

SPRAWOZDANIE

Zajęcia: Nauka o danych I

Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium Nr 4 Data 09.01.2025 Temat: Biblioteka Matplotlib Wariant 8	Imię Nazwisko Hubert Mentel Informatyka II stopień, niestacjonarne, 1 semestr, gr.1a
---	---

1. Zadanie:

Wykonanie wykresów:

- wykres liniowy,
- wykres słupkowy,
- histogram,
- wykres kołowy,
- wykres liniowy - kolory, linie i style markerów,
- wykres liniowy - dodanie siatki,
- wykres liniowy - dodanie adnotacji,
- wykres punktowy,
- wykres 3D,

Pliki dostępne są na GitHubie pod linkiem:

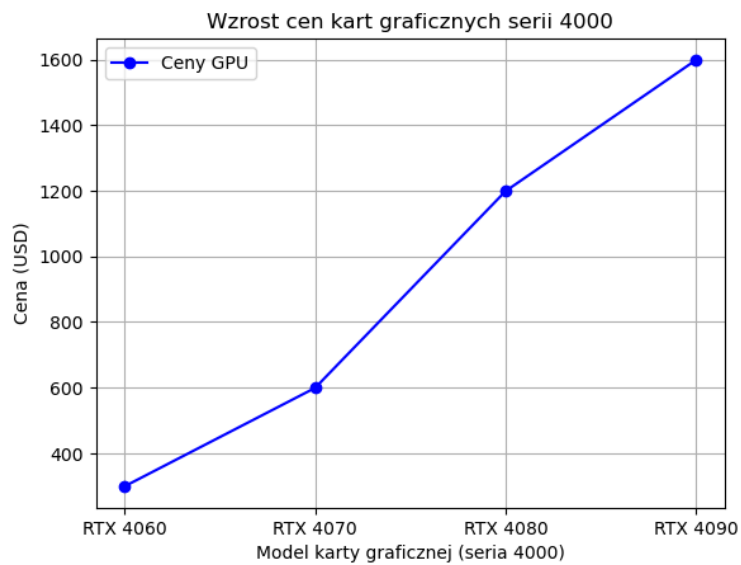
<https://github.com/HubiPX/NOD/tree/master/Zadanie%204>

2. Opis programu opracowanego (kody Źródłowe, zrzuty ekranu)

```
[1]: import matplotlib.pyplot as plt
import numpy as np

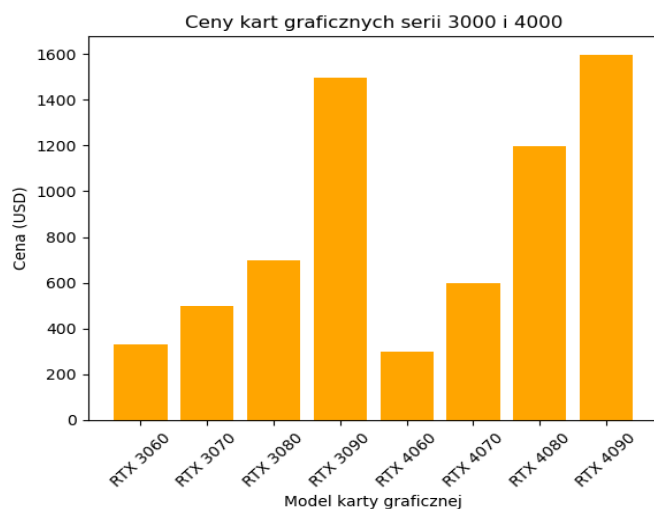
[3]: #WYKRES LINIOWY
# Dane o kartach graficznych i ich cenach
x = ["RTX 4060", "RTX 4070", "RTX 4080", "RTX 4090"]
y = [299, 599, 1199, 1599] # Cena w USD

# Tworzenie wykresu
plt.plot(x, y, marker='o', linestyle='-', color='b', label="Cena GPU")
plt.xlabel("Model karty graficznej (seria 4000)")
plt.ylabel("Cena (USD)")
plt.title("Wzrost cen kart graficznych serii 4000")
plt.legend()
plt.grid(True)
plt.show()
```



```
[5]: #WYKRES SŁUPKOWY
# Dane o kartach graficznych serii 3000 i 4000 oraz ich cenach
kategorie = [
    "RTX 3060", "RTX 3070", "RTX 3080", "RTX 3090",
    "RTX 4060", "RTX 4070", "RTX 4080", "RTX 4090"
]
wartosci = [
    329, 499, 699, 1499, 299, 599, 1199, 1599 # Ceny w USD
]

# Tworzenie wykresu słupkowego
plt.bar(kategorie, wartosci, color='orange')
plt.xlabel("Model karty graficznej")
plt.ylabel("Cena (USD)")
plt.title("Ceny kart graficznych serii 3000 i 4000")
plt.xticks(rotation=45)
plt.show()
```

