



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря
Сікорського”

Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

Комп’ютерний практикум №1

Моделювання систем

Виконав студент групи ІІІ-15: Плугатирьов Д.В.		Перевірив:
		Стеценко І.В.
		Дата:
		Оцінка:

Київ 2025

Завдання

1. Згенерувати 10000 випадкових чисел трьома вказаними нижче способами. **45 балів.**

а. Згенерувати випадкове число за формулою $x_i = -\frac{1}{\lambda} \ln(\xi_i)$, де ξ_i – випадкове число, рівномірно розподілене в інтервалі $(0;1)$. Числа ξ_i можна створювати за допомогою вбудованого в мову програмування генератора випадкових чисел. Перевірити на відповідність експоненційному закону розподілу $F(x) = 1 - e^{-\lambda x}$. Перевірку зробити при різних значеннях λ .

б. Згенерувати випадкове число за формулами:

$$x_i = \sigma \mu_i + a$$
$$\mu_i = \sum_{i=1}^{12} \xi_i - 6,$$

де ξ_i - випадкове число, рівномірно розподілене в інтервалі $(0;1)$. Числа ξ_i можна створювати за допомогою вбудованого в мову програмування генератора випадкових чисел. Перевірити на відповідність нормальному закону розподілу:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-a)^2}{2\sigma^2}\right).$$

Перевірку зробити при різних значеннях a і σ .

с. Згенерувати випадкове число за формулою $z_{i+1} = az_i \pmod{c}$, $x_{i+1} = z_{i+1}/c$, де $a = 5^{13}$, $c = 2^{31}$. Перевірити на відповідність рівномірному закону розподілу в інтервалі $(0;1)$. Перевірку зробити при різних значеннях параметрів a і c .

2. Для кожного побудованого генератора випадкових чисел побудувати гістограму частот, знайти середнє і дисперсію цих випадкових чисел. По

виду гістограми частот визначити вид закону розподілу. **20 балів.**

3. Відповідність заданому закону розподілу перевірити за допомогою критерію згоди χ^2 . **30 балів**

4. Зробити висновки щодо запропонованих способів генерування випадкових величин. **5 балів**

Хід роботи

У цій роботі представлено аналіз трьох різних методів генерації випадкових чисел:

- Експоненційний розподіл,
- Нормальний розподіл,
- Рівномірний розподіл.

Кожен метод реалізований у мові програмування Python. Для перевірки відповідності теоретичним розподілам було використано статистичні тести, гістограми та критерій узгодженості хі-квадрат.

Результати

```

# Скільки стовпчиків поміститься на гістограмі
def sturges_bins_amount(items_amt):
    return math.ceil(math.log2(items_amt) + 1)

# Створення пари бінів (межі відрізків на гістограмі)
def split_bins(bins):
    return [(bins[idx - 1], bins[idx]) for idx in range(1, bins.size)]

def calculate_mean_std(items):
    mean = np.mean(items)
    std = np.std(items, ddof=1)
    return (mean, std)

# Chi2 з таблиці (заздалегідь визначена) (функція відсоткової точки - ppf)
def calculate_chi2_k(alpha, bins_amt, params_amt):
    return stats.chi2.ppf(1 - alpha, bins_amt - params_amt - 1)

# Chi2 розраховується за формулою
def calculate_chi2(items_amt, values, bins, compute_step, *args):
    bins_pairs = split_bins(bins)
    probabilities_expected = np.array(
        [compute_step(left, right, *args) for left, right in bins_pairs]
    )

    values_expected = items_amt * probabilities_expected
    chi2 = np.sum(np.square(values - values_expected) / values_expected)
    return chi2

def is_matching_chi2_law(chi2, chi2_k):
    return chi2 <= chi2_k

```

Рисунок 1 – Функції обчислення статистичних характеристик та критерію хі-квадрат (χ^2)

