Lista 6 - Processos de Poisson

André Vinícius - nUSP = 10737290

June 9 2019

1. Considere uma sequência de v.a.i.i.d. $\{X_i\}_{i\in\mathbb{N}}$ com distribuição exponencial de média $1(\lambda = 1)$ e suas somas parciais, $S_n = \sum_{i=1}^{n} X_i$, com $S_0 = 0$. Considere o processo de contagem $N(t) = \max\{n : S_n \leq t\}$, ou seja, um Processo de Poisson. Em particular, $N(t) \sim P(\lambda t)$ sendo $E(N(t)) = Var(N(t)) = \lambda t$.

Baseados nas instruções do enunciado, vamos construir as funções que geram um Processo de Poisson. Por isso, faremos separadamente: somas parcias e contagem.

```
seqexpo <- function(lambda, t){

    #função que gera uma sequência de somas parciais
    #a partir de v.a.i.i.d. com distribuição exponencial de média "lambda"
    #o parâmetro "t" define o valor máximo desejado para a soma

    exposum <- numeric(1) # o primeiro elemento deve ser zero

    #preencher o vetor até atingir o valor máximo desejado
    while(tail(exposum, 1) < t)
      exposum <- c(exposum, (tail(exposum, 1) + rexp(1, lambda)))

    exposum

}

contapoisson <- function(seqexpo, t){

  #função que conta a quantidade de somas parciais
  #a partir de v.a.i.i.d. com distribuição exponencial
  #até o limite "t" dado, ou seja,
  #retorna o valor de uma variável X ~ Poisson((1/lambda)*t)
  #onde lambda é a média do vetor seqexpo

  length(seqexpo[seqexpo <= t])

}
```

Definidas as funções básicas, partimos para as simulações.

a. Considere 1.000 simulações para as trajetórias de $N(t)$, para $t \in [0, 100]$. Para cada $t \in 10, 20, ..., 100$ encontre os percentis $0,005$ e $0,995$ para os valores observados de S_n. Apresente estes valores em uma tabela.

Abaixo, seguem as funções que produzem a simulação.

Para armazenar os dados das simulações, utilizamos a estrutura de dados do R "lista", uma vez que a quantidade de variáveis numa trajetória é diferente entre as simulações.

Na sequência, definimos a função que gera o vetor de observações de $N(t)$.

Por último, simulamos o processo e coletamos os resultados, apresentando-os na tabela.

```
simulaseqexpo <- function(nsimul, lambda, t){
```

```r
  #função que simula trajetórias de somas parciais
  #a partir de v.a.i.i.d. com distribuição exponencial
  #até o limite "t" dado, e portanto,
  #gera Processos de Poisson de parâmetro 1/lambda

  simula <- vector("list", nsimul)
  for(i in 1:nsimul) simula[[i]] <- seqexpo(lambda, t)
  simula

}

vetorpoisson <- function(simula, t){

  #função que conta a quantidade de somas parciais até o limite "t" dado
  #em Processos de Poisson organizados numa lista de simulações

  nsimul <- length(simula)
  poiss <- numeric(nsimul)
  for(i in 1:nsimul) poiss[i] <- contapoisson(simula[[i]], t)
  poiss

}

meusdados <- simulaseqexpo(1000, 1, 100) # Gerando os dados
observado <- matrix(nrow = 0, ncol = 4) # inicializando a tabela

for(i in seq(10, 100, 10)){

  vetor <- vetorpoisson(meusdados, i)#gerando o vetor de contagens
  dado <- quantile(vetor, probs = c(0.005, 0.995)) #calculando os quantis
  dado <- c(t = i, dado, media = mean(vetor)) #gerando a linha da tabela
  observado <- rbind(observado, dado) #preenchendo a tabela

}

rownames(observado) <- c() #apagando nomes de linha da tabela
observado #mostrando a tabela
```

```
##         t   0.5%   99.5%   media
##  [1,]  10  3.000  20.005  10.876
##  [2,]  20 10.000  33.005  20.914
##  [3,]  30 18.000  46.000  30.927
##  [4,]  40 25.000  58.000  41.013
##  [5,]  50 34.000  70.000  51.182
##  [6,]  60 42.000  82.000  61.219
##  [7,]  70 50.000  93.005  71.365
##  [8,]  80 59.990 105.005  81.386
##  [9,]  90 68.995 116.005  91.366
## [10,] 100 78.000 127.005 101.295
```

