Project 2: Nonhomogeneous Poisson Processes

Max Barshay

due by 4pm Friday, May 15

Recall from class that a *nonhomogeneous* Poisson process $X(t)$ with intensity function $\lambda(t)$ is such that the number of events in any interval $(t_1, t_2]$ has a Poisson distribution with mean

$$\int_{t_1}^{t_2} \lambda(u)\,du$$

Define a *mean function* $\mu(t)$ by

$$\mu(t) = \int_0^t \lambda(u)\,du$$

Then $X(t)$ is Poisson with mean $\mu(t)$, and any increment $X(t_2) - X(t_1)$ is Poisson with mean $\mu(t_2) - \mu(t_1)$.

In this project, you will explore theoretical properties of $X(t)$ and learn how to simulate $X(t)$.

First, some math. Questions 1 and 2 are pretty easy.

1. What's $\mu'(t)$, the first derivative of the mean function?

$$\mu'(t) = \lambda(t)$$

2. In the homogeneous case where $\lambda(t) \equiv \lambda$, a constant, what's $\mu(t)$?

In this case $\mu(t) = \lambda t$.

3. Show that the pdf of Y_1 (aka T_1), the first arrival time, is

$$f_1(t) = \lambda(t)e^{-\mu(t)}$$

$$F_1(t) = P(T_1 \leq t) = 1 - P(T_1 > t) = 1 - P(no\ events\ in\ time\ interval\ (0, t])$$

$$= 1 - P(X(t) = 0)\ where\ X(t) \sim Poisson(\int_0^t \lambda(u)du = \mu(t))$$

$$F_1(t) = 1 - \frac{e^{-\mu(t)} \cdot \mu(t)^0}{0!} = 1 - e^{-\mu(t)}$$

$$f_1(t) = \frac{d}{dt}F_1(t) = \frac{d}{dt}[1 - e^{-\mu(t)}] = -e^{-\mu(t)} \cdot -\lambda(t) = \lambda(t)e^{-\mu(t)}$$

FYI, the pdf of Y_n, the nth arrival time, is

$$f_n(y) = \frac{\lambda(y)[\mu(y)]^{n-1}}{(n-1)!} e^{-\mu(y)}$$

You don't have to show this. But you can't just cite it to "solve" question 3!

4. Let $X(t)$ have intensity function $\lambda(t) = 3/(t+1)$. Determine the pdf of the first arrival time (simplify as much as possible), and then determine its mean and standard deviation.

Recall that: $\lambda(t) = \frac{3}{t+1}$ and $f_1(t) = \lambda(t)e^{-\mu(t)}$.

$$\mu(t) = \int_0^t \frac{3}{u+1} du = 3ln(t+1)$$

Plugging $\lambda(t)$ *and* $\mu(t)$ into the PDF for Y_1:

$$\frac{3}{t+1} e^{-3ln(t+1)} = \frac{3}{t+1} \cdot \frac{1}{e^{3ln(t+1)}} = \frac{3}{t+1} \cdot \frac{1}{(t+1)^3} = \frac{3}{(t+1)^4}$$

$$E(Y_1) = \int_0^\infty t \cdot \frac{1}{(t+1)^4} = \frac{1}{2}$$

$$Var(Y_1) = E(Y_1^2) - (E(Y_1))^2 = \int_0^\infty t^2 \cdot \frac{1}{(t+1)^4} - (\frac{1}{2})^2 = 1 - \frac{1}{4} = \frac{3}{4}$$

$$SD(Y_1) = \sqrt{Var(Y_1)} = \sqrt{\frac{3}{4}} = \frac{\sqrt{3}}{2} \approx .866$$

You're welcome to use a computer algebra system for your calculus.

FYI, no explicit formulas exist for the pdfs of $T_2, T_3, \ldots$. In particular, they don't have the same pdf as T_1 because of the lack of time-homogeneity. At best, recursive formulas can be developed.

On to the simulation! Because many properties of a nonhomogeneous Poisson process are mathematically intractable, having simulation tools is especially important. Here, we'll develop two different simulation methods.

