MIE-SPI 2014 – Homework No. 1

Team leader: Timur Tatarshaov (tatartim)
Date: 21.04.2014

1. Generating of a random sample and graphical verification of its distribution

1.I. Random values from Exp(L) generation

R commands according to the instructions:

```
K = 5;
L = 10;
n = K*20;
u = runif(n, 0, 1);
x = sapply(u, function(x) { -1 / L * log(1-x) });
x;
```

- Output:

```
 [1] 0.0477547540 0.1935880087 0.1989734088 0.0533036868 0.0027216545
  [6] 0.0217711377 0.0138585473 0.0812654112 0.0112832934 0.0666132807
 [11] 0.0705783986 0.0456461431 0.1662214387 0.0082814566 0.0389500995
 [16] 0.1569878067 0.0448024870 0.1246314573 0.0687633875 0.0899837389
 [21] 0.1095666151 0.0358022250 0.0030876778 0.1000982958 0.2651562485
 [26] 0.1905200013 0.0701409023 0.0211179416 0.0004592880 0.2987919585
 [31] 0.0796705788 0.0308097198 0.0799226338 0.0151366775 0.0809905616
 [36] 0.0468561319 0.2171485731 0.0502073190 0.0476231678 0.2210240551
 [41] 0.0623761522 0.2762411479 0.0066581587 0.1320692034 0.0281953656
 [46] 0.0291576761 0.0136004243 0.0655022532 0.0282261527 0.0398570165
 [51] 0.3134427655 0.0281679021 0.0308823490 0.0731974040 0.1134317155
 [56] 0.0344868151 0.1282048039 0.0391394589 0.0931966082 0.1534367358
 [61] 0.0008044644 0.0218343000 0.1066272312 0.2591080060 0.0449072825
 [66] 0.0562408326 0.0086155730 0.0029871897 0.0087703232 0.3475782376
 [71] 0.0075114799 0.0711884866 0.0594977693 0.1362101789 0.0683507591
 [76] 0.0002116523 0.2017399140 0.1808688799 0.1335807161 0.1352568365
 [81] 0.2450275079 0.0356711332 0.1664624215 0.1059594295 0.1002653521
 [86] 0.1050785293 0.0014468705 0.0345229535 0.2608948838 0.0108154042
 [91] 0.1422872449 0.0413805450 0.1861395373 0.0396794734 0.0748059115
 [96] 0.0495237562 0.1071789142 0.0034995034 0.0042909242 0.1063855024
```

