2_proc_non_pos

Frances Lin

4/9/2021

```
library(tidyverse)
library(ggplot2)
```

```
set.seed(1) # for reproducibility

t_max <- 10
t <- 0

lmbda <- function(t){
  t
}

X <- numeric(0) # vector of t

lmbda_star_f <- function() {
  # not sure
  max(sapply(seq(1, t_max, length.out = 1000), lmbda)) * 2

}

lmbda_f <- function(t_upper) {
  # this lmbda_fun function takes in t_upper and integrate f =
  # from t = 0 to t = t_upper
  integrate(f = lmbda, lower = 0, upper = t_upper)
}

while(t <= t_max){
  # thinning
  u       <- runif(1)
  t       <-  t - log(u)/lmbda_star_f()
  if(runif(1) < lmbda(t)/lmbda_star_f()) {
    X <- c(X,t)
  }
}
```

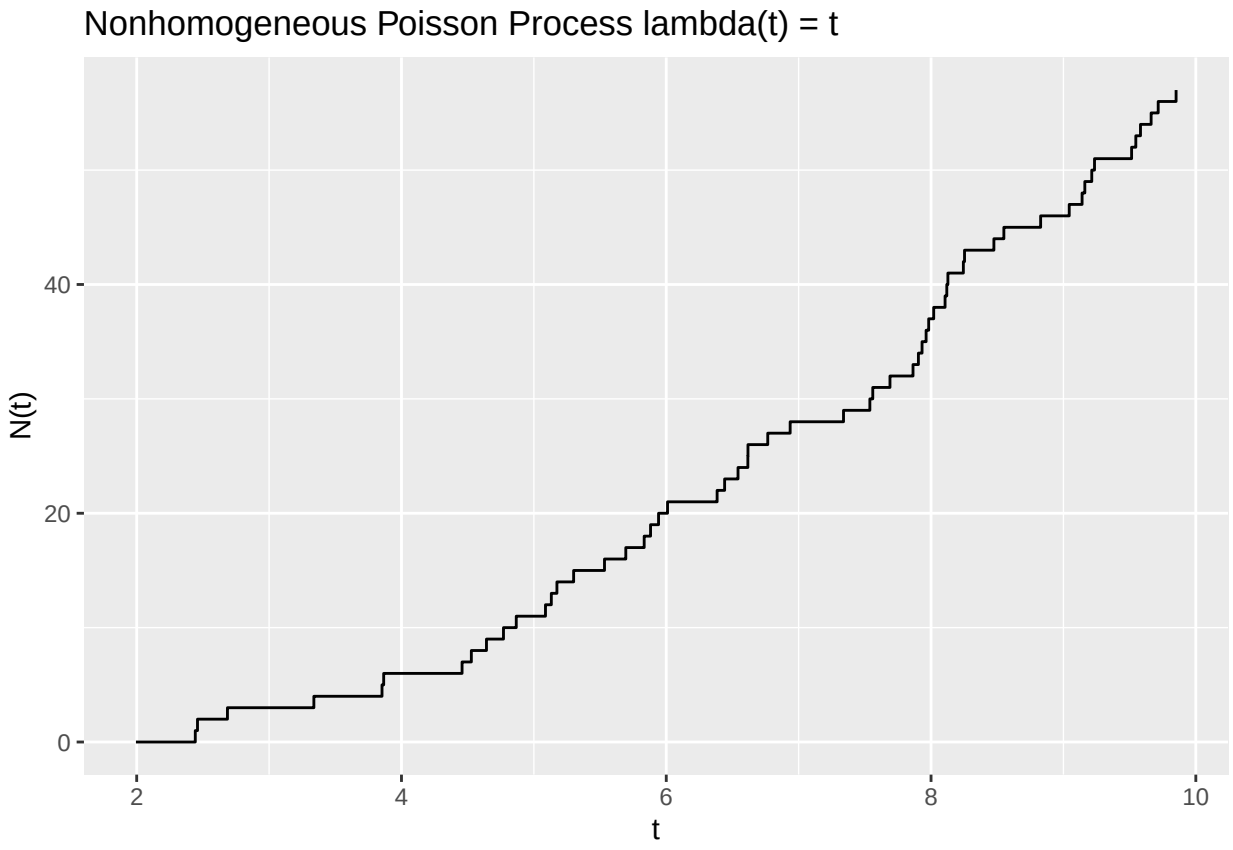
```
# X
```

```
# length(X)
```

```
# N <- 0:length(X)
# length(N)
```

```
# # Plot using base R
# plot(stepfun(x=X, y=N))
```

