ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ & ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΕΡΓΑΣΤΗΡΙΟ ΔΙΑΧΕΙΡΙΣΗΣ ΚΑΙ ΒΕΛΤΙΣΤΟΥ ΣΧΕΔΙΑΣΜΟΥ ΔΙΚΤΥΩΝ ΤΗΛΕΜΑΤΙΚΗΣ - NETMODE

Συστήματα Αναμονής

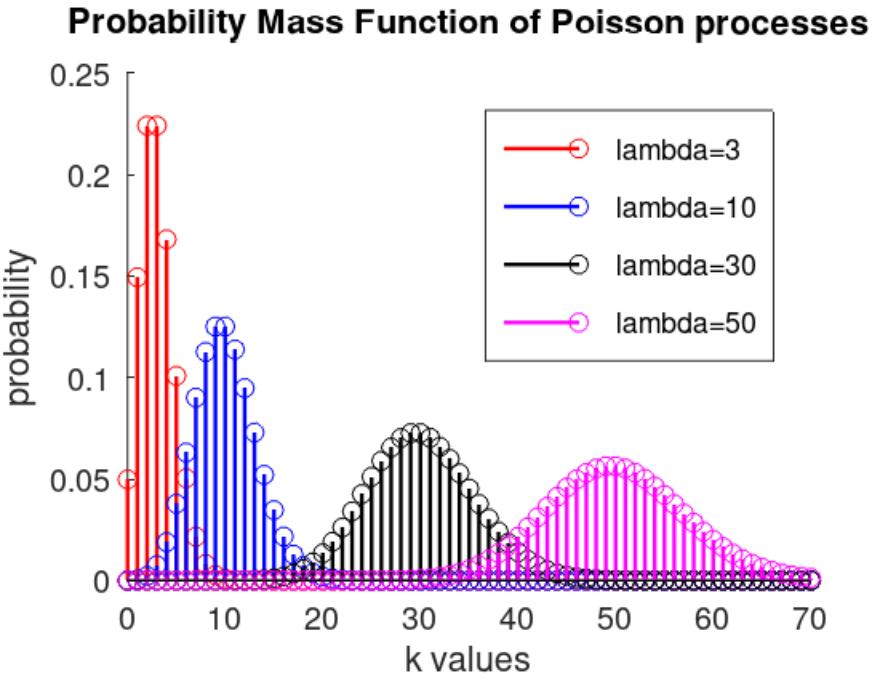
1η ΟΜΑΔΑ ΑΣΚΗΣΕΩΝ | ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2022-2023

ΟΝΟΜΑΤΕΠΩΝΥΜΟ : Κουστένης Χρίστος

A.M : el20227

Κατανομή Poisson

Α) Οι γραφικές που προκύπτουν εκτελώντας τον κώδικα που αντιστοιχεί σε αυτό το ερώτημα της ενότητας αυτής είναι οι ακόλουθες:



Για τη σχεδίαση των γραφικών εξυπηρετεί η συνάρτηση stem() αφού θέλουμε να απεικονίσουμε την συνάρτηση μάζας πιθανότητας διαδικασιών Poisson όπου ο χώρος ενδεχομένων είναι διακριτός.

Η συνάρτηση Poisson έχει τύπο : $P(x) = \frac{e^{-\lambda}\lambda^x}{x!}$.

Προκύπτει λοιπόν ότι έχει μέση τιμή : $E(X) = \lambda$

και διακύμανση : $Var(X) = \lambda$

Επομένως, αυξάνοντας την τιμή του λ παρατηρούμε ότι μετατοπίζεται προς τα δεξιά το ολικό μέγιστο των κατανομών αφού αυξάνεται η μέση τιμή λ(και άρα η πιθανότερη τιμή που λαμβάνει η τυχαία μεταβλητή), όμως συγχρόνως μειώνεται η πιθανότητα P[X = λ] αφού αύξηση του λ συνεπάγεται και αύξηση της διακύμανσης της συνάρτησης μάζας πιθανότητας και άρα η πιθανότητα ενδεχομένων του δειγματικού χώρου απλώνεται κατά μήκος του οριζόντιου άξονα αφού η συνολική πιθανότητα των ενδεχομένων πρέπει να αθροίζει πάντα στη μονάδα.

