Matplotlib II – subplotok, több grafikon egy ábrán, grid, stílus, layout

Subplot alapok

Egy ábrán több grafikon megjelenítése:

```
import matplotlib.pyplot as plt
import numpy as np

x = np.arange(0, 10)

plt.subplot(1, 2, 1)
plt.plot(x, x)
plt.title("y = x")

plt.subplot(1, 2, 2)
plt.plot(x, x**2)
plt.title("y = x^2")

plt.tight_layout()
plt.show()

#Magyarázat: subplot(sorok, oszlopok, pozíció)
```

Objektumorientált (OO) megközelítés

Komplexebb ábrák esetén strukturáltabb és tisztább.

Példa

```
fig, axes = plt.subplots(1, 2)

axes[0].plot(x, x)
axes[0].set_title("y = x")

axes[1].plot(x, x**2)
axes[1].set_title("y = x^2")

plt.tight_layout()
plt.show()
```

Feladat: Készíts 2×2 subplot elrendezést. Ábrázold:

- x
- x^2
- x^3
- sqrt(x)

```
import numpy as np
import matplotlib.pyplot as plt

x = np.arange(1, 11)

fig, axes = plt.subplots(2, 2)

axes[0, 0].plot(x, x)
axes[0, 0].set_title("x")

axes[0, 1].plot(x, x**2)
axes[0, 1].set_title("x^2")

axes[1, 0].plot(x, x**3)
axes[1, 0].set_title("x^3")

axes[1, 1].plot(x, np.sqrt(x))
axes[1, 1].set_title("sqrt(x)")

plt.tight_layout()
plt.show()
```

Ábraméret és grid

Ábraméret

```
plt.figure(figsize=(8, 4))
plt.plot(x, x**2)
plt.show()
```

Rácsvonal

```
plt.plot(x, x**2)
plt.grid(True)
plt.show()
```

OO módszerrel:

```
```

```
fig, ax = plt.subplots()
ax.plot(x, x**2)
ax.grid(True)
plt.show()
```

Kombinált vizualizáció

Feladat:

- Egy cég havi bevétele 3 évre (NumPy mátrix):
- Felső grafikon: évenkénti összbevétel (bar)
- Alsó grafikon: havi átlagbevétel (line)

```
import numpy as np
import matplotlib.pyplot as plt

np.random.seed(1)
bevetel = np.random.randint(100, 501, size=(3, 12))

ossz = bevetel.sum(axis=1)
havi_atlag = bevetel.mean(axis=0)

fig, axes = plt.subplots(2, 1, figsize=(8, 6))

axes[0].bar(["1. év", "2. év", "3. év"], ossz)
axes[0].set_title("Évenkénti bevétel")

axes[1].plot(havi_atlag)
axes[1].set_title("Havi átlagbevétel")
axes[1].grid(True)

plt.tight_layout()
plt.show()
```

Stílus és olvashatóság

Fontos alapelvek

- Ne használj túl sok görbét egy grafikonon
- Ha összehasonlítás a cél → subplot
- Használj gridet trendekhez
- Az ábra válaszoljon egy konkrét kérdésre

Összefoglaló feladatok

1. Feladat:

- Generálj 200 véletlen pontot.
- Bal oldalon scatter plot
- Jobb oldalon ugyanaz, de négyzetre emelt y értékkel

```
import numpy as np
import matplotlib.pyplot as plt

x = np.random.rand(200)
y = np.random.rand(200)

fig, axes = plt.subplots(1, 2, figsize=(10, 4))

axes[0].scatter(x, y)
axes[0].set_title("Eredeti")

axes[1].scatter(x, y**2)
axes[1].set_title("Transzformált")

plt.tight_layout()
plt.show()
```

2. Feladat – Több függvény összehasonlítása 2×2 subplot rácsban

1. Generálj 0–10 intervallumon 300 pontot.
2. Ábrázold 2×2-es subplot elrendezésben:
    - y = x
    - y = x^2
    - y = sqrt(x)
    - y = sin(x)
3. Minden grafikon kapjon külön címet.
4. Használj egységes stílust.
5. Alkalmazz tight_layout()-ot.

```
import numpy as np import matplotlib.pyplot as plt

plt.style.use("seaborn-v0_8")

x = np.linspace(0, 10, 300)

fig, axs = plt.subplots(2, 2, figsize=(10,8))

axs[0,0].plot(x, x)
axs[0,0].set_title("y = x")
axs[0,0].grid(True)

axs[0,1].plot(x, x**2)
```

```
axs[0,1].set_title("y = x^2")
axs[0,1].grid(True)

axs[1,0].plot(x, np.sqrt(x))
axs[1,0].set_title("y = sqrt(x)")
axs[1,0].grid(True)

axs[1,1].plot(x, np.sin(x))
axs[1,1].set_title("y = sin(x)")
axs[1,1].grid(True)

plt.tight_layout()
plt.show()
```