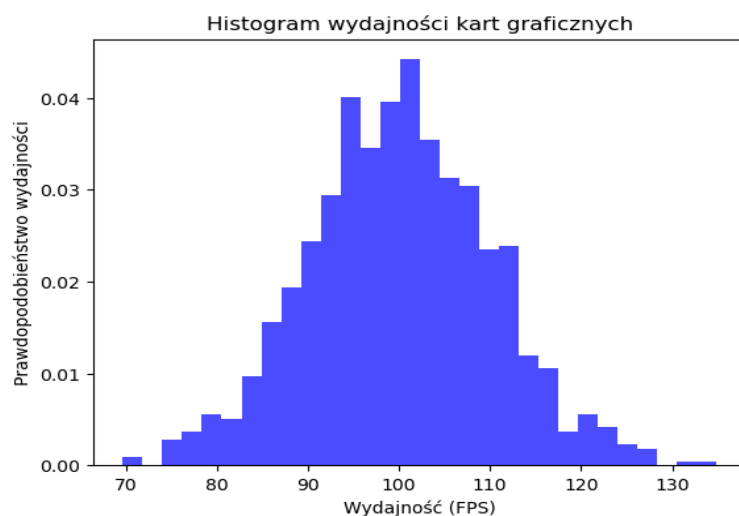


```
[7]: #HISTOGRAM
# Ręcznie generowane dane dotyczące wydajności kart graficznych FPS
# Średnia wydajność to 100 FPS, a odchylenie standardowe to 10 FPS
dane = np.random.normal(100, 10, 1000)

# Tworzenie histogramu
plt.hist(dane, bins=30, color='blue', alpha=0.7, density=True)

# Etykiety i tytuł wykresu
plt.xlabel("Wydajność (FPS)")
plt.ylabel("Prawdopodobieństwo wydajności")
plt.title("Histogram wydajności kart graficznych")

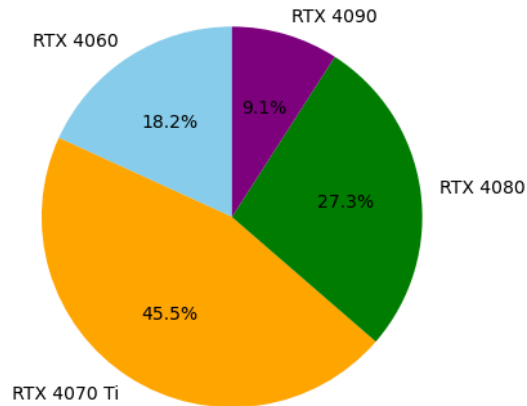
# Wyświetlenie wykresu
plt.show()
```



```
[9]: #WYKRES KOŁOWY
# Dane dotyczące sprzedaży kart graficznych z serii RTX 4000
modele = ['RTX 4060', 'RTX 4070 Ti', 'RTX 4080', 'RTX 4090']
sprzedaz = [20000, 50000, 30000, 10000] # Przykładowe dane

# Tworzenie wykresu kołowego
plt.pie(sprzedaz, labels=modele, autopct='%1.1f%%', startangle=90, colors=['skyblue', 'orange', 'green', 'purple'])
plt.title("Udział sprzedaży kart graficznych NVIDIA RTX 4000")
plt.show()
```