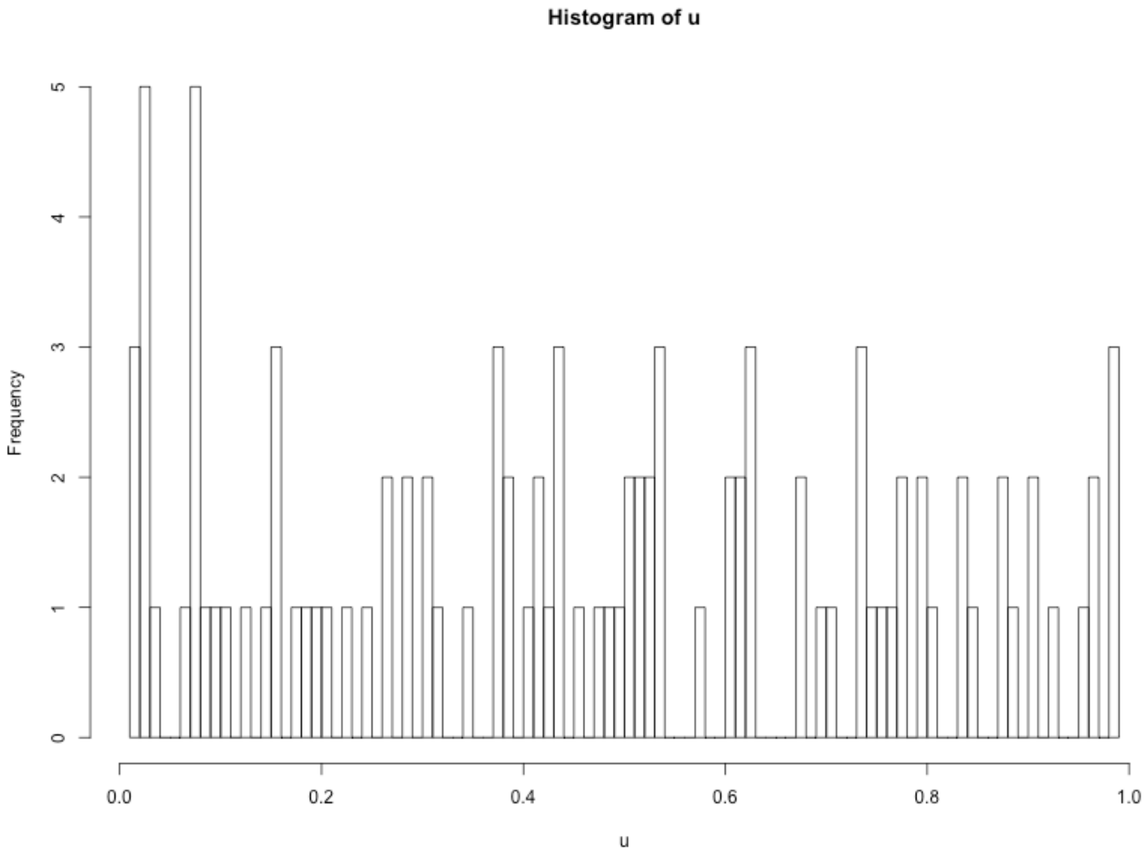
1.II. Plotting the histograms for data in the variables u a x

R commands according to the instructions:

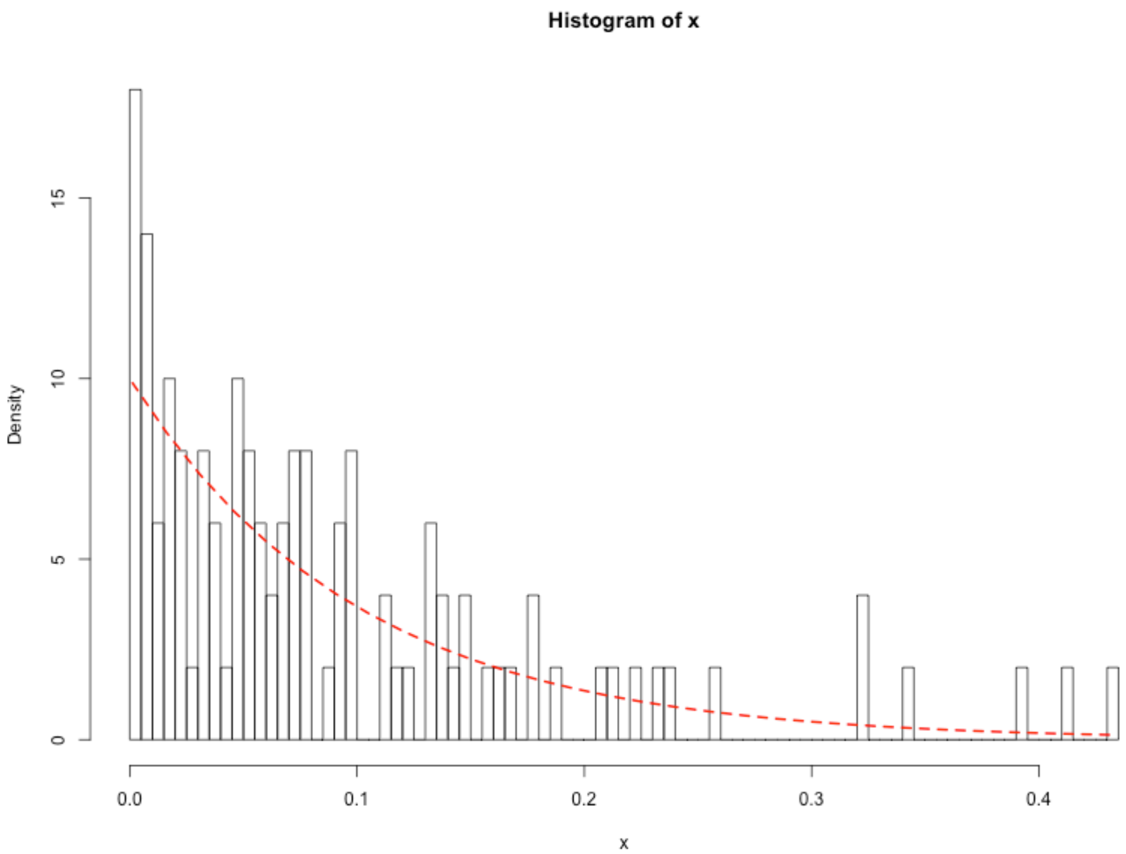
```
# 1.1
png(filename = '1.2_1.png');
hist(u, breaks=length(u), freq=TRUE);
graphics.off();

# 1.2
png(filename = '1.2_2.png');
hist(x, breaks=length(u), freq=FALSE);

xGrid=seq(min(x),max(x),length=30)
lines (xGrid,dexp(xGrid, rate=L), col='red', lw=2, lty=2)

graphics.off();
```

