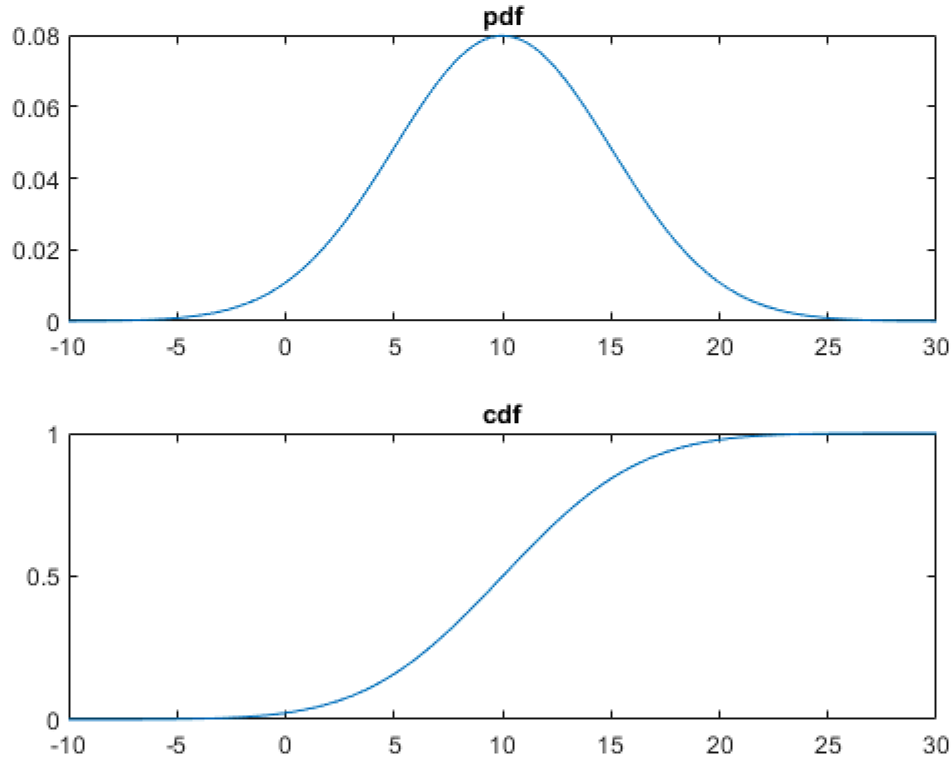
Marwin B. Alejo 2020-20221 EE214_Module2-LabEx3