Udział sprzedaży kart graficznych NVIDIA RTX 4000



```
[11]: # KOLORY, LINIE I STYLE MARKERÓW
# Liczba rdzeni CUDA dla kart RTX 4060, 4070, 4080 i 4090
rdzenie_cuda = [3584, 5888, 8704, 10496]

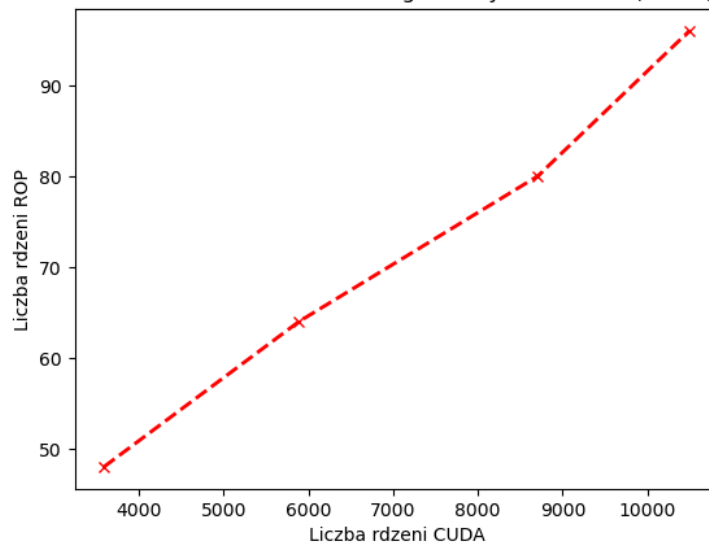
# Liczba rdzeni ROP dla kart RTX 4060, 4070, 4080 i 4090 (fikcyjne dane)
rdzenie_rop = [48, 64, 80, 96]

# Dostosowany wykres liniowy
plt.plot(rdzenie_cuda, rdzenie_rop, color='red', marker='x', linestyle='--', linewidth=2)

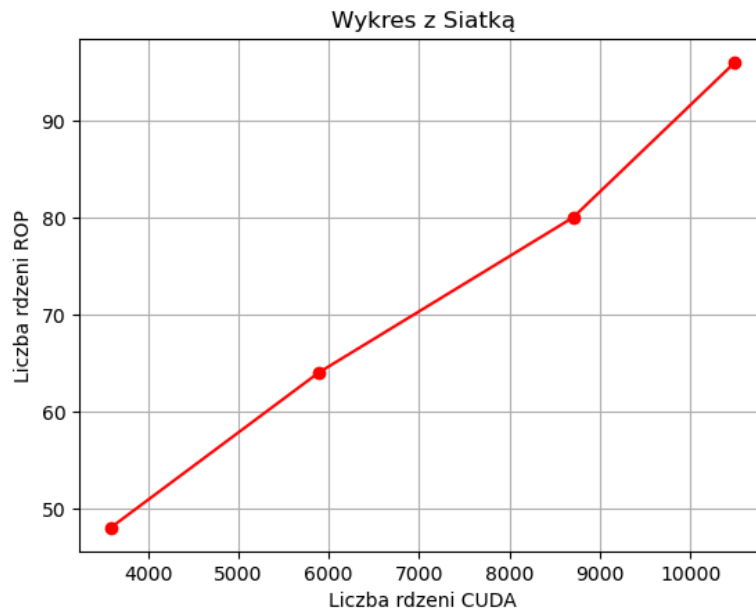
# Etykiety i tytuł wykresu
plt.xlabel("Liczba rdzeni CUDA")
plt.ylabel("Liczba rdzeni ROP")
plt.title("Porównanie rdzeni CUDA i ROP w kartach graficznych RTX 4060, 4070, 4080, 4090")

# Wyświetlenie wykresu
plt.show()
```

