

# 10. gyakorlat

Python kurzus



**Big Data adatfeldolgozás és  
adatvizualizáció**

# 1. A Matplotlib könyvtár

- A Matplotlib az adatvizualizáció egyik eszköztára
- Integrálható a Pandas és a NumPy adatfeldolgozó könyvtárakkal
- Telepítés: **pip install matplotlib**
- Vonaldiagramok, hisztogramok, scatter plotok és más diagramok készítésére: **plt.plot()**, **plt.subplot(1, 2, 1)**, **plt.scatter()**, **plt.hist()**,  
**plt.boxplot()**, **plt.subplots()** és **twinx()**,

# Vonaldiagram

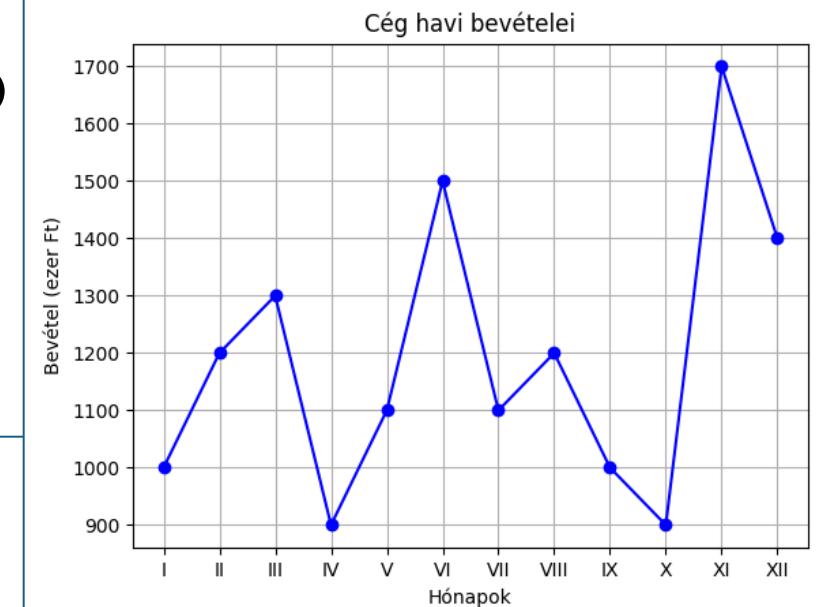
**1. Példa:** Egy cég havi bevételeit ábrázoljuk vonaldiagramon.

A `plt.plot()` függvényt alkalmazzuk a grafikon létrehozásához.

Tengelyfeliratozás és -cím hozzáadása: `plt.xlabel()`, `plt.ylabel()`, `plt.title()`

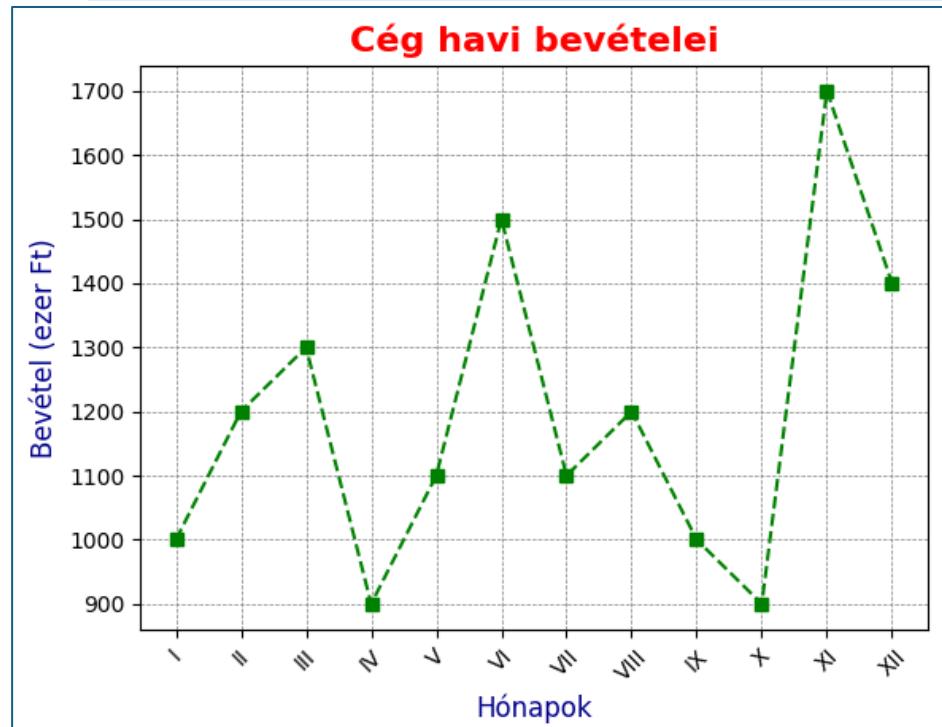
A **marker** argumentum a pontok kiemelésére szolgál, a `linestyle` a vonal típusát állítja be. A diagram megjelenítés a `plt.show()`-val történik.

```
import matplotlib.pyplot as plt
# Adatok: hónapok és bevételek
hónapok = ['I', 'II', 'III', 'IV', 'V', 'VI', 'VII', 'VIII', 'IX', 'X', 'XI', 'XII']
bevételek = [1000, 1200, 1300, 900, 1100, 1500, 1100, 1200, 1000, 900, 1700, 1400]
# Vonaldiagram készítése
plt.plot(hónapok, bevételek, marker='o', linestyle='-', color='b')
plt.title('Cég havi bevételei')
plt.xlabel('Hónapok')
plt.ylabel('Bevétel (ezer Ft)')
plt.grid(True)
plt.show()
```



# Tengelyek, címek és feliratok testreszabása

```
plt.plot(hónapok, bevételek, marker='s', linestyle='--', color='g')
plt.title('Cég havi bevételei', fontsize=16, fontweight='bold', color='navy')
plt.xlabel('Hónapok', fontsize=12, color='darkgreen')
plt.ylabel('Bevétel (ezer Ft)', fontsize=12, color='darkgreen')
plt.xticks(rotation=45)
plt.grid(color='gray', linestyle='--', linewidth=0.5)
plt.show()
```