Table of Contents

- Date Performed (d/m/y): 23/10/2021 to 28/10/2021

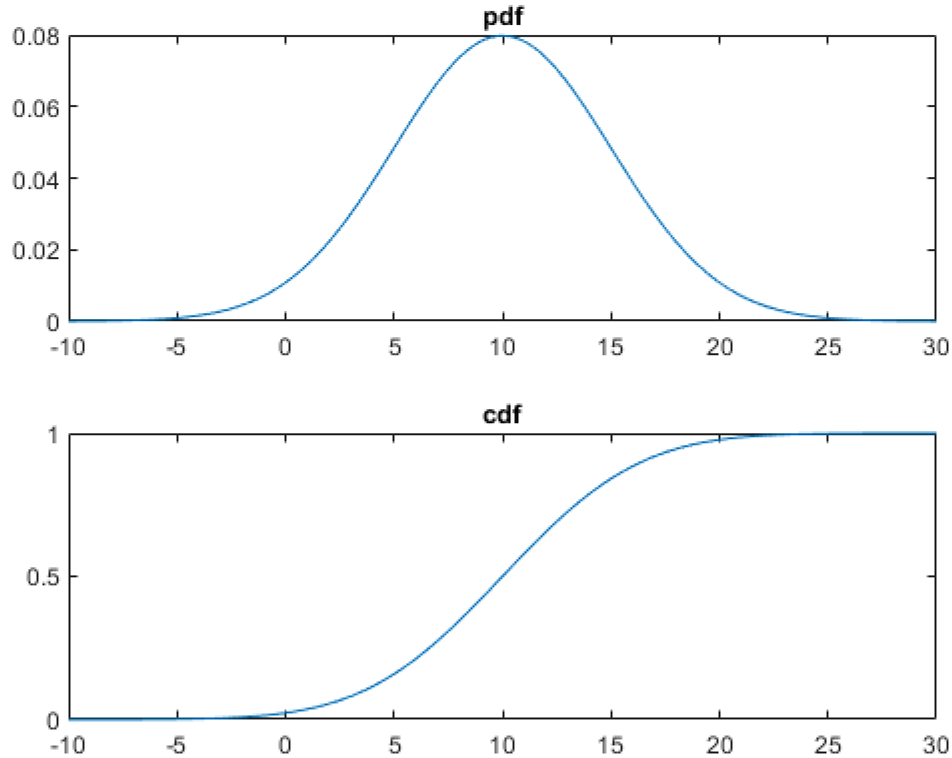
I. Gaussian Random Variable

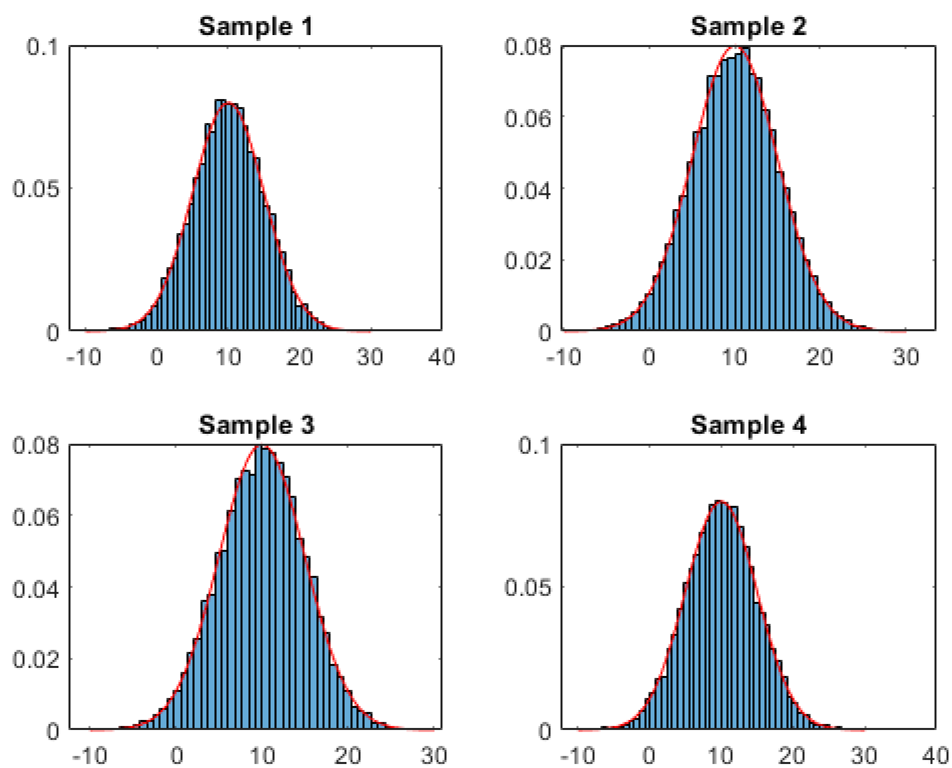
```
gaussdist = makedist('normal','mu',10,'sigma',5);
x = -10:0.1:30;
pdf1 = pdf(gaussdist,x);
cdf1 = cdf(gaussdist,x);

N = 10000;
samples1 = random(gaussdist,N,4);

figure(); title('Figure 1')
subplot(2,1,1); plot(x,pdf1); title('pdf');
subplot(2,1,2); plot(x,cdf1); title('cdf');
```



Considering the definition of guassian distribution, PDF, CDF, and the values provided for the parameters of the gaussian distribution, the author expected the generated PDF and CDF figure above. As for the **PDF** , the author expected that the center of the bell curve would be at 10 since the provided mean value of the gaussian distribution is 10. As with the **CDF** , the author expected that the slope of CDF would be smoother provided that the sample size is 10000 and the variance is 5 and the center of the slope is at 10 due to the assigned mean value (10). If the variance of the provided gaussian distribution is higher that 5, the slope of the CDF would be steeper. In general, the 'mean' define the position of the PDF and CDF along the horizntal axis while the variance define the steepness of their curves. Moreover, PDF is just a derivative of CDF hence, for this case sample, the curves of either PDF and CDF reflect each other and are expected to be as is, as shown above.

```
figure();
subplot(2,2,1);histogram(samples1(:,1),50,'Normalization','pdf');hold on;
plot(x,pdf1,'r'); title('Sample 1');
subplot(2,2,2);histogram(samples1(:,2),50,'Normalization','pdf');hold on;
plot(x,pdf1,'r'); title('Sample 2');
subplot(2,2,3);histogram(samples1(:,3),50,'Normalization','pdf');hold on;
plot(x,pdf1,'r'); title('Sample 3');
subplot(2,2,4);histogram(samples1(:,4),50,'Normalization','pdf');hold on;
plot(x,pdf1,'r'); title('Sample 4');
```


pdf
0.08
0.06
0.04
0.02
0
-10
-5
0
5
10
15
20
25
30
cdf
1
0.5
0


Sample 1
Sample 2
Sample 3
Sample 4
0.1
0.05
0
0.08
0.06
0.04
0.02
-10
0
10
20
30
40

Mean and Standard Deviation of Sample 1

```
fitdist(samples1(:,1),'normal')
```

```
ans =

  NormalDistribution

  Normal distribution
       mu = 10.0344   [9.93681, 10.132]
    sigma = 4.97961   [4.91155, 5.0496]
```

Mean and Standard Deviation of Sample 2

```
fitdist(samples1(:,2),'normal')
```

```
ans =

  NormalDistribution

  Normal distribution
       mu = 10.0351   [9.93625, 10.1339]
    sigma = 5.04132   [4.97241, 5.11217]
```

Mean and Standard Deviation of Sample 3

```
fitdist(samples1(:,3),'normal')
```

```
ans =

  NormalDistribution

  Normal distribution
       mu = 10.0196   [9.92118, 10.1181]
    sigma = 5.02259   [4.95393, 5.09318]
```

Mean and Standard Deviation of Sample 4

```
fitdist(samples1(:,4),'normal')
```

```
ans =

  NormalDistribution

  Normal distribution
       mu = 10.0354   [9.93789, 10.1329]
    sigma = 4.97496   [4.90696, 5.04489]
```

Considering the outputs above, the mean and standard deviation the four samples are approximately equivalent to 10 and 5 which is the original mean and stadard deviation value of the first gaussian distribution (gaussdist) of this section. Moreover, the four samples might have unbalanced or asymetric distribution on both sides of their curve without this original value of mean and standard deviation and their sample-unit would vary from each other.

