Міністерство освіти та науки України

Карпатський національний університет імені Василя Стефаника

Семанькiв М.В.

МЕТОДИЧНІ ВКАЗІВКИ ДО ЛАБОРАТОРНИХ РОБІТ З ДИСЦИПЛІНИ “PYTHON ДЛЯ DATA SCIENCE”

Івано-Франківськ, 2025

УДК 004.43
ББК 32.973-018.1

Рекомендовано до друку Вченою радою факультету математики та інформатики Карпатського національного університету імені Василя Стефаника
(*протокол № 9 від 22.10.2025р.*).

Рецензенти :

Горєлов В.О., кандидат технічних наук, доцент (Карпатський національний університет імені Василя Стефаника)

Іляш Ю.Ю., кандидат технічних наук, доцент (Карпатський національний університет імені Василя Стефаника)

Семаньків М.В. Методичні вказівки до лабораторних робіт з дисципліни “Python для Data Science”. – Івано-Франківськ: Карпатський національний університет імені Василя Стефаника, 2025. – 54 с.

Наведено завдання до лабораторних робіт з мови програмування Python та використання бібліотек NumPy, Pandas, Matplotlib, Seaborn. Призначено для проведення лабораторних занять з курсу “Python для Data Science”.
Для студентів спеціальностей Комп’ютерні науки та Інформаційні системи та технології.

ЗМІСТ:

ВСТУП

Мета курсу “Python для Data Science” — сформувати у студентів цілісне розуміння принципів аналізу даних, опанування інструментів та методів обробки, візуалізації й моделювання інформації за допомогою мови програмування Python.

Курс спрямований на розвиток практичних навичок роботи з бібліотеками NumPy, Pandas, Matplotlib, Seaborn та іншими засобами аналітики, що дозволяють ефективно виконувати підготовку даних, статистичний аналіз, побудову моделей машинного навчання й представлення результатів у зручній формі.

Курс має на меті навчити студентів застосовувати здобуті знання для розв’язання прикладних задач у сфері штучного інтелекту, бізнес-аналітики, фінансового прогнозування та наукових досліджень, розвиваючи аналітичне мислення, алгоритмічну культуру та здатність працювати з великими обсягами даних.

Лабораторна робота №1
NumPy

1. Імпортувати пакет NumPy під псевдонімом np	import numpy as np
Порівняння array vs list: - array використовує менше пам'яті ніж списки - array має значно більший функціонал - array вимагає щоб дані були однорідними, а списки ні	
2. Створити звичайний список з п'яти чисел 3. Перезаписати його через масив даного пакету	a= b=np.array(a)
4. Перевірити тип для змінних a і b	type()
5. Помножити список і масив на 4 та вивести	
6. Піднести масив до другого степеня, спробувати це зробити зі списком	
7. До масиву додати число, спробувати до списку	
8. Знайти остачі від ділення на 2 всіх елементів масиву і списку	
9. Створити ще один масив з 5 елементами	
10. Порівняти масиви **Питання: Чи можуть бути масиви з різною кількістю змінних в порівнянні?**	b==c b<c b>c
11. Додати два масиви **Питання: Що буде результатом додавання двох списків? Чи кількість елементів повинна співпадати?**	
12. Здійснити **фільтрацію**: задати маску для знаходження чисел більше 10 13. Вивести всі числа згідно даної маски	c>10 c[c>10]
14. Використовуючи маску перевірити чи **всі** значення з масиву *c* більші 5	_.all(__)
15. Чи **хоча б одне** значення в масиві *c* більше 10	_.any(__)
Операції: • min() / max() мінімальний/максимальний елемент • argmin(), argmax() індекс мінімального, максимального елементу • mean() середнє значення • sum() сума елементів • prod() добуток елементів • tolist() перетворення масиву у список • len() повертає довжину першого виміру (осі) • dtype - атрибут для виведення типу даних масиву	
16. Задати новий масив **Питання: як записати ці команди для списку?**	
17. На якій позиції знаходиться максимальний елемент? Мінімальний? **Питання: якщо максимальний елемент повторюється в масиві декілька раз, що виведе argmax**	

18. Якого типу дані в масиві b	
Питання: якого типу дані в масивах **b=np.array([5, 7, 0, 9.5, 3])** **b=np.array([5, 7, '0', 9.5, 3])** **b=np.array([5, 7, 0, True, 3])** **b=np.array([5, 7, '0', True, 3])** **b=np.array([5, 7, 0, None, 3])** **b=np.array([5, 7, 0, np.nan, 3])**	
Найбільш важливі атрибути об'єктів ndarray: ndarray.**ndim** - число вимірювань (частіше їх називають "осі") масиву. ndarray.**shape** - розміри масиву, його форма. Це кортеж натуральних чисел, що показує довжину масиву по кожній осі ndarray.**size** - кількість елементів масиву. Очевидно, дорівнює добутку всіх елементів атрибута shape.	
19. Визначити атрибути масивів ndim (розмірність), shape (розмір кожного виміру) , size (загальний розмір масиву)	
Питання: яка розмірність, розмір кожного виміру та загальний розмір для d=np.array([[2, 4], [1, 7], [5, 9]])	
Створення масивів • Явно m=np.array([9,8,7,6,5,4]). • Функція arange (start, stop, step, dtype). Аналогічна вбудованій в Python range(), тільки замість списків вона повертає масиви заданого типу. • Функція linspace(start, stop, num) num – кількість елементів. • Функція zeros(shape, dtype) створює масив із нулів. • Функція ones(shape, dtype) — масив із одиниць. • Функція eye() створює одиничну матрицю (двовимірний масив). • Функція empty() створює масив без його заповнення.	
Приклади:	
Зі списку Python *Якщо типи елементів у списку не співпадають, то NumPy спробує привести дані до одного типу*	np.array([4,6,8,0,9,5])
Задання типу даних dtype	np.array([1, 2, 3, 4], dtype='float32')
Створення масиву чисел, заповнених нулями	np.zeros(10, dtype=int)
Створити масив з одиниць	np.ones(3)
Створити двовимірний масив з нулів з заданим типом даних *Для створення двовимірних масивів необхідно передати кортеж: перший елемент - кількість рядків, другий- кількість стовпців*	np.ones((3, 5), dtype=float)
Двовимірний масив запоненений числом 100	np.full((2, 2), 100)
Створити список за допомогою функції arange	np.arange(10, 101, 10)
Створення масиву лінійнорозподіленихелементів із задного проміжку - функція linspace	np.linspace(10, 100, num=15)
Створення масиву випадкових чисел за допомогою функції rand() модуля random.	np.random.rand(4)

Створення двовимірного масиву випадкових чисел	np.random.random((3, 3))
Створення масиву випадкових чисел 3 х 3 на проміжку [0, 10)	np.random.randint(0, 10, (3, 3))
Створення діагональної матриці	np.eye(3)
Індекси масивів	
Створити двовимірний список	m=[[1,2],[3,4]]
Зробити з нього масив	n=np.array(m)
Вивести останній елемент списку та масиву	n[1][1] або n[1,1]
Змінювати значення масивів можна використовуючи індекси	x2[0, 0] = 12
20. Створити два одновимірні масиви та об'єднати їх в один об'єднання масивів	np.concatenate(())
21. Розділити масив пополам розбиття масивів x = np.array([1, 2, 3, 99, 99, 3, 2, 1]) x1, x2, x3 = np.split(x, [3, 5])	
Модифікація масивів • np.**append**(a, values)- додає елементи в масив а • np.**insert**(a, ndx, values)- вставляє елементи в масив фзаданою значенням індексу ndx • np.**delete**(a, ndx)- видаляє елементи з масиву а заданою значенням індексу ndx !!!!! Всі три функції створюють новий масив, як модифіковану копію оригіналу	
22. До масиву додати [1, 2, 3]	
23. Знищити перший елемент масиву	

Лінійна алгебра: Вектори та операції над ними

Задання векторів	Вектор в NumPy є одновимірним масивом, що відповідає інтуїтивному визначенню вектора
Довжина вектора, норма, модуль вектора	from numpy.linalg import norm norm(a)
Скалярний добуток	np.dot(_,_)

Завдання для виконання:

Частина 1.

1) Дано температуру за місяць: [4,1,0,-2,5,7,-3,0,0,5,0,-1,3,6,9,5,7,11,9,6,0]	
2) Яка в середньому температура були цього місяця	
3) Яка була найвища температура	
4) В який день (порядковий номер) було найтепліше	
5) Замінити максимальну температуру на 0	
6) Порахувати кількість днів, коли температура була менша нуля	
7) Чи перевищувала температура 10? 5?	

8) Чи всі дні температура була менша 10?	
9) Чи були випадки коли зміна температури між двома сусідніми днями була більша 5? скільки раз це було	
10) Збільшити всі дані температури на 1	
11) Для скількох днів показано дані	
12) Згенерувати масив даних з чисел проміжку від -5 до 5, кількість: скільки днів не вистачає до попереднього масиву, щоб було 31 день (тобто якщо у попередньому масиві 10 чисел, то згенерувати випадкове 21 число)	
13) Об'єднати обидва масиви в один	

Частина 2.

Знайти в numpy відповіді на тести та завдання контрольної роботи школяра

1.Дано $\vec{a}$ (-5;3), $\vec{b}$(2;4). Знайти координати вектора $\vec{c} = \vec{a} + \vec{b}$.
А) (7;1); Б) (-3;7); В) (-1;1); Г) (-7;-1).

2.Дано $\vec{a}$ (3;-6), $\vec{b}$(1;-2). Знайти координати вектора $\vec{c} = \vec{b} - \vec{a}$.
А) (2;-4); Б) (-2;4); В) (4;-8); Г) (-2;-8).

3.Дано $\vec{a}$ (-3;2), $\vec{b}$(1;-4). Чому дорівнює їх скалярний добуток?
А) 5; Б) -10; В) -11; Г) -7.

4.Знайдіть модуль вектора $\overrightarrow{AB}$, якщо А(4;-3), В(0;-6).
А) 5; Б) $\sqrt{97}$; В) 12; Г) $\sqrt{19}$.

5. Дано $\vec{a}$ (-4;8), $\vec{b}$(3;7). Знайти координати вектора $\vec{c}$ =2 $\vec{a}$ - $\vec{b}$.
А) (-11;9); Б) (-10;-6); В) (-5;9); Г) (-7;1).

6. Знайти косинус кут між векторами a(0; 2; -2) і b(1; 0; -1).

7. Чи вектори перпендикулярні (3,-3,3), (3, 5,2)

8. Чи вектори колінеарні (3,-1,2), (-6, 2,-4)

Лабораторна робота №2
Двовимірні масиви в Numpy

Створення двовимірних масивів

1. Зі списку Python [[1,2,3],[4,5,6],[7,8,9]]	
2. Створення масиву 3 на 4, заповненого нулями	
3. Створити масив 3 на 4 з одиниць типу float	
4. Створення масиву випадкових цілих чисел 3 x 3 на проміжку [0, 10)	

Перехід від одновимірного до двовимірного

5. Задати одновимірний масив чисел з проміжку від 0 до 12	
6. Зробити з нього двовимірний 3 на 4 Метод reshape(). У цей метод потрібно просто передати нові розмірності матриці. Якщо вказати розмірність -1, NumPy на основі вашої матриці зможе визначити правильну розмірність **Питання: Що отримаємо в результаті** `arr = np.array([[1, 2, 3], [4, 5, 6]])` `newarr = arr.reshape(-1)`	____.reshape(__,___)

Арифметичні операції над матрицями

Якщо дві матриці мають однакову розмірність, над нами можна виконувати арифметичні операторії (+ - * /). NumPy обробляє такі дії як позиційні операції:

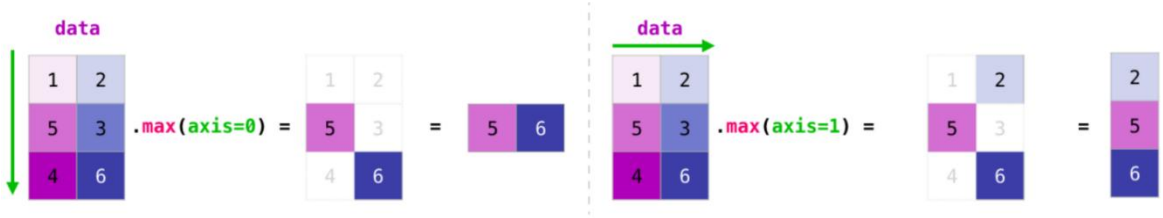
7. Задати два двовимірні масиви. Додати їх, відняти	
Зауваження: При заданні операції множення - множаться елементи по позиціях, але це не добуток матриць згідно математичного терміну	
Питання: Що буде в результаті множення m=np.array([[1,2],[3,4], [5,6]]) і h=np.array([1,0])	
Зауваження: *Якщо матриці мають різну розмірність, арифметичні операції до них можна застосовувати, тільки якщо розмірність однієї з матриць дорівнює одиниці (наприклад, матриця має лише один стовпець або один рядок)*	
8. **Множення матриць**: задати матриці 2 на 3, та 3 на 2, знайти добуток матриць	m.dot(n)

Базові операції над масивами

9. Знайти суму та добуток елементів масиву	sum(), prod()
Деякі функції надають можливість оперувати статистичними даними. Це такі фу як середнє арифметичне (*mean*), дисперсія (*var*) і стандартне відхилення (*std*).	
10. Враховувати комірки Not a Number (NaN) в масиві як нуль при обчисленні суми та добутку	np.nansum(a)

Агрегування матриць

Агрегатори - це методи NumPy дозволяють замінювати дані інтегральними характеристиками вздовж деяких осей. Форма нового масиву буде містити всі осі вихідного масиву, крім тих, уздовж яких підраховується агрегатор.



