

# L1\_Michał\_Szmuksta\_3\_4

October 6, 2022

## 1 Laboratorium 1

Michał Szmuksta, gr. 7

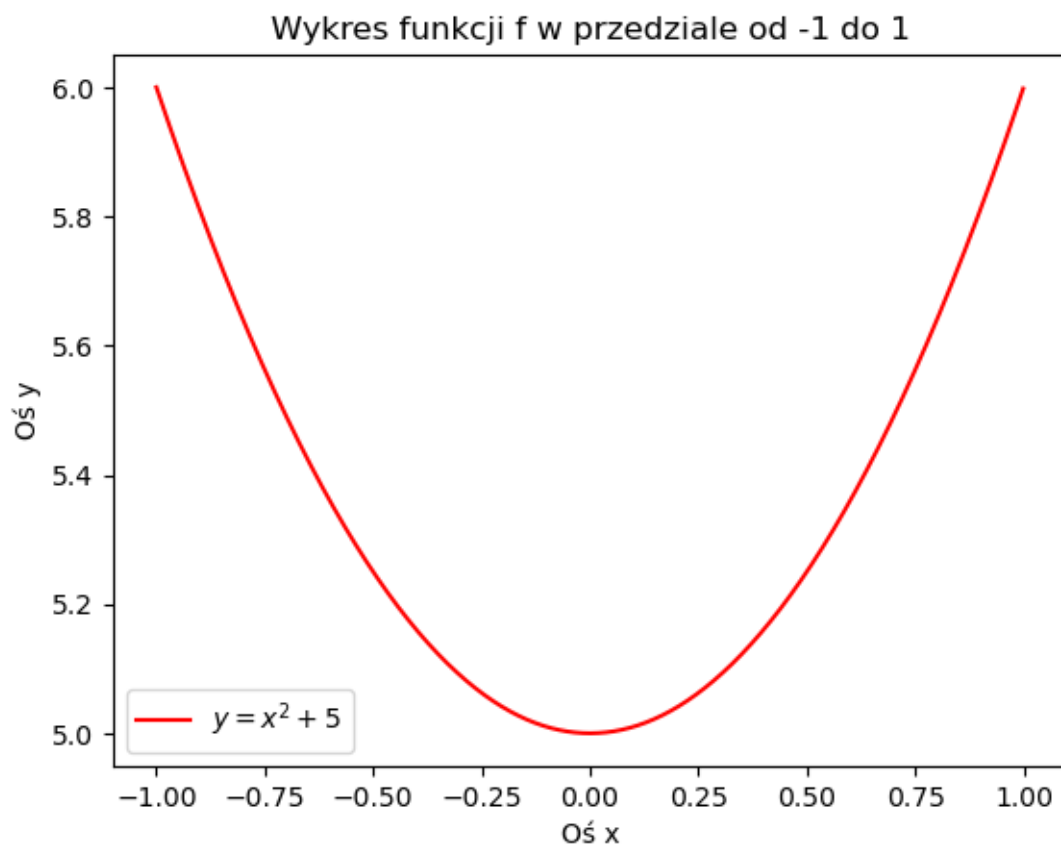
**Wstęp** Celem ćwiczenia było przypomnienie sobie podstawowych funkcji biblioteki matplotlib oraz poznanie obiektu dataframe z biblioteki pandas. Na początku zaimportowano niezbędne biblioteki:

```
[22]: import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

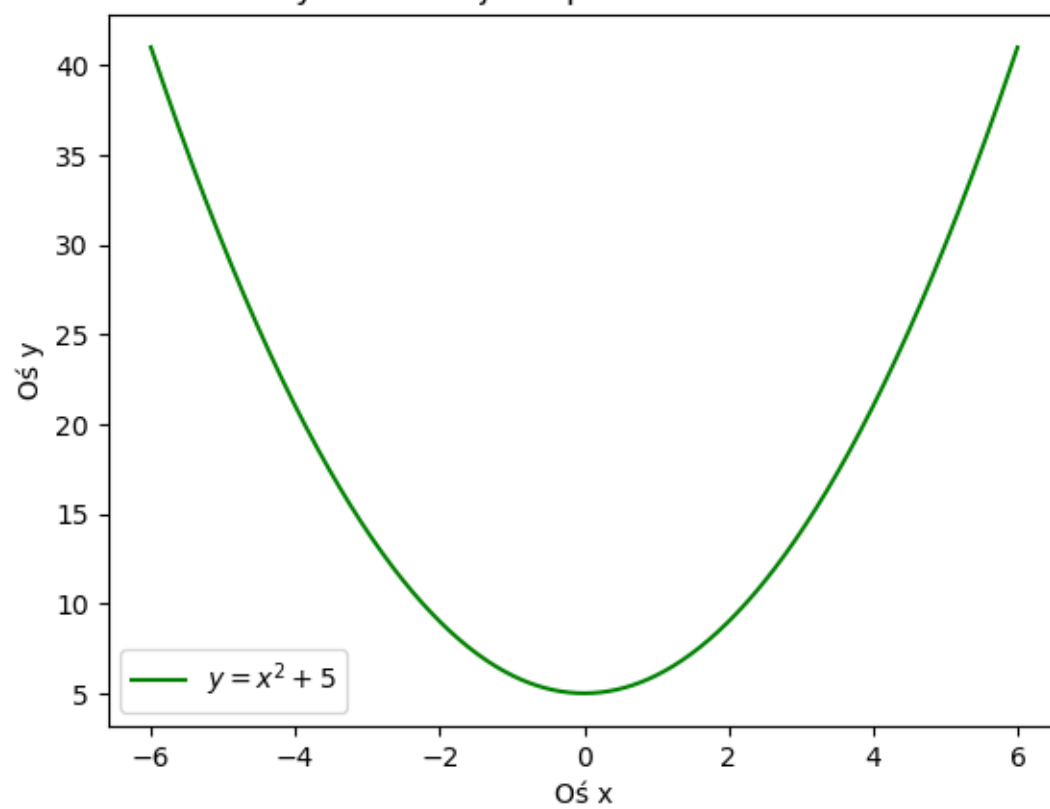