Overall, gaussian or normal distribution is a bell-shaped curve ditribution and is uniformly balanced or symmetric at its mean. The variance or standard deviation in the bell-shaped curve determine the steepness of the distribution and how far the sample-unit lies away from the mean. Furthermore, PDF is the probability that a random variable will take an exact value while CDF is of the logic 'less than or equal'.

II. Binary Symmetric Channel

Considering the given given AWGN paramters and values of mean of 0 and standard deviation of 1, the probability of error (Pe) of the binary symmetric channel with both '1' and '0' have equal apriori probabilities may be modeled using the ff. expressions below:

Equation 1: $P(e) = P(1_{decided}), 0_{transmitted}) + P(0_{decided}, 1_{transmitted})$

and by applying the Bayes Theorem, P(e) in Equation 1 may be expressed as shown in Equation 2:

Equation 2: $P(e) = 0.5 \int_0^\infty f(r|0_T)dr + 0.5 \int_{-\infty}^0 f(r|1_T)dr$

and Equation 2 may be expressed through Q-function which represents the region of error probability of the gaussian model. This statement may be translated through Equation 3:

Equation 3: $P(1_{decided}|0_{transmitted}) = Q(\sqrt{\frac{E_s}{N_0/2}})$ or $P(0_{decided}|1_{transmitted}) = Q(\sqrt{\frac{E_s}{N_0/2}})$

furthermore, the probability of error P(e) may be expressed as shown in Equation 4.

Equation 4: $P(e) = 0.5Q(\sqrt{\frac{E_s}{N_0/2}}) + 0.5Q(\sqrt{\frac{E_s}{N_0/2}}) = Q(\sqrt{\frac{2E_s}{N_0}}) = 0.5erfc(\sqrt{\frac{E_s}{N_0}})$

hence, the P(e) of the given conditions above may be generated using the code below:

```
P_e=0.5*erfc(sqrt(((0.5^2)/2)/(2))); % probability of error
P_s = 1-P_e; % probability of success
```

To compute PR1T1, PR0T1, PR1T0, PR0T0, PT1R1, PT0R1, PT1R0, PT0R0, PR0 and PR1, the following commands and code snippets below must be considered:

Manual Computation when N=1000

```
N_1000 = 1000; % consider 1000 as sample number of bits
tx_1000 = rand(1,N_1000) > 0.5; % generate 1000 random bits
tx0_1000 = (sum(tx_1000(:)==0))/N_1000; % rate of transmitted 0's from
 1000 sample bits
tx1_1000 = (sum(tx_1000(:)==1))/N_1000; % rate of transmitted 1's from
 1000 sample bits
txe_1000 = P_e; % transmission error probability rate
txs_1000 = P_s; % transmission success probability rate
rx1_1000 = tx1_1000*txs_1000+tx0_1000*txe_1000; % computes the P(R'1')
```

```
rx0_1000 = tx0_1000*txs_1000+tx1_1000*txe_1000; % computes the P(R'0')
PR1T1_1000 = (txs_1000*rx1_1000)/tx1_1000; % computes P(R'1'|T'1')
PR0T1_1000 = (txe_1000*rx0_1000)/tx1_1000; % computes P(R'0'|T'1')
PR1T0_1000 = (txe_1000*rx1_1000)/tx0_1000; % computes P(R'1'|T'0')
PR0T0_1000 = (txs_1000*rx0_1000)/tx0_1000; % computes P(R'0'|T'0')
PT1R1_1000 = (txs_1000*tx1_1000)/
((txs_1000*tx1_1000)+(txe_1000*tx0_1000)); % computes P(T'1'|R'1')
PT0R1_1000 = (txe_1000*tx0_1000)/
((txe_1000*tx0_1000)+(txs_1000*tx1_1000)); % computes P(T'0'|R'1')
PT1R0_1000 = (txe_1000*tx1_1000)/
((txe_1000*tx1_1000)+(txs_1000*tx0_1000)); % computes P(T'1'|R'0')
PT0R0_1000 = (txs_1000*tx0_1000)/
((txs_1000*tx0_1000)+(txe_1000*tx1_1000)); % computes P(T'0'|R'0')

fprintf('MANUAL N=1000:\n PR1=%.4f,\n PR0=%.4f, \n PR1T1=%.4f, \n
 PR0T1=%.4f, \n PR1T0=%.4f, \n PR0T0=%.4f, \n PT1R1=%.4f, \n PT0R1=
%.4f, \n PT1R0=%.4f, \n PT0R0=%.4f',...

 rx1_1000,rx0_1000,PR1T1_1000,PR0T1_1000,PR1T0_1000,PR0T0_1000,PT1R1_1000,PT0R1_10
```

```
MANUAL N=1000:
 PR1=0.5050,
 PR0=0.4950,
 PR1T1=0.6221,
 PR0T1=0.3458,
 PR1T0=0.3791,
 PR0T0=0.6554,
 PT1R1=0.6546,
 PT0R1=0.3454,
 PT1R0=0.3786,
 PT0R0=0.6214
```

