ΣΥΣΤΗΜΑΤΑ ΑΝΑΜΟΝΗΣ

<u>1η ΟΜΑΔΑ ΑΣΚΗΣΕΩΝ</u>

ΘΟΔΩΡΗΣ ΑΡΑΠΗΣ – EL18028

30 ΑΠΡΙΛΙΟΥ, 2023

Ονοματεπώνυμο: Θοδωρής Αράπης	**AM:** el18028

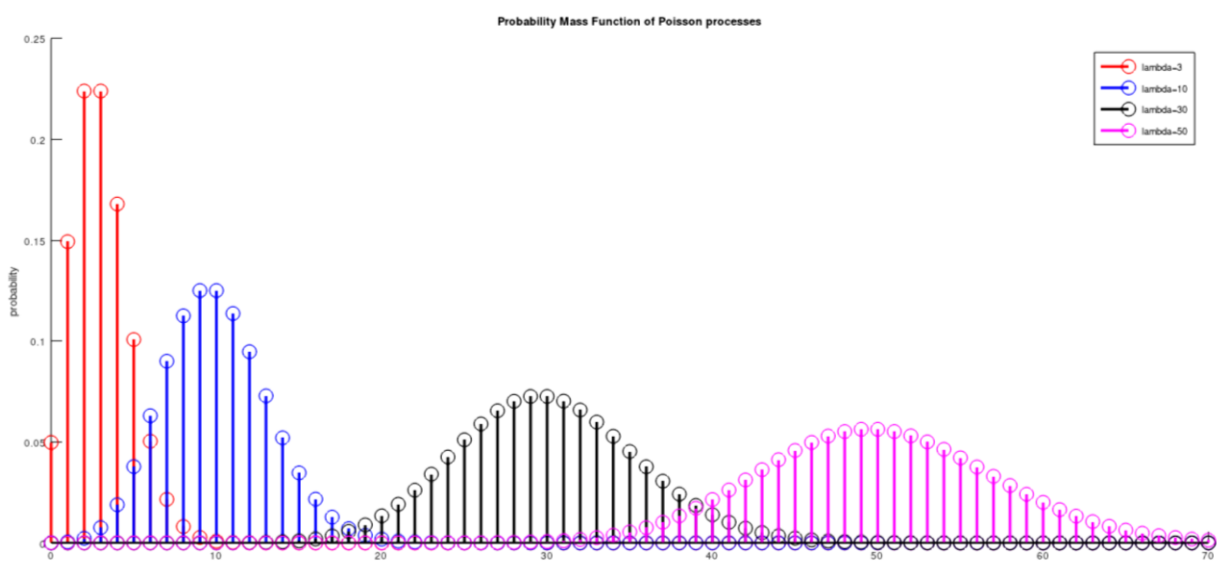
Κατανομή Poisson

A)

Αποθηκεύουμε των ακόλουθο κώδικα σε ένα αρχείο τον τρέχουμε:

```
clc;
clear all;
close all;

# Κατανομή Poisson
# Task A

k = 0:1:70;
lambda = [3, 10, 30, 50];

for i = 1 : columns(lambda)
  poisson(i, :) = poisspdf(k, lambda(i));
endfor

colors = "rbkm";
figure(1);
hold on;
for i = 1 : columns(lambda)
  stem(k, poisson(i, :), colors(i), "linewidth", 1.2);
endfor
hold off;

title("Probability Mass Function of Poisson processes");
xlabel("k values");
ylabel("probability");
legend("lambda=3", "lambda=10", "lambda=30", "lambda=50");
```

Λαμβάνουμε το εξής plot των συναρτήσεων μαζών πιθανότητας για τις ζητούμενες τιμές λ:



Γνωρίζουμε ότι για μία κατανομή $X \sim Poisson(\lambda)$ ισχύει:

$$E(x) = \lambda \; (Μέση\ Τιμή)$$

$$Var(x) = \lambda \; (Διασπορά)$$

Από τις παραπάνω σχέσεις συμπεραίνουμε ότι η μέση τιμή και η διασπορά είναι ίσες με την παράμετρο λ, άρα αναμένουμε για μεγαλύτερα λ τα μέγιστα των γραφικών παραστάσεων να βρίσκονται δεξιότερα, με όλο και μικρότερες τιμές, ενώ ακόμα οι γραφικές να φαίνονται πιο «απλωμένες» ως προς τον άξονα x.

