

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ДНІПРОВСЬКА ПОЛІТЕХНІКА»**

**ІНСТИТУТ ЕКОНОМІКИ
Фінансово-економічний факультет
Кафедра економіки та економічної кібернетики**

Аналіз даних з використанням Python

**Методичні рекомендації
до проведення практичних занять
за спеціальністю 051 Економіка
(спеціалізація “Інвестиційна економіка”)
для студентів всіх форм навчання**

**Дніпро
НТУ «ДП»
2024**

Аналіз даних з використанням Python. Методичні рекомендації до проведення практичних занять за спеціальністю 051 «Економіка» (спеціалізація «Інвестиційна економіка») для студентів всіх форм навчання / Антонюк О.П., Пістунов І.М. – Нац. техн. ун-т. «Дніпровська політехніка», каф. ЕЕК. – Д.: НТУ «ДП», 2024.– 14 с.

Автори:

Антонюк О.П., канд. екон. наук, доц.

Пістунов І.М., докт. техн. наук, проф.

Погоджено рішенням методичної комісії спеціальності 051 «Економіка»
(протокол № 1 від 30.08.2020 р.).

Відповідальний за випуск завідувачка кафедри економіки та економічної кібернетики, д-р екон. наук,
доц. О.Ю. Чуріканова

Зміст

1. Мета, предмет та завдання викладання дисципліни	4
2. Практичні роботи.....	4
2.1 Практична робота №1	4
<u>2.2 Практична робота №1</u>	10
<u>2.3 Практичні роботи №3, 4.....</u>	13

1. Мета, предмет та завдання викладання дисципліни

Мета дисципліни – формування компетентностей щодо застосування інструментарію мови програмування *Python* для аналізу, прогнозування та оптимізації явищ і процесів у фінансових системах.

Предметом дисципліни є методи і процеси обробки економічної інформації.

Завдання дисципліни полягають у вивченні моделей соціально-економічних явищ з погляду фундаментальних принципів і знань на основі розуміння основних напрямів розвитку економічної науки; знань теоретичних та методичних засад електронної економіки, вмінь виконувати аналітичні методи; навиків працювати з електронними платіжними системами та проводити прості фінансові операції

Результатами навчання мають бути вміння:

- Візуалізувати економічні процеси, системи, явища, використовуючи апарат математичного та комп'ютерного моделювання мови програмування *Python*.
- Розробляти моделі розвитку соціально-економічних процесів та систем з використанням мови програмування *Python*.
- Розуміти теоретичні засади електронної економіки та методів захисту інформації.

2. Практичні роботи

2.1 Практична робота №1

1. Необхідно зареєструватися на сайті <https://campus.datacamp.com>
2. Обираємо курс **Intermediate Python** за посиланням нижче:
<https://campus.datacamp.com/courses/intermediate-python/matplotlib?ex=1>

Необхідно прослухати першу лекцію та виконати завдання до неї. Для використання функцій для побудови графіків необхідно імпортувати бібліотеку `matplotlib.pyplot`.

За допомогою `matplotlib` ви можете створити різні види графіків у Python. Найпростіший графік - це лінійний графік. Приклад, побудови:

```
import matplotlib.pyplot as plt
plt.plot(x,y)
plt.show()
```

На відео ви вже бачили, наскільки за останні роки зросло населення світу. Чи буде тенденція продовжуватись? Світовий банк має оцінки населення світу за 1950 - 2100 роки. Роки завантажуються у вашому робочому просторі як список (*list*), що називається *year*, а відповідне населення - як список, що називається *pop*.

Цей курс стосується багатьох концепцій, які ви, можливо, забули, тому, якщо вам коли -небудь знадобиться швидке оновлення, завантажте шпаргалку [Python for data data](#) та тримайте її під рукою!