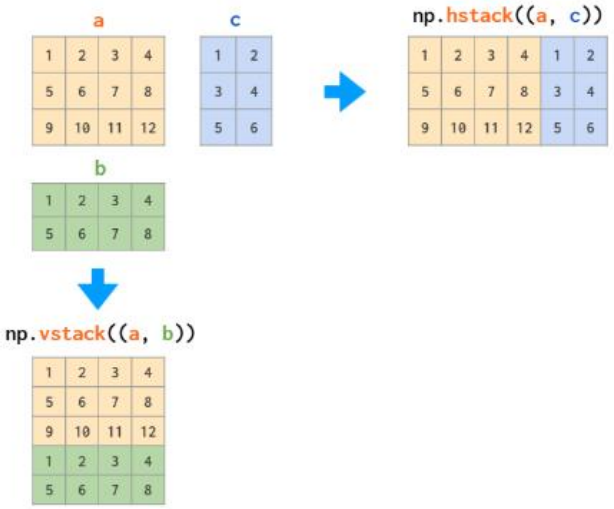
11. Задати двовимірний масив. Знайти мінімальні значення в стовпцях	
12. Знайти суму значень в рядках	
13. Знайти середнє значення по всій таблиці	

Приєднання масивів

14. Задати масив[[5,5],[6,6]]	

15. Додати масив [[1, 2,3], [4,5,6]] через hstack() горизонтально
Питання: що буде у випадку одновимірних масивів
`f = np.array([1,2,3])`
`g = np.array([4,5,6])`
	v

14 Додати масив [[0, 1], [2,3], [4,5]] vstack() через вертикально	



Транспонування і зміна форми матриць

16. Для транспонування матриці в масивах NumPy передбачено властивість T	a.T
17. З двовимірного масиву створити одновимірний **Питання: яка різниця між resize reshape**	ravel()

Додаткові команди

18. Знайти **унікальні** елементи масиву	np.unique(a)
19. Фільтрації індексів даних, що задовольняють певну вимогу. Вивести на яких позиціях числа більші нуля	np.where(a>0)

Матриці

1 спосіб Створення матриці як двовимірного масиву 1. Створити матрицю зі своїми даними	a = np.array([[0, 1, 2], [4,5,6]])
2. Транспонування матриці $T\left(\begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}\right) = \begin{bmatrix} a_1 & a_3 \\ a_2 & a_4 \end{bmatrix}$	З допомогою методу transpose(): A_t = A.transpose() *Другий спосіб: (A.T)*
3. Перевірити, що $(A^T)^T = A.$	
4. Знайти визначник матриці	np.linalg.det(A)
5. Обчислити обернену матрицю	np.linalg.inv(A)
6. Ранг матиці	linalg.matrix_rank( )
7. Розв'язати систему лінійних рівнянь 1*x1 + 5*x2 = 11	Задати двовимірний масив a з коефіцієнтами системи, та

2*x1 + 3*x2 = 8	одновимірний масив b з вільними членами системи a = np.array([[1, 5], [2,3]]) b= np.array([11,8]) x = np.linalg.solve(a,b) Перевірка np.dot(a,x)
8. Власні значення та власні вектори	w, v=np.linalg.eig( )

9. Порівняння матриць
`arr1 = np.array([0.15, 0.20, 0.25, 0.17])`
`arr2 = np.array([0.14, 0.21, 0.27, 0.15])`
	`np.allclose(arr1, arr2, 0.1)`
	`False`
	`np.allclose(arr1, arr2, 0.2)`
	`True`

ЗАВДАННЯ ДЛЯ ВИКОНАННЯ:

Частина 1. Виконати завдання:

1. Дана виручка за місяць з продажу товару в магазині [739, 642, 125, 984, 490, 712, 635, 643, 189, 370, 752, 397, 384, 756, 171, 153, 155, 695, 831, 618, 898, 115, 624, 592, 684, 799, 461, 96]
2. Задати подання в вигляді тижнів: перетворити у масив по 7 елементів (в кожному рядку виручка за 7 днів тижня). Задати, щоб кількість рядків обчислювалась
3. Знайти загальну виручку за дні роботи магазину
4. Знайти середню арифметичну виручку кожного тижня місяця
5. Для днів тижня (стовпців) знайти загальні суми виручки та вказати в який робочий день тижня (число) були найкращі продажі
6. В які дні тижня (стовпці) виручка перевищила 2500
7. Чи правдиве твердження, що виручка за вихідні дні місяця перевищує виручку за всі інші дні?
8. Всі значення в масиві за вихідні дні обнулити
9. Зменшити двовимірний масив відкинувши перший день тижня (перший стовпчик)
10. Надійшли нові дані за продаж в понеділок [205, 190, np.NaN, 117]. Утворити з них масив, задати форму 1 стовпчик та 4 рядки. Приєднати зліва до початкового масиву з виручками, щоб вони стали значеннями першого стовпця
11. Яка виручка за місяць тепер?
12. Транспонувати масив

Частина 2. Розв'язати систему алгебраїчних рівнянь згідно **СВОГО** варіанту

1) За допомогою функції numpy.linalg.solve();

2) За допомогою оберненої матриці; $X = A^{-1} \cdot B.$

де *A* – матриця коефіцієнтів системи рівнянь,

B – матриця – стовпець вільних членів рівнянь

3) за допомогою методу Крамера;

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \end{cases}$$

Определители:

$$\Delta = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}, \quad \Delta_1 = \begin{vmatrix} b_1 & a_{12} & a_{13} \\ b_2 & a_{22} & a_{23} \\ b_3 & a_{32} & a_{33} \end{vmatrix},$$

$$\Delta_2 = \begin{vmatrix} a_{11} & b_1 & a_{13} \\ a_{21} & b_2 & a_{23} \\ a_{31} & b_3 & a_{33} \end{vmatrix}, \quad \Delta_3 = \begin{vmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ a_{31} & a_{32} & b_3 \end{vmatrix}$$

Решение:

$$x_1 = \frac{\Delta_1}{\Delta}, \quad x_2 = \frac{\Delta_2}{\Delta}, \quad x_3 = \frac{\Delta_3}{\Delta}$$

4) Порівняти всі рішення за допомогою функції numpy.allclose().

Частина 3. Обчислити значення МАТРИЧНОГО виразу згідно свого варіанту

Варіанти:

Варіант 1	2) $\begin{cases} x_1 + x_2 + 2x_3 + 3x_4 = 1 \\ 3x_1 - x_2 - x_3 - 2x_4 = -4 \\ 2x_1 - 3x_2 - x_3 - x_4 = -6 \\ x_1 + 2x_2 + 3x_3 - x_4 = -4 \end{cases}$ 3) $2(A+B)(2B-A)$, де $A = \begin{pmatrix} 2 & 3 & -1 \\ 4 & 5 & 2 \\ -1 & 0 & 7 \end{pmatrix}$, $B = \begin{pmatrix} -1 & 0 & 5 \\ 0 & 1 & 3 \\ 2 & -2 & 4 \end{pmatrix}$	
Варіант 2	2) $\begin{cases} x_1 + 2x_2 + 3x_3 - 2x_4 = 6 \\ x_1 - x_2 - 2x_3 - 3x_4 = 8 \\ 3x_1 + 2x_2 - x_3 + 2x_4 = 4 \\ 2x_1 - 3x_2 + 2x_3 + x_4 = -8 \end{cases}$ 3) $3A - (A - 2B)B$, де $A = \begin{pmatrix} 4 & 5 & -2 \\ 3 & -1 & 0 \\ 4 & 2 & 7 \end{pmatrix}$, $B = \begin{pmatrix} 2 & 1 & -1 \\ 0 & 1 & 3 \\ 5 & 7 & 3 \end{pmatrix}$	

Варіант 3	2) $\begin{cases} x_1+2x_2+3x_3+4x_4=5 \\ 2x_1+x_2+2x_3+3x_4=1 \\ 3x_1+2x_2+x_3+2x_4=1 \\ 4x_1+3x_2+2x_3+x_4=-5 \end{cases}$ 3) $2(A-B)(A^2+B)$, де $A=\begin{pmatrix} 5 & 1 & 7 \\ -10 & -2 & 1 \\ 0 & 1 & 2 \end{pmatrix}$, $B\begin{pmatrix} 2 & 4 & 1 \\ 3 & 1 & 0 \\ 7 & 2 & 1 \end{pmatrix}$
Варіант 4	2) $\begin{cases} x_2-3x_3+4x_4=-5 \\ x_1-2x_3+3x_4=-4 \\ 3x_1+2x_2-5x_4=12 \\ 4x_1+3x_2-5x_3=5 \end{cases}$ 3) $\left(A^2-B^2\right)\cdot(A+B)$, де $A=\begin{pmatrix} 7 & 2 & 0 \\ -7 & -2 & 1 \\ 1 & 1 & 0 \end{pmatrix}$, $B=\begin{pmatrix} 0 & 2 & 3 \\ 1 & 0 & -2 \\ 3 & 1 & 1 \end{pmatrix}$
Варіант 5	2) $\begin{cases} x_1+3x_2+5x_3+7x_4=12 \\ 3x_1+5x_2+7x_3+x_4=0 \\ 5x_1+7x_2+x_3+3x_4=4 \\ 7x_1+x_2+3x_3+5x_4=16 \end{cases}$ 3) $\left(A-B^2\right)(2A+B)$, де $A=\begin{pmatrix} 5 & 2 & 0 \\ 10 & 4 & 1 \\ 7 & 3 & 2 \end{pmatrix}$, $B=\begin{pmatrix} 3 & 6 & -1 \\ -1 & -2 & 0 \\ 2 & 1 & 3 \end{pmatrix}$
Варіант 6	2) $\begin{cases} x_1+5x_2+3x_3-4x_4=20 \\ 3x_1+x_2-2x_3=9 \\ 5x_1-7x_2+10x_4=-9 \\ 3x_2-5x_3=1 \end{cases}$ 3) $\left(A-B^2\right)\cdot(2A+B)$, де $A=\begin{pmatrix} 5 & 2 & 0 \\ 10 & 4 & 1 \\ 7 & 3 & 2 \end{pmatrix}$, $B=\begin{pmatrix} 3 & 6 & -1 \\ -1 & -2 & 0 \\ 2 & 1 & 3 \end{pmatrix}$

Варіант 7	2) $\begin{cases} 2x_1 + x_2 - 5x_3 + x_4 = 8 \\ x_1 - 3x_2 - 6x_4 = 9 \\ 2x_2 - x_3 + 2x_4 = -5 \\ x_1 + 4x_2 - 7x_3 + 6x_4 = 0 \end{cases}$ 3) $2(A - 0{,}5B) + AB$, де $A = \begin{pmatrix} 5 & 3 & -1 \\ 2 & -2 & 0 \\ 3 & -1 & 2 \end{pmatrix}$, $B = \begin{pmatrix} 1 & 4 & 16 \\ -3 & -2 & 0 \\ 5 & 7 & 2 \end{pmatrix}$
Варіант 8	2) $\begin{cases} 2x_1 - x_2 + 3x_3 + 2x_4 = 4 \\ 3x_1 + 3x_2 + 3x_3 + 2x_4 = 6 \\ 3x_1 - x_2 - x_3 + 2x_4 = 6 \\ 3x_1 - x_2 + 3x_3 - x_4 = 6 \end{cases}$ 3) $(A - B)A + 3B$, де $A = \begin{pmatrix} 3 & 2 & -5 \\ 4 & 2 & 0 \\ 1 & 1 & 2 \end{pmatrix}$, $B = \begin{pmatrix} -1 & 2 & 4 \\ 0 & 3 & 2 \\ -1 & -3 & 4 \end{pmatrix}$
Варіант 9	2) $\begin{cases} x_1 + 2x_2 + x_3 + x_4 = 8 \\ 2x_1 + x_2 + x_3 + x_4 = 5 \\ x_1 - x_2 + 2x_3 + x_4 = -1 \\ x_1 + x_2 - x_3 + 3x_4 = 10 \end{cases}$ 3) $2A - (A^2 + B)B$, де $A = \begin{pmatrix} 1 & 4 & 2 \\ 2 & 1 & -2 \\ 0 & 1 & -1 \end{pmatrix}$, $B = \begin{pmatrix} 4 & 6 & -2 \\ 4 & 10 & 1 \\ 2 & 4 & -5 \end{pmatrix}$
Варіант 10	1) $\begin{cases} 4x_1 + x_2 - x_4 = -9 \\ x_1 - 3x_2 + 4x_3 = 7 \\ 3x_2 - 2x_3 + 4x_4 = 12 \\ x_1 + 2x_3 + 4x_4 = 12 \end{cases}$ 3) $3(A^2 - B^2) - 2AB$, де $A = \begin{pmatrix} 4 & 2 & 1 \\ 3 & -2 & 0 \\ 0 & 1 & 2 \end{pmatrix}$, $B = \begin{pmatrix} 2 & 0 & 2 \\ 5 & -7 & -2 \\ 1 & 0 & -1 \end{pmatrix}$

