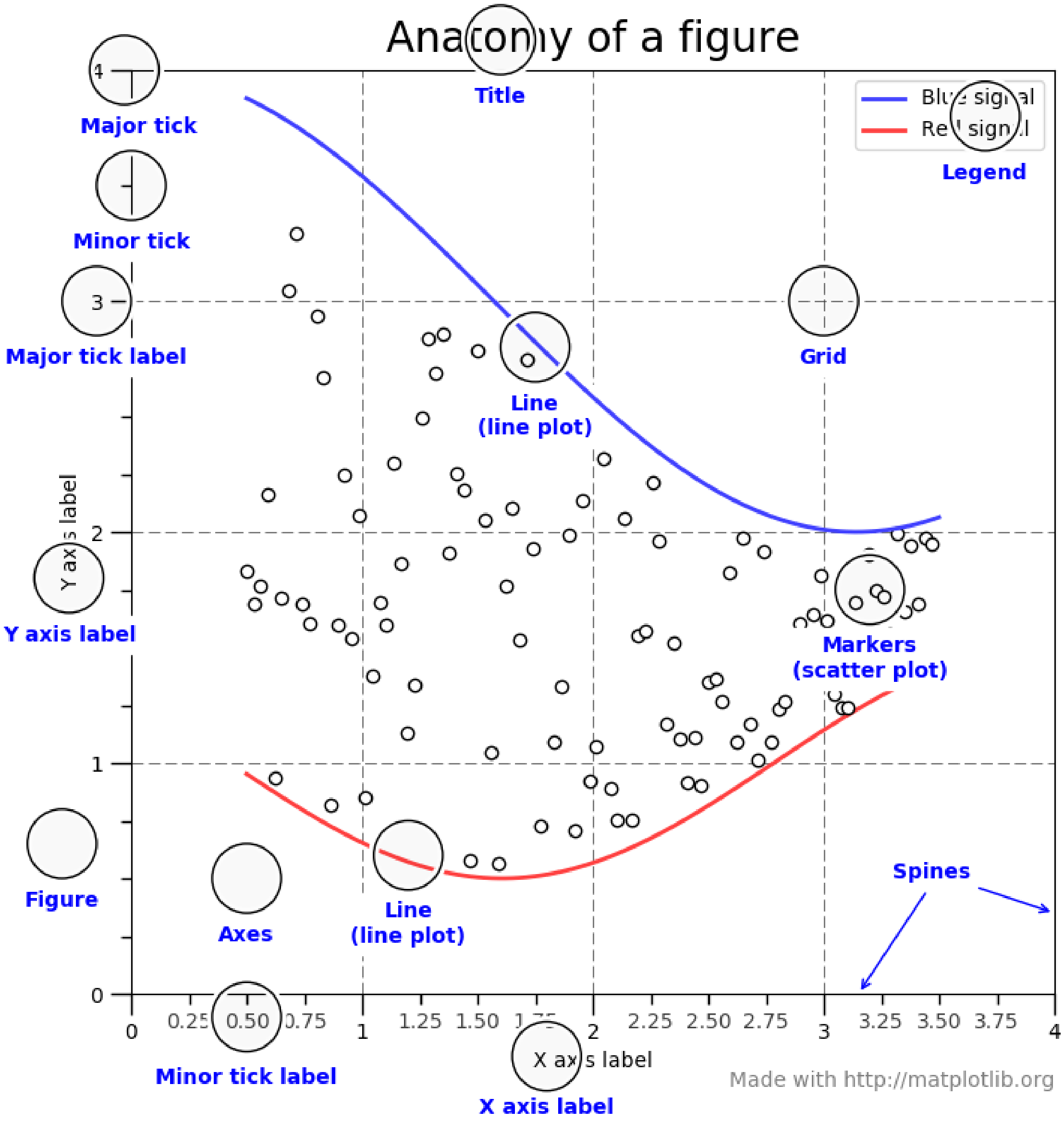
Wizualizacja Danych - Biblioteka matplotlib

