03_proc_Hawkes

Frances Lin

4/22/2021

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

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

t_max <- 10
t <- 0

lmbda <- 10

t_vec <- numeric(0) # vector of t # consider change it to t_vec

while(t <= t_max){
  u       <- runif(1)
  t       <-  t - log(u)/lmbda                # t ~ exp(1/lambda)
  if(t < t_max) {
    t_vec <- c(t_vec,t)
  }
}
```

```
# Writing the thinning algorithm using James's way

# Initialize
# Note that mu > 0 and 0 < alpha < beta ??
mu = 0.5
alpha = 0.7
beta = 0.5

# Create lambda(t) function
lmbda_fun <- function(time){
  diff = time - t_vec
  diff = diff[diff > 0]
  a = sum(alpha * exp(-beta * diff))
  out = mu + a
  return(out)
}

# Apply the lmbda_fun function
lmbda_star <- sapply(X = t_vec, FUN = lmbda_fun)
length(lmbda_star)
```

```
## [1] 114
```

```r
# lmbda_star #114
```

```r
# t_vec # 114
```

```r
# t_vec[runif(length(prob_keep)) < min(prob_keep, 1)] #75
```

```r
# lmbda_star[runif(length(prob_keep)) < min(prob_keep, 1)] #75
```

```r
set.seed(1)
lmbda <- 10
prob_keep <- lmbda / lmbda_star
#t_keep <- t_vec[runif(length(prob_keep)) < min(prob_keep, 1)]
check <- runif(length(prob_keep)) < min(prob_keep, 1)
lmdba_keep <- lmbda_star[check]
t_keep <- t_vec[check]
length(lmdba_keep)
```

```
## [1] 62
```

```r
length(t_keep)
```

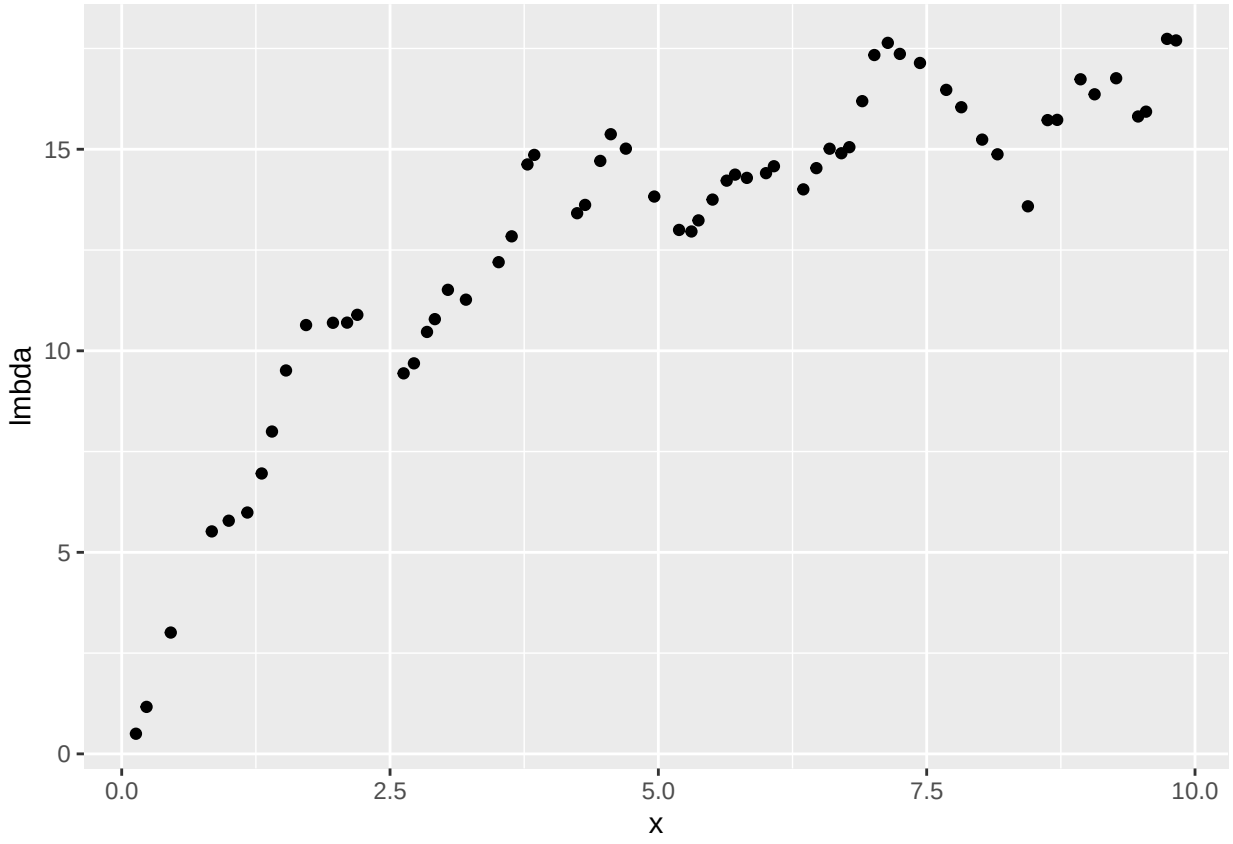
```
## [1] 62
```