Завдання 1:

- 1) Виведіть на екран `print()` останній елемент із *year* та *pop* -списків, щоб побачити прогнозовану кількість населення на 2010 рік. Використовуйте дві функції `print()`.
- 2) Перш ніж почати, вам слід імпортувати `matplotlib.pyplot` як `plt`. `pyplot`-це підпакет `matplotlib`.
- 3) Використовуйте `plt.plot ()` для побудови лінії. *Year* слід відобразити по горизонтальній осі, *pop* - по вертикальній осі. Не забудьте закінчити з функцією `show()`, щоб фактично відобразити графік.

Подивіться на побудований графік, створений у попередній вправі. Виходячи з графіку, приблизно в якому році на цій планеті буде більше десяти мільярдів людських істот?

Лінійні графіки

Завдання 2 Line plot (3):

Дані зібрані у 2007 році. Для вас доступні два списки:

life_exp, який містить тривалість життя для кожної країни та

gdp_cap, що містить ВВП на душу населення (тобто на особу) для кожної країни, виражений у доларах США.

ВВП означає Валовий внутрішній продукт. В основному це розмір економіки країни поділений на кількість населення, і ви отримаєте ВВП на душу населення.

Роздрукуйте останній елемент як зі списку *gdp_cap*, так і зі списку *life_exp* - інформація по Зімбабве.

Побудуйте лінійну діаграму з *gdp_cap* на осі *x* та *life_exp* по осі *y*. Чи має сенс представляти ці дані на лінійному графіку?

Не забудьте закінчити командою *plt.show()*, щоб фактично відобразити графік.

Завдання 3 Scatter Plot (1)

Коли у вас є шкала часу уздовж горизонтальної осі, доцільно використовувати лінійний графік. Але в багатьох інших випадках, коли ви намагаєтесь оцінити, чи існує кореляція між двома змінними, наприклад, діаграма розсіювання є кращим вибором. Нижче наведено приклад того, як побудувати діаграму розсіювання.

```
import matplotlib.pyplot as plt
plt.scatter(x,y)
plt.show()
```

Давайте побудуємо графік *gdp_cap* від *life_exp*, дані про ВВП та тривалість життя для різних країн у 2007 році.

Змініть вид графіку у попередньому завданні на діаграму розсіювання.

Кореляція стане зрозумілою, якщо відобразити ВВП на душу населення в логарифмічній шкалі. Додайте рядок *plt.xscale('log')*. Завершіть свій скрипт за допомогою *plt.show()*, щоб відобразити графік.

Завдання 4 Scatter plot (2)

У попередньому завданні ви побачили, що вищий ВВП зазвичай відповідає більшій тривалості життя. Іншими словами, існує позитивна кореляція. Як ви вважаєте, чи існує зв'язок між населенням та тривалістю життя країни? Список *life_exp* з попередньої вправи вже доступний. Крім того, зараз також доступна

`pop` - список з переліком відповідних груп населення для країн у 2007 році. Населення представлено в мільйонах осіб.

Почніть: імпортуйте `matplotlib.pyplot` як `plt`.

Побудуйте діаграму розсіювання, де `pop` відображається на горизонтальній осі, а `life_exp` - на вертикальній осі.

Завершіть сценарій за допомогою `plt.show()`, щоб фактично відобразити графік. Чи існує кореляція?

Гістограми

Теоретичний матеріал за посиланням:

<https://campus.datacamp.com/courses/intermediate-python/matplotlib?ex=7>

Завдання 1 Build a histogram (1)

`life_exp` - список, що містить дані про тривалість життя для різних країн у 2007 році, доступний у вашій оболонці Python.

Щоб побачити, як розподіляється тривалість життя в різних країнах, створимо гістограму `life_exp`. `matplotlib.pyplot` уже доступний як `plt`.

Використовуйте `plt.hist()` для створення гістограми значень з `life_exp`. Не вказуйте кількість інтервалів; за замовчуванням Python встановить для вас кількість інтервалів = 10.

Додайте `plt.show()`, щоб фактично відобразити гістограму. Чи можете ви сказати, який інтервал містить найбільше спостережень?