Κοιτώντας τις γραφικές παραστάσεις εύκολα επιβεβαιώνουμε τα συμπεράσματά μας. Πιο αναλυτικά, έχοντας ορίσει $k = 1, 2, 3, \ldots, 70$, θέλουμε να ισχύει:

$$\sum_{k=0}^{70} P[X = k] = 1$$

Ως εκ τούτου, όσο μικρότερο μέγιστο έχει η γραφική, τόσο θα πρέπει οι γειτονικές πιθανότητες να παρουσιάζονται πιο ομοιόμορφα κατανεμημένες ώστε να αντισταθμίσουν την διαφορά.

B)

Στο προηγούμενο αρχείο που φτιάξαμε προσθέτουμε τις παρακάτω γραμμές κώδικα και λαμβάνουμε το αποτέλεσμα που ακολουθεί:

```
# Task B

index = find(lambda == 30);
chosen = poisson(index, :);
mean_value = 0;

for i=0:(columns(poisson(index, :)) - 1)
  mean_value = mean_value + i .* poisson(index,i+1);
endfor

display("mean value of Poisson with lambda 30 is");
display(mean_value);

second_moment = 0;
for i = 0 : (columns(poisson(index, :)) - 1)
  second_moment = second_moment + i .* i .* poisson(index, i + 1);
endfor

variance = second_moment - mean_value .^ 2;
display("Variance of Poisson with lambda 30 is");
display(variance);
```

```
mean value of Poisson with lambda 30 is
mean_value = 30.000
Variance of Poisson with lambda 30 is
variance = 30.000
```

Είναι γνωστό από την θεωρία ότι οι σχέσεις για τη μέση τιμή και τη διακύμανση είναι αντίστοιχα:

$$E[X] = \sum k \cdot P[X = k] \ \ \kappa\alpha\iota$$

$$V[X] = E[X^2] - (E[X])^2,$$

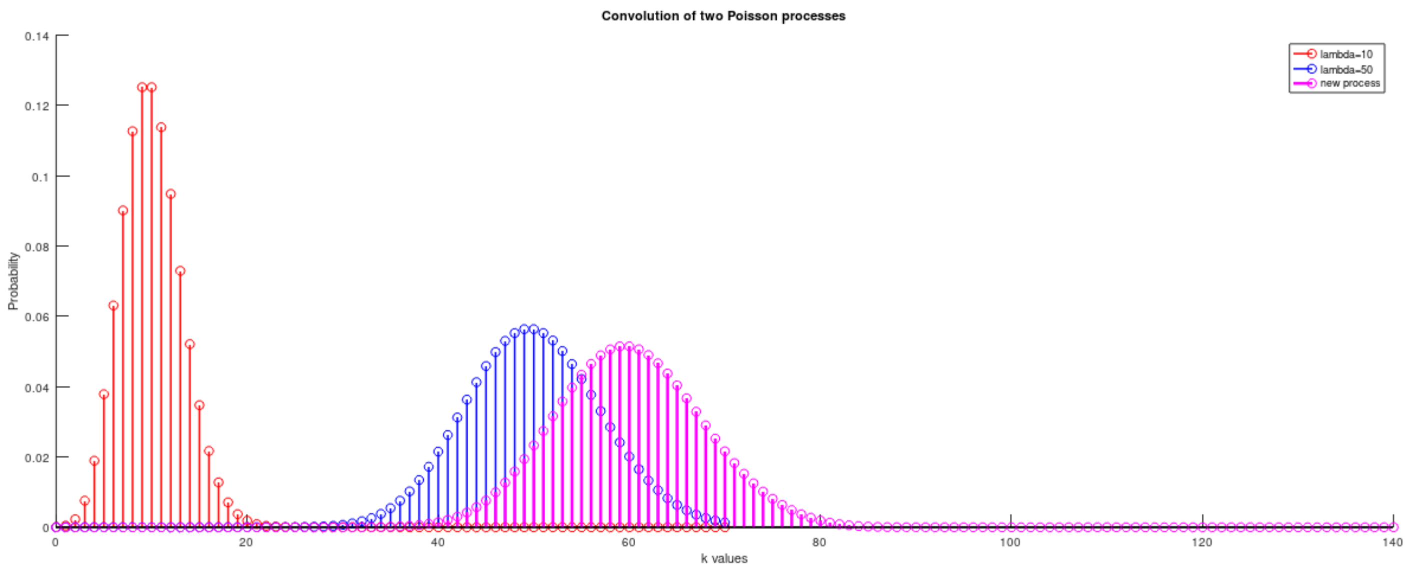
όπου $k \in [0, 70] \subseteq Z$ στην περίπτωσή μας. Εκτελώντας τις πράξεις, καταλήγουμε στα δύο μεγέθη να ισούνται με την παράμετρο λ, όπως υποδείξαμε και στο προηγούμενο ερώτημα. Επομένως, όπως αναμέναμε, για $\lambda = 30$ έχουμε $E[X] = 30$ και $V[X] = 30$.