Method 1: Suppose events are occurring according to a (regular) Poisson process with constant rate λ. Then, suppose there exists a function $p(t)$ such that an event that occurs at time t is "counted" with probability $p(t)$. It can be shown that the thinned process of "counted" events constitutes a nonhomogeneous Poisson process with intensity function $\lambda(t) = \lambda \cdot p(t)$.

We'll use this fact to reverse-engineer our $X(t)$ from a regular Poisson process.

5. How should we define $p(t)$ in order to exploit the method described? (This should be easy, assuming we know $\lambda(t)$ and have selected an appropriate constant λ.)

We should define $p(t) = \frac{\lambda(t)}{\lambda}$. This simply follows from the definition above: $\lambda(t) = \lambda \cdot p(t)$.

6. The choice of the constant λ isn't specified. But why must we require that $\lambda \geq \lambda(t)$ for all $t > 0$?

We must require that $\lambda \geq \lambda(t)$ for all $t > 0$ because if that was not the case p(t) (which is a probability for any fixed value of t) would be above 1 for some values of t, which goes against the definition of a probability.

7. Write a function to simulate a single realization of $X(t)$ with intensity function $\lambda(t) = 3/(t+1)$. Do this by simulating a regular Poisson process and then using the thinning method. The output of your function should be a list of arrival times up to some specified time `tmax`.

I am going to let $\lambda = 3$. I am going to then have $p(t) = \frac{1}{t+1}$ as the product $\lambda \cdot p(t) = \lambda(t) = \frac{3}{t+1}$.

```
nonhomogenouspp <- function(lambda, tmax){
  X.max <- rpois(1, lambda * tmax)
  Y <- sort(runif(X.max, 0, tmax))
  newY <- NULL
  for (i in 1:length(Y)){
    P.t <- 1/(Y[i]+1)
    u <- runif(1)
    if (u < P.t){
      newY <- c(newY, Y[i])
    }
  }
  return (newY)
}
```

8. Iterate your function many times, storing the value of the first arrival time from each iteration. Investigate the values of this variable, and verify that they comport with your earlier theoretical answers.

```
N <- 10000
results1 <- rep(NA, N)
while (any(is.na(results1))){
  for (i in 1:N){
    results1[i] <- nonhomogenouspp(3, 100)[1]
  }
}
```

```
mean(results1)
```

```
## [1] 0.4878818
```

This mean is close to the mean that I got of $\frac{1}{2}$ above.

```
sd(results1)
```

```
## [1] 0.7797434
```

This standard deviation is close to the standard deviation of $\frac{\sqrt{3}}{2} \approx .866$ that I got above. The fact that I am throwing away arrays of first arrivals where any first arrival occurs after tmax likely contributes to a slightly lower standard deviation estimate since those values would be outliers and thus increase the standard deviation.

9. Iterate your function many times, storing the value of the *third* arrival time Y_3 from each iteration. Investigate the values of this variable, and verify that their distribution matches $f_3(y)$ above.

```r
N <- 10000
results3 <- rep(NA, N)
while (any(is.na(results3))){
for (i in 1:N){
  results3[i] <- nonhomogenouspp(lambda = 3, 100)[3]
  }
}
```

```r
mean(results3)
```

```
## [1] 2.368221
```

```r
sd(results3)
```

```
## [1] 3.390828
```

By plugging in to the generic function above:

$$f_3(t) = \frac{\lambda(t)[\mu(t)]^2}{2}e^{-\mu(t)} = \frac{\frac{3}{t+1}(3ln(t+1))^2}{2}e^{-3ln(t+1)}$$

$$= \frac{27ln^2(t+1)}{2(t+1)^4}$$

$$E(Y_3) = \int_0^\infty t\frac{27ln^2(t+1)}{2(t+1)^4}dt = 2.375$$

This is close to the simulated mean value above.

