TELECOMMUNICATIONS ENGINEERING

STATISTICS

2022-2023

COMPUTER LAB SESSION # 2:
PROBABILITY AND RANDOM VARIABLES

AIMS: Introduction to probability by means of random experiments simulation. Introduction to random variables.

1. Probability

1. Simulate the experiment of tossing 1000 times a coin with probability of head (coded as 1) equal to 0.4 and probability of tail (coded as 0) equal to 0.6. How can you check if the simulation of the experiment has been running as expected?

```
>> n=1000; p=0.4;
>> u=rand(n,1);
>> m=1*(u<=p)+0*(u>p);
>> mean(m) % aprox. 0.40
```

2. Simulate the experiment of tossing 10 times a coin with probability of head (coded as 1) equal to 0.4 and probability of tail (coded as 0) equal to 0.6. Locate in a vector the tosses where a head was obtained.

```
>> n=10; p=0.4;
>> u=rand(n,1);
>> m=1*(u<=p)+0*(u>p);
>> toss=(1:n)';
>> data=[toss m]
>> locate=data(find(data(:,2)==1))
```

3. The probability of rain in a day is $p = 0.5$. Give the MATLAB/Octave's code to obtain the probability of rain during a weekend (assume independence between raining days). Compare the results with the theoretical probability.

```
>> p=0.5;
>> prob_theoretical=p+p-p^2
   prob_theoretical =
                  0.7500
```

```
% By simulation:
>> n=10000;
>> u=rand(n,2) ;  % simulate 2 columns (saturday and sunday)
>> x=(u<=p);      % if u<=p, takes the value 1 and 0 otherwise
>> w=(x(:,1)==1 | x(:,2)==1); % if it rains on saturday, sunday
                              % or both days, w=1 and w=0  otherwise
>> prob=sum(w)/n; % aprox. 0.75
```

4. Give the MATLAB/Octave's code to compute the reliability of a serial system of 5 components with probability of failure equal to 0.05 for each component. Assume failures are independent. Compare the results with the theoretical probability.

```
>> p=1-0.05;
>> prob_theoretical_reliability=p^5;
   prob_theoretical_reliability=
                           0.7738
% By simulation:
>> n=10000;
>> u=rand(n,5);
>> x=(u<=p);
>> w=(x(:,1)==1 & x(:,2)==1 & x(:,3)==1 & x(:,4)==1 & x(:,5)==1);
>> prob=sum(w)/n % aprox. 0.7738
```

5. The company "Aikon" has a manufacturing factory of mobile phones. It is known that the 30 % of the mobile phones are defective. If a mobile phone is defective, the probability that a robot from the Quality Department detects the failure is 0.9, and if the mobile phone is not defective, the probability that the robot detects it as defective and takes it out is 0.2. If a customer buys two mobile phones not taken out, what is the probability that both are defective?

Theoretical solution: Given the events $D =$ "is defective", $\bar{D} =$ "not defective", $R =$"the robot detects the failure", and $\bar{R} =$ "the robot does not detect the failure", we have that:

$$P(D) = 0.3,\ P(R|D) = 0.9 \text{ and } P(R|\bar{D}) = 0.2,$$

therefore $P(\bar{D}) = 1 - P(D) = 0.7$, $P(\bar{R}|D) = 0.1$ and $P(\bar{R}|\bar{D}) = 0.8$. Hence, the probability that a mobile phone not taken out is not defective is:

$$P(\bar{D}|\bar{R}) = \frac{P(\bar{R}|\bar{D})P(\bar{D})}{P(\bar{R})} = \frac{P(\bar{R}|\bar{D})P(\bar{D})}{P(\bar{R}|\bar{D})P(\bar{D}) + P(\bar{R}|D)P(D)} = 0.949$$

Finally, the probability that both mobiles are defective is $0.949 \times 0.949 = 0.9006$.

Using MATLAB, we can calculate that probability (without the use of Bayes Theorem) by simulation and the probabilities given in the problem:

1. Simulate n mobile phones (for example: $n = 10000$). We know that the 30 % are defective (and 70 % are not defective).

```
>> n=10000;
>> n_def=0.3*n; % nº of defective
>> n_no_def=0.7*n; % nº of non defective
```

2. We create a vector M, constructed by DEF and NDEF, which takes values 1 if the phone is defective and 0 otherwise.

```
>> DEF=repmat(1,n_def,1) ;
>> NDEF=repmat(0,n_no_def,1) ;
>> M=[DEF ; NDEF];
```

3. Generate random numbers by means of command rand, of size `n_def` and `n_no_def`.

```
>> u_def=rand(n_def,1);
>> u_no_def=rand(n_no_def,1);
```