Γ)

Στο ίδιο αρχείο με πριν προσθέτουμε τον ακόλουθο κώδικα:

```
# Task Γ

first = find(lambda == 10);
second = find(lambda == 50);
poisson_first = poisson(first, :);
poisson_second = poisson(second, :);

composed = conv(poisson_first, poisson_second);
new_k = 0 : 1 : (2 * 70);

figure(2);
hold on;
stem(k, poisson_first(:), colors(1), "linewidth", 1.2);
stem(k, poisson_second(:), colors(2), "linewidth", 1.2);
stem(new_k, composed, "mo", "linewidth", 2);
hold off;
title("Convolution of two Poisson processes");
xlabel("k values");
ylabel("Probability");
legend("lambda=10", "lambda=50", "new process");
```

Τρέχουμε το συνολικό και λαμβάνουμε τις ακόλουθες γραφικές παραστάσεις:



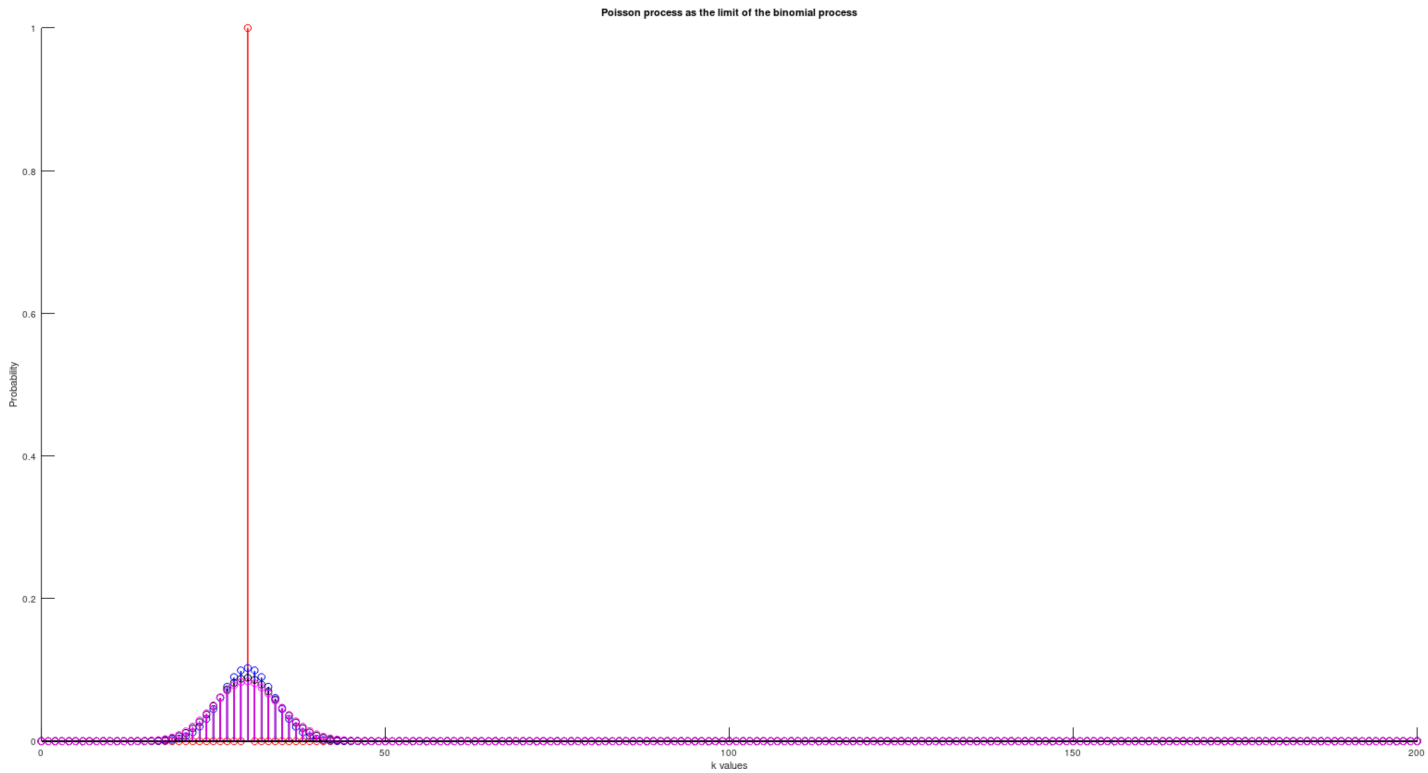
Παρατηρούμε ότι η συνέλιξη (υπέρθεση) των δύο κατανομών Poisson δίνει ως αποτέλεσμα μια νέα κατανομή Poisson με μεγαλύτερη μέση τιμή και διασπορά από τις άλλες δύο. Αυτό είναι αποτέλεσμα της σχέσης: $\boldsymbol{\lambda_{new} = \lambda_{10} + \lambda_{50}}$, η οποία ισχύει **μόνο** όταν οι προς συνέλιξη κατανομές είναι μεταξύ τους ανεξάρτητες.

