

ПРАКТИЧНА РОБОТА №7. ВІЗУАЛІЗАЦІЯ ДАНИХ ЗА ДОПОМОГОЮ МОДУЛЯ MATPLOTLIB

Мета роботи – вивчити та засвоїти навички роботи з базовим функціоналом модуля Matplotlib та mplot3d Toolkit. Ознайомитися з принципами та особливостями візуалізації даних шляхом побудови графіків, тривимірних графіків та засвоїти практичні навички їх використання.

Основні теоретичні відомості

Введення в Matplotlib

Matplotlib – бібліотека на мові програмування Python для візуалізації даних двовимірної (2D) графікою (3D графіка також підтримується). Одержувані зображення можуть бути використані в якості ілюстрацій в публікаціях.

Matplotlib написаний і підтримувався в основному Джоном Хантером (англ. John Hunter) і поширюється на умовах BSD-подібної ліцензії. Генеруються в різних форматах зображення можуть бути використані в інтерактивній графіці, в наукових публікаціях, графічному інтерфейсі користувача, веб-додатках, де потрібна побудова діаграм. Для установки використовуйте пакетний Matplotlib менеджер:

```
conda install matplotlib
```

або

```
pip install matplotlib
```

Імпорт модуля:

```
[5]: import matplotlib.pyplot as plt
```

Для відображення графіків в notebook введіть:

```
[6]: %matplotlib inline
```

Завдяки цьому всі графіки будуть виводитися в векторному форматі svg:

```
[7]: %config InlineBackend.figure_format = 'svg'
```

Для відображення графіків у відриві від jupyter notebook використовуйте `plt.show()` після всіх команд для виведення графіків.

API matplotlib в функціональному стилі

Matplotlib може будувати графіки зі звичайних списків або масивів numpy.

```
[10]: import numpy as np
x = np.linspace(0, 5, 11)
y = x**2
print(x,y,sep="\n\n")
```

[0. 0.5 1. 1.5 2. 2.5 3. 3.5 4. 4.5 5.]

[0. 0.25 1. 2.25 4. 6.25 9. 12.25 16. 20.25 25.]

Рисунок 7.1 – Створення масиву numpy

```
[11]: plt.plot(x, y, 'r') # r - red
plt.xlabel('Вісь X')
plt.ylabel('Вісь Y')
plt.title('Заголовок графіку')
plt.show();
```

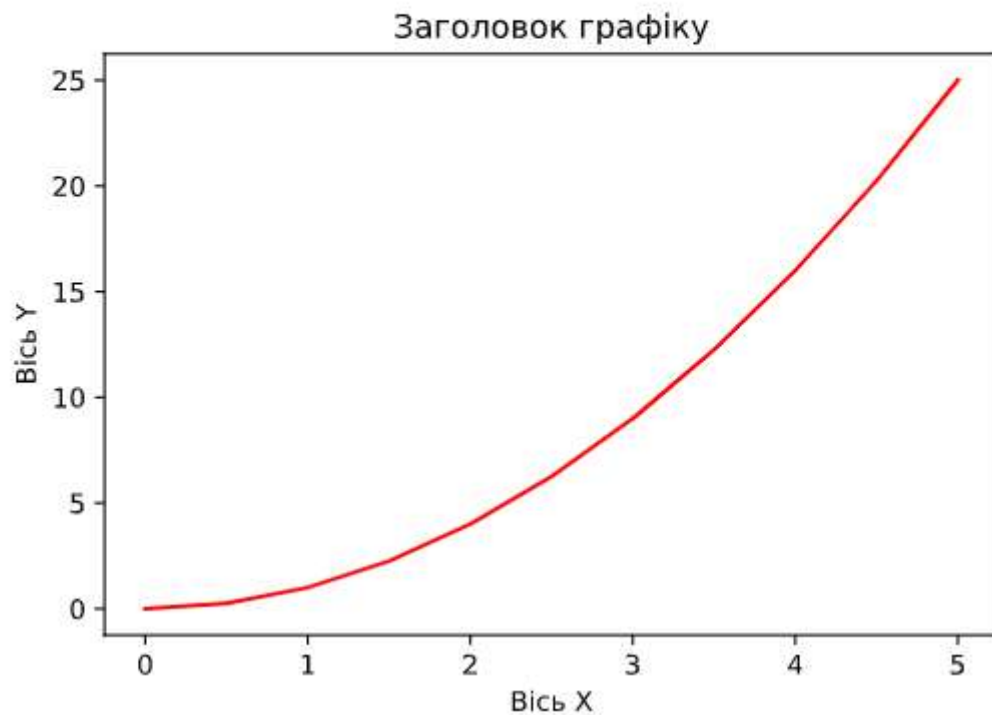


Рисунок 7.2 – Базові команди matplotlib

Кілька графіків на одному полотні:

```
# plt.subplot(nrows, ncols, plot_number)
plt.subplot(1, 2, 1)
plt.plot(x, y, 'y--') # "--" - вид лінії
plt.subplot(1, 2, 2)
plt.plot(y, x, 'b.-');# "-.-" - вид лінії
```

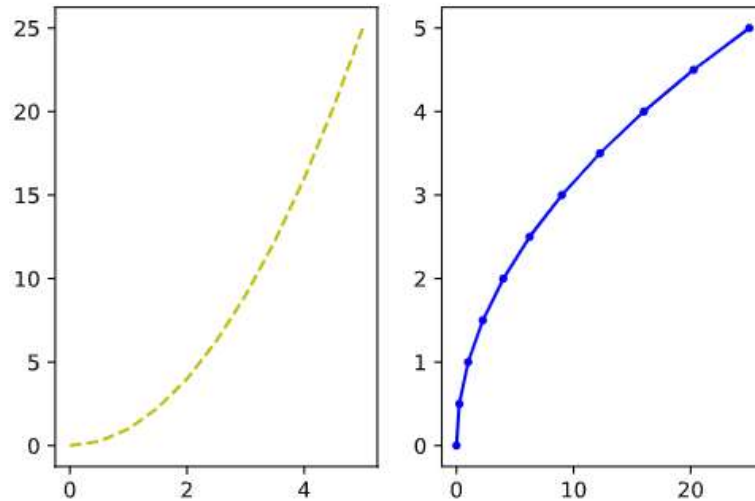


Рисунок 7.3 – Створення кількох графіків

Об'єктно-орієнтований стиль створення графіків:

```
[18]: # Create Figure (empty canvas)
fig = plt.figure()
# Add set of axes to figure
axes = fig.add_axes([0.1, 0.1, 0.8, 0.8]) # left, bottom, width, height (range 0 to 1)
# Plot on that set of axes
axes.plot(x, y, 'g->')
axes.set_xlabel('Вісь X') # Notice the use of set_ to begin methods
axes.set_ylabel('Вісь Y')
axes.set_title('Заголовок');
```

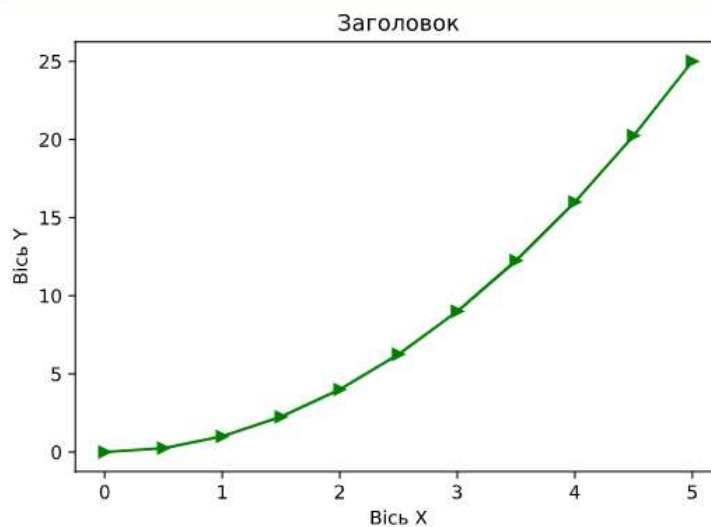


Рисунок 7.4 – Створення графіків з використанням стилю ООП

Тепер ми маємо повний контроль над вісями графіка:

```
[39]: # Створимо пусте полотно
fig = plt.figure()
# Вісі задаються через список из 4-х чисел: координати лівого нижнього угла, ширину та висоту.
axes1 = fig.add_axes([0.1, 0.1, 0.8, 0.8])
axes2 = fig.add_axes([0.2, 0.5, 0.4, 0.3])
axes1.plot(x, y, 'k')
axes1.set_xlabel('Підпис вісі X великого графіку')
axes1.set_ylabel('Підпис вісі Y великого графіку')
axes1.set_title('Заголовок великого графіку')

axes2.plot(y, x, 'p')
axes2.set_xlabel('Підпис вісі X\нмалого графіку')
axes2.set_ylabel('Підпис вісі Y\нмалого графіку')
axes2.set_title('Заголовок графіку');
```

