Simulations

Jason Luo

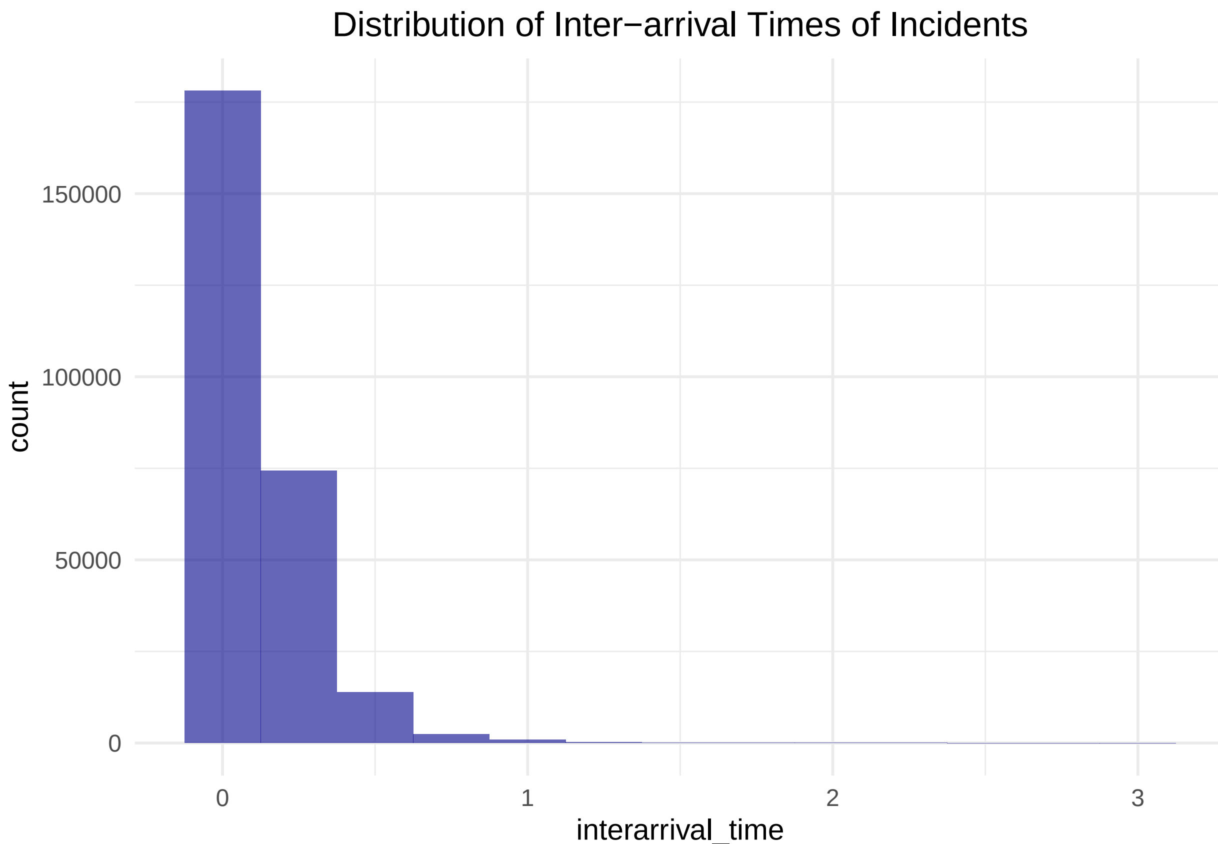
2024-07-30

```r
rm(list = ls())

setwd('/Users/jasonluo/Documents/R_crime_analysis')
library(tidyverse)

# Importing cleaned data
df <- read.csv('RMS_Crime_Incidents_Cleaned.csv')
```

Inter-arrival Times

Inter-arrival times follow an exponential distribution. The lambda parameter is estimated using the sample mean from our data.

```r
ggplot(data = df, aes(x = interarrival_time)) +
  geom_histogram(fill = 'darkblue', alpha = 0.6, bins = 30, binwidth = 0.25) +
  ggtitle("Distribution of Inter-arrival Times of Incidents") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```




