Exercise2-Scalas3.R*

Source on Save | Run | Source

```r
#Program unary.R
n<-10 #number of objects
g<-2 #number of categories
T<-1000 #number of Monte Carlo steps
alpha<-rep(0,times=g) #vector of parameters
bn<-rep(0,times=g) #occupation vector
probd<-rep(0,times=g) #vector of destruction probabilities
probc<-rep(0,times=g) #vector of construction probabilities
#creation parameters (leading to symmetric Polya if all equal)
alpha<-rep(0.5,times=g)
sumalpha<-sum(alpha) #Polya parameter
#initial occupation vector
bn[1]<-10
A<-bn #vector of results
#Main cycle
for (t in 1:T){
  #destruction (an object is removed from an occupied
  #category according to a hypergeometric probability)
  probd<-bn/n
  cumprobd<-cumsum(probd)
  #pointer to category
  indexsite<-min(which((cumprobd-runif(1))>0))
  bn[indexsite]<-bn[indexsite]-1
  #creation (the object reaccommodates according to a Polya probability)
  probc<-(alpha+bn)/(sumalpha+n-1)
  cumprobc<-cumsum(probc)
  #pointer to category
  indexsite<-min(which((cumprobc-runif(1))>0))
  bn[indexsite]<-bn[indexsite]+1
  A=c(A,bn) #data update
} # end for on t
A<-matrix(A,nrow=(T+1),ncol=g,byrow=TRUE)
#Frequencies
norm<-0
```

```r
f<-rep(c(0),times=(n+1))
for(j in 1:(n+1)){
  f[j]<-length(which(A==(j-1)))
  norm<-norm+f[j]
}#end for on j

f<-f/norm


#comparison with theory (g=2 only)
fth<-rep(0,times=(n+1))
for(i in 1:(n+1)){
  alpha1<-alpha[2]
  alpha0<-alpha[1]
  fth[i]<-factorial(n)/(gamma(sumalpha+n)/gamma(sumalpha))*
    +(gamma(alpha1+i-1)/gamma(alpha1))*(gamma(alpha0+n-i+1)/gamma(alpha0))/
    +factorial(i-1)/factorial(n-i+1)
  } #end for on i
k<-c(0:n)
#Plot of results
dev.new()
plot(k,f,xlab="k",ylab=expression(pi(k)))
points(k,fth, type="p", pch=3)
legend("topright",legend=c("MC estimates","Polya distribution"),pch=c(1,3))


A<-matrix(A,nrow=T+1,ncol=g,byrow=TRUE)
t<-c(0:T)
#Plot of the first column of A
dev.new()
plot(t,A[,1],xlab="t",ylab="k",type="l",main="Time-evolution for alpha1=alpha2=0.5")
```

Exercise no 2 (Scalas3)