```r
# Plot Hawkes
# Create a df
df_Hawkes = tibble(
  x = t_keep,
  y = 1:(length(t_keep)), #0:(length(X) - 1)
  lmbda = lmdba_keep
)

p_Hawkes <- ggplot(data=df_Hawkes, mapping=aes(x=x, y=y)) +
  geom_step() +
  labs(title = "Hawkes Process",
       x = "t",
       y = "N(t)")
#p_Hawkes
```

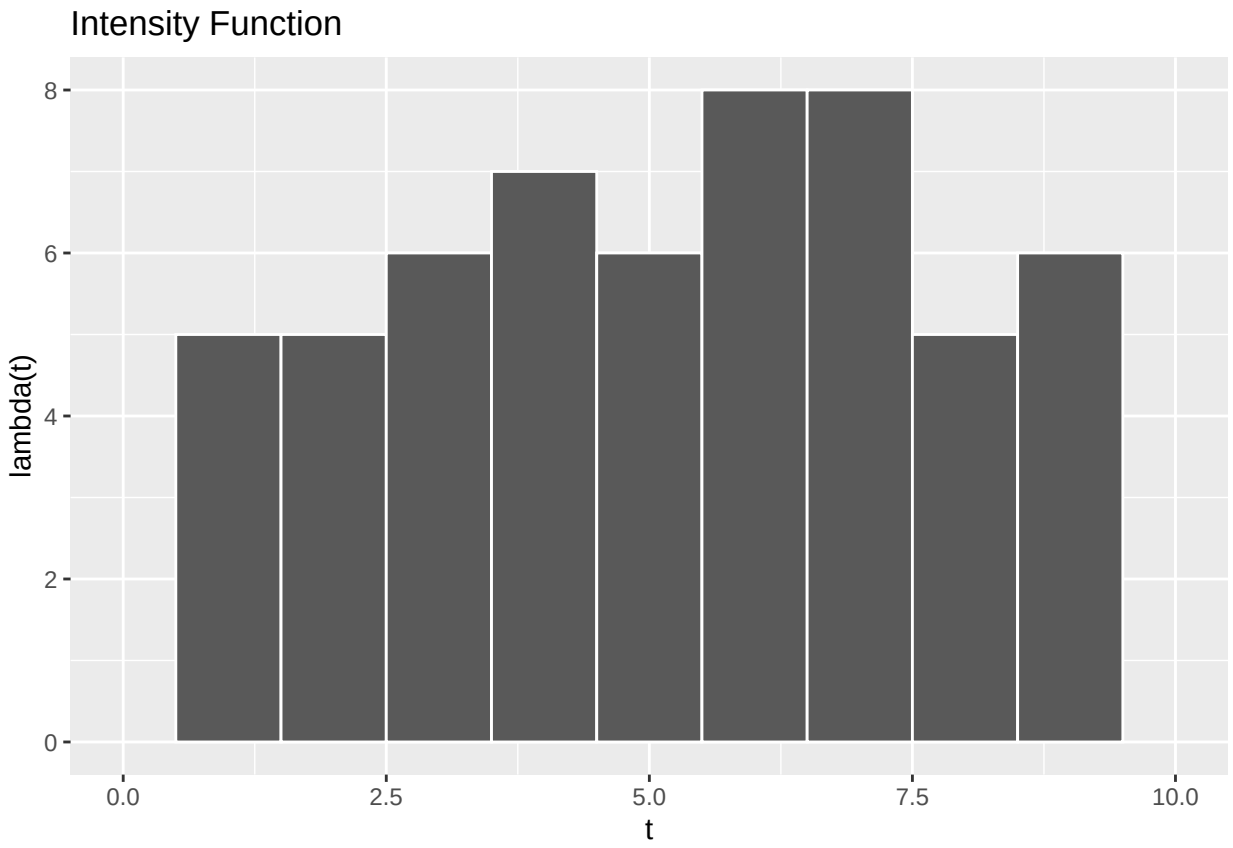
```r
# Plot time plot
p_Hawkes_time <- ggplot(data=df_Hawkes, mapping=aes(x=x, ymin = -0.5, ymax = 0.5)) +
  geom_linerange() +
  geom_hline(aes(yintercept = 0), linetype = "dashed") +
  labs(title = "Corresponding Inter-Arrivial Times",
       x = "t",
       y = "") +
  scale_x_continuous(limits = c(0, 10), breaks = seq(0, 10, by = 1)) +
  theme(axis.text.y=element_blank(), axis.ticks.y=element_blank())
#p_Hawkes_time
```