Β)

Στο ερώτημα αυτό ασχολούμαστε με την κατανομή Poisson.

Η μέση τιμή της κατανομής Poisson υπολογίζεται ως εξής:

$\mu = E(X) = \sum_{x=0}^{\infty} x \frac{\lambda^x e^{-\lambda}}{x!} = \lambda \sum_{x=1}^{\infty} \frac{\lambda^{x-1} e^{-\lambda}}{(x-1)!} = \lambda e^{-\lambda} \sum_{x=0}^{\infty} \frac{\lambda^x}{x!} = \lambda = 30$

Η διακύμανση κατανομής Poisson υπολογίζεται ως εξής:

$$Var(X) = E[(X-\mu)^2] = E[X^2 - 2\mu X + \mu^2] = E[X^2] - 2\mu E[X] + \mu^2 = \sum_{x=0}^{\infty} x^2 \frac{\lambda^x e^{-\lambda}}{x!} - 2\lambda \sum_{x=0}^{\infty} x \frac{\lambda^x e^{-\lambda}}{x!} + \lambda^2 =$$

$$\lambda e^{-\lambda} \sum_{x=0}^{\infty} x \frac{\lambda^{x-1}}{(x-1)!} - 2\lambda e^{-\lambda} \sum_{x=0}^{\infty} \frac{\lambda^x}{x!} + \lambda^2 = \lambda e^{-\lambda} \left[\lambda \sum_{x=1}^{\infty} \frac{\lambda^{x-1}}{(x-1)!} + \sum_{x=1}^{\infty} \frac{\lambda^x}{(x-1)!}\right] - 2\lambda^2 + \lambda^2 = \lambda e^{-\lambda}\left(\lambda e^{\lambda} + e^{\lambda}\right) - \lambda^2 = \lambda$$

Ο υπολογισμός των παραπάνω τιμών γίνεται στο Octave με τη χρήση του παρακάτω κώδικα:

```
# TASK: regarding the poisson process with parameter lambda 30, compute its mean
# value and variance

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);

```

Στο command window παρατηρούμε το αποτέλεσμα της παρακάτω εικόνας:

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

Επομένως μέση τιμή και συνδιακύμανση είναι ίσες με $\lambda = 30$ και οι δύο όπως αναμενόταν από τον θεωρητικό υπολογισμό τους.

Γ) Έστω X_1 *και* X_2 δύο τυχαίες μεταβλητές που ακολουθούν κατανομή Poisson με παραμέτρους λ_1 και λ_2, αντίστοιχα. Θέλουμε να δείξουμε ότι η τυχαία μεταβλητή που αποτελεί προκύπτει ως υπέρθεση των δύο κατανομών με $Y = X_1 + X_2$ έχει παράμετρο $\lambda = \lambda_1 + \lambda_2$

$$P(Y = y) = P(X_1 + X_2 = y)$$

$$= \sum_{k=0}^{y} P(X_1 = k) P(X_2 = y - k) =$$

$$= \sum_{k=0}^{y} \frac{e^{-\lambda_1} \lambda_1^k}{k!} \cdot \frac{e^{-\lambda_2} \lambda_2^{y-k}}{(y-k)!} =$$

$$= \frac{e^{-(\lambda_1+\lambda_2)}}{y!} \sum_{k=0}^{y} \frac{y!}{k!\,(y-k)!} \lambda_1^k \lambda_2^{y-k} =$$

$$= \frac{e^{-(\lambda_1+\lambda_2)}}{y!} \sum_{k=0}^{y} \binom{y}{k} \lambda_1^k \lambda_2^{y-k}$$

$$= \frac{e^{-(\lambda_1+\lambda_2)}}{y!}(\lambda_1+\lambda_2)^y$$

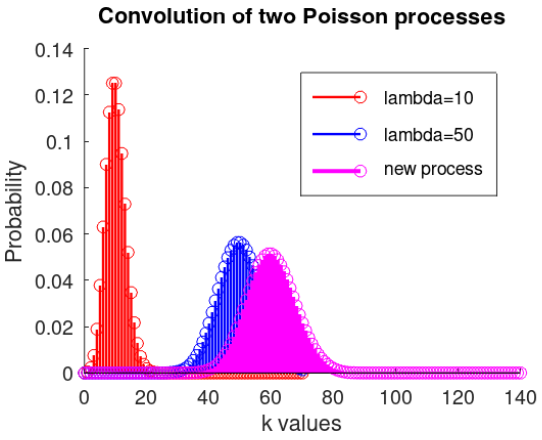
Επομένως η τυχαία μεταβλητή Y ακολουθεί Poisson με $\lambda = \lambda_1 + \lambda_2$.