Ostatnia aktualizacja: 2022-06-08 16:26:14

Matplotlib

https://matplotlib.org/



Import

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

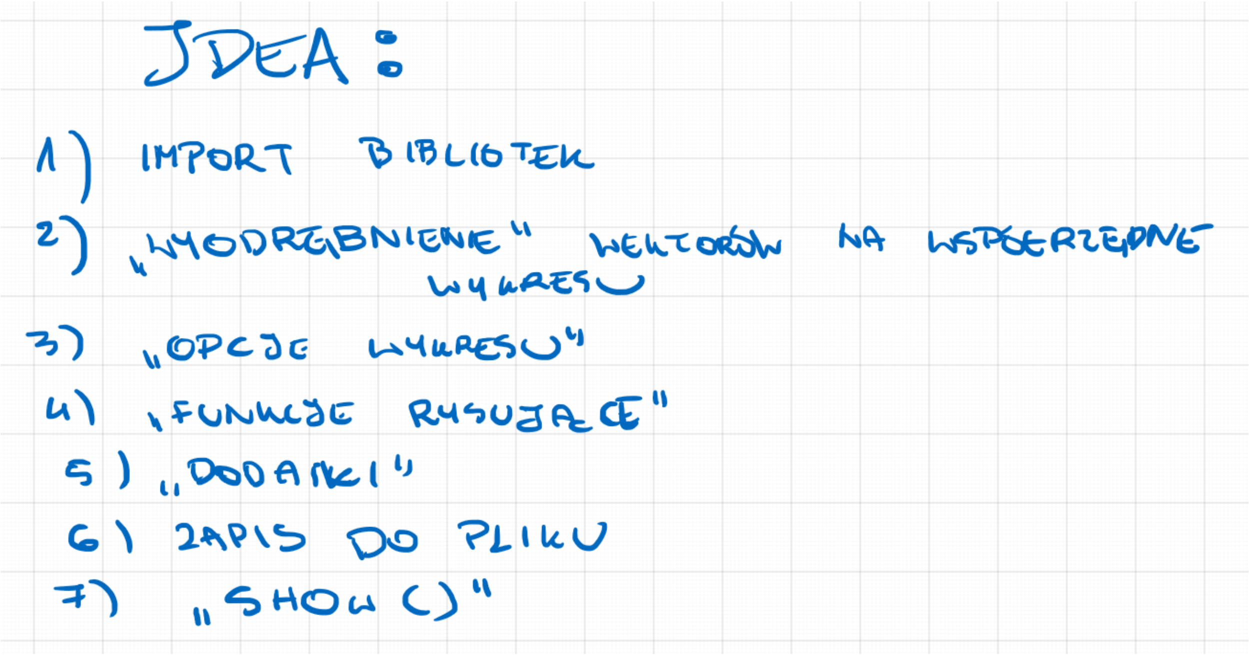
Galerie wykresów

https://matplotlib.org/gallery/index.html

https://python-graph-gallery.com/

https://github.com/rasbt/matplotlib-gallery

https://seaborn.pydata.org/examples/index.html



JDEA:

1) IMPORT BIBLIOTEK

2) „WYODRĘBNIENE" WEKTORÓW NA WSPÓŁRZĘDNE WYKRESU

3) „OPCJE WYKRESU"

