

Exercise – Understanding inverse transform sampling

1. Aim of the exercise

The aim of this exercise is to develop an understanding of how to sample from the exponential and truncated exponential distributions using the inverse transform method. By deriving the inverse CDFs and implementing them in MATLAB, this exercise illustrates both the general principle of inverse transform sampling and its application when built-in functions do not support truncated distributions directly.

2. Theory

Before proceeding with the example, review the theoretical foundation of inverse transform sampling as developed in the dedicated theory exercise. The current exercise focuses on their practical implementation.

3. Clear workspace

Remove all existing variables from the workspace to ensure a clean computational environment.

```
13 %% 3. Clear workspace
14
15 % Clear workspace
16 clear;
```

4. Sampling setup

We begin by specifying the sample size, which determines how many uniform draws will be generated. These draws will later be transformed into realizations from the exponential and truncated exponential distributions using the inverse cumulative distribution function (CDF).

```
18 %% 4. Sampling setup
19
20 % Sample size
21 N = 1000;
```

5. Sampling from the exponential distribution

Inverse transform sampling relies on the principle that if $U \sim \text{Uniform}(0, 1)$, then

$$X = F^{-1}(U)$$

has distribution function F , where F^{-1} denotes the inverse CDF.

For the exponential distribution with rate parameter $\lambda > 0$, the CDF is

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

where $x \geq 0$.

To apply inverse transform sampling, set

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

Solving for x :

$$\begin{aligned} e^{-\lambda x} &= 1 - U, \\ -\lambda x &= \ln(1 - U), \\ x &= -\frac{1}{\lambda} \ln(1 - U). \end{aligned}$$

In the MATLAB code, we first simulate N independent random variables U_1, U_2, \dots, U_N from the uniform distribution on $(0,1)$. These serve as the input for the inverse transform method.

Next, we parameterize the exponential distribution by a rate $\lambda > 0$. This parameter controls the expected value

$$\mathbb{E}[X] = \frac{1}{\lambda}$$

and variance

$$\text{Var}(X) = \frac{1}{\lambda^2}$$

of the distribution. In this case, $\lambda = 1$ is chosen for simplicity.

Finally, using the inverse transform sampling formula derived above, each uniform draw U_i is mapped to an exponential realization:

$$X_i = -\frac{1}{\lambda} \ln(1 - U_i).$$

This produces N independent samples from the exponential distribution with rate λ .

```
23 %% 5. Sampling from the exponential distribution
24
25 % 5.1 Generate N realizations from Uniform(0,1)
26 U = random('Uniform',0,1,[N 1]);
27
28 % 5.2. Rate parameter of the exponential distribution
29 lambda = 1;
30
31 % 5.3. Apply the inverse CDF of Exp(lambda)
32 ExponentialRealization = -(1/lambda)*log(1-U);
```

