# Звіт до лабораторної 1

Студента групи ТТП-42

Остренка Олександра

### Підкидання монети.

#### Умова

Изображение выглядит как текст, Шрифт, линия, снимок экрана

Автоматически созданное описание

#### Реалізація

Изображение выглядит как текст, снимок экрана, программное обеспечение, Операционная система

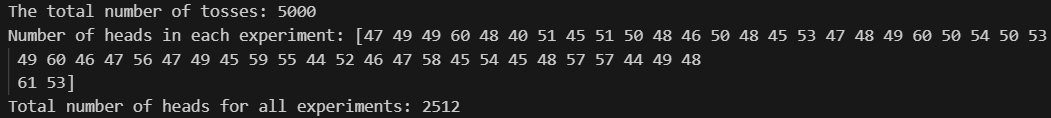
Автоматически созданное описание

За допомогою бібліотеки numpy можна з легкістю змоделювати підкидання монети, бо функція randint генерує цілу випадкову величину з рівномірного розподілу для проміжку [0;2). Змінна results являє собою матрицю K\*N, де один рядок – результати підкидань одного експеременту.

#### Тестування

Изображение выглядит как текст, Шрифт, снимок экрана, типография

Автоматически созданное описание



Изображение выглядит как снимок экрана, текст, шаблон, черно-белый

Автоматически созданное описание

Т.я. використовується функція, що моделює випадкові величини з рівномірного розподілу, то, в теорії, кількості появи герба та решки мають прямувати до одного і того самого числа. Можна спостерігати, що число появи герба і справді прямує до K\*N/2 при збільшені кількості підкидань, але на малих вибірках ми не отримаємо рівну кількість того і того через елемент випадковості при моделюванні.

### 1.2. Підкидання кубику.

#### Умова

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

#### Реалізація

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

За допомогою бібліотеки numpy можна з легкістю змоделювати підкидання кубику, бо функція randint генерує цілу випадкову величину з рівномірного розподілу для проміжку [1;7). Змінна results являє собою матрицю K\*N, де один рядок – результати підкидань одного експеременту.

#### Тестування

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Як і з підкиданням монети, в середньому, кожна цифра має випадати однакову кількість разів. В даному output кількість випадінь має прямувати до 18, але через випадкову складову моделювання кількості різняться.

### 2.2. Підкидання двох кубиків.

#### Умова

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

#### Реалізація

Для моделювання суми очок при підкиданні двох кубиків достатньо по прикладу задачі 1.2 змоделювати два рази один кубик і повернути суму очок.

Изображение выглядит как текст, Шрифт, снимок экрана

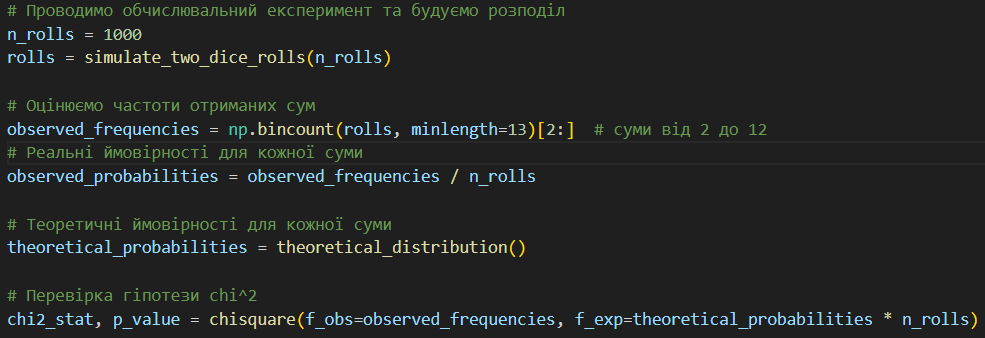
Автоматически созданное описание

Теоретичний розподіл знаходиться простим перерахуванням кількості комбінацій очок для конкретної суми.

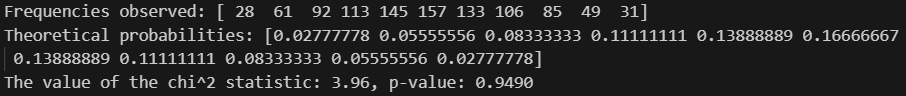
Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Далі проводимо обчислюваний експеримент, де отримуємо частоти випадання різних сум та обчислюємо на основі цих частот відповідні ймовірності. Отримуємо теоретичні ймовірності до кожної суми. За допомогою функції «chisquare» з бібліотеки «scipy.stats» обчислюємо значення статистики chi^2, а також відповідне p-value для перевірки відповідності отриманих (експериментально) частот з теоретичним розподілом. В даному випадку теоретичний розподіл задається порахованими частотами випадінь відповідних сум.



#### Тестування



Таке високе p-value (явно більше за прийняте 0.05) говорить про те, що немає підстав відкидати гіпотезу у тому, що експериментальні дані слідують теоретичному розподілу.

Для наочності я також відобразив теоретичні та емпіричні дані у вигляді стовпчастої діаграми.

Изображение выглядит как текст, снимок экрана, График, диаграмма

Автоматически созданное описание

З неї добре видно, що експериментальні ймовірності ідуть бік о бік з теоретичними, а сума в 7 очок буде випадати в більшій кількості підкидань.

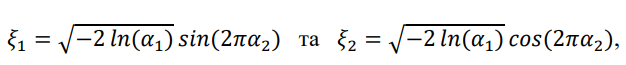
### 3.4. Моделювання випадкових величин з нормального розподілу.

#### Умова

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

#### Реалізація

Для функціонального моделювання було використано метод оберненої функції, а саме - перетворення Бокса-Мюллера, в якому пара випадкових величин генеруються за наступною формулою:  


Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Далі моделюємо випадкові величини з використанням бібліотечної функції random.gauss().

Изображение выглядит как текст, Шрифт, снимок экрана, линия

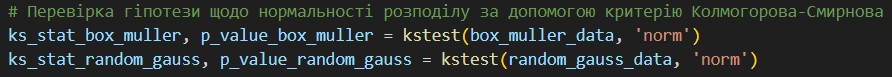
Автоматически созданное описание

Після отримання 1000 випадкових величин обома підходами, оцінюємо математичне сподівання та дисперсію отриманих вибірок.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

За допомогою критерію Колмогорова-Смирнова перевіряємо гіпотезу про нормальність отриманих вибірок.



#### Тестування

Изображение выглядит как текст, Шрифт, снимок экрана, черный

Автоматически созданное описание

Можна побачити, що обидва методи дають дуже хороші результати – мат. сподівання прямує до нуля, дисперсія до одиниці, а p-value набагато перевищує 0.05, що означає відсутність підстав відкидати гіпотезу про нормальність обох вибірок.

Візуалізована гістограма наглядно показує, що отримані вибірки дотримуються нормального розподілу

Изображение выглядит как диаграмма, снимок экрана, График, текст

Автоматически созданное описание

### 4.1.1. Обчислити інтеграл методом Монте-Карло

#### Умова