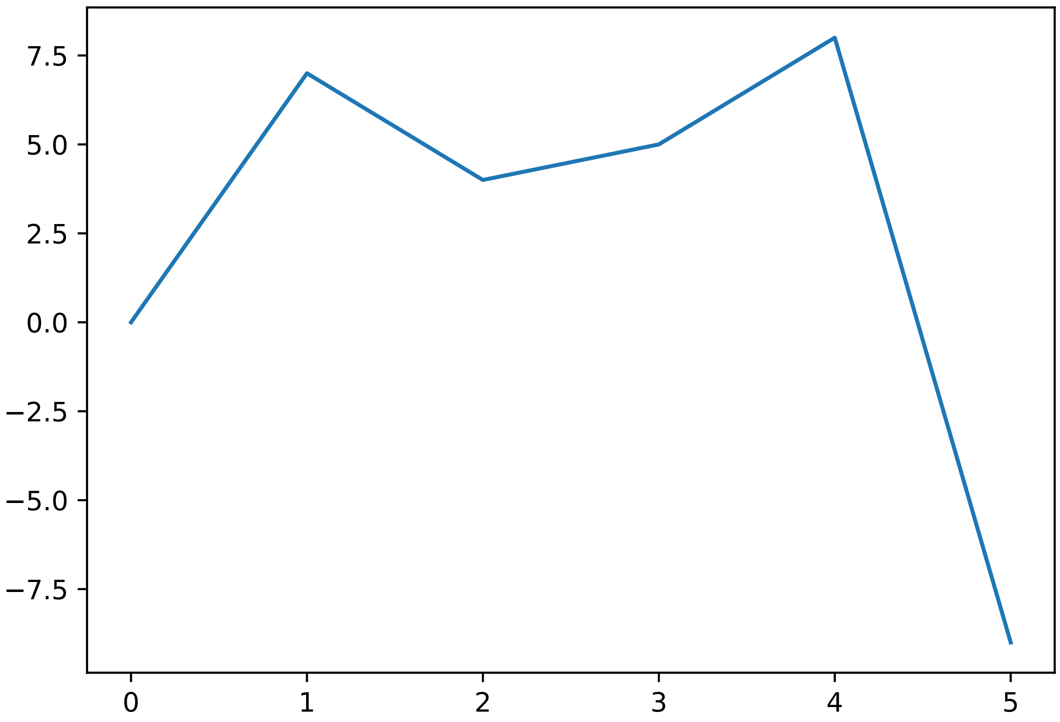
4) „FUNKCJE RYSUJĄCE"

5) „DODATKI"