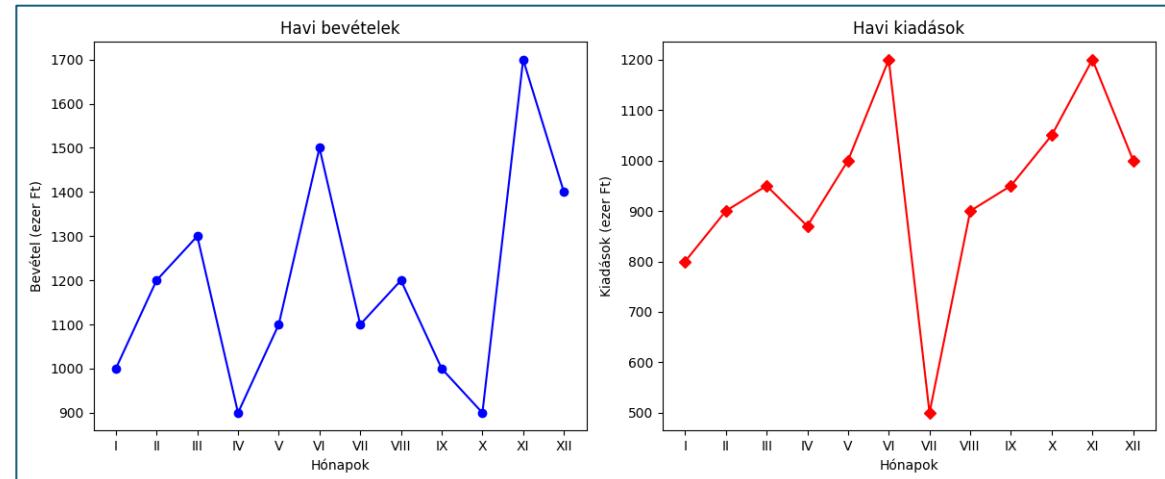
Cím és tengelyek formázása:  
betűméret, szín, stílus beállítása.  
Rotáció alkalmazása az x tengelyen:  
**plt.xticks(rotation=45)**  
Rácsvonalak vezérlése: **plt.grid()**

# Többszörös grafikonok egy ábrán belül - subplotok

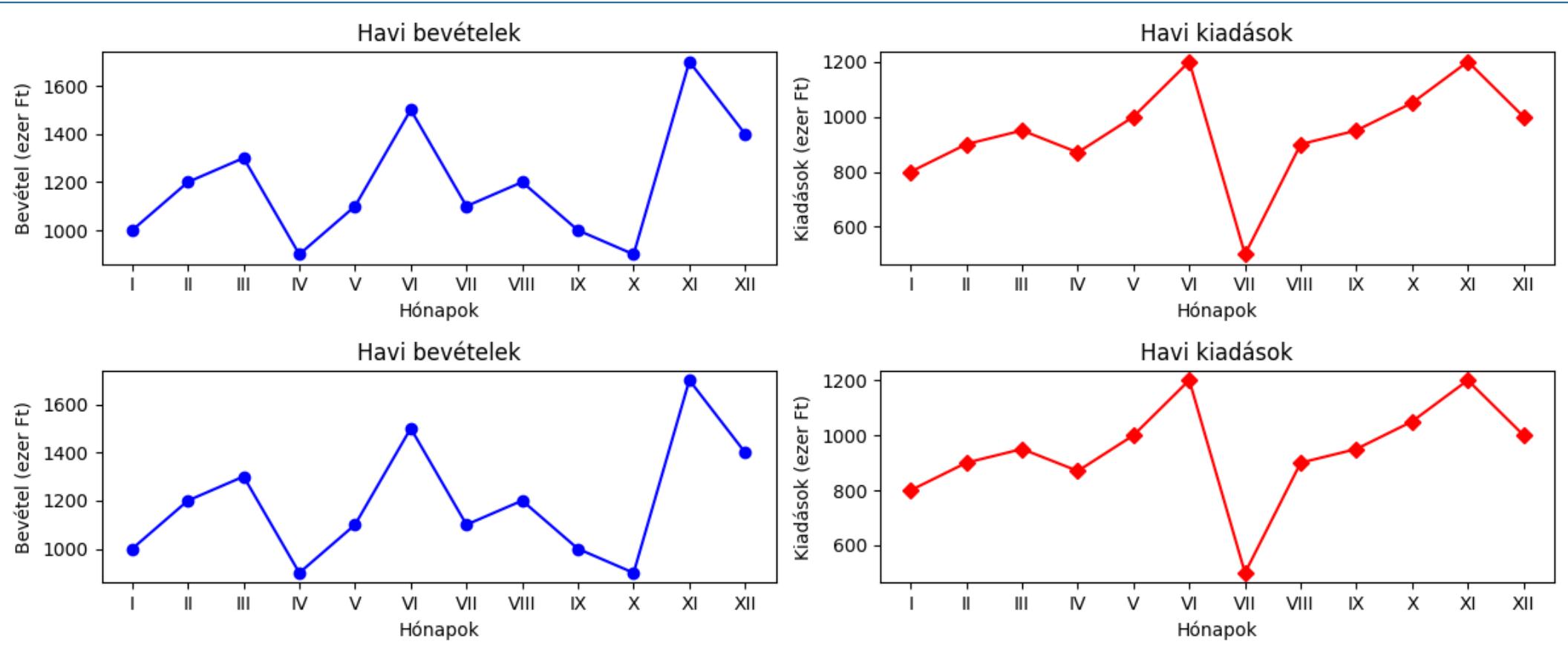
**2. Példa:** Ábrázoljunk egyszerre két diagramot – a bevételeket és a kiadásokat.

```
kiadások = [800, 900, 950, 870, 1000, 1200, 500, 900,  
950, 1050, 1200, 1000]  
plt.figure(figsize=(12, 5))  
# Első subplot  
plt.subplot(1, 2, 1) # 1 sor, 2 oszlop, 1. subplot  
plt.plot(hónapok, bevételek, color='blue', marker='o')  
plt.title('Havi bevételek')  
plt.xlabel('Hónapok')  
plt.ylabel('Bevétel (ezer Ft)')  
# Második subplot  
plt.subplot(1, 2, 2) # 1 sor, 2 oszlop, 2. subplot  
plt.plot(hónapok, kiadások, color='red', marker='D')  
plt.title('Havi kiadások')  
plt.xlabel('Hónapok')  
plt.ylabel('Kiadások (ezer Ft)')  
plt.tight_layout() # Igazítás a helyes megjelenítéshez  
plt.show()
```

Több diagram ábrázolása egy ablakban:  
**plt.subplot()** függvény használata.  
Paraméterek: sorok, oszlopok száma és  
a subplot pozíciója.  
**plt.tight\_layout()** az ábrák közötti  
térföző optimalizálásához.



