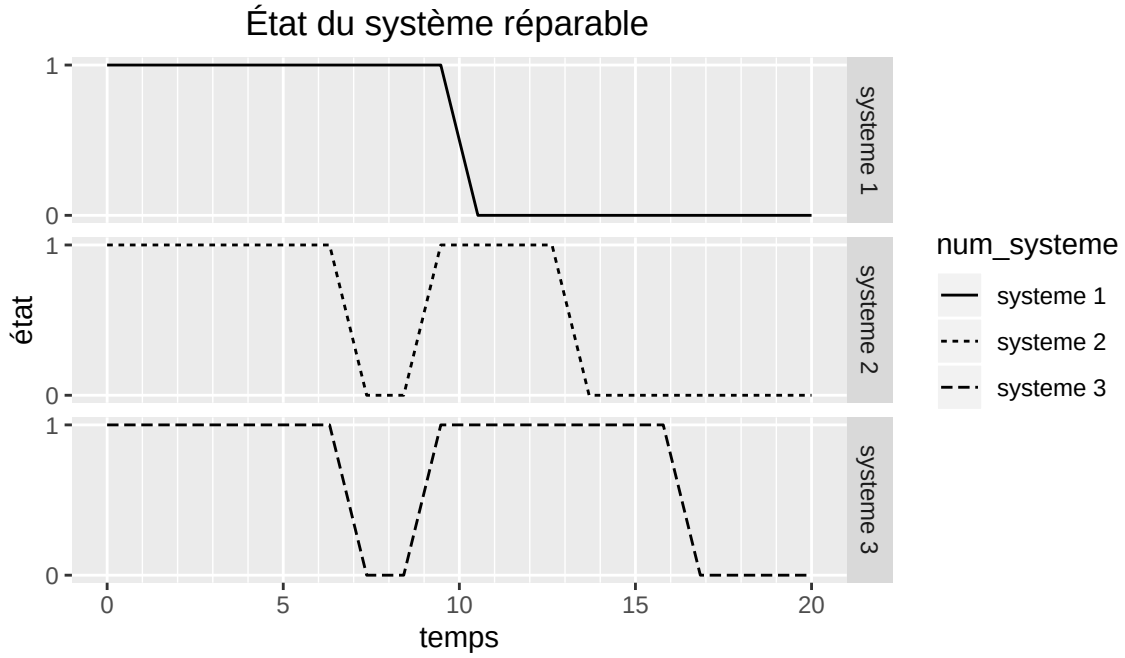
On voit sur ce graphe que le temps de vie moyen converge entre 3.9 et 4. Notre estimation a l'air de se stabiliser à partir de 3000 simulations.

Réparation du système

Trajectoire de l'état du système avec réparation

Le composant critique que nous avons choisis est le composant x_6. En effet, il est un composant critique car il est en série avec tout le reste du système et donc une panne de ce composant entraine forcément une panne du système. Nous allons maintenant implémenter le fait que ce composant soit réparable. On part du principe qu'un technicien vient tous les intervalles de temps δ pour le réparer : si le composant est en panne, il le répare, sinon il ne fait rien. Dans les deux cas, le déplacement du technicien implique un coût.

Nous allons tracer quelques trajectoires de l'état de notre nouveau système réparable. Cette fois-ci, on fixe $\lambda = 15$ et $\beta = 3$.



Pour la première trajectoire, on voit que le système tombe en panne au temps t = 10 et il n'y a pas de réparation ensuite, du moins pas de réparation qui remette le système en marche. Pour la deuxième trajectoire, on voit que le système tombe une première fois en panne au temps t = 7, puis il y a réparation au temps t = 9, puis le système retombe en panne au temps t = 13. On peut donc dire que la réparation du composant critique a permis au système de tenir un peu plus longtemps que prévu (il a gagné t = 4 de durée de vie). Pour le troisième système, on remarque qu'il tombe une première fois en panne et qu'il est réparé en même temps que le système 2 (aux temps t = 7 puis t = 9), seulement ce système met plus de temps que le précédent à retomber en panne (il retombe en panne à t = 16 contre t = 13 pour le précédent). Dans ce système, le fait que le composant critique soit réparable a doublé la durée de vie de notre système (il passe d'une durée de vie de 6 à 12).

