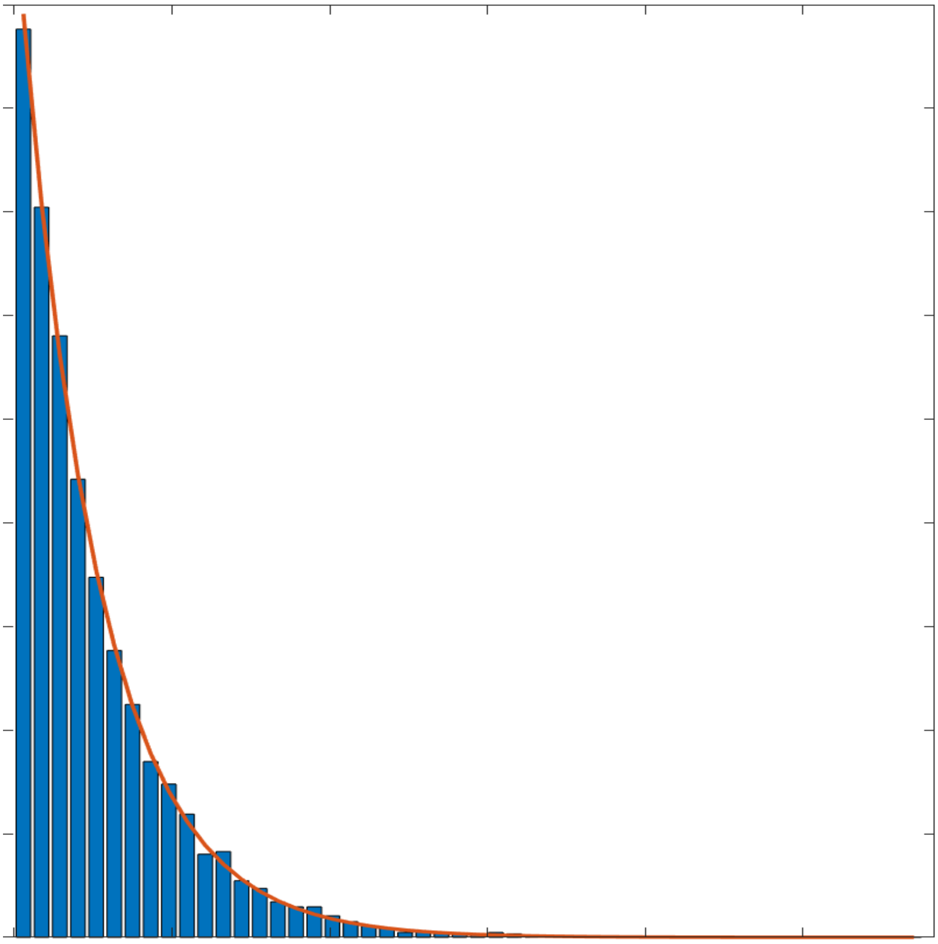
Escola Tècnica Superior
d'**Enginyeria de Telecomunicació de Barcelona**

Laboratory Session 1.
Study of the probability density function of random variables



Erik Solé & Gokarna Dumre
Lluís Llopis

GRUP 51
2022-23

Departament d'Enginyeria Telemàtica
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Index

1. Objectives of the session.

The objectives of this session may be summarized in the following points:

- To experiment with the creation of uniform and exponential random variables.
- To write functions in MATLAB.
- To obtain the occupancy probability of a system from the data generated by a simulator.

2. Previous work.

Obtain the inverse function of the cdf of an exponential random variable.