```
sample_mean <- mean(df$interarrival_time)
lambda <- 1/sample_mean
```

The sample mean is 0.1160534 and thus the value of lambda is 8.6167257. This means there are 0.1160534 hours between incidents and there are on average 8.6167257 incidents per hour.

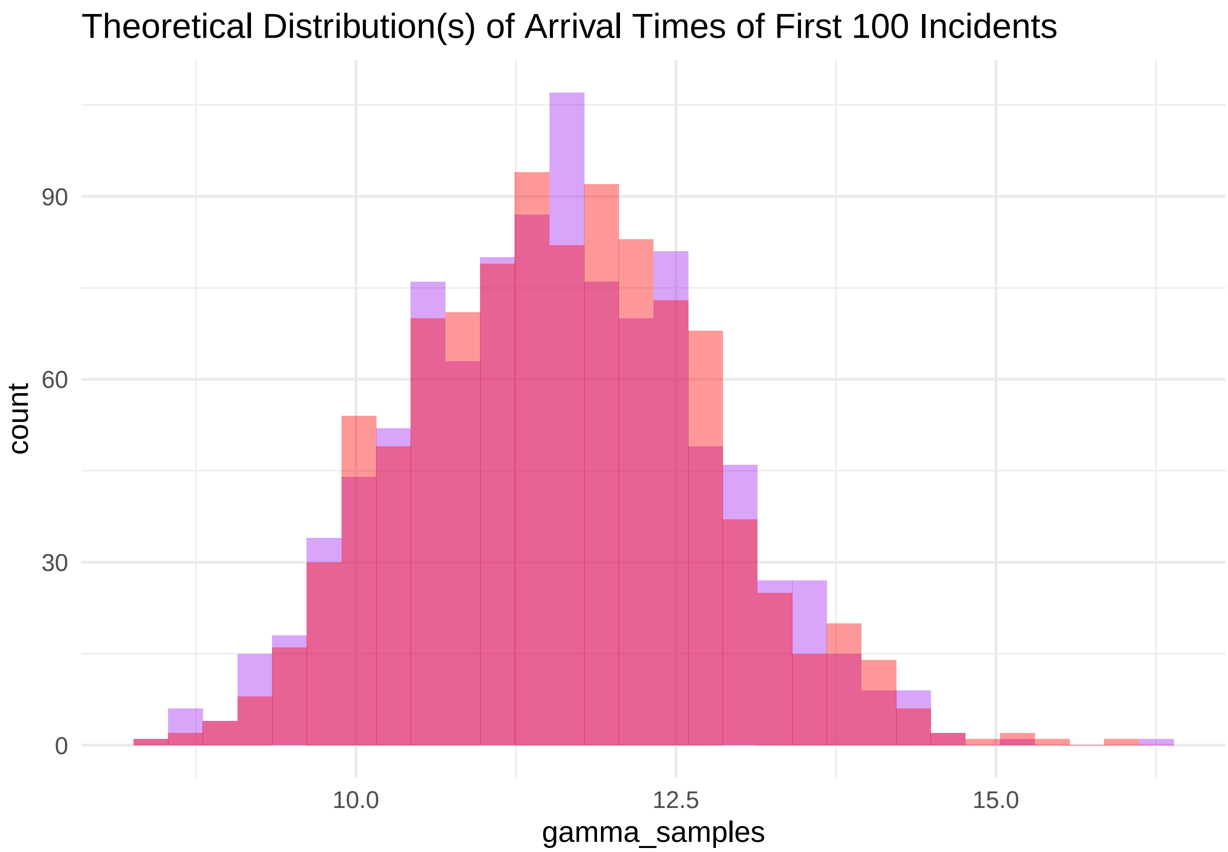
Since inter-arrival times are exponentially distributed, a sum of inter-arrival random variables has a gamma distribution. This sum is the arrival times (till the nth arrival)

```
n <- 100 # 100 arrivals
set.seed(1234)

generate_arrival_times <- function(n) {
  interarrival_time_samples <- rexp(n = n, rate = lambda)
  arrival_time_sample <- sum(interarrival_time_samples)
  return(arrival_time_sample)
}

arrival_times <- replicate(1000, generate_arrival_times(n)) # generates 1000 samples of arrival times
arrival_times_df <- as.data.frame(arrival_times)
```