## Feladat: Ábrázoljunk egyszerre négy diagramot - másolással



# Scatter plotok a korrelációk vizsgálatához

3. Példa: Látogatók száma és eladások közötti kapcsolat a plt.scatter() metódussal.

```
# Adatok generálása
```

```
látogatók = [120, 150, 180, 200, 170, 160, 140, 180, 190, 210, 190, 200]
```

```
eladások = [150, 180, 220, 240, 200, 230, 170, 190, 210, 250, 230, 220]
```

```
# Scatter plot készítése
```

```
plt.scatter(látogatók, eladások, color='red')
```

```
plt.title('Látogatók száma és eladások közötti kapcsolat')
```

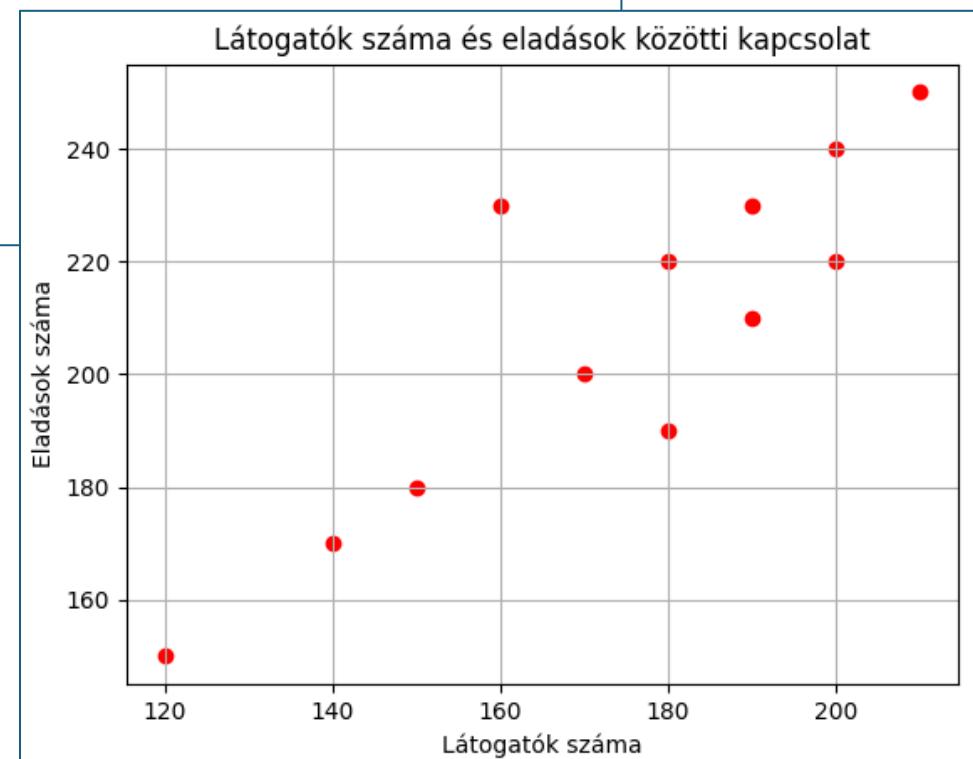
```
plt.xlabel('Látogatók száma')
```

```
plt.ylabel('Eladások száma')
```

```
plt.grid(True)
```

```
plt.show()
```

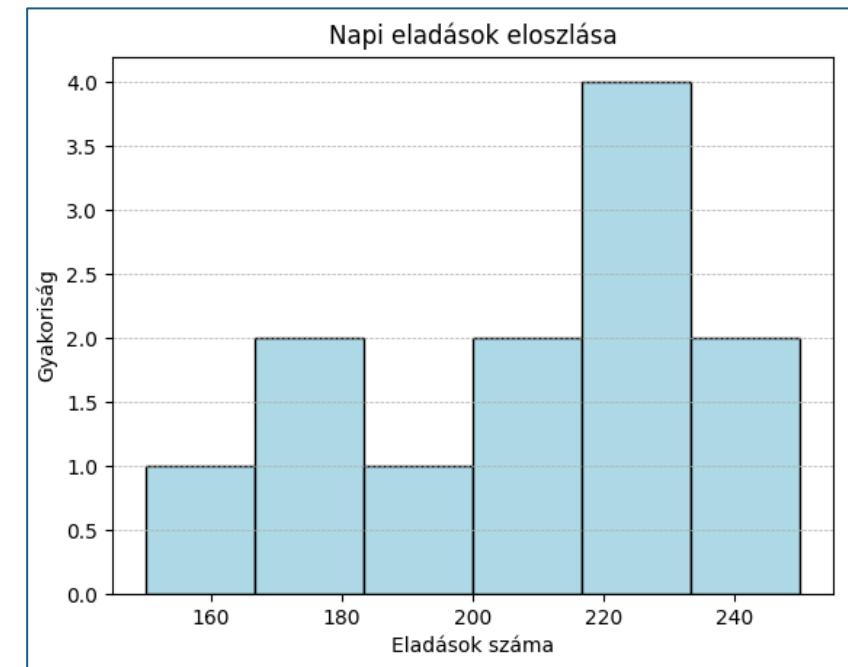
A plt.grid(True) jeleníti meg a rajzterület alapbeállítású vezetőrácsait.



# Hisztogramok és eloszlásvizsgálatok

**4. példa:** Hisztogram az adatok gyakoriságának bemutatására a `plt.hist()` függvényel.

```
# Adatok generálása  
eladások = [150, 180, 220, 240, 200, 230, 170, 190, 210, 250,  
           230, 220]  
# Hisztogram készítése  
plt.hist(eladások, bins=6, color='lightblue', edgecolor='black')  
plt.title('Napi eladások eloszlása')  
plt.xlabel('Eladások száma')  
plt.ylabel('Gyakoriság')  
plt.grid(axis='y', linestyle='--', linewidth=0.5)  
plt.show()
```



A **bins** paraméter határozza meg a sávok számát.

Az **edgecolor** az oszlopok körfonalának színét állítja be a jobb láthatóság érdekében.

# Box plotok használata adatok szórásának bemutatására

**5. példa:** A dobozábrák vizuálisan mutatják az adatok szórását, középértékét, valamint a kiugró értékeket. A `plt.boxplot()` függvény hozza létre a dobozábrát.