Παρατηρούμε ότι η $P(Y = y)$ εκφράζεται ως συνέλιξη των επιμέρους κατανομών στη δεύτερη σειρά της απόδειξης.

Πραγματοποιώντας συνέλιξη των κατανομών στο Octave με χρήση του παρακάτω κώδικα προκύπτει μια γραφική της οποίας η κορυφή είναι πιο δεξιά και πιο χαμηλά από τις κορυφές των κατανομών που συνελίξαμε αφού σύμφωνα με τη θεωρία $\lambda = \lambda1+\lambda2 = 50 + 10 = 60$.

```
# TASK: consider the convolution of the Poisson distribution with lambda 10 with
# the Poisson distribution with lambda 50.

first = find(lambda == 10); # finds the row number where lambda == 10
second = find(lambda == 50); # finds the row number where 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");
```

Το αντίστοιχο figure φαίνεται παρακάτω:



Δ)

Έστω ότι έχουμε μια αλληλουχία δοκιμών Bernoulli, κάθε μια με πιθανότητα επιτυχίας p, και έστω X ο αριθμός επιτυχιών σε n δοκιμές. Η συνάρτηση μάζας πιθανότητας του X δίνεται από την διωνυμική κατανομή:

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

Επιλέγουμε χρονικό διάστημα [0, t] και το διαιρούμε σε n υποδιαστήματα ίσου μήκους Δt = t/n με Y το πλήθος των γεγονότων στο $[0, t]$ n ανεξάρτητων διαδικασιών Poisson με ρυθμό λ = np. Η συνάρτηση μάζας πιθανότητας θα δίνεται από την παρακάτω σχέση :

$$P(Y = k) = e^{-\lambda} \frac{\lambda^k}{k!}$$

Θα δείξουμε ότι

$$\lim_{n\to\infty} P(X = k) = P(Y = k)$$

$$P(X = k) = \lim_{n\to\infty} \binom{n}{k} p^k (1-p)^{n-k}$$

$$= \lim_{n\to\infty} \frac{n!}{k!\,(n-k)!} \left(\frac{\lambda}{n}\right)^k \left(1 - \frac{\lambda}{n}\right)^{n-k}$$

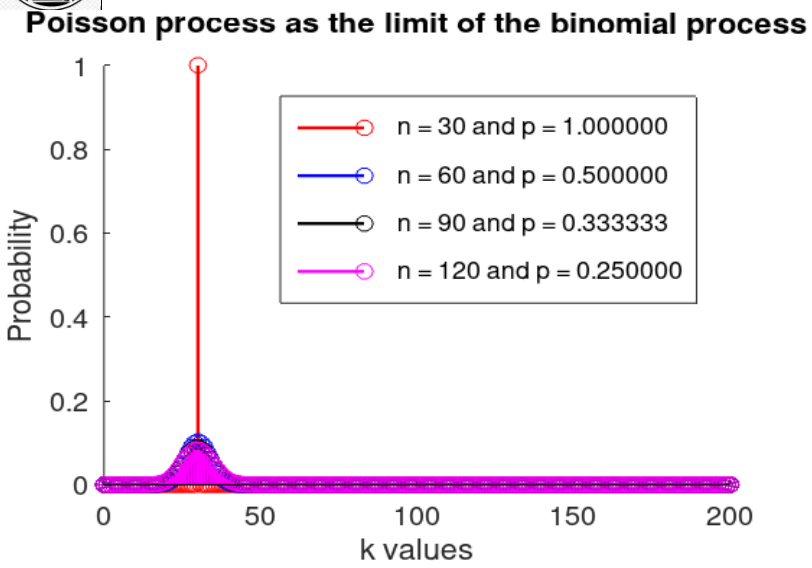
$$= \frac{\lambda^k}{k!} \lim_{n\to\infty} \frac{n(n-1)(n-2)\cdots(n-k+1)}{n^k} \left(1 - \frac{\lambda}{n}\right)^n \left(1 - \frac{\lambda}{n}\right)^{-k}$$

