

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний університет «Запорізька політехніка»

**МЕТОДИЧНІ ВКАЗІВКИ**

та завдання до лабораторних робіт з курсу  
**«Інформаційні системи і технології в маркетингу»**  
для студентів денної та заочної форм навчання спеціальності  
**075 – «Маркетинг»**

2024

Методичні вказівки та завдання до лабораторних робіт з курсу «Інформаційні системи і технології в маркетингу» для студентів денної та заочної форм навчання спеціальності 075 – «Маркетинг» / Укл.: Д.В. Широкорад., А.Є. Рябенко - Запоріжжя: НУ «Запорізька політехніка», 2024. - 33 с.

Укладачі: Д.В. Широкорад, доцент, к.ф.-м.н.,  
А.Є. Рябенко, доцент, к.ф.-м.н.

Рецензент: А.В. Бакурова, професор, д.е.н.

Відповідальний  
за випуск Е.В. Терещенко, доцент, к.ф.-м.н.

Затверджено на засіданні кафедри  
системного аналізу та  
обчислювальної математики.  
Протокол № 3 від 09.10.2024

Рекомендовано до видання  
НМК ФКНТ  
Протокол № 2 від 28.10.2024

## ЗМІСТ

Вступ .....	4
1 Лабораторна робота №1. Створення інтерактивних діаграм в MS EXCEL.....	5
1.1 Загальні теоретичні відомості .....	5
1.2 Завдання до лабораторної роботи .....	6
1.3 Вказівки до виконання лабораторної роботи .....	6
2 Лабораторна робота №2. Основи мови Python.....	17
2.1 Загальні теоретичні відомості .....	17
2.2 Завдання до лабораторної роботи .....	20
2.3 Вказівки до виконання лабораторної роботи .....	20
3 Лабораторна робота №3. Основи використання бібліотеки Pandas .	21
3.1 Теоретичні відомості.....	21
3.2 Завдання до лабораторної роботи .....	23
3.3 Вказівки до виконання лабораторної роботи .....	23
4 Бібліотеки Python для візуалізації даних .....	25
4.1 Теоретичні відомості.....	25
4.2 Завдання до лабораторної роботи .....	28
4.3 Вказівки до виконання лабораторної роботи .....	28
5 Інструмент для візуалізації даних TABLEAU.....	29
5.1 Теоретичні відомості.....	29
5.2 Завдання до лабораторної роботи .....	30
5.3 Вказівки до виконання лабораторної роботи .....	30
Література .....	33

## ВСТУП

Сучасний світ характеризується стрімким розвитком інформаційних технологій, які проникають у всі сфери життя, у тому числі й у маркетинг. Ефективне використання інформаційних систем та технологій стає ключовим фактором успіху для компаній, які прагнуть залишатися конкурентоспроможними на ринку.

Курс "Інформаційні системи і технології в маркетингу" покликаний ознайомити студентів з основними концепціями, інструментами та практиками застосування інформаційних технологій у маркетинговій діяльності. Він охоплює широкий спектр тем, починаючи від розуміння ролі інформаційних систем у маркетингу та закінчуючи конкретними прикладами їх використання для збору, аналізу та візуалізації маркетингових даних.

Під час курсу студенти дізнаються про переваги та виклики, пов'язані з впровадженням інформаційних систем у маркетинг, а також про те, як ці системи можуть підвищити ефективність маркетингових стратегій і кампаній. Вони опанують навички роботи з різноманітними інструментами та платформами, такими як Excel, Tableau, PowerBI, а також з мовою програмування Python для аналізу та візуалізації даних.

Завершивши цей курс, студенти матимуть ґрунтовні знання та практичні навички, необхідні для успішного застосування інформаційних систем і технологій у маркетинговій діяльності. Вони будуть готові до роботи в сучасному цифровому середовищі та зможуть ефективно використовувати переваги інформаційних технологій для досягнення маркетингових цілей.

# 1 ЛАБОРАТОРНА РОБОТА №1. СТВОРЕННЯ ІНТЕРАКТИВНИХ ДІАГРАМ В MS EXCEL

## 1.1 Загальні теоретичні відомості

Інтерактивні діаграми - це візуалізація даних в Excel, яка дозволяє користувачам взаємодіяти з графіком за допомогою елементів керування, таких як фільтри, перемикачі або кнопки. Ці елементи дозволяють динамічно змінювати представлення даних на діаграмі, не змінюючи вихідні дані.

Інтерактивні діаграми забезпечують більшу гнучкість та зручність для аналізу даних. Вони дозволяють користувачам швидко фільтрувати, сортувати та переглядати різні підмножини даних, не перебудовуючи діаграми вручну. Це заощаджує час і полегшує виявлення тенденцій та закономірностей у даних.

Найпоширенішими елементами керування для створення інтерактивних діаграм в Excel є:

- випадаючі списки (Data Validation);
- перемикачі (Check Boxes);
- кнопки (Form Controls);
- елементи керування прокручування (Scroll Bars).

Щоб зробити діаграму інтерактивною, необхідно зв'язати елементи керування з відповідними комітками або діапазонами даних. Це можна зробити за допомогою формул або макросів VBA (Visual Basic for Applications).

Для фільтрації даних на діаграмі можна використовувати функції Excel, такі як OFFSET, INDEX, MATCH та ін. Ці функції дозволяють динамічно змінювати діапазон даних, який візуалізується на діаграмі, на основі значень, вибраних в елементах керування.

Для більш складних сценаріїв взаємодії можна використовувати макроси VBA. VBA дозволяє створювати програмний код для обробки подій, таких як зміна значення в елементі керування, і відповідним чином оновлювати діаграму або інші об'єкти на аркуші.

Для кращої візуалізації та зручності використання інтерактивних діаграм рекомендується дотримуватися принципів належного оформлення, таких як використання відповідних заголовків, підписів осей, легенд та розміщення елементів керування в зручних місцях на аркуші.