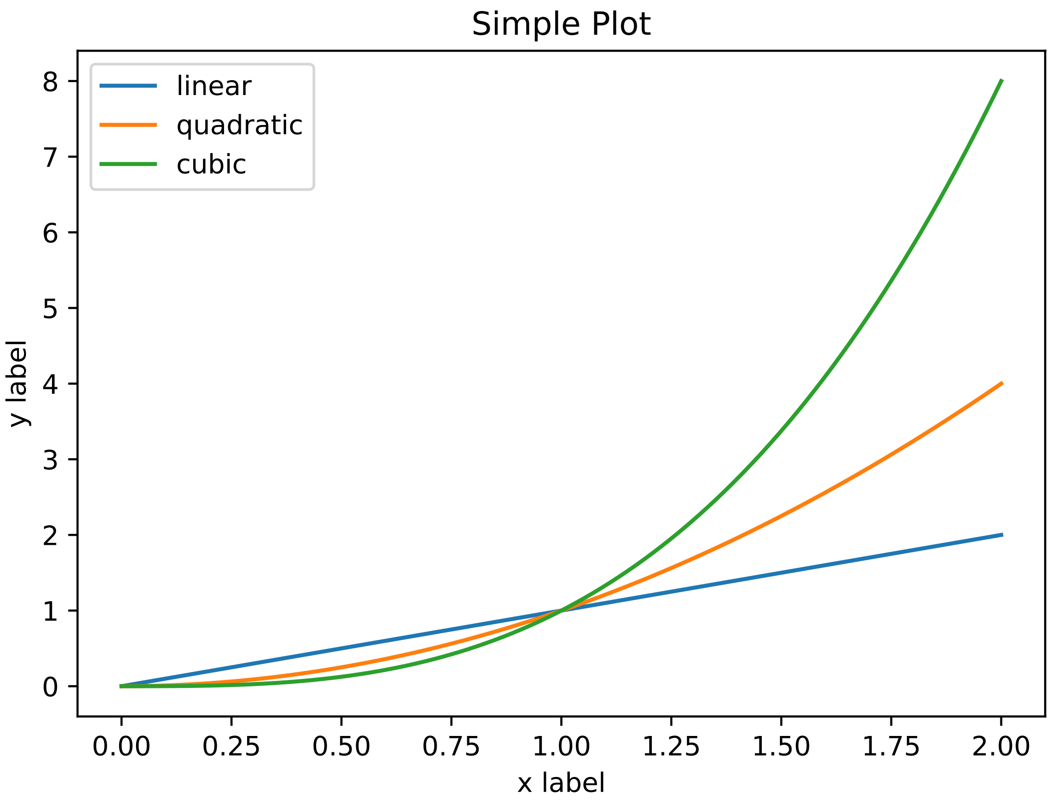
6) ZAPIS DO PLIKU

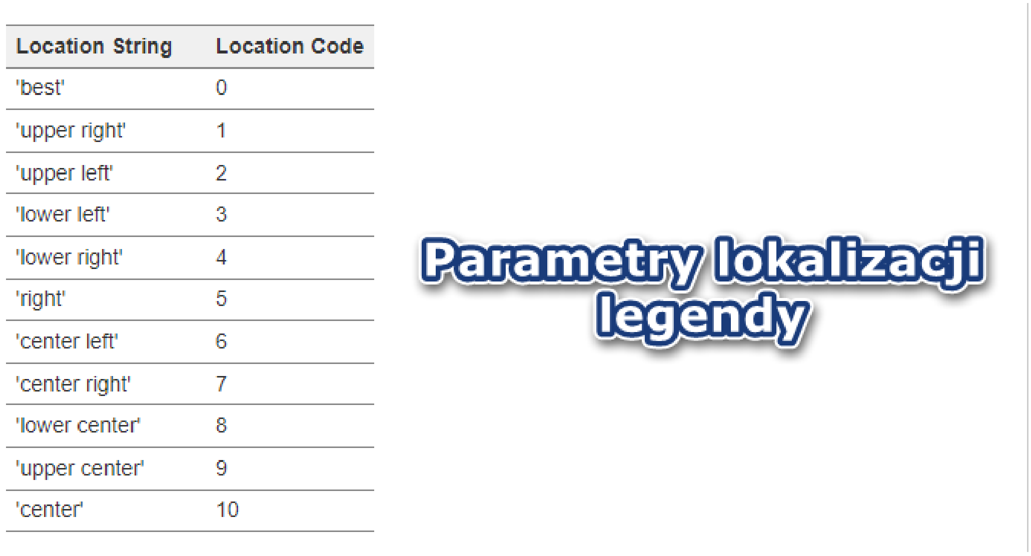
7) „SHOW()"

```
import matplotlib.pyplot as plt

x = [0, 7, 4, 5,8,-9]

plt.plot(x)

plt.show()
```



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

x = np.linspace(0, 2, 100)

plt.plot(x, x, label='linear')
plt.plot(x, x**2, label='quadratic')
plt.plot(x, x**3, label='cubic')

plt.xlabel('x label')
plt.ylabel('y label')

plt.title("Simple Plot")

plt.legend()