```
gamma_samples <- rgamma(n = 1000, shape = n, rate = lambda)
gamma_samples_df <- as.data.frame(gamma_samples)

# Plot 2 ways to simulate fist n arrival times, which follow a gamma distribution
ggplot() +
  geom_histogram(data = gamma_samples_df, aes(x = gamma_samples), fill = 'purple', alpha = 0.4) +
  geom_histogram(data = arrival_times_df, aes(x = arrival_times), fill = 'red', alpha = 0.4) +
```

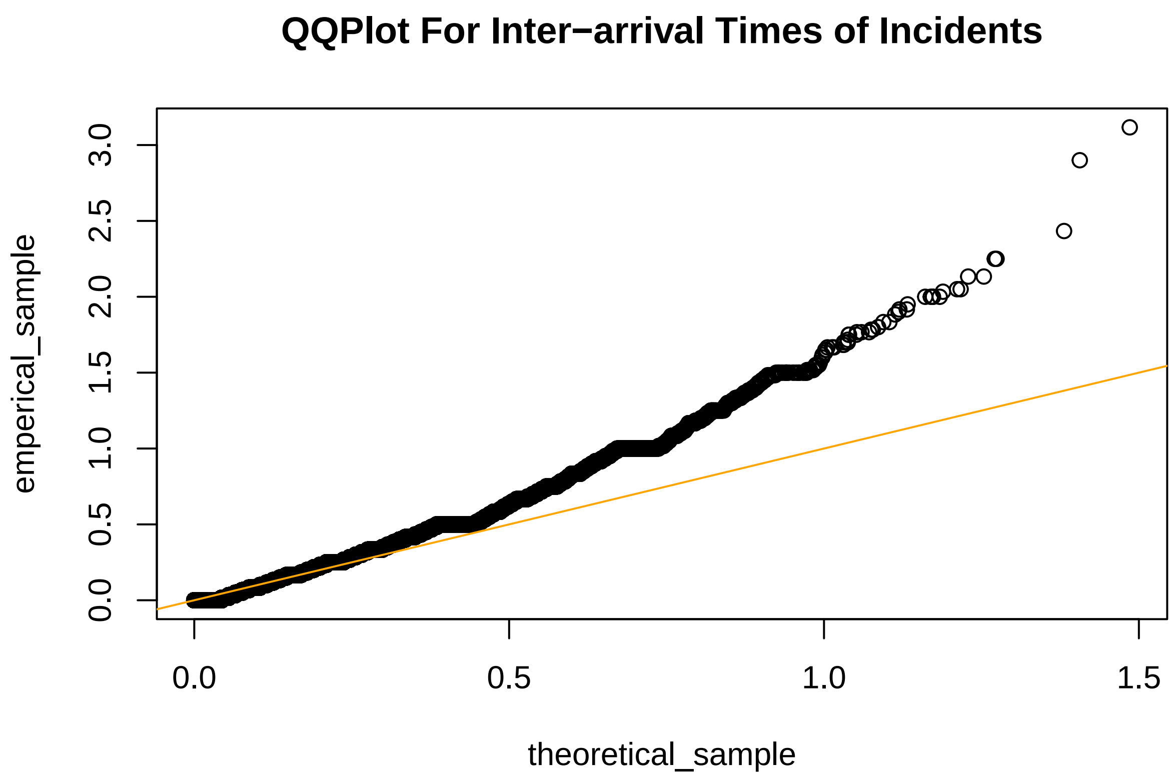
```
  ggtitle(paste0("Theoretical Distribution(s) of Arrival Times of First " , n, " Incidents")) +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme_minimal()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



We can check how accurate our sample of inter-arrival times follows an exponential distribution

```
set.seed(12345)
theoretical_sample <- rexp(n = dim(df)[1], rate = lambda)
emperical_sample <- df$interarrival_time

qqplot(theoretical_sample, emperical_sample, main = paste0("QQPlot For Inter-arrival Times of Incidents
abline(a = 0, b = 1, col = 'orange')
```



