Lista 5

Guilherme NºUSP:8943160 e Leonardo NºUSP:9793436

Exercício 1

Moscas e mosquitos pousam em sua sopa de modo independente e em instantes de tempo que seguem um processo de Poisson com taxas $\lambda = 1/\text{min}$ e $\mu = 2/\text{min}$, respectivamente.

(a) O objetivo deste item e ilustrar, via simulações, o fato que os tempos que insetos (apenas moscas ou mosquitos) pousam em sua sopa, seguem um processo de Poisson, estimando o valor da taxa. Para fazer isto voce vai precisar simular os dois processos separadamente (uma vez apenas a cada) durante um pereíodo de tempo fixado (300 minutos). Em seguida e necessário combinar os tempos de ocorrências dos dois processos em apenas um, ignorando a sua origem. Por último você precisa estimar a distribuição (e a média) dos tempos entre ocorrências no processo combinado. Plote os valores encontrados em um grafico e depois compare a média de suas estimativas com o valor teorico.

(b) Plote os valores observados para o processo de Poisson ($N(t)$), simulado no item anterior, bem como uma faixa de variação de $\pm$ 3 $\mathbb{D}[N(t)]$ em torno da media teórica ($\mathbb{E}[N(t)]$), verificando se a trajetorica resultante de sua simulação ficou ou não dentro dos limites da faixa de variação.

(c) Escolha um instante de tempo aleatoriamente ($T \sim U_c[0, 240]$) e verifique se o proximo evento depois deste tempo, no processo combinado, foi uma mosca ou um mosquito. Repita esta operação 100 vezes, estimando as probabilidades e comparando com os valores teóricos.

(d) Escolha de maneira aleatoria, digamos a de número $T \sim U_d\{1, 250\}$, uma mosca que tenha pousado. Encontre a mosca que pousou logo depois desta, anotando quantos mosquitos pousaram entre as duas. Repita esta operação 100 vezes. Apresente a distribuição dos valores observados, comparando esta distribuição com a teórica.

Exercício 2

Nos casinos americanos as roletas tem casas com os inteiros entre 1 e 36, alem das casas 0 e 00. Metade das casas com os numeros diferentes de zero são vermelhos enquanto a outra metade são pretos. As casas 0 e 00 sao verdes. Uma aposta comum neste jogo é colocar um dolar no vermelho. Se um número vermelho aparece, o apostador recebe seu dolar de volta além de outro dolar. Se um número é preto ou verde aparece, ele perde o dolar. E o problema conhecido como Ruina do jogador, ou seja, um passeio aleatorio simples ($p \in [0, 1]$) em $\mathbb{Z}$, partindo de $x(S_0 = x)$, com barreiras absorventes em c e d, com $c \leq x \leq d$. Inicialmente definimos $T_c = inf_n\{S_n = c\}$ e $T_c = inf_n\{S_d = d\}$. Da teoria dos processos estocasticos, sabemos que para $p \neq \frac{1}{2}$, $P(T_d = T_c = \infty) = 0$ e

$$P_x(T_d < T_c) = \frac{1 - (q/p)^{x-c}}{1 - (q/p)^{d-c}}$$

(a) Considere o um passeio onde p seja igual a probabilidade de vencer uma rodada na roleta americana, ou seja, $p = \frac{18}{38}$ (portanto $q = \frac{20}{38}$). Imagine um jogador que começa com com $x = \{10, 20, 30, 40\}$ e decide jogar ate chegar em \$50,00 ou até perder tudo. Faça 1.000 simulações para cada uma das fortunas iniciais $x = \{10, 20, 30, 40\}$ de modo a comparar os resultados simulados com os obtidos pelas formula acima.

(b) Considere agora o passeio simetrico, ou seja $p = 12$. Faça o mesmo que foi pedido no item acima, plotando as probabilidades obtidas pela simulação (também para $x = \{10, 20, 30, 40\}$) e inferindo sobre qual a função que melhor ajusta o conjunto de pontos obtidos. Fato: $P_x(T_d < T_c)$ e linear em x (fortuna inicial).

Exercício 3

Considere um processo de ramificação onde cada indivíduo, independente dos demais, pode ter 0,1 ou 2 filhos com probabilidades $p_0 = \frac{1}{2}$, $p_1 = \frac{1}{4}$ e $p_2 = \frac{1}{4}$. Seja X_0 o numero de indivíduos no início (geração 0).

(a) Simule este processo 100 vezes (ate a extinção) usando os dados para estimar o tamanho total população para o caso $X_0 = 1$. Compare com o valor teorico.

Resolução

O resultado teórico nos diz que se $X_0 = 1$ e $\mu < 1$ então o valor esperado do número total de indivíduos (de todas as geraçãoes) desta população dado por $\mu_n = \frac{1}{1-\mu}$ e como $\mu = 0 * \frac{1}{2} + 1 * \frac{1}{4} + 2 * \frac{1}{4} = \frac{3}{4} < 1 \Rightarrow \mu_n = \frac{1}{1-\frac{3}{4}} = 4$

E o resultado simulado nos retornou um valor de 4.1 simulando 100 vezes, logo podemos concluir que simulando o processo de contagem chegamos a mesma conclusão do resultado teórico.

(b) Considere $p_0 = \frac{1}{4}$, $p_1 = \frac{1}{4}$ e $p_2 = \frac{1}{2}$. Simule o processo ate que ele se extingua ou que atinja 100 indivíduos em uma mesma geração. Embora isto não seja garantia alguma, vamos considerar para efeitos praticos que a população, neste caso, não irá se extinguir. Estime a probabilidade de sobrevivencia da população e compare com o valor teórico para o caso $X_0 = 2$.