Gokama Dumrr Enik Sole AAL 15/05/2022

$$F(x) = 1 - e^{-\lambda x}$$

$$y = 1 - e^{-\lambda x}$$

$$1 - y = e^{-\lambda x}$$

$$\ln(1-y) = -\lambda x$$

$$\boxed{x = \frac{-\ln(1-y)}{\lambda}}$$

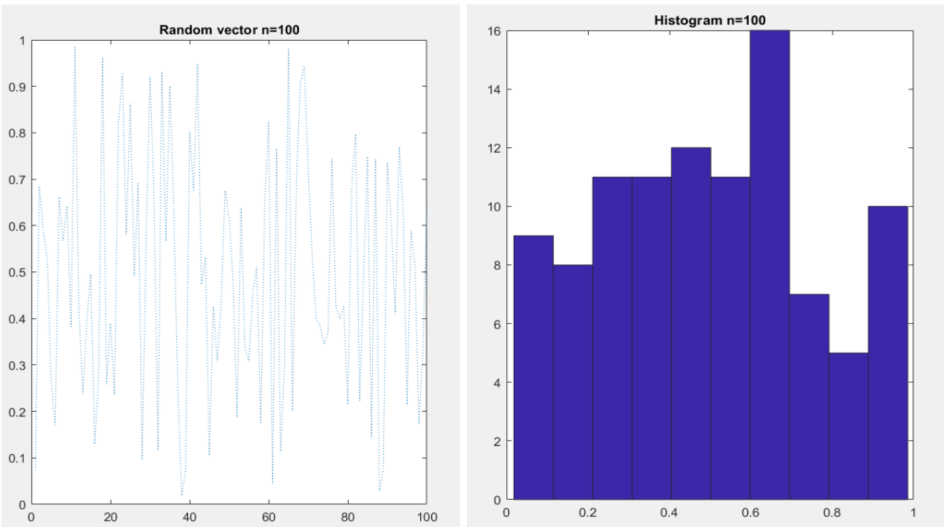
3. Uniform pdf.

Do the following steps in MATLAB:

- Generate a vector with size 100 (100x1 matrix) with uniformly distributed values in the interval [0,1] (Use the rand function, for instance `r=rand(100,1)`).
- Graph the vector (`plot(r)`).
- Obtain the mean value and the variance and compare with the theoretical values
- Generate and plot a histogram with the vector values (`hist(r)`).

```
%3.Uniform pdf
clear
clc
n=100
r=rand(n,1);
subplot(1,2,1)
plot(r,':');
title('Random vector n=100')
subplot(1,2,2)
hist(r);
title('Histogram n=100')
rmean=mean(r)
rvar=var(r)
```

Name	Value
n	100
r	100x1 double
rmean	0.4897
rvar	0.0955



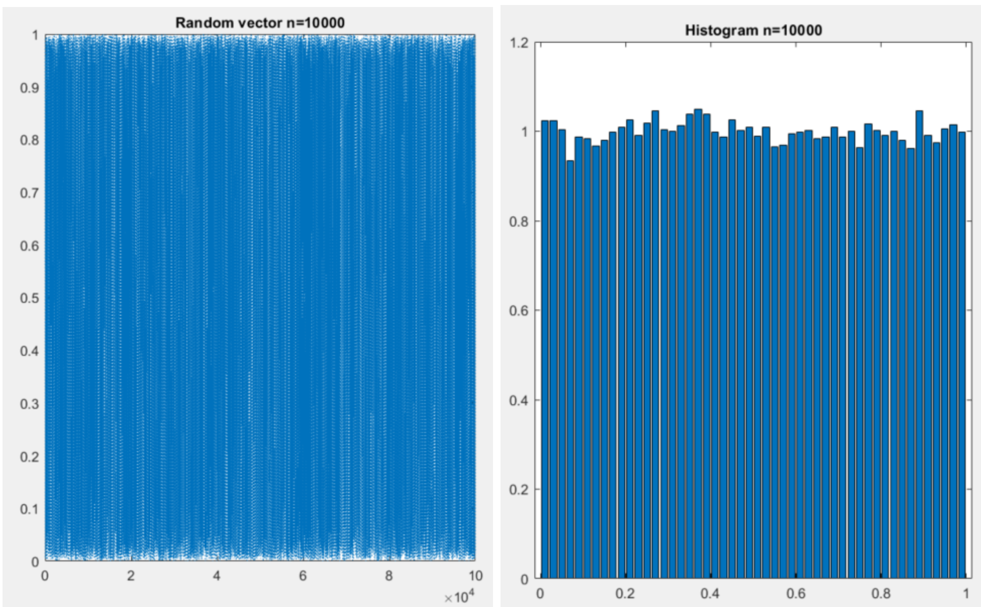
We can see the statistics of the uniform distribution. Specifically,the mean and the variance which is quite different from the theoreticals values. Its because the number of samples, in this case we got 10 samples so its logic that the theoretical values and experimental values are different.

▪ Repeat the previous steps but generating the vector with a size equal to 10000.
▪ Normalize the histogram area to obtain the pdf, doing the next steps (if needed, use the `help` function of MATLAB to understand the meaning and the objective of every sentence):

```
- [ht,x]=hist(r);
- bar(x,ht)
- area=(x(2)-x(1))*sum(ht);
- pdf=ht./area;
- figure
- bar(x,pdf);
```

```
%Normalized pdf with 100000 values
clear
clc
n=10000
nbin=50
r=rand(n,1);
subplot(1,2,1)
plot(r,':');
title('Random vector n=10000')
[ht,x]=hist(r,nbin);
area=(x(2)-x(1))*sum(ht);
pdf=ht./area;
subplot(1,2,2)
bar(x,pdf)
title('Histogram n=10000')
rmean=mean(r)
rvar=var(r)
```

Name	Value
area	1.9999e+03
ht	1x50 double
n	100000
nbin	50
pdf	1x50 double
r	100000x1 double
rmean	0.5014
rvar	0.0835
x	1x50 double



There is an improvement compared to the previous exercise, now the number of samples is quite large for that reason the theoretical and the experimental values are almost identical.