Simulating a Poisson Process

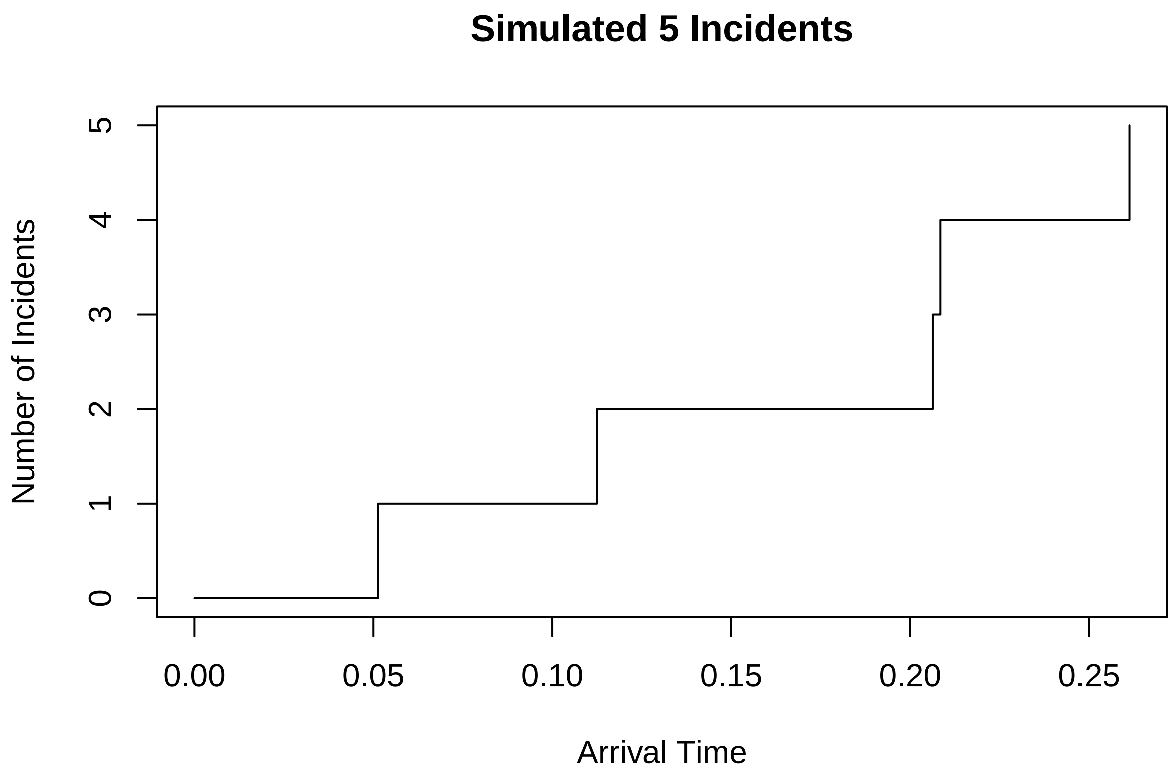
We estimate the rate parameter for our poisson process by computing `num_incidents / total_time`