lunedì 21 giugno 2021 12:46

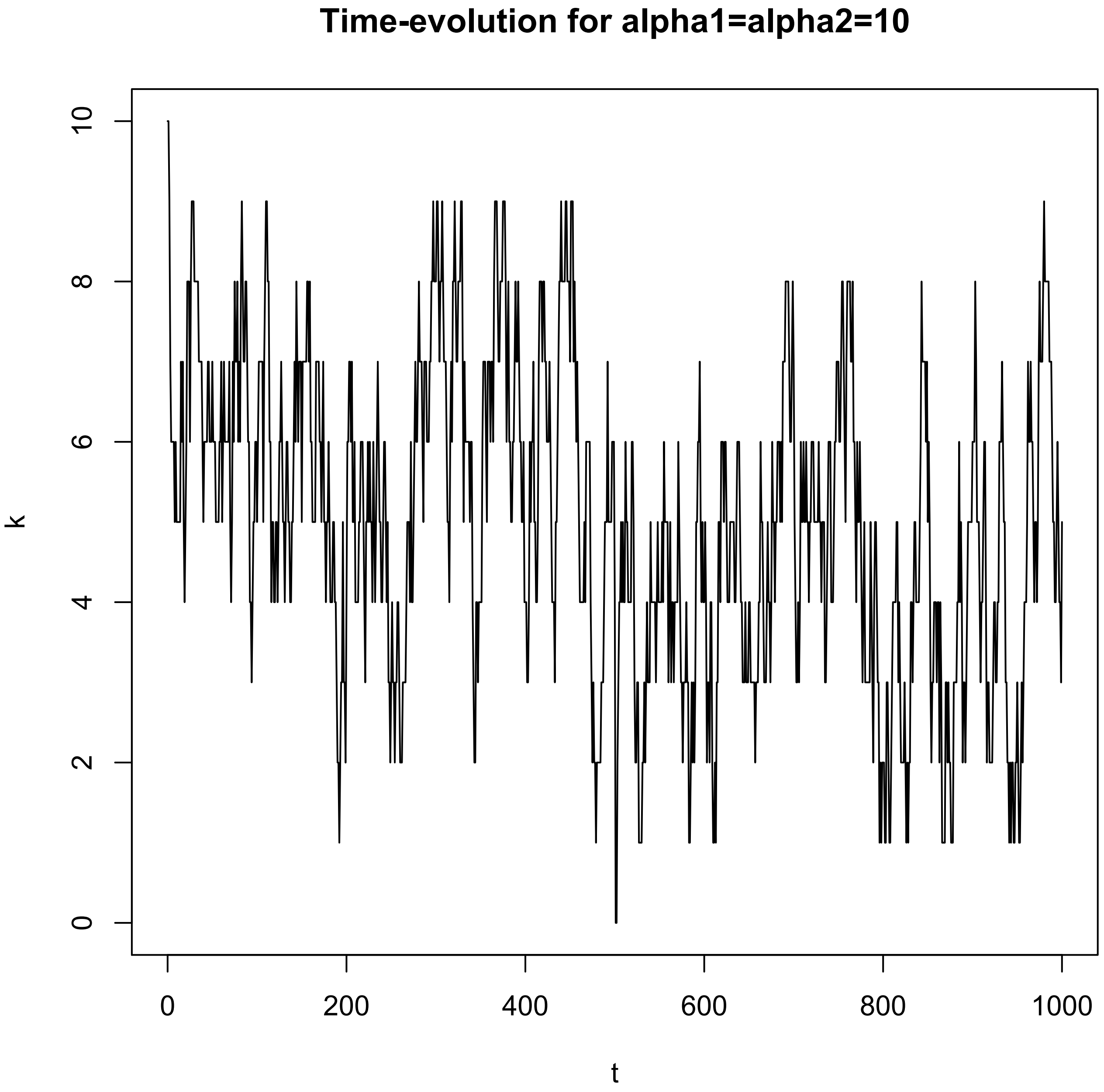
$$\pi(\underline{n}) = \frac{n!}{\alpha^{[n]}} \cdot \frac{\alpha_0^{[n_0]}}{n_0!} \cdot \frac{\alpha_1^{[n_1]}}{n_1!}$$