Δ)

Προσθέτουμε τώρα το εξής κομμάτι κώδικα:

```
# Task Δ

k = 0 : 1 : 200;

# Define the desired Poisson Process
lambda = 30;
i = 1 : 1 : 5;
n = lambda .* i;
p = lambda ./ n;

figure(3);
title("Poisson process as the limit of the binomial process");
xlabel("k values");
ylabel("Probability");
hold on;
for i = 1 : 4
  binomial = binopdf(k, n(i), p(i));
  stem(k, binomial, colors(i), 'linewidth', 1.2);
endfor
hold off;
```

Η γραφική που προκύπτει είναι:



Έστω μία θετική παράμετρος λ (σταθερά). Για κάθε $n \in \mathbb{N}$ ορίζουμε μία Τυχαία Μεταβλητή Y_n με Διωνυμική κατανομή (n, p) και πυκνότητα $P_n(k) = P[Y_n = k], για\ k = 0,1,2,\ldots,n$. Καθώς $n \longrightarrow \infty$, οι τιμές της πυκνότητας$P_n(k)$ συγκλίνουν στις αντίστοιχες τιμές της πυκνότητας $P(k)$ μίας Τ.Μ. X με κατανομή Poisson με παράμετρο λ. Ως εκ τούτου, για μεγάλο πλήθος γεγονότων $(n \geq 100)$ και μικρής πιθανότητας $p = \frac{\lambda}{n}$ $(p < 0.04)$, το γινόμενο $n \cdot p$ είναι της τάξης του 1 και μπορούμε με καλή ακρίβεια να προσεγγίσουμε μία διωνυμική κατανομή ως μία κατανομή Poisson.