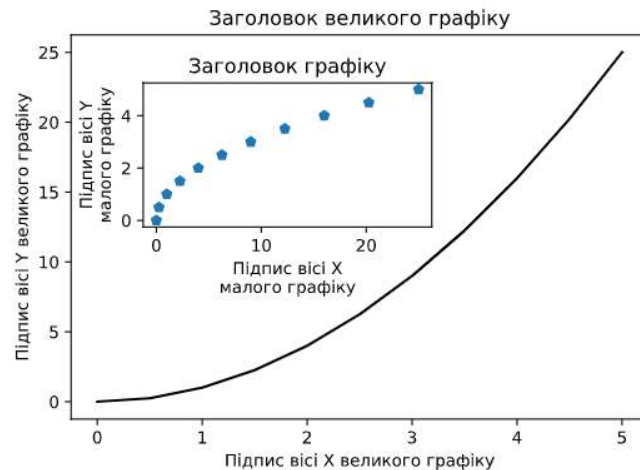


Рисунок 7.5 – Приклад створення графіків

`subplots()` – те ж, що і `plt.figure()`, тільки повертає відразу кортеж з фігурою:

```
[41]: fig, axes = plt.subplots()
axes.plot(x, y, 'ro')
axes.set_xlabel('x')
axes.set_ylabel('y')
axes.set_title('Заголовок')

[41]: Text(0.5, 1.0, 'Заголовок')
```

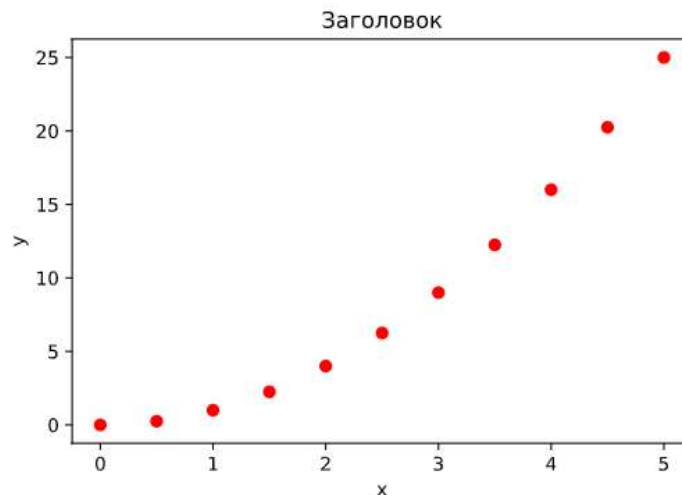


Рисунок 7.6 – Повернення кортежу з фігурою з використанням функції `subplots`

Якщо вказати кількість рядків і стовпців, можна створити кілька графіків.

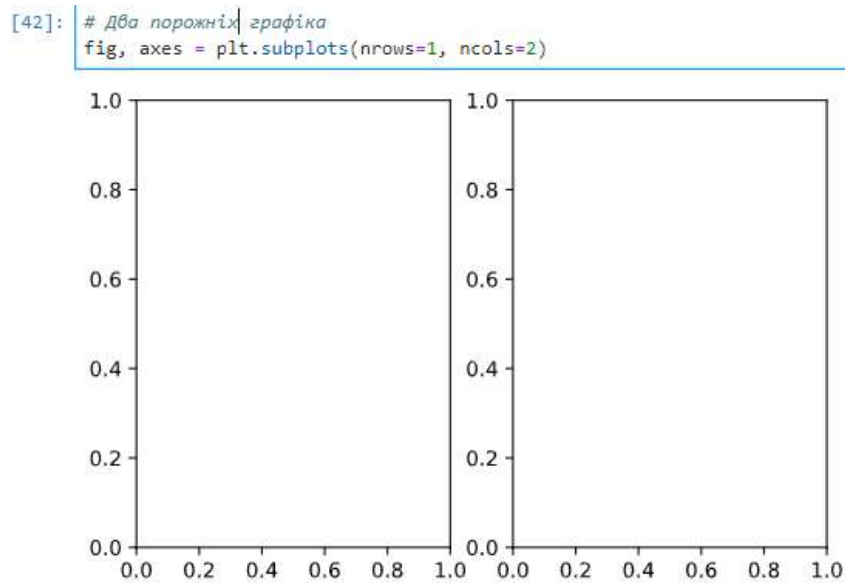


Рисунок 7.7 – Створення кількох графіків за допомогою функції subplots

Axes – це просто список осей « axes array ([,], dtype = object) » При побудові кількох графіків осі часто накладаються одна на одну. Для усунення цієї проблеми використовуйте методи `fig.tight_layout()` або `plt.tight_layout()`:

```
[43]: fig, axes = plt.subplots(nrows=1, ncols=2)
for ax in axes:
    ax.plot(x, y, 'gx')
    ax.set_xlabel('x')
    ax.set_ylabel('y')
    ax.set_title('Заголовок')
fig
plt.tight_layout()
```

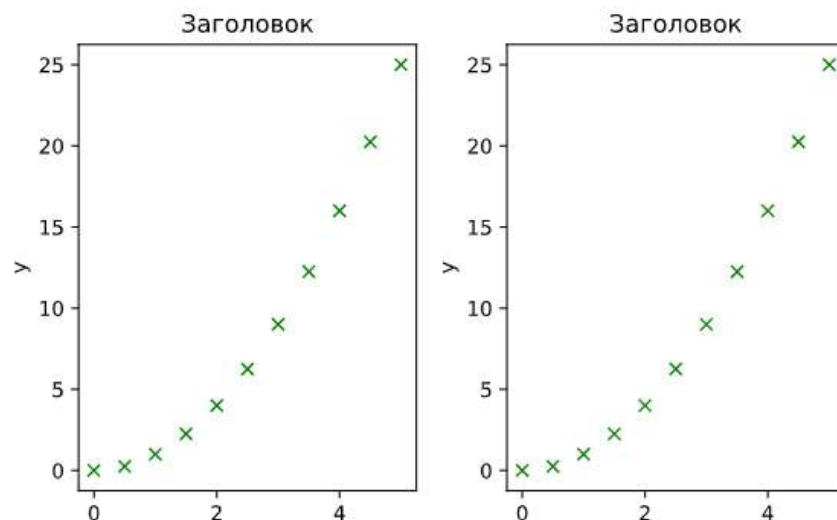


Рисунок 7.8 – Перебір циклом змінної axes для задання значень кожному графіку

Matplotlib дозволяє задавати розмір сторін графіка і DPI при створенні фігури. Для цього використовуйте наступні іменовані аргументи:

- **figsize** приймає кортеж і двох чисел - ширини і висоти графіка в дюймах;
- **dpi** - dots-per-inch.

Наприклад:

```
[44]: fig = plt.figure(figsize=(8,4), dpi=100)  
<Figure size 800x400 with 0 Axes>
```

Рисунок 7.9 – Використання функції figure

Можна зберігати графіки в форматах PNG, JPG, EPS, SVG, PGF і PDF. Для збереження графіка треба використовувати метод `savefig` класу `Figure`:

```
[ ]: fig.savefig("filename.png", dpi=200)|
```

Рисунок 7.10 – Використання функції `savefig` для збереження графіка

Легенди, підписи осей, заголовки

Заголовки графіків:

```
[ ]: ax.set_title("Заголовок");|
```

Рисунок 7.11 – Задання заголовку графіка

Підписи осей:

```
[48]: ax.set_xlabel("x")  
      ax.set_ylabel("y");
```

Рисунок 7.12 – Задання підписів осей

Легенди. При побудові графіків можна вказувати аргумент `label = "label text"`, і зазначені в ньому підписи будуть відображатися як підписи легенд і використанні методу `legend`:

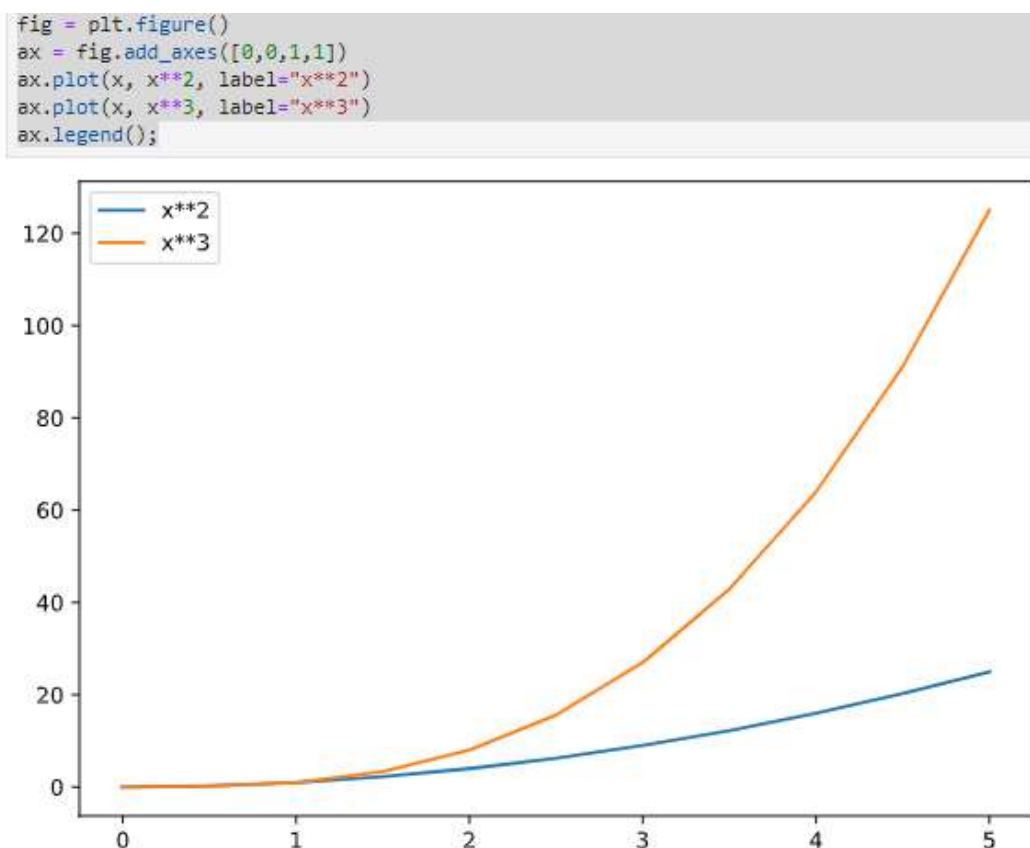


Рисунок 7.13 – Приклад вигляду легенд на графіку

Легенду можна розташовувати в різних місцях графіків:

```
[ ]: ax.legend(loc=1) # upper right
      ax.legend(loc=2) # upper left
      ax.legend(loc=3) # lower left
      ax.legend(loc=4) # lower right
      ax.legend(loc=0) # оптимальне розташування
```

Рисунок 7.14 – Демонстрація різних варіантів розташування легенди

Установка кольору, товщини і типу ліній

Стиль MatLab:

```
[53]: fig, ax = plt.subplots()
      ax.plot(x, x**2, 'b.-') # Синя лінія, точка пунктир
      ax.plot(x, x**3, 'g--'); # Зелена пунктирна лінія
```

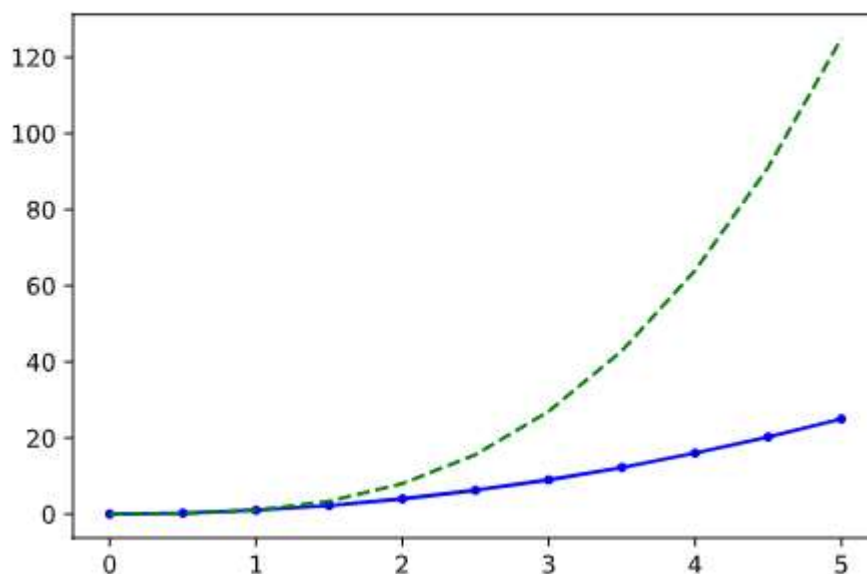


Рисунок 7.15 – Встановлення кольору для двох ліній

```
fig, ax = plt.subplots()
ax.plot(x, x+1, color="blue", alpha=0.5) # напівпрозора лінія
ax.plot(x, x+2, color="#8B008B")       # RGB
ax.plot(x, x+3, color="#FF8C00");       # RGB
```

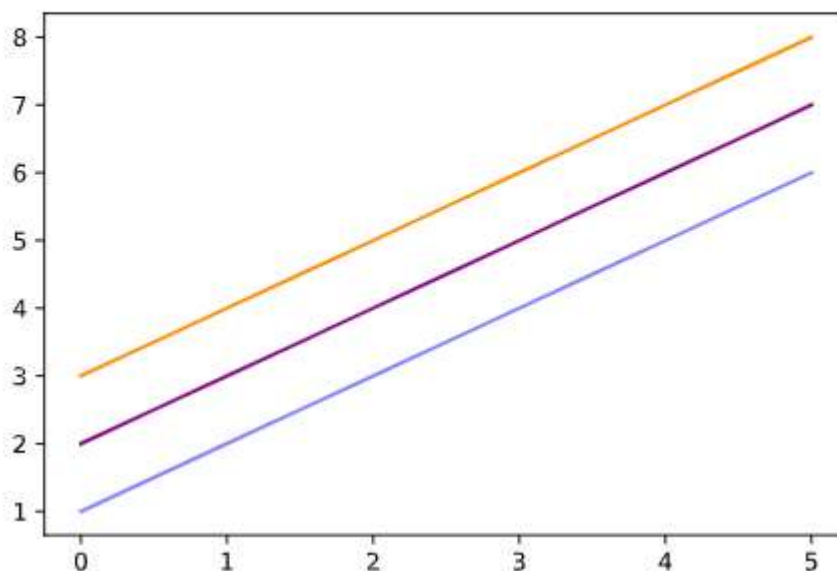


Рисунок 7.16 – Зазначення кольорів в параметрі color = і alpha:

Для зміни товщини лінії використовується ключове слово linewidth або lw.
Стиль лінії задається аргументами linestyle або ls:


```

fig, ax = plt.subplots(figsize=(12,6))
ax.plot(x, x+1, color="red", linewidth=0.25)
ax.plot(x, x+2, color="red", linewidth=0.50)
ax.plot(x, x+3, color="red", linewidth=1.00)
ax.plot(x, x+4, color="red", linewidth=2.00)
# Можливі варіанти '-', '-.-', '-.-.', ':', 'Steps'
ax.plot(x, x+5, color="green", lw=3, linestyle='-.-')
ax.plot(x, x+6, color="green", lw=3, ls='-.-.')
ax.plot(x, x+7, color="green", lw=3, ls=':')
# Кастомна пунктирна лінія
line, = ax.plot(x, x+8, color="black", lw=1.50)
line.set_dashes([5, 10, 15, 10]) # формат: довжина лінія, довжина пропуску, ...
# Можливі маркери: marker = '+', 'o', '*', 's', ',', '.', '1', '2', '3', '4', ...
ax.plot(x, x+9, color="blue", lw=3, ls='-', marker='+')
ax.plot(x, x+10, color="blue", lw=3, ls='--', marker='o')
ax.plot(x, x+11, color="blue", lw=3, ls='-', marker='s')
ax.plot(x, x+12, color="blue", lw=3, ls='--', marker='1')
# Розмір і колір маркерів
ax.plot(x, x+13, color="purple", lw=1, ls='-', marker='o', markersize=2)
ax.plot(x, x+14, color="purple", lw=1, ls='-', marker='o', markersize=4)
ax.plot(x, x+15, color="purple", lw=1, ls='-', marker='o', markersize=8, markerfacecolor="red")
ax.plot(x, x+16, color="purple", lw=1, ls='-', marker='s', markersize=8,
        markerfacecolor="yellow", markeredgewidth=3, markeredgewidth="green");