**Box plotok az adatok szórásának és kiugró értékeinek bemutatására.**

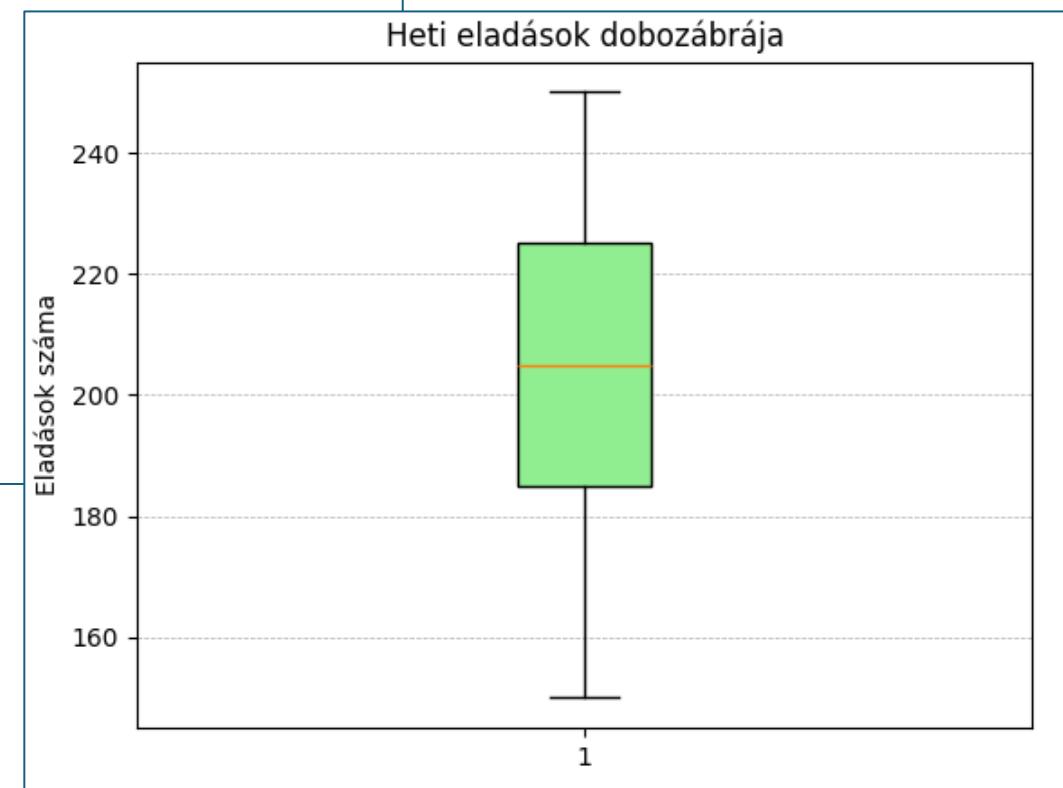
# Heti eladások adatai

```
heti_eladasok = [150, 180, 220, 240, 200, 230,  
    170, 190, 210, 250, 230, 220, 180, 195, 205]
```

# Box plot készítése

```
plt.boxplot(heti_eladasok, patch_artist=True,  
    boxprops=dict(facecolor='lightgreen'))  
plt.title('Heti eladások dobozárája')  
plt.ylabel('Eladások száma')  
plt.grid(axis='y', linestyle='--', linewidth=0.5)  
plt.show()
```

A `patch_artist` és `boxprops` paraméterek segítségével testreszabhatjuk a doboz megjelenését. Az ábra megmutatja a mediánt, a kvartiliseket és az esetleges kiugró értékeket.



```
heti_eladasok_1 = [150, 180, 220, 240, 200, 230, 170, 190, 210, 250, 230, 220, 180, 195, 205]
heti_eladasok_2 = [160, 170, 210, 230, 190, 220, 180, 200, 220, 240, 210, 200, 170, 185, 195]

fig, ax = plt.subplots()

box1 = ax.boxplot(heti_eladasok_1, positions=[1], patch_artist=True,
boxprops=dict(facecolor='lightgreen'))

box2 = ax.boxplot(heti_eladasok_2, positions=[2], patch_artist=True,
boxprops=dict(facecolor='lightblue'))

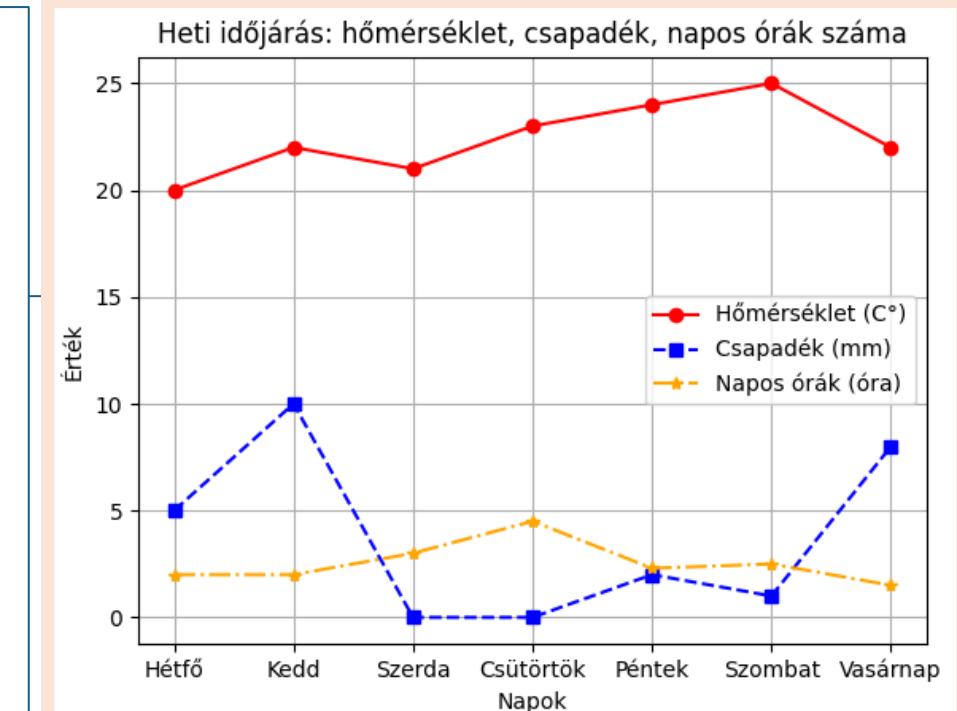
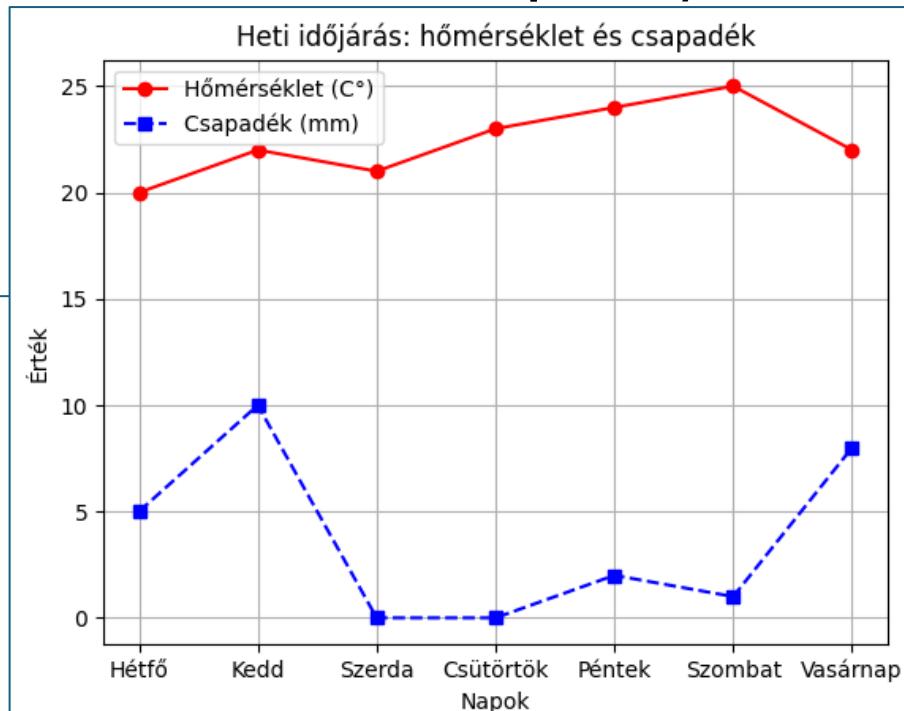
plt.title('Heti eladások dobozárája')
plt.ylabel('Eladások száma')
plt.xticks([1, 2], ['Heti eladások 1', 'Heti eladások 2'])
plt.grid(axis='y', linestyle='--', linewidth=0.5)

plt.show()
```