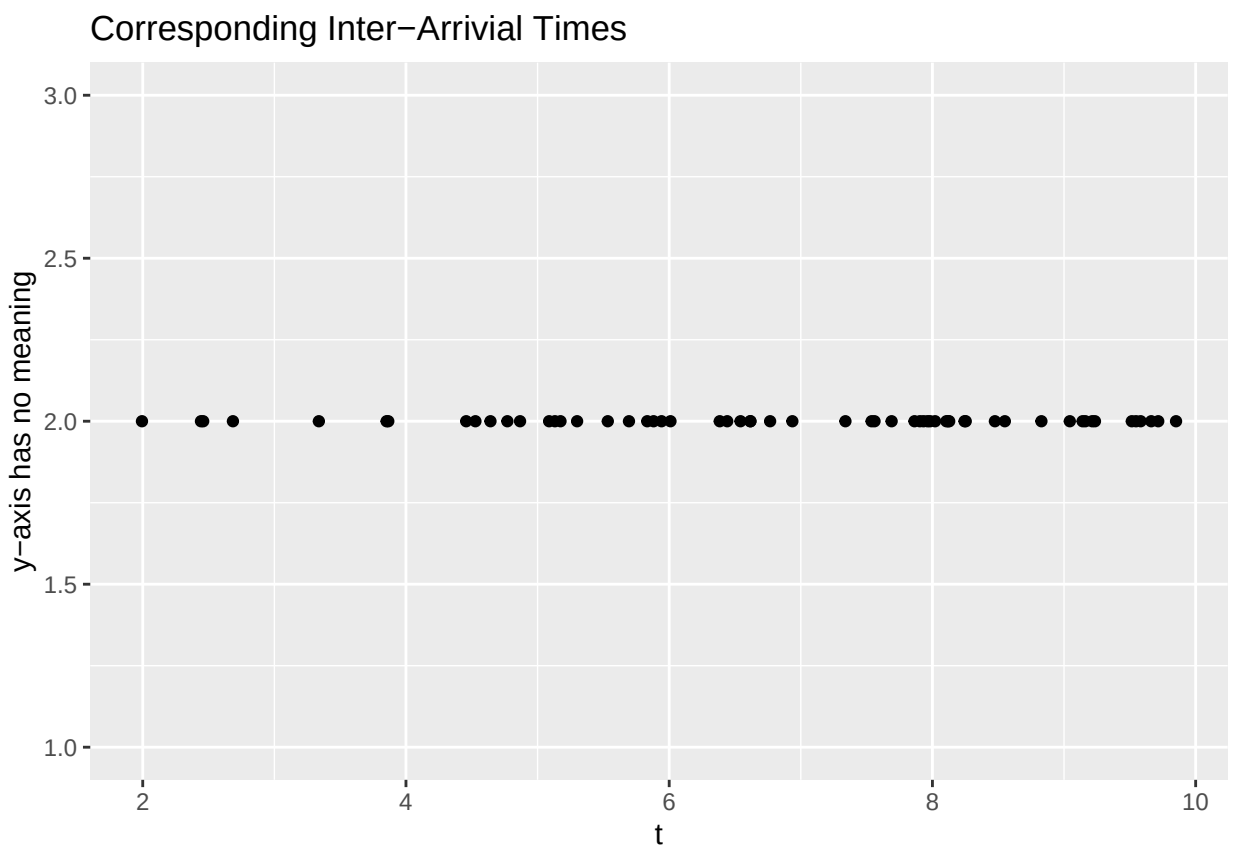
```r
# Plot using ggplot2
# Create a df
df_NPP = tibble(
  x = X,
  y = 0:(length(X) - 1)
)

p_NPP <- ggplot(data=df_NPP, mapping=aes(x=x, y=y)) +
  geom_step() +
  labs(title = "Nonhomogeneous Poisson Process lambda(t) = t",
       x = "t",
       y = "N(t)")
p_NPP
```




```r
# Plot time plot
df_NPP <- df_NPP %>% add_column(
  t_fix = rep(2, length(df_NPP$x))
)
```

```r
p_NPP_time <- ggplot(data=df_NPP, mapping=aes(x=x, y=t_fix)) +
  geom_point() +
  ylim(c(1,3)) +
  labs(title = "Corresponding Inter-Arrivial Times",
       x = "t",
       y = "y-axis has no meaning")
```

```
p_NPP_time
```



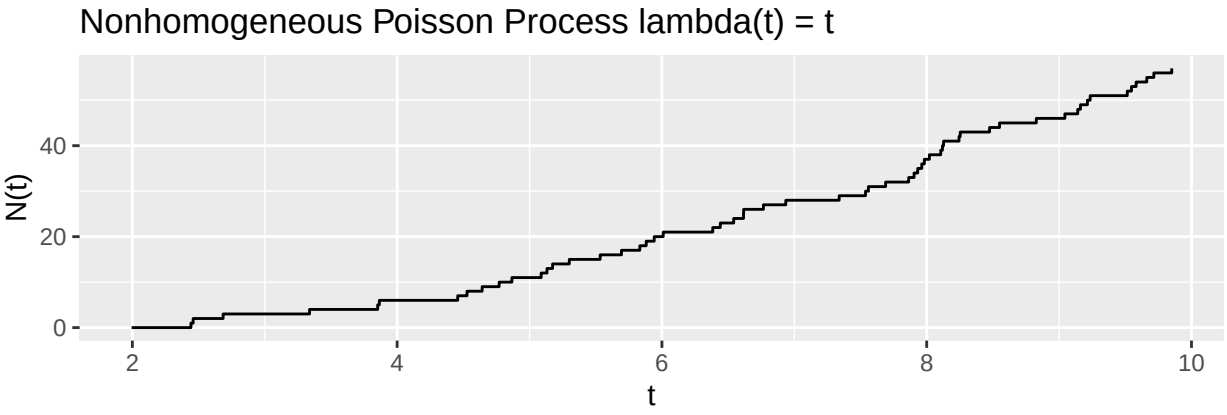

```
require(gridExtra)
```

```
## Loading required package: gridExtra
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
grid.arrange(p_NPP, p_NPP_time)
```


Nonhomogeneous Poisson Process lambda(t) = t
N(t)
0
20
40
2
4
6
8
10
t


Corresponding Inter−Arrivial Times
y−axis has no meaning
1.0
1.5
2.0
2.5
3.0
2
4
6
8
10
t