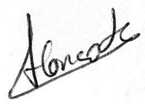
**Telecommunications Engineering**
**Universidad Carlos III de Madrid**
**Statistics**

## Assignment 2: Probability and Random Variables

| Group | Students | Signatures |
|---|---|---|
| | • Alonso Herreros Copete |  |

**IMPORTANT:** *The teachers of this course apply a 'zero tolerance' policy regarding academic dishonesty. Students that sign up this document agree to deliver an original work. The breach of this commitment will result in academic punishment.*

**Observations**:

Solve the exercises in the **Assignment2.pdf** file. **Note:** It is advisable to consult the manual for basic operation of MATLAB / Octave available on the website of the course.

---

# Probability

## Exercise 1.

Given the circuit shown in the figure, consisting of components A, B, C y D where all of them work independently.

a) Compute by simulation using MATLAB/Octave, the probability that the system works, assuming that P(A) = 0,80, P(B) = 0,85 y P(C) = P(D) = 0,75. What is the theoretical result that the simulation should be approximate? Justify your answer.

Since they are all independent, $P(X \cap Y) = P(X) \cdot P(Y)$.
$S$ is the event that the system works.

The theoretical result can be calculated as follows:

$$P(S) = P(A \cap B \cap (C \cup D)) = P(A) \cdot P(B) \cdot (P(C) + P(D) - P(C) \cdot P(D)) =$$
$$= 0.8 \cdot 0.85 \cdot (0.75 + 0.75 - 0.75 \cdot 0.75) = 0.6375$$

The system was simulated 10000 times in MatLab using logical operators, and the probability of a random simulation working turned out to be $0.6389$, which is close enough to the expected value to be considered a correct simulation.

```
n = 10000;
pA = 0.8; pB = 0.85;pC = 0.75; pD = 0.75; pE = 0.75; pF = 0.75;
x = (rand(n,1)<pA) & (rand(n,1)<pB) & ((rand(n,1)<pC) | (rand(n,1)<pD));

disp("a) Probability of working");
prob_theoretical = pA*pB*(pC+pD-pC*pD);
prob = mean(x);
disp("By calculation: " + prob_theoretical);
disp("By simulation: " + prob);
disp(" ");
```

```
a) Probability of working

By calculation: 0.6375
By simulation: 0.6389
```

b) Compute by simulation using MATLAB/Octave the increase in the system reliability (in percentage) when we add two components E y F (that work independently) with working probability equal to 0,75 in the final subsystem formed by the components C and D. What is the theoretical result that the simulation should be approximate? Justify your answer. Note: The increase in the reliability is calculated as: Prob. new syst.−Prob. original syst. Prob. original syst. × 100 %.

Since it is not really specified, I assumed the components E and F were added in parallel to the subsystem.

The theoretical result can be calculated as follows:

$$P(S') = P(A \cap B \cap (C \cup D \cup E \cup F)) =$$
$$= P(A) \cdot P(B) \cdot (P(C) + P(D) + P(E) + P(F) - P(C) \cdot P(D) - P(C) \cdot P(E) -$$
$$- P(C) \cdot P(F) - P(D) \cdot P(E) - P(D) \cdot P(F) - P(E) \cdot P(F) + P(C) \cdot P(D) \cdot P(E) +$$
$$+ P(C) \cdot P(D) \cdot P(F) + P(C) \cdot P(E) \cdot P(F) + P(D) \cdot P(E) \cdot P(F) -$$
$$- P(C) \cdot P(D) \cdot P(E) \cdot P(F)) =$$
$$= 0.8 \cdot 0.85 \cdot (0.75 + 0.75 + 0.75 + 0.75 - 0.75 \cdot 0.75 - 0.75 \cdot 0.75 -$$
$$- 0.75 \cdot 0.75 - 0.75 \cdot 0.75 - 0.75 \cdot 0.75 - 0.75 \cdot 0.75 + 0.75 \cdot 0.75 \cdot 0.75 +$$
$$+ 0.75 \cdot 0.75 \cdot 0.75 + 0.75 \cdot 0.75 \cdot 0.75 + 0.75 \cdot 0.75 \cdot 0.75 -$$
$$- 0.75 \cdot 0.75 \cdot 0.75 \cdot 0.75) = 0.6773$$

The theoretical increase is: $\frac{P(S')-P(S)}{P(S)} \cdot 100\% = \frac{0.6773-0.6375}{0.6375} \cdot 100\% = 6.25\%$

