Exercise – Understanding inverse transform sampling

1. Aim of the exercise

The aim of the exercise is to sampling from the exponential and truncated exponential distributions using inverse transform sampling.

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.

```
%% 3. Clear workspace

% Clear workspace
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).

```
%% 4. Sampling setup

% Sample size
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:

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

In the MATLAB code, we first simulate N independent random variables $U_1, U_2, \ldots, U_N$ 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

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

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.

```
%% 5. Sampling from the exponential distribution

% 5.1 Generate N realizations from Uniform(0,1)
U = random('Uniform',0,1,[N 1]);

% 5.2. Rate parameter of the exponential distribution
lambda = 1;

% 5.3. Apply the inverse CDF of Exp(lambda)
ExponentialRealization = -(1/lambda)*log(1-U);
```

6. Histogram of exponential sample

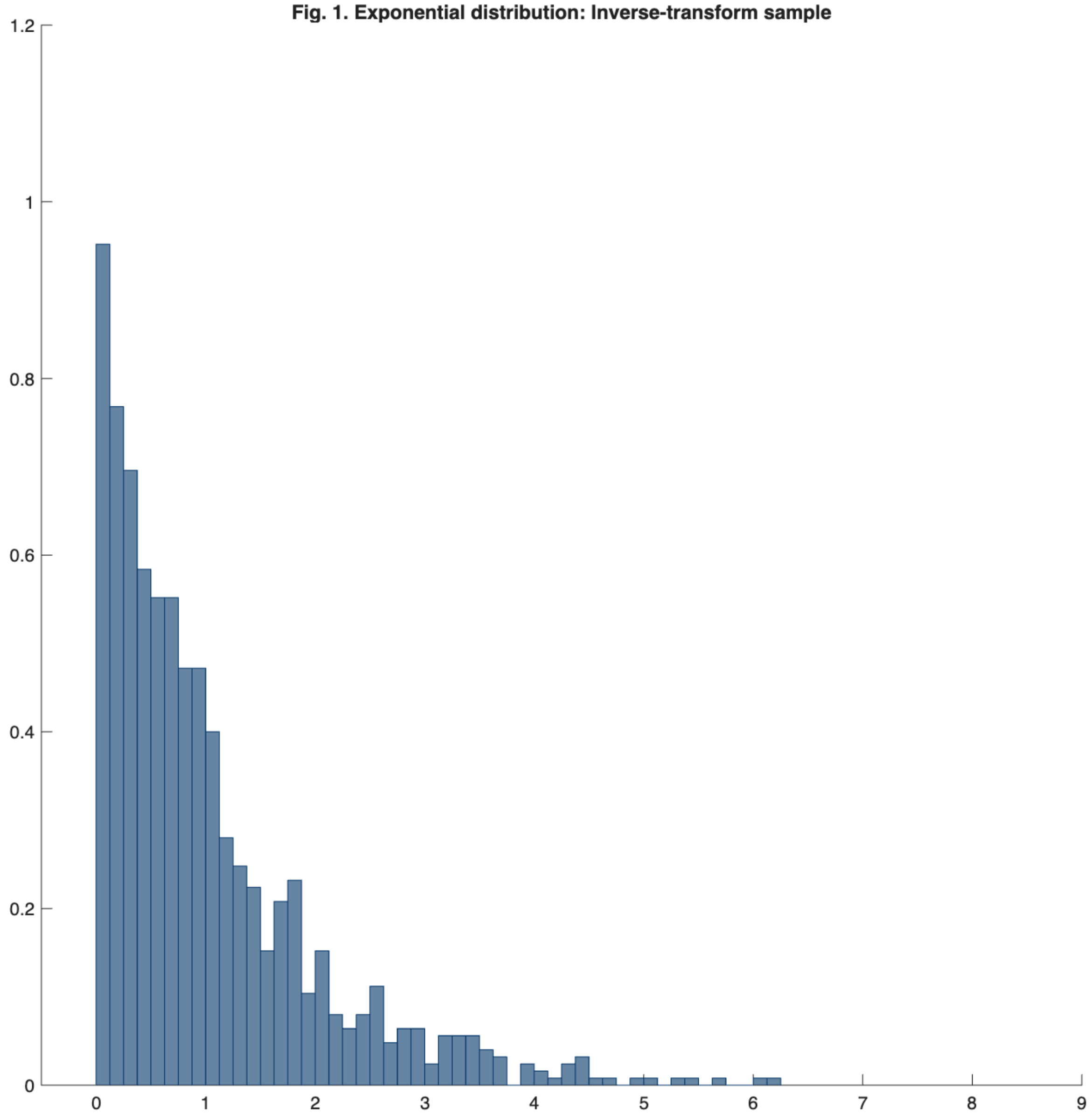
Here we produce a histogram of the simulated exponential realizations. The histogram is divided into 50 intervals, with bar heights adjusted so the total area equals 1. This makes the plot show the probability density rather than simple counts.

```
%% 6. Histogram of exponential sample

% Plot
figure
hold on
histogram(ExponentialRealization,50,'Normalization','pdf', ...
    'EdgeColor',[0.0 0.2 0.4], ...
    'FaceColor',[0.0 0.2 0.4]);
```

```
ylim([0.0 1.2])
xlim([-0.5 9])
title('Fig. 1. Exponential distribution: Inverse-transform sample');
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.