Завдання 2 Build a histogram (2): bins

У попередній вправі ви не вказали кількість інтервалів. За замовчуванням у цьому випадку Python встановлює кількість інтервалів = 10. Кількість інтервалів є досить важливою. Занадто мало інтервалів занадто спростить реальність і не покаже вам подробиць. Занадто багато інтервалів ускладнить дійсність і не покаже загальної картини.

Щоб контролювати кількість інтервалів для поділу даних, ви можете встановити аргумент `bins`.

Це саме те, що необхідно виконати в цій вправі. Ви будете робити тут два графіка. Код у скрипті вже містить виклики `plt.show()` та `plt.clf()`; `plt.show()` відображає графік; `plt.clf()` очищає, щоб ви могли розпочати заново.

Як і раніше, `life_exp` доступний, а `matplotlib.pyplot` імпортовано як `plt`.

Побудуйте гістограму `life_exp` з 5 інтервалами. Чи можете ви сказати, який інтервал містить найбільше спостережень?

Створіть ще одну гістограму `life_exp`, цього разу з 20 інтервалами. На якій гістограмі представлено дані краще?

Завдання 3 Build a histogram (3): compare

На відео ви побачили гістограми населення на сьогоднішній день і на майбутнє. Оскільки ми використовували гістограму, було дуже легко провести порівняння.

Давайте зробимо подібне порівняння. Список `life_exp` містить дані про тривалість життя для різних країн у 2007 році. Ви також маєте доступ до другого списку, `life_exp1950`, який містить подібні дані за 1950 рік. Чи можете ви побудувати гістограми для обох наборів даних?

Необхідно побудувати 2 графіка. Команди `plt.show()` та `plt.clf()` для гарного рендерингу вже включені. Також для вас імпортується `matplotlib.pyplot`, як `plt`.

Побудуйте гістограму за даними `life_exp` з 15 інтервалами.

Побудуйте гістограму за даними `life_exp1950`, також із 15 інтервалами. Чи є велика відмінність між гістограми для даних 2007 і 1950 року?

Питання 1: Виберіть правильний вид графіку:

Ви професор, який викладає *Data Science з Python*, і вам потрібно візуально оцінити, чи оцінки на вашому іспиті відповідають певному розподілу. Який графік ви використаєте?

Варіанти відповіді

- 1) Лінійний графік

- 2) Діаграма розкиду
- 3) Гістограма

Питання 2: Виберіть правильний вид графіку:

Ви - професор аналітики даних на Python, і вам потрібно візуально оцінити, чи довші відповіді на екзаменаційні питання призводять до вищих оцінок. Який вид графіку ви використаєте?

Варіанти відповіді

- 1) Лінійний графік
- 2) Діаграма розкиду
- 3) Гістограма

Налаштування

Теоретичний матеріал за посиланням:

<https://campus.datacamp.com/courses/intermediate-python/matplotlib?ex=13>

2.2 Практична робота №2

Завдання 1 Мітки

Настав час налаштувати власний графік. Це найцікавіша частина, ви побачите, як ваш графік оживає!

Ви збираєтесь працювати над діаграмою розсіювання з даними світового розвитку: ВВП на душу населення на осі x (логарифмічна шкала), тривалість життя на осі y. Код для побудови цього графіку доступний у скрипті.

```
# Basic scatter plot, log scale
plt.scatter(gdp_cap, life_exp)
plt.xscale('log')
```

```
# Strings
xlabel = 'GDP per Capita [in USD]'
ylabel = 'Life Expectancy [in years]'
title = 'World Development in 2007'

# Add axis labels

# Add title

# After customizing, display the plot
```

На першому кроці, додамо мітки осі та заголовок до графіку. Ви можете зробити це за допомогою функцій *xlabel()*, *ylabel()* та *title()*, доступних у `matplotlib.pyplot`. Цей підпакет уже імпортовано як `plt`.

Рядки *xlab* і *ylab* уже встановлені для вас. Використовуйте ці змінні, щоб встановити мітку осі *x* і *y*.