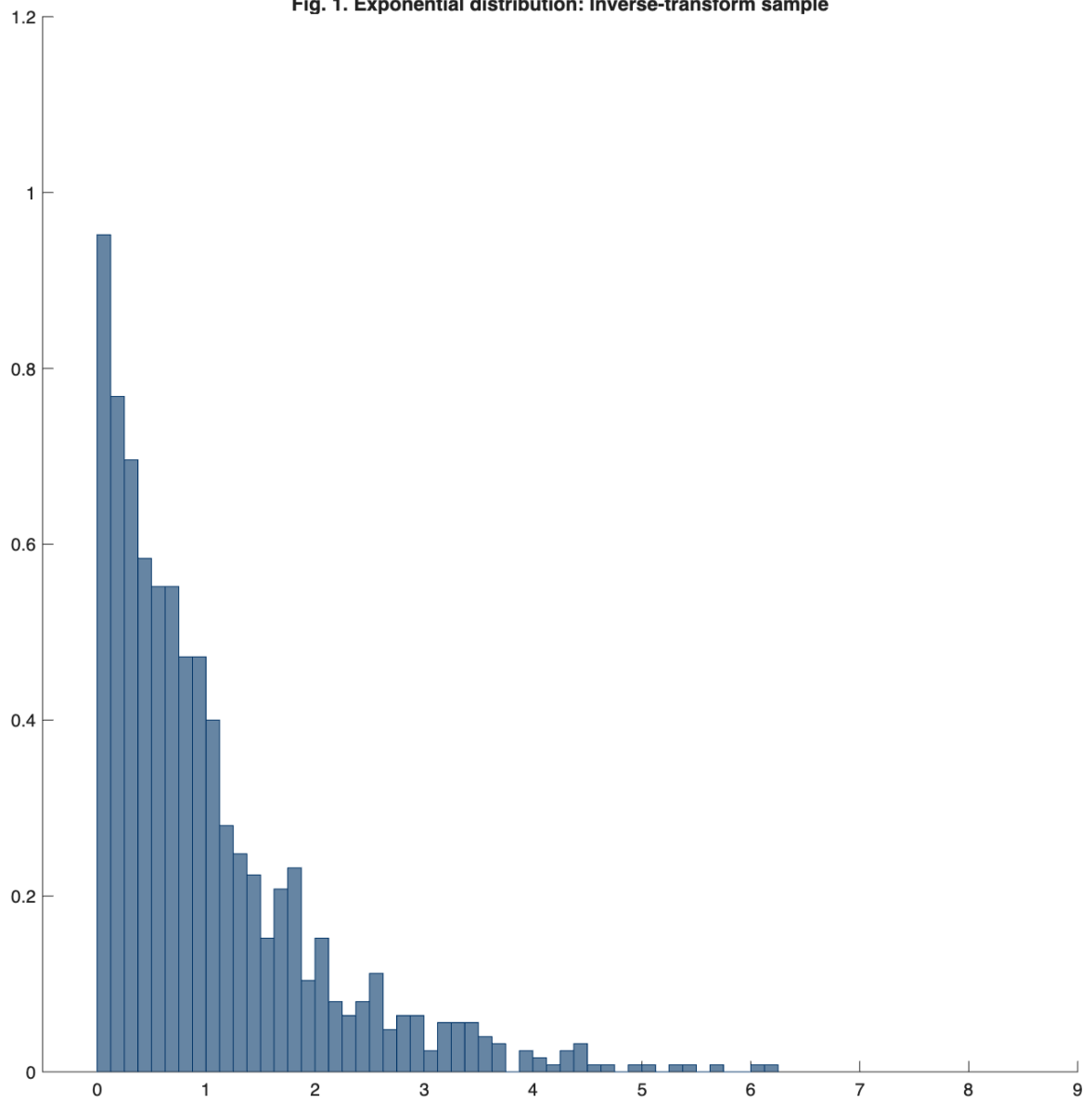
6. Histogram of exponential sample

Here we produce a histogram of the simulated exponential realizations. The histogram is divided into 50 intervals, with bar heights adjusted so the total area equals 1. This makes the plot show the probability density rather than simple counts.

```
34 %% 6. Histogram of exponential sample
35
36 % Plot
37 figure
38 hold on
```

```
39 histogram(ExponentialRealization,50,'Normalization','pdf', ...
40           'EdgeColor',[0.0 0.2 0.4], ...
41           'FaceColor',[0.0 0.2 0.4]);
42 ylim([0.0 1.2])
43 xlim([-0.5 9])
44 title('Fig. 1. Exponential distribution: Inverse-transform sample');
45 hold off
```

Fig. 1. Exponential distribution: Inverse-transform sample



7. MATLAB's built-in exponential generator

Here we show how to sample directly from the exponential distribution using MATLAB's built-in function. We generate N independent draws from an exponential distribution with mean $1/\lambda$. `random` is MATLAB's general random number generator function. It accepts as input arguments the distribution name, the mean parameter, and the desired output size specified as a column vector.