Exercício 4

Suponhamos que numa eleição (apenas A x B) o candidato A tenha obtido $a = 100$ votos e o candidato B, $b = 50$ votos. Simule o processo de contagem dos votos 1000 vezes. Verifique a proporção de vezes que A liderou a votação durante toda a contagem e compare a proporção encontrada com o valor teórico, que é dada por $\frac{a-b}{a+b}$.

Resolução

O resultado teórico nos diz que $\frac{(a-b)}{(a+b)} = \frac{(100-50)}{(100+50)} = \frac{50}{150} = 0.3333$

E o Resultado simulado 1000 vezes obtivemos uma proporção de 0.329

Logo podemos concluir que simulando o processo de contagem chegamos a mesma conclusão do resultado teórico.

```r
# Lista de Processos Estocáticos

# Bibliotecas
library(ggplot2)

# Exercício 1

# Exercício 2

c <- 0 # Perder tudo
d <- 50 # Quanto se quer ganhar
p <- 18/38 # probabilidade de vencer a rodada
q <- 20/38 # 1-p
```

```r
x0 <- c(10,20,30,40) # Vetor de valores iniciais
Nsim <-1000 # 1000 Numero de simulações

# item a

teo <- function(x,d,q,p,c){ # Função que retorna resultados teóricos
  prob <- (1-(q/p)^(x-c))/(1-(q/p)^(d-c))
  return(prob)
}
Sn <- numeric()
simula <- function(x1,p,q){
    x <- sample(x=c(1,-1),replace = T,size = 10000,prob = c(q,p))
    x0 <- c(x1,x)
    Sn <- cumsum(x0)
    for (i in 1:length(Sn)){
      b <- Sn[i]
      if (b<=c){return((Sn[1:i]))}
      if (b>=d) {return((Sn[1:i]))}
    }
  return(Sn)
}
# Resultado simulado
simula(20,p,q)
sds <- numeric(0)
for (i in 1:1000){
  sds[i] <- length(simula(x0[1],q,p))/10000
}
mean(sds)

# Resultado teórico

for (i in 1:4) {
  print(teo(x0[i],d,q,p,c))
}

# item b
p <- 1/2

teo1 <- function(x,c,d){ # Função que retorna resultados teóricos
  prob <- (x-c)/(d-c)
  return(prob)
}
s <- numeric()
for (i in 1 :1000){
  s[i] <- length(simula(x0[2],1/2,1/2))/1000
}

# Resultado teórico

for (i in 1:4) {
  print(teo1(x0[i],d,c))
}
```

```r
# Exercício 3

p0 <- 1/2 # Probabilidade de ter 0 filho
p1 <- 1/4 # Probabilidade de ter 1 filho
p2 <- 1/4 # Probabilidade de ter 2 filhos

# item a

x0 <- 1 # número de indivíduos no início (geração 0)
n <- 1 # número de indivíduos no início (geração 0)
Nsim <- 100 # número de Simulações
pop.size <- numeric(0) # Vetor da população média
sequ <- numeric(0)
filhos <- function(x0,n){ # função que calcula o número de filhos
  for(i in 1:n){
    sequ[i] <- runif(1)
    if(sequ[i]<p0){
      x0 <- c(x0,0)
    }
    if(sequ[i]>=p0 && sequ[i]<(p0+p1)){
      x0 <- c(x0,1)
      x1 <- filhos(1,1)
      x0 <- c(x0,x1)
    }
    if(sequ[i]>=(p0+p2)){
      x0 <- c(x0,2)
      x1 <- filhos(2,1)
      x0 <- c(x0,x1)
    }
  }
  return(x0)
}

for(i in 1:Nsim){ # Resultado simulado
  pop.size[i] <- sum(filhos(x0,n))
}

mean(pop.size) # média da poupulação simualda
(1/(1-(p1+2*p2))) # média da poupulação teoŕica

# item b
x0 <- 2 # número de indivíduos no início (geração 0)
n <- 2 # número de indivíduos no início (geração 0)
p0 <- 1/4 # Probabilidade de ter 0 filho
p1 <- 1/4 # Probabilidade de ter 1 filho
p2 <- 1/2 # Probabilidade de ter 2 filhos

pop.size <- vector() # Vetor da população média

filhos1 <- function(x0,n){ # função que calcula o número de filhos
  for(i in 1:n){
    sequ[i] <- runif(1)
    if(sequ[i]<p0){
```

```r
      x0 <- c(x0,0)
    }
    if(sequ[i]>=p1 && sequ[i]<p2){
      x0 <- c(x0,1)
      x1 <- filhos(1,1)
      x0 <- c(x0,x1)
    }
    if(sequ[i]>=p2){
      x0 <- c(x0,2)
      x1 <- filhos(2,1)
      x0 <- c(x0,x1)
    }
  }
  return(x0)
}

i <-1
while (i<100){ # Resultado simulado
  pop.size[i] <- sum(filhos1(x0,n))
  if (pop.size[i] >= 100){
    i <- i+1
  }
}
length(pop.size) # média da poupulação simualda
#(n/(1-(p1+2*p2))) # média da poupulação teoŕica

# Exercício 4

a <- 100 # Quantidade de votos do candidato A
b <- 50 # Quantidade de votos do candidato B
Total <- a+b # Total de votos

Nsim <- 1000 # número de simulações
soma <- numeric(0)

lid <- function(emb){ # Verifica se durante a contagem A esteve na liderança
  s <- cumsum(emb)
  for (i in 1:length(s)){
    b <- s[i]
    if (b <=0) return(0)
  }
  return(1)
}

for( j in 1:Nsim){ # SImulações do processo de contagem
  emb <- sample(rep(c(1,-1),c(a,b)),replace=T)
  soma[j] <- lid(emb)
}

sum(soma)/Nsim # Resultado Simulado
(a-b)/Total # Resultado Teórico
```