Назва рядка також кодується для вас. Використовуйте його, щоб додати назву графіку.

Після цих налаштувань завершіть сценарій за допомогою `plt.show()`, щоб фактично відобразити графік.

Завдання 2 Ticks

Встановлювати *y*-позначки можна, вказавши два аргументи:

```
plt.yticks ([0,1,2], ["один", "два", "три"])
```

У цьому прикладі позначки, що відповідають числам 0, 1 і 2, будуть замінені на слова «один», «два» та «три» відповідно.

Давайте зробимо подібне для осі *x* вашої світової діаграми розвитку з функцією *xticks()*. Значення позначок 1000, 10000 та 100000 слід замінити на 1к, 10к та 100к. З цією метою для вас уже створено два списки: *tick_val* та *tick_lab*.

Використовуйте *tick_val* і *tick_lab* як вхідні дані для функції *xticks()*, щоб зробити графік більш читабельним.

Як завжди, після додавання налаштувань відобразіть сюжет за допомогою *plt.show()*.

Завдання 3 Sizes

Наразі діаграма розсіювання - це просто набір синіх точок, які не відрізняються одна від одної. Давайте змінимо це. Чи не було б добре, якби розмір крапок відповідав даним?

Для цього у вашій робочій області завантажується список *pop*. Він містить кількість населення кожної країни, виражену в мільйонах. Ви можете бачити, що цей список додається до методу розсіювання, як аргумент *s*, для розміру.

Запустіть скрипт, щоб побачити, як змінюється графік.

Виглядає добре, бо збільшення розміру бульбашок зробить дані виразнішими.

Імпортуйте пакет *numru* як *nr*.

Використовуйте *nr.array()*, щоб створити масив *numru* зі спливаючого списку. Викличте цей масив *Numru nr_pop*.

Подвойте значення в *nr_pop*, встановивши значення *nr_pop*, що дорівнює *nr_pop * 2*. Оскільки *nr_pop* - це масив *Numru*, кожен елемент масиву буде подвоєний.

Змініть аргумент *s* всередині *plt.scatter ()* на *nr_pop* замість *pop*.

Завдання 4 Colors

Наступний крок - зробити сюжет більш барвистим! Для цього для вас створено список *col*. Це список із кольором для кожної відповідної країни, залежно від континенту, до складу якого входить країна.

Побудовано словник, який відображає континенти на кольори:

```
dict = {
    'Asia': 'red',
    'Europe': 'green',
    'Africa': 'blue',
    'Americas': 'yellow',
    'Oceania': 'black'
}
```

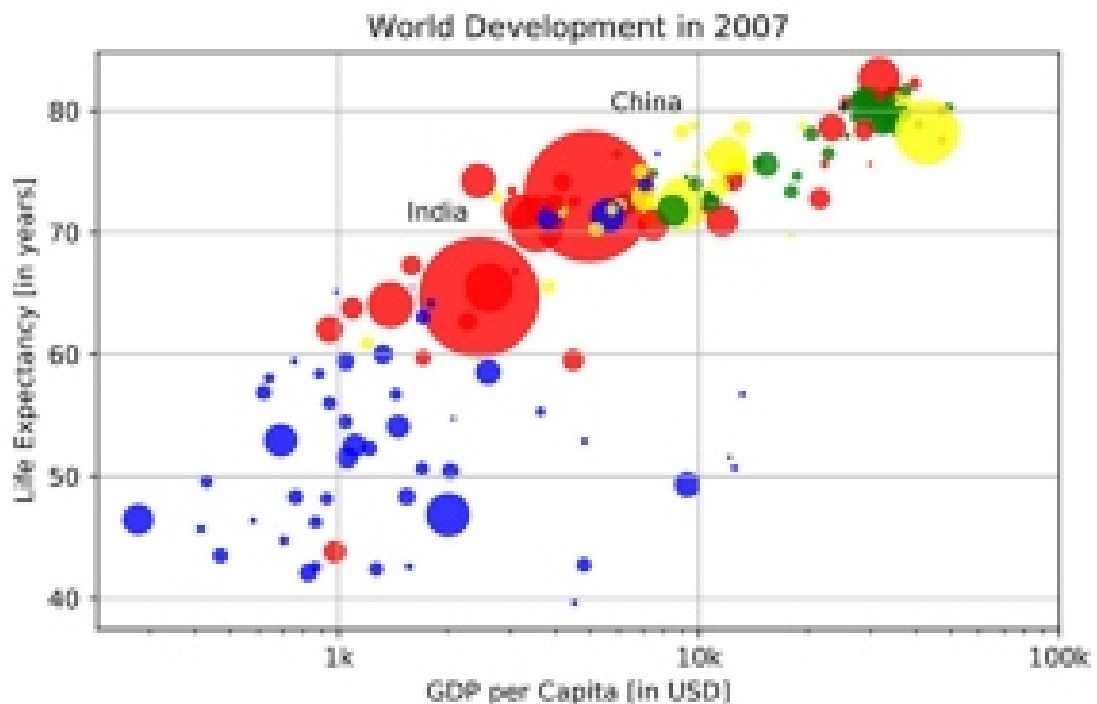
Додайте $c = col$ до аргументів функції `plt.scatter()`.

