Central Limit Theorem

Student: Giacomo Calabria

Introduction

In this first module we consider the problem of throwing dice multiple times and using and calculating the resulting probability distribution. In this scenario, N_{dice} dice are thrown N_{iter} times.

1 Figure 1

We starty by generating N_{iter} random numbers and use them for the calculation of the probability distribution.

The random variable of interest is

$$x = \mathrm{rand}(6) \tag{1}$$

and the theoretical distribution is uniform and is $p_i = \frac{1}{6}$.

We use the following code we generate the random distribution

```
1 Niter = 1000
2 dice_rolls = [random.randint(1, 6) for _ in range(Niter)]
3 counts = [dice_rolls.count(i) for i in range(1, 7)]
4 pdf = np.array(counts)/Niter
5 plt.bar(range(1, 7), pdf)
```

In line 4 we normalise the counter in order to obtain the probability distribution.

We generated the distribution for $N_{iter} = 10^3$ and $N_{iter} = 10^6$ and reported them in Figure 1

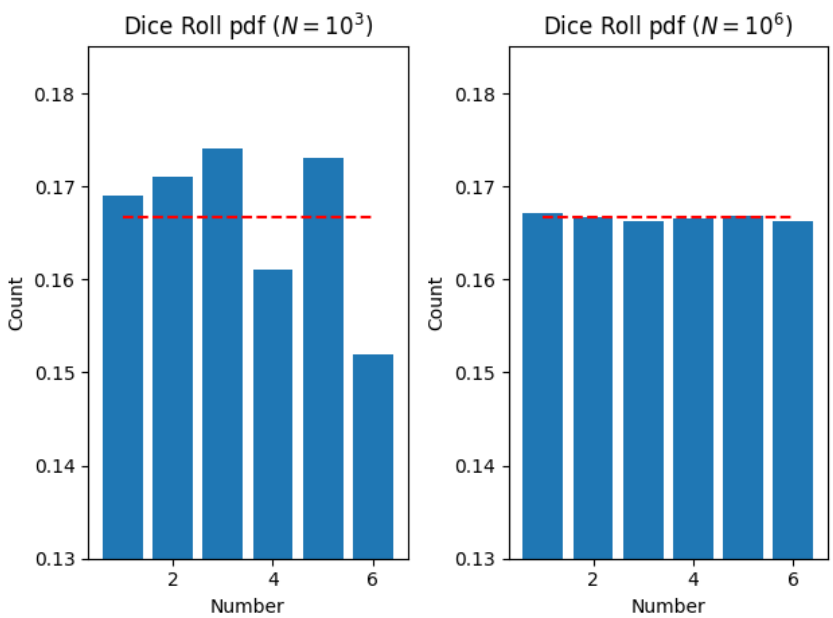

Figure 1: Simple distribution of dice roll

As can be seen as the number of N_{iter} iterations increases, the graph tends toward the normal distribution, which was highlighted with the dashed red line.

2 Figure 2

We now throw a dice twice, so $N_{dice} = 2$ and calculating the average value as

$$x = \frac{\text{rand}(6) + \text{rand}(6)}{2} \tag{2}$$

We are now going to calculate the the probability distribution of the outcome $x = (1, 1.5, 2, 2.5, \ldots, 6)$ and plot it into a figure. We use the following code we generate the random distribution

```
Niter = 100000
for i in range(Niter):
    a = random.randint(1, 6)
    b = random.randint(1, 6)
    dice_outcome[i] = (a + b) / 2
bin_edges = np.arange(1, 7, 0.5)
counts, _ = np.histogram(dice_outcome, bins=bin_edges)
pdf = counts / Niter
plt.bar(bin_edges[:-1], pdf, width=0.5)
```