Слід пам'ятати, що інтерактивні діаграми можуть стати складними для розуміння та обслуговування, якщо вони містять занадто багато елементів керування або складні формули чи макроси. Тому важливо дотримуватися балансу між функціональністю та простотою використання.

## 1.2 Завдання до лабораторної роботи

Створити інтерактивну діаграму для власного набору даних.

Забезпечити автоматичне оновлення даних з віддаленого ресурсу або з локального файлу.

Діаграма має містити мінімум три ряди даних, що можуть вимикатися/вмикатися за допомогою елемента керування на листі Excel.

Можливі ресурси з датасетами:

- <https://data.world/datasets/marketing>
- <https://www.kaggle.com/datasets/>.

## 1.3 Вказівки до виконання лабораторної роботи

В якості прикладу побудуємо інтерактивну діаграму з даними про продажі Adidas.

Імпортуємо дані з інтернет-сторінки. Для цього на вкладці дані потрібно натиснути на кнопку From Web. У вікні, що відкрилось вставимо

посилання <https://query.data.world/s/mmvrmyxmgdnnewhkhxgeodeinetglw>

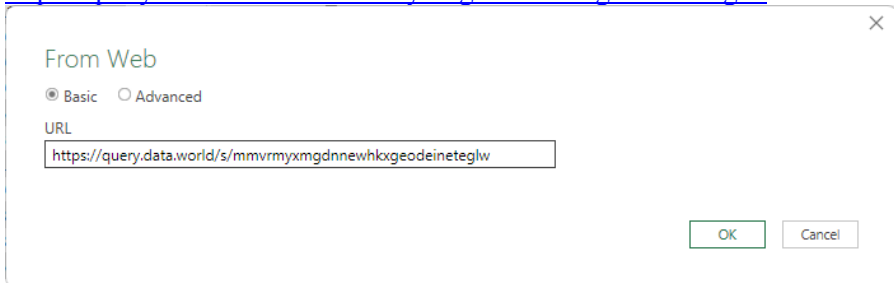


Рисунок 1.1 - Вікно імпорту з інтернет-сторінки

Натиснемо ОК, відкриється вікно PowerQuery. Видалимо непотрібні стовбці, порожні рядки зверху та перейменуємо стовбці за допомогою відповідних кнопок на панелі інструментів. В результаті отримаємо таблицю, показану на рис.



Invoice Date	Units Sold	Operating Profit	Invoice Date	Units Sold	Operating Profit
01.01.2020	2361	382131,84	01.01.2020	2361	382131,84
02.01.2020	2310	225947,16	02.01.2020	2310	225947,16
03.01.2020	2302	202278,51	03.01.2020	2302	202278,51
04.01.2020	2085	228235,52	04.01.2020	2085	228235,52
05.01.2020	1917	222391,56	05.01.2020	1917	222391,56
06.01.2020	1990	206013,7	06.01.2020	1990	206013,7
07.01.2020	2720	373843,48	07.01.2020	2720	373843,48
08.01.2020	2332	222531,22	08.01.2020	2332	222531,22
21.01.2020	2117	202423,16	21.01.2020	2117	202423,16
22.01.2020	1919	247812,14	22.01.2020	1919	247812,14
23.01.2020	1914	239177,22	23.01.2020	1914	239177,22
24.01.2020	1954	209383,3	24.01.2020	1954	209383,3
25.01.2020	2532	383683,66	25.01.2020	2532	383683,66
26.01.2020	2369	305106,56	26.01.2020	2369	305106,56
27.01.2020	2135	223685,35	27.01.2020	2135	223685,35
28.01.2020	1904	257825,52	28.01.2020	1904	257825,52
29.01.2020	1864	238511,14	29.01.2020	1864	238511,14

Рисунок 1.3 - Вихідна і допоміжні таблиці

До створених діапазоном можна застосувати команду **Форматувати як таблицю (Format as Table)** з вкладки **Головна (Home)**

**Додаємо прапорці (checkboxes)** проданої кількості (Units sold) та операційного прибутку (Operating profit).

В **Excel** для цього необхідно відобразити вкладку **Розробник (Developer)**. Для цього треба натиснути **Файл - Параметри - Налаштування стрічки - включити прапорець «Розробник» (File - Options - Customize Ribbon - Developer)**.

Після цього на вкладці стрічки **Розробник (Developer)** в списку, що розкривається **Вставити (Insert)**, вибираємо інструмент **Прапорець (Checkbox)** і малюємо два прапорці-галочки для включення-виключення обох показників, проданої кількості (Units sold) та операційного прибутку (Operating profit).

Текст прапорців можна поміняти, клацнувши по ним правою кнопкою миші і вибравши команду **Змінити текст (Edit text)**.

Тепер необхідно прив'язати прапорці до будь-яких комірок для визначення того, чи включений прапорець чи ні. Для цього клацніть правою кнопкою миші по черзі по кожному доданому прапорцю і виберіть команду **Формат об'єкта (Format Control)**, а потім у вікні задайте **Зв'язок з коміркою (Cell link)**.

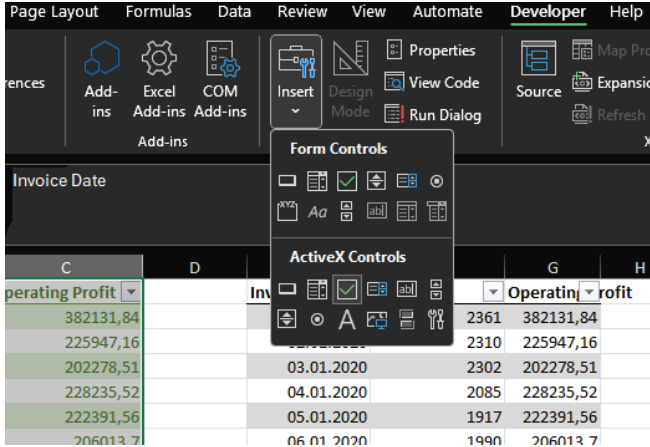


Рисунок 1.4 - Вставка елемента «Checkbox» на аркуш

	K	L
	TRUE	TRUE
	<input checked="" type="checkbox"/> Uits sold	<input checked="" type="checkbox"/> Profit

Рисунок 1.5 - Елементи керування та пов'язані комірки

Кожен прапорець прив'язаний до відповідної клітинки над ним. При включенні прапорця в пов'язану комірку буде виводитися **TRUE**, при вимиканні - **FALSE**. Це дозволить надалі перевіряти за допомогою формул пов'язані комірки і виводити в додаткову таблицю або відповідний показник з вихідної таблиці для побудови графіка, або #Н/Д (#N/A), щоб графік не будувався.