MATLAB Computation when N=1000

```
ch_1000 = rand(1,N_1000) > txs_1000;
rx_1000 = xor(tx_1000, ch_1000);
matlab_rx0_1000 = (sum(rx_1000(:)==0))/N_1000; % rate of received 0's
 from 1000 sample bits
matlab_rx1_1000 = (sum(rx_1000(:)==1))/N_1000; % rate of received 1's
 from 1000 sample bits
matlab_PR1T1_1000 = (txs_1000*matlab_rx1_1000)/tx1_1000; % computes
 P(R'1'|T'1')
matlab_PR0T1_1000 = (txe_1000*matlab_rx0_1000)/tx1_1000; % computes
 P(R'0'|T'1')
matlab_PR1T0_1000 = (txe_1000*matlab_rx1_1000)/tx0_1000; % computes
 P(R'1'|T'0')
matlab_PR0T0_1000 = (txs_1000*matlab_rx0_1000)/tx0_1000; % computes
 P(R'0'|T'0')
matlab_PT1R1_1000 = (txs_1000*tx1_1000)/
((txs_1000*tx1_1000)+(txe_1000*tx0_1000)); % computes P(T'1'|R'1')
matlab_PT0R1_1000 = (txe_1000*tx0_1000)/
((txe_1000*tx0_1000)+(txs_1000*tx1_1000)); % computes P(T'0'|R'1')
matlab_PT1R0_1000 = (txe_1000*tx1_1000)/
((txe_1000*tx1_1000)+(txs_1000*tx0_1000)); % computes P(T'1'|R'0')
```

```
matlab_PT0R0_1000 = (txs_1000*tx0_1000)/
((txs_1000*tx0_1000)+(txe_1000*tx1_1000)); % computes P(T'0'|R'0')

fprintf('MATLAB N=1000:\n PR1=%.4f,\n PR0=%.4f, \n PR1T1=%.4f, \n
 PR0T1=%.4f, \n PR1T0=%.4f, \n PR0T0=%.4f, \n PT1R1=%.4f, \n PT0R1=
%.4f, \n PT1R0=%.4f, \n PT0R0=%.4f',...

 matlab_rx1_1000,matlab_rx0_1000,matlab_PR1T1_1000,matlab_PR0T1_1000,matlab_PR1T0_
```

```
MATLAB N=1000:
 PR1=0.4840,
 PR0=0.5160,
 PR1T1=0.5963,
 PR0T1=0.3604,
 PR1T0=0.3633,
 PR0T0=0.6832,
 PT1R1=0.6546,
 PT0R1=0.3454,
 PT1R0=0.3786,
 PT0R0=0.6214
```

Manual Computation when N=10000

```
N_10000 = 10000; % consider 10000 as sample number of bits
tx_10000 = rand(1,N_10000) > 0.5; % generate 10000 random bits
tx0_10000 = (sum(tx_10000(:)==0))/N_10000; % rate of transmitted 0's
 from 10000 sample bits
tx1_10000 = (sum(tx_10000(:)==1))/N_10000; % rate of transmitted 1's
 from 10000 sample bits
txe_10000 = P_e; % transmission error probability rate
txs_10000 = P_s; % transmission success probability rate
rx1_10000 = tx1_10000*txs_10000+tx0_10000*txe_10000; % computes the
 P(R'1')
rx0_10000 = tx0_10000*txs_1000+tx1_10000*txe_10000; % computes the
 P(R'0')
PR1T1_10000 = (txs_10000*rx1_10000)/tx1_10000; % computes P(R'1'|T'1')
PR0T1_10000 = (txe_10000*rx0_10000)/tx1_10000; % computes P(R'0'|T'1')
PR1T0_10000 = (txe_10000*rx1_10000)/tx0_10000; % computes P(R'1'|T'0')
PR0T0_10000 = (txs_10000*rx0_10000)/tx0_10000; % computes P(R'0'|T'0')
PT1R1_10000 = (txs_10000*tx1_10000)/
((txs_10000*tx1_10000)+(txe_10000*tx0_10000)); % computes P(T'1'|R'1')
PT0R1_10000 = (txe_10000*tx0_10000)/
((txe_10000*tx0_10000)+(txs_10000*tx1_10000)); % computes P(T'0'|R'1')
PT1R0_10000 = (txe_10000*tx1_10000)/
((txe_10000*tx1_10000)+(txs_10000*tx0_10000)); % computes P(T'1'|R'0')
PT0R0_10000 = (txs_10000*tx0_10000)/
((txs_10000*tx0_10000)+(txe_10000*tx1_10000)); % computes P(T'0'|R'0')

fprintf('MANUAL N=10000:\n PR1=%.4f,\n PR0=%.4f, \n PR1T1=%.4f, \n
 PR0T1=%.4f, \n PR1T0=%.4f, \n PR0T0=%.4f, \n PT1R1=%.4f, \n PT0R1=
%.4f, \n PT1R0=%.4f, \n PT0R0=%.4f',...

 rx1_10000,rx0_10000,PR1T1_10000,PR0T1_10000,PR1T0_10000,PR0T0_10000,PT1R1_10000,P
```

```
MANUAL N=10000:
```

```
 PR1=0.4975,
 PR0=0.5025,
 PR1T1=0.6467,
 PR0T1=0.3704,
 PR1T0=0.3536,
 PR0T0=0.6299,
 PT1R1=0.6297,
 PT0R1=0.3703,
 PT1R0=0.3535,
 PT0R0=0.6465
```