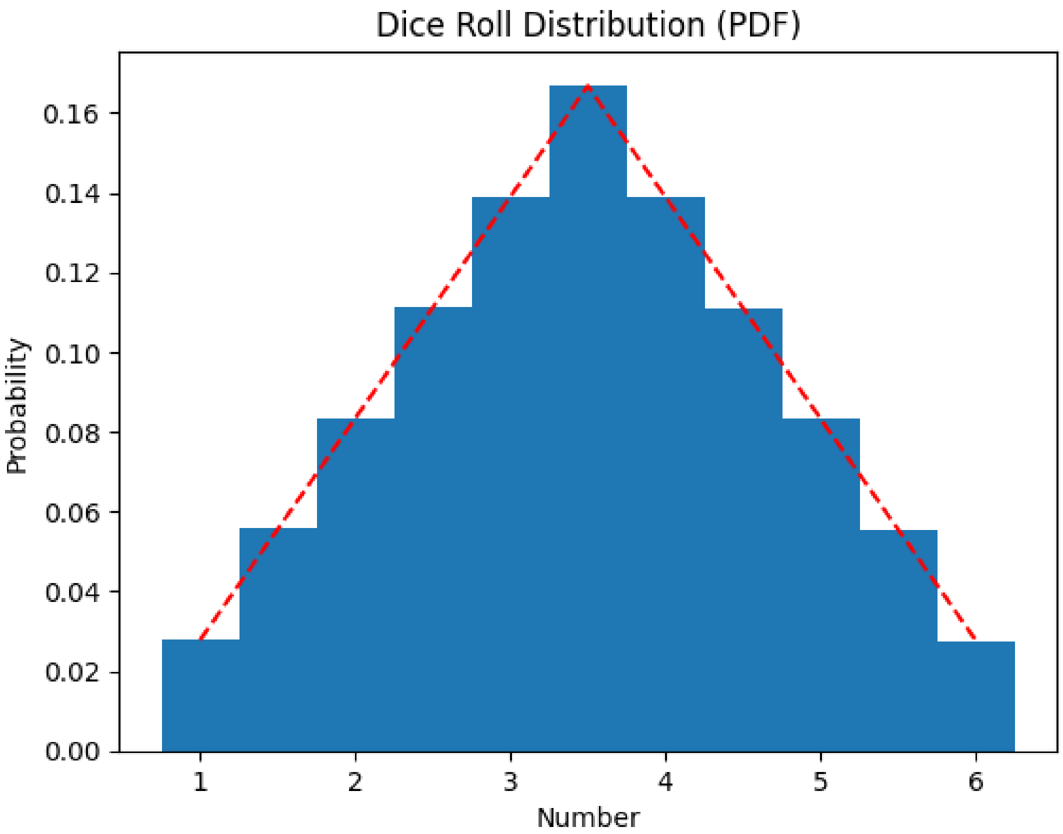

Figure 2: Distribution of the average value of two dice roll

The graph tends toward the normal distribution, which was highlighted with the dashed red line.

3 Figure 3

We now consider the follow random event: throwing a dice N_{iter} times and calculating the average value.

$$x = \frac{\sum_{i=1}^{N_{iter}} \text{rand}(6)}{N_{iter}} \tag{3}$$

With that random event x we calculate the probability distribution $p(x)$. As we use large N we can consider the random value x as a continuous variable. And then we normalise the PDF as $\int p(x)dx = 1$. With this code we generate calculate the probability distribution:

```
Ndice = 100000
Niter = 1000
def random_event():
    return np.sum([random.randint(1, 6) for _ in range(Niter)])/Niter

random_outcome = [random_event() for _ in range(Ndice)]
hist, bins = np.histogram(random_outcome, bins=50, density=True)
bin_centers = (bins[:-1] + bins[1:]) / 2
plt.plot(bin_centers, hist, label='Computed distribution')

x = np.linspace(min(random_outcome)-0.1, max(random_outcome)+0.1, 1000)
gaussian = norm.pdf(x, loc=3.5, scale=np.std(random_outcome))
plt.plot(x, gaussian, label='Gaussian distribution',color='red', linestyle='--')
```