```

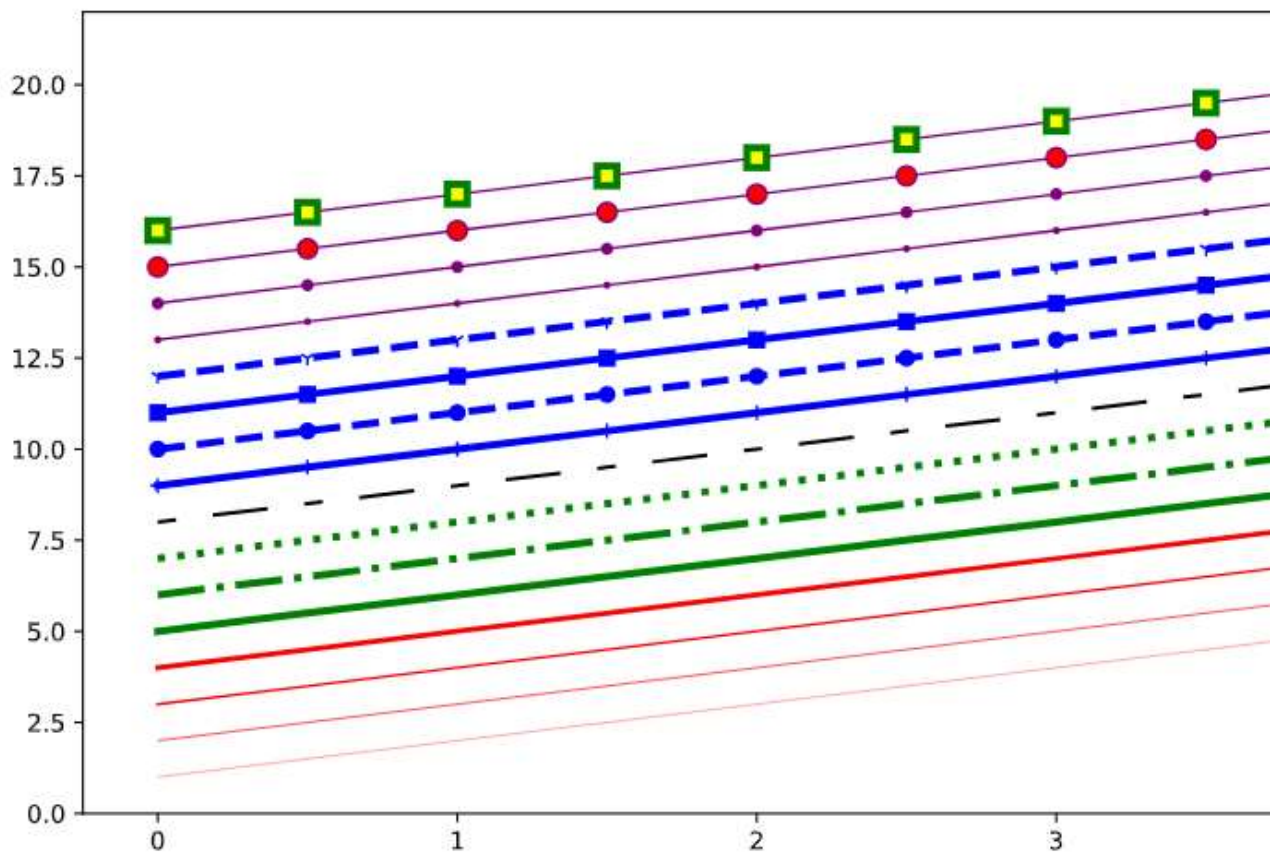


Рисунок 7.17 – Задання різних стилей для ліній

Діапазон значень графіка

Діапазон можна змінювати за допомогою методів `set_ylim`, `set_xlim` і `axis` ('tight'):

```
fig, axes = plt.subplots(1, 3, figsize=(12, 4))
axes[0].plot(x, x**2, x, x**3)
axes[0].set_title("Стандартний діапазон осей")
axes[1].plot(x, x**2, x, x**3)
axes[1].axis('tight')
axes[1].set_title("«Тісні» осі")
axes[2].plot(x, x**2, x, x**3)
axes[2].set_ylim([0, 60])
axes[2].set_xlim([2, 5])
axes[2].set_title("Заданий точними значеннями\ndіапазон осей");
```

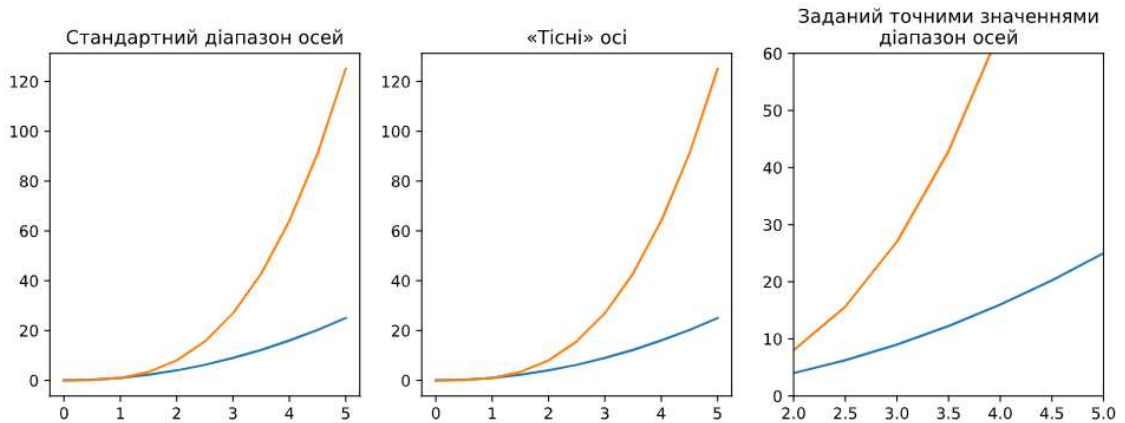


Рисунок 7.18 – Використання зміни діапазону

Інші типи графіків

За допомогою `matplotlib` можна створювати барплоти, гістограми, точкові діаграми і багато ще чого:

```
[57]: plt.scatter(x,y)
[57]: <matplotlib.collections.PathCollection at 0x1a107a50f48>
```

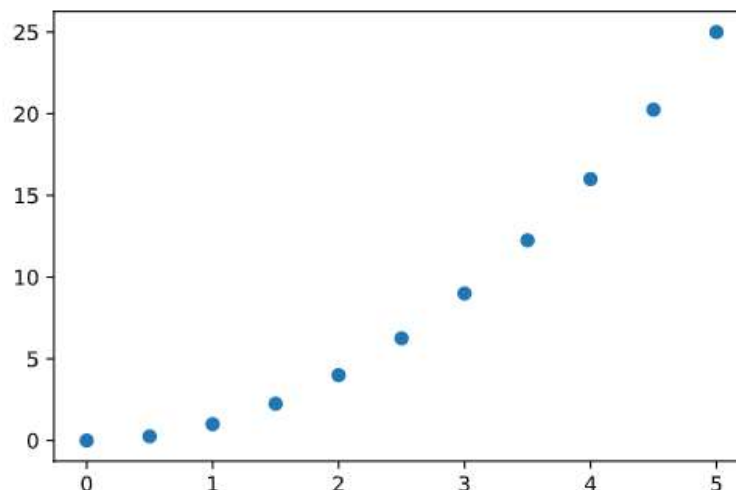


Рисунок 7.19 – Створення точкового графіку

```
[60]: fruits = ["apple", "peach", "orange", "bannana", "melon"]
      counts = [34, 25, 43, 31, 17]
      plt.bar(fruits, counts)
      plt.title("Fruits!")
      plt.xlabel("Fruit")
      plt.ylabel("Count")
```

```
[60]: Text(0, 0.5, 'Count')
```

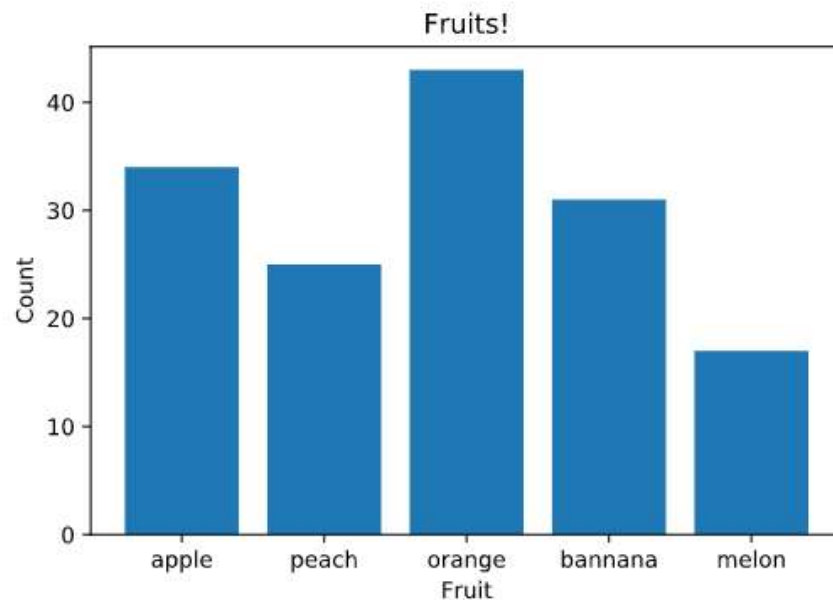


Рисунок 7.20 – Створення стовпчастої діаграми

```
[59]: data = [np.random.normal(0, std, 100) for std in range(1, 4)]

# rectangular box plot
plt.boxplot(data, vert=True, patch_artist=True);
```