Another 10000 simulations were made, and the simulated working probability increased by around $6.836\%$, which is close to the expected result.

```
disp("b) Components E and F added to subsystem CD"); % I will assume that E and F are added in parallel.
prob2_theoretical = pA*pB*(pC+pD+pE+pF-pC*pD-pC*pE-pC*pF-pD*pE-pD*pF-pE*pF+pC*pD*pE+pC*pD*pF+pC*pE*pF+pD*pE*
prob2_inc_theoretical = prob2_theoretical / prob_theoretical - 1;
disp("Theoretical increase: " + prob2_inc_theoretical*100 + "%");

x2 = (rand(n,1)<pA) & (rand(n,1)<pB) & ((rand(n,1)<pC) | (rand(n,1)<pD) | (rand(n,1)<pE) | (rand(n,1)<pF));
prob2 = mean(x2);
prob2_inc = prob2 / prob - 1;
disp("Simulated increase: " + prob2_inc*100 + "%");
disp(" ");
```

```
b) Components E and F added to subsystem CD

Theoretical increase: 6.25%

Simulated increase: 6.8362%
```

c) Compute by simulation using MATLAB/Octave, the change in the original reliability of the system when:

- the working probabilities of components A and B are increased by 0,1;
- the working probabilities of components C and D are increased by 0,2.

In which of these cases do we observe a major impact on the original reliability of the system?

In this case, there was no theoretical calculation.

The results were simulated in matlab:

```
disp("c) Changing the working probabilities of A and B or C and D");
pA_ = pA + 0.1;
pB_ = pB + 0.1;
x3 = (rand(n,1)<pA_) & (rand(n,1)<pB_) & ((rand(n,1)<pC) | (rand(n,1)<pD));
prob3 = mean(x3);
prob3_inc = prob3 / prob - 1;
disp("Simulated increase with working probabilities of A and B increased: " + prob3_inc*100 + "%

pC_ = pC + 0.2;
pD_ = pD + 0.2;
x4 = (rand(n,1)<pA) & (rand(n,1)<pB) & ((rand(n,1)<pC_) | (rand(n,1)<pD_));
prob4 = mean(x4);
prob4_inc = prob4 / prob - 1;
disp("Simulated increase with working probabilities of C and D increased: " + prob4_inc*100 + "%
```

```
c) Changing the working probabilities of A and B or C and D

Simulated increase with working probabilities of A and B increased: 25.6322%

Simulated increase with working probabilities of C and D increased: 5.2301%
```

As shown in the simulation, the increase in working probability of the system is almost 5 times greater when increasing the working probabilities of A and B. This is because these components are in serial connection. A single failure of any of these two components will cause the system to fail, so they are crucial for the working of the system.

# Random Variables

## Exercise 2.

Let p = P(S = s) be a probability function defined in the sample space S = {1, 2, 3, 4, 5} with the following probabilities:

| S | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| p | 0.3 | 0.2 | 0.2 | 0.2 | 0.1 |

a) Generate in MATLAB/Octave two independent random variables ($X_1$ and $X_2$) with probability function p. What are the approximate values that the mean and variance of $X_1$ and $X_2$ should have?

The variables $X_1$ and $X_2$ were generated with 1000 samples, stored in vectors x1 and x2. This was done by dividing the range (0,1) in intervals of size corresponding to their probabilities, and then assigning the generated random values between 0 and 1 to integer values, depending on the range in which they fell.

The sum of each value multiplied by its probability can be easily expressed as the dot product between the weights and the values vectors. The variance can be calculated similarly by multiplying the square of the difference between each value and the mean, by its probability.

```
% s |  1  |  2  |  3  |  4  |  5
% p | 0.3 | 0.2 | 0.2 | 0.2 | 0.1
X_values = [1 2 3 4 5];
X_weights = [0.3 0.2 0.2 0.2 0.1];
p_ranges = [0 0.3 0.5 0.7 0.9];

mean_theor = X_weights * X_values';
var_theor = X_weights * (X_values - mean_theor).^2';

n = 1000;
[~, x1] = max((rand(1, n)' >= p_ranges) .* X_values, [], 2);
[~, x2] = max((rand(1, n)' >= p_ranges) .* X_values, [], 2);

disp("a) Mean and variance");
disp("Theoretical: mean = " + mean_theor + ", variance = " + var_theor);
disp("X1:          mean = " + mean(x1) + ", variance = " + var(x1));
disp("X2:          mean = " + mean(x2) + ", variance = " + var(x2));
disp(" ");
```

```
a) Mean and variance
Theoretical: mean = 2.6, variance = 1.84
X1:          mean = 2.577, variance = 1.894
X2:          mean = 2.57, variance = 1.7368
```