```r
# Plot lambda_star vs t
p_Hawkes_Int <- ggplot(df_Hawkes, aes(x=x, y=lmbda)) + # y is not right
  geom_point()
p_Hawkes_Int
```

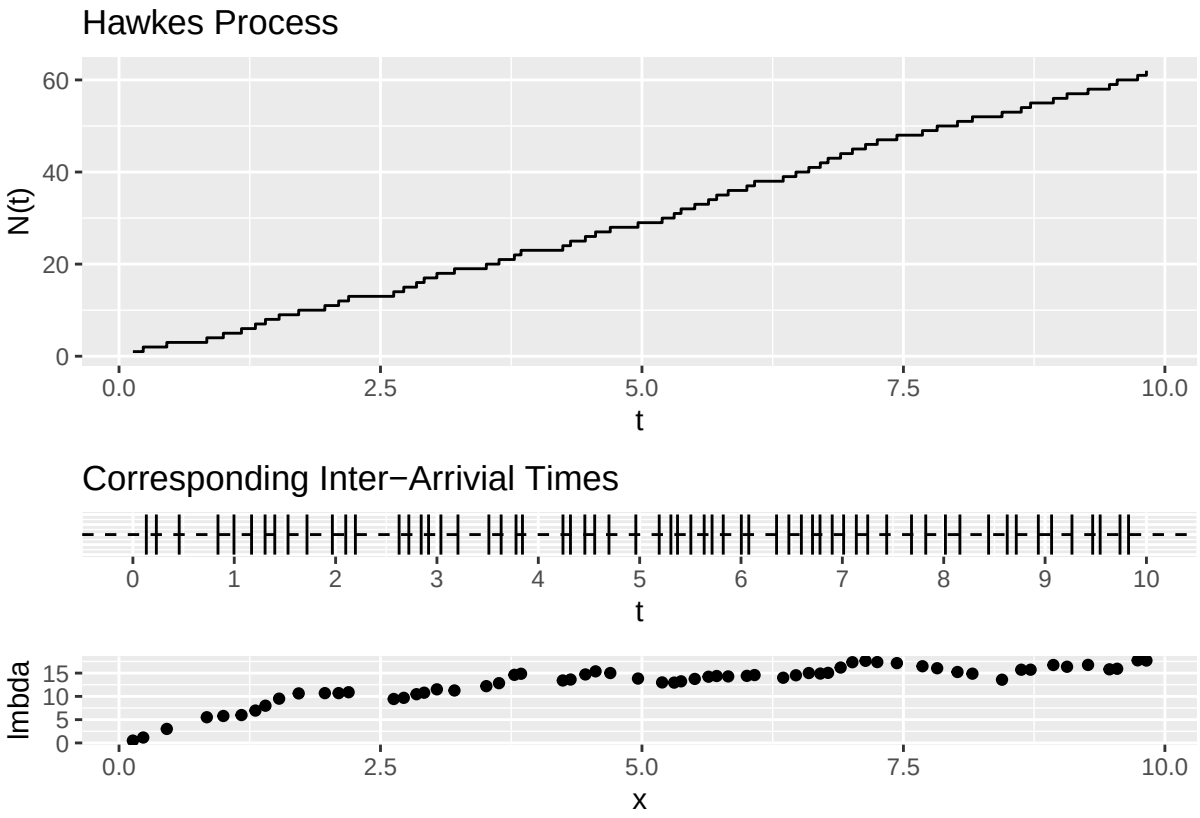