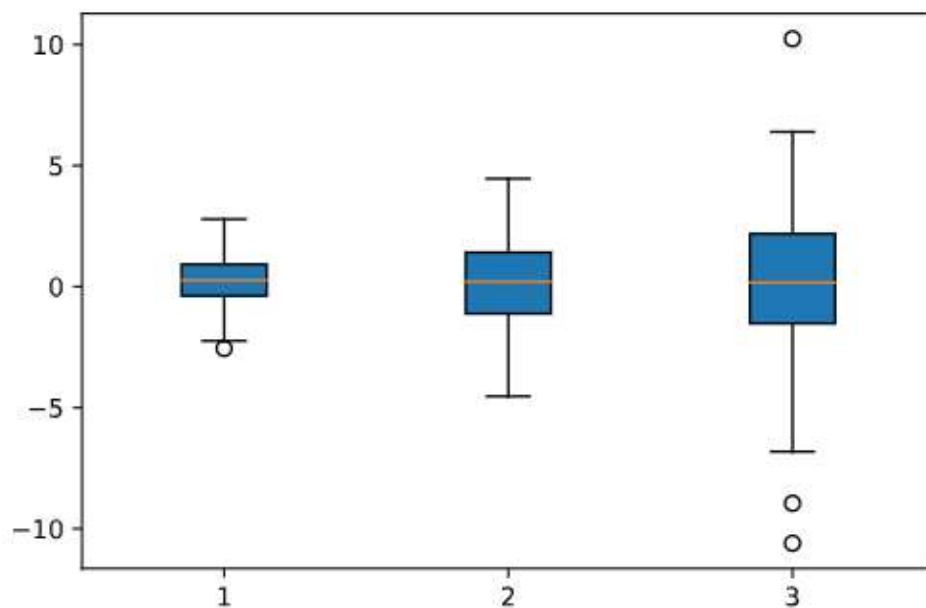


Рисунок 7.21 – Створення коробчатого графіку

```
[63]: import matplotlib as mpl
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
import datetime as dt
import csv
data_names = ['Київ', 'Харків', 'Суми', 'Донецьк',
              'Одесса', 'Чернівці', 'Полтава', 'Херсон',
              'Кривий ріг', 'Львів']
data_values = [1076, 779, 222, 189, 1137, 134, 124, 124, 91, 789]
dpi = 80
fig = plt.figure(dpi = dpi, figsize = (512 / dpi, 384 / dpi) )
mpl.rcParams.update({'font.size': 9})
plt.title('Розподіл кафе по городах України (%)')
xs = range(len(data_names))
plt.pie(
    data_values, autopct='%1f', radius = 1.1,
    explode = [0.15] + [0 for _ in range(len(data_names) - 1)] )
plt.legend(
    bbox_to_anchor = (-0.16, 0.45, 0.25, 0.25),
    loc = 'lower left', labels = data_names )
```

[63]: <matplotlib.legend.Legend at 0x1a10901abc8>



Рисунок 7.22 – Створення кругової діаграми

Побудова 3D графіків

До цього моменту всі графіки, які ми будували були двовимірні, Matplotlib дозволяє будувати 3D графіки.

- Лінійний 3D-графік;

- Точковий 3D-графік;
- Каркасна поверхня;
- Поверхня.

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D as ax3
```

Рисунок 7.23 – Імпорт необхідних бібліотек для роботи з 3D

У бібліотеці доступні інструменти для побудови різних типів графіків. Розглянемо деякі з них більш детально.

Лінійний графік

Для побудови лінійного графіка використовується функція `plot()`.

*`Axes3D.plot(self, xs, ys, *args, zdir = 'z', **kwargs)`*

- `xs`: 1D масив (x координати);
- `ys`: 1D масив (y координати);
- `zs`: скалярне значення або 1D масив (z координати). Якщо переданий скаляр, то він буде присвоєний всім точкам графіка;
- `zdir`: {'x', 'y', 'z'} - Визначає вісь, яка буде прийнята за z напрямом, значення за замовчуванням: 'z';
- `**kwargs` - Додаткові аргументи, аналогічні тим, що використовуються в функції `plot()` для побудови двовимірних графіків.


```
[26]: x = np.linspace(-np.pi, np.pi, 50)
y = x
z = np.cos(x)
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot(x, y, z)
ax.set_title("параметрична крива")
ax.set_xlabel("X")
ax.set_ylabel("Y")
ax.set_zlabel("Z")
fig.savefig("MyFile.png", dpi=800)
```

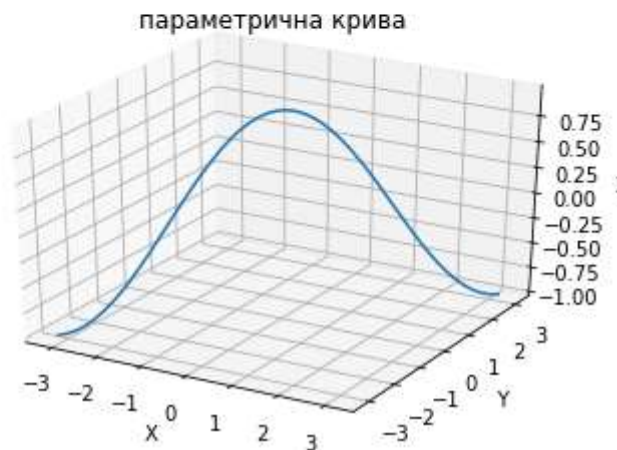


Рисунок 7.24 – Приклад побудови 3D лінійного графіку

Точковий графік

Для побудови точкового графіка використовується функція `scatter()`.

*Axes3D.scatter (self, xs, ys, zs = 0, zdir = 'z', s = 20, c = None, depthshade = True, * args, ** kwargs)*

- **xs, ys:** масив. Координати точок по осях x і y ;
- **zs:** float або масив, optional. Координати точок по осі z . Якщо переданий скаляр, то він буде присвоєний всім точкам графіка. Значення за замовчуванням: 0;
- **zdir:** { 'x', 'y', 'z', '-x', '-y', '-z' }, optional. Визначає вісь, яка буде прийнята за z напрямком, значення за замовчуванням: 'z';
- **s:** скаляр або масив, optional. Розмір маркера. Значення за замовчуванням: 20;
- **c:** color, масив, масив значень кольору, optional. Колір маркера. Можливі значення: Строкове значення кольору для всіх маркерів. Масив строкових

значень кольору. Масив чисел, які можуть бути відображені як кольори через функції `star` і `norm`;

- 2D масив, елементами якого є RGB або RGBA;
- ****kwargs** - Додаткові аргументи, аналогічні тим, що використовуються в функції `scatter()` для побудови двовимірних графіків.

```
[31]: np.random.seed(123)
x = np.random.randint(-5, 5, 40)
y = np.random.randint(0, 10, 40)
z = np.random.randint(-5, 5, 40)
s = np.random.randint(10, 100, 20)
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.scatter(x, y, z, s=s)
ax.set_xlabel("X")
ax.set_ylabel("Y")
ax.set_zlabel("Z")
fig.savefig("MyFile.png", dpi=800)
```

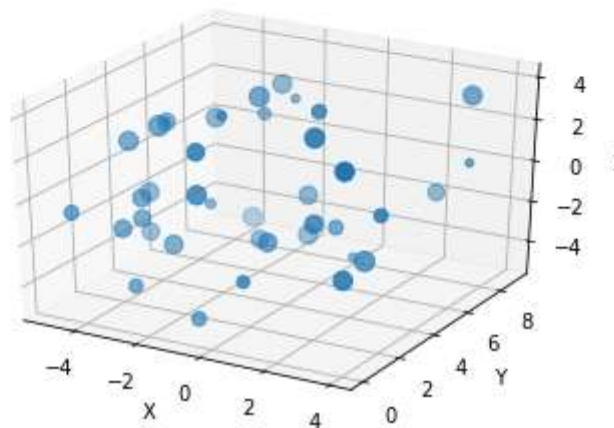


Рисунок 7.25 – Приклад побудови 3D точкового графіку

Каркасна поверхня

Для побудови каркасної поверхні використовується функція `plot_wireframe()`.

*plot_wireframe(self, X, Y, Z, *args, **kwargs)*

- X, Y, Z: 2D масиви. Дані для побудови поверхні;
- rcount, scount: int - максимальна кількість елементів каркаса, яке буде використано в кожному з напрямків. Значення за замовчуванням: 50;

- `rstride, cstride: int` - ці установки впливають на величину кроку, з яким будуть братися елементи рядка/стовпця з переданих масивів. Параметри `rstride, cstride` і `rcount, ccount` є взаємовиключними;
- `** kwargs` - додаткові аргументи, які визначаються `Line3DCollection`.

```
[34]: u, v = np.mgrid[0:2*np.pi:20j, 0:np.pi:10j]
x = np.cos(u)*np.sin(v)
y = np.sin(u)*np.sin(v)
z = np.cos(v)
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot_wireframe(x, y, z)
ax.set_xlabel("X")
ax.set_ylabel("Y")
ax.set_zlabel("Z")
fig.savefig("MyFile.png", dpi=800)
```

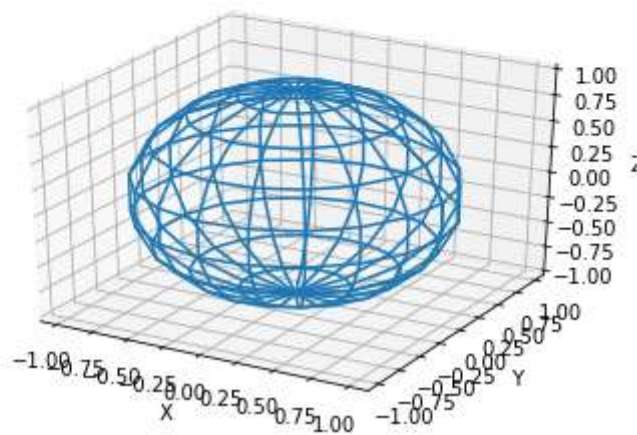


Рисунок 7.26 – Приклад побудови каркасної поверхні

Поверхня

Для побудови поверхні використовуйте функцію `plot_surface()`.

*`plot_surface(self, X, Y, Z, * args, norm = None, vmin = None, vmax = None, lightsource = None, ** kwargs)`*

- `X, Y, Z`: 2D масиви. Дані для побудови поверхні;
- `rcount, ccount: int` - див. `rcount, ccount` в "Каркасна поверхня";
- `rstride, cstride: int` - див. `rstride, cstride` в "Каркасна поверхня";
- `color: color` - колір для елементів поверхні;
- `cmap: Colormap` - `Colormap` для елементів поверхні;

- `facecolors`: масив елементів `color` - індивідуальний колір для кожного елемента поверхні;
- `norm`: `Normalize` - нормалізація для `colormap`;
- `vmin`, `vmax`: `float` - межі нормалізації;
- `shade`: `bool` - використання тіні для `facecolors`. Значення за замовчуванням: `True`;
- `lightsource`: `LightSource` - об'єкт класу `LightSource` - визначає джерело світла, використовується, тільки якщо `shade = True`;
- `**kwargs` - додаткові аргументи, які визначаються `Poly3DCollection`.

