

02__proc__non__pos

Frances Lin

4/9/2021

```
library(tidyverse)
library(ggplot2)

set.seed(1) # for reproducibility

t_max <- 10
t <- 0

lambda <- function(t){
  t
}

X <- numeric(0) # vector of t

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

lambda_f <- function(t_upper) {
  # this lambda_fun function takes in t_upper and integrate f =
  # from t = 0 to t = t_upper
  integrate(f = lambda, lower = 0, upper = t_upper)
}

while(t <= t_max){
  # thinning
  u <- runif(1)
  t <- t - log(u)/lambda_star_f() # set t = t + w where w ~ exp
  if(runif(1) < lambda(t)/lambda_star_f()) { # accpet with accepting probabiliy
    X <- c(X,t) # update index
  }
}

# X

length(X)

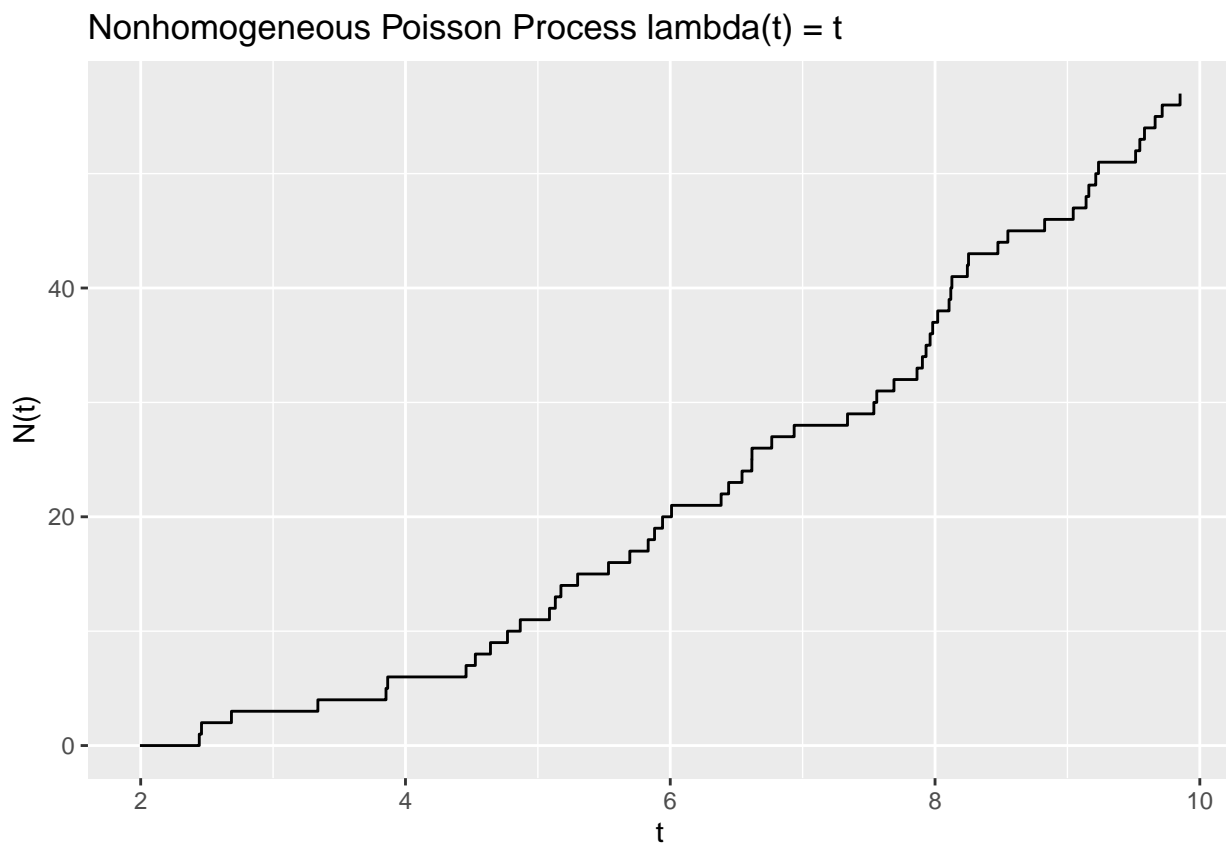
## [1] 58

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

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