- Output

  u:


Histogram of u
Frequency
0
1
2
3
4
5
0.0
0.2
0.4
0.6
0.8
1.0
u

x:


Histogram of x
Density
0
5
10
15
0.0
0.1
0.2
0.3
0.4
x

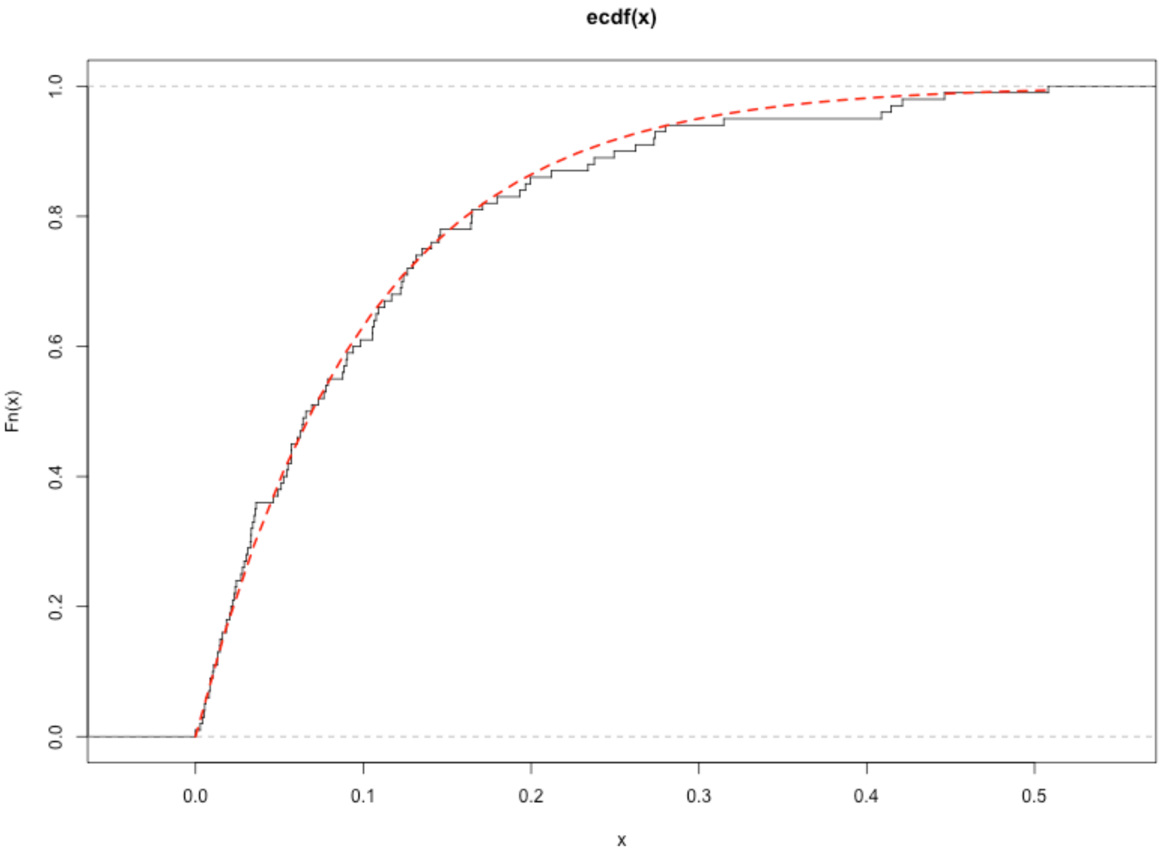
We can see that values issued by transforming values generated with uniform distribution reminds theoretically exponential distributed values with parameter L.

