# 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 = 0.025
alpha = 0.5
beta = 0.7

# Create lambda(t) function
lmbda_fun <- function(time, obs){
  diff = time - obs
  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, obs = t_vec)
```

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

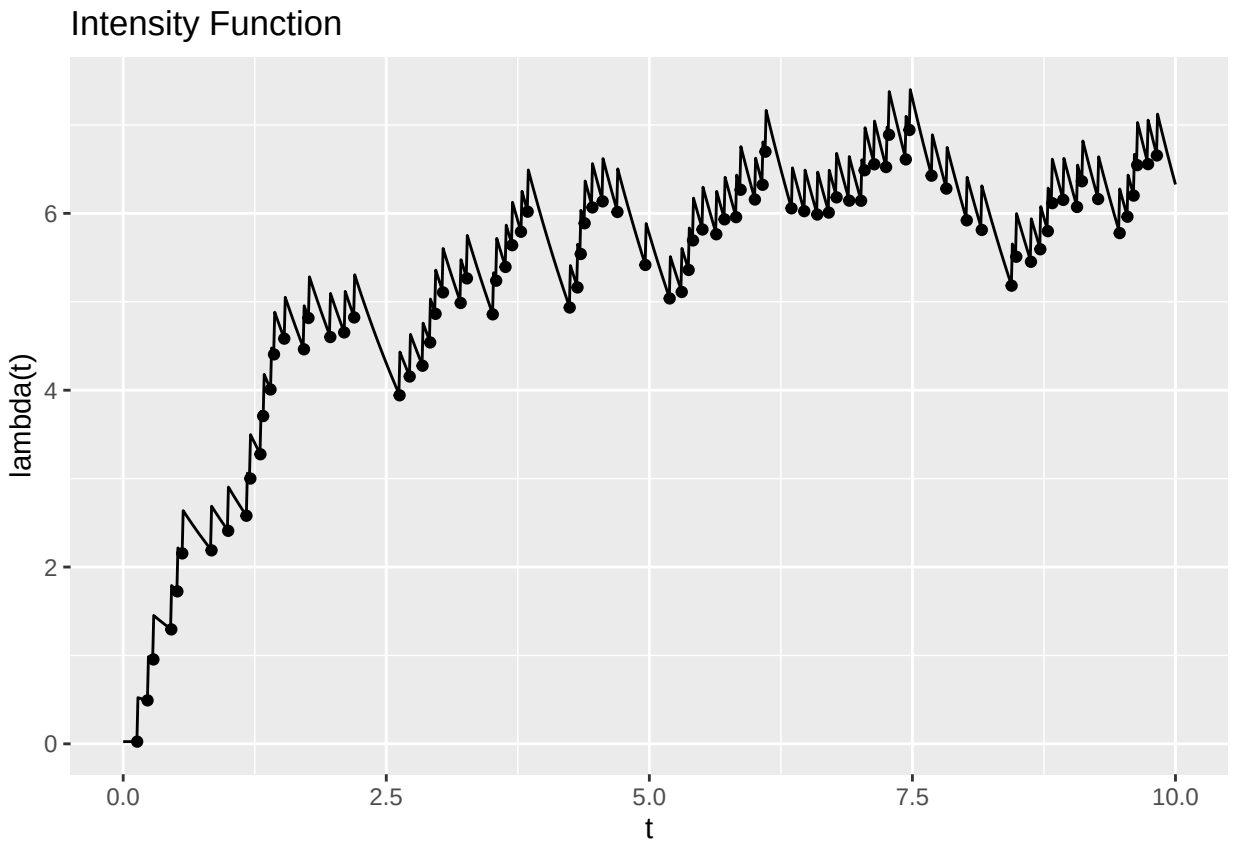
```r
check <- runif(length(prob_keep)) < min(prob_keep, 1)
lmdba_keep <- lmbda_star[check]
t_keep <- t_vec[check]
# length(lmbda_star) #114
# length(lmdba_keep) #87
# length(t_vec) #114
# length(t_keep) #87
```

