Chaîne de Markov: Réservoir

Luis González, Estia Maliqari

14/11/2019

Préliminaires: Bibliothèques nécessaires

```
install.packages("markovchain")
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/3.6'
## (as 'lib' is unspecified)
```

```
install.packages("ggplot2")
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/3.6'
## (as 'lib' is unspecified)
```

```
install.packages("reshape2")
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/3.6'
## (as 'lib' is unspecified)
```

```
library(markovchain)
```

```
## Package:   markovchain
## Version:   0.8.4.1
## Date:      2020-05-04
## BugReport: http://github.com/spedygiorgio/markovchain/issues
```

```
library(ggplot2)
library(reshape2)
```

Question 1 : *Montrer que Z__n est une chaîne de Markov. Donner son espace d'état et sa fonction de répartition.*

Pour Z__n+1 on a :
- Z__n + X__n+1 pour 1 <= Z__n + X__n+1 <= c
- c-1 pour Z__n + X__n+1 > c
- 0 pour Z__n + X__n+1 <= 1

Alors, on a Z__n+1 = f(Z__n , X__n+1). L'état futur dépend seulement de l'état présent et pas du passé. (Z__n) est bien une chaîne de Markov. Le fonction de transition est donné par le fonction *populate(i, j)*, où i et j sont deux états.
La suite (X__n) suit la loi p qui est donné ci-dessous.
Tenons en compte qu'en langage R, l'indice du vecteur commence de 1 et pas de 0.

```
c=7
p <- c(0.1, 0.6, 0.1, 0.05, 0.05, 0.05, 0.05)
etats <- 0:(c-1)
```

```r
h <- function(j) {
  sum=0
  for(k in etats) {
    if(k>=j)
      sum=sum+p[k+1]
  }
  return (sum)
}
P <-matrix(0, nrow=c, ncol=c)
populate <- function(i, j) {
  if((i==j) & (i==0)) p[1]+p[2]
  else if (j==i-1) p[1]
  else if (j==c-1) h(c-i)
  else if (j>=i) p[j-i+2]
  else 0

}
for(i in etats){
  for(j in etats){
    P[i+1,j+1] <- populate(i,j)
  }
}
```

Espace d'états

L'espace d'états est l'ensemble de 0 à c-1 : E = $\{0, 1, \ldots, \text{c-1}\}$.

```r
et <-as.character(etats)
print(et)
```

```
## [1] "0" "1" "2" "3" "4" "5" "6"
```

Matrice de transition

La matrice de transition P est construit à travers des fonctions au-dessus.

```r
CM <- new("markovchain", states=et, byrow=TRUE, transitionMatrix=P, name="Volume du reservoir")
print(CM)
```

```
##     0   1    2    3    4    5    6
## 0 0.7 0.1 0.05 0.05 0.05 0.05 0.00
## 1 0.1 0.6 0.10 0.05 0.05 0.05 0.05
## 2 0.0 0.1 0.60 0.10 0.05 0.05 0.10
## 3 0.0 0.0 0.10 0.60 0.10 0.05 0.15
## 4 0.0 0.0 0.00 0.10 0.60 0.10 0.20
## 5 0.0 0.0 0.00 0.00 0.10 0.60 0.30
## 6 0.0 0.0 0.00 0.00 0.00 0.10 0.90
```

Question 2 : *Tracer une trajectoire du réservoir dans l'intervalle de temps [0,100].*