1.III. Plotting the graph of the empirical distribution function for data in x, combined with the graph of the distribution function of the Exp(L)

R commands according to the instructions:

```
# 1.3
png(filename = '1.2_3.png');
plot(ecdf(x), verticals=TRUE, do.points = FALSE);
xGrid=seq(min(x),max(x),length=30)
lines (xGrid, pexp(xGrid, rate=L), col='red', lw=2, lty=2)

graphics.off();
```

- Output



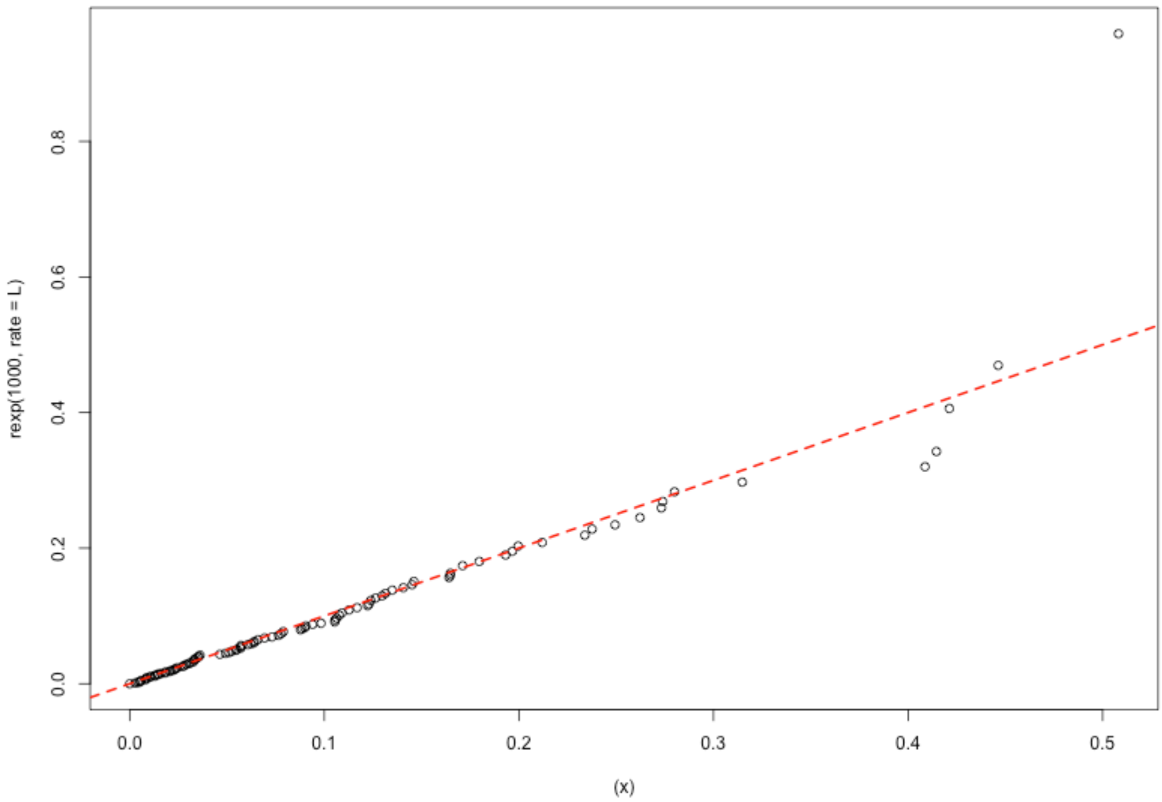
We can see that empirical cdf reminds theoretical nicely.