```r
# Plot Hawkes
# Create a df
df_Hawkes = tibble(
  x = t_keep,
  y = 1:(length(t_keep)), #0:(length(X) - 1)
  lmbda = lmdba_keep,
  lmbda2 = sapply(X = t_keep, FUN = lmbda_fun, obs = t_keep)
)

df_Hawkes2 = tibble(
  x = seq(0, 10, length.out = 1001),
  y = sapply(X = x, FUN = lmbda_fun, obs = t_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
```

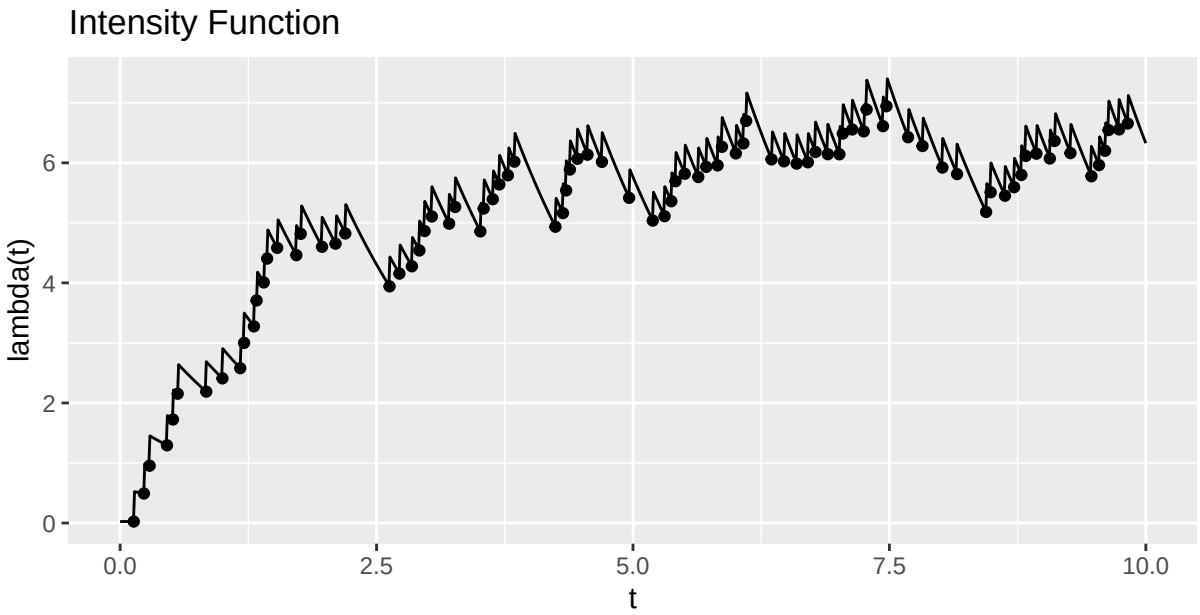
```r
# Plot lambda_star vs t
# p_Hawkes_Int <- ggplot(df_Hawkes, aes(x=x, y=lmbda2)) +
#   geom_line() +
#   labs(title = "Intensity Function",
#        x = "t",
#        y = "lambda(t)")
# p_Hawkes_Int

p_Hawkes_Int2 <- ggplot(df_Hawkes2, aes(x=x, y=y)) +
  geom_line() +
  geom_point(aes(x=x, y=lmbda2), data = df_Hawkes) +
  labs(title = "Intensity Function",
       x = "t",
```

```
       y = "lambda(t)")
p_Hawkes_Int2
```




```
# # 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
```

```
# # Combine plots
# require(gridExtra)
# grid.arrange(p_Hawkes, p_Hawkes_time)
p_Hawkes_Int2 / p_Hawkes_time + plot_layout(heights = c(0.9, 0.1))
```





```
# Save output and adjust size
png(file = '/Users/franceslinyc/Hawkes-Process-2021/results/plot_1D_Hawkes.png', width = 450*2, height =
p_Hawkes_Int2 / p_Hawkes_time + plot_layout(heights = c(0.9, 0.1))
```

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

```
# # Plot lmbda_star
# # Not thinned yet
# length(t_keep) # 75
# x_max = length(t_keep) - 1
# plot(x = 0:x_max, y = t_keep)
```