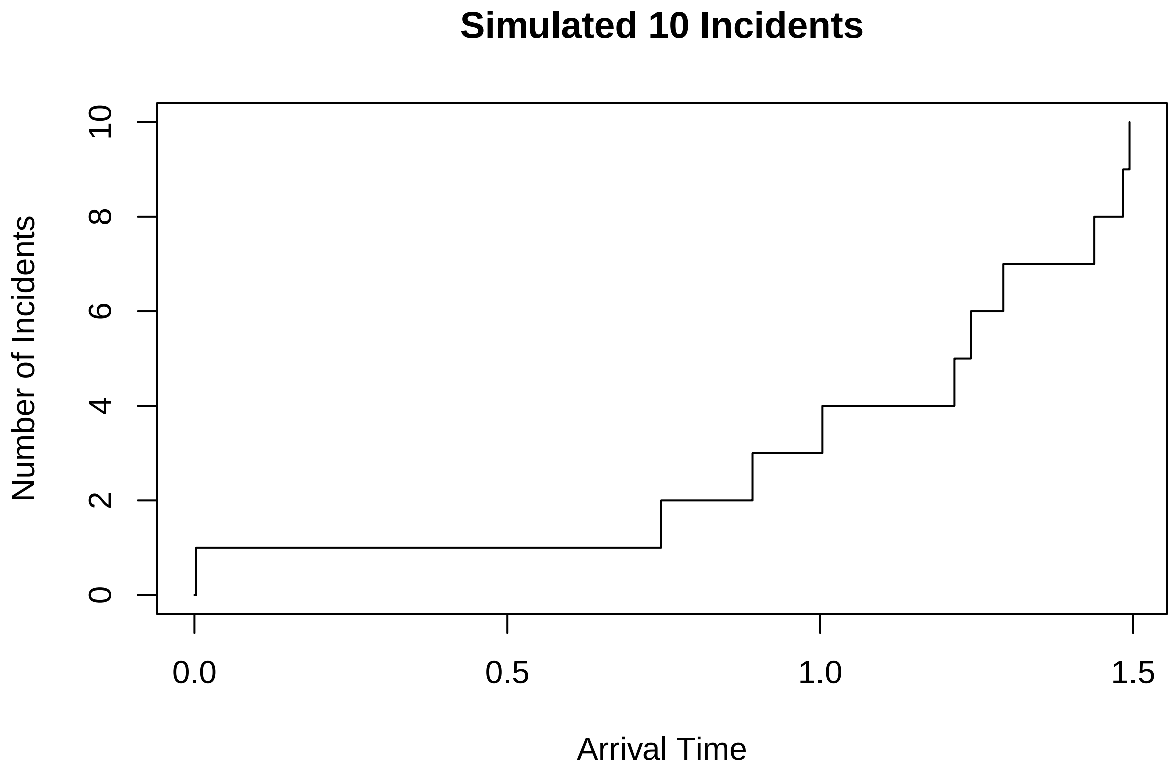
```
num_incidents <- nrow(df)
total_time <- as.numeric(difftime(df$incident_timestamp[dim(df)[1]], df$incident_timestamp[1], units =
rate <- num_incidents / total_time
```

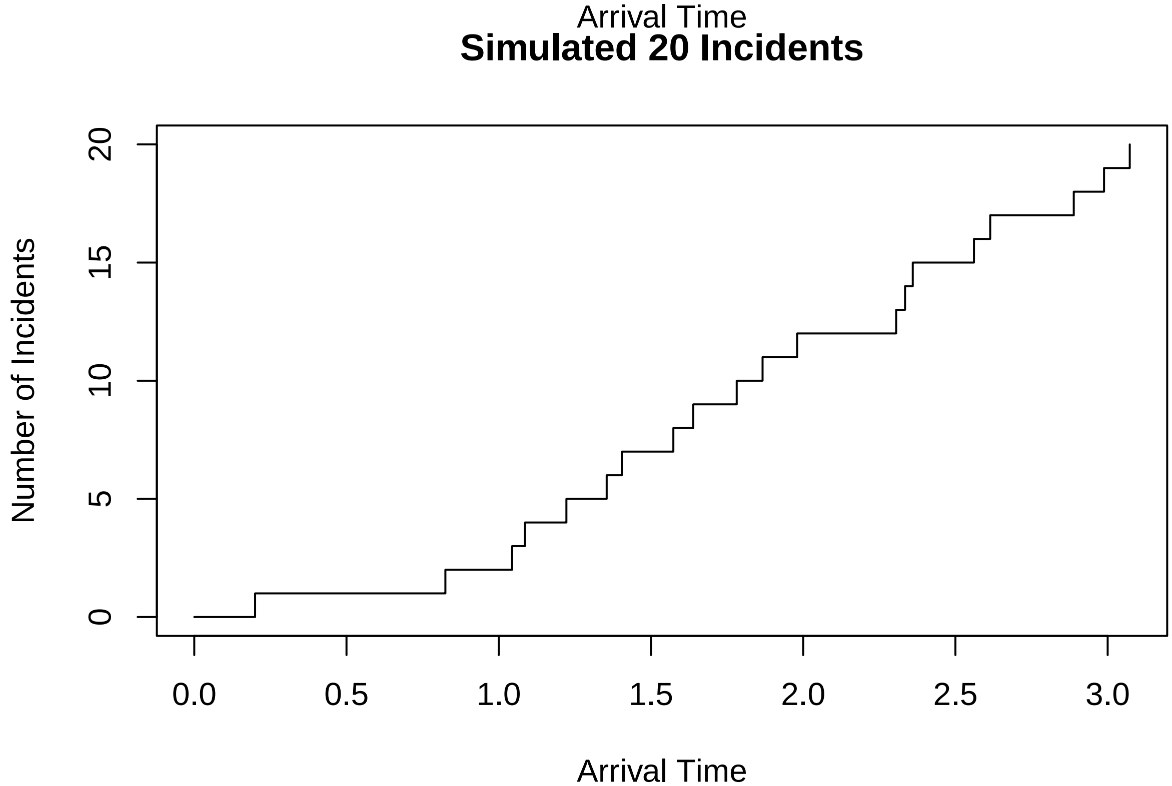
There are 8.6167257 incidents an hour. This is the exact same as the lambda value from the exponential distribution computed prior. We can now simulate a poisson process:

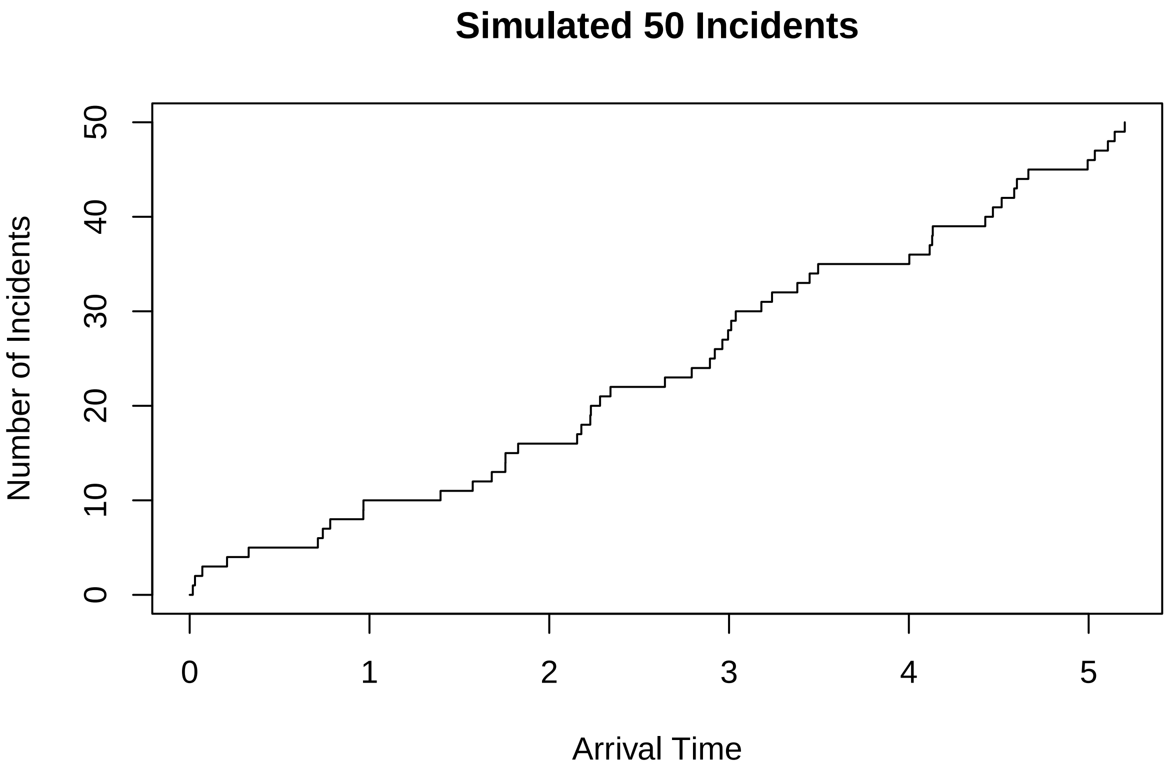
```
set.seed(12345)

pois_process <- function(num_arrivals, lambda) {
  arrival_time_stamps <- c() # vector of arrival times

  for(i in 1:num_arrivals) {
    inter_arrival_time <- rexp(n = 1, rate = lambda) # inter-arrival times ~ exp(lambda)

    if (i == 1) {
      temp <- 0
    } else {
      temp <- arrival_time_stamps[i-1]
    }

    arrival_time <- sum(temp, inter_arrival_time)
    # sum of N inter-arrival times is the arrival time for the Nth event
    arrival_time_stamps <- c(arrival_time_stamps, arrival_time)
    # append arrival times to vector
  }
```