1.IV. Plotting the "probability plot"

R commands according to the instructions:

```
png(filename = '1.2_4.png');
qqplot((x), rexp(1000, rate=L), plot.it = TRUE);
abline (0, 1, col='red', lw=2, lty=2)
graphics.off();
```

- Output



We can see that issued QQ image proves that x values distributed exponentially with parameter L.

2. Generating of a non-homogeneous Poisson process and graphical verification of its distribution

2.I. Poisson process with varying rate lambda(t)

There was actually no task. Just prepare data for the second part.

R commands according to given data:

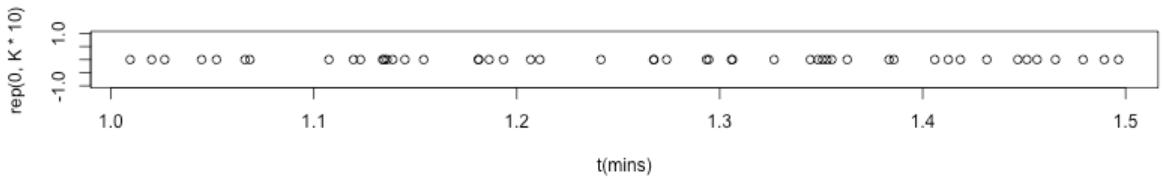
```
lambda = function(t) { 100 + 50/exp((t - 420)^2/(3600*L)) + 100/exp((L*(t - 480 -
30*L)^2)/360000)};
arrival_times = numeric();
current_time = 1;
while (current_time <= 24 * 60)
{
    increment = rexp(1, lambda(current_time));
    current_time = current_time + increment;
    arrival_times = append(arrival_times, current_time);
}
```

2.II. Plotting arrival times for this Poisson process during the first day for the first the first K*10 arrivals

R commands according to the instructions:

```
png(filename = "2.2.png", width = 800, height = 600);
plot(x = arrival_times[1:(K * 10)], y = rep(0, K * 10), type = 'p', xlab = "t(mins)");
graphics.off();
```