Зауважте, що при використанні команди **Форматувати як таблицю (Format as Table)** на першому кроці, формула має використовувати ім'я таблиці і назву колонки. У разі звичайного діапазону, формула буде більш звичного вигляду:

Units Sold	Operating Profit	Invoice Date	Units Sold	Operating Profit	TRUE
2361	382131,84	01.01.2020	=IF(K\$1;Data_Sales_Adidas@[Units Sold];\"		
2310	225947,16	02.01.2020	=IF(logical_test; [value_if_true]; [value_if_false])		
2302	202278,51	03.01.2020	2302	202278,51	

Рисунок 1.6 - Приклад формули для побудови допоміжної таблиці

Зверніть увагу на часткове закріплення посилання на жовту клітинку (K\$1), тому вона повинна зміщатися вправо, але не повинна - вниз, при копіюванні формули на весь діапазон.

Тепер при включенні-виключенні прапорців додаткова таблиця заповнюється або даними з вихідної таблиці, або штучно створеної помилкою # Н /Д, яка не генерує лінії на графіку.

Тепер додамо на аркуш **Excel** смуги прокрутки, за допомогою яких користувач зможе легко зрушувати графік по осі часу і змінювати масштаб його збільшення.

**Смугу прокрутки (Scroll bar)** необхідно обрати там, де і прапорці - на вкладці **Розробник (Developer)**.

Рисуємо на аркуші в будь-якому відповідному місці одну за одною дві смуги - для зсуву за часом і масштабу.

Минуле	105	Майбутнє	
Ближче	334	Далі	

Рисунок 1.7 - Смуги керування

Кожну смугу прокрутки треба пов'язати зі своєю коміркою (праворуч від смуги), куди буде виводитися числове значення положення повзунка. Його ми потім будемо використовувати для визначення масштабу і зсуву. Для цього клацніть правою кнопкою миші по смугі і виберіть в контекстному меню команду **Формат об'єкта (Format control)**. У діалоговому вікні можна задати пов'язану комірку і мінімум-максимум, в будь-яких межах де буде пересуватись повзунок.

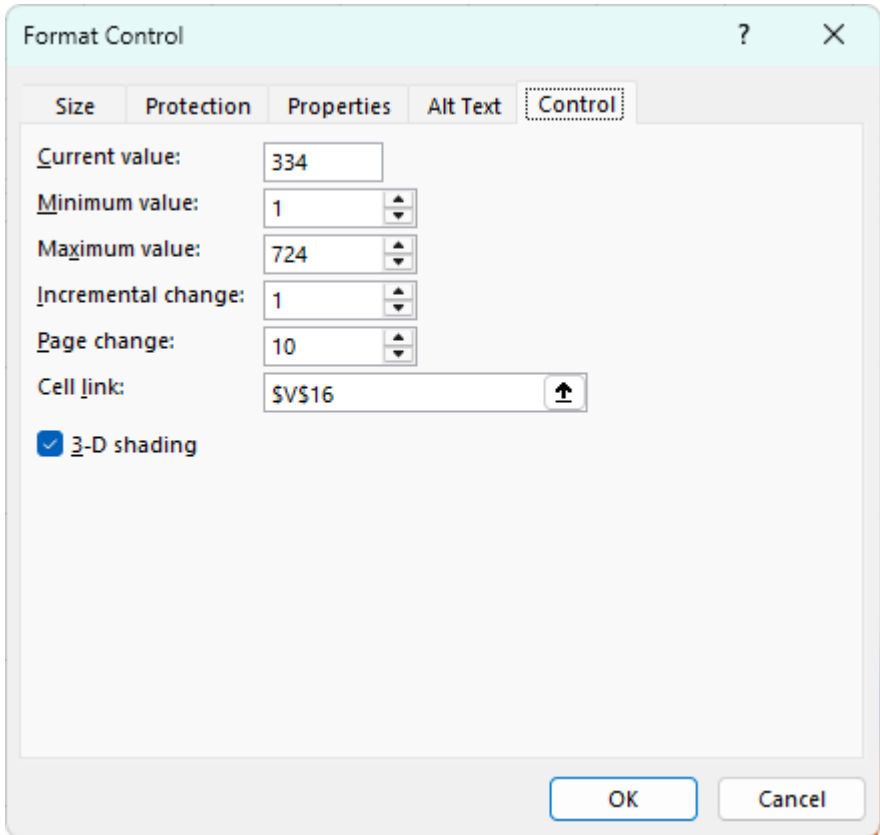


Рисунок 1.8 - Налаштування смуги прокрутки

Щоб відображати на графіку дані тільки за певний інтервал часу, створимо іменованний діапазон, який буде посилатися тільки на потрібні комірки в додатковій таблиці. Цей діапазон буде характеризуватися двома параметрами:

- відступом від початку таблиці вниз на задану кількість рядків, тобто відступом по часовій шкалі минуле-майбутнє (верхня комірка на рис. 1.7);
- кількістю комірок по висоті, тобто масштабом (верхня комірка на рис.1.7);

Цей іменованний діапазон ми пізніше будемо використовувати як вихідні дані для побудови діаграми.