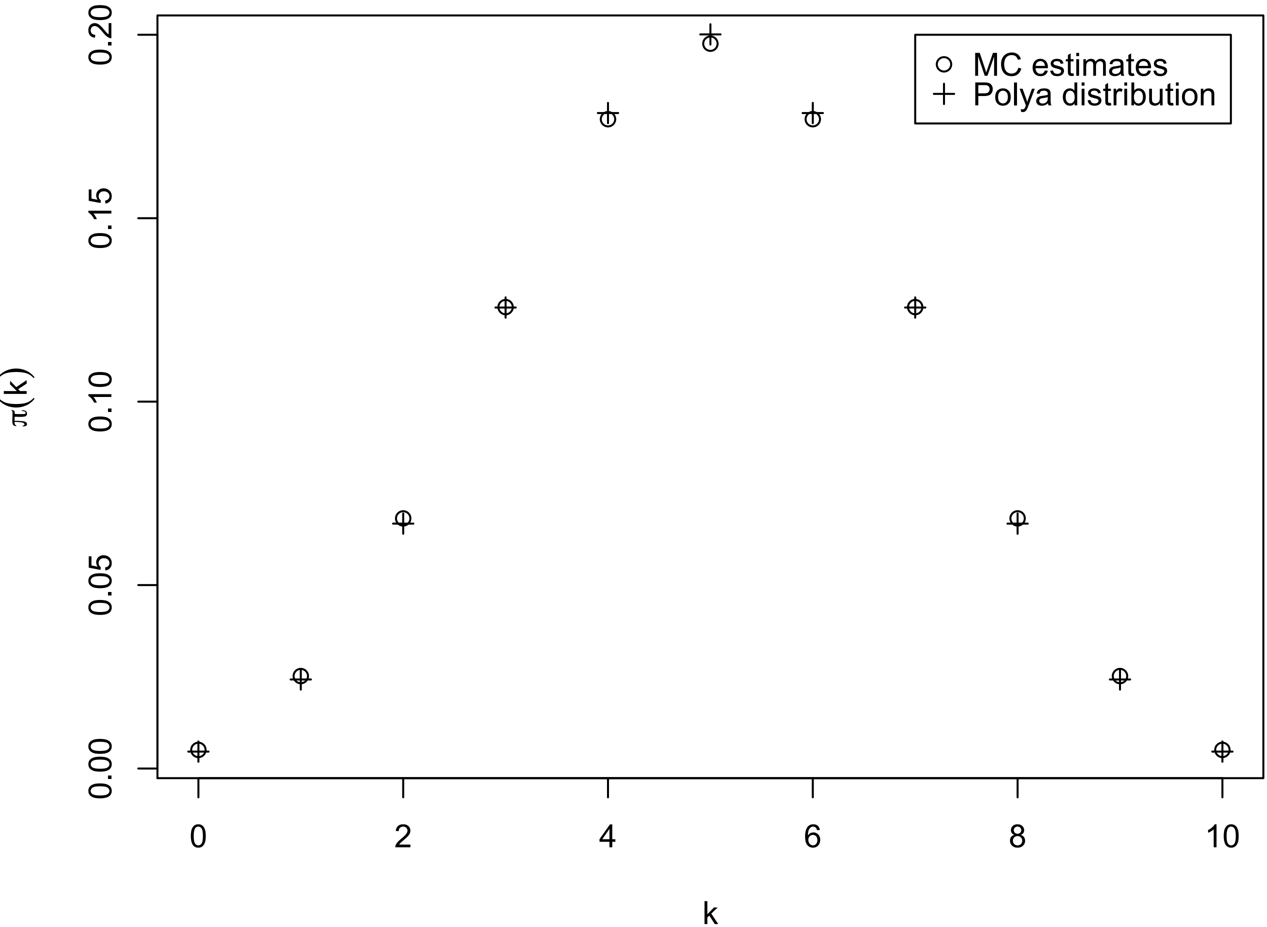
$\rightarrow n = n_0 + n_1 \Rightarrow n_0 = n - n_1$