MATLAB Computation when N=10000

```
ch_10000 = rand(1,N_10000) > txs_10000;
rx_10000 = xor(tx_10000, ch_10000);
matlab_rx0_10000 = (sum(rx_10000(:)==0))/N_10000; % rate of received
 0's from 10000 sample bits
matlab_rx1_10000 = (sum(rx_10000(:)==1))/N_10000; % rate of received
 1's from 10000 sample bits
matlab_PR1T1_10000 = (txs_10000*matlab_rx1_10000)/tx1_10000; %
 computes P(R'1'|T'1')
matlab_PR0T1_10000 = (txe_10000*matlab_rx0_10000)/tx1_10000; %
 computes P(R'0'|T'1')
matlab_PR1T0_10000 = (txe_10000*matlab_rx1_10000)/tx0_10000; %
 computes P(R'1'|T'0')
matlab_PR0T0_10000 = (txs_10000*matlab_rx0_10000)/tx0_10000; %
 computes P(R'0'|T'0')
matlab_PT1R1_10000 = (txs_10000*tx1_10000)/
((txs_10000*tx1_10000)+(txe_10000*tx0_10000)); % computes P(T'1'|R'1')
matlab_PT0R1_10000 = (txe_10000*tx0_10000)/
((txe_10000*tx0_10000)+(txs_10000*tx1_10000)); % computes P(T'0'|R'1')
matlab_PT1R0_10000 = (txe_10000*tx1_10000)/
((txe_10000*tx1_10000)+(txs_10000*tx0_10000)); % computes P(T'1'|R'0')
matlab_PT0R0_10000 = (txs_10000*tx0_10000)/
((txs_10000*tx0_10000)+(txe_10000*tx1_10000)); % computes P(T'0'|R'0')

fprintf('MATLAB N=10000:\n PR1=%.4f,\n PR0=%.4f, \n PR1T1=%.4f, \n
 PR0T1=%.4f, \n PR1T0=%.4f, \n PR0T0=%.4f, \n PT1R1=%.4f, \n PT0R1=
%.4f, \n PT1R0=%.4f, \n PT0R0=%.4f',...

 matlab_rx1_10000,matlab_rx0_10000,matlab_PR1T1_10000,matlab_PR0T1_10000,matlab_PR
```

```
MATLAB N=10000:
 PR1=0.4965,
 PR0=0.5035,
 PR1T1=0.6454,
 PR0T1=0.3711,
 PR1T0=0.3529,
 PR0T0=0.6311,
 PT1R1=0.6297,
 PT0R1=0.3703,
 PT1R0=0.3535,
 PT0R0=0.6465
```

Comparison of Manual and MATLAB generated probabilities when N=1000.

```
fprintf('Manual | MATLAB - when N=1000 \n PR1: %.4f | %.4f, \n PR0:
 %.4f | %.4f, \n PR1T1: %.4f | %.4f, \n PR0T1: %.4f | %.4f, \n PR1T0:
 %.4f | %.4f,\n PR0T0: %.4f | %.4f, \n PT1R1: %.4f | %.4f, \n PT0R1:
 %.4f | %.4f, \n PT1R0: %.4f | %.4f, \n PT0R0: %.4f | %.4f',...

 rx1_1000,matlab_rx1_1000,rx0_1000,matlab_rx0_1000,PR1T1_1000,matlab_PR1T1_1000,PR

 PT1R1_1000,matlab_PT1R1_1000,PT0R1_1000,matlab_PT0R1_1000,PT1R0_1000,matlab_PT1R0
```

```
Manual | MATLAB - when N=1000
 PR1: 0.5050 | 0.4840,
 PR0: 0.4950 | 0.5160,
 PR1T1: 0.6221 | 0.5963,
 PR0T1: 0.3458 | 0.3604,
 PR1T0: 0.3791 | 0.3633,
 PR0T0: 0.6554 | 0.6832,
 PT1R1: 0.6546 | 0.6546,
 PT0R1: 0.3454 | 0.3454,
 PT1R0: 0.3786 | 0.3786,
 PT0R0: 0.6214 | 0.6214
```