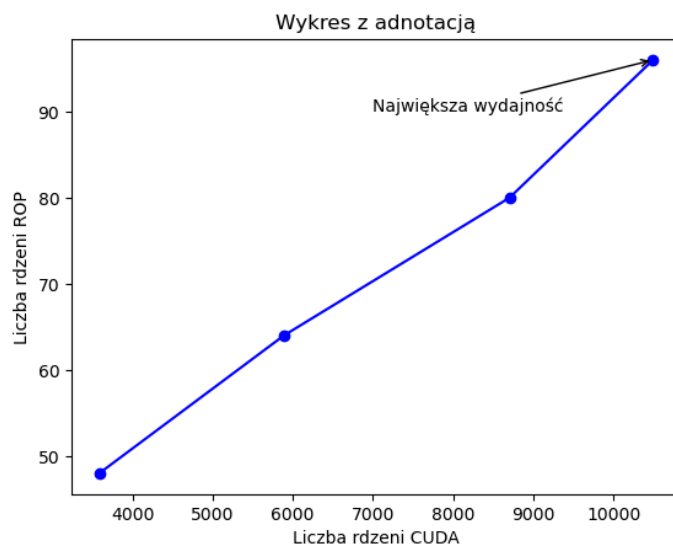
Porównanie rdzeni CUDA i ROP w kartach graficznych RTX 4060, 4070, 4080, 4090



```
[13]: #DODAWANIE SIATKI
plt.plot(rdzenie_cuda, rdzenie_rop, color='red', marker='o')
plt.grid(True)
plt.xlabel("Liczba rdzeni CUDA")
plt.ylabel("Liczba rdzeni ROP")
plt.title("Wykres z Siatką")
plt.show()
```



```
: #DODAWANIE ADNOTACJI
plt.plot(rdzenie_cuda, rdzenie_rop, color='blue', marker='o')
plt.annotate('Największa wydajność', xy=(10496, 96), xytext=(7000, 90), arrowprops=dict(facecolor='black', arrowstyle='->'))
plt.xlabel("Liczba rdzeni CUDA")
plt.ylabel("Liczba rdzeni ROP")
plt.title("Wykres z adnotacją")
plt.show()
```



```
#WYKRES PUNKTOWY
# Dane: Wydajność (TFLOPS), Liczba rdzeni CUDA i ceny kart graficznych z serii 2000, 3000 i 4000
#"RTX 2060", "RTX 2070", "RTX 2080", "RTX 3060", "RTX 3070", "RTX 3080", "RTX 4060", "RTX 4070", "RTX 4080", "RTX 4090"
wydajnosc_tflops = [6.5, 7.5, 10.0, 13.0, 20.0, 29.8, 20.0, 25.0, 35.0, 40.0]
rdzenie_cuda = [1920, 2304, 2944, 3584, 5888, 8704, 3584, 5888, 8704, 10496] # Liczba rdzeni CUDA
ceny = [350, 500, 700, 800, 1100, 1500, 600, 800, 1200, 1600] # Przykładowe ceny w dolarach

sizes = np.array(rdzenie_cuda) / 10 # Mniejszy współczynnik, aby rozmiar punktów nie był za duży

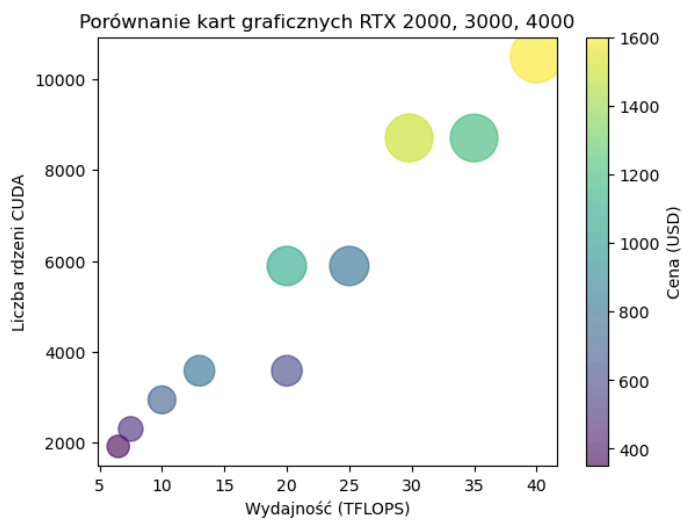
# Kolory punktów na podstawie ceny
colors = np.array(ceny)

# Tworzenie wykresu punktowego
plt.scatter(wydajnosc_tflops, rdzenie_cuda, s=sizes, c=colors, alpha=0.6, cmap='viridis')

# Dodanie paska kolorów
plt.colorbar(label='Cena (USD)')