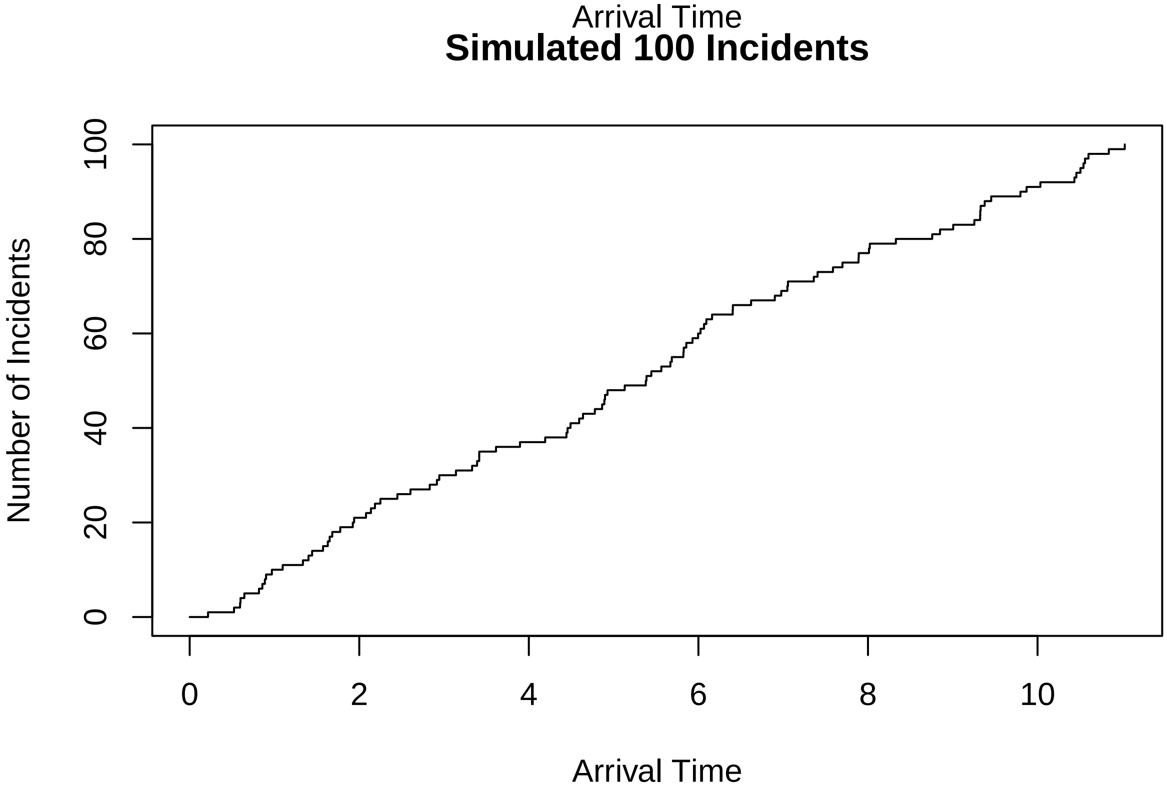
```r
  Nt <- 1:length(arrival_time_stamps)
  plot(c(0, Nt) ~ c(0, arrival_time_stamps), type = 's', main = paste0("Simulated " , num_arrivals, " I
       xlab = "Arrival Time", ylab = "Number of Incidents")

}


for (i in c(5,10,20,50, 100)) {
  pois_process(num_arrivals = i, lambda = rate)
}
```