- Output

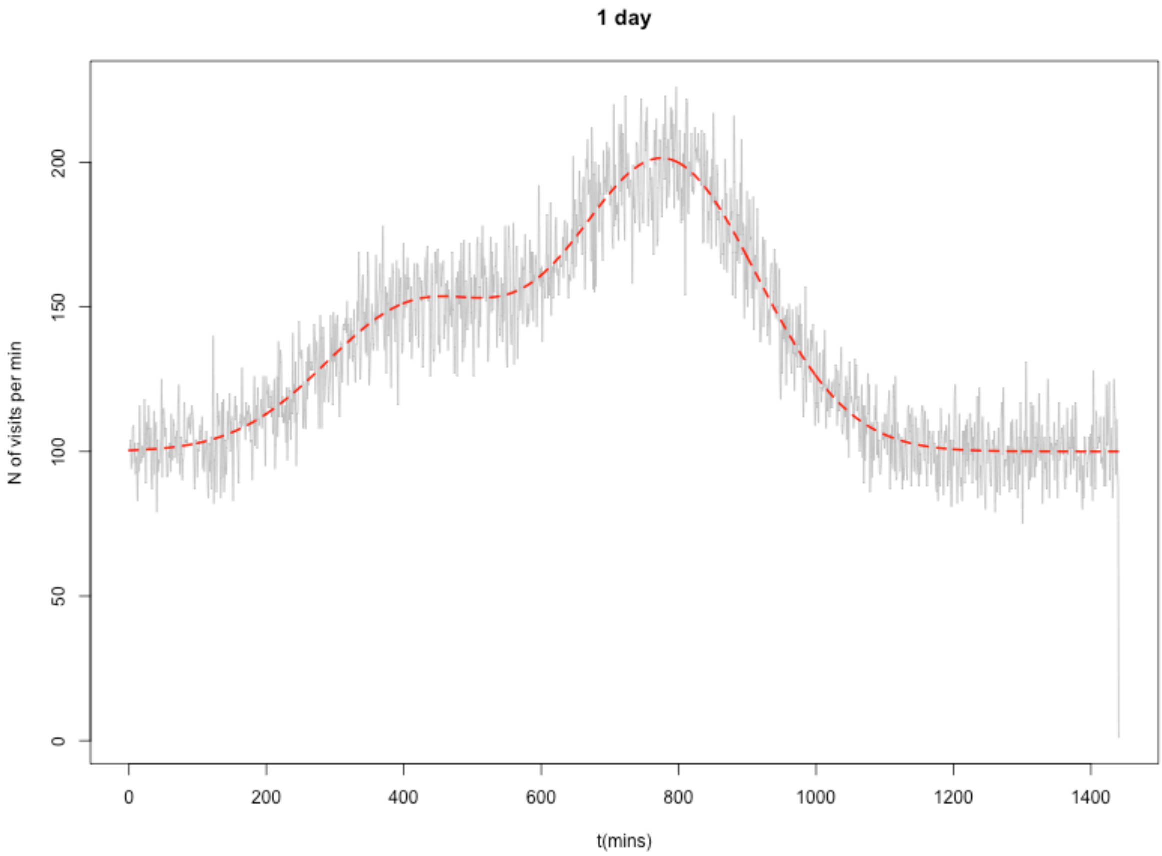


2.III. Plotting the observed arrival counts for each minute during the entire day

R commands according to the instructions:

```
png(filename = "2.3.png", width = 800, height = 600);
h = hist(arrival_times, breaks = 24 * 60, freq = TRUE, plot = FALSE, warn.unused =
FALSE);
plot(h$mids, h$counts, col = "gray", type = 'l', xlab = "t(mins)", ylab = "N of visits
per min", main = "1 day");
curve(lambda(x), from = 0, to = 24 * 60, n = 24 * 60, add = TRUE, lty = "dashed", col =
"red", lwd = 2);
graphics.off();
```

- Output



We can see that graph of arrival counts generated during each minute of the day reminds graph of rate lambda(t).

3. Simulation of an internet shop with orders arriving according to a non-homogeneous Poisson process

3.I. Data given

Assuming that K/(K+L)*100% of all placed orders will use a courier delivery, while the rest will use a state postal service.

No task given.