plt.show()
```

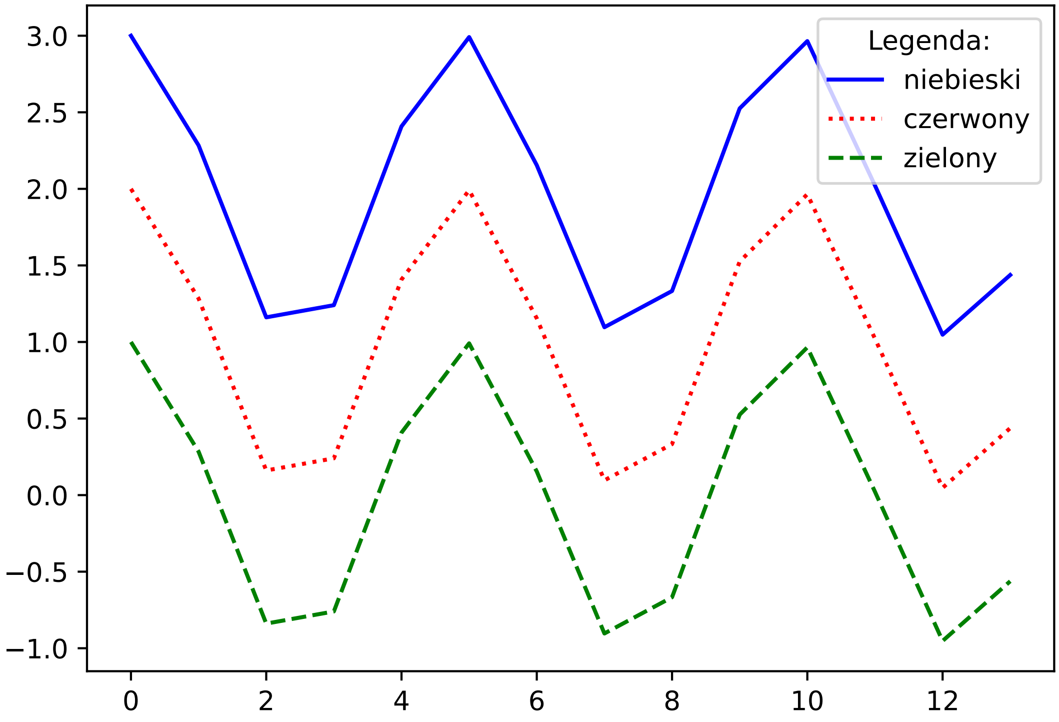


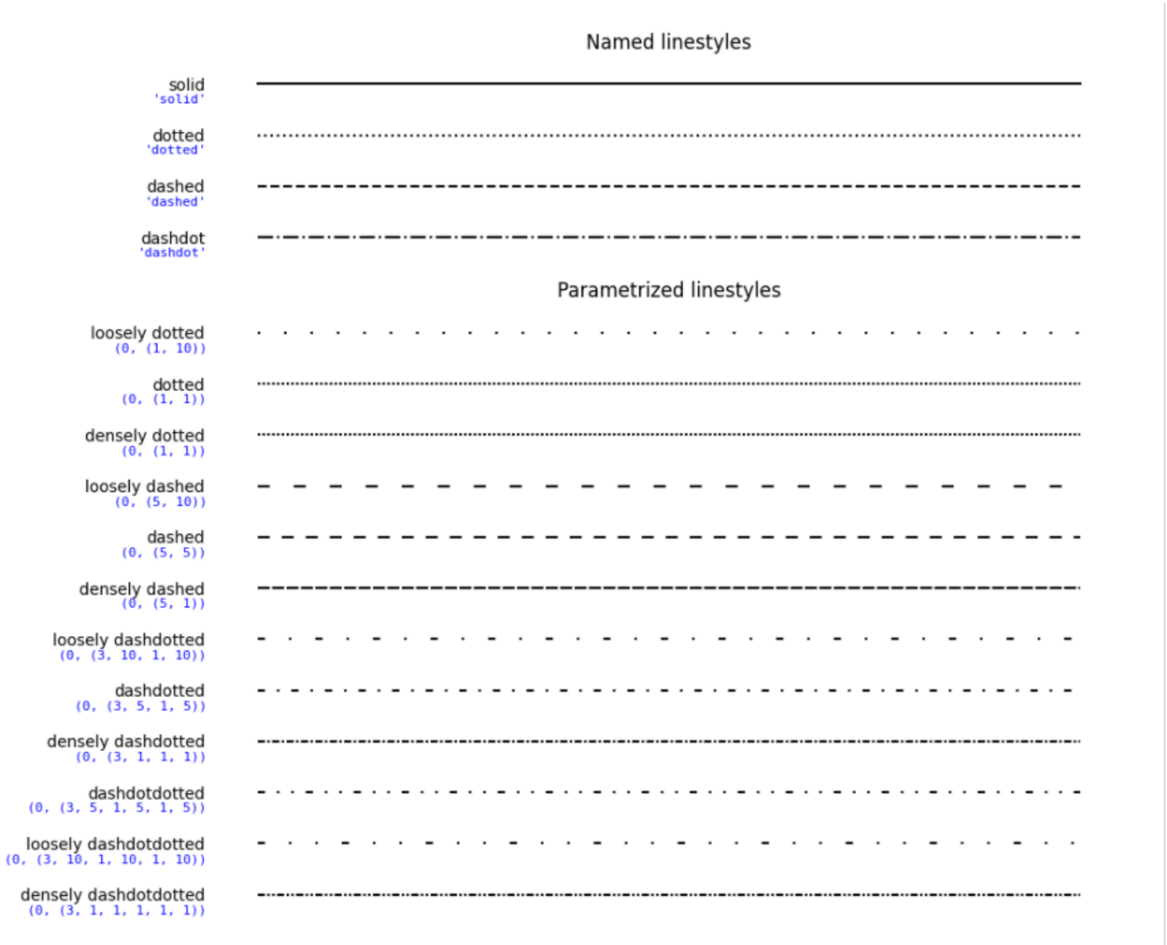


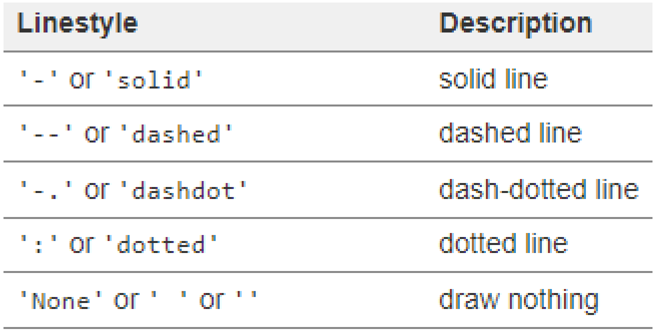
Parametry lokalizacji legendy

Location String	Location Code
'best'	0
'upper right'	1
'upper left'	2
'lower left'	3
'lower right'	4
'right'	5
'center left'	6
'center right'	7
'lower center'	8
'upper center'	9
'center'	10

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

x = np.arange(14)
y = np.cos(5 * x)

plt.plot(x, y + 2, 'blue', linestyle="-", label="niebieski")
plt.plot(x, y + 1, 'red', linestyle=":", label="czerwony")
plt.plot(x, y, 'green', linestyle="--", label="zielony")
plt.legend(title='Legenda:')
plt.show()
```