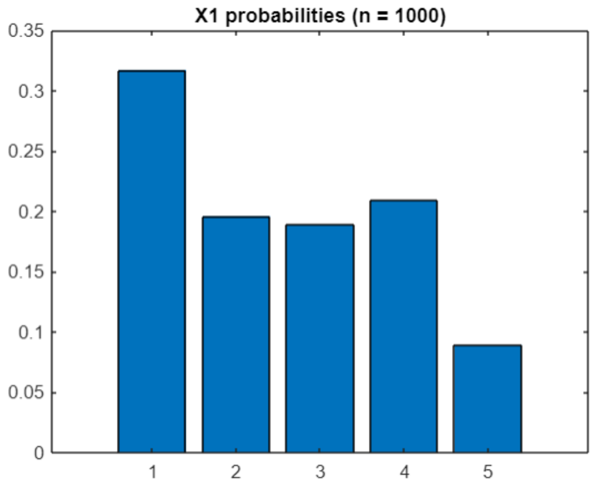
The results are quite similar to the expected values, indicating a reliable simulation.

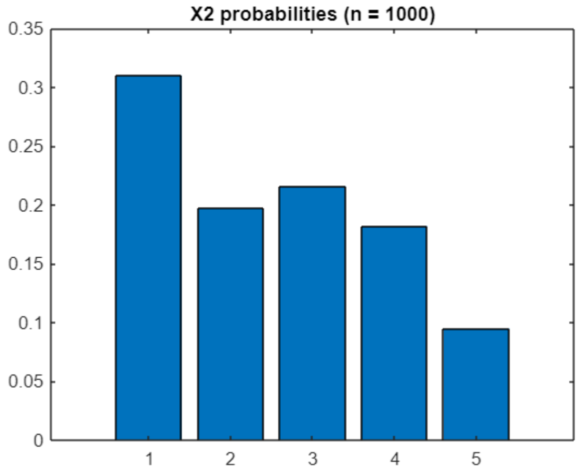
b) Check with a frequency table and a bar diagram that the generation of X1 and X2 is correct

Here are the frequency tables and bar diagrams for both $X_1$ and $X_2$.

| | X1 | Count | Probability |
|---|---|---|---|
| 1 | 1 | 317 | 31.7000 |
| 2 | 2 | 196 | 19.6000 |
| 3 | 3 | 189 | 18.9000 |
| 4 | 4 | 209 | 20.9000 |
| 5 | 5 | 89 | 8.9000 |

| | X2 | Count | Probability |
|---|---|---|---|
| 1 | 1 | 310 | 31 |
| 2 | 2 | 197 | 19.7000 |
| 3 | 3 | 216 | 21.6000 |
| 4 | 4 | 182 | 18.2000 |
| 5 | 5 | 95 | 9.5000 |




These results seem consistent with the parameters established: around 30% for value 1, around 20% for values 2 through 4, and around 10% for value 5.

c) Compute using simulation in MATLAB/Octave the probability $P(2 \times X_1 = X_2)$. What is the theoretical value for the obtained result? Justify your answer.

The theoretical value was calculated using a loop, going over each possible value and calculating the probability of the intersection of $X_1$=i and $X_2$=2·i. The sum function was applied to make sure that the added value was always a number.

```
disp("c) Probability of 2*X1 = X2");

prob_theoretical = 0;
for i = X_values
    prob_theoretical = prob_theoretical + sum(X_weights(X_values == i) .* X_weights(X_values == 2*i));
end

prob = mean(2*x1 == x2);

disp("Theoretical: " + prob_theoretical);
disp("Simulation:  " + prob);
```

```
c) Probability of 2*X1 = X2

Theoretical: 0.1
Simulation:  0.105
```

The results are consistent with the expected values.

## **EXERCISE 3.**

Use the inverse transformation method of the distribution function to generate a continuous random variable with distribution function $F_X(x)$ given by

$$F_X(x) = \begin{cases} 0 & x < 0 \\ \frac{x^2}{2} & 0 \le x \le 1 \\ 1 - \frac{(2-x)^2}{2} & 1 < x \le 2 \\ 1 & x > 2 \end{cases}$$

a) Determine analytically the density function $f_X(x)$.

