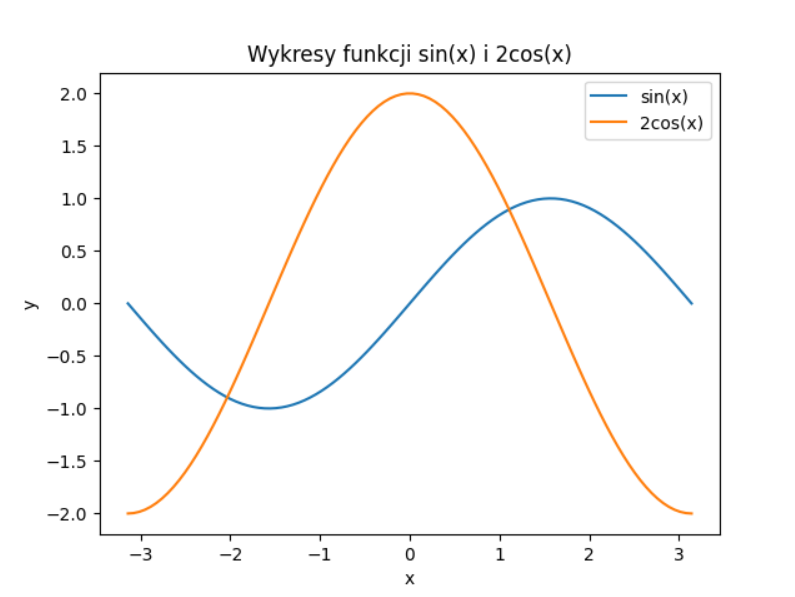
|  |  |  |  |
| --- | --- | --- | --- |
| Obraz zawierający symbol, design  Opis wygenerowany automatycznie | Politechnika Bydgoska im. J. J. Śniadeckich  **Wydział Telekomunikacji,**  **Informatyki i Elektrotechniki** | | Obraz zawierający symbol, logo, Grafika, Czcionka  Opis wygenerowany automatycznie |
| **Przedmiot** | Skryptowe języki programowania | | |
| **Prowadzący** | mgr inż. Martyna Tarczewska | | |
| **Temat** | *Matplotlib* | | |
| **Student** | Marcin Ogórkiewicz | | |
| **Nr ćw.** | 10 | **Data wykonania** | 06.12.2023 |
| **Ocena** |  | **Data oddania spr.** | 06.12.2023 |

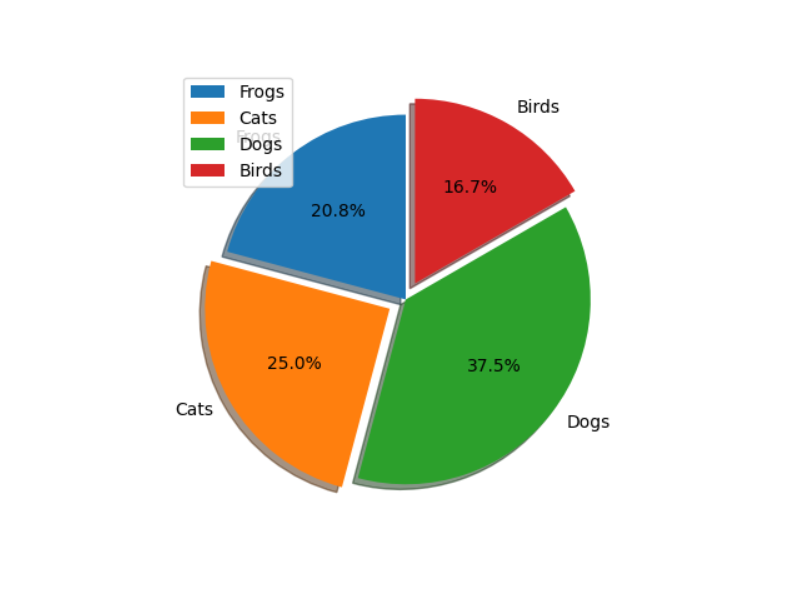
**Zadanie 1.**

#!C:\Users\koksu\AppData\Local\Programs\Python\Python310  
  
# importy  
import typing  
import matplotlib.pyplot as plt  
import numpy as np  
# stałe i zmienne globalne  
  
# funkcje  
  
  
def main() -> None:  
 x = np.linspace(-np.pi, np.pi, 100)  
 y\_sin = np.sin(x)  
 y\_cos = 2 \* np.cos(x)  
 plt.plot(x, y\_sin, label='sin(x)')  
 plt.plot(x, y\_cos, label='2cos(x)')  
 plt.title('Wykresy funkcji sin(x) i 2cos(x)')  
 plt.xlabel('x')  
 plt.ylabel('y')  
 plt.legend()  
 plt.show()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