Експоненційний закон розподілу

Формулювання закону розподілу

```
lambda_value = 4  
xi = np.random.uniform(size=numbers_to_generate)  
exponential_law = -1 / lambda_value * np.log(xi)  
exponential_law
```

[156] ✓ 0.0s

```
... array([0.01621837, 0.10029877, 0.80938324, ..., 0.28842959, 0.24805342,  
        0.69039281])
```

Побудова графіку для перевірки закону

```
counts, bins = np.histogram(exponential_law, bins=exponential_bins)
```

```
<Axes: title={'center': 'Закон - Експоненціальний'}, xlabel='Значення', ylabel='Частота'>
```

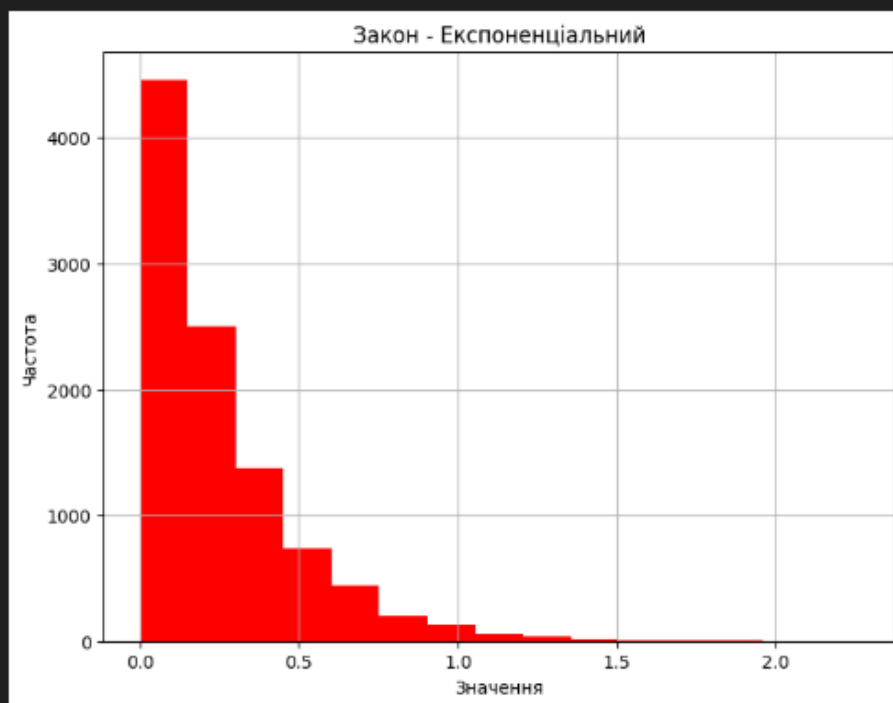


Рисунок 2 – Результати виконання для генерації та візуалізації експоненційного розподілу

Статистика за законом

```
exponential_stats = calculate_mean_std(exponential_law)
exponential_mean, exponential_std = exponential_stats
fmt = format_statistics_info(exponential_stats, 'Експоненціальний')
print(fmt)
```

[158] ✓ 0.0s

... Експоненціальний:
Середнє: 0.250, Std: 0.246

Перевірка за критерієм згоди χ^2

```
def calculate_exponential_step(left, right, lambda_value):
    return np.exp(-lambda_value * left) - np.exp(-lambda_value * right)

lambda_expected = 1 / exponential_mean
exponential_chi2 = calculate_chi2(numbers_to_generate, counts, bins,
                                  calculate_exponential_step, lambda_expected)
exponential_chi2_k = calculate_chi2_k(0.05, exponential_bins, 1)
fmt = format_chi_square_info(exponential_chi2, exponential_chi2_k, 'Експоненціальний')
print(fmt)
```

[159] ✓ 0.0s

... Експоненціальний:
Chi^2: 11.486, Chi^2 k: 22.362032494826934

```
if is_matching_chi2_law(exponential_chi2, exponential_chi2_k):
    print("Експоненціальний закон працює!")
else:
    print("Експоненціальний закон не працює!")
```

[160] ✓ 0.0s

... Експоненціальний закон працює!