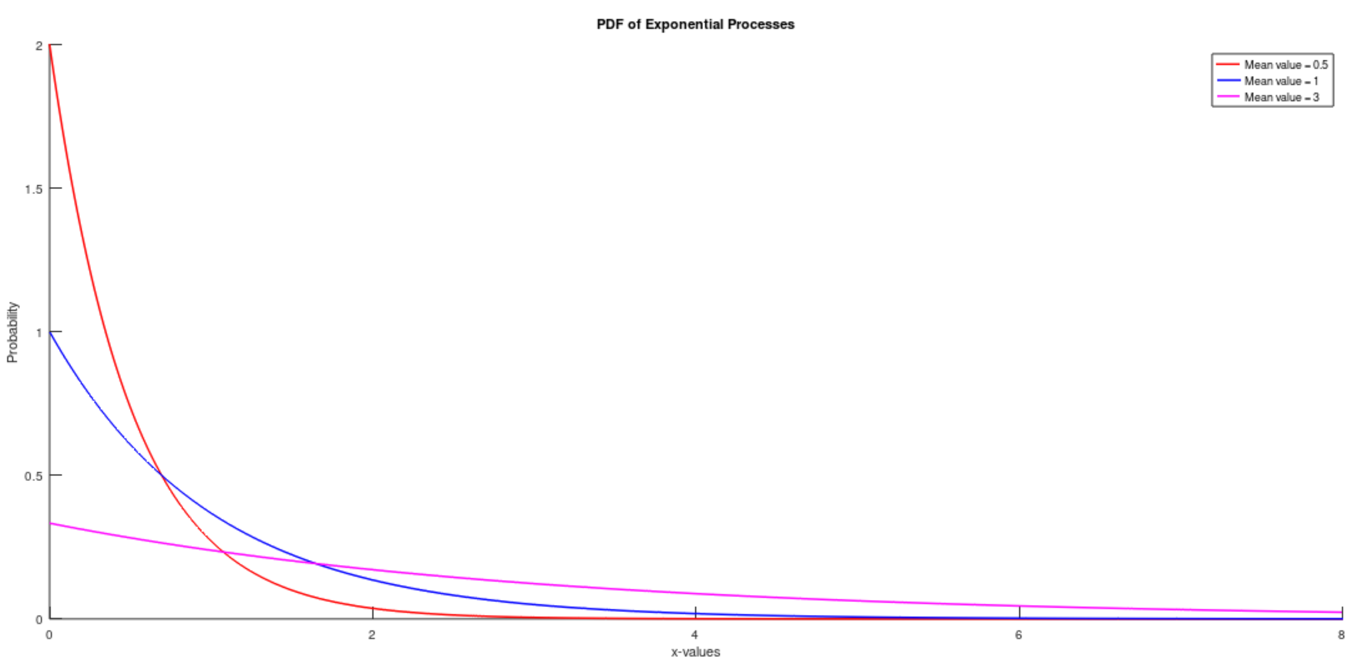
<u>Εκθετική κατανομή</u>

A)

Σχεδιάζουμε τη Συνάρτηση Πυκνότητας Πιθανότητας για εκθετική κατανομή με μέσο όρο $\left(\frac{1}{\lambda}\right) = \{0.5, 1, 3\}$, χρησιμοποιώντας τον ακόλουθο κώδικα:

```
pkg load statistics

clc;
clear all;
close all;

# Εκθετική κατανομή
# Task A

k = 0 : 0.0001 : 8;
mean_value = [0.5, 1, 3]; # (1/lambda)

for i = 1 : columns(mean_value)
  exponential_pdf(i, :) = exppdf(k, mean_value(i));
endfor

colors = "rbm";
figure(1);
hold on;
for i = 1 : columns(mean_value)
  plot(k, exponential_pdf(i, :), colors(i), "linewidth", 1.2);
endfor

hold off
title("PDF of Exponential Processes");
xlabel("x-values");
ylabel("Probability");
legend("Mean value = 0.5", "Mean value = 1", "Mean value = 3");
```

Η γραφική παράσταση που προκύπτει είναι:

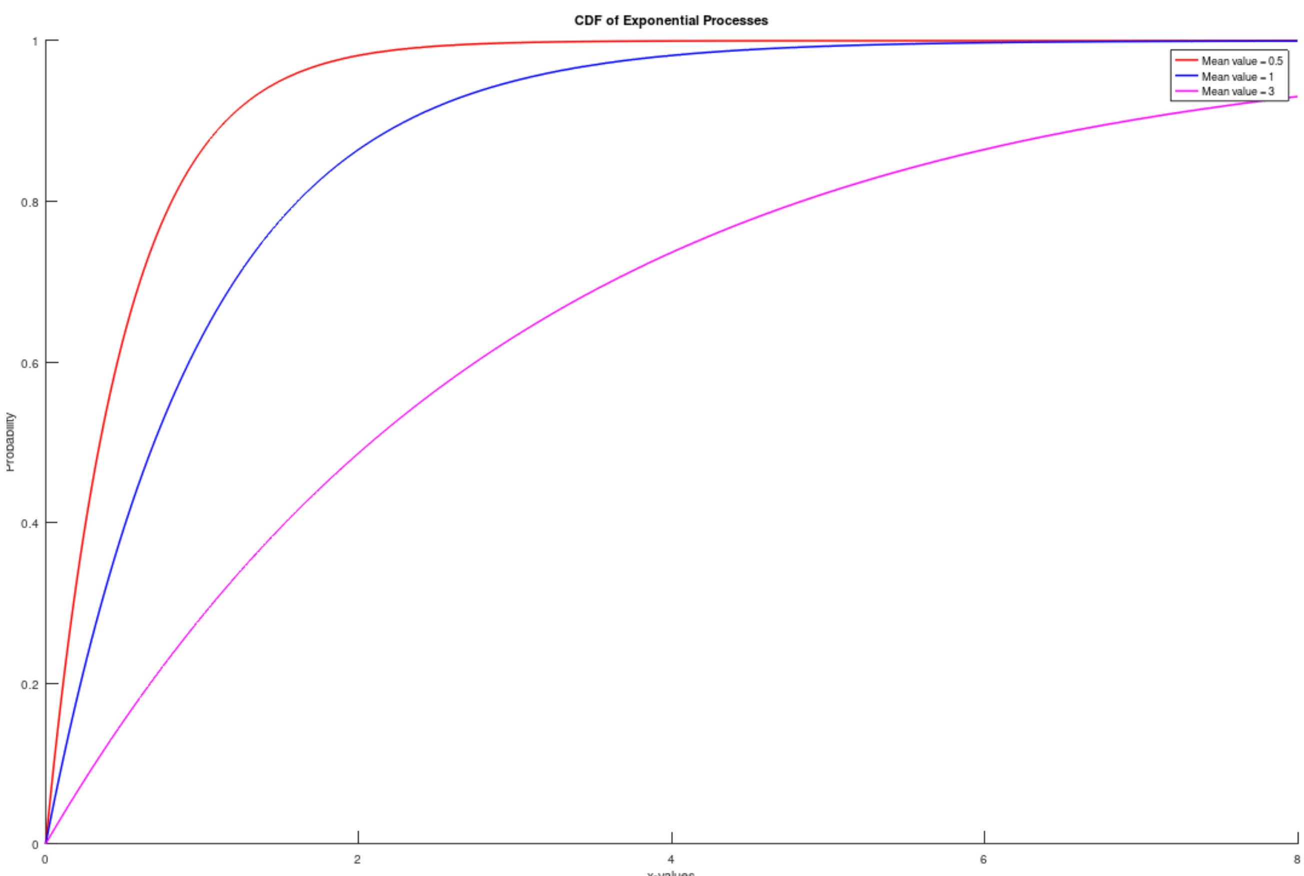


B)

Σχεδιάζουμε τώρα την Αθροιστική Συνάρτηση Κατανομής (CDF) των εκθετικών κατανομών που φτιάξαμε προηγουμένως. Συνεπώς στον προηγούμενο κώδικα προσθέτουμε τις εξής γραμμές:

```
#Task B

for i = 1 : columns(mean_value)
  exponential_cdf(i, :) = expcdf(k, mean_value(i));
endfor

figure(2);
hold on
for i = 1 : columns(mean_value)
  plot(k, exponential_cdf(i, :), colors(i), "linewidth", 1.2);
endfor
hold off

title("CDF of Exponential Processes");
xlabel("x-values");
ylabel("Probability");
legend("Mean value = 0.5", "Mean value = 1", "Mean value = 3");
```

Οι ζητούμενες γραφικές είναι:



Γ)

Εφόσον δουλεύουμε με Εκθετικές Κατανομές ισχύει:

$$F[x] = 1 - e^{-\lambda x}, \qquad x \geq 0$$

Συνεπώς για τις πιθανότητες $P[X > 30000]$ και $P[X > 50000 \mid X > 20000]$ έχουμε:

$$\boldsymbol{P[X > 30000]} = 1 - P[X \leq 30000] = 1 - F[30000] = 1 - \left(1 - e^{-\frac{30000}{2.5}}\right) = e^{-12000}$$

$$\boldsymbol{P[X > 50000 \mid X > 20000]} = \frac{P[X > 50000] \wedge P[X > 20000]}{P[X > 20000]} = \frac{P[X > 50000]}{P[X > 20000]} =$$

$$\frac{1 - P[X \leq 50000]}{1 - P[X \leq 20000]} = \frac{1 - F[50000]}{1 - F[20000]} = \frac{1 - \left(1 - e^{-\frac{50000}{2.5}}\right)}{1 - \left(1 - e^{-\frac{20000}{2.5}}\right)} = \frac{e^{-\frac{50000}{2.5}}}{e^{-\frac{20000}{2.5}}} = e^{-12000}$$