# Kombinált grafikonok: 1. Overlay grafikon készítése

**6. Példa:** Két különböző adatsor ábrázolása egy grafikonon. Egy diagramon belül két grafikon, amelyek egy város napi átlaghőmérsékletét és csapadékmennyiségét mutatják.

```
napok = ['Hétfő', 'Kedd', 'Szerda', 'Csütörtök', 'Péntek', 'Szombat', 'Vasárnap']
homerseklet = [20, 22, 21, 23, 24, 25, 22] # Celsius fok
csapadek = [5, 10, 0, 0, 2, 1, 8] # mm
plt.plot(napok, homerseklet, marker='o', color='red', label='Hőmérséklet (C°)')
plt.plot(napok, csapadek, marker='s', color='blue', linestyle='--', label='Csapadék (mm)')
plt.title('Heti időjárás: hőmérséklet és csapadék')
plt.xlabel('Napok')
plt.ylabel('Érték')
plt.legend()
plt.grid(True)
plt.show()
```

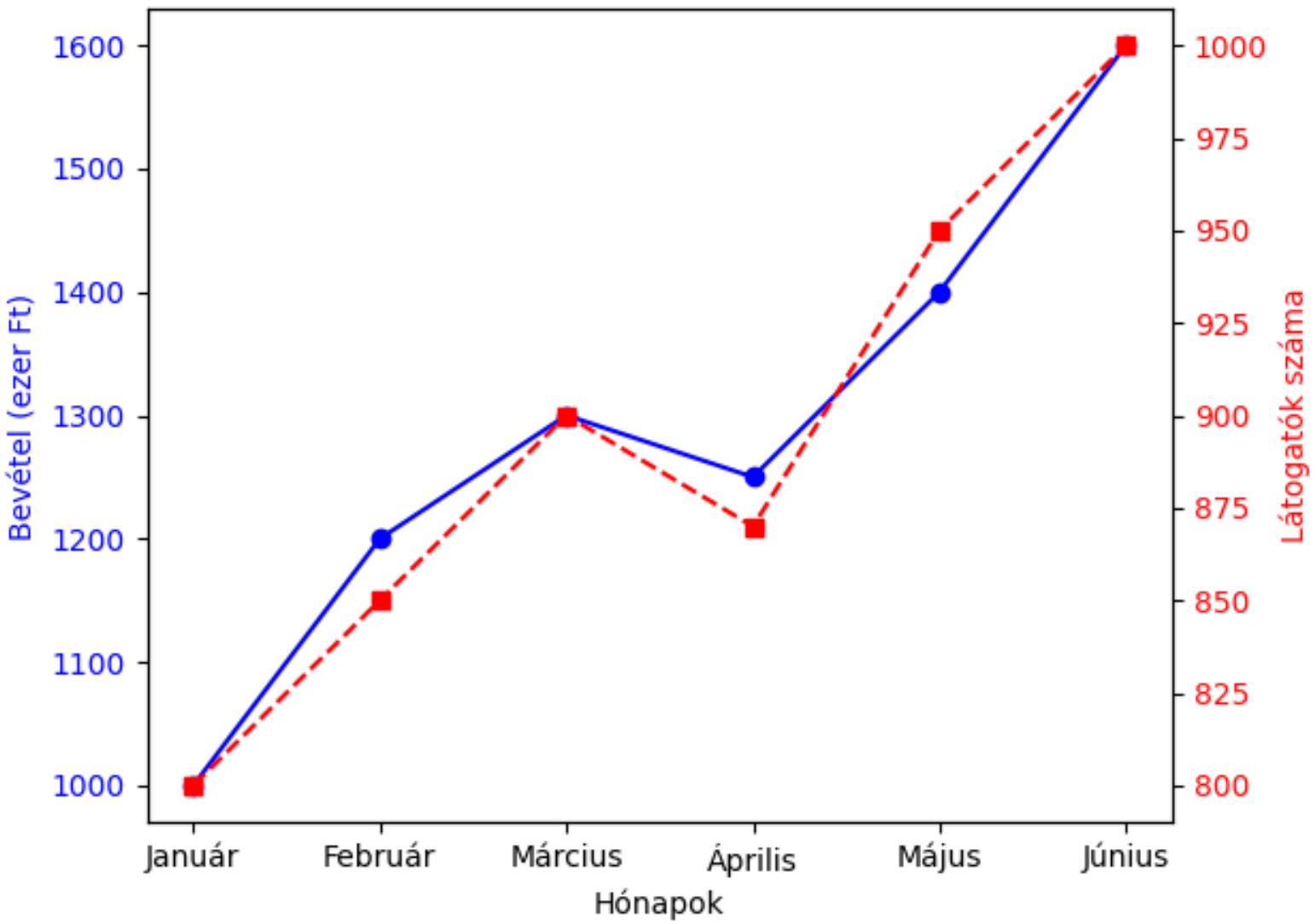


## Kombinált grafikonok: 2. Két y tengely egy diagramon

7. Példa: Egy bolt havi bevételeit és látogatóinak számát egy grafikonon jelenítjük meg a plt.subplots() és a twinx() metódusokkal.

```
# Adatok generálása
honapok = ['Január', 'Február', 'Március', 'Április', 'Május', 'Június']
bevetelek = [1000, 1200, 1300, 1250, 1400, 1600]
latogatok = [800, 850, 900, 870, 950, 1000]
# Kombinált diagram készítése
fig, ax1 = plt.subplots()    #létrehoz egy ábrát és egy ax1 tengelyt
# Első y tengely: bevételek
ax1.plot(honapok, bevetelek, color='blue', marker='o', label='Bevétel')
ax1.set_xlabel('Hónapok')
ax1.set_ylabel('Bevétel (ezer Ft)', color='blue')