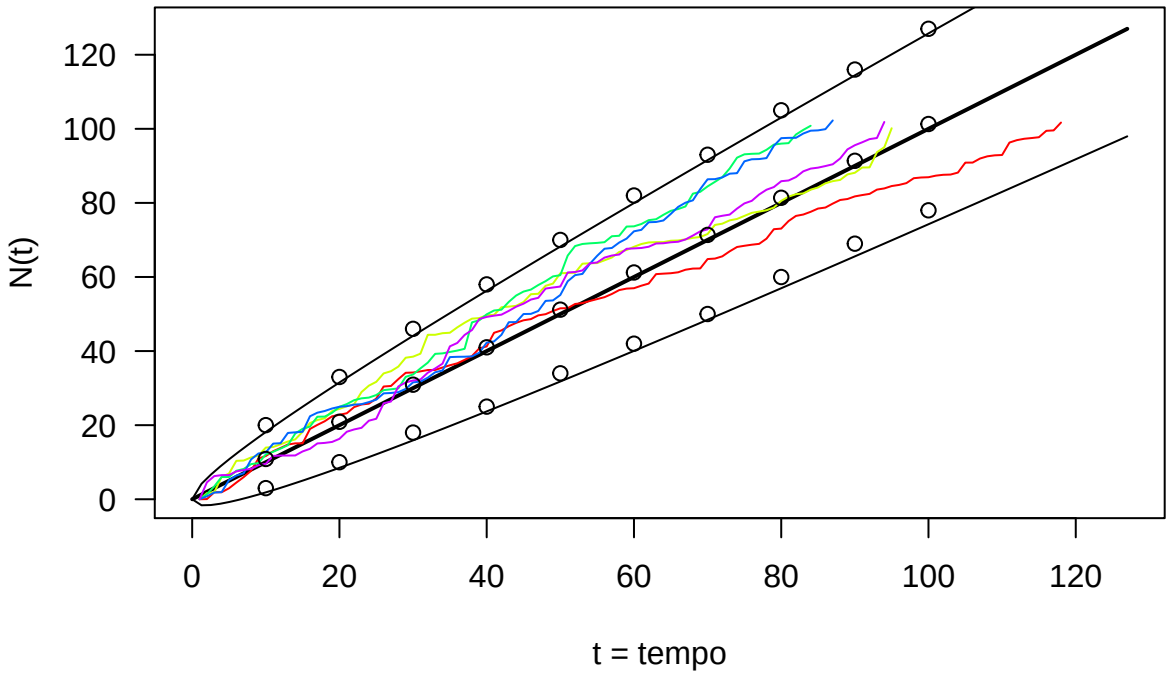
b. Apresente em um só gráfico:

i) a trajetória de 5 simulacões,
ii) os valores da tabela do item a,
iii) os valores estimados para os mesmos percentis pelo Teorema Central do Limite,

iv) a função $E(N(t))$.