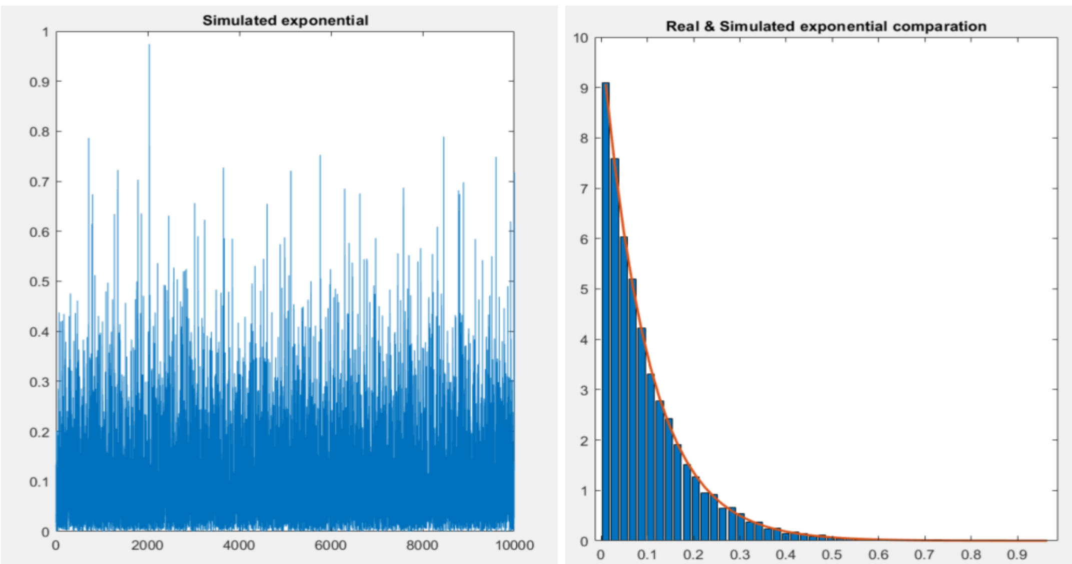
4. Exponential pdf.

Do the following steps en MATLAB:

- Starting with a uniform distribution (10000 values), obtain a new vector with exponentially distributed values (see the annex). Choose yourself the mean of the exponential.
- Check the mean value, the variance and the variation coefficient of the new vector.
- Obtain and plot the pdf (50 bins).
- Compare graphically the obtained pdf with a "real" exponential of the same mean value.

```
%4.Exponential pdf
clear
clc
n=10000
nbin=50
lamb=10
%simulated exponential
r=rand(n,1);
expo=-log(1-r)/lamb;
meanexp=mean(expo)
varexp=var(expo)
subplot(1,2,1)
plot(expo)
title('Simulated exponential')
%simulated exponential histogram
[ht,x]=hist(expo,nbin)
area=(x(2)-x(1))*sum(ht);
pdf=ht./area;
%real exponential pdf
expo2=lamb*exp(-x*lamb)
subplot(1,2,2)
bar(x,pdf)
hold on
plot(x,expo2,'LineWidth',1.75)
title('real & simulated exponential comparation')
hold off
```

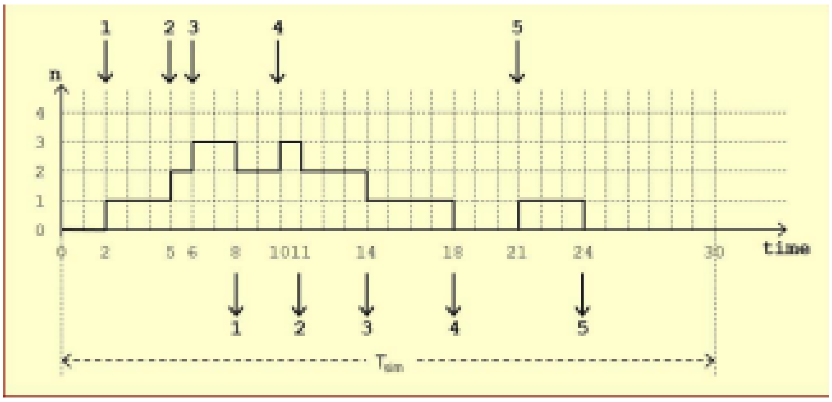
Name	Value
area	194.8679
expo	10000x1 double
expo2	1x50 double
ht	1x50 double
lamb	10
meanexp	0.0997
n	10000
nbin	50
pdf	1x50 double
r	10000x1 double
varexp	0.0100
x	1x50 double




```
In our case, our simulated exponential pdf and the real exponential
look similar.
```