Εύκολα παρατηρούμε πως οι 2 πιθανότητες δίνουν το ίδιο αποτέλεσμα (e^{-12000}). Το αποτέλεσμα αυτό προκύπτει λόγω της ιδιότητας απώλειας μνήμης της εκθετικής κατανομής. Πιο ειδικά, η ιδιότητα λέει:

$$P[X \geq \alpha + \beta \mid X \geq \alpha] = P[X \geq \beta], \qquad \alpha, \beta > 0$$

Με άλλα λόγια, η δεσμευμένη πιθανότητα της μορφής $P[X \geq \alpha + \beta \mid X \geq \alpha]$ εξαρτάται μόνο από την τιμή του β. Άρα στην περίπτωσή μας έχουμε $\alpha = 20000, \beta = 30000$:

$$P[X > 50000 \mid X > 20000] = P[X > 20000 + 30000 \mid X > 20000] =$$

$$P[X > \alpha + \beta \mid X > \alpha] = P[X > \beta] = P[X > 30000] = e^{-12000}$$

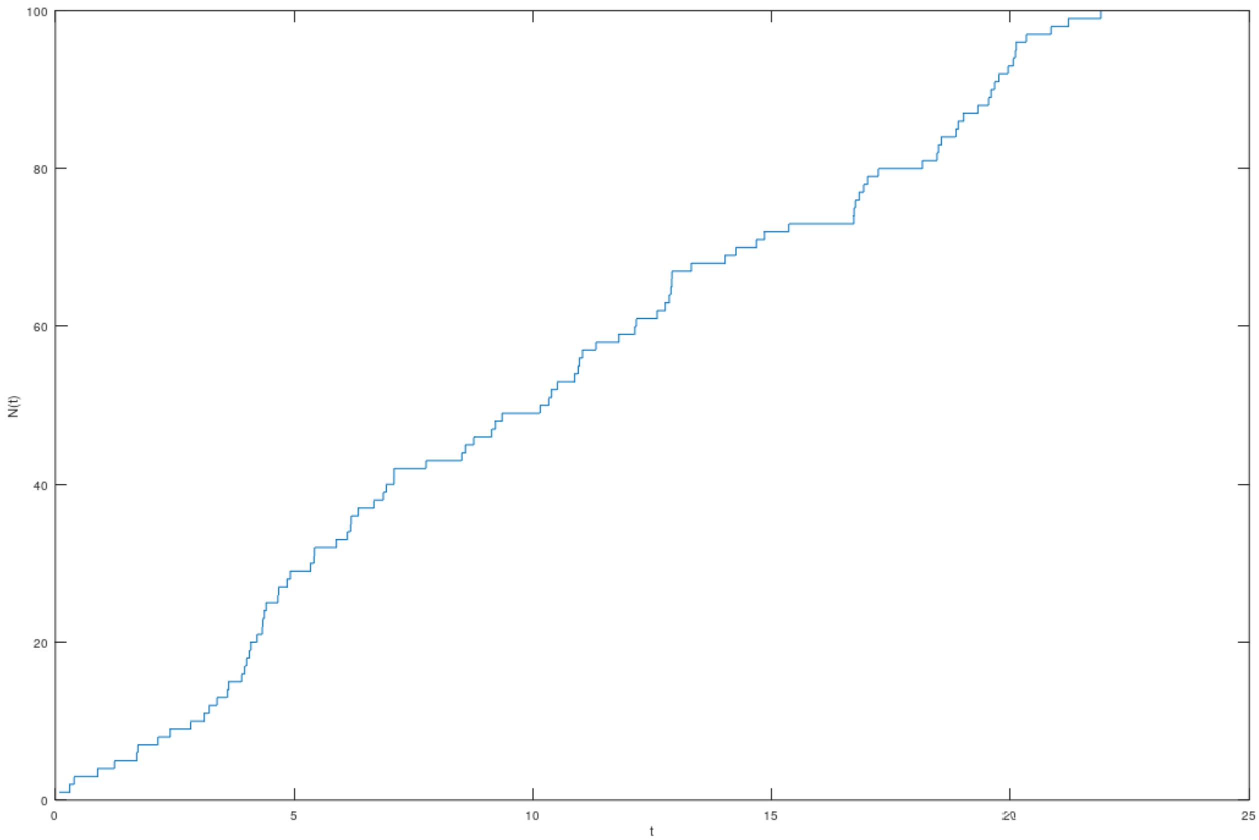
<u>Διαδικασία Καταμέτρησης Poisson</u>

A)