Simulated 10 Incidents
Number of Incidents
0
2
4
6
8
10
0.0
0.5
1.0
1.5
Arrival Time


Simulated 20 Incidents
Number of Incidents
0
5
10
15
20
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Arrival Time





Binomial Poisson Model

```
categories <- unique(df$offense_category)
probs <- c()
```

```r
for (i in 1:length(categories)) {
  prob_of_category <- sum(df$offense_category == categories[i]) / nrow(df)
  print(paste0("Probability of commiting ", categories[i], " offense: ", prob_of_category))
  probs[i] <- prob_of_category
}
```

```
## [1] "Probability of commiting STOLEN VEHICLE offense: 0.108019269964795"
## [1] "Probability of commiting LARCENY offense: 0.145262182694089"
## [1] "Probability of commiting WEAPONS OFFENSES offense: 0.0474041133963313"
## [1] "Probability of commiting AGGRAVATED ASSAULT offense: 0.09461552714471"
## [1] "Probability of commiting STOLEN PROPERTY offense: 0.0323994811932555"
## [1] "Probability of commiting DAMAGE TO PROPERTY offense: 0.132199370020382"
## [1] "Probability of commiting ASSAULT offense: 0.190950528071151"
## [1] "Probability of commiting HOMICIDE offense: 0.0030461367426348"
## [1] "Probability of commiting FRAUD offense: 0.0845025013896609"
## [1] "Probability of commiting OBSTRUCTING THE POLICE offense: 0.0105354826755605"
## [1] "Probability of commiting ROBBERY offense: 0.017124328330554"
## [1] "Probability of commiting ARSON offense: 0.00580322401334075"
## [1] "Probability of commiting BURGLARY offense: 0.0605447470817121"
## [1] "Probability of commiting SEX OFFENSES offense: 0.0111802853437095"
## [1] "Probability of commiting SEXUAL ASSAULT offense: 0.00797850657772837"
## [1] "Probability of commiting EXTORTION offense: 0.000629979618306467"
## [1] "Probability of commiting DANGEROUS DRUGS offense: 0.0109208819714656"
## [1] "Probability of commiting RUNAWAY offense: 0.00914952751528627"
## [1] "Probability of commiting DISORDERLY CONDUCT offense: 0.00250509542338336"
## [1] "Probability of commiting OBSTRUCTING JUDICIARY offense: 0.0101315545673522"
## [1] "Probability of commiting FAMILY OFFENSE offense: 0.00448397257735779"
## [1] "Probability of commiting FORGERY offense: 0.00198258291643506"
## [1] "Probability of commiting KIDNAPPING offense: 0.00148230498425051"
## [1] "Probability of commiting OUIL offense: 0.0018195293681675"
## [1] "Probability of commiting LIQUOR offense: 0.000511395219566426"
## [1] "Probability of commiting MISCELLANEOUS offense: 0.000837502316101538"
## [1] "Probability of commiting OTHER offense: 0.00356494348712248"
## [1] "Probability of commiting SOLICITATION offense: 4.81749119881416e-05"
## [1] "Probability of commiting JUSTIFIABLE HOMICIDE offense: 0.000352047433759496"
## [1] "Probability of commiting GAMBLING offense: 1.48230498425051e-05"
```

Binomial Poisson model for simulating the number of particular incidents

```r
set.seed(123)

binPois <- function(n, prob) {
  y <- rpois(n, lambda = lambda) # y ~ pois
  x_y <- rbinom(n, size = y, prob = prob) # x | y ~ bin

  x_y_df <- as.data.frame(x_y)

  graph <- ggplot(data = x_y_df, aes(x_y)) +
    geom_histogram(bins = 20, alpha = 0.6, fill = "lightgreen") +
    theme_minimal() +
    ggtitle(paste0("Estimated Distribution of ", n, " Incidents")) +
    theme(plot.title = element_text(hjust = 0.5))

  return(graph)
}
```

```
binPois(n = 10000, prob = max(probs))
```