Comparison of Manual and MATLAB generated probabilities when N=10000.

```
fprintf('Manual | MATLAB - when N=10000 \n PR1: %.4f | %.4f, \n PR0:
 %.4f | %.4f, \n PR1T1: %.4f | %.4f, \n PR0T1: %.4f | %.4f, \n PR1T0:
 %.4f | %.4f,\n PR0T0: %.4f | %.4f, \n PT1R1: %.4f | %.4f, \n PT0R1:
 %.4f | %.4f, \n PT1R0: %.4f | %.4f, \n PT0R0: %.4f | %.4f',...

 rx1_10000,matlab_rx1_10000,rx0_10000,matlab_rx0_10000,PR1T1_10000,matlab_PR1T1_10

 PT1R1_10000,matlab_PT1R1_10000,PT0R1_10000,matlab_PT0R1_10000,PT1R0_10000,matlab_
```

```
Manual | MATLAB - when N=10000
 PR1: 0.4975 | 0.4965,
 PR0: 0.5025 | 0.5035,
 PR1T1: 0.6467 | 0.6454,
 PR0T1: 0.3704 | 0.3711,
 PR1T0: 0.3536 | 0.3529,
 PR0T0: 0.6299 | 0.6311,
 PT1R1: 0.6297 | 0.6297,
 PT0R1: 0.3703 | 0.3703,
 PT1R0: 0.3535 | 0.3535,
 PT0R0: 0.6465 | 0.6465
```

Conclusion: Given the parameters of AWGN, the probability of error of BSC is determined through an unorthodox approach. It is observed that the probabilities generated through manual and MATLAB approach yield similar to same values in both N samples. Unlike the probability values determined in the previous activity, by the influence of Gaussian distribution through AWGN and the computed ERFC, the normal distribution of samples and their probabilities are balance regardless of the number of samples and the included AWGN.

III. Queuing Theory

20 different mu (or lambda as we speak in poisson) values.

```
lambda = [0:1:10,20:10:100];
```

Humans in queue, must be whole numbers but fractioned-human may hypothetically be considered.

```
humans = 0:1:30;
```

This will create poisson dist objects using the given mu values (lambda).

```
for ctr = 1:length(lambda)
    poisdist(ctr) = makedist('Poisson','lambda',lambda(ctr));
end
```

This will compute the CDF of generated poisson dist with the length of humans in queue.

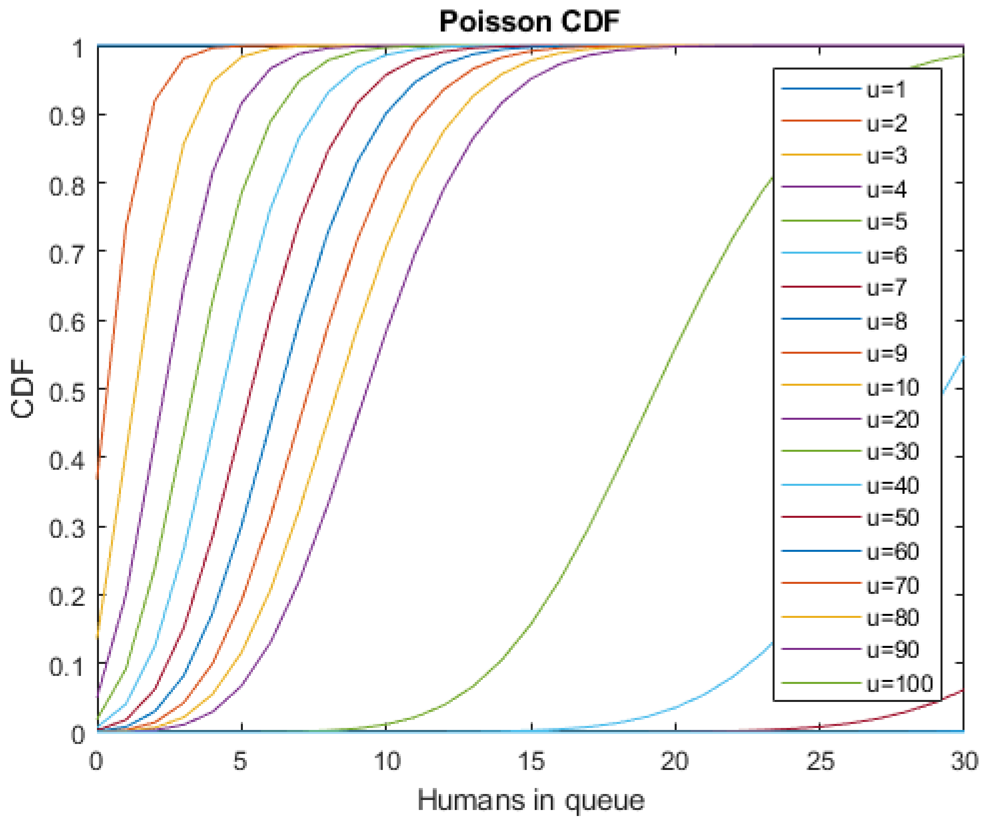
```
cdf2=zeros(length(lambda),length(humans)); % initialize pmf mtrx
 cells.
ctr=1;
while ctr<=length(lambda)
    cdf2(ctr,:)= cdf(poisdist(ctr),humans);
    ctr=ctr+1;
end
```

