Exercise – Understanding the central limit theorem (CLT) using simulation

1. Aim of the exercise

The CLT studies how the sampling distribution of a sample mean behaves when the sample size increases. We illustrate the theorem using simulation.

2. Theory

Let $\{x_1, ..., x_n\}$ denote a random sample of size n from a population with expected value μ and finite variance σ^2. Consider, the sample mean,

$$\bar{x}_n = \frac{x_1 + \cdots + x_n}{n},$$

as the estimator of the population mean. $\bar{x}_n$ is a random variable. If we take repeated samples of a same size from the population, and obtain a $\bar{x}_n$ from each sample, $\bar{x}_n$ has a sampling distribution. This is called the sampling distribution of the sample mean. Assume that x_i are i.i.d., but importantly, do not assume a specific distribution for them. By the law of large numbers, as $n \to \infty$, the sample average converges in probability to the expected value μ – see our simulation exercise on the law of large numbers. Building on this, the Lindeberg-Levy version of the CLT states that, as $n \to \infty$, the sampling distribution of $\bar{x}_n$ converges to a normal distribution, $\mathcal{N}(\mu, \frac{\sigma^2}{n})$. This is denoted as

$$\bar{x}_n \xrightarrow{d} \mathcal{N}\left(\mu, \frac{\sigma^2}{n}\right).$$

We do not provide the proof of this result. The CLT tells us that, no matter from which distribution our repeated samples come from, the sampling distribution of the mean of these samples will be normally distributed as long as the size of the random samples is large enough. The CLT is powerful because it allows us to make statistical inferences about the population mean using the normal distribution, which is well understood and easy to work with.

The CLT has an important implication. In practice, we can use a limiting distribution as a tool to approximate the true distribution of a statistic when the sample size is finite, that is, its exact distribution, which describes the actual probabilities the statistic takes across repeated samples of a given size. As long as the sample is sufficiently large and the underlying data isn't too extreme (e.g., heavy-tailed or highly skewed), this approximation is often accurate enough for statistical inference. According to the CLT, for example, we can use the limiting normal distribution to construct confidence intervals and perform hypothesis tests when we do not know the true distribution of the sample mean in finite samples. In this way, the limiting distribution becomes a powerful approximation tool for real-world data analysis.

3. Set the parameters of the simulation

We are interested in simulating the behaviour of the sampling distribution of the sample mean as the sample size increases. For this simulation exercise, we will draw random samples of different sizes from a population. Therefore, here we define a population size, alternative sample sizes, and the number of alternative sample sizes. We also define how many samples we will draw from the population at a given sample size.

```
%% 3. Set the parameters of the simulation

% 3.1. Clear the memory
clear;

% 3.2. Define the population size
N_obs_population = 10000;

% 3.3. Define alternative sample sizes
N_obs_sample = [2,15,30,90];

% 3.4. Define the number of samples
N_samples = size(N_obs_sample,2);

% 3.5. Define the number of simulated samples
N_sim = 1000;
```

4. An exponential random variable

4.1. Define the population

We demonstrate the CLT using an exponential random variable. Here we create a population of random values with an exponential distribution, which we will later use to draw random samples. The population is generated using the built-in `random` function, which takes three input arguments. The first argument specifies the distribution type. The second argument is the mean of the exponential distribution, which we set to 1. This value will also be the mean that the sample mean converges to in the simulation. The third argument specifies the size of the population.