****

**Zadanie 2**

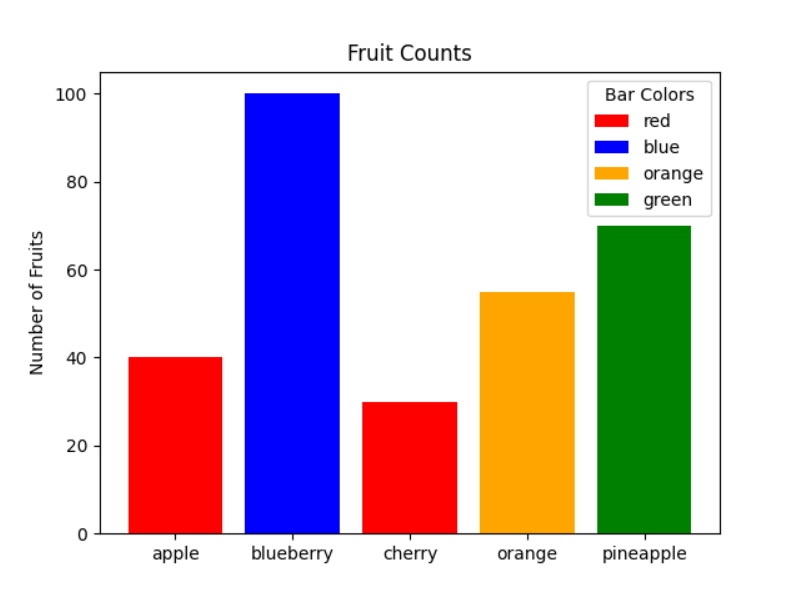
#!C:\Users\koksu\AppData\Local\Programs\Python\Python310  
  
# importy  
import typing  
import matplotlib.pyplot as plt  
# stałe i zmienne globalne  
  
# funkcje  
  
  
def main() -> None:  
 labels = 'Frogs', 'Cats', 'Dogs', 'Birds'  
 sizes = [25, 30, 45, 20]  
 explode = (0, 0.1, 0, 0.1) # Dodane przesunięcia dla sektorów  
 shadow = True # Dodany cień  
 start\_angle = 90 # Zmieniony kąt początkowy  
 fig1, ax1 = plt.subplots()  
 ax1.pie(  
 sizes,  
 explode=explode,  
 labels=labels,  
 autopct='%1.1f%%',  
 shadow=shadow,  
 startangle=start\_angle  
 )  
 ax1.legend(labels, loc="best")  
 plt.show()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

****

**Zadanie 3.**

Podkreślenie pomija dodanie nowej etykiety w spisie.

#!C:\Users\koksu\AppData\Local\Programs\Python\Python310  
  
# importy  
import typing  
import matplotlib.pyplot as plt  
# stałe i zmienne globalne  
  
# funkcje  
  
  
def main() -> None:  
 fig, ax = plt.subplots()  
 fruits = ['apple', 'blueberry', 'cherry', 'orange', 'pineapple']  
 counts = [40, 100, 30, 55, 70]  
 bar\_labels = ['red', 'blue', '\_red', 'orange', 'green']  
 bar\_colors = ['red', 'blue', 'red', 'orange', 'green']  
 ax.bar(fruits, counts, label=bar\_labels, color=bar\_colors)  
 ax.set\_ylabel('Number of Fruits')  
 ax.set\_title('Fruit Counts')  
 ax.legend(title='Bar Colors')  
 plt.show()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

****

**Zadanie 4.**

#!C:\Users\koksu\AppData\Local\Programs\Python\Python310  
  
# importy  
import typing  
import matplotlib.pyplot as plt  
import numpy as np  
# stałe i zmienne globalne  
  
# funkcje  
  
  
def main() -> None:  
 x = np.linspace(0, 9, 10)  
 y1 = np.random.randint(0, 100, size=10)  
 y2 = np.random.randint(0, 100, size=10)  
 y3 = np.random.randint(0, 100, size=10)  
 plt.plot(x, y1, 'gD-', label='Zielone diamenty')  
 plt.plot(x, y2, 'y\*--', label='Żółte gwiazdki')  
 plt.plot(x, y3, 'mp-.', label='Fioletowe pięciokąty')  
 plt.xlabel('Oś X')  
 plt.ylabel('Oś Y')  
 plt.title('Wykres liniowy z różnymi formatowaniami linii')  
 plt.legend()  
 plt.show()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