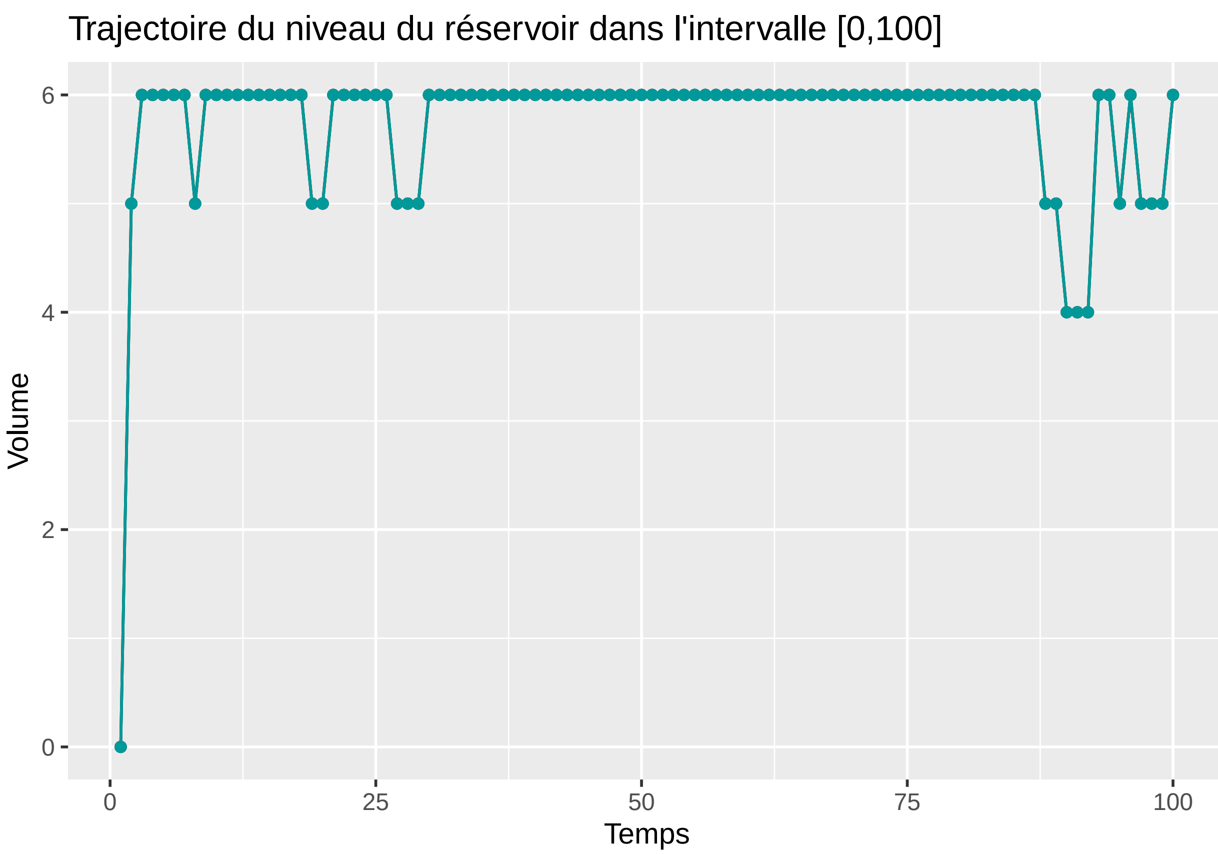
Dans l'enoncé il est demandé de prendre c=10 comme niveau maximal du réservoir. Cependant, la loi suggerée p de X_n nous donne des probabilités pour un niveau maximal de 6. C'est pourquoi les solutions sont réalisés tenant en compte c=7. Le fonction ci-dessous nous donne une réalisation d'une trajectoire de la chaîne Z_n prenant pour chaque paire d'états son probabilité.

```r
sim.mc <- function(i0, P, n.sim){
  S=1:nrow(P)
  Z=rep(0, n.sim)
  Z[1]<-i0
  for(i in 2:n.sim){
    Z[i]<-sample(x=S, size=1, prob=P[Z[i-1],])
  }
  return(Z)
}
```

Graphique du trajectoire

Dans le graphique suivant, on peut voir la trajetoire du niveau du réservoir dans l'intervalle [0,100]

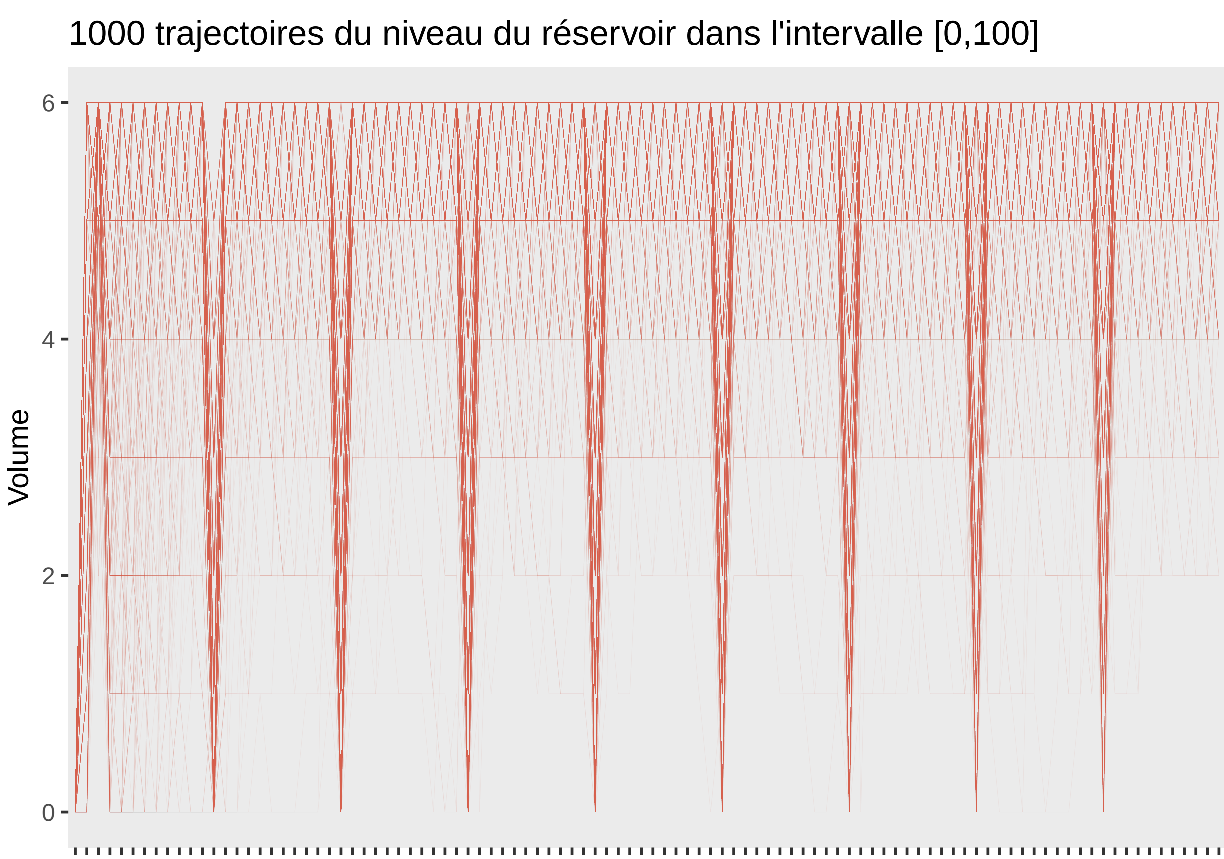
```r
niv <- sim.mc(1, P, 100)-1
qplot(x=1:100,y=niv, main="Trajectoire du niveau du réservoir dans l'intervalle [0,100]",
xlab ="Temps", ylab="Volume",
geom = c("line", "point")) + geom_point(color='#009999') + geom_line(color="#009999")
```