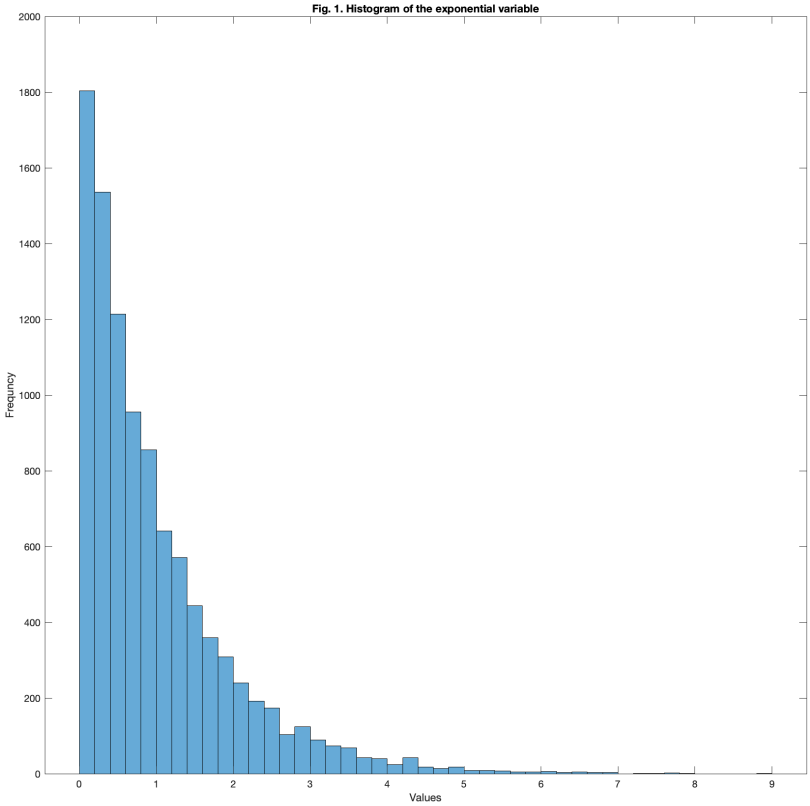
```
%% 4.1. Define the population

% 4.1.1. Define Lambda
Lambda = 1;

% 4.1.2. Define the population
  population = random('Exponential',Lambda,[N_obs_population 1]);
% population = random('Uniform',0,2,[N_obs_population 1]);
```

4.2. Plot the frequency distribution of the exponential variable

Figure 1. presents the histogram of the generated population values.

```
%% 4.2 Plot the frequency distribution of the exponential variable

% 4.2.1. Create the plot
figure
histogram(population);
title('Fig. 1. Histogram of the exponential variable');
ylabel('Frequncy');
```

```
xlabel('Values');
```



4.3. Plot the sampling distribution of the sample mean

Here we draw N_sim random samples from the defined population to construct a sampling distribution for the samples mean. We repeat this exercise N_samples times to generate distributions that differ with respect to the N_obs_sample, that is, sizes of samples we draw from the population. To draw the random samples, we use the randrample function. We supply the function with the input arguments population and N_obs_sample. We also supply the function with the true input argument that allows for sampling with replacement, meaning that the same observation can be selected more than once. After taking the random samples, we calculate their mean using the mean function.

Figure 2 plots the four sampling distributions, in particular their PDF estimates using the function ksdensity. The figure demonstrates the asymptotic behaviour of the sampling distribution of the sample mean as the sample size increases. The sampling distributions approximate the normal distribution well as the sample size increases.

```
%% 4.3. Plot the sampling distribution of the sample mean

% 4.3.1. Preallocate an array to store means of samples
means_samples = NaN(N_sim,N_samples);

```

```
% 4.3.2. Draw random samples from the population and take their mean
for i = 1:N_sim
    for j = 1:N_samples
        sample = randsample(population,N_obs_sample(j),true);
        means_samples(i,j) = mean(sample);
    end
end

% 4.3.3. Create the plot
colors = ['b','r','g','m'];
figure
hold on
for j = 1:N_samples
    [estimated_function_values_j,evaluation_points_j] = ...
        ksdensity(means_samples(:,j));
    plot(evaluation_points_j,estimated_function_values_j, ...
        colors(mod(j-1,length(colors))+1));
    title('Fig. 2. Sampling distribution of the sample mean');
    ylabel('Probability density');
    xlabel('Values');
end
legend_labels = arrayfun(@(x) sprintf('Sample size = %d', ...
    N_obs_sample(x)),1:N_samples,'UniformOutput',false);
legend(legend_labels);
hold off
```