ax1.tick_params(axis='y', labelcolor='blue')
# Második y tengely: látogatók száma
ax2 = ax1.twinx()
ax2.plot(honapok, latogatok, color='red', marker='s', linestyle='--', label='Látogatók száma')
ax2.set_ylabel('Látogatók száma', color='red')
ax2.tick_params(axis='y', labelcolor='red')
plt.title('Havi bevételek és látogatók száma')
fig.tight_layout()
plt.show()
```

### Havi bevételek és látogatók száma



## 2. Adatvizualizáció a Pandas eszközeivel

- A Pandas és a Matplotlib jól integráltak, ez lehetővé teszi a DataFrame objektumok közvetlen vizualizálását.
- A Matplotlib alapértelmezés szerint megjelenik, ha a **Pandas plot()** metódusát meghívjuk a **DataFrame-re**.
- Fontos csomagok importálása: **import pandas as pd**  
**import matplotlib.pyplot as plt**

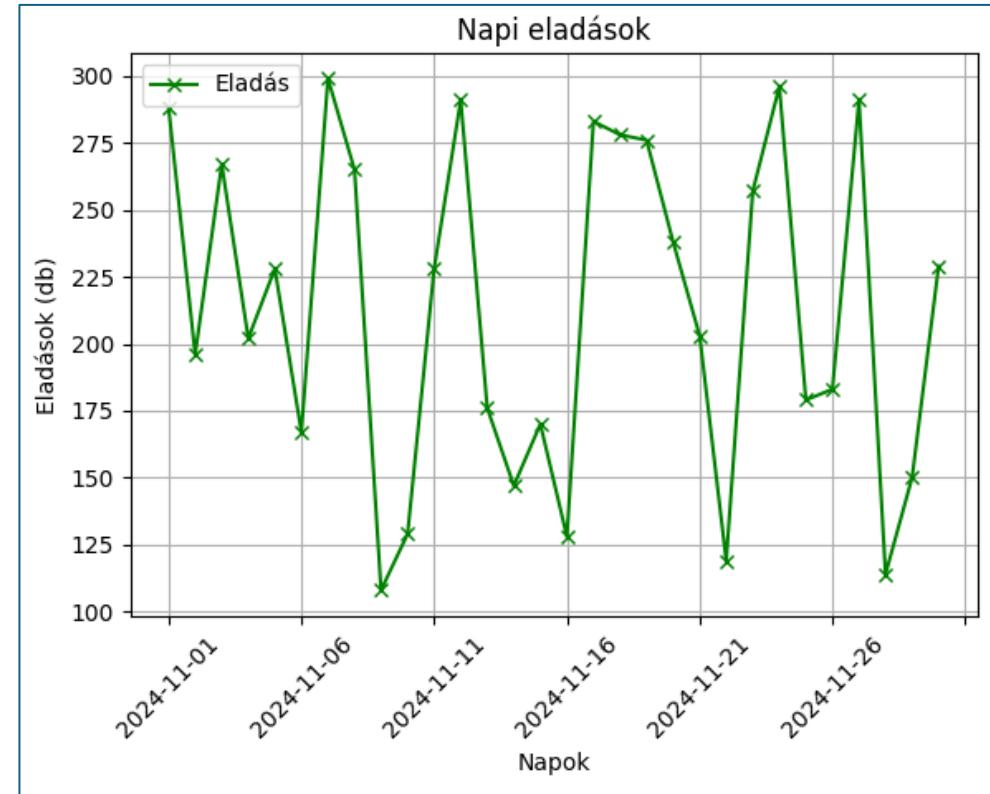
# Beépített Pandas plot metódusok

A Pandas DataFrame és Series objektumai beépített **plot()** metódussal rendelkeznek.

## 8. Példa: Vonal-diagram készítése közvetlenül Pandas segítségével

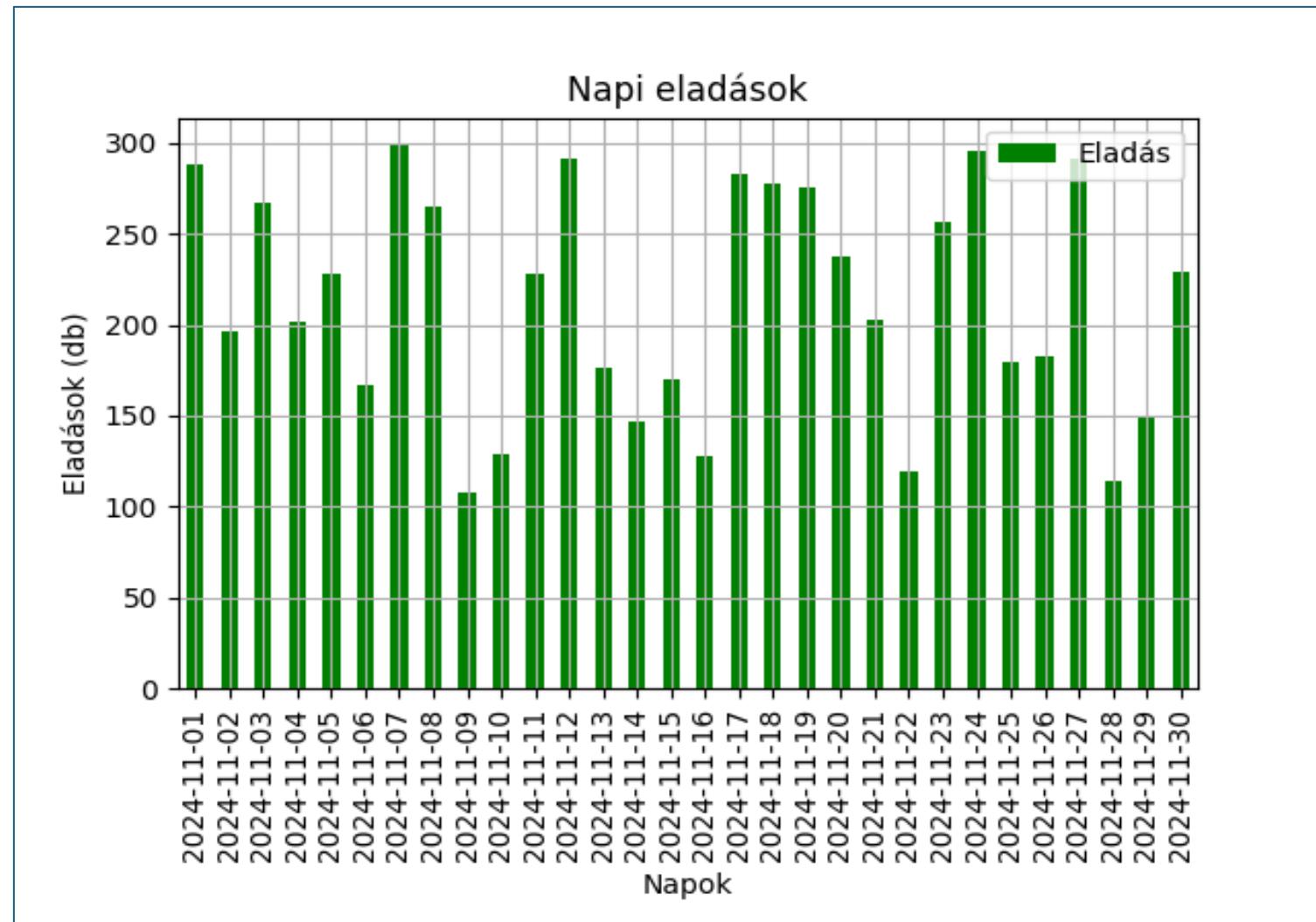
```
import pandas as pd
# CSV adatállomány betöltése
adatok = pd.read_csv('eladasok_30.csv')
adatok.plot(x='Nap', y='Eladás', kind='line', marker='x',
             color='green', title='Napi eladások')
plt.xlabel('Napok')
plt.ylabel('Eladások (db)')
plt.grid(True)
plt.show()
# A beolvasott adatok megtekintése
print(adatok.head())
```