5. Probability function of the occupancy states in a system.

Write a MATLAB function to obtain the probability function of the occupancy states in a system. The parameter for this function will be obtained from a matrix (two columns, any number of rows). The first column is the time at which the system moves to a new state. The second column is the new state. This is similar to the format that the ScalevLite simulator (which will be used in next sessions) provides (in a text file). Check the correct behavior of your function with the example shown in the figure.



time	n
0	0
2	1
5	2
6	3
8	2
10	3
11	2
14	1
18	0
21	1
24	0
30	0

```
%5.Probability function of the occupancy states in a system
clear
clc
time=[0 2 5 6 8 10 11 14 18 21 24 30]; %tiempo
n=[0 1 2 3 2 3 2 1 0 1 0 0]; %estados
matrix=[time;n].' %hacemos 2 columnas, de tiempo y de estados
Max_time=max(time); %la cantidad total de observacio
Max_stats=max(n) % el número de estats
prob=zeros(1,max(n)+1) %vector de possibilitats

%iteramos el vector tiempo uno por uno y cuando se produce el cambio de estado en el que estamos calculamos cuánto tiempo ha estado en ese estado haciendo la resta, y este valor lo sumamos a la probabilidad del estado.
for i=1:length(time)-1
   prob(n(i)+1)=prob(n(i)+1)+(time(i+1)-time(i))
end
prob=prob./Max_time;
```

$$p_0 = \frac{2+3+6}{30} = \frac{11}{30} \qquad p_1 = \frac{3+4+3}{30} = \frac{10}{30} \qquad p_2 = \frac{1+2+3}{30} = \frac{6}{30} \qquad p_3 = \frac{2+1}{30} = \frac{3}{30}$$

To obtain this probability function, we define a matrix of two columns (time and state). After that, we iterate the time vector one by one and if there is a change of state , we calculate how long it has been in that state by doing the subtraction, and we add this value to the probability of the state.

Name	Value
i	11
matrix	12x2 double
Max_stats	3
Max_time	30
n	1x12 double
prob	[0.3667,0.3333,0.2000,0.1000]
time	1x12 double

Annex: Matlab script

```matlab
%% Erik Sole && Gokarna Dumre      AAX grup 51
%3.Uniform pdf
clear
clc
n=100
r=rand(n,1);
subplot(1,2,1)
plot(r,':');
title('Random vector n=100')
subplot(1,2,2)
hist(r);
title('Histogram n=100')
rmean=mean(r)
rvar=var(r)
%%
%Normalized pdf with 100000 values
clear
clc
n=10000
nbin=50
r=rand(n,1);
subplot(1,2,1)
plot(r,':');
title('Random vector n=10000')
[ht,x]=hist(r,nbin);
area=(x(2)-x(1))*sum(ht);
pdf=ht./area;
subplot(1,2,2)
bar(x,pdf)
title('Histogram n=10000')
rmean=mean(r)
rvar=var(r)
%%
%4.Exponential pdf
clear
clc
n=10000
nbin=50
lamb=10
%simulated exponential
r=rand(n,1);
expo=-log(1-r)/lamb;
meanexp=mean(expo)
varexp=var(expo)
subplot(1,2,1)
plot(expo)
title('Simulated exponential')
%simulated exponential histogram
[ht,x]=hist(expo,nbin)
area=(x(2)-x(1))*sum(ht);
pdf=ht./area;
%real exponential
expo2=lamb*exp(-x*lamb)
subplot(1,2,2)
bar(x,pdf)
hold on
plot(x,expo2,'LineWidth',1.75)
title('real & simulated exponential comparation')
hold off
```

```
%%
%5.Probability function of the occupancy states in a system
clear
clc
time=[0 2 5 6 8 10 11 14 18 21 24 30]; %tiempo
n=[0 1 2 3 2 3 2 1 0 1 0 0]; %estados
matrix=[time;n].'
Max_time=max(time); %la cantidad total de observacio
Max_stats=max(n) % el numero de estats
prob=zeros(1,max(n)+1) %vector de possibilitats
%Hacer un for de 0 hasta size(time)
%iteramos el vector tiempo uno por uno y cuando se produce el cambio
de
%estado en el q estamos calculamos cuanto tiempo ha estado en ese
estado
%haciendo la resta, i este valor lo sumamos a la probalidad del
estado.
for i=1:length(time)-1
   prob(n(i)+1)=prob(n(i)+1)+(time(i+1)-time(i))
end
prob=prob./Max_time;
```