3.II. Splitting arrival process into two processes

R commands according to the instructions:

```
atc = numeric();
atp = numeric();
for (i in 1:length(arrival_times))
{
        t = runif(1, min = 0, max = 1);
        if (t < K / (K + L))
                atc = append(atc, arrival_times[i])
        else
                atp = append(atp, arrival_times[i]);
}
```

- Output

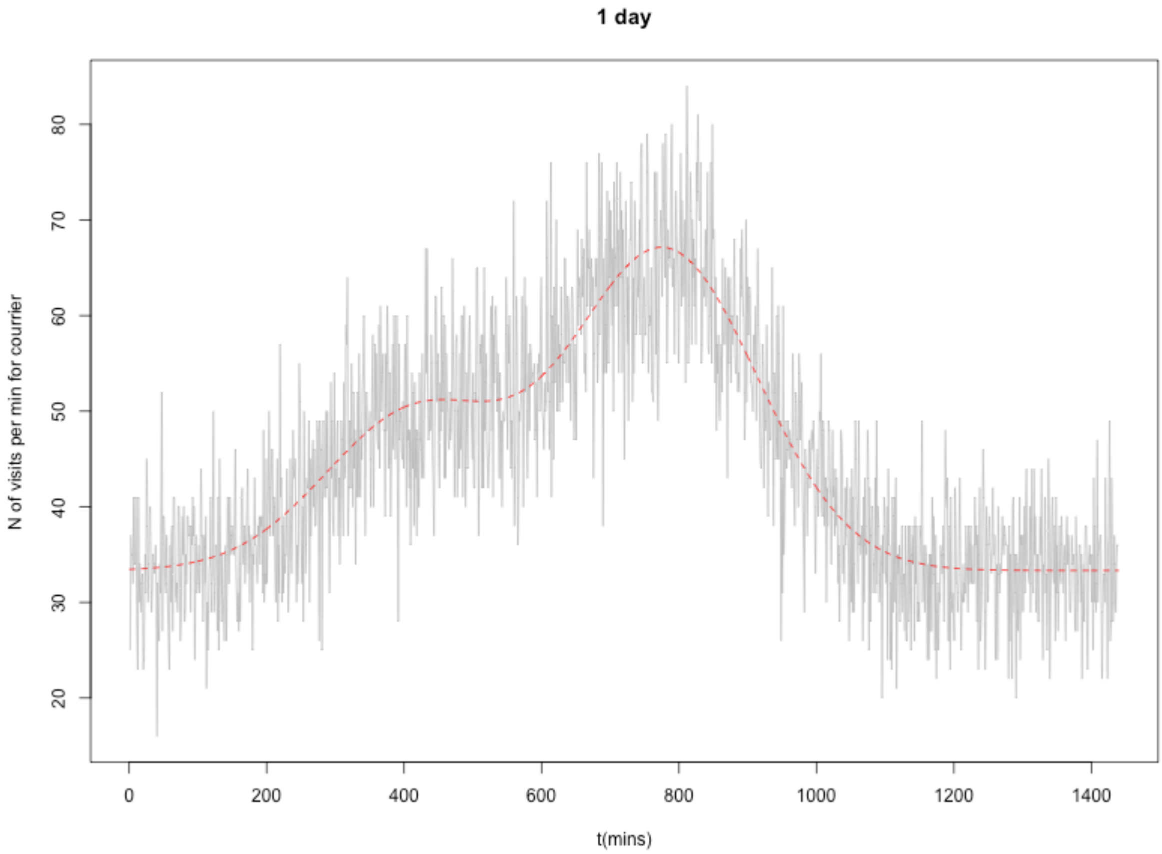
```
We've got two processes out of one according to given probability
```