```
%% 7. MATLAB's built-in exponential generator

% Generate exponential sample using MATLAB's built-in function
X_builtin = random('Exponential',1/lambda,[N 1]);
```

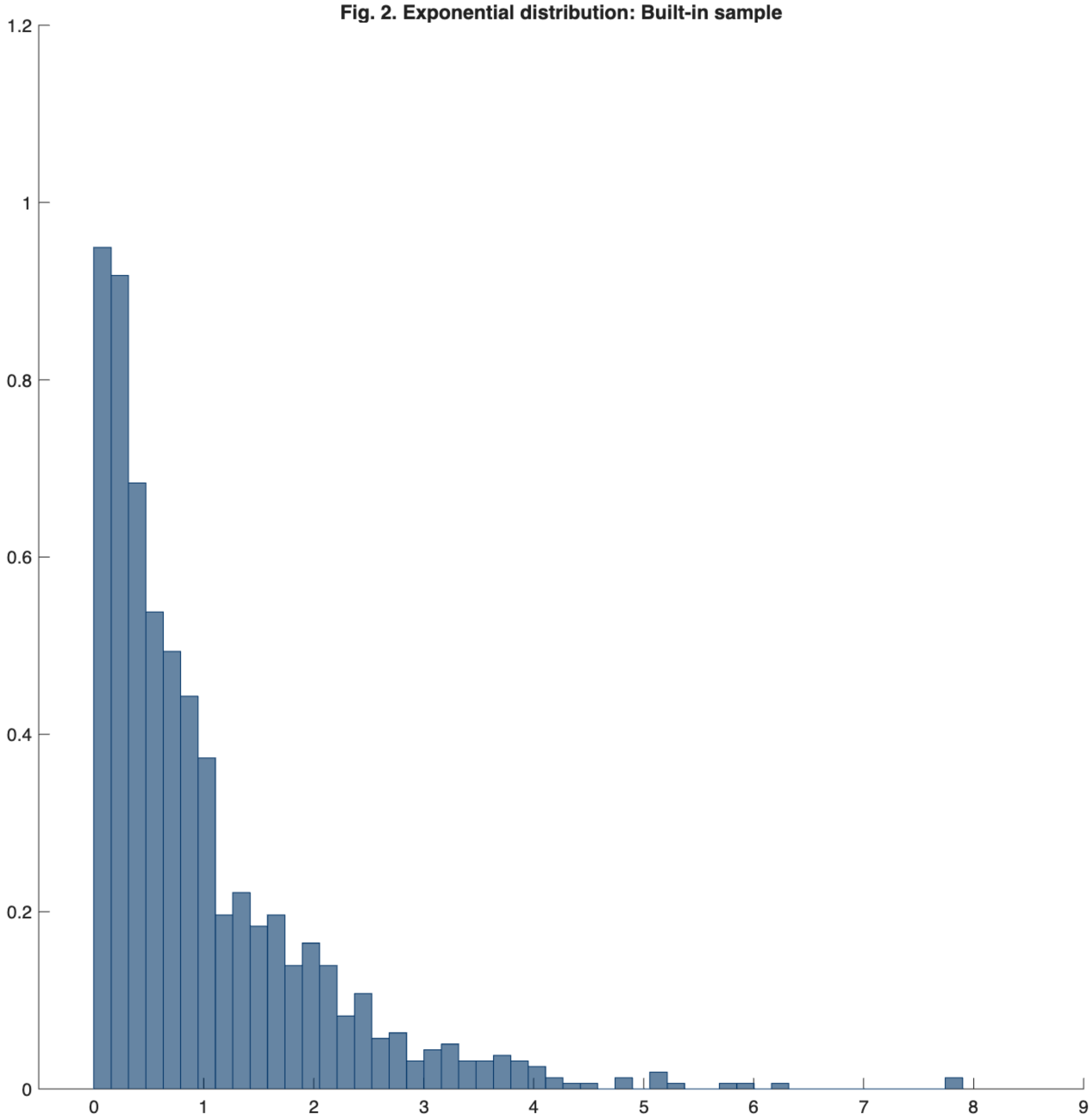
8: Histogram of MATLAB's built-in exponential sample

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.

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

% Plot
figure
hold on
histogram(ExponentialRealizationBuiltIn,50,'Normalization','pdf', ...
    'EdgeColor',[0.0 0.2 0.4], ...
    'FaceColor',[0.0 0.2 0.4]);
ylim([0.0 1.2])
xlim([-0.5 9])
title('Fig. 2. Exponential distribution: Built-in sample');
hold off
```

Fig. 2. Exponential distribution: Built-in sample



9. Sampling from the truncated exponential

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

```
%% 9. Sampling from the truncated exponential

% 9.1. Truncation point
a = 2;

% 9.2. Draw random numbers between 0 and 1
U = random('Uniform',0,1,[N 1]);

% 9.3. Create the sample that begins at a and extends to infinity
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 $[a, \infty)$.