Варіант 11	2) $\begin{cases} 2x_1 - x_2 + x_3 - x_4 = 1 \\ 2x_1 - x_2 - 3x_4 = 2 \\ 3x_1 - x_3 + x_4 = -3 \\ 2x_1 + 2x_2 - 2x_3 + 5x_4 = -6 \end{cases}$ 3) $(2A - B)(3A + B) - 2AB$, де $A = \begin{pmatrix} 1 & 0 & 3 \\ -2 & 0 & 1 \\ -1 & 3 & 1 \end{pmatrix}$, $B = \begin{pmatrix} 7 & 5 & 2 \\ 0 & 1 & 2 \\ -3 & -1 & -1 \end{pmatrix}$
Варіант 12	2) $\begin{cases} x_1 + x_2 - x_3 - x_4 = 0 \\ x_2 + 2x_3 - x_4 = 2 \\ x_1 - x_2 - x_4 = -1 \\ -x_1 + 3x_2 - 2x_3 = 0 \end{cases}$ 3) $A(A^2 - B) - 2(B + A)B$, де $A = \begin{pmatrix} 2 & 3 & 1 \\ -1 & 2 & 4 \\ 5 & 3 & 0 \end{pmatrix}$, $B = \begin{pmatrix} 2 & 7 & 13 \\ -1 & 0 & 5 \\ 5 & 13 & 21 \end{pmatrix}$
Варіант 13	2) $\begin{cases} 5x_1 + x_2 - x_4 = -9 \\ 3x_1 - 3x_2 + x_3 + 4x_4 = -7 \\ 3x_1 - 2x_3 + x_4 = -16 \\ x_1 - 4x_2 + x_4 = 0 \end{cases}$ 3) $(A + B)A - B(2A + 3B)$, де $A = \begin{pmatrix} 1 & -2 & 3 \\ 2 & 3 & 5 \\ 1 & 4 & -1 \end{pmatrix}$, $B = \begin{pmatrix} 4 & 11 & 3 \\ 1 & 6 & 1 \\ 2 & 2 & 16 \end{pmatrix}$
Варіант 14	2) $\begin{cases} 2x_1 + x_3 + 4x_4 = 9 \\ x_1 + 2x_2 - x_3 + x_4 = 8 \\ 2x_1 + x_2 + x_3 + x_4 = 5 \\ x_1 - x_2 + 2x_3 + x_4 = -1 \end{cases}$ 3) $A(2A + B) - B(A - B)$, де $A = \begin{pmatrix} 2 & 3 & 1 \\ 4 & -1 & 0 \\ 0 & 1 & 2 \end{pmatrix}$, $B = \begin{pmatrix} 9 & 8 & 7 \\ 2 & 7 & 3 \\ 4 & 3 & 5 \end{pmatrix}$

Варіант 15	2) $\begin{cases} 2x_1 - 6x_2 + 2x_3 + 2x_4 = 12 \\ x_1 + 3x_2 + 5x_3 + 7x_4 = 12 \\ 3x_1 + 5x_2 + 7x_3 + x_4 = 0 \\ 5x_1 + 7x_2 + x_3 + 3x_4 = 4 \end{cases}$ 3) $3(A+B)(AB-2A)$, де $A = \begin{pmatrix} 2 & 1 & 3 \\ 1 & -2 & 0 \\ 4 & -3 & 0 \end{pmatrix}$, $B = \begin{pmatrix} 22 & -14 & 3 \\ 6 & -7 & 0 \\ 11 & 3 & 15 \end{pmatrix}$	

Лабораторна робота №3
PANDAS
import pandas as pd

Ядром pandas є дві структури даних, в яких відбуваються операції:

1. **Series**
2. **Dataframes**

Series — структура, що використовується для роботи з послідовністю одновимірних даних, а **Dataframe** — для декількох.

Series (серії)

Series — для подання одновимірних структур даних, що подібні до масивів тільки з додатковими можливостями. Він складається з двох пов'язаних між собою масивів. Основний містить дані (дані будь-якого типу NumPy), а в допоміжному index містяться мітки.

Series	
index	value
0	12
1	-4
2	7
3	9

	import pandas as pd
Створення об'єктів Series	
• Конструктор **Series**()	s = pd.Series([12,-4,12,9])
Однак краще створювати Series, використовуючи мітки з певним сенсом, щоб у майбутньому відокремлювати та ідентифікувати дані незалежно від того, в якому порядку вони зберігаються	s = pd.Series([12,-4,12,9], index=['a','b','c','d'])
Якщо необхідно побачити обидва масиви, з яких складається структура, можна викликати два атрибути: **index** і **values**. Атрибут values є вже знайомим нам масивом NumPy	s.values s.index
Об'єкт Series як узагальнений масив NumPy. Може здатися, що об'єкт Series та одновимірний масив бібліотеки NumPy взаємозамінні. Основна відмінність між ними – індекс. У той час як індекс масиву NumPy, що використовується для доступу до значень, є цілим і описується неявно, індекс об'єкта Series бібліотеки Pandas описується явно і зв'язується зі значеннями. Явний опис індексу розширює можливості об'єкта Series	
• Створення Series з масивів NumPy	arr = np.array([1,2,3,4]) s3 = pd.Series(arr)
• Створення на основі словника: Аргумент може бути словником, в якому стандартний index є відсортованими ключами цього словника	pd.Series({2:'a', 1:'b', 3:'c'})
Фільтрація значень Завдяки тому, що основною бібліотекою в pandas є NumPy, багато операцій, що застосовуються до масивів NumPy, можуть бути використані і у випадку з Series. Однією з таких є фільтрація значень у структурі даних за допомогою умов	s[s > 8]
Операції та математичні функції Інші операції, такі як оператори (+, -, * і /), а також математичні функції, що працюють з масивами NumPy, можуть використовуватися і для Series.	

• Вивести масив з унікальними значеннями	s.unique()
• як часто елементи зустрічаються в Series	s.value_counts()
Доступ до елементів	
За індексом:	s[1]
loc отримує рядки (або стовпці) з певними мітками з індексу. **iloc** отримує рядки (або стовпці) у певних позиціях в індексі (тому він приймає лише цілі числа).	s.iloc[:3] s.loc[:3]
Перетворення в список	s.to_list()

DataFrame (датафрейм)

Dataframe — це таблична структура даних. Її основна задача – дозволити використання багатовимірних Series. Dataframe складається з упорядкованих колекцій колонок, кожна з яких містить значення різних типів (числове, рядкове, логічне і т.д.)

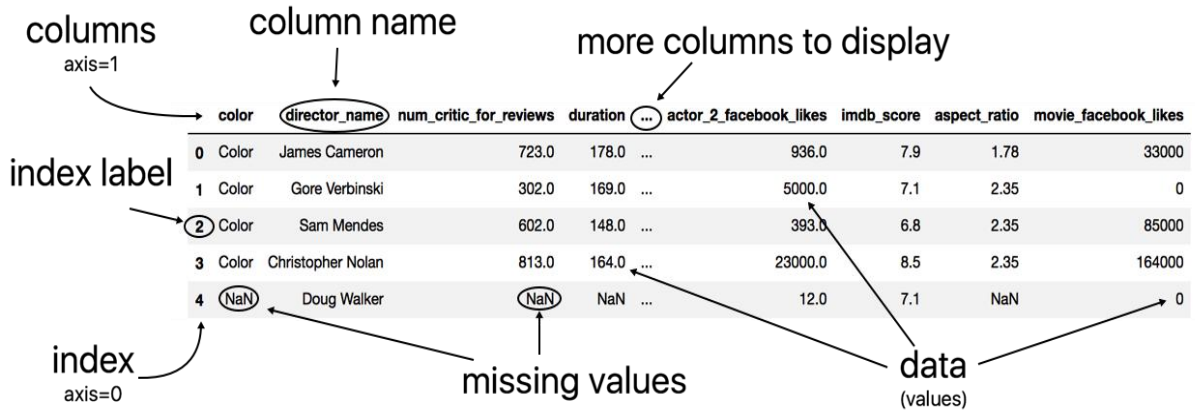
DataFrame

	columns		
index	color	object	price
0	blue	ball	1.2
1	green	pen	1.0
2	yellow	pencil	0.6
3	red	paper	0.9
4	white	mug	1.7

На відміну від Series у якого є масив індексів з мітками, асоційованих з кожним з елементів, Dataframe має відразу два таких. Перший асоційований з рядками (рядами) і нагадує так Series. Кожна мітка асоційована з усіма значеннями у рядгу. Другий містить мітки для кожної з колонок.

Dataframe можна сприймати як dict, що складається з Series, де ключі — назви колонок, а значення — об'єкти Series, які формують колонки самого об'єкта Dataframe. Всі елементи в кожному об'єкті Series пов'язані відповідно до масиву міток, що називається index.

Анатомія Python Pandas DataFrame — Column, Index, Data



	color	director_name	num_critic_for_reviews	duration	...	actor_2_facebook_likes	imdb_score	aspect_ratio	movie_facebook_likes
0	Color	James Cameron	723.0	178.0	...	936.0	7.9	1.78	33000
1	Color	Gore Verbinski	302.0	169.0	...	5000.0	7.1	2.35	0
2	Color	Sam Mendes	602.0	148.0	...	393.0	6.8	2.35	85000
3	Color	Christopher Nolan	813.0	164.0	...	23000.0	8.5	2.35	164000
4	NaN	Doug Walker	NaN	NaN	...	12.0	7.1	NaN	0

Три компоненти DataFrame:

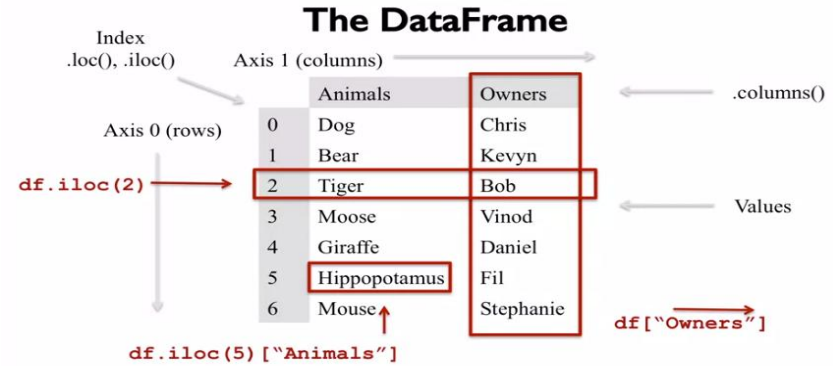
DataFrame складається з трьох різних компонентів: **індексу, стовпців і даних**. Дані також відомі як значення.

Індекс - це послідовність значень в лівій частині DataFrame. Кожне окреме значення індексу називається index label, іноді індекс згадується як заголовки рядків. У наведеному вище прикладі мітки рядків не дуже цікаві і є цілими числами, починаючи з 0 до n-1, де n — кількість рядків у таблиці.

Стовпці являють собою послідовність значень у верхній частині DataFrame.

Все інше є **даними** або значеннями. Іноді датафрейми називають табличними даними. Це просто ще одне ім'я даних прямокутної таблиці з рядками і стовпцями.

Кожен рядок має **мітку**, кожний стовпчик має **мітку.** Ці мітки використовуються для посилання на конкретні рядки або стовпці в DataFrame.



Створення	
• **з pd.Series**	purchase_1 = pd.Series({'Name': 'Chris', 'Item Purchased': 'Dog Food', 'Cost': 22.50}) purchase_2 = pd.Series({'Name': 'Kevyn', 'Item Purchased': 'Kitty Litter', 'Cost': 2.50}) purchase_3 = pd.Series({'Name': 'Vinod', 'Item Purchased': 'Bird Seed', 'Cost': 5.00}) df = pd.DataFrame([purchase_1, purchase_2, purchase_3], index=['Store 1', 'Store 1', 'Store 2'])
• **Зі словників**	data = {'color' : ['blue', 'green', 'yellow', 'red', 'white'], 'object' : ['ball', 'pen', 'pencil', 'paper', 'mug'], 'price' : [1.2, 1.0, 0.6, 0.9, 1.7]} frame = pd.DataFrame(data)
Навіть для об'єктів Dataframe якщо мітки явно не задані в масиві index, pandas автоматично присвоює числову послідовність, починаючи з нуля. Якщо ж індексам Dataframe потрібно присвоїти мітки, необхідно використовувати параметр index і присвоїти йому масив з мітками.	