4. Using a boolean condition, and provided the probabilities $P(R|D) = 0.9$ and $P(R|\bar{D}) = 0.2$, we create the vector R, which indicates whether the robot detects the failure and it actually is (1) and 0 if it detects a failure when it is not.

```
>> R_DEF=1*(u_def<=0.9) + 0*(u_def>0.9);
>> R_NDEF=1*(u_no_def<=0.2) + 0*(u_no_def>0.2);
>> R=[R_DEF ; R_NDEF];
```

5. Finally, we create the matrix `datos`, which contains in the first column the vector M and the vector R in the second. Therefore we can approximate the probability $P(\bar{D}|\bar{R})$, counting how many times $R = 0$ and $M = 0$, we can use command *find*.

```
>> datos = [ M R ];
>> AIKON = datos(find(datos(:,2)==0));
% with tabulate
>> tabulate(AIKON)

Value Count Percent
0     5631   94.80%
1      309    5.20%
```

We obtain $P(\bar{D}|\bar{R}) \approx 0.949$, in this simulation we obtained 0.9480.

2. Random variables

The following table summarizes some useful MATLAB/Octave's functions to generate random numbers:

Function	Description	Syntax
`rand`	random numbers $\in (0,1)$	`rand(m,n)`
`unifrnd`	random numbers $\in (a,b)$	`unifrnd(a,b,m,n)`
`unidrnd`	random numbers $\in \{1,2,...,N\}$	`unidrnd(N,m,n)`

where `m` and `n` are the numbers of rows and columns, respectively.

2.1. Useful preliminary functions

1. Generate 100 data from a continuous uniform distribution $\mathcal{U}(0,1)$.

```
>> x=rand(100,1)          %   x=unifrnd(0,1,100,1)
```

2. Generate with MATLAB/Octave 5 tosses of a dice.

```
>> x=unidrnd(6,5,1)       % generate 5 tosses
                          % from a discrete uniform between 1 and 6
```

3. Generate with MATLAB/Octave 5 tosses of 3 dices.

```
>> x=unidrnd(6,5,3)       % generate 5 tosses of 3 dices
                          % from a discrete uniform between 1 and 6
```

2.2. Random number generation

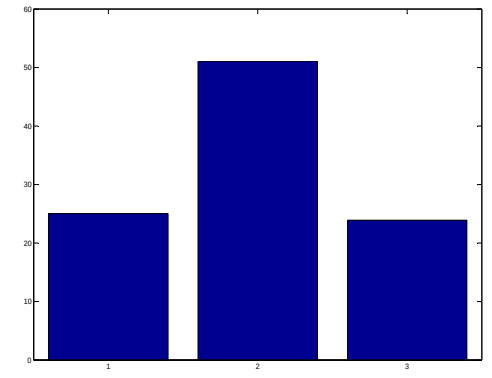
1. Suppose an experiment of tossing two coins. Let X be the random variable of "number of tails". Compute in MATLAB/Octave's the probability function of X.

```
>> n=10000;
>> u1=rand(n,1);
>> u2=rand(n,1);

>> m1=1*(u1<=1/2)+0*(u1>1/2);   % generate n tosses of coins 1 and 2,
>> m2=1*(u2<=1/2)+0*(u2>1/2);   % with equal probability

>> x=m1+m2;    % x is the sum of tails of both coins

>> tabulate(x) % with tabulate command, we obtain the
               % absolute and relative frequencies table

          Value    Count   Percent
              0     2483     24.83%
              1     5006     50.06%
              2     2511     25.11%
```

- Then we can approximate the theoretical probability function:

$$p(x) = P(X = x)$$

```
>> tab=tabulate(x) % save the table in variable tab
>> bar(tab(:,3))   % represent the bar diagram of relative
                   % frequencies
```



X	$p(x)$
0	1/4
1	1/2
2	1/4

2. Given the random variable X of previous exercise, check the theoretical values $\mathrm{E}[X]$ and $\mathrm{Var}[X]$.

SOLUTION:

✓ The expectation of X is:

$$\mu = \mathsf{E}[X] = \sum_{i=1}^{n} x_i \times P(X = x_i) = 0 \times \frac{1}{4} + 1\frac{1}{2} + 2 \times \frac{1}{4} = 1$$

And variance:

$$\sigma^2 = \mathsf{Var}[X] = \sum_{i=1}^{n}(x_i - \mu)^2 P(X = x_i)$$
$$= (0-1)^2 \times \frac{1}{4} + (1-1)^2 \times \frac{1}{2} + (2-1)^2 \times \frac{1}{4} = \frac{1}{2}$$