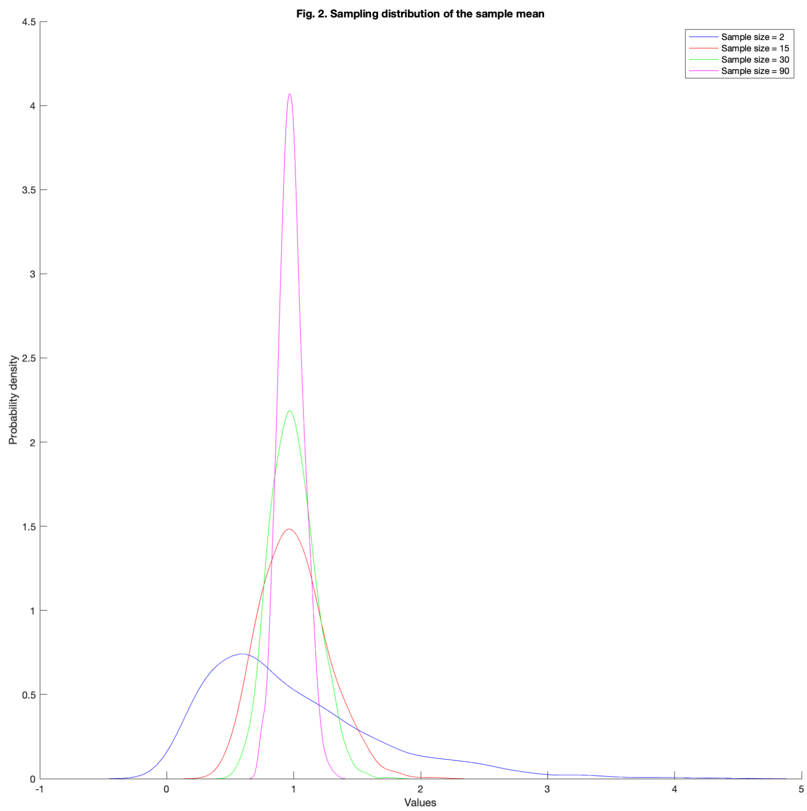


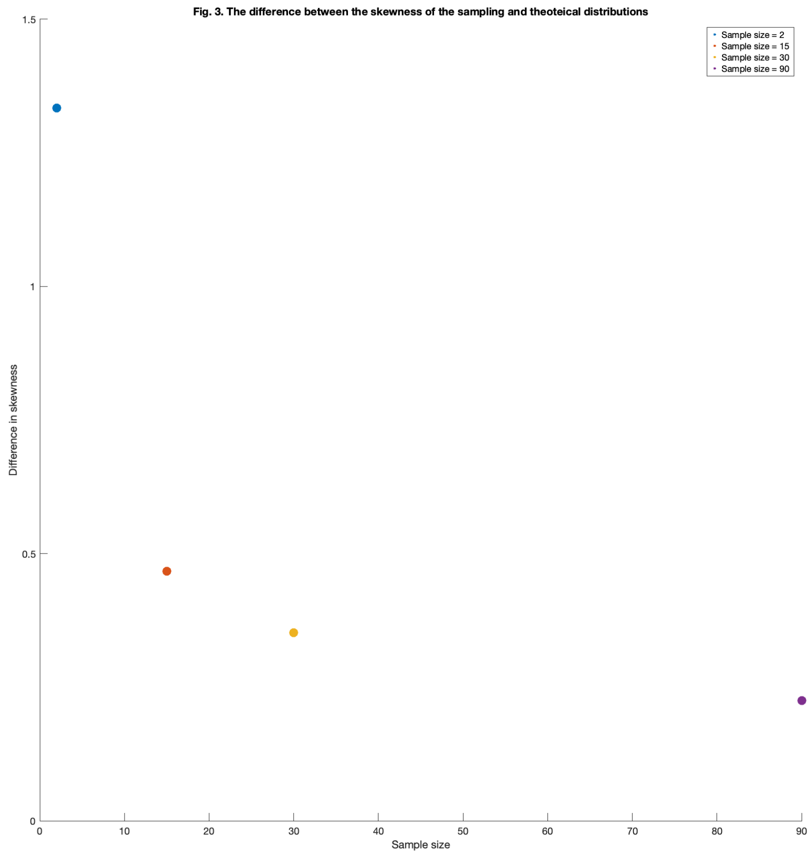
Fig. 2. Sampling distribution of the sample mean

4.4. Speed of convergence of the sampling distribution

To formally analyze the speed at which the sampling distribution of the sample mean converges to the normal distribution, we examine how the skewness of the sampling distribution approaches the theoretical skewness of the normal distribution (which is 0) as the sample size increases. We utilize the `skewness` function for this analysis. This exercise can also be conducted for kurtosis by using the `kurtosis` function instead of the `skewness` function. In Figure 3, we plot the difference between the skewness of the sampling distribution and the theoretical normal distribution for alternative sample sizes. The figure illustrates that, even with a sample size of just 30, the approximation is already quite close. This illustrates the commonly accepted rule of thumb for the sample size required for the CLT to be applicable.