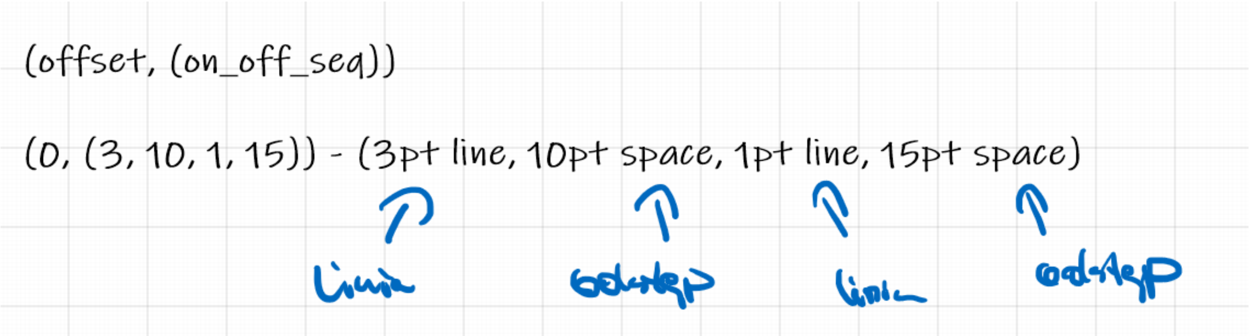






Linestyle	Description
'-' or 'solid'	solid line
'--' or 'dashed'	dashed line
'-.' or 'dashdot'	dash-dotted line
':' or 'dotted'	dotted line
'None' or ' ' or ''	draw nothing