• визначення **трьох аргументів** в конструкторі в наступній послідовності: матриця даних, масив значень для параметру index і масив з назвами колонок для параметра columns.	frame3 = pd.DataFrame(np.arange(16).reshape(4,4), index=['red', 'blue', 'yellow', 'white'], columns=['ball', 'pen', 'pencil', 'paper'])
• **Завантаження** даних з файлу. Найчастіше дані зберігаються в еселівських таблицях або csv-файлах.	df = pd.read_csv('data.csv', sep=';') (зчитати без заголовка df = pd.read_csv (" data.txt", header= None)) df = pd.read_excel('file.xlsx')
Вибір елементів	
• Інформація про датафрейм: стовпці та їх типи	frame.info()
• Назви всіх колонок	frame.columns
• Список індексів.	frame.index
• Весь набір даних	frame.values
Вибірка даних	
• Вивести перших 5 елементів (для іншої кількості елементів можна вказати в дужках число)	frame.head()
• Вивести останні 5 елементів	frame.tail()
• Вивести стовпчик: вказавши у квадратних дужках назву **стовпця**	frame['price'] frame[['Name', 'Cost']]
• Назву **колонки** можна використовувати в якості атрибута.	frame.price
• Для **рядків** використовують атрибут loc зі значенням **індекса** рядка	frame.loc[2]
• Для вибору кількох **рядків** можна вказати масив із їх послідовністю.	frame.loc[[2,4]]
• Частина датафрейму	frame[0:1]
• .loc робить вибір рядків, і він може приймати два параметри, індекс рядків та список назв стовпців.	frame.loc['Store 1']['Cost'] frame.loc[:,['Name', 'Cost']] .loc також підтримує нарізку. Якщо ми хотіли вибрати всі рядки, ми можемо використовувати стовпчик, щоб вказати повний фрагмент від початку до кінця. А потім додайте ім'я стовпця як другий параметр як рядок.
• Додавання нової **колонки**	frame['new'] = 12 frame['new'] = [3.0, 1.3, 2.2, 0.8, 1.1]
• Знищення **колонки**	del frame['new']
• Перейменування **колонки**	`df = df.rename(columns={'Name': 'FullName'})`
• Транспонування	frame.T
• Змінити індекс	frame.index=[, , , ,] frame. set_index("Name", inplace = True)

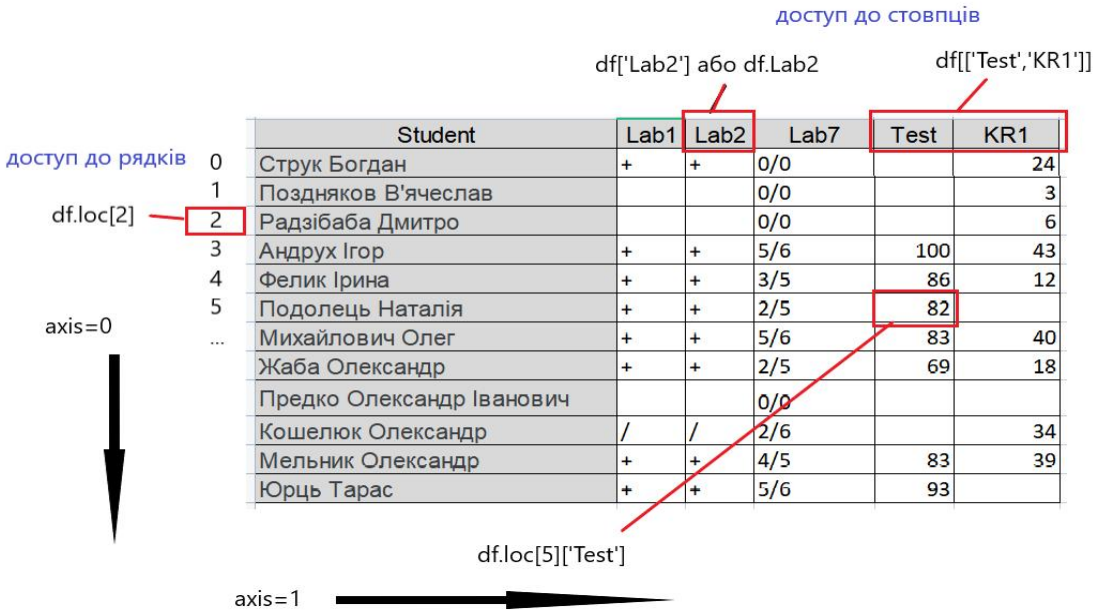
Робота з порожніми комірками

Реальні дані часто містять відсутні значення,
які в багатьох випадках замінюються **nan** (nan розшифровується як «Not-A-Number»).

isnull() — генерує булеву маску для відсутніх даних.
notnull() — протилежний до методау isnull().
dropna() — повертає відфільтрований варіант даних без порожніх комірок.
fillna() — повертає копію даних, в якій порожні комірки заповнені

• Заповнити nan клітинки певним значенням і зберегти зміни в датафреймі	df=df.fillna(2023)
• Видалити рядки, де є nan значення і зберегти зміни в датафреймі	df=df.dropna(axis=0)

Завдання для виконання



	Student	Lab1	Lab2	Lab7	Test	KR1
0	Струк Богдан	+	+	0/0		24
1	Поздняков В'ячеслав			0/0		3
2	Радзібаба Дмитро			0/0		6
3	Андрух Ігор	+	+	5/6	100	43
4	Фелик Ірина	+	+	3/5	86	12
5	Подолець Наталія	+	+	2/5	82	
...	Михайлович Олег	+	+	5/6	83	40
	Жаба Олександр	+	+	2/5	69	18
	Предко Олександр Іванович			0/0		
	Кошелюк Олександр	/	/	2/6		34
	Мельник Олександр	+	+	4/5	83	39
	Юрць Тарас	+	+	5/6	93	

1)	Дані знаходяться у файлі KN-4.csv. Завантажте файл.	Зверніть увагу на розділювач (sep=';')
2)	Вивести всі дані	
3)	Вивести розмір таблиці	Кількість стовпців та рядків
4)	Вивести перелік стовпців	
5)	Вивести назви стовпців їх типи	Звернути увагу на типи даних у стовпцях
6)	Вивести дані стовпчика Student	
7)	Вивести дані двох стовпців Student та Test	
8)	Вивести дані рядка з індексом 7	
9)	В кого зі студентів контрольна більша 30 балів	(маска як в numpy)
10)	Бали за тест перевести у 12-бальну систему оцінювання (помножити на 12, поділити на 100)	df.Test=
11)	Який середній бал за контрольну?	
12)	Додати новий стовпчик KR2 з оцінкою 10 для всіх	
13)	Вивести дані про тих, хто не здав першу	**Робота з порожніми**

контрольну. Скільки їх?	**комірками**
14) Які студенти здали абсолютно все? (в їх рядках н відкиньте рядки з nan	
15) Витягнути прізвища студентів, що здали все. Скі	
16) Невизначені значення NaN заповнити нулями	
18) Зарахувати по 2 бали за '+' за лабораторну 1 та лабораторну 2	`df.Name= df.Name.replace('ABC', 'A')`
19) Зарахувати 1 бал за '/' за лабораторну 1 та лабораторну 2	
20) Створити стовпчики Lab71 Lab72 на основі оцінок зі стовпця Lab7	
`data[['A_v3', 'B_v3']] = data['A/B'].str.split('/', expand=True)`	
21) Змінити тип даних в числовий для утворених стовпців	df['Data']=pd.to_numeric(df['Data']) Перевірте через info()
22) Додати стовпчик для Lab3 зі значенням 2 для всіх рядків	
23) Створити стовпчик Залік, де порахувати суму за Lab1, Lab2, Lab3, Lab71, Lab72, Test, KR1, KR2	сума в рядку (axis=1)
24) Перейменувати стовпчики Lab71 і Lab72 на Lab8 І Lab9	
25) Знищити стовпчик Lab7	
26) Чи всі отримали залік (>=50)	___.all()
27) Який максимальний бал за залік?	
28) Який номер рядка,що має найбільшу суму-залік?	
29) Вивести дані про студента, що має найбільшу суму	
30) Вивести дані про студентів, що отримали залік (>=50)	
31) Скільки студентів мають залік	
32) Посортувати по стовпцю суми по спаданню. Вивести трьох найкращих студентів	___.sort_values(by='___', ascending=False)
33) Вивести список студентів, які не отримали залік і прийдуть перездати тест, бо не здавали його	
34) Чи відрізняються дані двох стовпців Lab1 та Lab2? чим відрізняються стовпці Lab1 та Lab3	df.compare(df2)
35) Хто здав першу лабораторну на 1 бал, або не здав (отримав 0) 36) Створити список з прізвищ	df[df['____'].isin([0,1])]
37) Замінити значення у стовпці Заліку з числового на словесний: зарах, незарах	____=np.where(умова, значення_при виконанні умови, значення_при_невиконанні)
38) Вивести прізвища і заліки	

Лабораторна робота №4
PANDAS. Модифікація, фільтрація, групування даних. Зведена таблиця

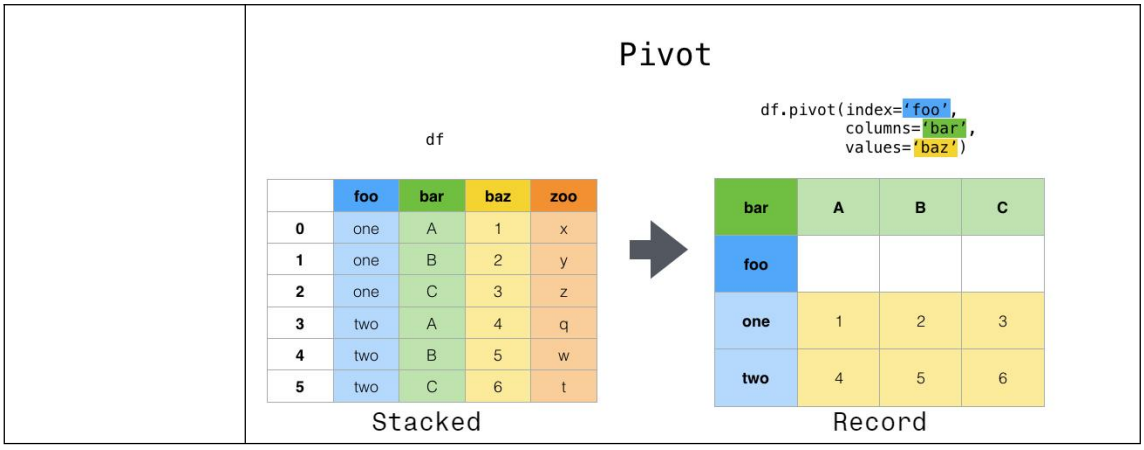
Зчитування		
Зчитування з файлу	frame=pd.**read_csv**('data_file.csv') pd.read_table('table.txt')	
Аргументи:		
sep	pd.read_csv('data_file.csv', sep=';')	Задати розділювач «;»
index_col	pd.read_csv('data_file.csv', index_col=0)	index_col = n (n ціле число) вказує стовчик n для індексації DataFrame
skip_blank_lines	pd.read_csv('data_file.csv', index_col=0, skip_blank_lines=False)	skip_blank_lines=False щоб виключити пусті рядки
Доступ до даних		
З номером	df.iloc[4]	
За допомогою .loc	df.loc[row_selection,column_selection]	df.loc[150:163,['Продаж','Менеджер'])
За назвою стовпця	frame.country	frame[[‘kod’, ‘country’]]
Модифікація таблиці		
Збереження змін	24. переприсвоєння	df = df.drop([0, 4])
	25. Задання параметру **inplace=True**	df.fillna(0, inplace=True)
Додати рядок	df.**append**(*other*, *ignore index=False*, *verify_integrity=False*)	frame=frame.append(new, *ignore_index=True*)
Додавати стовпчик	frame[‘new_col’]=True	Створити елемент, як додавання елементу в словник, де назва стовпця – ключ, а список даних – значення ключа
Знищити рядок та стовпчик	df.drop([0,4], inplace=True) або df = df.drop([0,4])	
	Якщо знищуємо рядок, то вказуємо номер (назву) індекса та вказуємо, що це рядок axis=0	frame.drop([156:178],axis=0, inplace=True)
	Якщо стовпчик, то записати назву стовпця та axis=1	
Знищення стовпчика	**1) del** **2) drop**	del df['C'] df.drop(['B', 'E'], axis='columns', inplace=True)

	• **drop з номерами стовпців**	# or df = df.drop(['B', 'E'], axis=1) df.drop(df.columns[[0, 2]], axis='columns')
Фільтрація даних		
# за маскою	df['Cost']>10 df['Name']=='Alla'	df[df['Cost']>10] df[df['Name']=='Alla']
# маска у функції доступу loc	df.loc[df['Cost']>10]	df[(df['Cost']>10) & (df['Amount']>15)]
# на мові запитів	df.query('Cost>10')	
# за функцією isin()	df['Name'].isin(['Alla'])	df[df['Name'].isin(['Alla'])] df[df['Name'].isin(['Alla','Chris'])]
Модифікація даних		
apply		df['More'] = df['Amount'].apply(lambda x: x+100)
		format = lambda x: '%.2f' % x frame.applymap(format)

	def change_values(val):
	  if val >= 20:
	  return val*1.3
	  else:
	  return val*2.1
	
	df['Change_cost'] = df['Cost'].apply(char
	df
		df['Name'] = df['Name'].apply(str.upper)

Перевірка на входження символів у значенні стовпця		
# містить	str.contains	df[df['Name'].str.contains('EE')]
# починається	str.startswith	df[df['Name'].str.startswith('D')]
# знайти і замінити	str.replace	df['Name'] = df['Name'].str.replace('I', 'A')
Запис у файл		
to_csv		Frame.to_csv('rezult.csv',sep=' ', header=True, index=True)
Операції з даними		
середнє значення одного стовпчика	df['points'].mean()	
#find mean of points and rebounds columns	df[['rebounds', 'points']].mean()	
індекс мінімального (максимального) значення	df.idxmin() df.idxmax()	

кількість унікальних значень (розподіл)	Index.value_counts()	
Значення без повторень	df.unique()	
Кількість значень без повторень	df.nunique()	
# сортування по індексу	frame.sort_index()	
# сортування по стовпцю	frame.sort_values(by='b')	
# кількість нечислових значень	count	
Групування		
Групування по одному стовпцю	df.groupby('по чому групувати')['дані якого стовпця використовувати'] .обч_функція()	df.groupby('legs')['localized_name'].nunique()
		df.groupby('legs')['localized_name'].count()
		concentrations.groupby(['genus']).aggregate(['mean','min', 'max'])
Зведена таблиця		
pd.pivot_table(data, values=None, index=None, columns=None, aggfunc='mean', fill_value=None, margins=False, dropna=True, margins_name='All')	-data — початкова таблиця; -values — стовпчик, значення якого визначають значення зведеної таблиці; -index — ключі для групування, що відносяться до індексів; -columns— ключі для групування, що відносяться до столбців; -aggfunc — функція, яка буде застосована до кожної групи значень значень, згрупованих за значеннями індексу і стовпців. Значення цієї функції і є значення вихідної таблиці.	df.pivot_table(values='(kW)', index='YEAR', columns='Make', aggfunc=np.mean) df.pivot_table(values='(kW)', index='YEAR', columns='Make', aggfunc=np.max)