$$Var(Y_3) = E(Y_3^2) - (E(Y_3))^2 = \int_0^\infty t^2\frac{27ln^2(t+1)}{2(t+1)^4}dt - (2.375)^2 = 21.25 - 5.641 \approx 15.609$$

$$SD(Y_3) = \sqrt{Var(Y_3)} = \sqrt{15.609} = 3.951$$

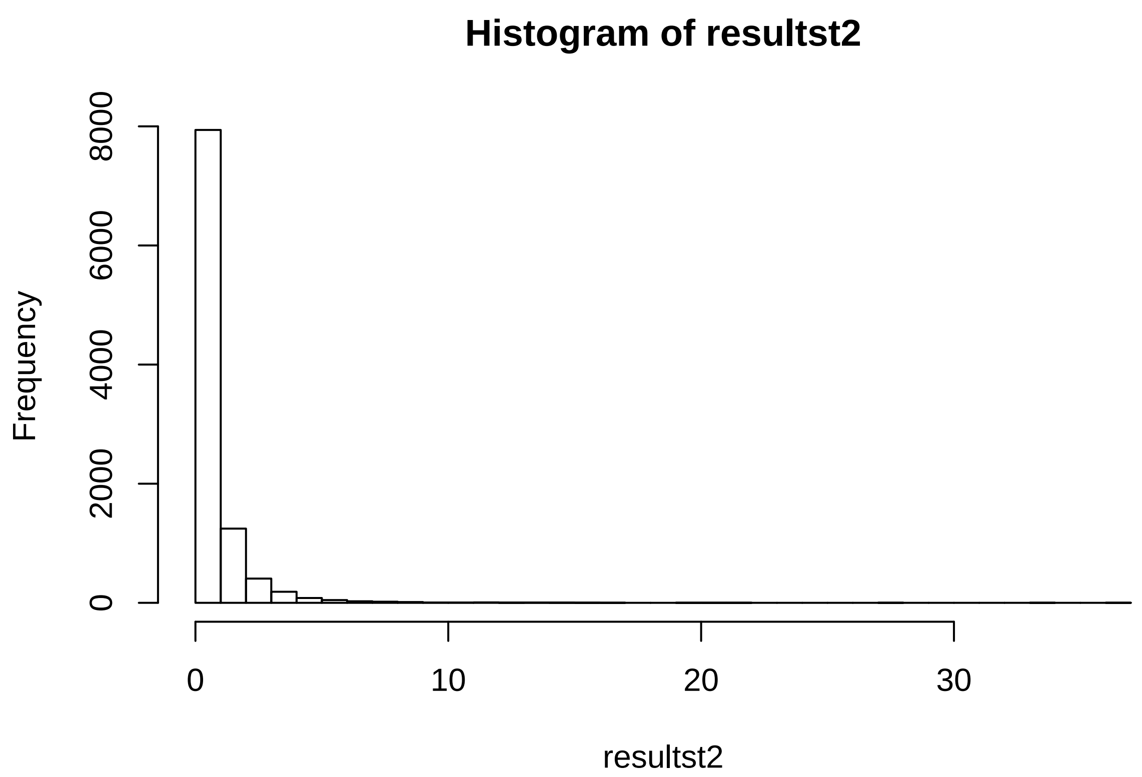
This is somewhat close to the simulated standard deviation above. The fact that I am throwing away arrays of third arrivals where any third arrival occurs after tmax likely contributes to a slightly lower standard deviation estimate since those values would be outliers and thus increase the standard deviation.

10. Iterate your function many times, storing the second *interarrival* time T_2 from each iteration. Make a graph of these values and report the estimated mean and standard deviation of T_2.

```r
N <- 10000
resultst2 <- rep(NA, N)
while (any(is.na(resultst2))){
  for (i in 1:N){
    run <- nonhomogenouspp(lambda = 3, tmax = 500)
    resultst2[i] <- (run[2] - run[1])
  }
}
```

The graph is not very good (informative) because there are values that are very large that affect the scale of the graph.

```r
hist(resultst2, breaks = 50)
```




```r
mean(resultst2)
```

```
## [1] 0.7477043
```

```r
sd(resultst2)
```

```
## [1] 1.341627
```

[Hint: How can you determine the second interarrival time from the vector of arrival times?]

Method 2: Conditional on $Y_n = s$, the cdf of the next interarrival time T_{n+1} is

$$
\begin{aligned}
F(t) &= 1 - P(0 \text{ events in } (s, s+t]) = 1 - P(X(s+t) - X(s) = 0) \\
&= 1 - \frac{e^{-[\mu(s+t)-\mu(s)]}[\mu(s+t) - \mu(s)]^0}{0!} = 1 - e^{-[\mu(s+t)-\mu(s)]}
\end{aligned}
$$

How does that help? We can generate the T's and Y's sequentially. Having generated Y_n, use the formula above and the inverse cdf method to generate T_{n+1}. Then, the next arrival time is $Y_{n+1} = Y_n + T_{n+1}$. (For the first iteration, define $Y_0 = 0$.)