```
[36]: u, v = np.mgrid[0:2*np.pi:20j, 0:np.pi:10j]
      x = np.cos(u)*np.sin(v)
      y = np.sin(u)*np.sin(v)
      z = np.cos(v)
      fig = plt.figure()
      ax = fig.add_subplot(111, projection='3d')
      ax.plot_surface(x, y, z, cmap='inferno')
      fig.savefig("MyFile.png", dpi=800)
```

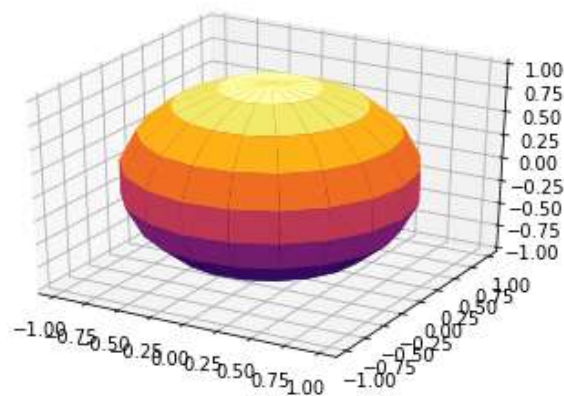


Рисунок 7.27 – Приклад побудови поверхні

Завдання до виконання

Використовуючи засоби matplotlib побудувати графіки відповідно до таблиці варіантів (таб. 7.1).

Таблиця 7.1 – Варіанти завдань

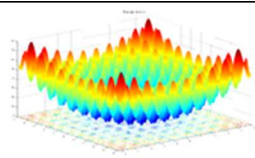
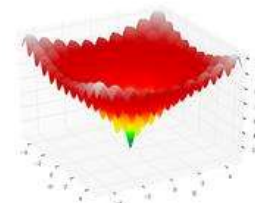
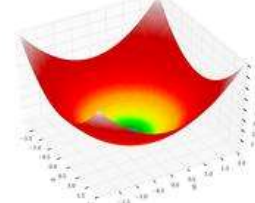
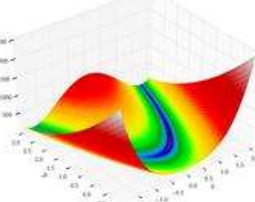
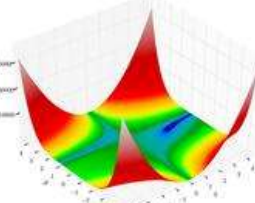
№	Завдання
1	Побудувати графік функції $y = 5x^2 - 2x + 8$ тип лінії – “-”, розташування легенди (ліво низ) діапазон значень – y (-10:50) x (-25:50), колір лінії - #80AC55 , прозорість лінії – 69%
	Побудувати кругову діаграму (не менше 10 сегментів) "Продажі автомобілів в Україні за 2019 рік", дані можна взяти довільні або знайти в мережі Інтернет. У діаграмі передбачити назви сегментів в поле легенд.
2	Побудувати графік функції $y = 5x^3 - 4$ тип лінії – “.”, розташування легенди (ліво верх) діапазон значень – y (5:80) x (-40:60), колір лінії - #225533 , прозорість лінії – 76%
	Побудувати стовпчикову діаграму (не менше 10 стовпчиків) "Рівень продажу гречки за останні 10 місяців", дані можна взяти довільні або знайти в мережі Інтернет. У діаграмі передбачити назви стовпчиків. Колір використовувати як для попереднього графіку.
3	Побудувати графік функції $y = 6x^{1/2} + 8$ тип лінії – “-”, розташування легенди (право низ) діапазон значень – y (10:70) x (-15:55), колір лінії - #8EC0BC , прозорість лінії – 41%
	Побудувати кругову діаграму (не менше 10 сегментів) "Популяція різновидів їжаків", дані можна взяти довільні або знайти в мережі Інтернет. У діаграмі передбачити назви сегментів в поле легенд.
4	Побудувати графік функції $y = x/2 + 7$ тип лінії – “-”, розташування легенди (право верх) діапазон значень – y (-5:35) x (-60:60), колір лінії - #FF0B0B , прозорість лінії – 36%
	Побудувати стовпчикову діаграму (не менше 10 стовпчиків) "Температура води Азовського моря за останні 10 місяців", дані можна взяти довільні або знайти в мережі Інтернет. У діаграмі передбачити назви стовпчиків. Колір використовувати як для попереднього графіку
5	Побудувати графік функції $y = 4e^x - 9$ тип лінії – “.”, розташування легенди (ліво низ) діапазон значень – y (-10:100) x (-35:40), колір лінії - #F0F0F0 , прозорість лінії – 91%
	Побудувати кругову діаграму (не менше 10 сегментів) "Кількість постраждалих від пандемії по місяцях", дані можна взяти довільні або знайти в мережі Інтернет. У діаграмі передбачити назви сегментів в поле легенд.
6	Побудувати графік функції $y = \ln x + 4$ тип лінії – “-”, розташування легенди (ліво верх) діапазон значень – y (-15:100) x (-45:60), колір лінії - #0F0F0F , прозорість лінії – 88%
	Побудувати стовпчикову діаграму (не менше 10 стовпчиків) "Рівень атмосферного тиску за останні 10 днів", дані можна взяти довільні або знайти в мережі Інтернет. У діаграмі передбачити назви стовпчиків. Колір використовувати як для попереднього графіку
7	Побудувати графік функції $\log x + 6$ тип лінії – “-”, розташування легенди (право низ) діапазон значень – y (-5:100) x (-25:55), колір лінії - #E8E0BD , прозорість лінії – 76%

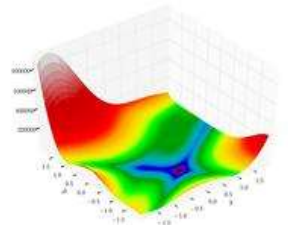
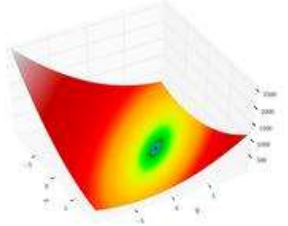
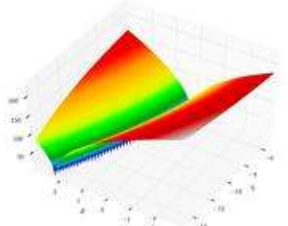
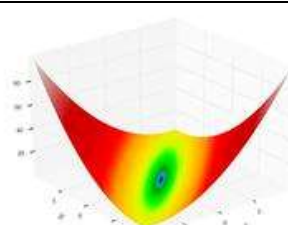
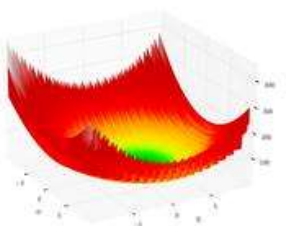
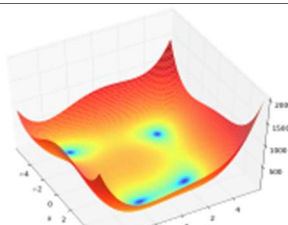
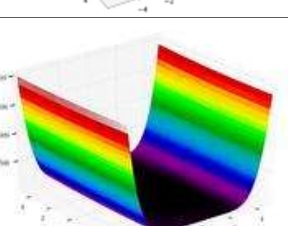
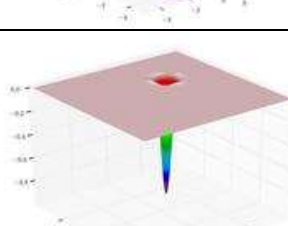
№	Завдання
	Побудувати кругову діаграму (не менше 10 сегментів) "Кількість проданого туалетного паперу фірмами за 1й квартал", дані можна взяти довільні або знайти в мережі Інтернет. У діаграмі передбачити назви сегментів в поле легенд.
8	<p>Побудувати графік функції $y = 2\cos x - 2$ тип лінії – "--", розташування легенди (право верх) діапазон значень – $y (-10:10) \times (-40:40)$, колір лінії - #80F0AB , прозорість лінії – 54%</p> <p>Побудувати стовпчикову діаграму (не менше 10 стовпчиків) "Кількість пожеж за 2019 рік", дані можна взяти довільні або знайти в мережі Інтернет. У діаграмі передбачити назви стовпчиків. Колір використовувати як для попереднього графіку</p>
9	<p>Побудувати графік функції $y = 3\sin x + 2$ тип лінії – ".", розташування легенди (ліво низ) діапазон значень – $y (5:75) \times (-30:30)$, колір лінії - #00CFFC , прозорість лінії – 68%</p> <p>Побудувати кругову діаграму (не менше 10 сегментів) "Кількість проданих смартфонів компаніями за 2019", дані можна взяти довільні або знайти в мережі Інтернет. У діаграмі передбачити назви сегментів в поле легенд.</p>
10	<p>Побудувати графік функції $y = \operatorname{tg} x - 4$ тип лінії – "-.", розташування легенди (ліво верх) діапазон значень – $y (10:80) \times (-60:35)$, колір лінії - #55AABB , прозорість лінії – 70%</p> <p>Побудувати стовпчикову діаграму (не менше 10 стовпчиків) "Рівень привабливості котів різних порід згідно голосування", дані можна взяти довільні або знайти в мережі Інтернет. У діаграмі передбачити назви стовпчиків. Колір використовувати як для попереднього графіку</p>
11	<p>Побудувати графік функції $y = \frac{1}{2}\operatorname{ctg} x + 7$ тип лінії – "--", розташування легенди (право низ) діапазон значень – $y (15:65) \times (-50:50)$, колір лінії - #8F0FBF , прозорість лінії – 90%</p> <p>Побудувати кругову діаграму (не менше 10 сегментів) "загальний обіг різновидів печива магазину «Вацак» за поточний квартал", дані можна взяти довільні або знайти в мережі Інтернет. У діаграмі передбачити назви сегментів в поле легенд.</p>
12	<p>Побудувати графік функції $y = 5e^x + 8$ тип лінії – "-", розташування легенди (право верх) діапазон значень – $y (-20:50) \times (-55:45)$, колір лінії - #F8F0FB , прозорість лінії – 80%</p> <p>Побудувати стовпчикову діаграму (не менше 10 стовпчиків) "Популяція горобців по регіонах проживання", дані можна взяти довільні або знайти в мережі Інтернет. У діаграмі передбачити назви стовпчиків. Колір використовувати як для попереднього графіку</p>
13	<p>Побудувати графік функції $y = \ln x - 8$ тип лінії – ".", розташування легенди (право верх) діапазон значень – $y (20:55) \times (-30:55)$, колір лінії - #55FFBB , прозорість лінії – 75%</p> <p>Побудувати кругову діаграму (не менше 10 сегментів) "Зареєстровані порушення ПДД у 2019", дані можна взяти довільні або знайти в мережі Інтернет. У діаграмі передбачити назви сегментів в поле легенд.</p>
14	<p>Побудувати графік функції $y = x/5 + 17$ тип лінії – "-.", розташування легенди (право низ) діапазон значень – $y (-10:60) \times (-35:25)$, колір лінії - #FFBBFF , прозорість лінії – 100%</p> <p>Побудувати стовпчикову діаграму (не менше 10 стовпчиків) "Середній вік жителів України за останні 10 років", дані можна взяти довільні або знайти в мережі Інтернет. У діаграмі передбачити назви стовпчиків. Колір використовувати як для попереднього графіку</p>
15	Побудувати графік функції $y = 5x^{1/3} - 3$ тип лінії – "--", розташування легенди (ліво верх) діапазон значень – $y (-15:70) \times (-40:40)$, колір лінії - #00BBFF , прозорість лінії – 25%

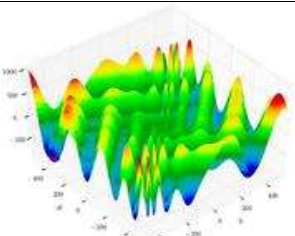
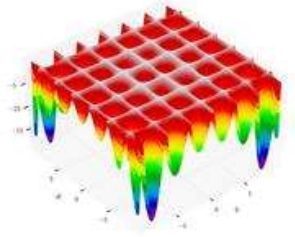
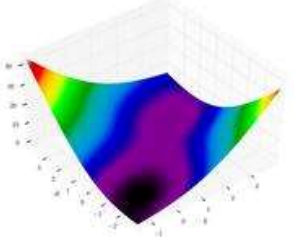
№	Завдання
	Побудувати кругову діаграму (не менше 10 сегментів) "врожай картоплі за 2019 по областях", дані можна взяти довільні або знайти в мережі Інтернет. У діаграмі передбачити назви сегментів в поле легенд.
16	Побудувати графік функції $y = 2x^2 + 3$ тип лінії – “-”, розташування легенди (ліво низ) діапазон значень – y (0:100) x (-50:50), колір лінії - #8800BB , прозорість лінії – 50%
	Побудувати стовпчикову діаграму (не менше 10 стовпчиків) "Курс долара до гривні за останні 10 місяців", дані можна взяти довільні або знайти в мережі Інтернет. У діаграмі передбачити назви стовпчиків. Колір використовувати як для попереднього графіку

Використовуючи засоби matplotlib побудувати графіки 3D відповідно до таблиці варіантів (таб. 7.2).

Таблиця 7.2 – Варіанти завдань

	Назва	Формула	Малюнок
1.	Функція Растрігіна	$f(\mathbf{x}) = An + \sum_{i=1}^n [x_i^2 - A \cos(2\pi x_i)]$ <p>where: $A = 10$</p>	
2.	Функція Еклі	$f(x, y) = -20 \exp \left[-0.2 \sqrt{0.5 (x^2 + y^2)} \right] - \exp[0.5 (\cos 2\pi x + \cos 2\pi y)] + e + 20$	
3.	Функція сфери	$f(\mathbf{x}) = \sum_{i=1}^n x_i^2$	
4.	Функція Розенброка	$f(\mathbf{x}) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$	
5.	Функція Біла	$f(x, y) = (1.5 - x + xy)^2 + (2.25 - x + xy^2)^2 + (2.625 - x + xy^3)^2$	

6.	Функція Гольдман- Прайса	$f(x, y) = \left[1 + (x + y + 1)^2 (19 - 14x + 3x^2 - 14y + 6xy + 3y^2) \right] \\ \left[30 + (2x - 3y)^2 (18 - 32x + 12x^2 + 48y - 36xy + 27y^2) \right]$	
7.	Функція Бута	$f(x, y) = (x + 2y - 7)^2 + (2x + y - 5)^2$	
8.	Функція Букіна N6	$f(x, y) = 100 \sqrt{ y - 0.01x^2 } + 0.01 x + 10 .$	
9.	Функція Матяся	$f(x, y) = 0.26 (x^2 + y^2) - 0.48xy$	
10.	Функція Леві N 13	$f(x, y) = \sin^2 3\pi x + (x - 1)^2 (1 + \sin^2 3\pi y) \\ + (y - 1)^2 (1 + \sin^2 2\pi y)$	
11.	Функція Гіммельблау	$f(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2.$	
12.	Функція трихгорбого верблюда	$f(x, y) = 2x^2 - 1.05x^4 + \frac{x^6}{6} + xy + y^2$	
13.	Функція Ізома	$f(x, y) = -\cos(x) \cos(y) \exp\left(-\left((x - \pi)^2 + (y - \pi)^2\right)\right)$	

14.	Функція "підставка для яєць" (Eggholder function)	$f(x, y) = -(y + 47) \sin \sqrt{\left \frac{x}{2} + (y + 47) \right } - x \sin \sqrt{ x - (y + 47) }$	
15.	Таблична функція Хольдера	$f(x, y) = - \left \sin x \cos y \exp \left(1 - \frac{\sqrt{x^2 + y^2}}{\pi} \right) \right $	
16.	Функція МакКорміка	$f(x, y) = \sin(x + y) + (x - y)^2 - 1.5x + 2.5y + 1$	

Приклад виконання

Варіант – 16.