Завдання для виконання

Формування датасету даних	
14) Дані знаходяться у файлі countries.csv. Завантажте файл.	Зверніть увагу на розділювач (sep)
15) Залишіть тільки рядки даних 2022 року	
16) Стовпців теж забагато: відокремте новий датасет з даними 'Country Name', 'Population', 'Continent Name', 'GDP','Education Expenditure (% GDP)','Health Expenditure (% GDP)' та виконайте всі подальші дії в ньому	
Повторення	
17) Скільки країн подано у таблиці?	
18) Які континенти представлено в таблиці? Скільки їх?	Через команди pandas
19) Скільки європейських країн в таблиці?	
20) Скільки рядків кожного континенту в таблиці? Що за це структура (тип)	____.value_counts()
21) Країн якого континенту найбільше	___.idxmax() з попереднього
22) Чи існують країни, які витрачають на освіту більше 25% Хто це?	**Education Expenditure (% GDP)**
23) Для скількох країн не вказано значення в Health Expenditure (% GDP)	
24) Заповнити порожні комірки в стовпці Health Expenditure (% GDP) значенням 5	
25) Вивести дані про країни, де назва континенту включає 'America'	
Робота з індексами	
26) **Обновити індекс** (нумерація буде здійснена заново)	`df = df.reset_index()` Щоб значення старого індекса не зберігалось окремим стовпцем, додайте параметр (drop=True)
27) Вивести дані по Україні	
28) Вивести ВВП України (GDP)	

29) В якої країни найменше ввп GDP	Вивести назву країну
30) Задати **нову індексацію** по стовпцю 'Country Name'	___.set_index(_____,inplace=True)
31) Вивести знову дані по Україні	
32) Вивести ВВП України (GDP)	
33) В якої країни найменше ввп GDP	Вивести назву країну
34) Обновити -“скинути” індекс	drop=True не вказувати, щоб не пропали назви країн
35) Створити індекс за стовпцем 'Continent Name'	
36) Вивести всі дані про країни South America	
37) Вивести знову дані по Україні	
38) Вивести ВВП України (GDP)	
39) Скинути індекс	
40) Задати індекс назви країн	drop=True не вказувати
41) Яка країна з Південної Америки виділяє найбільший відсоток з ВВП на освіту	
42) Вивести перелік країн, де кількість населення перевищує 200 мільйонів	
Групування	
43) Погрупувати по континентах та знайти загальну кількість населення на кожному континенті Зберегти дані та показати континент з найбільшою кількістю населення	
44) Скинути індекс	
45) Скільки країн на кожному континенті	
46) у попередній команді замість функції обчислення кількості задати команду запису результатів у список	.apply(list)
Об’єднання	
47) Витягнути дані з country-list.csv зі столицями країн	
48) Задати об’єднання двох датасетів за стовпцями назв країн	pd.merge(df1, df2, left_on='col1', right_on='col2', how='inner')
49) Вивести столицю України	
50) Змінити параметр об’єднання, щоб відобразити всі країни з першого датасету	how='left'
Модифікація даних	
51) Перетворити дані стовпця Health Expenditure (% GDP) в цілі числа	
52) У стовпці Education Expenditure (% GDP) підправити значення: якщо значення >=20 - то задати 20, якщо >=10, то 10, в іншому випадку 5	

Лабораторна робота №5
Аналіз даних в датафреймі

Завдання для виконання

1. Завантажити інформацію про злочинність 2010 року в Лос-Анджелесі з файлу **lab4.csv**
(розділювач-сепаратор **‘,’**) та перший стовпчик зробити індексом **index_col=0**

ШВИДКИЙ ПОПЕРЕДНІЙ АНАЛІЗ ДАНИХ:

Загальне ознайомлення з об’ємом інформації, типами даних	
2. Скільки рядків та стовпців в таблиці?	
3. Які назви стовпців	
4. Які типи даних стовпців? Вивести загальні дані про стовпці	
Перевірка на відсутність даних або неточності	
5. Кількість ненульових елементів	df.count()

6. Скільки пропущених даних (невизначених, nan) в стовпцях?	`for i in df.columns:`
	`  print(i, df[i].isna().sum`
	
	`#або`
	`print(df.isna().sum())`

7. Скільки в стовпцях унікальних даних?	Перевірте таким чином дані про стать жертви 'Vict Sex'.
	`print(df['AREA NAME'].unique()) #масив`
	`print(df['AREA NAME'].nunique()) #кіль`
	`print(df['AREA NAME'].value_counts())`

Попередні статистичні дані	

8. Виведіть статистику за певними числовими стовпцями	Статистика за стовпцем “Вік жертви”
	`df['Vict Age'].describe()`
	`count  4553.000000`
	`mean  33.244454`
	`std  17.815441`
	`min  0.000000`
	`25%  23.000000`
	`50%  34.000000`
	`75%  46.000000`
	`max  99.000000`
	`Name: Vict Age, dtype: floa`
	Середній вік жертви злочину - 33
	Std - стандартне відхилення - 18
	Мінміальний - 0
	Максимальний - 99
	50% - половина жертв має вік більше 34 і половина має вік більше 34

	25% -четвертина жертв має вік більше 23 років, та четвертина має вік більше 23 І т.д.
9. Виведіть розподіли значень певних стовпців	Перевірити стовпчик про вік жертви `df['Vict Age'].value_counts()`
ПІДГОТОВКА ДАНИХ ДЛЯ ПОДАЛЬШОЇ РОБОТИ	
10. Видалити стовпчик Vict Descent	
На основі попереднього аналізу в стовпці 'Vict Sex' є значення 'X'. 11. Замінимо значення 'X' стовпці 'Vict Sex' на порожні клітинки np.NaN	Перевірте дію вивівши унікальні значенні стовпця
Аналіз показав, що стовпцю 'Date Rptd' відповідає тип object, що по замовчуванні задається для текстових полів. 12. Задати стовпцю 'Date Rptd' тип date	`df['Date Rptd']=df['Date Rptd'`
13. Сформувати два стовпці з поля 'Date Rptd' на основі **року** та **місяця**	`df['Year'] = df['Date Rptd'` `df['Month'] = df['Date Rptd`
14. Стовпчик 'Date Rptd' знищити	
15. Перевірити стовпчик Year на унікальність даних. Якщо там вказано один рік, то відкинути стовпчик Year і в аналізі використовувати стовпчик з місяцями	
16. Дані скількох місяців показано в таблиці	
На основі попереднього аналізу у стовпці Crm Cd Desc є порожні значення 17. Відкинути рядки з порожніми значеннями у стовпці Crm Cd Desc	*Перевірка: df[df['Crm Cd Desc'].isnull()]* Команда: `df=df.loc[df['Crm Cd Desc']` Перевірити чи знищили рядок
На основі попередньому аналізу відзначено, що при невідомому віці жертви поставлено число 0 18. Всі нульові значення в стовпці Vict Age замінити на np.NaN	
АНАЛІЗ ДАНИХ	
19. Для зручності роботи створимо окремо датафрейм про жертви, вибравши тільки стовпці 'Crm Cd Desc', 'Vict Age', 'Vict Sex'	
20. **Скільки різних видів злочинів зафіксовано** в 'Crm Cd Desc'	
21. Визначити розподіл значень по 'Crm Cd Desc'	value_counts()
22. Вивести **злочин, який найчастіше відбувався** 23. Скільки випадків "найпопулярнішого" злочину відбулося та скільки це відсотків від загальної кількості	Використати попередню команду

24. **Якого віку найчастіше були жертви**? Задати розподіл значень по віку жертви та вивести перші 5 значень індексів	
25. **Скільки випадків пограбувань** ROBBERY зафіксовано в таблиці?	231
26. **Скільки було випадків, що жертвами пограбування були чоловіки**?	() & () 175
27. **Скільки злочинів було пов'язано з автомобілем?** Підрахувати кількість викрадень автомобіля або спроб викрадення 'VEHICLE - ATTEMPT STOLEN' та 'VEHICLE - STOLEN'	() | () або __.isin([,]) 222
28. Обчислене вище значення показати в процентному відношенні до кількості всіх рядків-злочинів датафрейму	4.88
29. **Скільки було випадків використання вогнепальної зброї (зафіксовано в назві злочинів)?** Відфільтрувати рядки де в назві злочину вказана зброя. Скільки їх? Скільки процентів?	WEAPON 456 10.02
30. Сформувати перелік злочинів з використанням зброї.	В утвореному вище вибрати назви злочинів без повторень
31. **Чи правда, що жінки стають частіше жертвами злочину ніж чоловіки?**	
32. **Жертвою якого злочину найчастіше стають жінки**	Розподіл BATTERY - SIMPLE ASSAULT
Розглянемо стовпці-дані про сам злочин	
33. Яке місце найчастіше було місце злочину	Premis Desc
34. Чи були місцем злочину навчальні заклади	SCHOOL
35. Які статуси для злочинів подано в таблиці	Status Desc
36. Скільки проведено арештів (надано статус Adult Arrest)? Який це відсоток від загальної кількості?	
37. Підрахувати кількість рядків для яких поле Weapon Desc заповнено. 38. Що найчастіше зустрічається в даному стовпчику як “зброя”	
39. В датафрреймі погрупувати по статі, порахувати кількість злочинів для кожної статі	
40. Порахувати кількість жертв-жінок для кожного типу злочину	Спочатку профільтрувати жінок і погрупувати
41. Погрупувати по даті. Скільки злочинів скоєно кожного місяця	
42. Для стовпців 'Crm Cd Desc', 'Vict Age' зробити групування по злочину та визначити максимальний вік жертви для кожного злочину	
43. В результаті групування з попереднього завдання знайти злочин, де вік є найменший з обчислений	

максимумів	
44. Створити зведену таблицю, в якій рядки **'Month'**, стовпці -'Crm Cd Desc', а сама таблиця містить результати обчислень кількості жертв 'Vict Sex'	Zv = df.pivot_table(_____, aggfunc=np.count_nonzero) Теорія з попередньої лаби
45. В утвореній зведеній таблиці вивести стовпчик для злочину ROBBERY	
46. Скільки максимум пограбувань в місяць було вчинено	
47. **Які види злочинів зустрічаються кожного місяця?**	*У зведеній таблиці відкинути рядки-місяці, коли злочин не було вчинено* ____.dropna(axis=1)
48. Створити зведену таблицю, що підраховує кількість злочинів для кожної статті кожного місяця	

ВИСНОВОК:

Проведено аналіз даних з файлу lab4.csv, де подано дані про злочинність (дані про злочини та жертви) за 8 місяців 2010 року та отримано наступні результати:

- серед злочинів найчастіше зафіксовано напад та побиття жертв, вони становлять 17 % від загальної кількості зафіксованих злочинів;
- найчастіше жертвами злочинів ставали особи віком від 23 до 27 років;

І т.д.

Лабораторна робота №6
Візуалізація даних

Імпортувати бібліотеку	import matplotlib.pyplot as plt
Побудова графіка	
Задати list comprehension: формування квадратів чисел з проміжку від 0 до 10	x=[ ]
Побудувати графік для списку чисел	plt.plot(x) plt.show()
Задання двох функцій на одній площині	
Задати другий list comprehension: формування кубів чисел з проміжку від 0 до 10	y=
Задати побудову другого графіку	**Plot** можна вказувати декілька раз, команда **plt.show()** записується один раз
Задання аргументу x	

Задати проміжок від -10 до 10 та дві функції	`import numpy as np`
	`x = np.arange(-10,10,0.1)`
	
	`y1 = [i**2+200 for i in x]`
	`y2 = [abs(i)*100 for i in x]`

Побудувати обидва графіки згідно аргументу x	`plt.plot(x,y1)` `plt.plot(x,y2)`
Форматування ліній графіку	
linestyle або ls	solid', 'dashed', 'dashdot', 'dotted' '-', '--', '-.', ':'
Для першого графіку задати **тип лінії** словом, для другої - символом	plt.plot(x,y1, ls='dotted')
color або c	‘red’ ‘r’ ‘#ff0000’
Задати графікам різні **кольори**	
linewidth або lw	lw=5.5
Задати **товщину ліній**	
Заливка області	
# fill_between()	
Задати третю функцію	`y3 = [100*math.sin(i)+150 for i in x]`
Для неї графік задати через заливку області	`plt.fill_between(x,y3)`
Точкові графіки	

Задати x і y	`import numpy as np`
	`x = np.arange(8)`
	`y = np.array([9,2,6,4,0,6,3,1])`

Задати точковий графік	plt.scatter(x, y)
Форматування: - size - marker ".": point "o": circle "s": square "^": triangle "v": upside down triangle "+":	`plt.scatter(x, y, marker='^')`

plus "x": X	
Стовпчасті діаграми	

Задати список чисел
Аргументом буде проміжок в залежності від кількості елементів списку.
Побудувати стовпчасту діаграму
	`y = [5, 8, 2, 3, 5]`
	`x = range(len(y))`
	`plt.bar(x, y)`