Рисунок 3 – Аналіз експоненційного розподілу та перевірка відповідності

критерію χ^2 -квадрат (χ^2)

Нормальний закон розподілу

Формулювання закону розподілу

```
sigma_value = 4
a_value = -5

normal_law = np.zeros(numbers_to_generate)
for idx in range(numbers_to_generate):
    mu_value = np.random.uniform(size=12).sum() - 6
    normal_law[idx] = sigma_value * mu_value + a_value
normal_law
```

[162] ✓ 0.0s

```
... array([ -4.9297405 , -11.88015949, -5.69515272, ..., -6.39305029,
          -2.67458217, -0.3684195 ])
```

Побудова графіку для перевірки закону

```
normal_counts, normal_bins = np.histogram(normal_law, bins=normal_init_bins)
build_histogram(normal_law, normal_bins, title='Закон - Нормальний розподіл', color='black')
```

[163] ✓ 0.1s

```
... <Axes: title={'center': 'Закон - Нормальний розподіл'}, xlabel='Значення', ylabel='Частота'>
```

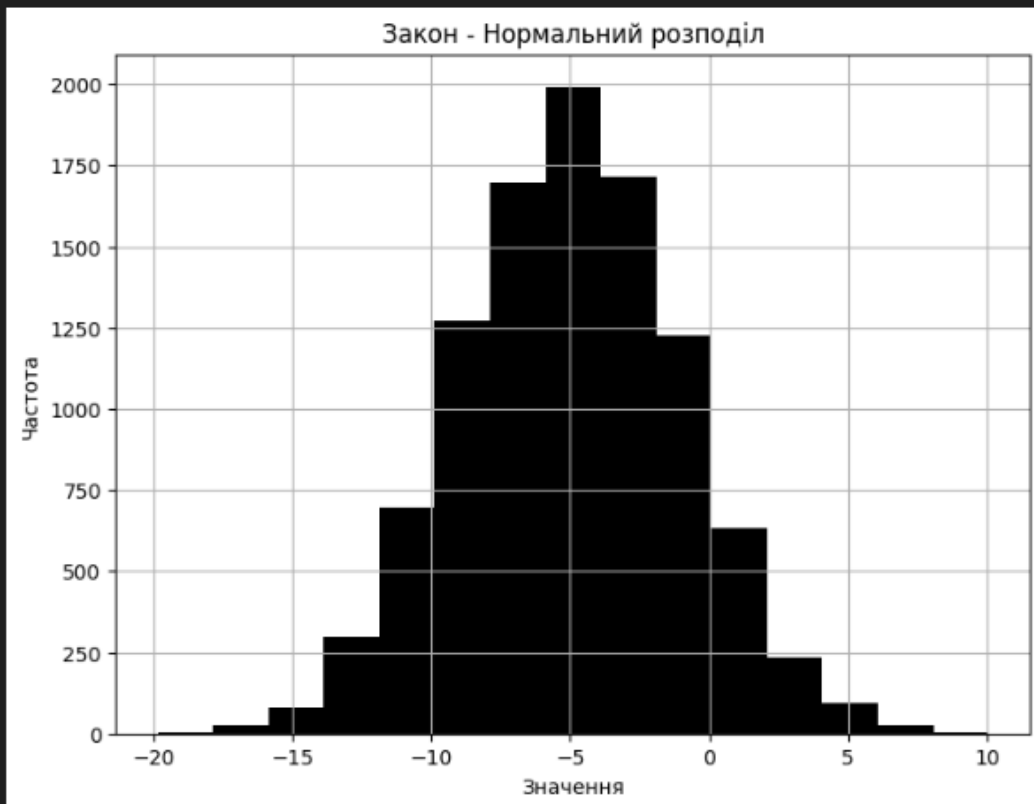


Рисунок 4 – Генерація нормального розподілу та його візуалізація

Статистика за законом

```
normal_stats = calculate_mean_std(normal_law)
normal_mean, normal_std = normal_stats
fmt = format_statistics_info(normal_stats, 'Нормальний розподіл')
print(fmt)
```