Para gerar o gráfico, organizamos o código em uma função. Apesar desta não servir para propósito geral, uma vez que só constrói corretamente o gráfico para $\lambda = 1$, julgamos ser mais conveniente para organizar o raciocínio. Os comentários explicam os detalhes de cada passo.

```
plotdata <- function(simula,# lista de trajetórias a serem plotadas
                     obs,# tabela com pontos a serem plotados
                     qtd# quantidade de trajetórias a serem plotadas
                     ){

  #função que plota simulações sorteadas de uma lista de
  #trajetórias de somas parciais de v.a.i.i.d. de distribuição exponencial
  #além dos quantis observados nessas simulações, em 0,5% e 99,5%

  sorteio <- sample(1:length(simula), qtd) #sortear as trajetórias a serem plotadas
  processo <- sorteio[1] # obter a posição da trajetória na lista
  t <- simula[[processo]] # obter os dados da trajetória
  N <- length(t) # contar o tamanho da trajetória
  min <- t[1]+qnorm(0.005)*sqrt(t[1]) # calcular o limite inferior do gráfico
  max <- tail(t, 1)+qnorm(0.995)*sqrt(tail(t, 1)) # e o limite superior
  cores <- rainbow(qtd) # gerar um vetor de cores diferentes para as trajetórias
  plot(0:max, #plotar E(N(t))
       0:max,
       main = "Processo de Poisson", #título do gráfico
       xlab = "t = tempo", #label dos xzes
       ylab = "N(t)", #label dos yzes
       las = 1, #orientação horizontal dos números nos eixos
       type = "l", #gráfico de linhas
       lwd = 2, #espessura da linha desta plotagem
       ylim = c(min, max)) #limites superior e inferior do gráfico

  curve(x+qnorm(0.005)*sqrt(x), add = TRUE)#limite de 0,5% dos dados
  curve(x+qnorm(0.995)*sqrt(x), add = TRUE)#limite de 99,5% dos dados

  for(i in 1:qtd){

    processo <- sorteio[i]
    lines(simula[[processo]], col = cores[i]) #plotando cada uma das linhas

  }

  for(i in seq(10, 100, 10)){

    points(c(i, i, i), obs[obs[,"t"] == i,2:4]) #plotando os pontos

  }

}

plotdata(meusdados, observado, 5)#plotando 5 simulações
```



2. Além do Processo de Poisson ($N(t)$) definido no item 1, considere agora uma sequência de v.a.i.i.d.$\{Y_i\}_{i\in\mathbb{N}}$ com distribuição $U[0,1]$, e o Processo de Poisson Composto (ou com Recompensa) $R(t) = \sum_{i=1}^{n} Y_i$. Em particular, $E(R(t)) = \lambda t E(Y_1)$ e $Var(R(t)) = \lambda t E(Y_1^2)$.

Para o Processo de Poisson com Recompensa, utilizamos como base os dados já gerados no Processo Simples, uma vez que basta gerar outra lista com dados das respectivas recompensas. Portanto, as funções aqui contém apenas algumas alterações em relação às originais, aproveitando sua estrutura.

a. Considere 1.000 simulações para as trajetórias de $R(t)$, para $t \in [0, 100]$. Para cada $T \in \{10, 20, ..., 100\}$, encontre os percentis $0,005$ e $0,995$ para os valores observados de $R(t)$. Apresente estes valores em uma tabela.

Seguem as funções alteradas que produzem esta outra simulação.

A função que simula as somas parciais das recompensas recebe como entrada a lista de simulações de somas parcias de variável exponencial. A distribuição das recompensas está fixada em $\sim U(0, 1)$.

Depois, foi definida a função que retorna a soma parcial da recompensa para dado t. Para encontrar o valor acumulado desta recompensa, buscamos sua posição através da soma parcial de exponencias, pois é a variável que está atrelada à t.

Em seguida, está a função que gera o vetor de observações de $R(t)$. O processo se assemelha ao original, aterada a função de contagem.