Змініть непрозорість бульбашок, встановивши для `alpha` -аргументу значення 0,8 усередині `plt.scatter()`. `Alpha` може бути встановлена від нуля до одиниці, де нуль повністю прозорий, а одиниця зовсім не прозора.

Завдання 5 Additional Customizations

Виведіть підписи найбільших даних за допомогою `plt.text()`. Додайте `plt.grid(True)` після `plt.text()`, щоб вивести написи India та China на графік.

```
plt.text(1550, 71, 'India')
plt.text(5700, 80, 'China')
```



2.3 Практичні роботи №3, 4

У цих роботах студенти повторюють завдання роти 1 та 2, але для даних, взятих з Інтернету у такому порядку:

1. Вибрати статистичну інформацію згідно номеру студента за списком групи за якнайбільший період. Конкретну таблицю в групі даних обирати довільно.

2. Ввести дані з сайту в програму та зберегти цю інформацію на диску у файлі Excel. Наприклад, застосовуючи таку групу команд:

```
import requests
import xlrd
import numpy as np
from statistics import mean
import matplotlib.pyplot as plt

#ссылка на таблицю на сайті
url1 = 'https://ukrstat.gov.ua/operativ/operativ2020/zf/zf_reg/zf_reg19_eu.xls'
#скачати файл за посиланням
def download(url=""):
    try:
        res = requests.get(url=url1)
        with open('tab.xls', 'wb') as file:
            file.write(res.content)
        return "завантажено"
    except Exception as _ex:
        return "напевно ви ввели ссылку з похибкою"
```

3. Для введених даних побудувати графіки згідно завдань 1 та 2.

4. Розроблені програми мають бути представлені як у текстовому форматі так і у форматі GUI.

№ за списком групи	Статистична інформація
1.	Ринок праці
2.	Освіта
3.	Охорона здоров'я
4.	Доходи та умови життя
5.	Соціальний захист
6.	Населені пункти та житло
7.	Правосуддя та злочинність
8.	Культура
9.	Макроекономічна статистика
10.	Національні рахунки
11.	Економічна діяльність
12.	Діяльність підприємств
13.	Внутрішня торгівля
14.	Капітальні інвестиції
15.	Основні засоби
16.	Сільське, лісове та рибне господарство
17.	Енергетика
18.	Промисловість
19.	Будівництво
20.	Транспорт
21.	Туризм
22.	Реєстр статистичних одиниць
23.	Навколишнє середовище
24.	Державні фінанси, податки та публічний сектор
25.	Зовнішньоекономічна діяльність
26.	Ціни
27.	Наука, технології та інновації
28.	Інформаційне суспільство