Для створення такого діапазону будемо використовувати функцію OFFSET - ця функція створює посилання на діапазон заданого розміру в заданому місці аркуша і має такі аргументи:

=OFFSET(старт;A;B;C;D)

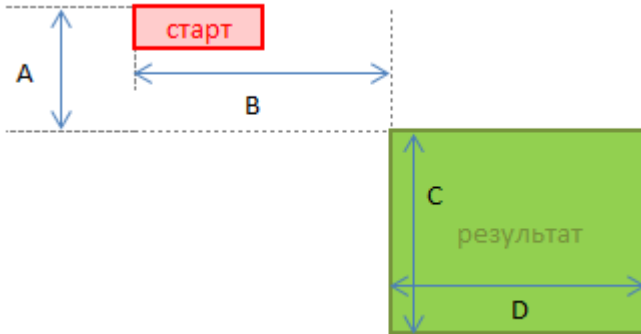


Рисунок 1.9 - Параметри функції OFFSET

Константи в цій формулі можна замінити на посилання на комірки з перемінним вмістом - в нашому випадку, на комірки біля смуг прокрутки. Зробити це можна, створивши динамічний іменованний діапазон з функцією OFFSET. Для цього натисніть кнопку Диспетчер імен (Name Manager) на вкладці Формули (Formulas).

Для створення нового іменованого діапазону потрібно натиснути кнопку Створити (Create) та ввести ім'я діапазону та посилання на комірки у вікні,.

Спочатку створимо два простих статичних іменованих діапазону з іменами, наприклад, Shift та Zoom, які будуть посилатися на верхню та нижню комірки відповідно.

The screenshot shows the 'Edit Name' dialog box in Microsoft Excel. The title bar reads 'Edit Name'. The 'Name' field contains the text 'Shift'. The 'Scope' dropdown menu is set to 'Workbook'. The 'Comment' field is empty. The 'Refers to' field contains the formula '=Data Sales Adidas!\$V\$10'. At the bottom, there are 'OK' and 'Cancel' buttons.

Рисунок 1.10 - Іменованний діапазон Shift

The screenshot shows the 'Edit Name' dialog box in Microsoft Excel. The title bar reads 'Edit Name'. The 'Name' field contains the text 'Zoom'. The 'Scope' dropdown menu is set to 'Workbook'. The 'Comment' field is empty. The 'Refers to' field contains the formula '=Data Sales Adidas!\$V\$16'. At the bottom, there are 'OK' and 'Cancel' buttons.

Рисунок 1.11 - Іменованний діапазон Zoom

Тепер створимо діапазон з ім'ям Sold, який буде посилатися за допомогою функції OFFSET на дані з кількості проданих одиниць за вибраний відрізок часу, використовуючи щойно створені до цього діапазони Shift і Zoom і комірку E1 в якості точки відліку:

The screenshot shows a dialog box titled "Edit Name" with the following fields:

- Name:** Sold
- Scope:** Data Sales Adidas
- Comment:** (empty text area)
- Refers to:** =OFFSET('Data Sales Adidas'!\$E\$1;Shift;1;Zoom;1)

Buttons for "OK" and "Cancel" are located at the bottom right.

Рисунок 1.12 - Іменованій діапазон Sold

Зверніть увагу, що перед ім'ям діапазону використовується ім'я поточного аркуша - це звужує коло дії іменованого діапазону, тобто робить його доступним в межах поточного аркуша, а не всієї книги. Це необхідно для побудови діаграми в майбутньому.

Аналогічним чином створюється іменованій діапазон Profit для даних отриманому прибутку.

The screenshot shows a dialog box titled "Edit Name" with the following fields:

- Name:** Profit
- Scope:** Data Sales Adidas
- Comment:** (empty text area)
- Refers to:** =OFFSET('Data Sales Adidas'!\$E\$1;Shift;2;Zoom;1)

Buttons for "OK" and "Cancel" are located at the bottom right.

Рисунок 1.13 - Іменованій діапазон Profit

Також створюємо діапазон Labels, який вказує на підписи до осі X, тобто дати для обраного відрізка.

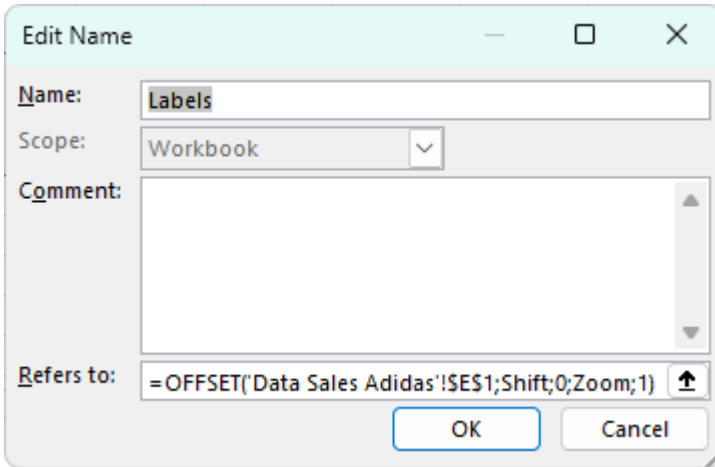


Рисунок 1.14 - Іменованій діапазон Labels

Виділимо кілька рядків у верхній частині допоміжної таблиці, наприклад діапазон E1:G10 і побудуємо по ньому діаграму типу Графік (Line). Для цього в Excel потрібно перейти на вкладку Вставка (Insert) і в групі Діаграма (Chart) вибрати тип Графік (Line). Якщо виділити одну з ліній на створеній діаграмі, то в рядку формул буде видна функція РЯД (SERIES), що обслуговує виділений ряд даних.