$\rightarrow x^{[n]} = \frac{\Gamma(x+n)}{\Gamma(x)}$

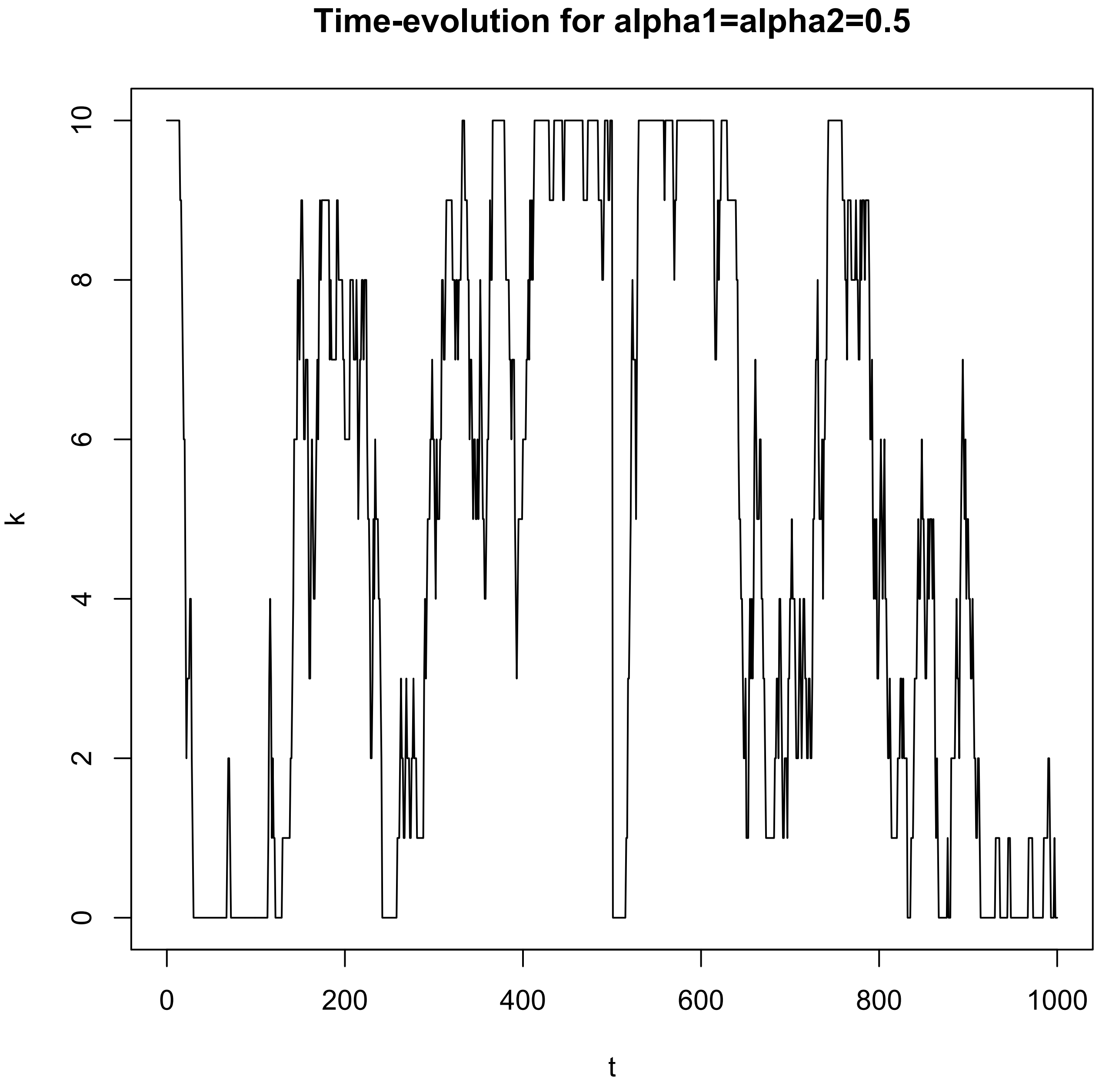
$$= \frac{n!}{\frac{\Gamma(\alpha+n)}{\Gamma(\alpha)}} \cdot \frac{\frac{\Gamma(\alpha_0+n-n_1)}{\Gamma(\alpha_0)}}{(n-n_1)!} \cdot \frac{\frac{\Gamma(\alpha_1+n_1)}{\Gamma(\alpha_1)}}{(n_1)!}$$