**Obraz zawierający linia, tekst, diagram, Wykres

Opis wygenerowany automatycznie**

**Zadanie 5.**

#!C:\Users\koksu\AppData\Local\Programs\Python\Python310  
  
# importy  
import typing  
import matplotlib.pyplot as plt  
import numpy as np  
# stałe i zmienne globalne  
  
# funkcje  
  
  
def main() -> None:  
 data = np.random.randint(0, 10, 100)  
 unique\_values, counts = np.unique(data, return\_counts=True)  
 plt.figure(figsize=(10, 5))  
 plt.subplot(1, 2, 1)  
 plt.bar(unique\_values, counts, color='blue')  
 plt.title('Wykres Słupkowy')  
 plt.xlabel('Wartość')  
 plt.ylabel('Liczba Powtórzeń')  
 plt.subplot(1, 2, 2)  
 plt.hist(data, bins=np.arange(11) - 0.5, color='green', edgecolor='black')  
 plt.title('Histogram')  
 plt.xlabel('Wartość')  
 plt.ylabel('Liczba Powtórzeń')  
 plt.show()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

**Obraz zawierający zrzut ekranu, diagram, Wykres, Wielobarwność

Opis wygenerowany automatycznie**

**Zadanie 6.**

Parametr loc wykorzystuje się do ustalenia lokalizacji legendy na wygenerowanym rysunku. Wartości słowne wybierają jedną ze zdefiniowanych pozycji dla legendy, a cyfry pozwalają na wprowadzenie dokładnych współrzędnych.

**Zadanie 7.**

#!C:\Users\koksu\AppData\Local\Programs\Python\Python310  
  
# importy  
import typing  
import matplotlib.pyplot as plt  
import numpy as np  
# stałe i zmienne globalne  
  
# funkcje  
  
  
def main() -> None:  
 x = np.linspace(0, 2 \* np.pi)  
 y1 = np.sin(x)  
 y2 = np.cos(x)  
 y3 = np.tan(x)  
 y4 = np.arctan(x)  
 plt.figure(figsize=(12, 8))  
 plt.subplot(2, 2, 1)  
 plt.plot(x, y1, label='sin(x)')  
 plt.title('sin(x)')  
 plt.legend()  
 plt.subplot(2, 2, 2)  
 plt.plot(x, y2, label='cos(x)', color='orange')  
 plt.title('cos(x)')  
 plt.legend()  
 plt.subplot(2, 2, 3)  
 plt.plot(x, y3, label='tan(x)', color='green')  
 plt.title('tan(x)')  
 plt.ylim(-5, 5) # Ustawienie ograniczenia osi Y dla lepszej widoczności  
 plt.legend()  
 plt.subplot(2, 2, 4)  
 plt.plot(x, y4, label='arctan(x)', color='red')  
 plt.title('arctan(x)')  
 plt.legend()  
 plt.tight\_layout()  
 plt.show()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

**Obraz zawierający diagram, Wykres, linia, tekst

Opis wygenerowany automatycznie**

**Zadanie 8.**

#!C:\Users\koksu\AppData\Local\Programs\Python\Python310  
  
# importy  
import typing  
import matplotlib.pyplot as plt  
import numpy as np  
# stałe i zmienne globalne  
  
# funkcje  
  
  
def main() -> None:  
 data = np.genfromtxt('../Lab09/oceny.csv', delimiter='\t', skip\_header=1)  
 lab\_grades = data[:, :-1]  
 flat\_lab\_grades = lab\_grades.flatten()  
 grades, grades\_counts = np.unique(flat\_lab\_grades, return\_counts=True)  
 plt.figure(figsize=(8, 8))  
 plt.pie(grades\_counts, labels=grades, autopct='%1.1f%%', startangle=90,  
 colors=['red', 'orange', 'yellow', 'green', 'blue', 'purple'])  
 plt.title('Liczba Poszczególnych Oceny z Laboratoriów')  
 plt.legend(grades, title='Oceny', loc="best")  
 plt.savefig('Ilość ocen.png')  
 plt.show()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

**Obraz zawierający tekst, zrzut ekranu, diagram, Wielobarwność

Opis wygenerowany automatycznie**

**Wnioski**

Ćwiczenie nauczyło mnie posługiwania się modułem Matplotlib.