Ця функція задає діапазони даних і підписів для виділеного ряду діаграми. Необхідно підмінити статичні діапазони в її аргументах на динамічні, створені раніше. Це можна зробити прямо в рядку формул, наприклад:

```
=SERIES('Data Sales Adidas'!$F$1;'Interactive plot.xlsx'!Labels;'Data Sales Adidas'!Sold;2)
```

Виконавши цю процедуру послідовно для рядів даних проданої кількості та отриманого прибутку, ми отримаємо інтерактивну діаграму – вона буде будуватися за динамічними діапазонами Sold і Profit, а підписи до осі X будуть братися з динамічного ж діапазону Labels. При зміні положення повзунків будуть змінюватися діапазони і, як наслідок, діаграма. При включенні-виключенні прапорців - відображатися тільки ті ряди даних, які нам потрібні.

Таким чином ми маємо повністю інтерактивну діаграму, де можемо відобразити саме той фрагмент даних, що нам потрібен для аналізу.

## 2 ЛАБОРАТОРНА РОБОТА №2. ОСНОВИ МОВИ PYTHON

### 2.1 Загальні теоретичні відомості

#### Виведення інформації на консоль за допомогою функції print()

Функція print() використовується для виведення тексту або даних на консоль. Вона може приймати кілька аргументів, розділених комами, і виводити їх послідовно:

```
print("Вивчайте Python!", "Це цікаво.")
```

Цей код виведе на екран: Вивчайте Python! Це цікаво.

Також можна використовувати форматування рядків для виведення змінних:

```
name = "Анна"
print(f"Привіт, {name}!")
```

#### Змінні та типи даних

Змінна — це ім'я, яке посилається на певне значення в пам'яті. В Python змінну можна створити шляхом присвоєння їй значення:

```
age = 25
name = "Олександр"
```

#### Типи даних

Основні типи даних у Python:

**Цілі числа (int):** цілі значення, наприклад, 5, -3, 100.

**Числа з плаваючою точкою (float):** числа з дробовою частиною, наприклад, 3.14, -0.5.

**Рядки (str):** послідовність символів, наприклад, "Привіт", "Python".

**Логічні значення (bool):** True або False.

#### Арифметичні операції

Python підтримує стандартні арифметичні операції:

**Додавання:** +

**Віднімання:** -

**Множення:** \*

**Ділення:** /

**Цілочисельне ділення:** //

**Залишок від ділення:** %

**Піднесення до степеня:** \*\*

Приклад:

```
a = 10
b = 3
print(a + b) # 13
print(a - b) # 7
print(a * b) # 30
print(a / b) # 3.3333333333333335
print(a // b) # 3
print(a % b) # 1
print(a ** b) # 1000
```

### Робота з рядками

Рядки можна об'єднувати, множити та використовувати спеціальні символи, такі як новий рядок (`\n`) та табуляція (`\t`):

```
greeting = "Привіт"
name = "Світ"
message = greeting + ", " + name + "!"
print(message) # Привіт, Світ!
```

```
multiline_text = "Це перший рядок.\nЦе другий рядок."
print(multiline_text)
```

```
tabbed_text = "Ім'я\tВік\nАнна\t23\nОлексій\t30"
print(tabbed_text)
```

### Умовні оператори

Умовні оператори використовуються для прийняття рішень у програмі:

```
age = 18
```

```

if age >= 18:
    print("Ви повнолітні.")
else:
    print("Ви неповнолітні.")

```

Python використовує відступи для позначення блоків коду, тому важливо правильно форматувати код.

## Цикли

### Цикл for

Цикл for використовується для ітерації по елементах послідовності (наприклад, список, рядок):

```

for i in range(5):
    print(i)

```

Цей код виведе числа від 0 до 4.

### Цикл while

Цикл while виконується, поки умова є істинною:

```

count = 0
while count < 5:
    print(count)
    count += 1

```

## Функції

Функції дозволяють повторно використовувати код. Вони визначаються за допомогою ключового слова def:

```

def greet(name):
    return f"Привіт, {name}!"

message = greet("Іван")
print(message)

```

Функція greet приймає аргумент name і повертає привітання.

## 2.2 Завдання до лабораторної роботи

Завантажити Jupiter ноутбук за посиланням [https://moodle.zp.edu.ua/pluginfile.php/286101/mod\\_assign/introattachments/0/Python\\_basics.ipynb?forcedownload=1](https://moodle.zp.edu.ua/pluginfile.php/286101/mod_assign/introattachments/0/Python_basics.ipynb?forcedownload=1) та виконайте завдання запропоновані в ньому.

## 2.3 Вказівки до виконання лабораторної роботи

Виконувати роботу рекомендується в середовищі Google Colaboratory

1. **Збережіть файл .ipynb на своєму комп'ютері.** Переконайтеся, що у вас є доступ до файлу Jupiter Notebook, який ви хочете відкрити.

2. **Відкрийте Google Colab.** Перейдіть за посиланням <https://colab.research.google.com/> у вашому браузері.

3. **Завантажте файл у Colab.** У верхньому правому куті натисніть на "Файл" (File) та оберіть "Завантажити ноутбук" (Upload notebook). Потім оберіть свій файл .ipynb на комп'ютері.

4. **Очікуйте завершення завантаження.** Після вибору файлу він автоматично завантажиться та відкриється в новій вкладці Google Colab.

5. **Напишіть та запустіть код у блоках.** Щоб запустити код, клацніть на потрібному блоці та натисніть на кнопку "Виконати" (Run), яка позначена значком ► або скористайтесь комбінацією клавіш Shift + Enter.

6. **Зберігання змін.** Щоб зберегти зміни у вашому ноутбуці, оберіть "Файл" (File) > "Зберегти" (Save) або "Зберегти на Діску" (Save a copy in Drive), щоб зберегти копію у вашому Google Діску.

7. Розшарте ноутбук викладачу.

- Після збереження ноутбука на Google Діску натисніть на кнопку "Поділитися" (Share) у верхньому правому куті вікна Colab.