[164] ✓ 0.0s

... Нормальний розподіл:
Середнє: -4.994, Std: 3.978

Перевірка за критерієм згоди χ^2

```
# Кумулятивна функція розподілу - cdf
def calculate_normal_step(left, right, sigma_value, a_value):
    left_cdf, right_cdf = stats.norm.cdf([left, right], loc=sigma_value, scale=a_value)
    return right_cdf - left_cdf

sigma_expected = normal_mean
normal_expected = normal_std
normal_chi2 = calculate_chi2(numbers_to_generate, normal_counts, normal_bins,
                             calculate_normal_step, sigma_expected, normal_expected)
normal_chi2_k = calculate_chi2_k(0.05, normal_init_bins, 2)
fmt = format_chi_square_info(normal_chi2, normal_chi2_k, 'Нормальний розподіл')
print(fmt)
```

[165] ✓ 0.0s

... Нормальний розподіл:
Chi^2: 14.754, Chi^2 k: 21.02606981748307

```
if is_matching_chi2_law(normal_chi2, normal_chi2_k):
    print("Закон нормального розподілу працює!")
else:
    print("Закон нормального розподілу не працює!")
```

[166] ✓ 0.0s

... Закон нормального розподілу працює!

Рисунок 5 – Аналіз нормального розподілу та перевірка його відповідності критерію χ^2 Рівномірний закон розподілу

Формуляція закону розподілу

```
a_value = 5**13
c_value = 2**31
zi_value = np.random.uniform()

uniform_law = np.zeros(numbers_to_generate)
for idx in range(numbers_to_generate):
    zi_value = (a_value * zi_value) % c_value
    uniform_law[idx] = zi_value / c_value
uniform_law
```

[168] ✓ 0.0s

... array([0.07858199, 0.1854437 , 0.32018286, ..., 0.82644033, 0.7305522 ,
0.24644232])

Побудова графіку для перевірки закону

```
uniform_counts, uniform_bins = np.histogram(uniform_law, bins=uniform_init_bins)
build_histogram(uniform_law, uniform_bins, title='Закон - Рівномірний розподіл', color='blue')
```

[169] ✓ 0.1s

... <Axes: title={'center': 'Закон - Рівномірний розподіл'}, xlabel='Значення', ylabel='Частота'>