```
[34]: import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
%config InlineBackend.figure_format = 'svg'
x = np.linspace(-50, 50, 100)
fig=plt.figure()
ax= fig.add_axes([0.1,0.1,0.9,0.9])
ax.plot(x, 2*(x**2)+3, color="#880088", ls="--",label="2x^2+3",alpha=0.5)
ax.set_ylim([0, 100])
ax.set_xlim([-50, 50])
ax.legend(loc=3)
ax.set_title("Графік функції")
ax.set_xlabel("X")
ax.set_ylabel("Y")
fig.savefig("MyFile.png",dpi=800)
```

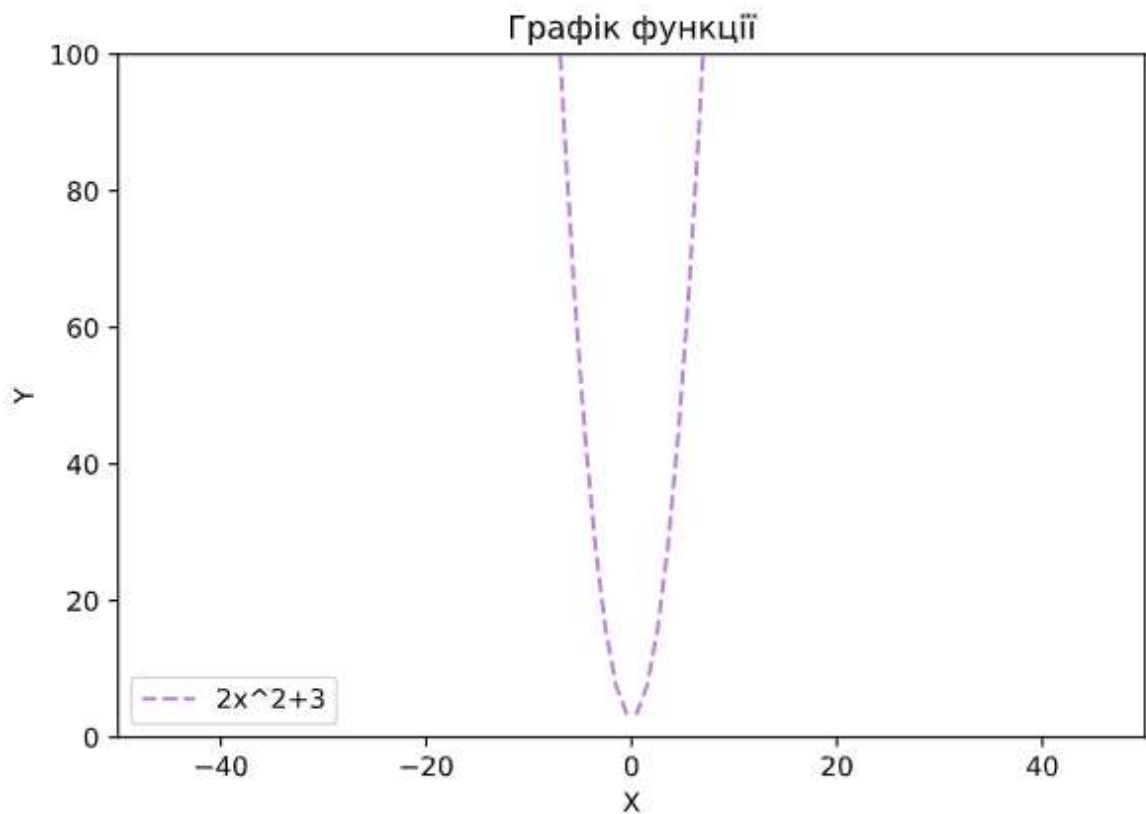


Рисунок 7.28 – Приклад виконання першого завдання