```
%% 10. Histogram of truncated exponential sample

% Plot
```

```
figure
hold on
histogram(TruncExponentialRealization,50,'Normalization','pdf', ...
    'EdgeColor',[0.0 0.2 0.4], ...
    'FaceColor',[0.0 0.2 0.4]);
ylim([0.0 1.2])
xlim([-0.5 9])
title(['Fig. 3. Truncated exponential distribution: ' ...
    'Inverse-transform sample']);
hold off
```

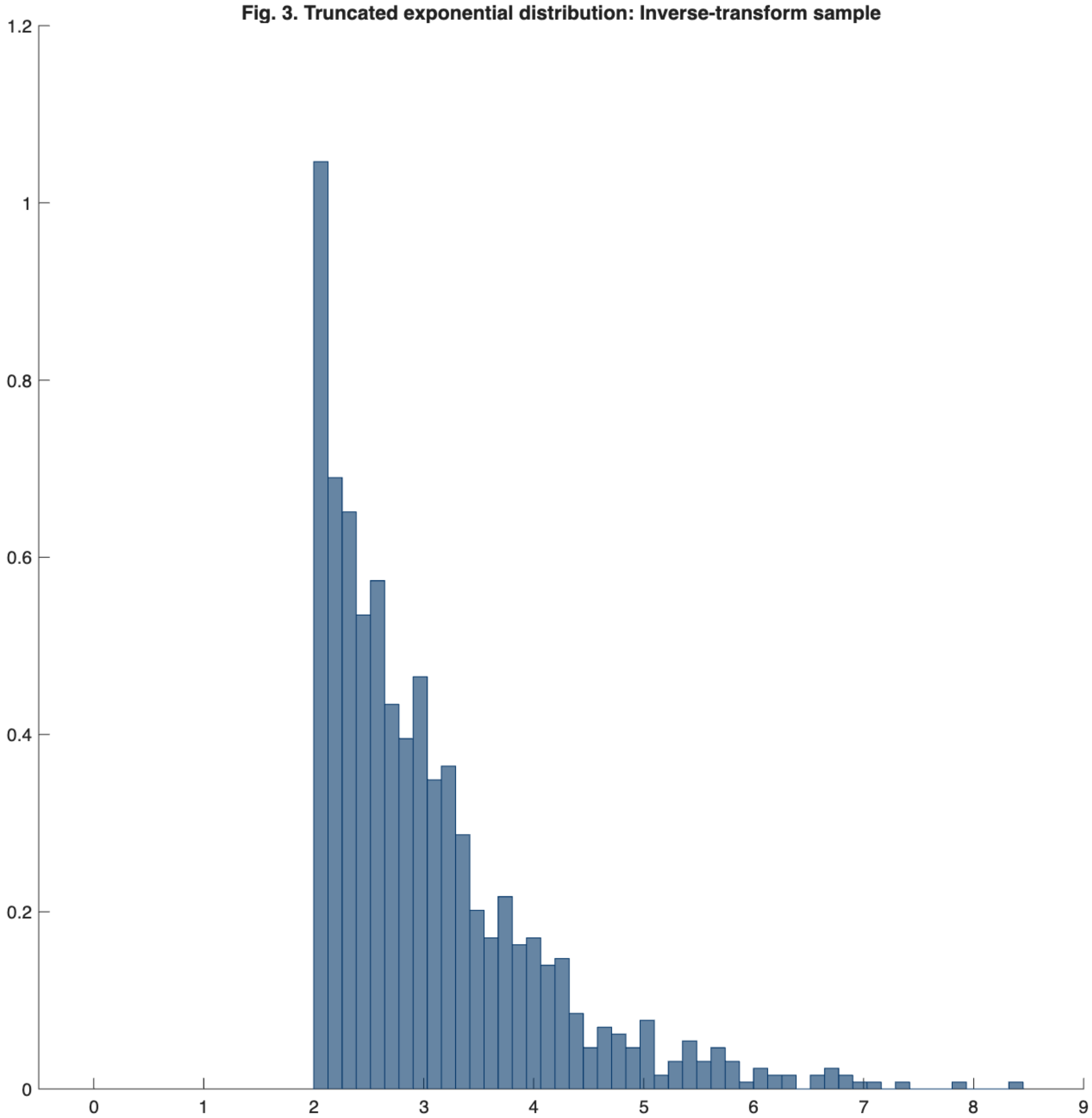


Fig. 3. Truncated exponential distribution: Inverse-transform sample
1.2
1
0.8
0.6
0.4
0.2
0
0
1
2
3
4
5
6
7
8
9

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.