- Налаштуйте права доступу (редагування або лише перегляд).

- Натисніть "Копіювати посилання" та вставте у відповідь до лабораторної роботи в системі Moodle.

## 3 ЛАБОРАТОРНА РОБОТА №3. ОСНОВИ ВИКОРИСТАННЯ БІБЛІОТЕКИ PANDAS

### 3.1 Теоретичні відомості

#### Імпорт бібліотеки Pandas

**Pandas** – це потужна бібліотека Python для роботи з даними, особливо з табличними. Для початку роботи з Pandas необхідно імпортувати бібліотеку:

```
import pandas as pd
```

Ця бібліотека забезпечує зручні структури даних та функції для обробки та аналізу даних, такі як DataFrame та Series.

#### Завантаження даних з CSV-файлу

**CSV (Comma-Separated Values)** – це формат текстових файлів, що використовується для зберігання табличних даних. Pandas пропонує функцію `pd.read_csv()` для завантаження даних з CSV-файлів у DataFrame, що є основною структурою даних Pandas.

Основні параметри функції `pd.read_csv()`:

- **filename:** шлях до файлу або URL.
- **parse\_dates=True:** якщо True, Pandas намагається автоматично розпізнати дати у зазначених стовпцях. Це корисно, коли один зі стовпців містить дати, і вам потрібно їх обробляти як тип даних `datetime`.
- **index\_col=0:** цей параметр вказує на те, що перший стовпець файлу (нумерація з 0) буде використовуватися як індекс DataFrame. Індекс - це мітка, яка допомагає ідентифікувати рядки.

```
df = pd.read_csv('aapl.csv', parse_dates=True,
index_col=0)
```

#### Перегляд перших рядків DataFrame

Щоб швидко ознайомитися зі структурою та вмістом даних, можна скористатися методом `.head()`, який повертає перші п'ять рядків DataFrame за замовчуванням. Якщо потрібно більше або менше рядків, можна вказати потрібну кількість у параметрі методу.

```
df.head() # Повертає перші 5 рядків
```

```
df.head(10) # Повертає перші 10 рядків
```

### Операції з індексами та колонками

DataFrame має два основні компоненти: **індекси** та **колонки**.

- **Індекс** (.index) представляє мітки рядків, що можуть бути числовими або текстовими.
- **Колонки** (.columns) зберігають назви всіх стовпців у вигляді списку.

Ці властивості дозволяють вам отримувати доступ до імен стовпців і до самого індексу.

```
index = df.index # Отримати індекс
columns = df.columns # Отримати список колонок
```

### Типи даних

Метод .dtypes дозволяє перевірити тип даних кожного стовпця в DataFrame. Це важливо, оскільки різні типи даних можуть вимагати різної обробки або підходів до аналізу.

```
df.dtypes # Визначаємо типи даних у стовпцях
```

### Вибірка даних із DataFrame

Pandas надає кілька способів вибору даних:

- **Вибірка за колонками:** можна отримати доступ до колонок DataFrame як через індексацію, так і через атрибути.
- **Вибірка за рядками:** можна використовувати .loc[] для вибірки за мітками або .iloc[] для вибірки за індексом.

```
df['ColumnName'] # Вибірка однієї колонки
df[['Col1', 'Col2']] # Вибірка декількох колонок
```

```
df.loc['RowLabel'] # Вибірка рядка за міткою
df.iloc[0] # Вибірка першого рядка за індексом
```

### Обробка пропущених значень

Часто в даних можуть бути пропущені значення (NaN). Pandas пропонує функції для їх виявлення та обробки:

- **df.isnull()**: повертає DataFrame того ж розміру з булевими значеннями, що показують, де значення пропущене.
- **df.dropna()**: видаляє рядки або колонки з пропущеними значеннями.
- **df.fillna(value)**: заповнює пропущені значення певним значенням.

```
df.isnull() # Визначення пропущених значень
df.dropna() # Видалення рядків з пропущеними значеннями
df.fillna(0) # Заповнення пропущених значень нулями
```

### Групування даних

Pandas надає зручний метод `.groupby()` для групування даних за певними критеріями. Це дозволяє виконувати агрегації, такі як сума, середнє значення, підсумовування тощо, для груп даних.

```
grouped = df.groupby('ColumnName').sum() # Групування та підсумовування за колонкою
```

### 3.2 Завдання до лабораторної роботи

Завантажити Jupiter ноутбук за посиланням <https://colab.research.google.com/drive/1laamt5bS6Y-OuhKyPOFSiT5Me0wmLFrF?usp=sharing> та виконайте завдання запропоновані в ньому.

Допоміжні файли розташовані в системі Moodle <https://moodle.zp.edu.ua/mod/assign/view.php?id=128236>

### 3.3 Вказівки до виконання лабораторної роботи

Для завантаження ноутбука в Google Colab та роботи з ним використовуйте інструкцію із лабораторної роботи №2.

Щоб мати доступ до файлів, що зберігаються на Google Диску, використовуйте наступний код:

```
from google.colab import drive
drive.mount('/content/drive')
```

- Після виконання цього коду вам буде надано посилання для авторизації. Перейдіть за посиланням, скопіюйте код і вставте його в поле в Colab.
- Ваш Google Диск буде змонтовано, і ви зможете звертатися до файлів за шляхом `/content/drive/My Drive/`.

### **Завантаження файлів з комп'ютера**

Якщо вам потрібно завантажити файли безпосередньо з комп'ютера, скористайтеся наступним кодом:

```
from google.colab import files
uploaded = files.upload()
```

- Після виконання цього коду відкриється вікно вибору файлів. Ви можете вибрати один або кілька файлів для завантаження.
- Завантажені файли будуть доступні в робочій директорії `/content/`.

Також є можливість просто перетягнути файл на панель зліва з провідника за допомогою миші.