The density function can be calculated as the derivative of the distribution function.

$$f_X(x) = \frac{d}{dx} F_X(x) = \begin{cases} 0 & x < 0 \\ x & 0 \le x \le 1 \\ 2 - x & 1 < x \le 2 \\ 1 & x > 2 \end{cases}$$

b) Write the **pseudocode** to generate random numbers from the random variable X using the inverse transformation method.

1. Calculate $F_X^{-1}$, the inverse function of the density function $F_X$
2. Generate a random number $u$ in the interval $[0, 1]$.
3. Calculate the random number $x = F_X^{-1}(u)$, where $F_X^{-1}$ is the inverse function of $F_X$

c) Write the MATLAB/Octave code to generate random numbers from the random variable X.

First, we need to find the inverse of the density function $F_X$. This will be done by parts:

$u_1 = \frac{x^2}{2} \Leftrightarrow x = \sqrt{2u_1}$

$u_2 = 1 - \frac{(2-x)^2}{2} \Leftrightarrow 2 - x = \sqrt{2\left(1 - u_2\right)} \Leftrightarrow x = 2 - \sqrt{2\left(1 - u_2\right)}$

Adjusting part ranges:

$0 \leq x \leq 1 \Leftrightarrow 0 \leq \sqrt{2u_1} \leq 1 \Leftrightarrow 0 \leq u_1 \leq \frac{1}{2}$

$1 < x \leq 2 \Leftrightarrow 1 < 2 - \sqrt{2\left(1 - u_2\right)} \leq 2 \Leftrightarrow 2 - 2 \leq \sqrt{2\left(1 - u_2\right)} < 2 - 1 \Leftrightarrow$

$\Leftrightarrow 0 \leq 1 - u_2 < \frac{1}{2} \Leftrightarrow \frac{1}{2} < u_2 \leq 1$

Inverse density function:

$$F_X^{-1}(u) = \begin{cases} 0 & u < 0 \\ \sqrt{2u} & 0 \leq u \leq \frac{1}{2} \\ 2 - \sqrt{2(1 - u)} & \frac{1}{2} < u \leq 1 \\ 0 & u > 1 \end{cases}$$

We can use this function in MatLab to generate 1000 random numbers that follow the distribution of the random variable X and store them in the vector x. The inverse function defined as F_1, then the random numbers between 0 and 1 are generated, and then we apply the inverse function to those in order to generate the random numbers for x.

```
disp("c) Matlab code for the inverse transformation method");
% Simulate X for F(x)
% u1 = x^2/2 => x = sqrt(2*u1)
% u2 = 1 - (2-x)^2/2 => x = 2 - sqrt(2*(1-u2))
%           { 0                 if u <= 0
% F^-1(u) = { sqrt(2*u)         if u <= 1/2
%           { 2 - sqrt(2*(1-u)) if u > 1/2
%           { 0                 if u > 1
F_1 = @(x) (x<=0.5).*sqrt(2*x) + (x>0.5).*(2-sqrt(2*(1-x)));
n = 1000;
u = rand(n,1);
x = F_1(u);
disp(" ");
```

d) Obtain analytically $\mathbb{E}[X]$ and $\mathbb{V}ar[X]$. How can you check if the simulation in c) is performing correctly?

I calculated $\mathbb{E}[X]$ and $\mathbb{V}ar[x]$ by applying their definitions. I used MatLab to calculate the integrals, and also checked the results manually.

```
disp("d) Compute E(X) and Var(X)");
mean_theor = integral(@(x) x.*f(x), -100, 100);
var_theor = integral(@(x) ((x-mean_theor).^2).*f(x), -100, 100);

disp("Theoretical: mean = " + mean_theor + ", variance = " + var_theor);
disp("Simulated: mean = " + mean(x) + ", variance = " + var(x));
```

```
d) Compute E(X) and Var(X)
```

```
Theoretical: mean = 1, variance = 0.16667
Simulated: mean = 1.0197, variance = 0.16243
```

By comparing these theoretical calculations with the generated random numbers, which provide very similar mean and variance, we can check that the simulation is (most likely) working correctly.