Recall:

$$\mathsf{Var}[X] = \mathsf{E}[X^2] - (\mathsf{E}[X])^2$$

```
>> mean(x)            % is aprox. 1
>> var(x)             % is aprox. 0.5
```

3. The **inverse transform method**[1] of the distribution function, $F_X(x)$, says "*for a given random variable X with distribution function $F_X(x)$ which admits inverse, then it is verified that the transform variable $\mathcal{U} = F_X(X)$ always follows a continuous uniform distribution $\mathcal{U}$*". This results can be applied in practice considering the equality $u = F_X(x)$ and isolating x as a function of u, given by $x = F_X^{-1}(u)$. Therefore generating $u \sim \mathcal{U}(0,1)$, we can obtain $x = F_X^{-1}(u)$ which follows the distribution of X.

   In order to generate in MATLAB/Octave $u \sim \mathcal{U}$ we can apply the commands `rand` or `unifrnd`.

   a) Let X be a random variable with distribution function $F_X(x)$, given by:

$$F_X(x) = \begin{cases} 0 & , \ x < 0 \\ 1 - e^{-2x} & , \ x \geq 0 \end{cases}$$

   How would you simulate X?

   b) Compute in MATLAB/Octave the values of $\mathrm{E}[X]$ and $\mathrm{Var}[X]$.

SOLUTION:

a) Using inverse transform method and considering the equality $u = F_X(x)$, we have that for $x \geq 0$:

$$1 - e^{-2x} = u$$
$$1 - u = e^{-2x}$$
$$-\frac{1}{2}\log(1-u) = x$$

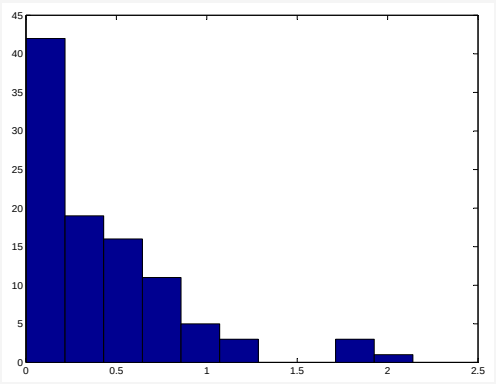
If we generate $u \sim \mathcal{U}(0,1)$, we obtain $x \sim f(x)$. The pseudocode would be:

1: `Fix` $n = 100$
2: `Generate` n `data` $u \sim \mathcal{U}(0,1)$
3: `Apply inverse transform method` $(x = -\frac{1}{2}\log(1-u))$
4: `We have` $x \sim f(x)$

[1] For more details: link

In MATLAB/Octave, in the `Command Window`:

```
>> n=100;
>> u=rand(n,1);
>> x=(-1/2)*log(1-u);
>> hist(x)
```



Recall: Inverse transform method allows an easy way to simulate random variables from a uniform. In this exercise we obtain exponential random variables of parameter $\lambda = 2$.

b) For given distribution function X, $F_X(x)$, we can obtain the density function $f(x)$ as:

$$f(x) = \frac{dF_X(x)}{dx}$$

therefore:

$$f(x) = \begin{cases} 2e^{-2x} & , \ x \geq 0 \\ 0 & , \ x < 0 \end{cases}$$

$$\mu = \mathsf{E}[X] = \int_{-\infty}^{\infty} x f(x) dx = \frac{1}{2}$$

$$\sigma^2 = \mathsf{Var}[X] = \int_{-\infty}^{\infty} (x-\mu)^2 f(x) dx = \int_{-\infty}^{\infty} \left(x - \frac{1}{2}\right)^2 f(x) dx = \frac{1}{4}$$

```
>> mean(x)                  % is aprox. 0.5
>> var(x)                   % is aprox. 0.25
```

Recall: As noticed previously, $X \sim \mathcal{E}xp(\lambda = 2)$, where $\mathsf{E}[X] = 1/\lambda$ and $\mathsf{Var}[X] = 1/\lambda^2$.

4. Let X be a Rayleigh random variable with parameter α with density function $f\,(x)$ given by:

$$f(x) = \begin{cases} \alpha x \exp\left(-\frac{\alpha}{2}x^2\right) & , \ x > 0 \\ 0 & , \ x \leq 0 \end{cases}$$

with mean and variance:

$$\mu = E\,[X] = \sqrt{\frac{\pi}{2\alpha}}$$

$$\sigma^2 = V\,[X] = \frac{4-\pi}{2\alpha}$$

a) Indicate MATLAB/Octave's code to generate 50 values X with $\alpha = 0.17$?

b) How can you check numerically if the simulation of X is correct?