Поміняти команду на **plt.barh(x, y)**	
Задати список кольорів ['red', 'orange', 'yellow', 'green', 'blue'] та задати їх для діаграми в параметр color	
Кругова діаграма	
Задати список з чисел	`z = [15, 84, 52, 3]` `plt.pie(z)`
- формат текстової мітки в долі	plt.pie([30,12,25,9], autopct='%1.1f%%')
- винесення з діаграми	ex = (0.3, 0, 0) plt.pie([25,35,55], explode=ex)
- порожнє коло з визначеним діаметром	plt.pie([25,35,55], wedgeprops=dict(width=0.7))
Форматування області діаграми	
Задати list comprehension: формування квадратів чисел з проміжку від -10 до 10	
Побудувати графік для довільного списку чисел	
Задати заголовок	`plt.title('Function')`
Підписати осі	`plt.xlabel('Some data')` `plt.ylabel('Some other data')`
Задати легенду	`plt.plot(x, label='Data')` `plt.legend(loc='best')`
Задати сітку	`plt.grid()`
Додати текст в діаграму	plt.text(5, 5, "Ось тут буде текст")
Додати до тексту форматування:	plt.text(5, 35, "Ось тут буде текст",fontweight="bold", rotation=25, color="red")
Задати межі відображення осей	`plt.xlim(4, 10)` `plt.ylim(80, 100)`

Лабораторна робота №7
Візуалізація даних попередніх лабораторних робіт

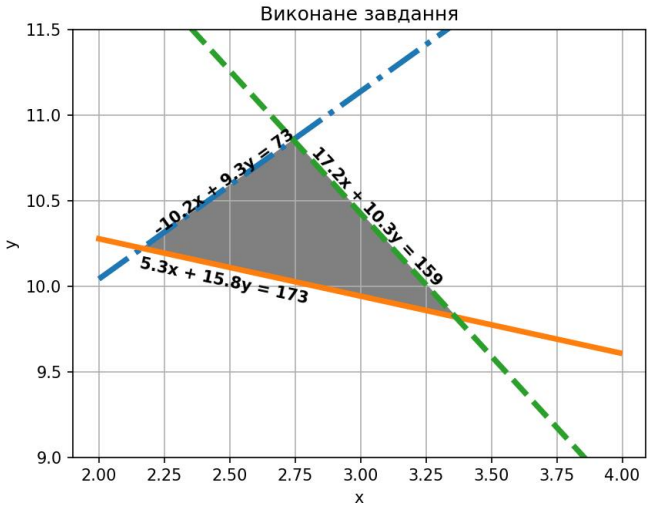
ЗАВДАННЯ 1: Візуалізація до лаб2
Для обраної системи лінійних рівнянь зобразити область, яку вона обмежує своїми трьома прямими.

Розмістити всі лінії на одному рисунку.
– всі прямі повинні бути різного кольору;
– всі прямі повинні мати різний тип ліній (пунктирна, точка-тире тощо);
– підібрати масштаб таким чином, щоб всі три точки перетину прямих були в області видимості;
– додати сітку;
– зробити підписи рівнянь прямих вздовж лінії з відповідним нахилом;
– додати підписи осей та назву графіку;
– заповнити кольором область, що утворена перетином всіх прямих (використовуйте метод fill_between)

Наприклад:
r4=np.minimum(r1,r3) # обмеження зверху
plt.fill_between(x, r4,r2, where = (x > 2.16) & (x <= 3.36), color='gray')
# заливка між значеннями
Або np.maximum

Зразок:

10	$-10.2\,x_1 + 9.3\,x_2 = 73$ $5.3\,x_1 + 15.8\,x_2 = 173$ $17.2\,x_1 + 10.3\,x_2 = 159$



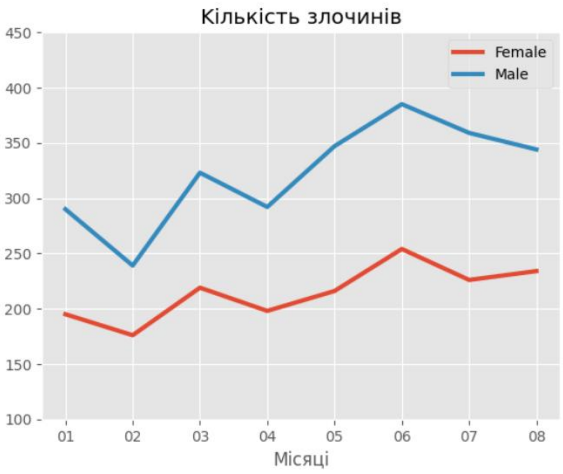
Завдання до лабораторної роботи

Вар.	Система рівнянь	Вар.	Система рівнянь
1	$4.2\,x_1 + 10\,x_2 = 136$ $-10.1\,x_1 + 13.8\,x_2 = 132$ $18.3\,x_1 - 7.6\,x_2 = 108$	**6**	$-10.3\,x_1 + 10.2\,x_2 = 70$ $4.7\,x_1 + 12.3\,x_2 = 173$ $13.2\,x_1 + 8.8\,x_2 = 282$
2	$6.2\,x_1 + 6.6\,x_2 = 83$ $-9.6\,x_1 + 13.8\,x_2 = 72$ $-13.2\,x_1 + 5.7\,x_2 = 305$	**7**	$3.2\,x_1 + 8.8\,x_2 = 89$ $-14.2\,x_1 + 10.8\,x_2 = 125$ $20.3\,x_1 - 7.2\,x_2 = 142$
3	$8.2\,x_1 + 7.9\,x_2 = 123$ $11.2\,x_1 + 16.3\,x_2 = 201$ $-16.3\,x_1 - 8.3\,x_2 = 169$	**8**	$13.1\,x_1 + 9.2\,x_2 = 173$ $8.3\,x_1 + 21\,x_2 = 271$ $3.2\,x_1 - 8.3\,x_2 = 34$
4	$10.2\,x_1 - 3.2\,x_2 = 149$ $-5.8\,x_1 + 16\,x_2 = 83$ $10.3\,x_1 + 7.3\,x_2 = 234$	**9**	$8.3\,x_1 + 8.2\,x_2 = 134$ $-10.2\,x_1 + 15.2\,x_2 = 102$ $14.5\,x_1 - 7.3\,x_2 = 141$
5	$8.1\,x_1 + 8.5\,x_2 = 72$ $-3.2\,x_1 + 15.2\,x_2 = 42$ $10.2\,x_1 + 8.8\,x_2 = 205$	**10**	$-10.2\,x_1 + 9.3\,x_2 = 73$ $5.3\,x_1 + 15.8\,x_2 = 173$ $17.2\,x_1 + 10.3\,x_2 = 159$

ЗАВДАННЯ 2: Візуалізація до лаб3
Для файлу “KN-4.csv” з лабораторної робота №3 для результатів KR-1 побудувати стовпчасту діаграму з відображенням прізвищ студентів як аргументів
- Діаграма plt._____(стовпчик прізвищ, стовпчик оцінок)
- Напрямок відображення підписів осі x: plt.xticks(rotation=величина_кута)
- Надпис, сітка
−plt.tight_layout() для того щоб прізвища вмстились

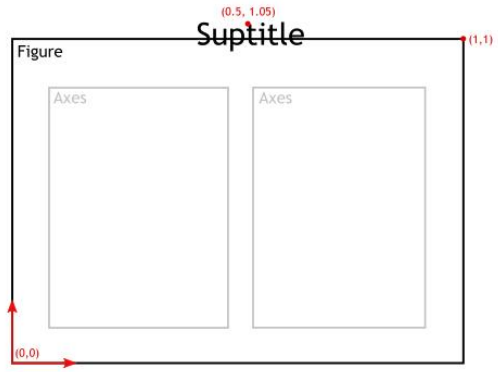
ЗАВДАННЯ 3: Візуалізація до лаб4
Для сформованого датасету даних в лаб4 підрахувати загальну кількість населення кожного континенту та показати результати в круговій діаграмі:
- задати формат текстової мітки в долі (autopct)
- заголовок
- підписи складових частин

ЗАВДАННЯ 4: Візуалізація до лаб5
Для результату п.48. “Створити зведену таблицю, що підраховує кількість злочинів для кожної статті кожного місяця” з лабораторної робота №5 показати у вигляді графіка (стиль plt.style.use("ggplot")).

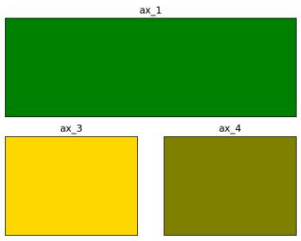
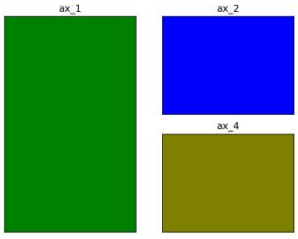

Кількість злочинів
Female
Male
450
400
350
300
250
200
150
100
01
02
03
04
05
06
07
08
Місяці

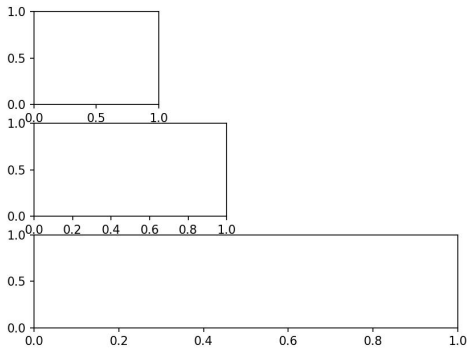
Лабораторна робота №8.
Взаємне розміщення декількох графіків

Завдання для виконання

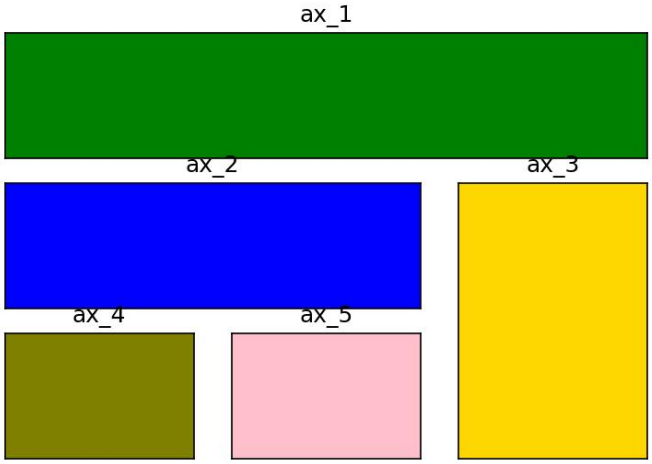
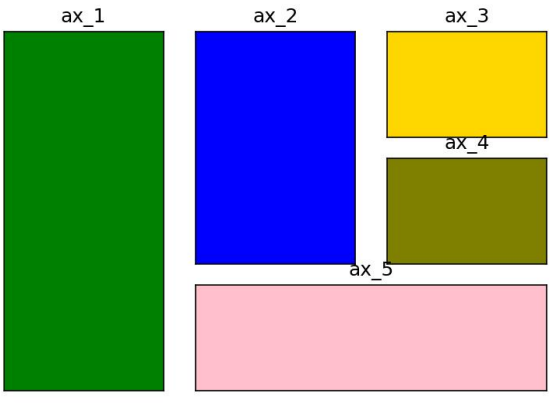
Задано словник кількостей ліцензійних місць університетів м. Івано-Франківськ за 2023 рік.	univer = {'MedUn':1770, 'PNU': 11175, 'KD':1612, 'IFUNG':4729}
Витягнути для осі абсцис ключі словника, для осі ординат - значення ключів словника	
Побудова	Теорія https://matplotlib.org/stable/gallery/subplots_axes_and_figures/subplots_demo.html
Задати змінні для визначення фігури та областей: кількість рядків 1, стовпців 2	`fig, ax = plt.subplots(nrows=1, ncols=2)`
У першій області задати графік, в другій - стовпчасту діаграму для університетських даних	`ax[0].plot(x, y)` `ax[1].bar(x,y)`
Підправити команду задавши розмір figsize та визначити одну вісь ординат на дві діаграми sharey	`fig, ax = plt.subplots(nrows=1, ncols=2, figsize=(6`
Розширити фігуру ще на один графік та розмір збільшити	... nrows=1, ncols=3... figsize=...
Додати в третю область точковий графік	.scatter
Форматування:	
Підписати кожну область	`ax[0].set_title('Перша область')`
Залити кольором кожну область	`ax[0].set_facecolor('gray')`
Задати заголовок всього зображення	fig.suptitle('Лабораторна робота №7')
Змістити заголовок вище, адже він накладається на інші заголовки 	fig.subplots_adjust(top=0.8)
Задати підписи осей абсцис кожної області	ax[0].set_xlabel('Лінійний графік') …
Змістити низ області вище на 0,2, щоб підписи областей помістились	Підправити команду fig.subplots_adjust … bottom=0.2
Отримане зображення здати	
Структури різної форми	

Дано код поділу областей 2 на 2:
fig = plt.figure()
# перше число - кількість рядків; друге - кількість стовпців; третє - індекс клітинки. 	
ax_1 = fig.add_subplot(2, 2, 1)
ax_2 = fig.add_subplot(2, 2, 2)
ax_3 = fig.add_subplot(2, 2, 3)
ax_4 = fig.add_subplot(2, 2, 4)
 	
ax_1.set(title = 'ax_1', xticks=[], yticks=[],facecolor = 'green')
ax_2.set(title = 'ax_2', xticks=[], yticks=[],facecolor = 'blue')
ax_3.set(title = 'ax_3', xticks=[], yticks=[],facecolor = 'gold')
ax_4.set(title = 'ax_4', xticks=[], yticks=[],facecolor = 'olive')	