### **Читання файлів з Google Диска**

Після підключення до Google Диска, ви можете працювати з файлами, як зазвичай, вказуючи повний шлях до файлу на диску:

```
import pandas as pd

# Приклад читання CSV-файлу з Google Диска
file_path = '/content/drive/My
Drive/path_to_your_file/aapl.csv'
df = pd.read_csv(file_path)
```

### **5. Збереження результатів на Google Диск**

Ви можете зберігати результати роботи або модифіковані файли назад на Google Диск:

```
df.to_csv('/content/drive/My
Drive/path_to_save/modified_aapl.csv')
```

## 4 БІБЛІОТЕКИ PYTHON ДЛЯ ВІЗУАЛІЗАЦІЇ ДАНИХ

### 4.1 Теоретичні відомості

#### Бібліотеки для візуалізації даних

Для виконання завдання необхідно використовувати такі бібліотеки:

- **Pandas:** Бібліотека для роботи з табличними даними. Вона надає зручні методи для обробки даних і підготовки їх до візуалізації.
- **Matplotlib:** Основна бібліотека Python для створення статичних, анімованих та інтерактивних графіків.
- **Seaborn:** Бібліотека для візуалізації даних на основі Matplotlib, яка дозволяє створювати більш складні та привабливі графіки з меншою кількістю коду.

#### Створення графіків

##### Лінійний графік (Line Plot)

Лінійний графік використовується для відображення даних у вигляді точок, з'єднаних лініями. Він ідеально підходить для показу зміни величини через певний період часу або іншу послідовність.

```
import matplotlib.pyplot as plt
import pandas as pd

# Приклад даних
data = pd.DataFrame({'x': [1, 2, 3, 4, 5], 'y': [1,
4, 9, 16, 25]})

# Побудова лінійного графіка
plt.plot(data['x'], data['y'])
plt.xlabel('X axis')
plt.ylabel('Y axis')
plt.title('Line Plot')
plt.show()
```

### Точковий графік (Scatter Plot)

Точковий графік використовується для відображення відношення між двома змінними. Кожна точка на графіку представляє значення двох змінних.

```
import seaborn as sns

# Побудова точкового графіка
sns.scatterplot(x='x', y='y', data=data)
plt.xlabel('X axis')
plt.ylabel('Y axis')
plt.title('Scatter Plot')
plt.show()
```

### Гістограма (Histogram)

Гістограма показує розподіл даних, представляючи частоту появи значень у певних інтервалах.

```
# Генерація випадкових даних
data = pd.Series([1, 2, 2, 3, 3, 3, 4, 4, 4, 4])

# Побудова гістограми
plt.hist(data, bins=4)
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.title('Histogram')
plt.show()
```

### Графік типу "ящик з вусами" (Box Plot)

Box Plot використовується для показу розподілу даних і виявлення потенційних викидів.

```
# Генерація випадкових даних
data = pd.DataFrame({'group': ['A', 'A', 'A', 'B', 'B', 'B'], 'value': [7, 8, 5, 6, 7, 8]})

# Побудова box plot
```

```
sns.boxplot(x='group', y='value', data=data)
plt.xlabel('Group')
plt.ylabel('Value')
plt.title('Box Plot')
plt.show()
```

### Скрипичний графік (Violin Plot)

Violin Plot поєднує в собі властивості Box Plot і гістограми. Він показує розподіл даних через товщину графіка.

```
# Побудова скрипичного графіка
sns.violinplot(x='group', y='value', data=data)
plt.xlabel('Group')
plt.ylabel('Value')
plt.title('Violin Plot')
plt.show()
```

### Маски для фільтрації даних та агрегатні функції

Для обробки даних перед візуалізацією можна використовувати **маски** (фільтрацію) та **агрегатні функції**.

#### Маски (Фільтрація даних)

Маски дозволяють вибрати підмножину даних, що відповідає певним умовам.

```
# Маска для вибору даних, де значення більше 6
mask = data['value'] > 6
filtered_data = data[mask]
```

#### Агрегатні функції

Агрегатні функції, такі як `mean`, `median`, `max`, використовуються для підсумовування або обчислення середніх значень даних.

```
mean_value = data['value'].mean() # Середнє значення
median_value = data['value'].median() # Медіана
max_value = data['value'].max() # Максимальне значення
```

## 4.2 Завдання до лабораторної роботи

1. Підготуйте довільні дані для візуалізації. Можливі джерела даних:

- <https://www.kaggle.com/datasets>
- <https://datasetsearch.research.google.com/>

2. Побудуйте мінімум 5 графіків:

лінійний графік

точковий графік

гістограму

графік типу "ящик з вусами"

скрипичний графік

Використовувати бібліотеки Pandas, Matplotlib та/або Seaborn.

Обов'язково використовувати маски для фільтрації даних та агрегатні функції (mean, median, max тощо).

## 4.3 Вказівки до виконання лабораторної роботи

Для завантаження ноутбука в Google Colab та роботи з ним, а також з файлами з локального комп'ютера використовуйте інструкцію із лабораторної роботи №2, 3.

## 5 ІНСТРУМЕНТ ДЛЯ ВІЗУАЛІЗАЦІЇ ДАНИХ TABLEAU

### 5.1 Теоретичні відомості

Tableau — це інструмент для візуалізації даних, що дозволяє створювати інтерактивні дашборди для аналізу інформації. Дашборд — це набір візуальних елементів (діаграм, карт, таблиць), які в одному вікні надають узагальнену інформацію з можливістю гнучкого налаштування.

Основні елементи дашборда

Карта

Карта відображає географічну інформацію, наприклад, розподіл даних по країнах, містах або регіонах. У Tableau карти створюються на основі географічних полів (наприклад, широта, довгота або назви місць). Карты дозволяють швидко виявляти тенденції у розподілі даних по територіях.

Стовпчаста діаграма (Bar Chart)

Стовпчасті діаграми ефективно показують порівняння між категоріями, ілюструючи значення різних показників. Такі діаграми легко інтерпретувати, що робить їх популярними для відображення фінансових, демографічних та інших даних.