Uproszczone

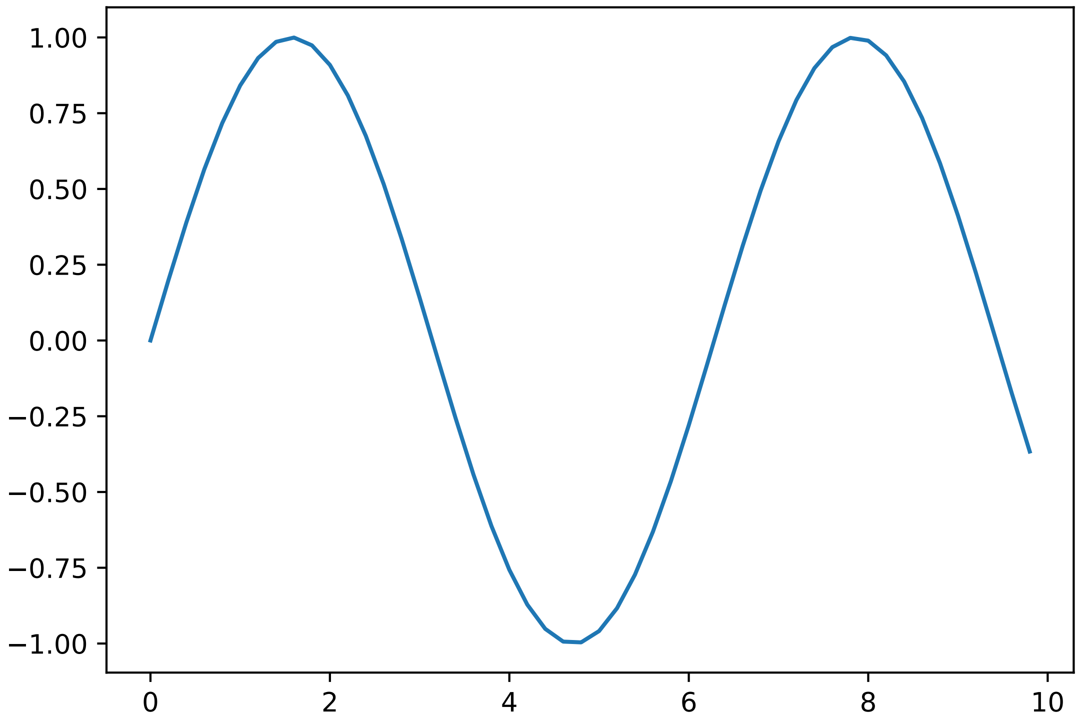


(offset, (on_off_seq))

(0, (3, 10, 1, 15)) - (3pt line, 10pt space, 1pt line, 15pt space)