$$= \frac{\lambda^k}{k!} \lim_{n\to\infty} \left(1 - \frac{\lambda}{n}\right)^n \left(1 - \frac{\lambda}{n}\right)^{-k}$$

$$= \frac{\lambda^k}{k!} e^{-\lambda}$$

Κάνοντας χρήση της προσέγγισης του Stirling : $\lim_{n\to\infty} \left(1 + \frac{x}{n}\right)^n = e^x$ και της σχέσης $\lim_{n\to\infty} \frac{n(n-1)(n-2)\cdots(n-k+1)}{n^k} = 1$

Επομένως, πρακτικά η κατανομή Poisson αποτελεί μια ειδική περίπτωση της διωνυμικής κατανομής όπου ο αριθμός των δοκιμών είναι πολύ μεγάλος και η πιθανότητα επιτυχίας μικρή. Στην παρακάτω γραφική φαίνεται πως αυξάνοντας το πλήθος των δοκιμών και μειώνοντας την πιθανότητα επιτυχίας οι γραφικές συγκλίνουν στην κατανομή Poisson.



Ο κώδικας που χρησιμοποιήθηκε είναι ο ακόλουθος:

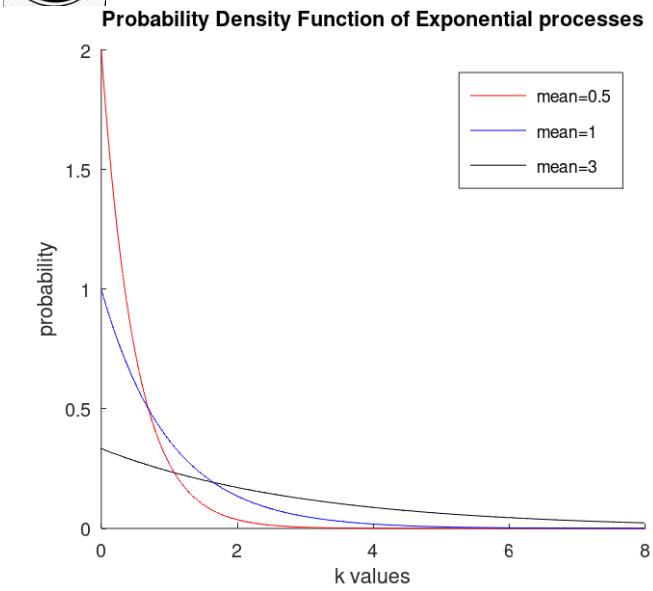
```
# TASK: show that Poisson process is the limit of the binomial distribution.
k = 0 : 1 : 200;
# Define the desired Poisson Process
lambda = 30;
i = 1 : 1 : 5;
n = lambda .* i;
p = lambda ./ n;

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
legend(sprintf("n = %d and p = %f",n(1),p(1)),sprintf("n = %d and p = %f",n(2),p(2)),sprintf("n = %d and p = %f",n(3),p(3)),sprintf("n = %d and p = %f",n(4),p(4)
hold off;
```

Εκθετική Κατανομή

Α) Στο ερώτημα αυτό καλούμαστε να σχεδιάσουμε την συνάρτηση πυκνότητας πιθανότητας(PDF) της εκθετικής κατανομής με μέσο όρο μ = 1/λ= {0.5, 1, 3} και τύπο:

$f(x;\lambda) = \lambda e^{-\lambda x}$ για $x \geq 0$, $f(x;\lambda) = 0$ για $x < 0$

Τα αποτελέσματα για διαφορετικές τιμές της παραμέτρου λ φαίνονται παρακάτω:



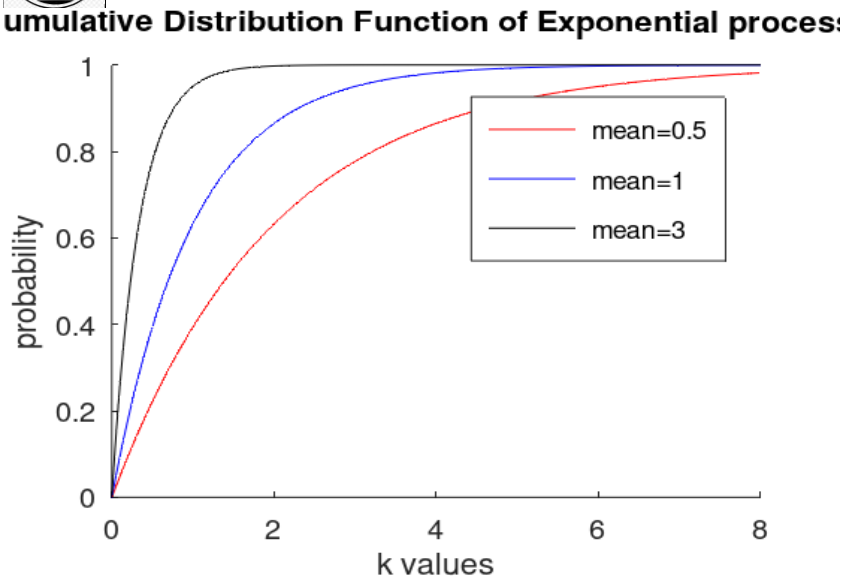
Ακολουθεί ο κώδικας που πάραξε το παραπάνω αποτέλεσμα:

```
# Exponential distribution

# Task A
m = [0.5 1 3];
lambda = 1./m;
k = 0:0.00001:8;

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

figure(1);
hold on;
colors = 'rbkm';
for i = 1 : columns(m)
  plot(k,exponential_pdf(i,:),colors(i));
endfor
hold off;
title("Probability Density Function of Exponential processes");
xlabel("k values");
ylabel("probability");
legend("mean=0.5", "mean=1","mean=3");
```

Β) Ακολουθούμε αντίστοιχη διαδικασία για τον σχεδιασμό της αθροιστικής συνάρτησης εκθετικής κατανομής(CDF) με τύπο : $F(x;\lambda) = 1 - e^{-\lambda x}$ για $x \geq 0$ $και$ $F(x;\lambda) = 0$ $για$ $x < 0$

για διαφορετικές τιμές της παραμέτρου λ.

Ακολουθούν οι αντίστοιχες γραφικές και ο ανάλογος κώδικας:




```
# Task B
m = [0.5 1 3];
lambda = 1./m;
k = 0:0.00001:8;
for i = 1 : columns(lambda)
  exponential_cdf(i,:) = expcdf(k,lambda(i));
endfor

figure(2);
hold on;
colors = 'rbkm';
for i = 1 : columns(lambda)
  plot(k,exponential_cdf(i,:),colors(i));
endfor
hold off;
title("Cumulative Distribution Function of Exponential processes");
xlabel("k values");
ylabel("probability");
legend("mean=0.5", "mean=1","mean=3");
```

Γ) Η ιδιότητα έλλειψης μνήμης της εκθετικής κατανομής αποδεικνύεται ως εξής:

$$P(X > s+t|X > s) = \frac{P(X > s+t \cap X > s)}{P(X > s)}$$

$$= \frac{P(X > s+t)}{P(X > s)} \quad (\text{since } s+t > s)$$

$$= \frac{\int_{s+t}^{\infty} \lambda e^{-\lambda x} dx}{\int_{s}^{\infty} \lambda e^{-\lambda x} dx}$$

$$= \frac{e^{-\lambda(s+t)}}{e^{-\lambda s}}$$

$$= e^{-\lambda t}$$

[7]

$$= \int_t^{\infty} \lambda e^{-\lambda x} dx \; = P(X > t)$$

Επομένως, $P(X > 50000|X > 20000) =$, $P(X > 30000 + 20000 \mid X > 20000) = P(X > 30000)$

και $P(X > 30000) = 1 - F(30000)$.

και $P(X > 50000|X > 20000) = \frac{P(X>50000, X>20000)}{P(X>20000)} = \frac{P(X>50000)}{P(X>20000)} = (1 - F(50000))/(1 - F(20000))$.