# Etykiety i tytuł wykresu
plt.xlabel("Wydajność (TFLOPS)")
plt.ylabel("Liczba rdzeni CUDA")
plt.title("Porównanie kart graficznych RTX 2000, 3000, 4000")

# Wyświetlenie wykresu
plt.show()
```



```
[19]: #WYKRES 3D
from mpl_toolkits.mplot3d import Axes3D

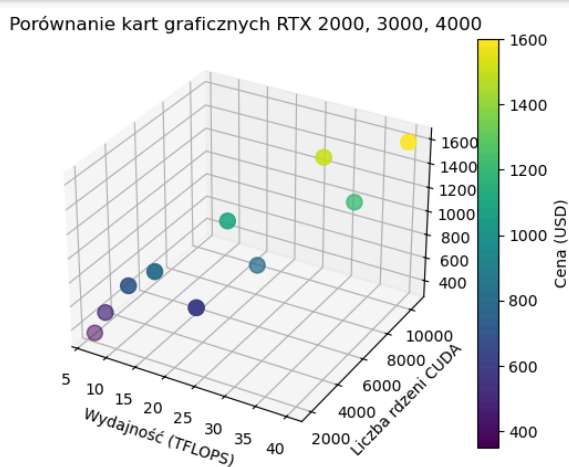
# Tworzenie wykresu 3D
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

# Wykres 3D, gdzie:
# X = wydajność w TFLOPS
# Y = Liczba rdzeni CUDA
# Z = cena kart graficznych
ax.scatter(wydajnosc_tflops, rdzenie_cuda, ceny, c=ceny, cmap='viridis', s=100)

# Etykiety i tytuł wykresu
ax.set_xlabel('Wydajność (TFLOPS)')
ax.set_ylabel('Liczba rdzeni CUDA')
ax.set_zlabel('Cena (USD)')
ax.set_title('Porównanie kart graficznych RTX 2000, 3000, 4000')

# Dodanie paska kolorów
fig.colorbar(ax.scatter(wydajnosc_tflops, rdzenie_cuda, ceny, c=ceny, cmap='viridis', s=100), ax=ax, label='Cena (USD)')

# Wyświetlenie wykresu
plt.show()
```



3. Wnioski

Wykonanie różnych typów wykresów w Jupyterze umożliwia lepszą wizualizację danych, co ułatwia ich interpretację i analizę. Wykresy liniowe i punktowe świetnie nadają się do analizy zależności między zmiennymi, podczas gdy wykresy słupkowe i kołowe pomagają porównać kategorie lub rozkład procentowy. Histogramy dostarczają informacji o rozkładzie danych, a wykresy 3D umożliwiają prezentację bardziej złożonych zależności w trzech wymiarach. Dodanie elementów takich jak siatka czy adnotacje poprawia czytelność wykresów i ułatwia komunikację wyników.