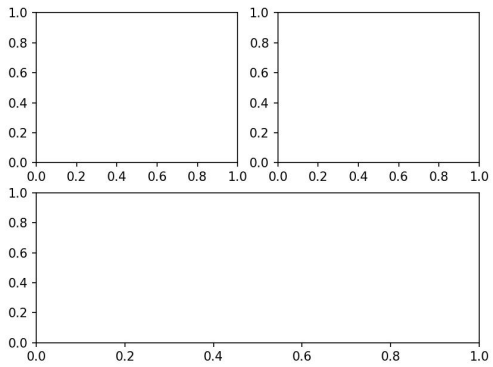
Сформувати таку область 	ax_1 бачить поділ області на 2 рядки, 1 колонку і займає 1 клітинку ax_1 = fig.add_subplot(2, 1, 1) ax_3 бачить поділ області 2 на 2 і займає 3 клітинку по рахунку при такому поділі ax_4 бачить поділ області 2 на 2 і займає 4 клітинку ax_2 зайва Не здавати
По аналогії зробити 	Не здавати
Випадки 3 на 3: fig = plt.figure() ax_1 = fig.add_subplot(3, 3, 1) ax_2 = fig.add_subplot(3, 3, 5) ax_3 = fig.add_subplot(3, 3, 9) Не будуючи сказати як виглядатиме	
Зробити:	

<table>
<tr><td>  </td><td></td></tr>
<tr><td>Побудувати наступну структуру за допомогою gridspec

import matplotlib.gridspec as gridspec
plt.figure(figsize=(6, 4))
G = gridspec.GridSpec(3, 3)

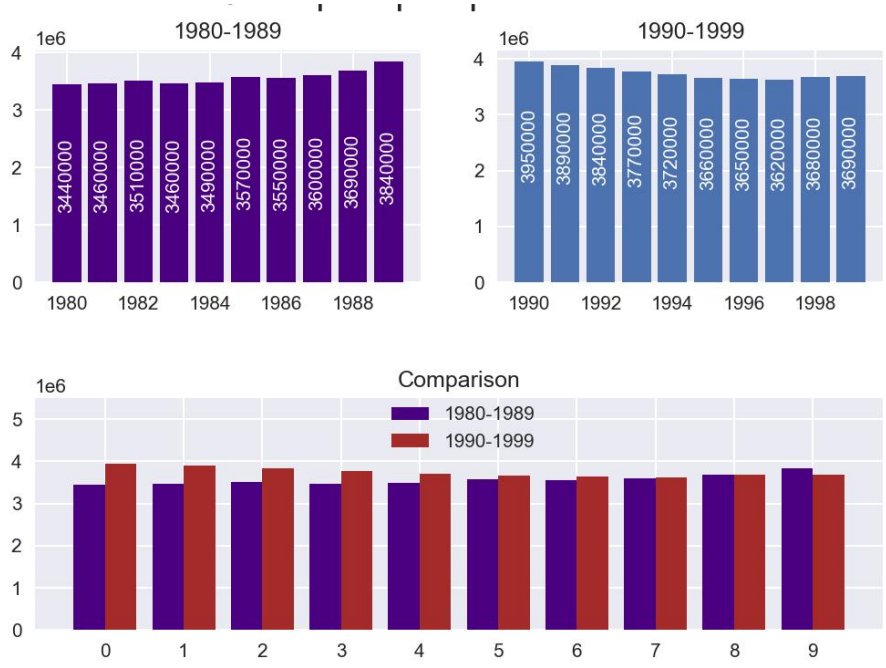
ax_1 = plt.subplot(G[0, :])
ax_1.set(title = 'ax_1', xticks=[], yticks=[],facecolor = 'green')
ax_2 = plt.subplot(G[1, :-1])
ax_2.set(title = 'ax_2', xticks=[], yticks=[],facecolor = 'blue')
ax_3 = plt.subplot(G[1:, -1])
ax_3.set(title = 'ax_3', xticks=[], yticks=[],facecolor = 'gold')
ax_4 = plt.subplot(G[-1, 0])
ax_4.set(title = 'ax_4', xticks=[], yticks=[],facecolor = 'olive')
ax_5 = plt.subplot(G[-1, -2])
ax_5.set(title = 'ax_5', xticks=[], yticks=[],facecolor = 'pink')</td><td>  </td></tr>
<tr><td>Зробити і здати</td><td>  </td></tr>
</table>

Лабораторна робота №9
Візуалізації отриманих результатів аналізу

Витягувнути дані з файлу 'NationalNames.csv'. Знайдіть загальну кількість народжень за кожен рік	 Count Year 1880 201484 1881 192699 1882 221538 1883 216950 1884 243467
Визначити загальну кількість народжень за роки від 1980 по 1989 включно та загальну кількість народжень від 1990 по 1999 Зберегти результати групування	`data1 = df[(df['Year']>=1980)&(df['Year']<=1990)` `data2 = df[(df['Year']>=1990)&(df['Year']<=2000)`
Задати наступну структуру	
В першій області побудувати результати за 1980-89 роки у вигляді стовпчикової діаграми В другій за 1990-99	ax_1 .bar(data1.index, data1) …
У третій обидва результати не обираючи індекс як аргумент, а кількість років через проміжок np.arange	ax_3.bar(np.arange(len(data1.index)), data1) …
Щоб побачити в третій області всі стовпці задамо відступи та ширину для стовпців	`ax_3.bar(np.arange(len(data1))-0.2,data1` Для другого +0,2 і та сама ширина
Підравимо шкалу поділу для третьої діаграми	ax_3.locator_params (axis='x', nbins= 20) Для другої та першої області можна задати шкалу 10, оскільки через рік відображення теж сприймається добре
Вивести назви вбудованих стилів для налаштування зовнішнього вигляду діаграм	print(plt.style.available)
Обрати один з них та застосувати для діаграми https://matplotlib.org/stable/gallery/style_sheets/style_sheets_reference.html https://python-charts.com/matplotlib/styles/	plt.style.use('seaborn-v0_8') Команду задати **перед** визначенням фігури та діаграм
Задати відображення даних для третьої діаграми на проміжку від 0 до 5.5 мільйонів по ординаті	`ax_3.set_ylim([0,5500000])`

В утворений простір поставимо діаграму: до bar третьої області дописати label='1980-1989' та label='1990-1999' та задати відображення легенди	`ax_3.bar(np.arange(len(data1))-0.2,data1,width=0.4,` ax_3.legend()
Легенду можна перемістити наступною командою	ax_3.legend(bbox_to_anchor=(0.4, 0.7))

Задати заголовки діаграмам	`ax_1.set_title('1980-1989')`
	`ax_2.set_title('1990-1999')`
	`ax_3.set_title('Comparison')`

Збільшимо відстані між діаграмами	plt.subplots_adjust(wspace=0.2, hspace=0.5)
Задати заголовок всього зображення	
Значення стовпців першої та другої діаграм підписати, округлити до десятків тисяч та повернути підпис на 90 градусів, колір надпису - білий	rects =ax_1.bar(data1.index, data1, color='indigo') for rect in rects: height = rect.get_height() ax_1.text(rect.get_x() + rect.get_width()/2., 0.35*height,'%d' % round(int(height),-4),ha='center', va='bottom', color='white', rotation=90)
	
Додатково:	
Задати фігуру для побудови діаграм зі структурою 2 на 2	
Задати x на проміжку від -5 до 5 з кроком 0.1	
В першій області задати функцію sin x, в другій cos x, в третій sin 2x, в останній - sin x/2	`ax[0, 0].plot(x,np.sin(x))`
Розмістимо декартові осі по центру	Блок команд:

графіків	ax.spines['left'].set_position('center') ax.spines['bottom'].set_position('center') ax.spines['top'].set_visible(False) ax.spines['right'].set_visible(False) задати у цикл for ax in fig.get_axes(): щоб визначити для всіх областей

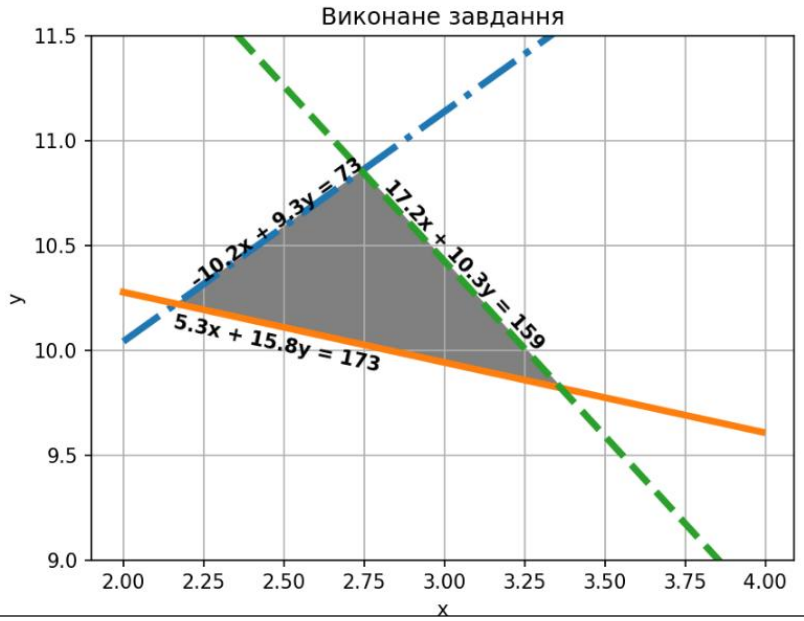
Лабораторна робота №10
Анімація в Matplotlib

26. Дано зразок анімації побудови лінії за допомогою FuncAnimation.

Jupyter notebook задати команду %matplotlib notebook

Зразок 1:

```
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.animation import FuncAnimation

x = [] # порожній список для абсцис
y = [] # порожній список для ординат
fig, ax = plt.subplots() #площина для графіка

# функція, що малюватиме кожен фрейм для анімації
def animate(i):
    t=-4+i*0.1 # початкове значення для x=-4 крок 0.1 (задавайте крок 1 для простих ліній)
    x.append(t) # формування нової абсциси
    y.append(np.sinc(t)) # обчислення ординати - значень нашої функції

    ax.clear() # очистити область
    ax.plot(x, y, lw=3) # побудувати новий графік з доданою точкою

    ax.set_xlim(-4, 4) # проміжок відображення  для x
    ax.set_ylim(-0.3, 1.1)  # проміжок відображення  для y

# запустити анімацію
ani = FuncAnimation(fig, animate, frames=80, interval=30, repeat=False)
# підберіть оптимальну кількість фреймів frames та інтервал interval  для Вашої анімації

plt.show()
```

27. Використавши завдання1 лабораторної роботи №7 згідно свого варіанту здійснити побудову трьох ліній з анімацією на основі зразка1.	
	Змінити t=-4+i*0.1 на проміжок для варіанту, в випадку , що на зображенні початок в x=2, тоді - t=2+i*0.1
	Змінити ax.set_xlim(2, 4) # проміжок відображення для x ax.set_ylim(9, 11,5) # проміжок відображення для y

	Задати
	`y1 = []`
	`y2 = []`
	`y3 = []`

	Обрати першу функцію з варіанту: 73/9.3+10.2*x/9.3 **вставити** у рядок y.append(np.sinc(t)) замість x поставити t і позначити y1 `y1.append(73/9.3+10.2*t/9.3)` Так всі три функції y1 y2 y3
	Задати для них графіки ax.plot(x, **y1**, lw=3)
	Якщо здійснювати заливку області, то параметр x визначте як через масив numpy ax.fill_between(**np.array(x)**, y4, y2, where=(**np.array(x)** > -30) & (**np.array(x)** < 4.5), color='gray')
	при потребі можна змінити параметри
28. Дано зразок 2	***Зразок 2:*** import matplotlib.pyplot as plt import numpy as np from matplotlib.animation import FuncAnimation import pandas as pd

	fig, ax = plt.subplots() # площина для графіка
	
	labels_group = ['SOI', 'KN', 'IST', 'PM', 'M'] # надписи
	
	ex = [0.2]*5 # відступи для **5** даних
	
	def animate(i):
	  ax.clear()
	  ax.pie(**[25, 50, 35, 13, 28]**, labels=labels_group, explode=ex, autopct='%1.1f%%')
	
	  ex[i-1]= 0 #знищення відступів
	
	ani = FuncAnimation(fig, animate, frames=50, interval=560, repeat=False)
	plt.show()

29. Взяти дані з файлу **kn.csv** та обчислити скільки студентів на яку оцінку здали Test1 (розподіл оцінок за перший тест)	stud_test = _____.value_counts() по встовпцю Test1
30. Згідно зразку 2 задати анімацію для кругової діаграми по даних stud_test	**labels_group=stud_test.index** для даних з файлу kn.csv
31. Додати легенду	ax.legend(bbox_to_anchor=(0.85,1.025), loc="upper left")
32. Додати форматування до діаграми	**ax.pie**(_____, wedgeprops={"edgecolor":"k",'linewidth': 1, 'linestyle': 'solid', 'antialiased': True},textprops={'fontsize': 14},shadow = True, startangle = 25)

33. Дано зразок використання стовпчикової діаграми	***Зразок 3:***
	from random import randint
	import matplotlib.pyplot as plt
	from matplotlib.animation import FuncAnimation
	import pandas as pd
	
	univer = {'MedUn':2231, 'PNU': 11812, 'KD':1508, 'IFUNG':7425}
	
	# create empty lists for the x and y data
	x = []
	y = []
	
	# create the figure and axes objects

	fig, ax = plt.subplots()
	
	# function that draws each frame of the animation
	def animate(i):
	x.append(**list(univer.keys())[i]**)
	y.append(**list(univer.values())[i]**)
	ax.clear()
	bars = ax.bar(x, y, color='mediumorchid', align='edge')
	
	ax.bar_label(bars)
	
	ax.tick_params(labelrotation=90)
	ax.set_xlim([0, len(**univer**)])
	ax.set_ylim([0, **max(univer.values())**])
	
	ax.spines['top'].set_color(None)
	ax.spines['right'].set_color(None)
	fig.subplots_adjust(bottom=0.22)
	
	# run the animation
	ani = FuncAnimation(fig, animate, frames=**10,** interval=400, repeat=False)
	
	plt.show()

34. По зразку задати анімацію з результатів студентів за KR-6 файлу kn.csv	x.append(df.iloc[i]['Student']) y.append(df.iloc[i]['KR-6']) І далі по коду
35. Виправити	ax.set_xlim([0, len(**univer**)]) ax.set_ylim([0, **max(univer.values())**])
11 кількість фреймів підправити	ani = FuncAnimation(fig, animate, frames=**10**, interval=400, repeat=False)

Лабораторна робота №11.
SEABORN

Relational plots

relplot	Figure-level interface for drawing relational plots onto a FacetGrid.		
	scatterplot	or **sns.relplot(** **,** kind="scatter") #the default	Draw a scatter plot with possibility of several semantic groupings.
	lineplot	or **sns.relplot(** **,** kind="line")	Draw a line plot with possibility of several semantic groupings.