This will compute the PMF of the poisson dist given the mu(lambda) values.

```
pmf=zeros(length(lambda),length(humans)); % initialize pmf mtrx cells.
ctr=1;
while ctr<=length(lambda)
    pmf(ctr,:)= ((exp(-1*lambda(ctr))).*(lambda(ctr).^humans))./
factorial(round(humans));
    ctr=ctr+1;
end
```

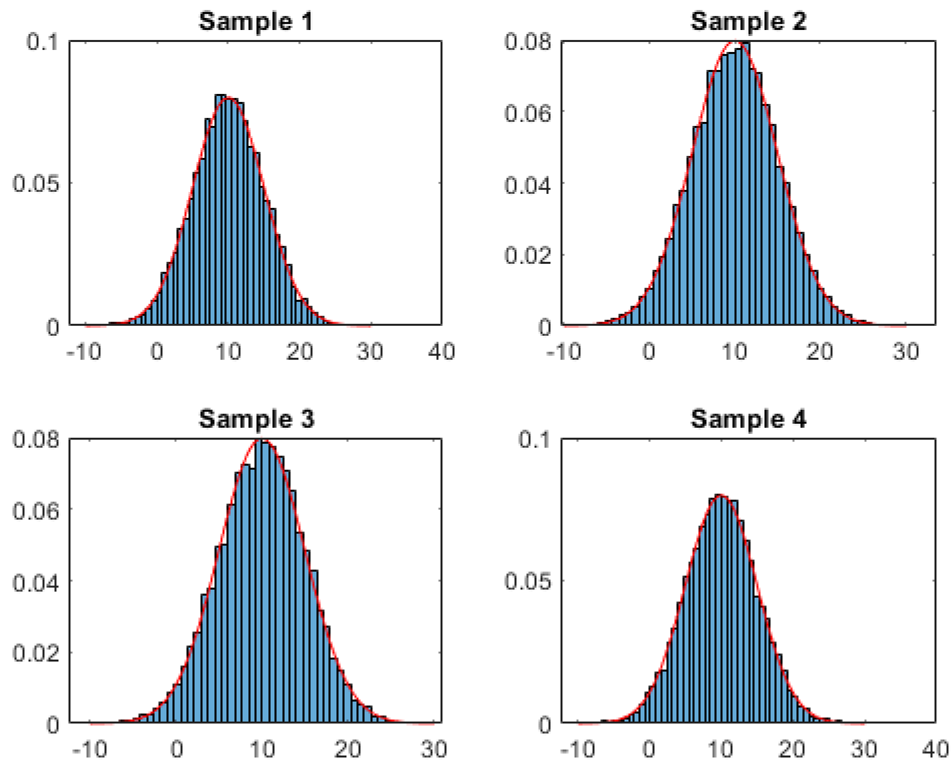
Show CDF of all mu(lambda) given the number of humans

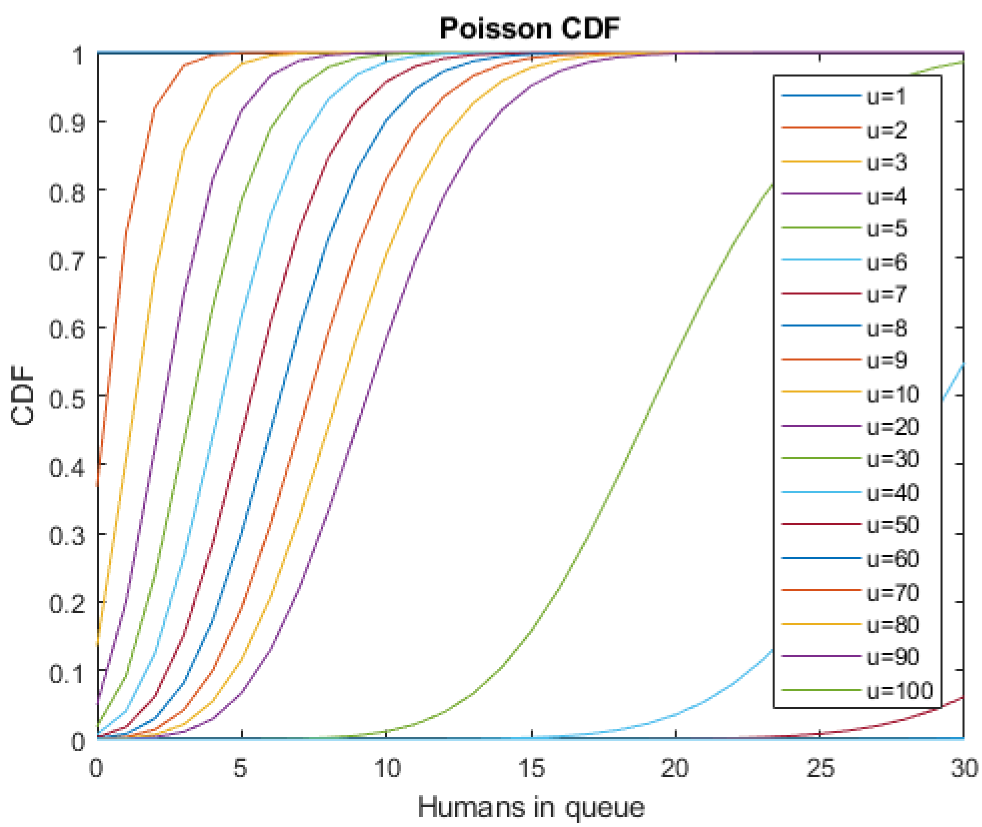
```
figure();
ctr=1;
while ctr<=length(lambda)
    plot(humans,cdf2(ctr,:)); title('Poisson CDF'); xlabel('Humans in
 queue'); ylabel('CDF');hold on;
    legend({'u=1','u=2','u=3','u=4','u=5','u=6','u=7','u=8','u=9',...

  'u=10','u=20','u=30','u=40','u=50','u=60','u=70','u=80','u=90',...
        'u=100'})
    ctr=ctr+1;
end
warning off;
```