Кругова діаграма (Pie Chart)

Кругові діаграми відображають частки кожної категорії в межах загального обсягу даних. Вони корисні для демонстрації пропорцій та часток, але їх застосування рекомендується лише при невеликій кількості категорій для уникнення перевантаження інформацією.

Таблиця

Таблиці показують детальні дані в табличній формі. Вони є зручними для демонстрації точних числових значень або переліку даних, які не завжди легко зобразити у вигляді діаграм.

Фільтри

Фільтри дозволяють користувачеві вибирати конкретні категорії або діапазони даних, щоб звужити обсяг інформації для аналізу. Це додає гнучкість до дашборду, дозволяючи користувачам зосередитися на певних аспектах даних.

Вимоги до дашборда

При оцінці дашборда важливо звернути увагу на:

- Повноту відображення інформації — усі ключові аспекти даних мають бути висвітлені.

- Легкість сприйняття — дашборд має бути зрозумілим і зручним для інтерпретації.
- Оформлення дашборда — важливо дотримуватись чіткої структури, зрозумілих підписів, узгодженої колірної гами.

## 5.2 Завдання до лабораторної роботи

Використовуючи один із запропонованих датасетів (<https://moodle.zp.edu.ua/mod/assign/view.php?id=129302>) або дані з інших відкритих джерел, створити дашборд в Tableau. Дашборд має містити:

- карту
  - столбчасту діаграму
  - кругову діаграму
  - таблицю
  - фільтри
- Оцінюється:
- повнота відображення інформації
  - легкість сприйняття
  - оформлення дашборда

## 5.3 Вказівки до виконання лабораторної роботи

### 1. Підготовка даних

Оберіть один із запропонованих датасетів або знайдіть інші дані з відкритих джерел (наприклад, Kaggle або Google Dataset Search).

Переконайтеся, що дані містять географічні поля (назви країн, міст або широта та довгота), щоб створити карту, та інші числові або категоріальні поля для побудови діаграм.

### 2. Імпорт даних у Tableau

Запустіть Tableau Desktop (або Tableau Public).

На початковому екрані оберіть Connect to Data та імпортуйте вибраний датасет (наприклад, у форматі Excel, CSV або Google Sheets).

Перевірте, чи коректно завантажені дані і чи правильно розпізнані типи полів (географічні, числові, текстові).

### 3. Створення основних елементів дашборда

Карта

Перетягніть географічне поле (наприклад, назву країни або координати) у область Rows або Columns.

Tableau автоматично створить карту. Додайте на карту інші показники, щоб проілюструвати розподіл даних.

В налаштуваннях карти виберіть відповідний тип кольорів для більшої зрозумілості.

Стовпчаста діаграма (Bar Chart)

Перетягніть категоріальне поле (наприклад, назви продуктів, регіони) в Rows і числове поле (наприклад, обсяги продажів) у Columns.

Оберіть тип візуалізації “Bar” у верхньому меню Marks для створення стовпчастої діаграми.

Кругова діаграма (Pie Chart)

Перетягніть потрібне числове поле в область Rows і категоріальне поле в Columns.

Виберіть тип діаграми Pie у меню Marks.

Перевірте, щоб кількість категорій була невеликою для уникнення перевантаження.

Таблиця

Створіть таблицю, розмістивши текстові та числові поля у Rows та Columns.

Перевірте налаштування форматування для зручності читання.

4. Додавання фільтрів

Щоб створити фільтри, перетягніть поле, яке хочете використовувати як фільтр, у розділ Filters.

Налаштуйте тип фільтру для більшої зручності: вибір окремих категорій, діапазон значень, мультिवибір тощо.

5. Створення дашборда

У головному меню Tableau оберіть Dashboard > New Dashboard.

Перетягніть на дашборд створені елементи (карту, діаграми, таблицю).

Налаштуйте розташування та розміри елементів для гармонійного вигляду.

Додайте фільтри на дашборд, щоб користувачі могли взаємодіяти з даними.

6. Оформлення дашборда

Налаштуйте кольорову гаму, щоб зробити дашборд зрозумілим і привабливим.

Додайте заголовки та підписи до елементів для полегшення сприйняття.

Переконайтеся, що текстові поля, підписи та легенди чіткі та видимі.

#### 7. Публікація дашборда на Tableau Public

У Tableau оберіть Server > Tableau Public > Save to Tableau Public.

Увійдіть у свій обліковий запис Tableau Public або створіть новий.

Після завантаження дашборда натисніть Share, скопіюйте отримане посилання та надішліть його як результат роботи.

#### 8. Перевірка дашборда

Перегляньте опублікований дашборд за посиланням.

Переконайтеся, що всі елементи відображаються коректно, фільтри працюють, а інформація легко сприймається.

## ЛІТЕРАТУРА

1. Microsoft: Analyzing and Visualizing Data with Excel. – Режим доступу: <https://www.edx.org/learn/excel/microsoft-analyzing-and-visualizing-data-with-excel>.
2. Хайгом Е. Essentials of Pricing Analytics: Tools and Implementation with Excel. – Taylor & Francis, 2020. – 280 с.
3. Lutz М. Programming Python: Powerful Object-Oriented Programming. – O'Reilly Media, 2011. – 1626 с.
4. Беппі П. Head First. Python. – Фабула, 2021. – 624 с.
5. Васильєв О. Програмування мовою Python. – Навчальна книга - Богдан, 2019. – 504 с.
6. Tutorials of visualizations by Lilla Rasztk. – Режим доступу: <https://public.tableau.com/app/profile/lilla.rasztk/viz/Tutorialsofvisualizations/Tutorial>
7. Райан Л. Visual Data Storytelling with Tableau. – Longman (Pearson Education), 2018. – 272 с.
8. Tableau Learning – Режим доступу: <https://www.tableau.com/learn>.