11. For the intensity function $\lambda(t) = 3/(t+1)$, calculate the conditional cdf $F(t) = P(T_{n+1} \leq t | Y_n = s)$ as defined above.

$$\mu(s+t) = \int_0^{s+t} \frac{3}{u+1} du = 3 \cdot ln(s+t+1)$$

$$\mu(s) = \int_0^{s} \frac{3}{1+u} du = 3 \cdot ln(s+1)$$

Therefore the conditional CDF with this $\lambda(t)$ is:

$$1 - e^{-[3 \cdot ln(s+t+1) - 3 \cdot ln(s+1)]} = 1 - (\frac{s+1}{s+t+1})^3$$

12. Determine the inverse of this cdf. (Remember, s is a fixed value in the expression, so treat it like a constant.)

$$u = F(t) = 1 - (\frac{s+1}{s+t+1})^3 \Rightarrow t = F^{-1}(u) = \frac{s + 1 - s(1-u)^{1/3} - (1-u)^{1/3}}{(1-u)^{1/3}}$$

The work to get there:

$$u = 1 - (\frac{s+1}{s+t+1})^3 \Rightarrow 1 - u = (\frac{s+1}{s+t+1})^3 \Rightarrow (1-u)^{1/3} = \frac{s+1}{s+t+1}$$
$$= (1-u)^{1/3} \cdot (s+t+1) = s+1 \Rightarrow s(1-3)^{1/3} + t(1-u)^{1/3} + (1-u)^{1/3} = s+1$$

$$t(1-u)^{1/3} = s + 1 - s(1-u)^{1/3} - (1-u)^{1/3} \Rightarrow t = \frac{s + 1 - s(1-u)^{1/3} - (1-u)^{1/3}}{(1-u)^{1/3}}$$

13. Write a function to simulate a single realization of $X(t)$ with intensity function $\lambda(t) = 3/(t+1)$. Do this by iteratively generating $Y_1, Y_2, \ldots$ using Method 2. The output of your function should be a list of arrival times up to some time `tmax`.

```r
iterativenonhmgpp <- function(tmax){
  Y <- NULL
  Y[1] <- 0
  i <- 1
  while (Y[i] < tmax){
    u <- runif(1)
     T.n1 <- (Y[i] + 1 - Y[i]*(1-u)^(1/3) - (1-u)^(1/3)) / (1-u)^(1/3)
    i <- i + 1
    Y[i] <- Y[i-1] + T.n1
  }
  return (Y[-c(1, length(Y))])
}
```

14. Iterate your function many times, storing the value of the third arrival time Y_3 from each iteration. Investigate the values of this variable, and verify that their distribution matches $f_3(y)$ above.

```r
N <- 10000
resultsiter <- rep(NA, N)
while (any(is.na(resultsiter))){
for (i in 1:N){
  resultsiter[i] <- iterativenonhmgpp(100)[3]
  }
}
```

```r
mean(resultsiter)
```

```
## [1] 2.384944
```

```r
sd(resultsiter)
```

```
## [1] 3.526247
```

These results are close to the theoretical results of $E(Y_3) = 2.375$ and $SD(Y_3) \approx 3.951$. The fact that I am throwing away arrays of third arrivals where any third arrival occurs after tmax likely contributes to a slightly lower standard deviation estimate since those values would be outliers and thus increase the standard deviation.

15. Iterate your function many times, storing the second interarrival time T_2 from each iteration. How do these simulated values compare with those you found using Method 1?

```r
N <- 10000
resultst2iter <- rep(NA, N)
while (any(is.na(resultst2iter))){
  for (i in 1:N){
    run <- iterativenonhmgpp(100)
    resultst2iter[i] <- (run[2] - run[1])
  }
}
```

```r
mean(resultst2iter)
```

```
## [1] 0.7651632
```

```r
sd(resultst2iter)
```

```
## [1] 1.458169
```

These values are very close to 0.7477043 and 1.3416269 that I found above.