```
[36]: mounts = ["december", "january", "february", "march", "april"]
counts = [24.7, 24.3, 25.8, 26.75, 28.20]
plt.bar(mounts, counts, color="#880088")
plt.title("Course")
plt.xlabel("Mount")
plt.ylabel("Hryvna")
```

```
[36]: Text(0, 0.5, 'Hryvna')
```

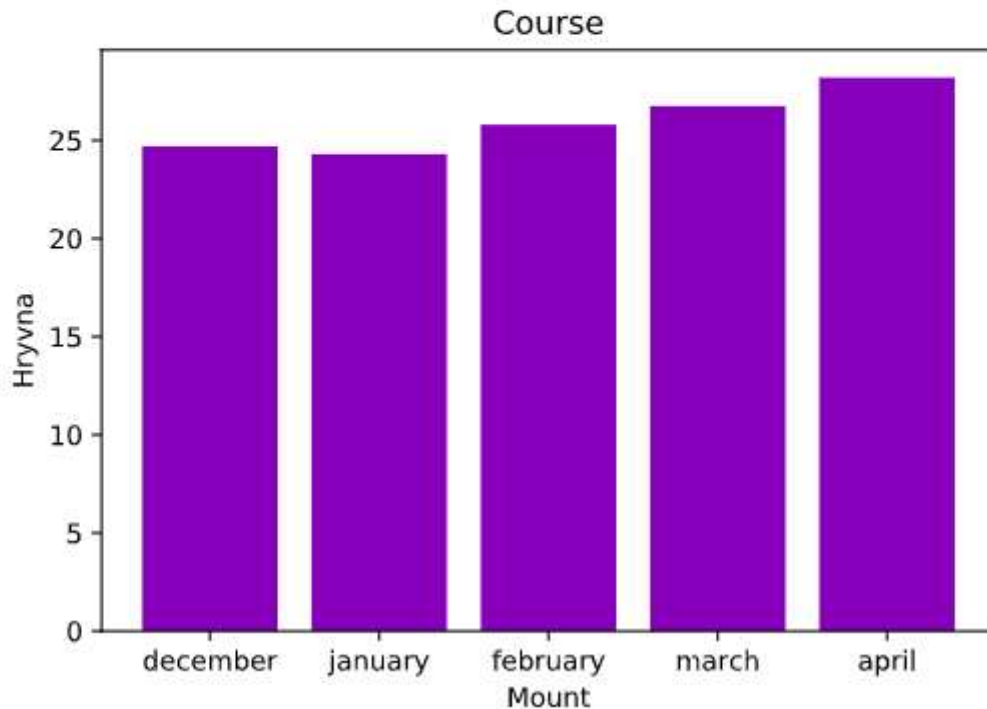


Рисунок 7.29 – Приклад виконання другого завдання

```
[4]: from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
from matplotlib import cm
import numpy as np
fig = plt.figure()
ax = fig.gca(projection='3d')
# Make data.
X = np.arange(-100, 100, 0.2)
Y = np.arange(-100, 100, 0.2)
X, Y = np.meshgrid(X, Y)
Z = np.sin(X+Y)+(X+Y)**2-(1.5*X)+(2.5*Y)+1
# Plot the surface.
surf = ax.plot_surface(X, Y, Z, rstride=5, cstride=5, cmap = cm.jet)
# Add a color bar which maps values to colors.
fig.colorbar(surf, shrink=0.5, aspect=4)
plt.show()
fig.savefig("MyFile.png", dpi=800)
```

Рисунок 7.30 – Приклад виконання третього завдання

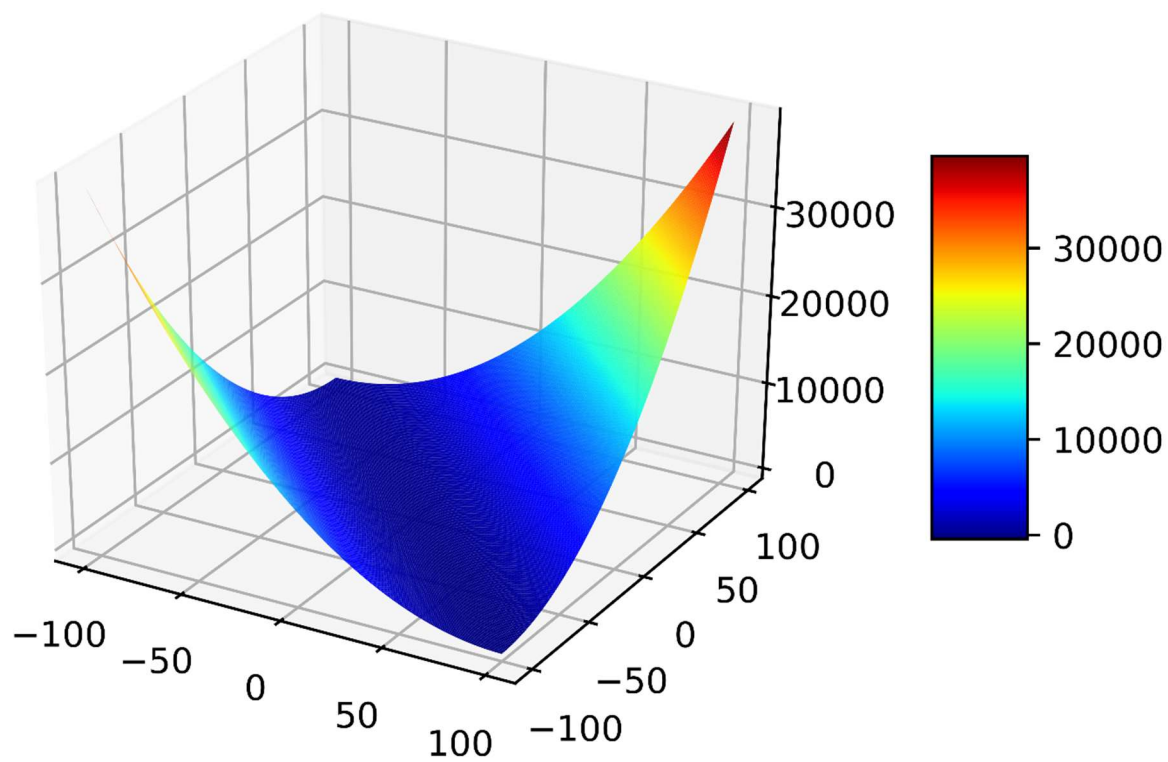


Рисунок 7.31 – Результат виконання третього завдання

Контрольні запитання

1. Модуль `matplotlib`. Призначення та базові принципи застосування. Наведіть приклади.
2. Опишіть базові принципи та механізми роботи з гістограмами у `matplotlib`.
3. Опишіть що таке легенди та як їх можна розташовувати.
4. Опишіть існуючі види налаштування відображення графіків, наведіть приклади.
5. Які види діаграм підтримуються у `matplotlib`, та якими методами будуються, наведіть приклади.
6. Наведіть приклади існуючих у `matplotlib` видів тривимірних графіків.
7. Опишіть що таке поверхня у `matplotlib` та чим вона відрізняється від каркасної поверхні. Наведіть приклади .
8. Які існують відмінності та переваги застосування тривимірних графіків перед двовимірними? Наведіть приклади.