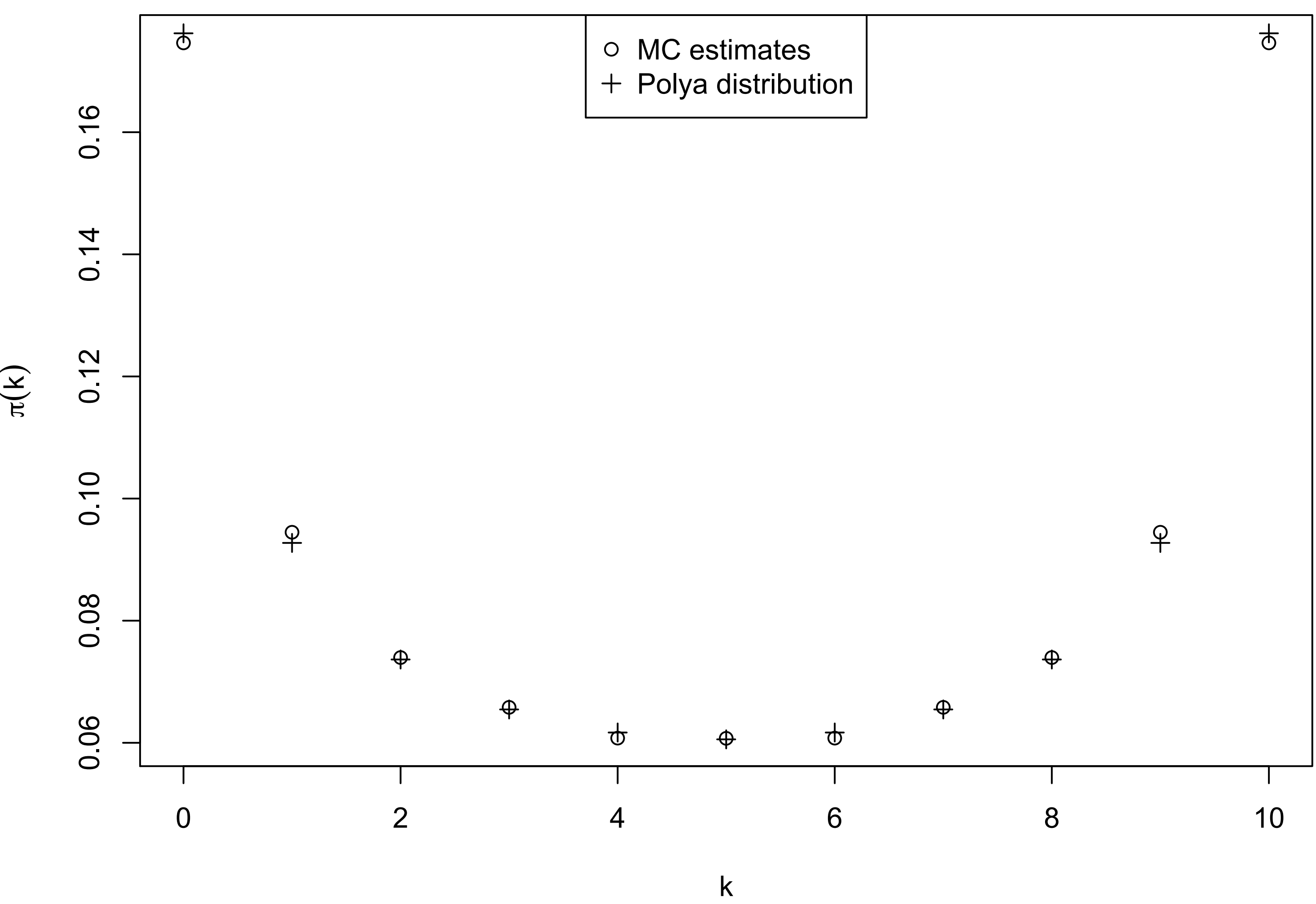
$n_1 = i - 1$ } in R we will use this notation


Time-evolution for alpha1=alpha2=10
k
t
0
2
4
6
8
10
0
200
400
600
800
1000



MC estimates
Polya distribution
0.00
0.05
0.10
0.15
0.20
π(k)
0
2
4
6
8
10
k


Time-evolution for alpha1=alpha2=0.5
k
t
0
2
4
6
8
10
0
200
400
600
800
1000



MC estimates
Polya distribution
0.06
0.08
0.10
0.12
0.14
0.16
π(k)
0
2
4
6
8
10
k