Por último, simulamos o processo e coletamos os resultados, apresentando-os na tabela.

```
recomppoiss <- function(simula){

  #função que simula Recompensas de distribuição Uniforme(0,1)
  #e as relaciona a v.a.i.i.d. com distribuição exponencial
  #gerando Processos de Poisson Compostos

  nsimul <- length(simula)
```

```
  simularcp <- vector("list", nsimul)

  for(v in 1:nsimul){

  nvetor <- length(simula[[v]])
  unifsum <- numeric(nvetor)
  for(u in 2:nvetor) unifsum[u] <- unifsum[u-1] + runif(1)

  simularcp[[v]] <- unifsum

  }

  simularcp

}

contapoissrcp <- function(seqexpo, sequnif, t){

  #função que retorna o valor acumulado para
  #a recompensa de um Processo de Poisson Composta
  #dado um intervalo de tempo "t"
  sequnif[length(seqexpo[seqexpo <= t])]

}

vetorpoissrcp <- function(simula, simularcp, t){

  #função que coleta as somas parciais de recompensas
  #até o intervalo de tempo limite "t" dado
  #em Processos de Poisson Compostos simulados

  nsimul <- length(simula)
  poiss <- numeric(nsimul)
  for(i in 1:nsimul) poiss[i] <- contapoissrcp(simula[[i]], simularcp[[i]], t)
  poiss

}

meusdadosrcp <- recomppoiss(meusdados) #Gerando os dados
observadorcp <- matrix(nrow = 0, ncol = 4) #inicializando a tabela

for(i in seq(10, 100, 10)){

  vetor <- vetorpoissrcp(meusdados, meusdadosrcp, i) #coletando recompensas
  dado <- quantile(vetor, probs = c(0.005, 0.995)) #calculando os quantis
  dado <- c(t = i, dado, media = mean(vetor)) #montando a linha da tabela
  observadorcp <- rbind(observadorcp, dado) #acrescentando a linha à tabela

}

rownames(observadorcp) <- c() #apagando os nomes das linhas da tabela
observadorcp #mostrando a tabela
```

```
##         t      0.5%    99.5%     media
```

```
##  [1,]  10  1.030818 10.24762  4.898793
##  [2,]  20  3.908168 16.81440  9.943430
##  [3,]  30  7.465931 23.76526 14.934619
##  [4,]  40 11.420192 30.02862 19.987244
##  [5,]  50 15.230349 35.13626 25.002534
##  [6,]  60 19.196750 42.53419 30.005856
##  [7,]  70 22.792650 47.88270 35.019929
##  [8,]  80 27.024211 53.55232 40.035692
##  [9,]  90 31.547433 59.84757 45.034235
## [10,] 100 36.014501 66.27872 49.989738
```

b. Apresente em um só gráfico

i) a trajetória de 5 simulações,
ii) os valores da tabela do item a,
iii) os valores estimados para os mesmos percentis pelo TCL,
iv) a função $E(R(t))$.

Novamente organizamos em uma função, que desta vez constrói corretamente o gráfico apenas para $X \sim Poisson(\lambda = 1)$ e $Y \sim U(0, 1)$. Gostaríamos de evidenciar a origem dos valores $1/2 = E(Y)$ e $1/3 = E(Y^2)$. Os comentários no código explicam os detalhes de cada passo.

```r
plotdatarcp <- function(simula, simularcp, obs, qtd){

  #função que plota simulações sorteadas de uma lista de
  #trajetórias de Processos de Poisson Compostos,
  #além dos quantis observados nessas simulações, em 0,5% e 99,5%

  sorteio <- sample(1:length(simula), qtd) #sorteando os dados a serem plotados
  processo <- sorteio[1] #obtendo o número de uma das simulações sorteadas
  t <- simula[[processo]] #dados da sequência exponencial (eixo x)
  R <- simularcp[[processo]] #dados da sequência uniforme (eixo y)
  min <- t[1]*1/2+qnorm(0.005)*sqrt(t[1]*1/3) #limites do gráfico
  max <- tail(t, 1)*1/2+qnorm(0.995)*sqrt(tail(t, 1)*1/3)
  cores <- rainbow(qtd)
  plot(0:max, #plotando E(R(t))
       (0:max)*1/2,
       main = "Processo de Poisson com Recompensa",
       xlab = "t = tempo", #label dos xzes
       ylab = "R(t)", #label dos yzes
       las = 1,
       type = "l", #gráfico de linhas
       lwd = 2, #espessura da plotagem dessa informação
       ylim = c(min, max)) #limites superior e inferior do gráfico

  #limite de 0,5% dos dados
  curve(x*1/2+qnorm(0.005)*sqrt(x*1/3), add = TRUE)
  #limite de 99,5% dos dados
  curve(x*1/2+qnorm(0.995)*sqrt(x*1/3), add = TRUE)

  for(i in 1:qtd){

    #plotando cada um dos trajetos sorteados
    processo <- sorteio[i]
    lines(simula[[processo]], simularcp[[processo]], col = cores[i])
```

```
  }

  for(i in seq(10, 100, 10)){

    #plotando os pontos da tabela
    points(c(i, i, i), obs[obs[,"t"] == i,2:4])

  }

}

plotdatarcp(meusdados, meusdadosrcp, observadorcp, 5)
```