	Nap	Eladás
0	2024-11-01	202
1	2024-11-02	279
2	2024-11-03	192
3	2024-11-04	114
4	2024-11-05	206



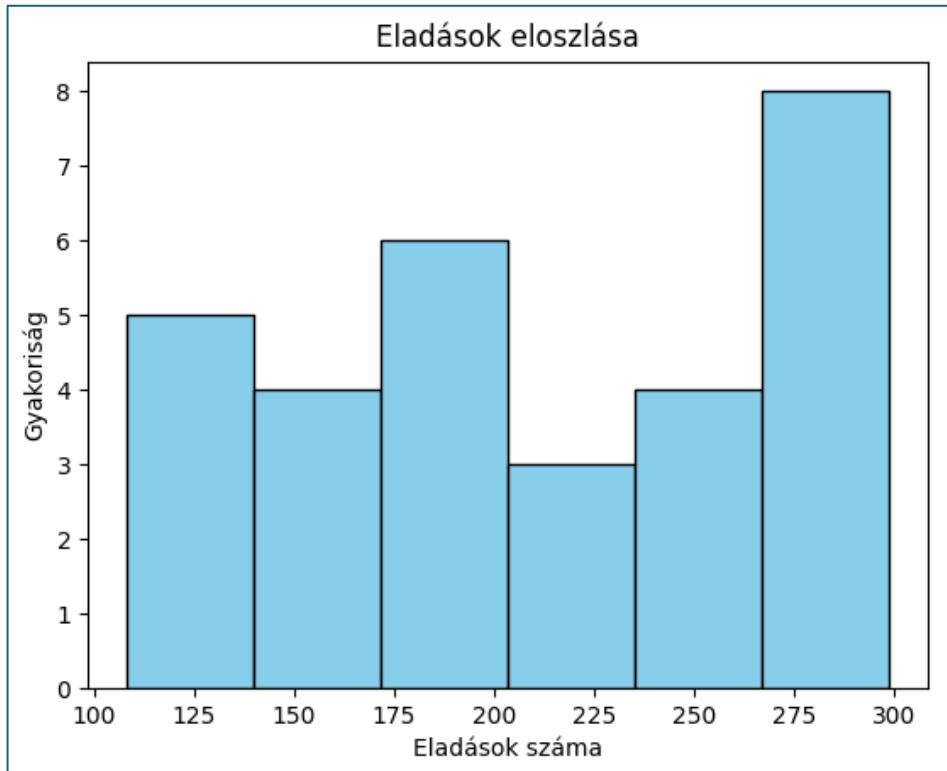
A **kind** paraméter határozza meg a diagram típusát (pl. **line**, **bar**), az **x** és **y** a tengelyek megadása, a **marker** és **color** paraméterek a megjelenést szabályozzák.

## Oszlopokkal: adatok.plot(x='Nap', y='Eladás', kind='bar', color='green', title='Napi eladások')



## 9. Példa: Hisztogram készítése az eladások eloszlásának bemutatására

```
# Hisztogram készítése  
adatok['Eladás'].plot(kind='hist', bins=6, color='skyblue', edgecolor='black',  
title='Eladások eloszlása')  
plt.xlabel('Eladások száma')  
plt.ylabel('Gyakoriság')  
plt.show()
```



A **kind='hist'** beállítással hisztogramot készítünk, amely az adatok eloszlását mutatja.

A **bins** paraméter határozza meg a csoportok számát.