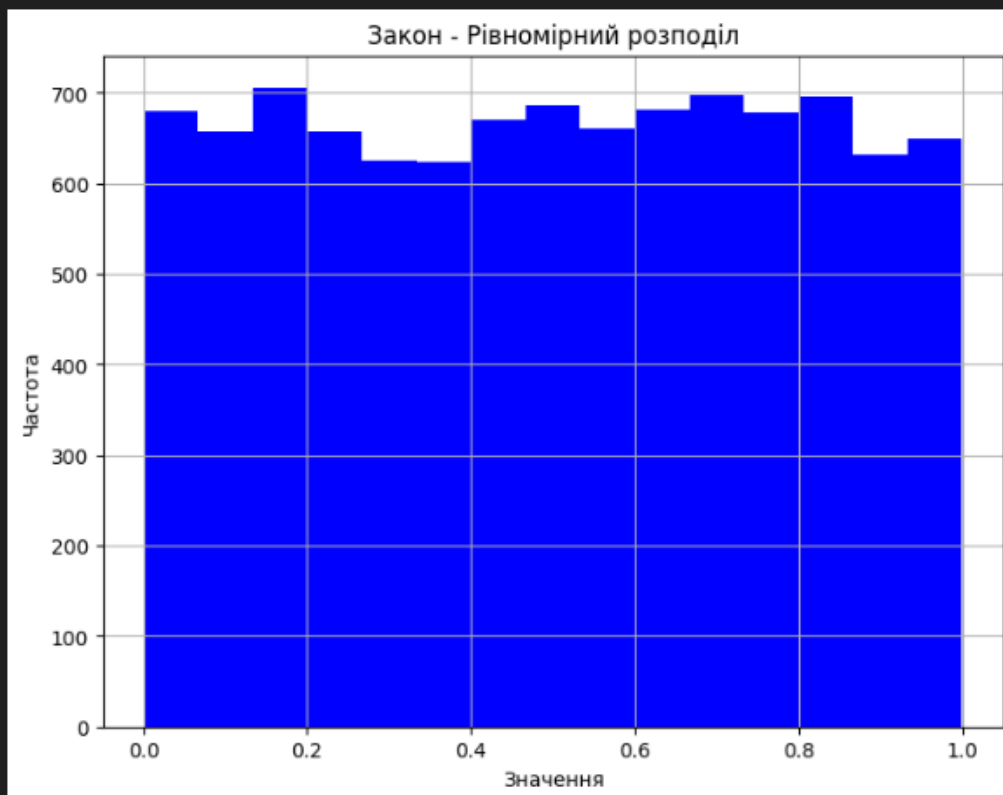


Рисунок 6 – Генерація рівномірного розподілу та його візуалізація

Статистика за законом

```
uniform_stats = calculate_mean_std(uniform_law)
uniform_mean, uniform_std = uniform_stats
fmt = format_statistics_info(uniform_stats, 'Рівномірний розподіл')
print(fmt)
```

[170] ✓ 0.0s

... Рівномірний розподіл:
Середнє: 0.500, Std: 0.289

Перевірка за критерієм згоди χ^2

```
def calculate_uniform_step(left, right, a_value, c_value):
    return (right - left) / (c_value - a_value)

a_expected = uniform_mean - 3 ** 0.5 * uniform_std
c_expected = uniform_mean + 3 ** 0.5 * uniform_std
uniform_chi2 = calculate_chi2(numbers_to_generate, uniform_counts, uniform_bins,
                             calculate_uniform_step, a_expected, c_expected)
uniform_chi2_k = calculate_chi2_k(0.05, uniform_init_bins, 2)
fmt = format_chi_square_info(uniform_chi2, uniform_chi2_k, 'Рівномірний розподіл')
print(fmt)
```

[171] ✓ 0.0s

... Рівномірний розподіл:
Chi^2: 14.552, Chi^2 k: 21.02606981748307

```
if is_matching_chi2_law(uniform_chi2, uniform_chi2_k):
    print("Закон рівномірного розподілу працює!")
else:
    print("Закон рівномірного розподілу не працює!")
```

[172] ✓ 0.0s

... Закон рівномірного розподілу працює!

Рисунок 7 – Перевірка рівномірного розподілу та його відповідності
критерію χ^2 -квадрат (χ^2)

Посилання на репозиторій з кодом: <https://github.com/I-delver-I/system-modelling>.

Висновок

У ході виконання комп'ютерного практикуму було розглянуто основні закони розподілу випадкових величин, їхню генерацію, статистичний аналіз та перевірку відповідності теоретичним очікуванням. Реалізовано алгоритми для експоненційного, нормального та рівномірного розподілів, проведено розрахунок їхніх основних характеристик і виконано перевірку критерієм χ^2 -квадрат.

Отримані результати продемонстрували, що згенеровані вибірки узгоджуються з відповідними математичними моделями. Обчислені середні значення та стандартні відхилення для кожного закону розподілу виявилися близькими до теоретичних, а значення критерію χ^2 -квадрат підтвердили відповідність емпіричних даних очікуваному розподілу.

Завдяки проведеному аналізу вдалося підтвердити правильність реалізованих алгоритмів та їхню здатність коректно відтворювати випадкові величини відповідно до заданих законів розподілу. Виконана робота дозволила закріпити навички програмної реалізації методів статистичного аналізу та оцінки якості випадкових вибірок.