Nous allons maintenant regarder avec quel intervalle de temps δ il faut demander au technicien d'intervenir pour que le système soit le plus rentable possible. La fonction de récompense est la suivante :

$$R(\delta) = a\mathbb{E}(T) - C\mathbb{E}(N)$$

. Le coefficients a correspond au gain du système associé au temps, et le coefficient C correspond au coût d'intervention du technicien. N est le nombre moyen d'intervention du technicien avant l'instant T de panne du système.

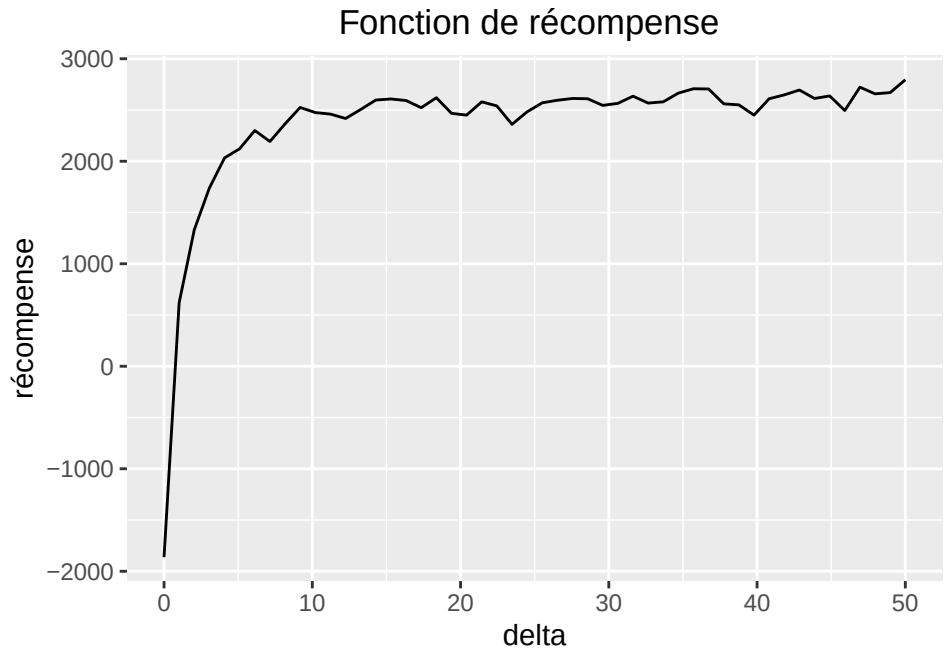
On implémente cette fonction, puis on cherche à la maximiser par rapport à δ. Comme précédemment les valeurs de $\mathbb{E}(T)$ et $\mathbb{E}(N)$ sont estimées par la loi des grands nombres.

Fonction de récompense

Pour la maximisation, on va fixer reprendre les paramètres utilisés précédemment : on fixe $\lambda = 15$, $\beta = 3$. Pour les coefficients de gain associé au temps et de cout d'intervention du technicien, on les fixe arbitrairement : on fixe $a = 250$ et $C = 100$.

```
## $maximum
## [1] 78.01667
##
## $objective
## [1] 2807.5
```

Lorsqu'on cherche à maximiser la fonction de récompense, on obtient qu'il faut que le technicien intervienne tous les 78 temps pour avoir un résultat optimal, et ce pour une récompense de 2807. Lorsque qu'on regarde l'évolution de la fonction recompense en fonction de δ, on obtient :



Ici on se limite à δ allant de 1 à 50 pour limiter le temps de compilation. On voit sur le graphe que la fonction optimisation a une allure croissante : plus l'intervalle de temps entre deux intervention du technicien est élevée (donc moins on demande au technicien d'intervenir), plus la récompense est élevée. On en déduit donc que l'intervention du technicien n'est pas rentable en vue de notre système.