```r
# Plot lambda (rate)
p_Hawkes_hist <- ggplot(data=df_Hawkes, mapping=aes(x=x)) +
  geom_histogram(bins = 11, color = "white") +
  labs(title = "Intensity Function",
       x = "t",
       y = "lambda(t)") +
  xlim(0, 10) # so that scale lines up
p_Hawkes_hist
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```




```
# # Combine plots
# require(gridExtra)
# grid.arrange(p_Hawkes, p_Hawkes_time)
p_Hawkes / p_Hawkes_time / p_Hawkes_Int + plot_layout(heights = c(0.7, 0.1, 0.2))
```



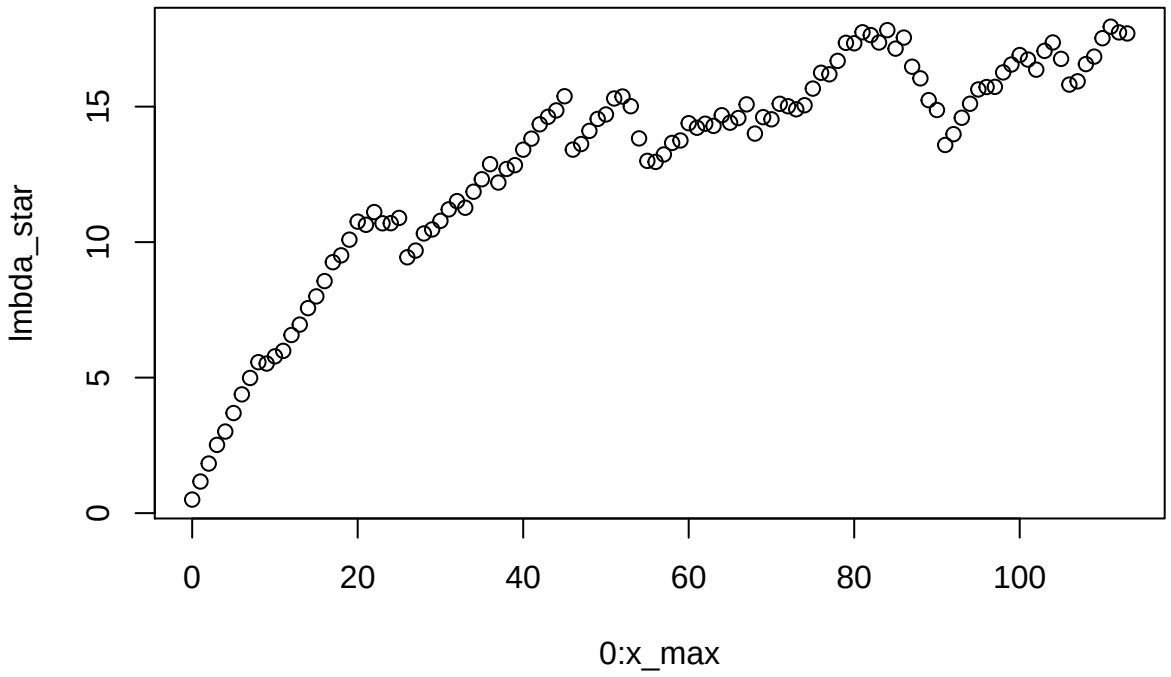

```
# Save output and adjust size
png(file = '/Users/franceslinyc/Hawkes-Process-2021/results/plot_1D_Hawkes.png', width = 450*2, height =
p_Hawkes / p_Hawkes_time / p_Hawkes_Int + plot_layout(heights = c(0.7, 0.1, 0.2))
```

```
# while(t <= t_max){
#   if(runif(1) < min(lmbda/lmbda_star, 1)){
#     X_keep <- c(X, t)
#   }
#   return(X_keep)
# }
```

```
# Plot lmbda_star
# Not thinned yet
length(lmbda_star) #114
```

```
## [1] 114
```

```
x_max = length(lmbda_star) - 1
plot(x = 0:x_max, y = lmbda_star)
```