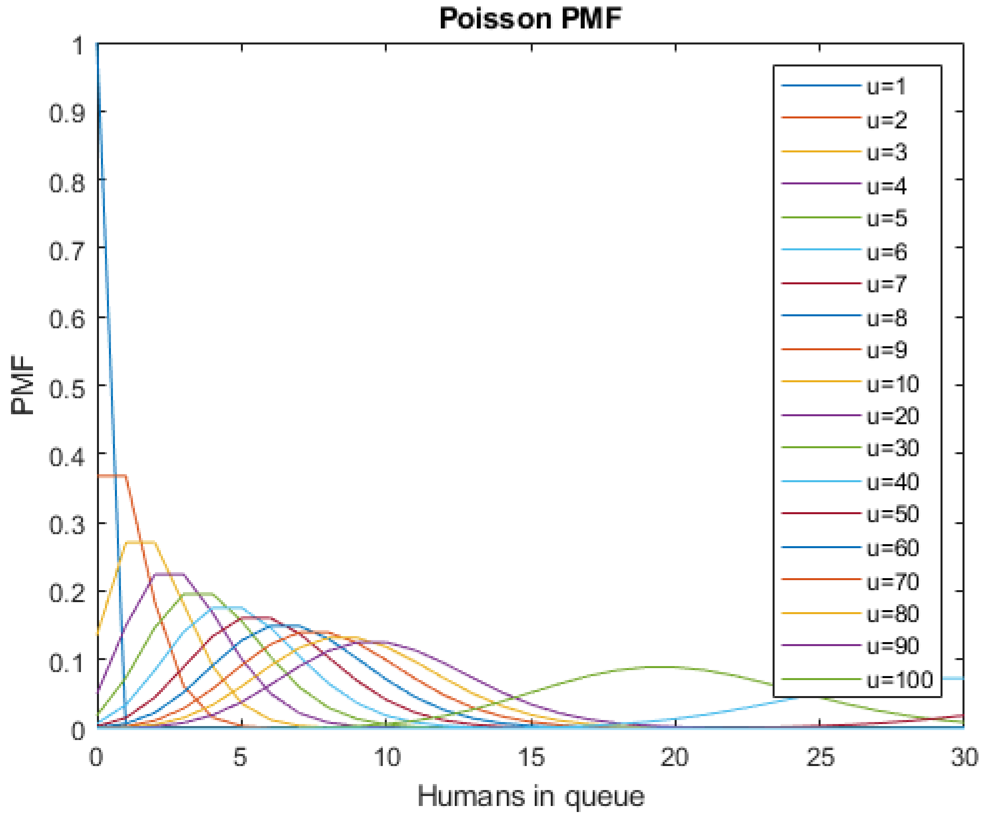


Show PMF of all mu(lambda) given the number of humans

```
figure();
ctr=1;
while ctr<=length(lambda)
    plot(humans,pmf(ctr,:)); title('Poisson PMF'); xlabel('Humans in
 queue'); ylabel('PMF');hold on;
    legend({'u=1','u=2','u=3','u=4','u=5','u=6','u=7','u=8','u=9',...

  'u=10','u=20','u=30','u=40','u=50','u=60','u=70','u=80','u=90',...
        'u=100'})
    ctr=ctr+1;
end
warning off;
```


Sample 1
Sample 2
Sample 3
Sample 4


Poisson CDF
CDF
Humans in queue
u=1
u=2
u=3
u=4
u=5
u=6
u=7
u=8
u=9
u=10
u=20
u=30
u=40
u=50
u=60
u=70
u=80
u=90
u=100



Considering the generated figures of Poisson Distribution above, the lower the interval of event "human in queue", the higher the yielded probability (PMF) is whilst the steeper the CDF slope is. Moreover, these observations are similar with the lambda, the smaller the lambda is, the higher the PMF and the steeper the CDF slope is. When the **lambda value is 2** , there exist a probability of **36.78%** that **atleast one patient is in the queue** and a probability of **22.40%** that **atleast three patients are in the queue** .

Overall, Poisson distribution gives the probability of a number of events in an interval generated by a Possion process. It is degined by the lambda(mu) which is the rate parameter, which is the expected number of events in the interval and the highest probability number of events.

Published with MATLAB® R2021a