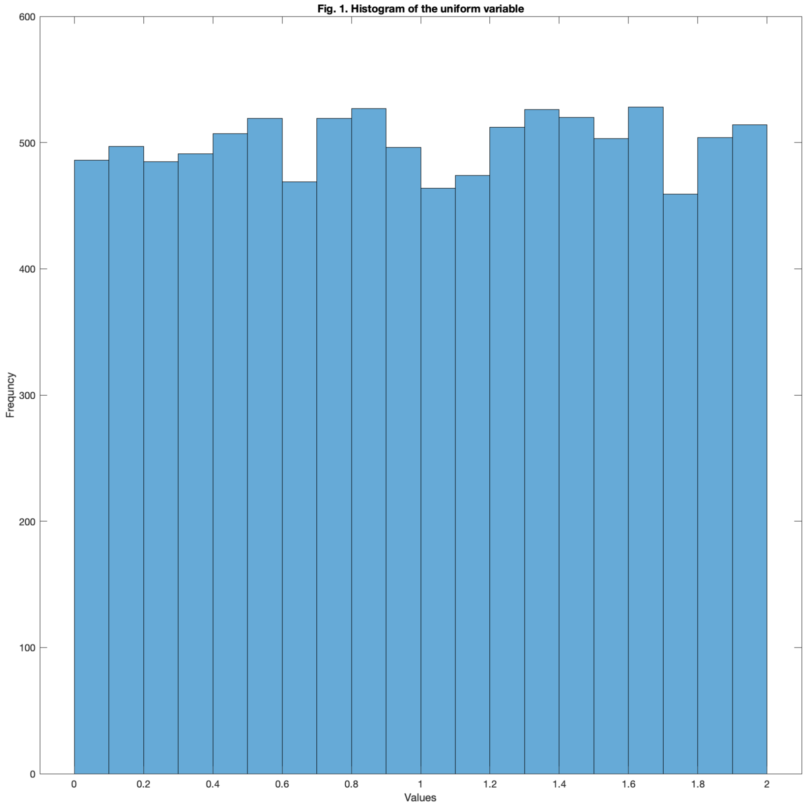
```
%% 4.4. Speed of convergence of the sampling distribution

% 4.4.1. Define the theoretical skewness of the normal distribution
theoretical_skewness = 0;

% 4.4.2. Preallocate matrix to store skewness values
means_samples_skewness = NaN(1,N_samples);

% 4.4.3. Calculate the skewness of the sampling distribution
for j = 1:N_samples
    means_samples_skewness(1,j) = skewness(means_samples(:,j));
end
```

```
% 4.4.4. Define the absolute difference
abs_dif = abs(means_samples_skewness-theoretical_skewness);

% 4.4.5. Create the plot
figure
hold on
for j = 1:N_samples
    scatter(N_obs_sample(j),abs_dif(j),1000,'Marker','.', ...
        'DisplayName',sprintf('Sample size = %d',N_obs_sample(j)));
end
ylim([0 1.5]);
title(['Fig. 3. The difference between the skewness of ' ...
    'the sampling and theoteical distributions']);
ylabel('Difference in skewness');
xlabel('Sample size');
legend('show');
hold off
```



In the remainder of the exercise, we demonstrate how the speed of convergence to normality changes when we sample from a population that follows a uniform distribution instead of an exponential distribution. Figure 4 shows the distribution of the uniform population and we compare it to Figure 1. Distributions that are skewed are known to have slower convergence to normality, while distributions that are symmetric show faster convergence. As compared to Figure 2, Figure 5 shows that, especially when the sample size is smaller, the sampling distribution of the sample mean approximates the normal distribution better. This shows that the exponential distribution requires a larger sample size to exhibit normality. Figure 6 shows that, at a sample size of, for example, 30, the difference between the skewness of the sampling distribution of the sample mean and that of the theoretical distribution is smaller compared to when we sample from an exponential distribution in Figure 3.