Ακολουθούμε αντίστοιχη διαδικασία στο περιβάλλον του Octave:

```
48  # Task C
49  k = 0:0.00001:8;
50  lambda = 2.5;
51  exponential_cdf_ = expcdf(k,lambda);
52
53  display("The P[X > 30000] = ");
54  display(1-exponential_cdf_(30000));
55
56  answer = (1-exponential_cdf_(50000))/(1-exponential_cdf_(20000));
57  display("P[X > 50000 | X > 20000] = ");
58  display(answer);
```

και το αποτέλεσμα είναι το θεωρητικά αναμενόμενο αφού οι πιθανότητες των 2 ενδεχομένων είναι ίσες:

```
The P[X > 30000] =
0.8869
P[X > 50000 | X > 20000] =
answer = 0.8869
>>
```

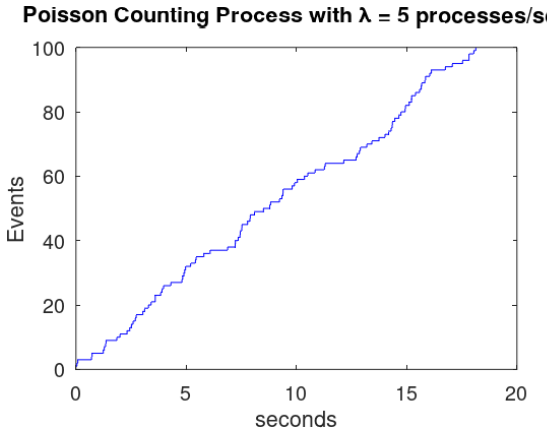
Γ

Διαδικασία Καταμέτρησης Poisson

Α) Γνωρίζουμε ότι οι χρόνοι που μεσολαβούν ανάμεσα στην εμφάνιση δύο διαδοχικών γεγονότων Poisson ακολουθούν εκθετική κατανομή με μέση τιμή 1/λ. Χρησιμοποιούμε λοιπόν την εντολή exprnd() για να παράξουμε 100 διαδοχικά τυχαία γεγονότα θεωρώντας ότι λ = 5 γεγονότα/sec. Άρα η εκθετική κατανομή που ακολουθούν διαδοχικά τυχαία γεγονότα θα έχει μέση τιμή 0.2.

Αθροίζουμε το πλήθος των γεγονότων στις διαδοχικές θέσεις του διανύσματος N και το διάνυσμα time αποθηκεύει μέσω συσσώρευσης την αντίστοιχη χρονική στιγμή για δεδομένο πλήθος γεγονότων. Παρακάτω βλέπουμε τον αντίστοιχο κώδικα και την οπτική αναπαράσταση της καταμέτρησης Poisson με χρήση της συνάρτησης stairs().

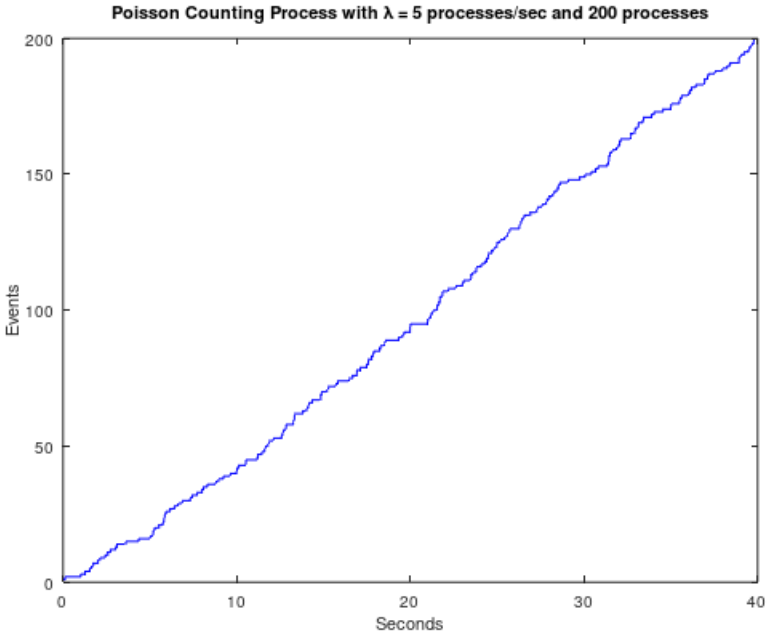
```
5  # Poisson Counting Process
6  # Task A
7  lambda = 5;
8  N = exprnd(1/lambda,1,100);
9  time = ones(100,1);
10 for i = 1:99
11   N(i+1) = N(i+1)+N(i);
12   time(i+1) = time(i+1)+time(i);
13 endfor
14 figure(1);
15 stairs(N,time,color='b');
16 title("Poisson Counting Process with λ = 5 processes/sec");
17 ylabel("Events");
18 xlabel("seconds");
19
```