SOLUTION:

a) One way to simulate X is using inverse transform method of the distribution function. First, we need to calculate the distribution function:

$$F(x) = \begin{cases} \int_{-\infty}^{x} 0dt = 0 & , \quad x \leq 0 \\ \int_{-\infty}^{x} \alpha t \exp\left(-\frac{\alpha}{2}t^2\right) dt = \int_{0}^{x} \alpha t \exp\left(-\frac{\alpha}{2}t^2\right) dt & , \quad x > 0 \end{cases}$$

Therefore

$$F(x) = \begin{cases} 0 & , \quad x \leq 0 \\ \left[\alpha\left(\frac{-2}{\alpha}\frac{1}{2}\right)\exp\left(-\frac{\alpha}{2}t^2\right)\right]_0^t = -\left(\exp\left(-\frac{\alpha}{2}x^2\right) - 1\right) = 1 - \exp\left(-\frac{\alpha}{2}x^2\right) & , \quad x > 0 \end{cases}$$

It holds the four properties of a distribution function ($F(-\infty) = 0, F(+\infty) = 1$, non-decreasing monotonous and right continuous).

With the theoretical result that $Y = F(X)$, if inverse of F exists, follows a $U(0,1)$. It is possible to generalized this result generating u as $U(0,1)$ equal to $F(x)$ and solving the equation for x as function of u. Given $u > 0$ the inverse of F is:

$$1 - \exp\left(-\frac{\alpha}{2}x^2\right) = u$$

$$1 - u = \exp\left(-\frac{\alpha}{2}x^2\right)$$

$$x^2 = \frac{-2}{\alpha} Ln(1-u)$$

$$x = \pm\sqrt{\frac{-2}{\alpha} Ln(1-u)}$$

From previous expression we take the positive square root such as $x > 0$. The pseudocode would be:

- Generate u as $U(0,1)$.

- Finally $x = \sqrt{\frac{-2}{\alpha} Ln(1-u)}$ follows the required random variable (Rayleigh).

```
>> n=50
>> u=rand(n,1);
>> alpha=0.17;
>> x=sqrt((-2/alpha)*log(1-u));
```

b) We can check the results for observed values of the mean and standard deviation for simulated values of x, comparing them with theoretical ones:

```
>> mean_theoretical = sqrt(pi/(2*alpha));
>> mean_theoretical =
                  3.0397

>> var_theoretical = (4-pi)/(2*alpha);
>> var_theoretical =
                  2.5247

>> mean(x)              % aprox. 3.0397
>> var(x)               % aprox. 2.5247
```

5. TRUE or FALSE. In case of TRUE demonstrate it and in case or FALSE give a counterexample or the correct result:

   The MATLAB/Octave's code to generate 100 values from a generalized Gompertz random variable with distribution function:

$$F(x) = \begin{cases} 1 - \exp\left(-\exp\left(x\right) + 1\right) & x > 0 \\ 0 & \text{otherwise} \end{cases}$$

   is

```
>> u=rand(100,1);
>> x=-log(1-u); % log is natural logarithm
```

   SOLUTION:

   ✓ It is FALSE. To generate continuous random variables whose distribution function admits inverse, we consider:

$$F\left(x\right) = u$$

   with $u \sim U\left(0,1\right)$ and isolate x. Therefore

$$F\left(x\right) = 1 - \exp\left(-\exp\left(x\right) + 1\right) = u$$

$$1 - u = \exp\left(-\exp\left(x\right) + 1\right)$$

$$\ln\left(1-u\right) = -\exp\left(x\right) + 1$$

$$x = \ln\left(1 - \ln\left(1-u\right)\right)$$

   Finally the MATLAB/Octave's code is:

```
>> u=rand(100,1);
>> x=log(1-log(1-u));
```

6. Imagine the next game: you have to choose a number among 1 and 6. You also have 3 dices. If you throw the dices and obtain the number you have chosen in three dices you get 3 euros, if only in two you get 2 euros, if only in one you get 1 euro and if you don't match any of them you pay 1 euro. Is it worth to play this game? Create a MATLAB/Octave function to solve the problem.

   SOLUTION:

   ✓ Create the function `ThreeDices.m`:

```
%%% *.m file
 function Expectation = ThreeDices(n,k)
   u=unidrnd(6,n,3);
   a=(u==k);
   s=sum(a,2);
   g=-1*(s==0)+1*(s==1)+2*(s==2)+3*(s==3);
   Expectation=sum(g)/n;
%%%
```

   Using function `ThreeDices.m`, we observe for any value of k, the expectation is negative. Therefore the game is not profitable. The values obtained are approximately close to the theoretical value of -0.08.

```
>> Expectation = ThreeDices(1000,k)
```