### 3. A Seaborn könyvtár bevezetése és használata

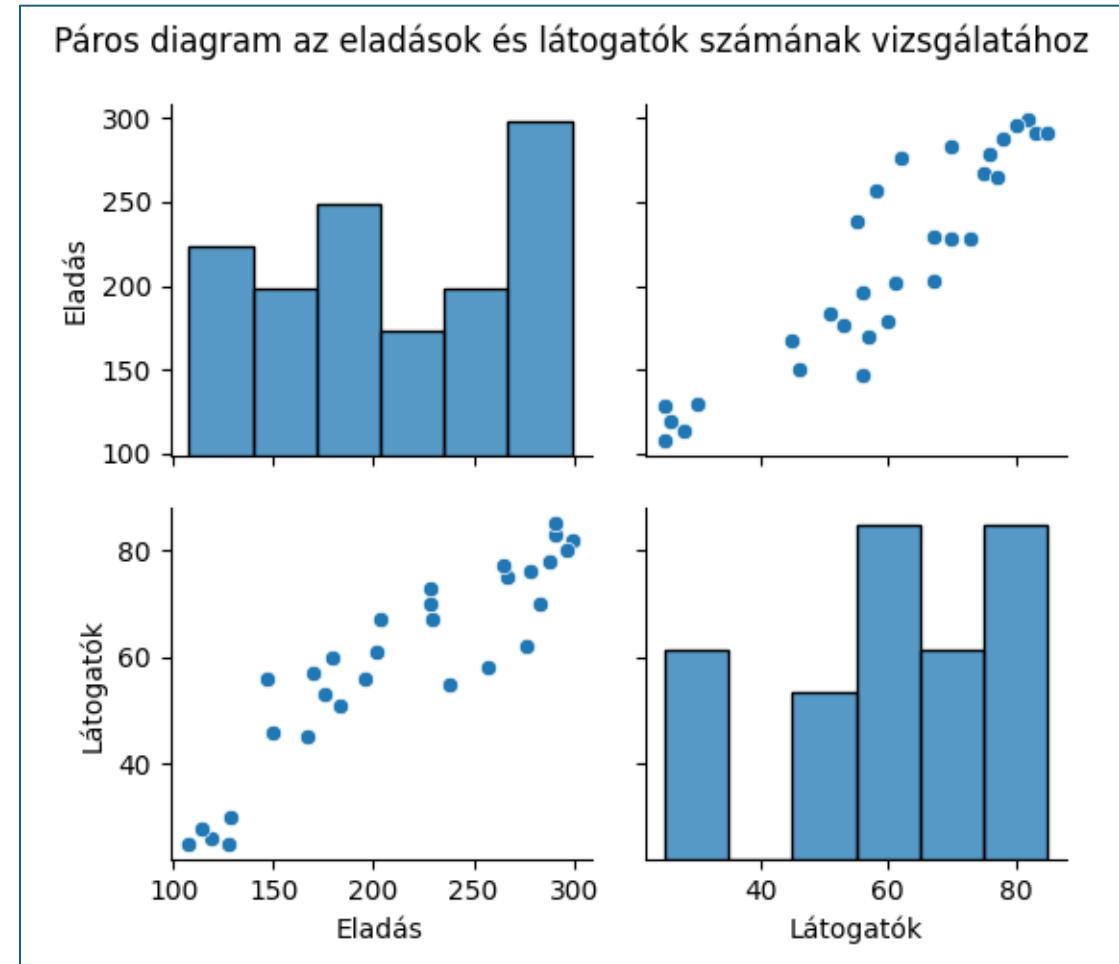
- A Seaborn egy magasabb szintű Python-könyvtár Matplotlib alapokon
- Előnyei: egyszerű használat, különleges diagramok, Pandas integráció
- Jól alkalmazható statisztikai adatok vizualizálására és összetett grafikonok létrehozására
- Hőtérképek, eloszlásdiagramok és trenddiagramok
- Beépített esztétikai beállítások
- Könnyen integrálható Pandas DataFrame-ekkel
- Több beépített plot típust tartalmaz, amelyekkel különböző típusú adatvizualizációkat készíthetünk.
- Telepítés: **pip install seaborn**

# Seaborn pairplot()

A pairplot() több változó közötti kapcsolatok és eloszlások bemutatására szolgál, és automatikusan elkészíti a scatter plotokat és hisztogramokat.

**10. Példa:** Többváltozós elemzés:

```
import seaborn as sns  
import pandas as pd  
import matplotlib.pyplot as plt  
# Pairplot használata  
sns.pairplot(adatok)  
plt.suptitle('Páros diagram az eladások  
és látogatók számának vizsgálatához', y=0.98)  
plt.show()
```



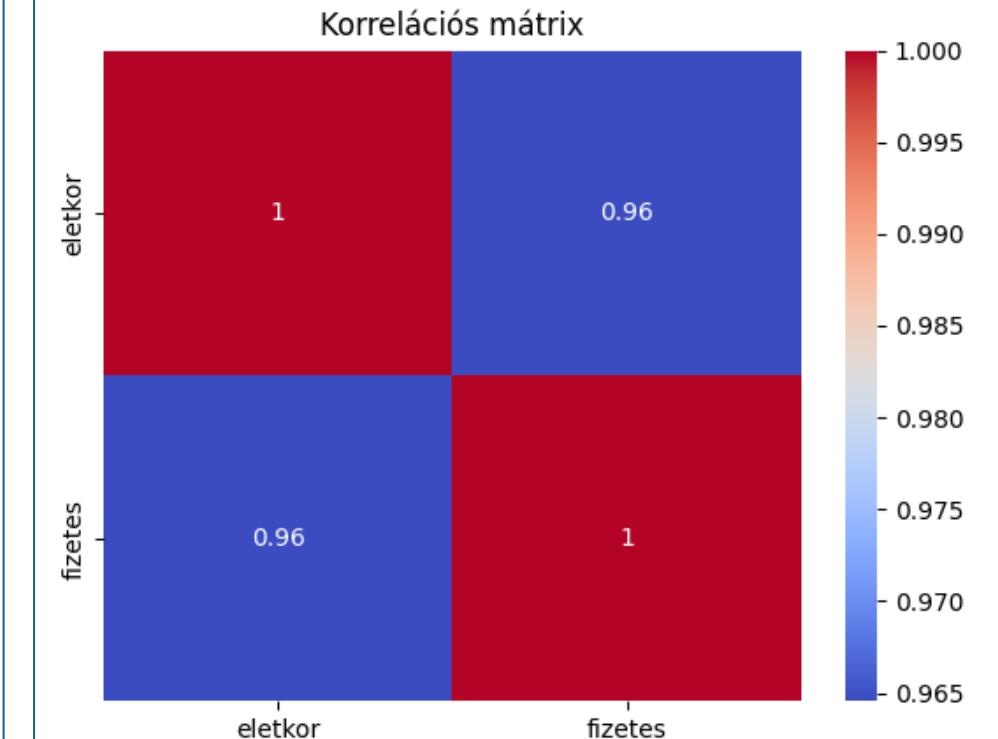
# Seaborn heatmap()

## 11. Példa: Adatok korrelációjának vizsgálata

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
adatok = {
    'eletkor': [23, 45, 34, 25, 32, 40, 29, 48, 37, 22],
    'fizetes': [250, 500, 300, 260, 320, 480, 290, 520, 410, 240]
}
adat_df = pd.DataFrame(adatok)
# Korrelációs mátrix kiszámítása
korrelacio = adat_df.corr()
print("\nKorrelációs mátrix:")
print(korrelacio)
# Korrelációs mátrix megjelenítése
sns.heatmap(korrelacio, annot=True, cmap='coolwarm')
plt.title('Korrelációs mátrix')
plt.show()
```

Korrelációs mátrix:

	eletkor	fizetes
eletkor	1.000000	0.964549
fizetes	0.964549	1.000000



Találkozunk az előadáson!