**Zadanie 3** Do generacji wykresów użyto napisanej funkcji lambda “f” opisującej  $f(x)$  oraz funkcji “plot” do generacji wykresów dla różnych parametrów wejściowych.

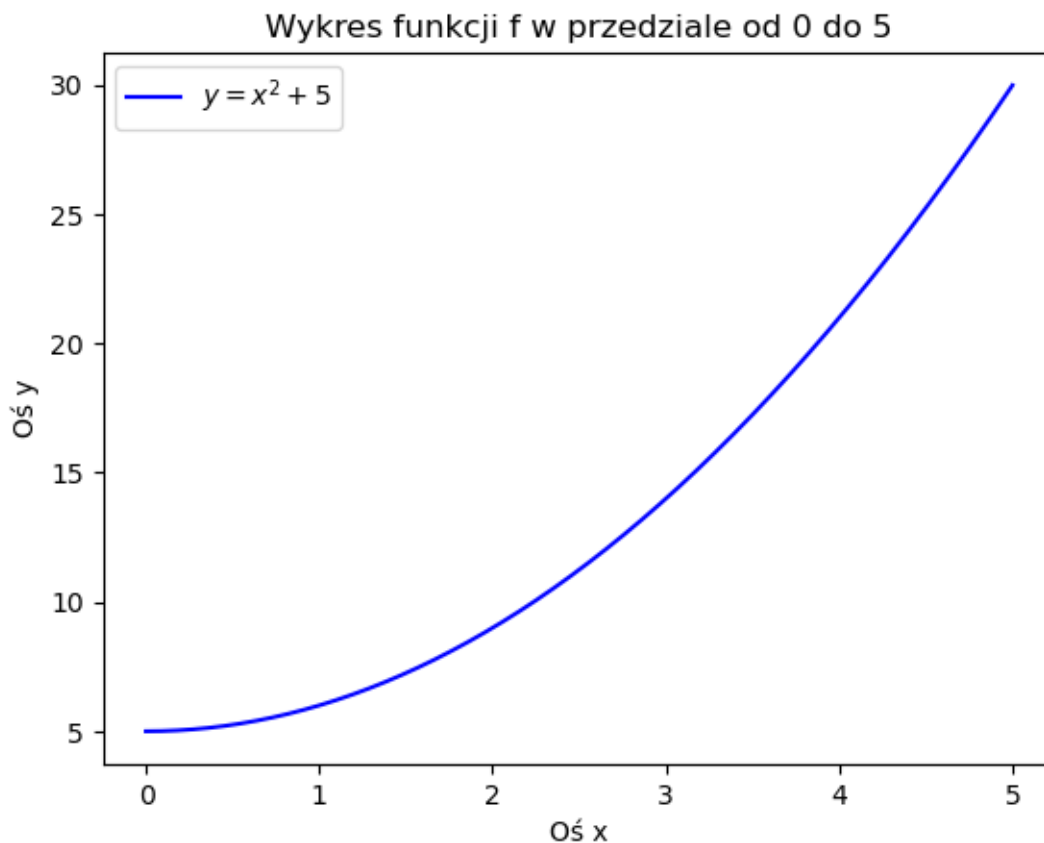
```
[23]: f = lambda x : x**2 + 5
def plot(function, min, max, step, xlabel, ylabel, title, legend, color):
    x = np.arange(min, max, step)
    y = function(x)
    plt.plot(x, y, label=legend, color=color)
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    plt.title(title)
    plt.legend()
    plt.show()
```