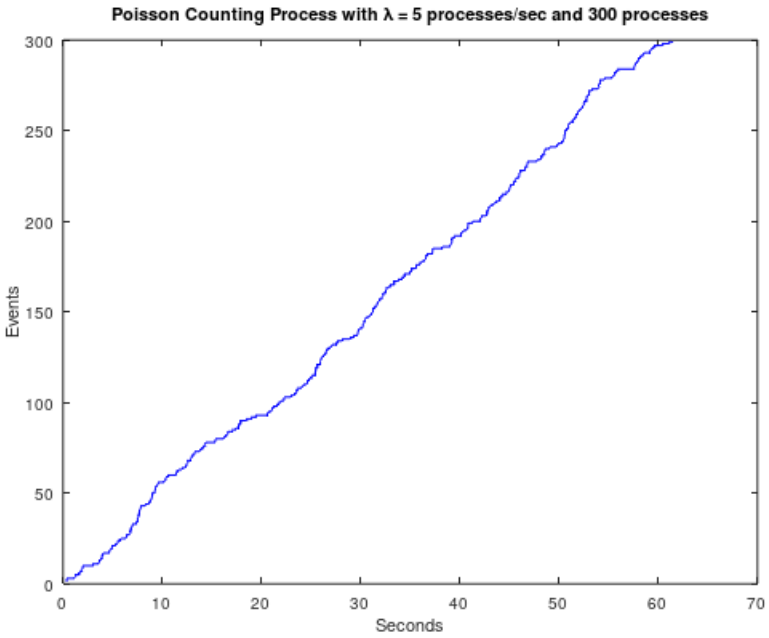
B)

Επαναλαμβάνουμε την παραπάνω διαδικασία για αυξανόμενο πλήθος γεγονότων και προκύπτουν οι ακόλουθες γραφικές για (i) 200, (ii) 300, (iii) 500, (iv) 1000, (v) 10000 διαδοχικά τυχαία γεγονότα.

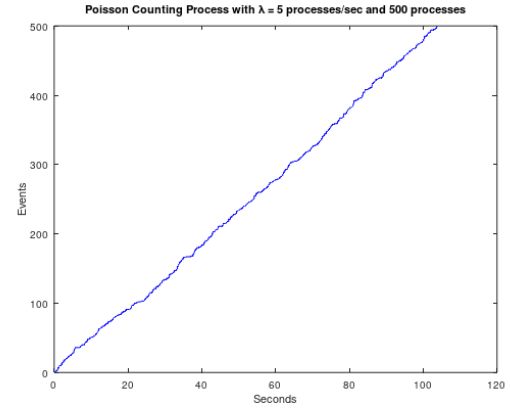
(i)


Poisson Counting Process with λ = 5 processes/sec and 200 processes
Events
Seconds
0
50
100
150
200
0
10
20
30
40

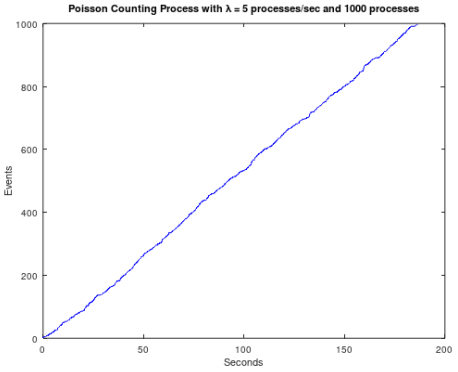
(ii)


Poisson Counting Process with λ = 5 processes/sec and 300 processes
Events
Seconds
0
50
100
150
200
250
300
0
10
20
30
40
50
60
70

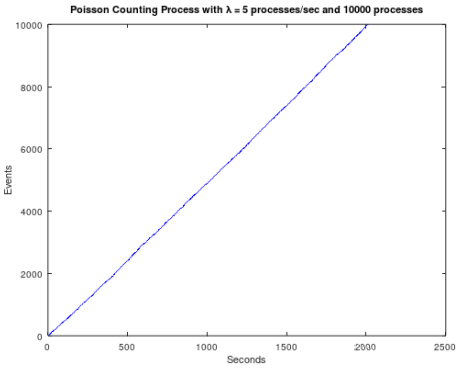
(iii)