Question 3 ***Tracer 1000 trajectoires, estimer le niveau moyen du réservoir et faire le calcul analytique***

Simulation des 1000 trajectoires

```
z1000 <- replicate(1000, sim.mc(1, P, 100)-1)
rownames(z1000) <- paste("Temps", seq(0:99))
colnames(z1000) <- paste("Trajectoire ", seq(1:1000))
z1000df <- as.data.frame(z1000)
z1000df$temps <- rownames(z1000df)
mz1000df <- melt(z1000df, id.vars="temps")
ggplot(mz1000df, aes(x=temps, y=value, group=variable)) +
    theme_gray() +
    theme(panel.grid=element_blank()) +
    geom_line(size=0.075, alpha=0.075, colour="#D6604D") +
    theme(axis.title.x=element_blank(),
    axis.text.x=element_blank()) +
    ggtitle("1000 trajectoires du niveau du réservoir dans l'intervalle [0,100]") +
    ylab("Volume")
```



Niveau moyen du réservoir par simulation

```
mean_sim <- mean(z1000)
print(mean_sim)
```

```
## [1] 5.28409
```

Niveau moyen du réservoir par analyse

```
n<-100
meanZ<- function() {
  mean<-0
  for(y in etats){
    mean<-mean+(y*(CM^n)[1, y+1])
  }
  return (mean)
}
print(meanZ())
```

```
## [1] 5.626953
```

Conclusion: *Quand on augmente le nombre de simulations, la moyenne de ces simulations tends vers la moyenne du calcul analytique.*

Question 4 : *Calculer et estimer la valeur moyenne de T lorsque le niveau initial est u = 4.*

```
z1000_u4 <- replicate(1000, sim.mc(5, P, 100)-1)
T <- c()
for(i in 1:100){
T[i] <- sum(z1000_u4[,i]==0)
}
print(mean(T))
```

```
## [1] 0.04
```

Question 5 : *Calculer la loi stationnaire de Z_n et le niveau moyen d'équilibre de ce réservoir.*

Loi stationnaire de Z_n

On trouve la loi stationnaire de Z_n à l'aide du code suivant :

```
K<-c
A_basic <- t(diag(rep(1,K))-P)
b_basic <- rep(0,K)
A_constr <- rbind(A_basic,rep(1,K))
b_constr <- c(b_basic,1)
pi_lineq <- t(solve(t(A_constr)%*%A_constr,t(A_constr)%*%b_constr))
print(pi_lineq%*%P)
```

```
##              [,1]        [,2]        [,3]       [,4]       [,5]      [,6]
## [1,] 0.0003613043 0.001083913 0.003974347 0.01463282 0.05383434 0.1979948
##           [,7]
## [1,] 0.7281185
```

Niveau moyen d'équilibre

Nous venons de trouver la loi stationnaire de Z_n, qui est également la loi d'équilibre de Z_n. On déduit que le dernier état, le "6" a la plus grande probabilité. La probabilité moyenne est notée ci-dessous. Elle est plus proche avec la probabilité stationnaire de l'état "5".

```
print(mean(pi_lineq))
```

```
## [1] 0.1428571
```

Pour le cas de la variance, on trouve :

```
print((sd(pi_lineq)^2))
```

```
## [1] 0.07160512
```