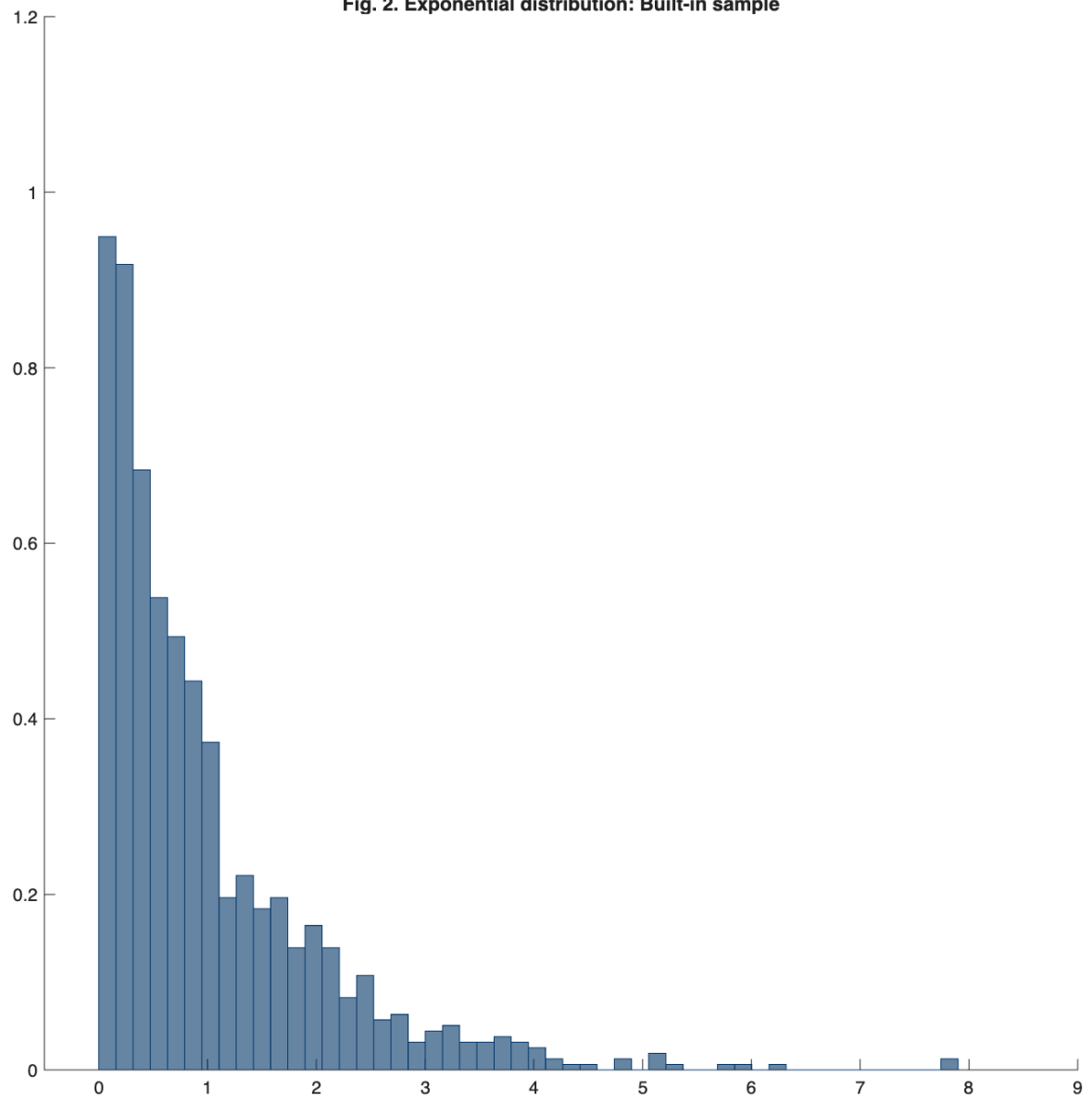
```
47 %% 7. MATLAB's built-in exponential generator
48
49 % Generate exponential sample using MATLAB's built-in function
50 X_builtin = random('Exponential',1/lambda,[N 1]);
```