```
# 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
```



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

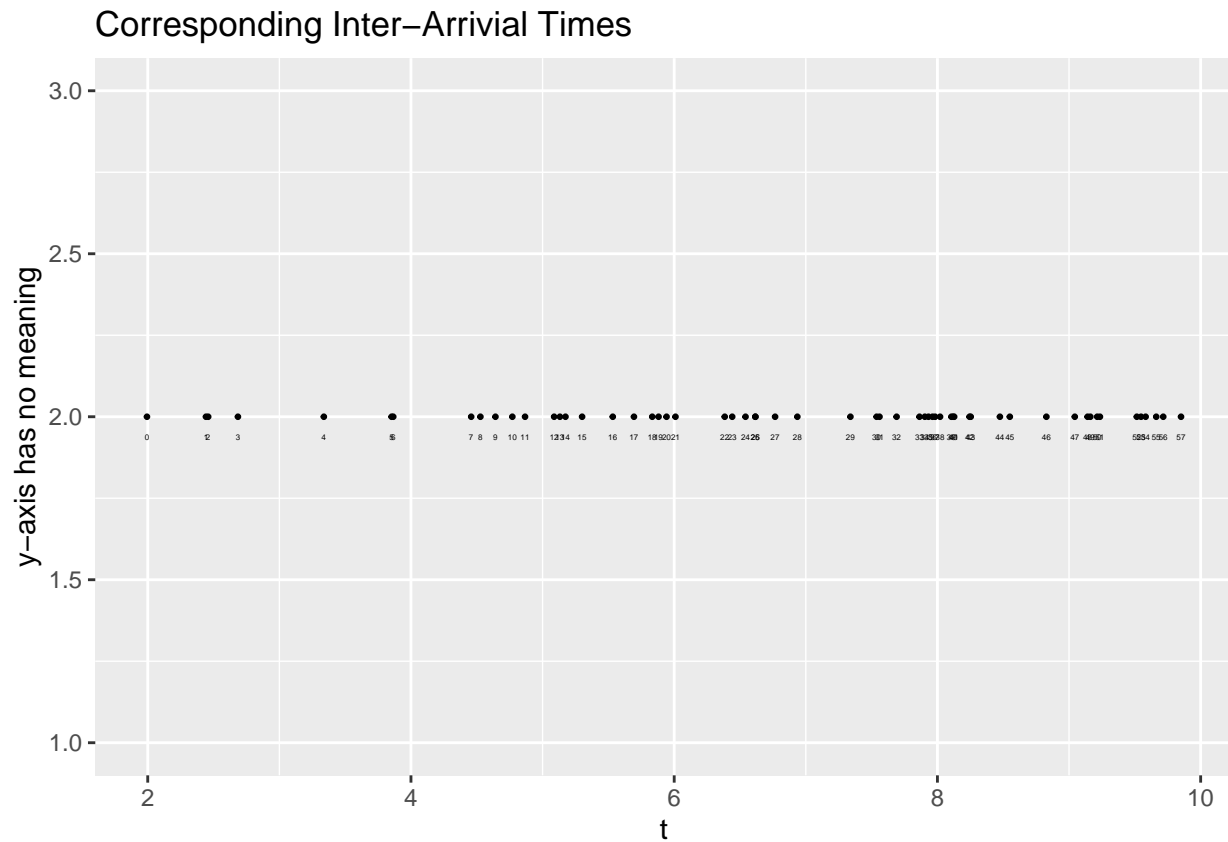
p_NPP_time <- ggplot(data=df_NPP, mapping=aes(x=x, y=t_fix, label = y)) +
  geom_point(size=0.5) +
  ylim(c(1,3)) +
  labs(title = "Corresponding Inter-Arrival Times",
```

```

x = "t",
y = "y-axis has no meaning") +
geom_text(vjust=4, size=1)

```

p_NPP_time



```
require(gridExtra)
```

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

```
##
```

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

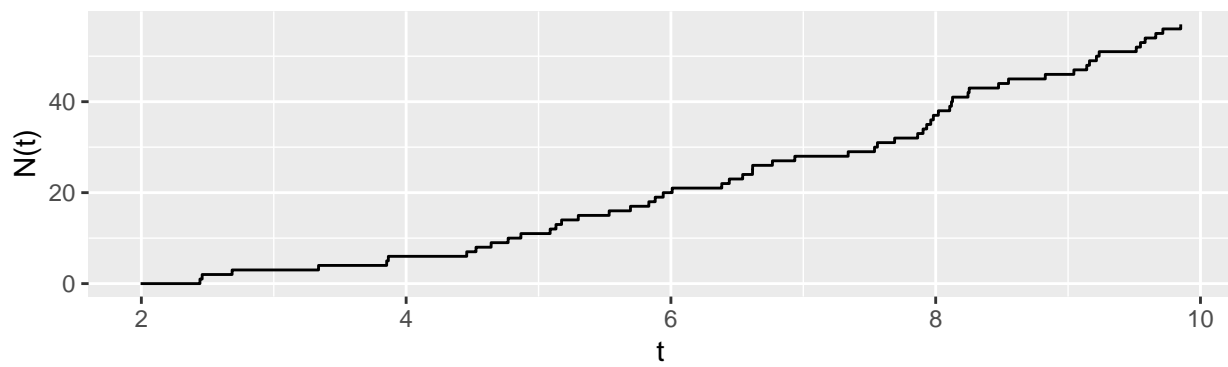
```
## The following object is masked from 'package:dplyr':
```

```
##
```

```
## combine
```

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

Nonhomogeneous Poisson Process $\lambda(t) = t$



Corresponding Inter-Arrival Times