(iv)



(v)



Μέσος αριθμός γεγονότων στη μονάδα χρόνου για κάθε περίπτωση φαίνεται στην έξοδο του command window:

```
Approximation for 200 events : 5.016041
Approximation for 300 events : 4.842246
Approximation for 500 events : 5.049038
Approximation for 1000 events : 4.977495
Approximation for 10000 events : 4.966595
>>
```

Παρατηρούμε από τις γραφικές ότι όσο αυξάνεται ο αριθμός των γεγονότων η γραφική της παραγωγής διαδικασιών Poisson τείνει να γίνει ευθεία με κλίση λ = 5 δηλ. συγκλίνει στην εμφάνιση 5 γεγονότων ανά

δευτερόλεπτο. Αυτό φαίνεται προφανώς και από τον μέσο ρυθμό γεγονότων που υπολογίσαμε που τείνει να έχει μικρότερο σφάλμα-απόκλιση από το 5 όσο αυξάνουμε το πλήθος των γεγονότων. Ακολουθεί ο αντίστοιχος κώδικας:

```
# Task B
# i)
lambda = 5;
event_num = 200; # the proposed number of events
N = exprnd(1/lambda,1,event_num);
time = ones(event_num,1);
for i = 1:(event_num-1)
  N(i+1) = N(i+1)+N(i);
  time(i+1) = time(i+1)+time(i);
endfor

figure(2);
stairs(N,time,color='b');
title(sprintf("Poisson Counting Process with λ = 5 processes/sec and %d processes",event_num));
ylabel("Events");
xlabel("Seconds");
display(sprintf("Approximation for %d events : %f",event_num,event_num/N(event_num)));
# ii)
lambda = 5;
event_num = 300; # the proposed number of events
N = exprnd(1/lambda,1,event_num);
time = ones(event_num,1);
for i = 1:(event_num-1)
  N(i+1) = N(i+1)+N(i);
  time(i+1) = time(i+1)+time(i);
endfor

figure(3);
stairs(N,time,color='b');
title(sprintf("Poisson Counting Process with λ = 5 processes/sec and %d processes",event_num));
ylabel("Events");
xlabel("Seconds");
display(sprintf("Approximation for %d events : %f",event_num,event_num/N(event_num)));
# iii)
lambda = 5;
event_num = 500; # the proposed number of events
N = exprnd(1/lambda,1,event_num);
time = ones(event_num,1);
for i = 1:(event_num-1)
  N(i+1) = N(i+1)+N(i);
  time(i+1) = time(i+1)+time(i);
endfor

figure(4);
stairs(N,time,color='b');
title(sprintf("Poisson Counting Process with λ = 5 processes/sec and %d processes",event_num));
ylabel("Events");
xlabel("Seconds");
display(sprintf("Approximation for %d events : %f",event_num,event_num/N(event_num)));
# iv)
lambda = 5;
event_num = 1000; # the proposed number of events
N = exprnd(1/lambda,1,event_num);
time = ones(event_num,1);
for i = 1:(event_num-1)
  N(i+1) = N(i+1)+N(i);
  time(i+1) = time(i+1)+time(i);
endfor

figure(5);
stairs(N,time,color='b');
title(sprintf("Poisson Counting Process with λ = 5 processes/sec and %d processes",event_num));
ylabel("Events");
xlabel("Seconds");
display(sprintf("Approximation for %d events : %f",event_num,event_num/N(event_num)));
# v)
lambda = 5;
event_num = 10000; # the proposed number of events
N = exprnd(1/lambda,1,event_num);
time = ones(event_num,1);
for i = 1:(event_num-1)
  N(i+1) = N(i+1)+N(i);
  time(i+1) = time(i+1)+time(i);
endfor

figure(6);
stairs(N,time,color='b');
title(sprintf("Poisson Counting Process with λ = 5 processes/sec and %d processes",event_num));
ylabel("Events");
xlabel("Seconds");
display(sprintf("Approximation for %d events : %f",event_num,event_num/N(event_num)));
```