3. Feladat – Bevételi dashboard (2×1 layout)

1. Generálj 12×3 bevételi adatot.
2. Felső subplot: vonaldiagram havi bevételről üzletenként.
3. Alsó subplot: oszlopdiagram az éves összbevételről.
4. Használj grid-et.
5. Igazítsd a layout-ot.

```
import numpy as np import matplotlib.pyplot as plt

np.random.seed(0)
bevetel = np.random.randint(1000, 5000, (12, 3))
honapok = np.arange(1, 13)

fig, axs = plt.subplots(2, 1, figsize=(10,8))

#Felső grafikon

axs[0].plot(honapok, bevetel[:,0], label="Üzlet 1")
axs[0].plot(honapok, bevetel[:,1], label="Üzlet 2")
axs[0].plot(honapok, bevetel[:,2], label="Üzlet 3")
axs[0].set_title("Havi bevételek")
axs[0].set_xlabel("Hónap")
axs[0].set_ylabel("Bevétel")
axs[0].legend()
axs[0].grid(True)

#Alsó grafikon

eves = np.sum(bevetel, axis=0)
axs[1].bar(["Üzlet 1", "Üzlet 2", "Üzlet 3"], eves)
axs[1].set_title("Éves összbevétel")
axs[1].set_ylabel("Bevétel")
axs[1].grid(axis="y")
```

```
plt.tight_layout()
plt.show()
```

4. Feladat – Scatter és eloszlás egy ábrán (1×2 layout)

1. Generálj 500 darab (x,y) normál eloszlású pontot.
2. Bal oldalon scatter plot.
3. Jobb oldalon hisztogram az y értékekről.
4. Használj közös stílust.
5. Adj címeket és rácsvonalat.

```
import numpy as np import matplotlib.pyplot as plt

plt.style.use("ggplot")

np.random.seed(1)
x = np.random.randn(500)
y = np.random.randn(500)

fig, axs = plt.subplots(1, 2, figsize=(12,5))

axs[0].scatter(x, y, alpha=0.6)
axs[0].set_title("Scatter plot")
axs[0].set_xlabel("X")
axs[0].set_ylabel("Y")
axs[0].grid(True)

axs[1].hist(y, bins=30)
axs[1].set_title("Y eloszlása")
axs[1].set_xlabel("Y")
axs[1].set_ylabel("Gyakoriság")
axs[1].grid(True)

plt.tight_layout()
plt.show()
```

5. Feladat – Hőtérkép és vonaldiagram kombináció

1. Generálj 20×20-as mátrixot.
2. Bal subplot: imshow() hőtérkép.
3. Jobb subplot: az oszlopátlagok vonaldiagramja.
4. Adj colorbar-t.
5. Használj layout optimalizálást.

```
import numpy as np import matplotlib.pyplot as plt
```

```
np.random.seed(2)
matrix = np.random.rand(20, 20)

fig, axs = plt.subplots(1, 2, figsize=(12,5))

img = axs[0].imshow(matrix)
axs[0].set_title("Hőtérkép")
axs[0].set_xlabel("Oszlop")
axs[0].set_ylabel("Sor")

plt.colorbar(img, ax=axs[0])

col_mean = np.mean(matrix, axis=0)
axs[1].plot(col_mean)
axs[1].set_title("Oszlopátlagok")
axs[1].set_xlabel("Oszlop index")
axs[1].set_ylabel("Átlag")
axs[1].grid(True)

plt.tight_layout()
plt.show()
```

6. Feladat – Komplex 2×2 adatvizualizációs panel

1. Generálj 1000 normál eloszlású adatot.
2. Készíts 2×2 panelt:
3. Hisztogram
4. Dobozdiagram (boxplot)
5. Kumulatív összeg grafikon
6. Szórásablakos (rolling std) grafikon
7. Használj egységes stílust és layout optimalizálást.

```
import numpy as np import matplotlib.pyplot as plt

plt.style.use("bmh")

np.random.seed(3)
data = np.random.randn(1000)

fig, axs = plt.subplots(2, 2, figsize=(10,8))

#Hisztogram

axs[0,0].hist(data, bins=30)
axs[0,0].set_title("Hisztogram")
axs[0,0].grid(True)

#Boxplot
```

```
axs[0,1].boxplot(data)
axs[0,1].set_title("Boxplot")
axs[0,1].grid(True)

#Kumulatív összeg

cumsum = np.cumsum(data)
axs[1,0].plot(cumsum)
axs[1,0].set_title("Kumulatív összeg")
axs[1,0].grid(True)

#Rolling std (ablak=50)

window = 50
rolling_std = [np.std(data[i:i+window]) for i in range(len(data)-window)]
axs[1,1].plot(rolling_std)
axs[1,1].set_title("Rolling szórás (50)")
axs[1,1].grid(True)

plt.tight_layout()
plt.show()
```

From:
https://edu.iit.uni-miskolc.hu/ - **Institute of Information Science - University of Miskolc**

Permanent link:
https://edu.iit.uni-miskolc.hu/tanszek:oktatas:muszaki_informatika:matplotlib_ii._toebb_plot_stilus_feliratok

Last update: **2026/02/27 08:29**