Annexe

projet

question 1

```
phi <- function(x){
  rep <- (x[5]*((1-(1-x[1])*(1-x[2]))*(1-(1-x[3])*(1-x[4])))+(1-x[5])*(1-(1-x[1]*x[3])*(1-x[2]*x[4])))*
  return(rep)
}
```

Question 2

```
simu_etat_systeme <- function(lambda, beta, t){
  X <- matrix(data = rep(0, 10*length(t)), ncol = 10, nrow = length(t))
  S <- rep(0,length(t))
  u <- runif(10, 0, 1)
  w <- rweibull(10, lambda, beta)
  max <- max(t)

  for(i in 1:length(t)){
    for(j in 1:10){
      if(t[i] < w[j]){
        X[i,j] <- 1
      }
    }
    S[i] <- phi(X[i,])
    if((S[i] == 0) && (S[i-1] == 1)){
      max <- t[i]
    }
  }
  return(list(etat = S, last = max))
}
```

```
set.seed(568)
lambda <- 5
beta <- 5
t <- seq(0, 20, 0.1)

set.seed(1)
systeme1 <- simu_etat_systeme(lambda, beta, t)
systeme2 <- simu_etat_systeme(lambda, beta, t)
systeme3 <- simu_etat_systeme(lambda, beta, t)
df1 <- data.frame(systeme = c(systeme1[[1]], systeme2[[1]], systeme3[[1]]),
                  num_systeme = c(rep("systeme 1",length(t)), rep("systeme 2",length(t)), rep("systeme 3
                  t = rep(t, 3))

g1 <- df1 %>%
  ggplot() +
  geom_line(aes(x = t, y = systeme, linetype = num_systeme)) +
  labs(title = "simulation état du système",
       x = "temps",
       y = "état") +
  scale_y_continuous(breaks = c(0, 1), minor_breaks = c(0,1)) +
  scale_x_continuous(breaks = seq(0, 20, 5), minor_breaks = seq(0, 20, 1)) +
  facet_grid(num_systeme ~ .)
g1
```

```
simu_T <- function(n){
  simu <- rep(-1, n)
  for(i in 1:n){
    simu[i] <- simu_etat_systeme(lambda, beta, t)[[2]]
  }
```

```
  return(simu)
}

set.seed(1)
n <- 10000
simu <- simu_T(n)

esperance <- mean(simu)

confint <- c(mean(simu) - 1.96 * sd(simu)/sqrt(n), mean(simu) + 1.96 * sd(simu)/sqrt(n))

esperance
confint
```

```
nb.simu <- matrix(1:n, n, 1)
esp.cumul <- cumsum(simu)/nb.simu
df.si <- data.frame(nb.simu, esp.cumul)
ggplot(df.si,aes(x=nb.simu, y=esp.cumul))+
 geom_line()+
 xlab("Nombre simulations")+
 ylab("Moyenne des estimations")
```