Distribution plots

displot	Figure-level interface for drawing distribution plots onto a FacetGrid.		
	histplot	kind="hist"; the default	Plot univariate or bivariate histograms to show distributions of datasets.
	kdeplot	kind="kde"	Plot univariate or bivariate distributions using kernel density estimation.
	ecdfplot	kind="ecdf"; univariate-only	Plot empirical cumulative distribution functions.
	rugplot	can be added to any kind of plot to show individual observations	Plot marginal distributions by drawing ticks along the x and y axes.

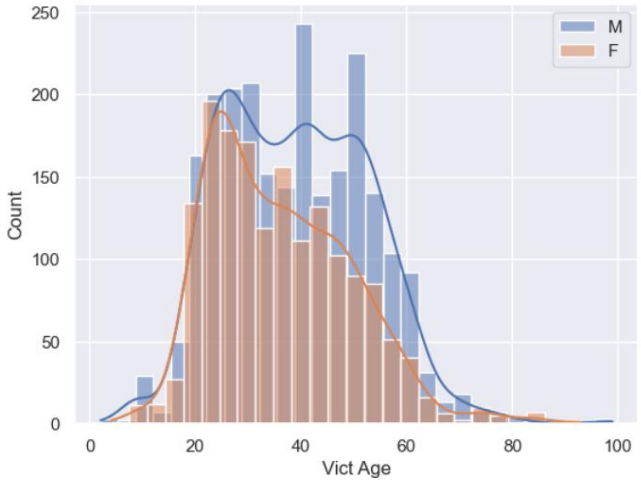
Categorical plots

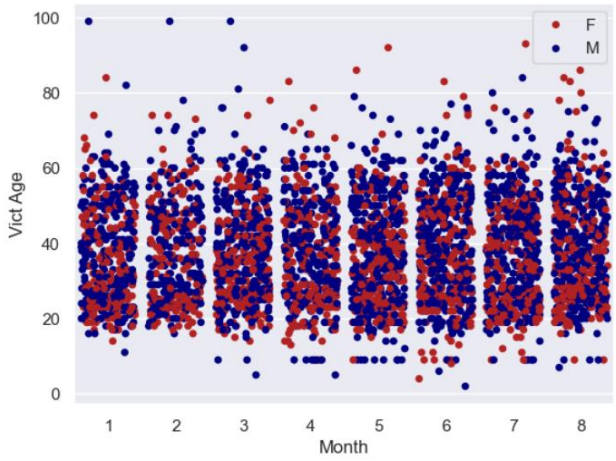
catplot	Figure-level interface for drawing categorical plots onto a FacetGrid.		
	Categorical scatterplots:		
	stripplot	kind="strip"; the default	Draw a categorical scatterplot using jitter to reduce overplotting.
	swarmplot	kind="swarm"	Draw a categorical scatterplot with points adjusted to be non-overlapping.
	Categorical distribution plots:		
	boxplot	kind="box"	Draw a box plot to show distributions with respect to categories.
	violinplot	kind="violin"	Draw a patch representing a KDE and add observations or box plot statistics.
	boxenplot	kind="boxen"	Draw an enhanced box plot for larger datasets.
	Categorical estimate plots:		
	pointplot	kind="point"	Show point estimates and errors using lines with markers.
	barplot	kind="bar"	Show point estimates and errors as rectangular bars.
	countplot	kind="count"	Show the counts of observations in each categorical bin using bars.

Завданн дл виконання

Імпортувати бібліотеку seaborn https://seaborn.pydata.org/index.html	**import seaborn as sns**
Використати дані з файлу lab5.csv	
Кількість жертв щомісяця:	
1 спосіб matplotlib	
Погрупувати по місяцях та порахувати кількість жертв кожного місяця. Зберегти результати	
Побудувати **стовпчикову** діаграму по місяцях і кількостях жертв	*Відзначити для висновку місяці де найбільша кількість жертв та найменша*

2 спосіб seaborn	`countplot` Show the counts of observations in each categorical bin using bars.
Задати діаграму в seaborn, яка сама з основного датасету групує по x та підраховує кількість значень в стовпці	`sns.countplot(x ='Month', data = df)` *Відзначити для висновку відмінності в діаграмах*
Додаткова можливість: додати параметр hue= "Vict Sex", щоб розділити обчисленні значення по статі за кольором	*Як це можна обчислити* Підправити кольори palette=["firebrick", "navy"]
3 спосіб seaborn	
Задати лінійний графік по основному датасету в seaborn за основною командою catplot вказавши тип count	`sns.catplot(x ='Month', data = df, kind='count',`
Додаткова можливість: додати параметр ***hue_order=['M','F'],*** *щоб вказати порядок відображення даних*	
Середній вік жертв щомісяця:	
1 спосіб matplotlib	
Погрупувати по місяцях та порахувати середній вік жертв кожного місяця. Зберегти результати. Побудувати **лінійну** діаграму	
2 спосіб seaborn	`lineplot` Draw a line plot with possibility of several semantic groupings.
Задати лінійний графік по основному датасету в seaborn, вказавши групування по x значень y	`sns.lineplot(x="Month", y="Vict Age", data=df)`
Додаткова можливість: додати параметр hue = "Vict Sex", щоб розділити обчисленні значення по статі за кольором	
Додаткова можливість: додати параметр для визначення функції обчислення максимального віку жертв щомісяця	sns.lineplot(_____, estimator=np.max)
3 спосіб seaborn	`relplot` Figure-level interface for drawing relational plots onto a FacetGrid.
Задати лінійний графік по основному датасету в seaborn за основною командою relplot вказавши тип line	sns.relplot(x=___, data=____, y=_____, kind='line')
Додаткова можливість: додати параметр col="Vict Sex"	*Якщо ще додати параметр hue, що зміниться?*

для розбиття по параметру на дві діаграми	
На базі основного датасету побудувати графік оцінки одновимірного розподілу по вибірці: - обрати стовпчик вік жертви - використати seaborn.histplot() для побудови гістограми з графіком щільності	sns.histplot(x= стовпчик, data=датасет) або `sns.histplot(df['Vict Age'])` *Відзначити для висновку віковий діапазон людей, що найчастіше стають жертвами злочину*
Відобразити лінію оцінки щільності	sns.histplot(_____, kde=True)
Застосувати інший стиль	sns.set_theme(style="darkgrid")
Побудувати дві гістограми з графіками щільності: одна для чоловіків, друга для жінок.	`sns.histplot(df[df['Vict Sex']=='M']['Vict Age'], ...)`
Визначте мітками де чоловіки, де жінки та задайте легенду	sns.histplot(, label='M') sns.histplot(, label='F')
 *Який висновок можна зробити на основі візуалізації?*	
Побудувати точкову діаграму залежностей місяця та віку жертв	`sns.stripplot(x="Month", y="Vict Age", data=df, jitter=0.4)` *Відзначити для висновку місяці де є наймолодші та найстарші жертви*
Додати параметр для розділення по статі та відзначення кольором людей різної статі на діаграмі	sns.stripplot(, hue="Vict Sex", palette=sns.color_palette(["firebrick", "navy"]))
Підправте розміщення легенди	*Відзначити для висновку якої статі були наймолодші та найстарші жертви*



Розмежувати стать	sns.stripplot(, hue="Vict Sex", **dodge=True**)
Показати спільний розподіл по двох змінних	`sns.jointplot(x="Month", y="Vict Age", data=df, kind='scatter')` *Відзначити які графіки вже будували окремо*
Показати автоматизований спільний графік для всього набору числових даних	`sns.pairplot(data = df)` *Відзначити які графіки вже будували окремо*

Завдання для самостійного виконання:

Завантажте набір даних про дитячі імена США з веб-сайту kaggle.com
https://www.kaggle.com/datasets/kaggle/us-baby-names
NationalNames.csv

1.	Отримайте загальну інформацію про дані у наборі даних
	Out[5]: Id Year Count count 1.825433e+06 1.825433e+06 1.825433e+06 mean 9.127170e+05 1.972620e+03 1.846879e+02 std 5.269573e+05 3.352891e+01 1.566711e+03 min 1.000000e+00 1.880000e+03 5.000000e+00 25% 4.563590e+05 1.949000e+03 7.000000e+00 50% 9.127170e+05 1.982000e+03 1.200000e+01 75% 1.369075e+06 2.001000e+03 3.200000e+01 max 1.825433e+06 2.014000e+03 9.968000e+04
2.	Знайдіть кількість унікальних імен у наборі даних
	Out[33]: 93889
3.	Знайдіть 5 найпопулярніших чоловічих імен у 2010 році
	Out[45]: Id Name Year Gender Count 1677392 1677393 Jacob 2010 M 22082 1677393 1677394 Ethan 2010 M 17985 1677394 1677395 Michael 2010 M 17308 1677395 1677396 Jayden 2010 M 17152 1677396 1677397 William 2010 M 17030
4.	Підрахуйте скільки років проводилось спостереження
	Out[218]: 'Спостереження проводилось 135 років'
5.	Підрахуйте кількість записів, для яких `Count` - мінімальне у наборі.
	Out[10]: 254615
6.	Порахуйте, скільки разів хлопчиків називали Barbara
	Out[99]: 4139
7.	Підрахуйте кількість гендерно-нейтральних імен (однакових для дівчат та хлопців)
	Очікуваний результат: Out[85]: 10221
8.	Підрахуйте кількість унікальних імен у кожному році

	Out[26]: Name Year 1880 1889 1881 1830 1882 2012 1883 1962 1884 2158
9.	Знайдіть рік із найбільшою кількістю унікальних імен.
	Out[32]: Name Year 2008 32488
10.	Знайдіть найпопулярніше ім'я в році з найбільшою кількістю унікальних імен (тобто у 2008 році)
	Out[24]: 'Jacob'
11.	Знайдіть загальну кількість народжень за кожен рік
	Out[56]: Count Year 1880 201484 1881 192699 1882 221538 1883 216950 1884 243467
12.	Знайдіть рік, коли народилося найбільше дітей
	Out[49]: 1957
13.	Знайдіть кількість дівчаток та хлопчиків, які народились кожного року
	Gender F M Year 1880 90993 110491 1881 91954 100745 1882 107850 113688 1883 112321 104629 1884 129022 114445
14.	Підрахуйте кількість років, коли дівчаток народжувалось більше, ніж хлопчиків
	Out[64]: 54

Список використаних джерел

1. Бахрушин В.Є. Методи аналізу даних : навчальний посібник для студентів. – Запоріжжя : КПУ, 2011. – 268 с.
2. Василенко О. А. Математично-статистичні методи аналізу у прикладних дослідженнях: навч. посіб. / О. А. Василенко, І. А. Сенча. – Одеса: ОНАЗ ім. О. С. Попова, 2011. – 166 с
3. Марченко О.О., Россада Т.В. Актуальні проблеми Data Mining: Навчальний посібник для студентів факультету комп'ютерних наук та кібернетики. – Київ. – 2017. – 150 с.
4. Ланде Д.В., Субач І.Ю., Бояринова Ю.Є. Основи теорії і практики інтелектуального аналізу даних у сфері кібербезпеки: навчальний посібник. – К.: ІСЗЗІ КПІ ім. Ігоря Сікорського», 2018. — 297 с
5. Bruce P., Bruce A., Gedeck P. Practical Statistics for Data Scientists: 50+ Essential Concepts Using R and Python. O'Reilly, 2020.
6. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly, 2019.
7. Aggarwal C. C. Neural Networks and Deep Learning: A Textbook. Springer, 2019.

інформаційні джерела:

1. Офіційний сайт Python. URL: https://www.python.org/
2. The Python Tutorial. URL: https://docs.python.org/3/tutorial/index.html
3. Путівник мовою програмування Python. URL: https://pythonguide.rozh2sch.org.ua/#_вступ
4. Основи програмування (Python). URL: https://dystosvita.gnomio.com/course/view.php?id=27.
5. Spyder. The Scientific Python Development Environment. URL: https://www.spyder-ide.org/
6. PEP 8 -- Style Guide for Python Code. URL: https://www.python.org/dev/peps/pep-0008/