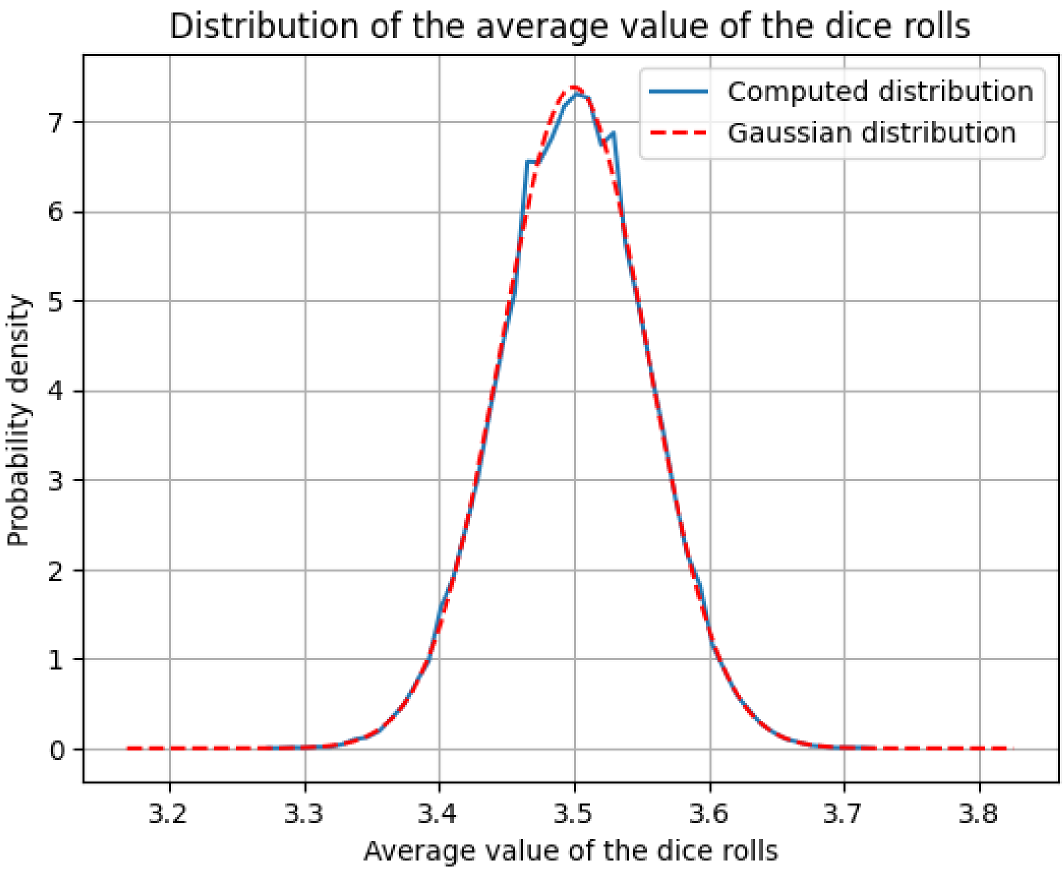

Figure 3: Distribution of the average value of multiple dice roll

As we can see as we use a large number of dice the resulting distribution is pretty similar to the Gaussian distribution expected by the Central Limit Theorem

4 Error estimation

We wanna now calculate the average value and estimate the statistical error, associated with such estimation. Assume that single-die throwing is used to estimate the mean value and the variance

$$\mu = \langle x \rangle \approx \frac{\sum_{i=1}^{N_{iter}} x_i}{N_{iter}} \tag{4}$$

$$\sigma^2 = \langle x^2 \rangle - \langle x \rangle^2 \approx \frac{\sum_{i=1}^{N_{iter}} x_i^2}{N_{iter}} - \left(\frac{\sum_{i=1}^{N_{iter}} x_i}{N_{iter}} \right)^2 \tag{5}$$

Calculate the mean value by throwing the dice $N_{iter} = 10$ and $N_{iter} = 100$ times and compare the estimation of the mean value and the variance with the exact values, given by

$$\mu = \langle x \rangle = \frac{\sum_{\ell=1}^{6} \ell}{6} = 3.5 \quad \sigma^2 = \frac{\sum_{\ell=1}^{6} \ell^2}{6} - \left(\frac{\sum_{\ell=1}^{6} \ell}{6} \right)^2 = \frac{35}{12} \approxeq 2.92$$

Using the following code

```
Niter = 10
dice_rolls = np.array([random.randint(1, 6) for _ in range(Niter)])

u = np.sum(dice_rolls)/Niter
var = np.sum(dice_rolls**2)/Niter-np.mean(dice_rolls)**2 # s^2
err = np.sqrt(var/Niter) # (s^2/N)
```

then we estimate the statistical error using

$$\varepsilon = \frac{\sigma}{\sqrt{N_{iter}}} \tag{6}$$

where the variance is the estimation of the statistical error. We obatin the results, which are reported in Table 1

N_{iter}	μ	σ^2	ε	$\mu(N_{iter}) - \mu_{exact}$
10	3.600	3.44	0.5865	0.1
100	3.640	2.55	0.1597	0.14
1000	3.498	2.96	0.0172	-0.002

Table 1: Results of error estimation

As we expected as the number of samples increases, the statistical error decreases and the mean value and the variance tends to the exact value. The acual error is not similar to the estimation of the statistical error.