Fig. 1. Histogram of the uniform variable



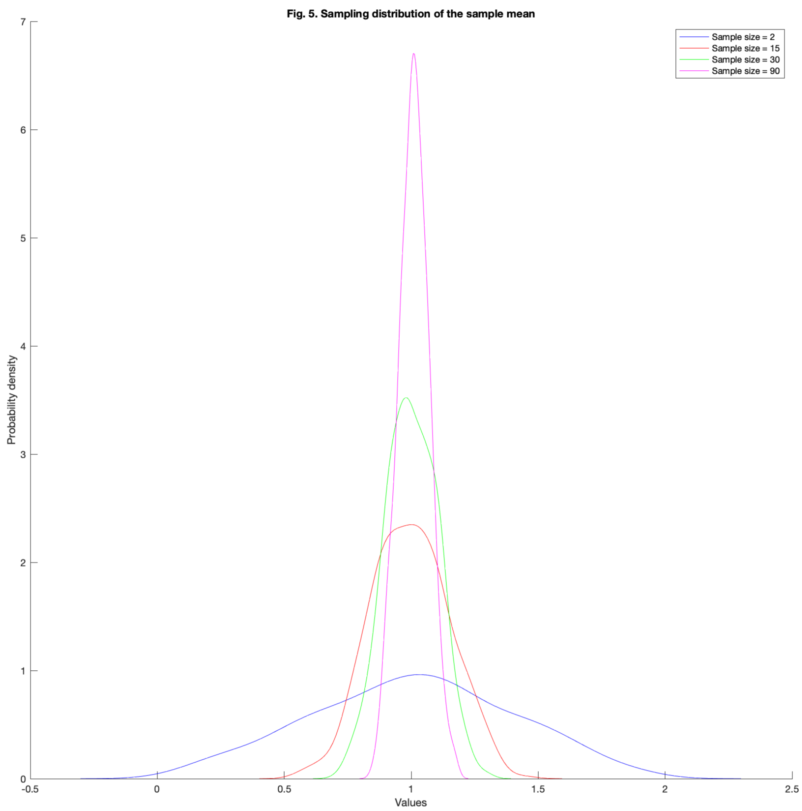
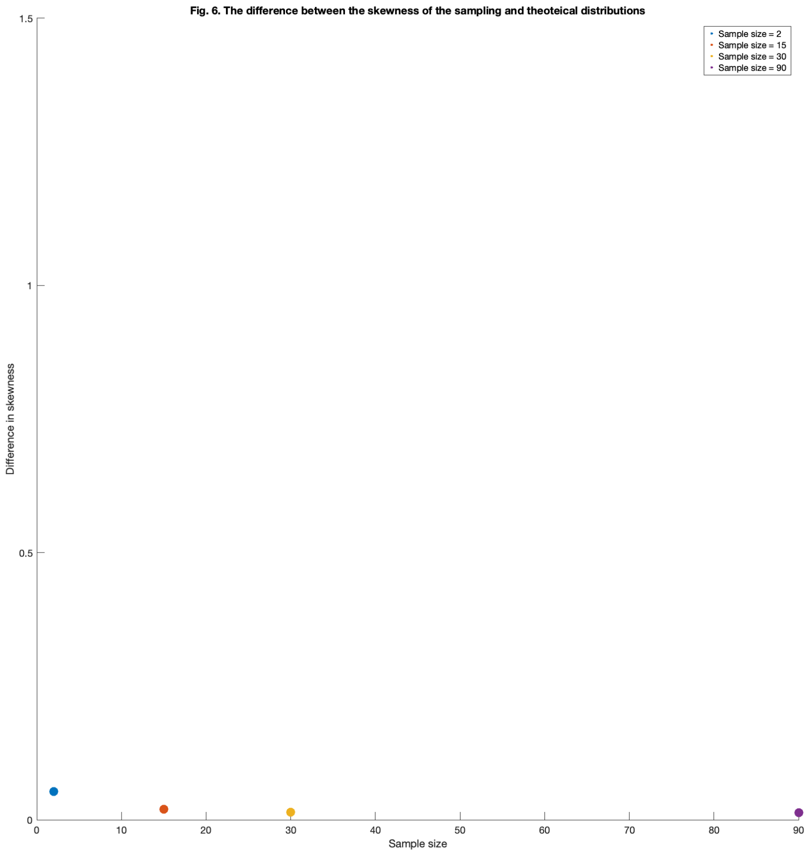


Fig. 5. Sampling distribution of the sample mean
Sample size = 2
Sample size = 15
Sample size = 30
Sample size = 90
Probability density
Values
0
1
2
3
4
5
6
7
-0.5
0
0.5
1
1.5
2
2.5



7. Final notes

This file is prepared and copyrighted by Simonas Stravinskas and Tunga Kantarcı. This file and the accompanying MATLAB file are available on GitHub and can be accessed via this link.