8: Histogram of MATLAB's built-in exponential sample

Here we plot a histogram of the built-in exponential sample, which shows the same distributional shape as when we generate the sample ourselves using the inverse-transform method.

```
52 %% 8: Histogram of MATLAB's built-in exponential sample
53
54 % Plot
55 figure
56 hold on
57 histogram(ExponentialRealizationBuiltIn,50,'Normalization','pdf', ...
58           'EdgeColor',[0.0 0.2 0.4], ...
59           'FaceColor',[0.0 0.2 0.4]);
60 ylim([0.0 1.2])
61 xlim([-0.5 9])
62 title('Fig. 2. Exponential distribution: Built-in sample');
63 hold off
```

Fig. 2. Exponential distribution: Built-in sample



9. Sampling from the truncated exponential

Consider the truncated exponential distribution $\mathcal{TE}\text{Expo}(1, a)$, obtained by truncating an $\text{Expo}(1)$ distribution at a . If $Y \sim \mathcal{TE}\text{Expo}(1, a)$, its PDF is

$$p_Y(z) = \frac{e^{-z}}{\int_a^\infty e^{-t} dt} \mathbf{1}\{z \geq a\} = e^{-(z-a)} \mathbf{1}\{z \geq a\},$$

where the denominator e^{-a} is the normalization constant.

The CDF of Y is

$$P_Y(x) = \int_a^x e^{-(t-a)} dt = 1 - e^{-(x-a)},$$

where $x \geq a$.

Applying inverse transform sampling, we set

$$u = P_Y(x) = 1 - e^{-(x-a)}.$$

Solving for x gives

$$x = a - \ln(1 - u).$$

Hence, for $U \sim \text{Uniform}(0, 1)$,

$$X = P_Y^{-1}(U) = a - \ln(1 - U)$$

has distribution $\mathcal{TE}\text{Expo}(1, a)$. The inverse transform method provides a simple way to sample from the truncated exponential, though it is limited to the univariate case.

MATLAB's `random` function does not support truncated distributions directly. Here we draw uniform samples and apply the inverse transform formula to generate samples from the truncated exponential distribution. In doing so, we bypass MATLAB's limitation and show the value of implementing the inverse transform ourselves.

```
65 %% 9. Sampling from the truncated exponential
66
67 % 9.1. Truncation point
68 a = 2;
69
70 % 9.2. Draw random numbers between 0 and 1
71 U = random('Uniform',0,1,[N 1]);
72
73 % 9.3. Create the sample that begins at a and extends to infinity
74 TruncExponentialRealization = a-log(1-U);
```