```
[24]: plot(f, -1, 1, 0.001, 'Oś x', 'Oś y', 'Wykres funkcji f w przedziale od -1 do 1', '$y = x^2 + 5$', 'r')
plot(f, -6, 6, 0.001, 'Oś x', 'Oś y', 'Wykres funkcji f w przedziale od -6 do 6', '$y = x^2 + 5$', 'g')
plot(f, 0, 5, 0.001, 'Oś x', 'Oś y', 'Wykres funkcji f w przedziale od 0 do 5', '$y = x^2 + 5$', 'b')
```



Wykres funkcji  $f$  w przedziale od -6 do 6





**Zadanie 4** Dataframe utworzono dodając wszystkie rekordy jako listy w konstruktorze klasy, a nazwy kolumn przekazano również w postaci listy jako dodatkowy argument.

```
[25]: column_names = ['Name', 'Surname', 'Age', 'Sex']
dataframe = pd.DataFrame([['Adam', 'Abramowicz', 32, 'Male'],
                           ['Bartosz', 'Bartosiewicz', 18, 'Male'],
                           ['Cyryl', 'Cęgielski', 44, 'Male'],
                           ['Dominik', 'Domagała', 12, 'Male'],
                           ['Engelbert', 'Edison', 87, 'Male']],
                           columns=column_names)
dataframe
```

```
[25]:
```

	Name	Surname	Age	Sex
0	Adam	Abramowicz	32	Male
1	Bartosz	Bartosiewicz	18	Male
2	Cyryl	Cęgielski	44	Male
3	Dominik	Domagała	12	Male
4	Engelbert	Edison	87	Male

Metoda “info” zwraca informację o wielkości struktury danych, typie danych w każdej kolumnie, ilości danych nie będących wartością niezdefiniowaną w kolumnie i zużyciu pamięci przez obiekt.

Metoda również sumuje liczbę poszczególnych typów danych w kolejnych kolumnach.

```
[26]: dataframe.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 4 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Name        5 non-null     object
1   Surname     5 non-null     object
2   Age         5 non-null     int64
3   Sex         5 non-null     object
dtypes: int64(1), object(3)
memory usage: 288.0+ bytes
```

Metodę describe użyto w trybie “all” aby zobaczyć wszystkie parametry. Metoda zwraca zliczenie wszystkich zdefiniowanych rekordów w każdej kolumnie, liczbę unikalnych rekordów w każdej kolumnie, najczęściej występującą daną, średnią z danych, odchylenie standardowe, wartość minimalną i maksymalną czy też kwantyle różnych rzędów w kolumnach.

```
[27]: dataframe.describe(include='all')
```

```
[27]:
```

	Name	Surname	Age	Sex
count	5	5	5.000000	5
unique	5	5	NaN	1
top	Adam	Abramowicz	NaN	Male
freq	1	1	NaN	5
mean	NaN	NaN	38.600000	NaN
std	NaN	NaN	29.779187	NaN
min	NaN	NaN	12.000000	NaN
25%	NaN	NaN	18.000000	NaN
50%	NaN	NaN	32.000000	NaN
75%	NaN	NaN	44.000000	NaN
max	NaN	NaN	87.000000	NaN

Metoda head zwraca zbiór pierwszych n rekordów podanych w argumencie funkcji.

```
[28]: dataframe.head(3)
```

```
[28]:
```

	Name	Surname	Age	Sex
0	Adam	Abramowicz	32	Male
1	Bartosz	Bartosiewicz	18	Male
2	Cyryl	Cęgielski	44	Male