Question 3

```
fonction_interval=function(n,inter,i){
  t=NULL
  d=NULL
  for (v in seq(1,n,inter)) {
    if(v==i){
      t[v]=1
    }
    else{
      t[v]=0
    }
  }
  t=t[!is.na(t)]
  w=sum(t)
  if(w==0){
    d=0
  }
  else{
    d=1
  }
  return(d)
}

phi2 <- function(a, n, lambda, beta,inter){
  t <- seq(0, a,length.out =n)
  y <- rep(0,n)
  u <- runif(10,0,1)
  x <- matrix(0, nrow = n, ncol = 10)

  for(i in 1:n){
```

```
    for(j in c(1,2,3,4,5,7,8,9,10)){
      if(t[i] < (lambda*(-log(u[j]))^(1/beta))){
        x[i,j] <- 1
      }
    }
  }
  d=0
  for(i in 1:n){
    if(t[i-d] < (lambda*(-log(u[6]))^(1/beta))){
      x[i,6] <- 1
    }
    if(fonction_interval(n,inter,i)==1 & x[i,6]==0){
      d <- i
      u[6] <- runif(1,0,1)
    }
  }
  # y[i] <- phi(x[i,])
  return(x)
}

phi3 <- function(a, n, lambda, beta,inter){
  y=NULL
  matrice=phi2(a, n, lambda, beta,inter)
  for (i in 1:n) {
    y[i]=phi(matrice[i,])
  }
  return(y)
}
```

```
a <- 20
n <- 20
lambda <- 15
beta <- 3
inter <- 4

set.seed(10560)
df2 <- data.frame(systeme = c(phi3(a,n,lambda,beta,inter), phi3(a,n,lambda,beta,inter), phi3(a,n,lambda
                  num_systeme = c(rep("systeme 1",n), rep("systeme 2",n), rep("systeme 3",n)),
                  t = rep(seq(0, a, length.out = n), 3))

g2 <- df2 %>%
  ggplot() +
  geom_line(aes(x = t, y = systeme, linetype = num_systeme)) +
  labs(title = "état du système réparable",
       x = "temps",
       y = "état") +
  scale_y_continuous(breaks = c(0, 1), minor_breaks = c(0,1)) +
  scale_x_continuous(breaks = seq(0, 20, 5), minor_breaks = seq(0, 20, 1)) +
  facet_grid(num_systeme ~ .)
g2
```

```
# Esperance de T
tp_i_de_panne <- function(a, n, lambda, beta,inter){
  y=NULL
```

```
  d=0
  matrice=phi2(a, n, lambda, beta,inter)
  for (i in 1:n) {
    y[i]=phi(matrice[i,])
  }
  for (i in n:1){
    if(y[i]==1 & i==n ){
      d=n
      break
    }
    if(y[i]==1){
      d=i+1
      break
    }
  }
  return(d)
}

nbtot_realisation_t=function(a, n, lambda, beta,inter,nbtot){
  vector=NULL
  for(i in 1:nbtot){
    vector[i]=tp_i_de_panne(a, n, lambda, beta,inter)
  }
  return(vector)
}

E_T=function(a, n, lambda, beta,inter,nbtot){
  t=nbtot_realisation_t(a, n, lambda, beta,inter,nbtot)
  moyenne=(1/nbtot)*sum(t)
  return(moyenne)
}
```

```
# Esperance de N
nb_intervention <- function(a, n, lambda, beta,inter){
  t <- seq(0, a,length.out =n)
  y <- rep(0,n)
  u <- runif(10,0,1)
  x <- matrix(0, nrow = n, ncol = 10)

  for(i in 1:n){
    for(j in c(1,2,3,4,5,7,8,9,10)){
      if(t[i] < (lambda*(-log(u[j]))^(1/beta))){
        x[i,j] <- 1
      }
    }
  }
  d=0
  intervention=0
  for(i in 1:n){
    if(t[i-d] < (lambda*(-log(u[6]))^(1/beta))){
      x[i,6] <- 1
    }
    if(fonction_interval(n,inter,i)==1){
      intervention <- intervention + 1
```

```
      if(x[i,6] == 0){
        d <- i
        u[6] <- runif(1,0,1)
      }
    }
  }
  return(intervention)
}

nb_realisation_intervention=function(a, n, lambda, beta,inter,nbtot){
  vector=NULL
  for(i in 1:nbtot){
    vector[i]=nb_intervention(a, n, lambda, beta,inter)
  }
  return(vector)
}

moyenne_intervention=function(a, n, lambda, beta,inter,nbtot){
  moyenne=(1/nbtot)*sum(nb_realisation_intervention(a, n, lambda, beta,inter,nbtot))
  return(moyenne)
}
```

```
# Fonction récompense
a_val <- 200
n_val <- 200
lambda_val <- 15
beta_val <- 3
nbtot_val <- 100
cout_val <- 100
gain_val <- 250

set.seed(1)
recompense<-function(inter, a=a_val, n=n_val, lambda=lambda_val, beta=beta_val,nbtot=nbtot_val,cout=cou
  gain_moyen <- gain*E_T(a, n, lambda, beta,inter,nbtot)
  cout_moyen <- cout*moyenne_intervention(a, n, lambda, beta,inter,nbtot)
  r <- gain_moyen - cout_moyen
  return(r)
}
solution <- optimize(recompense,c(0, 200),maximum = TRUE)
solution
```

```
## $maximum
## [1] 78.01667
##
## $objective
## [1] 2807.5
```

```
a_val <- 50
n_val <- 50
lambda_val <- 15
beta_val <- 3
nbtot_val <- 100
cout_val <- 100
gain_val <- 250
```

```r
set.seed(1)
value <- 0
for(i in 1:50){
  value[i] <- recompense(i)
}

df3 <- data.frame(value,
                  t = seq(0, a_val, length.out = n_val))

g3 <- df3 %>%
  ggplot() +
  geom_line(aes(x = t, y = value)) +
  labs(title = "fonction de récompense",
       x = "delta",
       y = "récompense")
g3
```