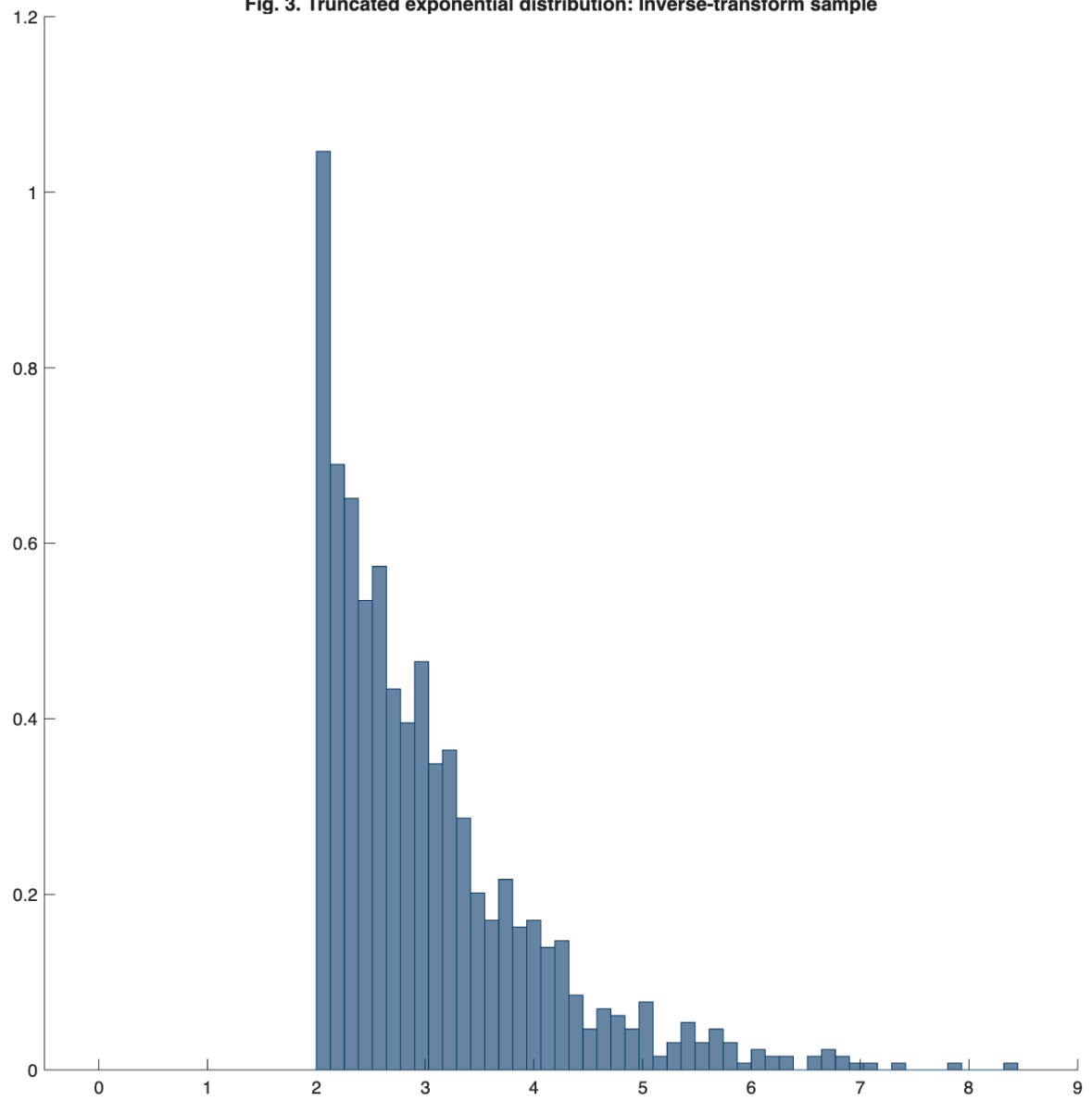
10. Histogram of truncated exponential sample

Here we plot a histogram of the truncated exponential sample. Because the distribution is truncated at a , the histogram begins at $x = a$ and shows the renormalized exponential density restricted to $[a, \infty)$.

```
76 %% 10. Histogram of truncated exponential sample
77
78 % Plot
```

```
79 figure
80 hold on
81 histogram(TruncExponentialRealization,50,'Normalization','pdf', ...
82           'EdgeColor',[0.0 0.2 0.4], ...
83           'FaceColor',[0.0 0.2 0.4]);
84 ylim([0.0 1.2])
85 xlim([-0.5 9])
86 title(['Fig. 3. Truncated exponential distribution: ' ...
87       'Inverse-transform sample']);
88 hold off
```


Fig. 3. Truncated exponential distribution: Inverse-transform sample



11. Final notes

This file is prepared and copyrighted by Jelmer Wieringa and Tunga Kantarcı. This file and the accompanying MATLAB file are available on GitHub and can be accessed using this [link](#).