Όπως γνωρίζουμε από την θεωρία, οι χρόνοι που μεσολαβούν ανάμεσα στην εμφάνιση δύο διαδοχικών γεγονότων Poisson ακολουθούν εκθετική κατανομή. Συνεπώς, είναι α-νεξάρτητοι από το παρελθόν. Με τον ακόλουθο κώδικα, σχεδιάζουμε μια διαδικασία κα-ταμέτρησης Poisson 100 τυχαίων γεγονότων, όπου θεωρούμε πως σε κάθε γεγονός έ-χουμε μοναδιαία αύξηση (μία προσθήκη στην ουρά):

```
pkg load statistics

clc;
clear all;
close all;

# Διαδικασία Καταμέτρησης Poisson
# Task A

N = 100;                # Number of successive events
lambda = 5;
mean = 1/lambda;
succ_events = exprnd(mean, 1, N);
sum_succ_events = succ_events;

for i = 1 : (N - 1)
  sum_succ_events(1, i+1) = sum_succ_events(1, i) + succ_events(1, i+1);
endfor

y = 1 : N;
stairs(sum_succ_events(1,:), y);
xlabel("t");
ylabel("N(t)");
```

Η ζητούμενη γραφική παράσταση είναι:



B)

Είναι γνωστό από την θεωρία ότι ο αριθμός των γεγονότων σε ένα διάστημα $(t, t+T)$ αποτελεί μια διακριτή Τ.Μ., που περιγράφεται από την σχέση: $\nu = N(t+T) - N(t)$. Επιπλέον, στην περίπτωσή μας δεν υπάρχει εξάρτηση από το παρελθόν μεταξύ των τυχαίων γεγονότων, επομένως, με βάση τη θεωρία, η Τ.Μ. ν θα ακολουθεί κατανομή Poisson. Συνεπώς, Ο μέσος αριθμός εμφανίσεων των γεγονότων θα ισούται με λT, οπότε ο μέσος και μέσος αριθμός εμφανίσεων στην μονάδα του χρόνου θα είναι ίσος με $\frac{\lambda T}{T} = 5$ events/sec. Για να ελέγξουμε τα συμπεράσματά μας γράφουμε τον κώδικα που ακολουθεί στην επόμενη σελίδα για να υπολογίσουμε το μέσο αριθμό εμφανίσεων στην μονάδα του χρόνου για τα ζητούμενα πλήθη τυχαίων γεγονότων. Τα αποτελέσματα που λαμβάνουμε συμφωνούν με τις παρατηρήσεις μας καθώς με την αύξηση του πλήθους των γεγονότων η τιμή πλησιάζει όλο και περισσότερο το λ:

Command Window

```
Mean number of successive events for N = 100 in a time period T is: 5.738122
Mean number of successive events for N = 200 in a time period T is: 5.433974
Mean number of successive events for N = 300 in a time period T is: 5.003122
Mean number of successive events for N = 500 in a time period T is: 4.872015
Mean number of successive events for N = 1000 in a time period T is: 5.033375
Mean number of successive events for N = 10000 in a time period T is: 4.998422
```

```
pkg load statistics

clc;
clear all;
close all;

# Διαδικασία Καταμέτρησης Poisson
# Task B

N = [100,  200, 300, 500, 1000, 10000];                # Number of successive events
lambda = 5;
mean = 1/lambda;

for i = 1 : columns(N)
  succ_events = exprnd(mean, 1, N(i));                 # random samples
  sum_succ_events = succ_events;
  for j = 1 : (N(i)-1)
    sum_succ_events(1, j+1) = sum_succ_events(1, j) + succ_events(1, j+1);
  endfor

   mean_succ_events = N(i)/sum_succ_events(1, N(i));     # means
   printf("Mean number of successive events for N = %d in a time period T is: %f \n", N(i), mean_succ_events)
endfor
```