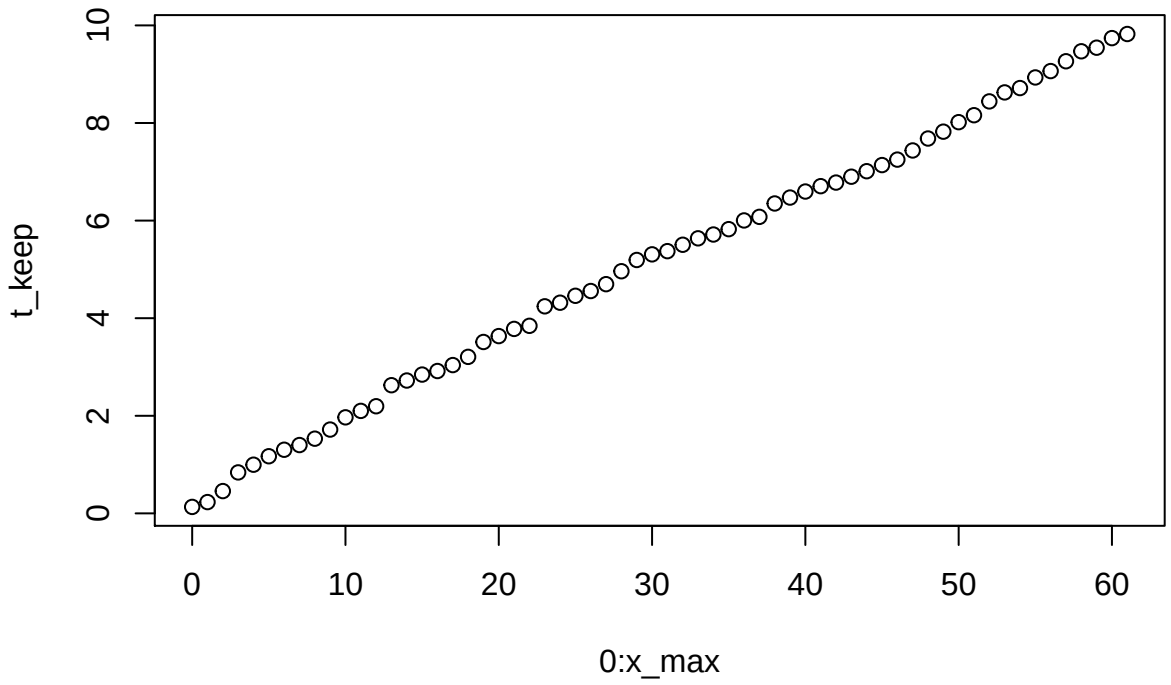

```
lmbda_star
```

```
##   [1]  0.500000  1.166243  1.828709  2.518966  3.009789  3.692638  4.381577
##   [8]  4.987644  5.568817  5.519164  5.784735  5.987685  6.572629  6.956240
##  [15]  7.563257  7.997083  8.562208  9.258648  9.511981 10.090501 10.755817
##  [22] 10.638557 11.108964 10.694468 10.698810 10.892377  9.440549  9.688126
##  [29] 10.319339 10.467464 10.785304 11.207891 11.512135 11.268051 11.859905
##  [36] 12.319467 12.876089 12.198703 12.699850 12.839321 13.411322 13.818262
##  [43] 14.347787 14.623249 14.859711 15.382651 13.413329 13.619000 14.105405
##  [50] 14.545215 14.710365 15.299386 15.372810 15.014675 13.826841 12.998089
##  [57] 12.959886 13.234890 13.660645 13.751166 14.387725 14.220679 14.370096
##  [64] 14.289392 14.681589 14.406944 14.578472 15.083097 14.004828 14.610564
##  [71] 14.530923 15.102575 15.013133 14.900957 15.051268 15.664532 16.249309
##  [78] 16.192795 16.687621 17.353729 17.337470 17.742754 17.639249 17.365307
##  [85] 17.822599 17.140228 17.547158 16.473199 16.042486 15.238914 14.874533
##  [92] 13.584656 13.982873 14.589516 15.105905 15.633533 15.722693 15.728607
##  [99] 16.262510 16.553932 16.902863 16.735650 16.363206 17.057126 17.367689
## [106] 16.761283 15.812665 15.932645 16.569077 16.844507 17.523966 17.949392
## [113] 17.737683 17.699430
```

```
# Plot lmbda_star
# Not thinned yet
length(t_keep) # 75
```

```
## [1] 62
```

```
x_max = length(t_keep) - 1
plot(x = 0:x_max, y = t_keep)
```


t_keep
0
2
4
6
8
10
0
10
20
30
40
50
60
0:x_max