## Exercise 4.

Let X be a random variable with density function $f(x)$ defined as

$$f(x) = \begin{cases} \alpha\beta x^{\beta-1}\exp\left(-\alpha x^{\beta}\right) & x > 0 \\ 0 & x \le 0 \end{cases}$$

a) Write a MATLAB/Octave code to generate n = 100000 values of X with α = 0,2 y β = 0,5. Justify your answer.

Since we don't have the distribution function, we'll have to calculate it:

$$F(x) = \int_0^x f(x)dx = \begin{cases} \int_0^x \alpha\beta x^{\beta-1} e^{\left(-\alpha x^{\beta}\right)} dx & x > 0 \\ 0 & x \le 0 \end{cases} =$$

$$= \begin{cases} \left[-e^{\left(-\alpha x^{\beta}\right)}\right]_0^x & x > 0 \\ 0 & x \le 0 \end{cases} = \begin{cases} 1 - e^{\left(-\alpha x^{\beta}\right)} & x > 0 \\ 0 & x \le 0 \end{cases}$$

Then we can use the inverse transform method once again. We just have to find the inverse function:

$$u = 1 - e^{\left(-\alpha x^{\beta}\right)} \Leftrightarrow -\alpha x^{\beta} = \log(1-u) \Leftrightarrow x = \left(-\frac{\log(1-u)}{\alpha}\right)^{\frac{1}{\beta}}$$

You can find a commented line with the $F^{-1}$ function definition using the pre-calculated expression with the values of α and β already applied. I decided to use the parametrized expression instead because it's more maintainable.

```
disp("a) Generate 100000 values of X for α = 0.2 and β = 0.5");
% Find the distribution function F(x) of X using the inverse transform method
% F(x) = 1 - exp(-αx^β)  if x > 0
% u = 1 - exp(-αx^β) => -αx^β = ln(1-u) => x^β = -ln(1-u)/α => x = (-ln(1-u)/α)^(1/β)
% F^-1(u) = (-ln(1-u)/α)^(1/β) = (-ln(1-u)/0.2)^(1/0.5) = (-ln(1-u)*5)^2 = 25*ln(1-u)^2
alpha = 0.2;
beta = 0.5;
F_1 = @(u) (-log(1-u)/alpha).^(1/beta);
%F_1 = @(u) 25.*log(1-u).^2;
n = 100000;
u = rand(n,1);
x = F_1(u);
disp(" ");
```

b) Given that X has mean and variance defined as:

$$\mu = \mathbb{E}[X] = \alpha^{-\frac{1}{\beta}}\Gamma\left(1 + \frac{1}{\beta}\right) \text{ and } \sigma^2 = \mathbb{V}ar[X] = \alpha^{-\frac{2}{\beta}}\left(\Gamma\left(1 + \frac{2}{\beta}\right) - \left(\Gamma\left(1 + \frac{1}{\beta}\right)\right)^2\right)$$

How would you check that the simulation of X is correct?

**Note:** The Gamma function $\Gamma(\cdot)$ for positive integer values is defined as $\Gamma(n) = (n - 1)$

First, I will calculate the theoretical mean and variance of the random variable X. Then, I will compare these theoretical values with the mean and variance of the randomly generated numbers in the vector x.

I calculated these values with the provided formulas and using MATLAB to do the numerical calculations, and also solved them manually to check I got the correct results. Then, I compared these theoretical calculations with the mean and variance of the samples in the vector x.

```
disp("b) Check if the simulation is correct");
% μ   = E[X]   = α^(-1/β)*Γ(1+1/β)                = 0.2^(-1/0.5)*Γ(1+1/0.5) = 0.2^(-2)*Γ(3)
% σ^2 = Var[X] = α^(-2/β)*(Γ(1+2/β) - Γ(1+1/β)^2) = 0.2^(-2/0.5)*(Γ(1+2/0.5) - Γ(1+1/0.5)^2
theoretical_mean = alpha^(-1/beta)*gamma(1+1/beta);
theoretical_variance = alpha^(-2/beta)*(gamma(1+2/beta) - gamma(1+1/beta)^2);
disp("Theoretical: mean = " + theoretical_mean + ", variance = " + theoretical_variance);
disp("Simulation: mean = " + mean(x) + ", variance = " + var(x));
disp(" ");
```

```
b) Check if the simulation is correct


Theoretical: mean = 50, variance = 12500
Simulation: mean = 49.9399, variance = 12729.7648
```

The simulated results were very close to the expected ones, so we can conclude that the simulation is correct.