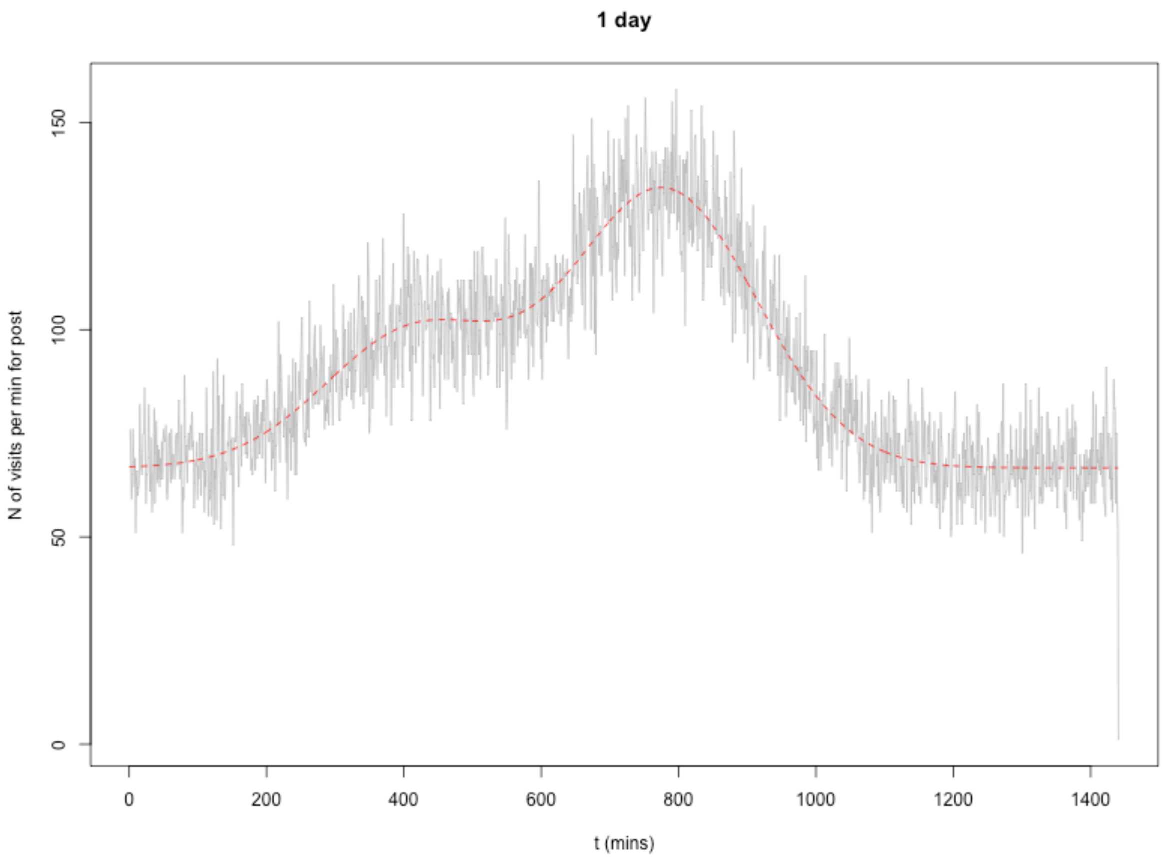
3.III. Plotting the observed arrival counts separately for each process

R commands according to the instructions:

```
png(filename = "3.1.png", width = 800, height = 600);
h = hist(atc, breaks = 24 * 60, freq = TRUE, plot = FALSE, warn.unused = FALSE);
plot(h$mids, h$counts, col = "gray", type = 'l', xlab = "t(mins)", ylab = "N of visits
per min for courrier", main = "1 day");
curve(lambda(x) * (K / (K + L)), from = 0, to = 24 * 60, n = 24 * 60, add = TRUE, col =
'red', lty = 2);
graphics.off();

png(filename = "3.2.png", width = 800, height = 600);
h = hist(atp, breaks = 24 * 60, freq = TRUE, plot = FALSE, warn.unused =
                FALSE);
plot(h$mids, h$counts, col = "gray", type = 'l', xlab = "t (mins)", ylab = "N of visits
per min for post", main = "1 day");
curve(lambda(x) * (L / (K + L)), from = 0, to = 24 * 60, n = 24 * 60, add = TRUE, col =
'red', lty = 2);
graphics.off();
```

- Output





We can see that issued processes reminds their theoretical functions.