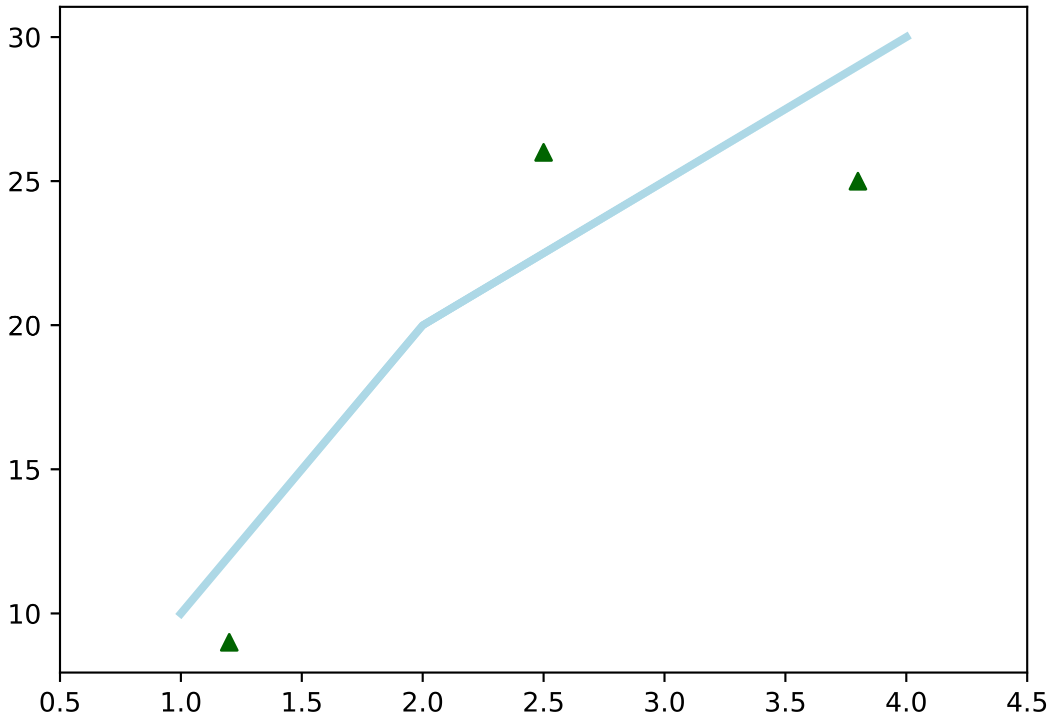
Linia
odstęp
linia
odstęp

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

x = np.arange(0, 10, 0.2)
y = np.sin(x)
fig, ax = plt.subplots()
ax.plot(x, y)
plt.show()
```

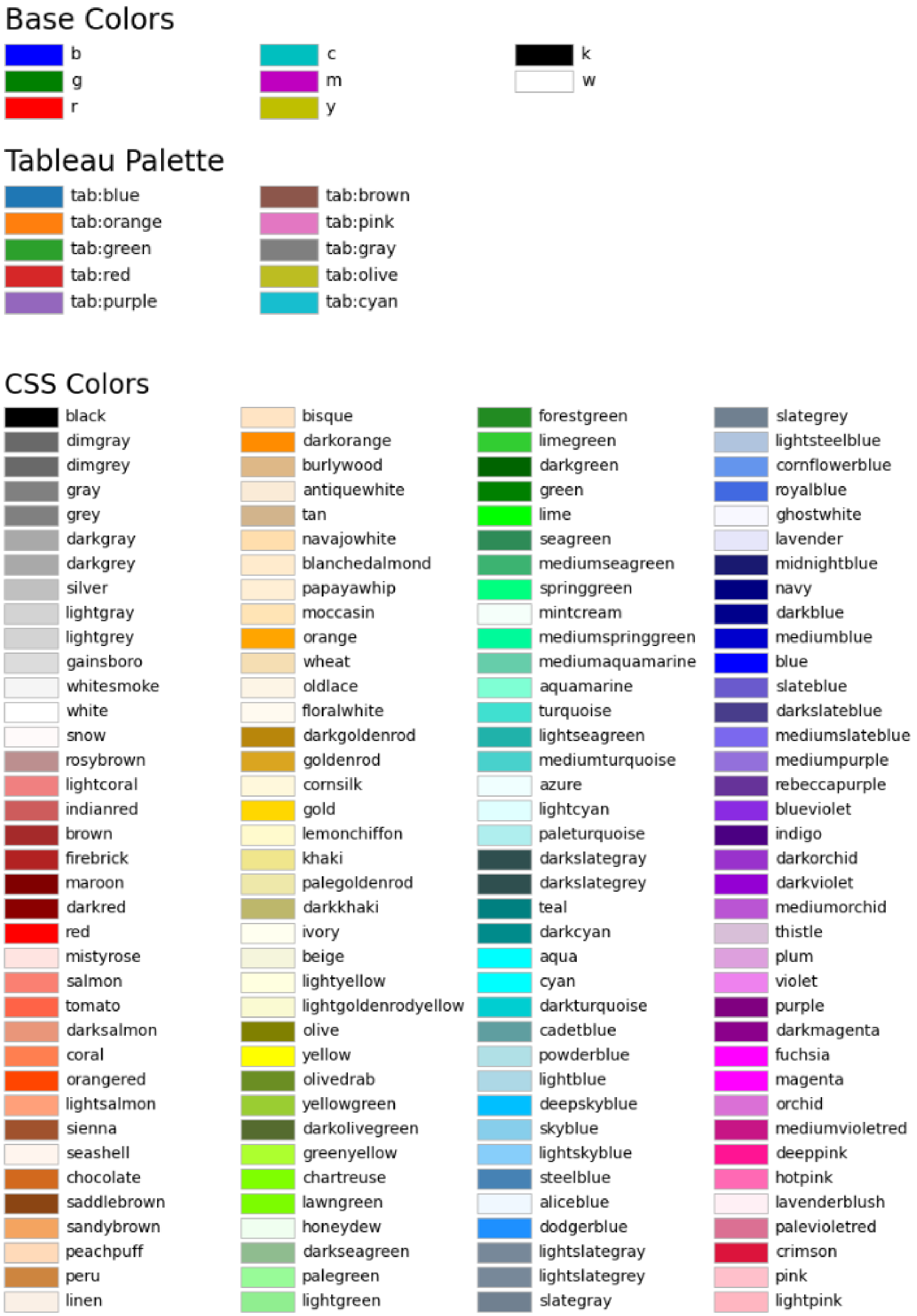


```
import matplotlib.pyplot as plt

fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot([1, 2, 3, 4], [10, 20, 25, 30], color='lightblue', linewidth=3)
ax.scatter([0.3, 3.8, 1.2, 2.5], [11, 25, 9, 26], color='darkgreen', marker='^')
ax.set_xlim(0.5, 4.5)
plt.show()
```

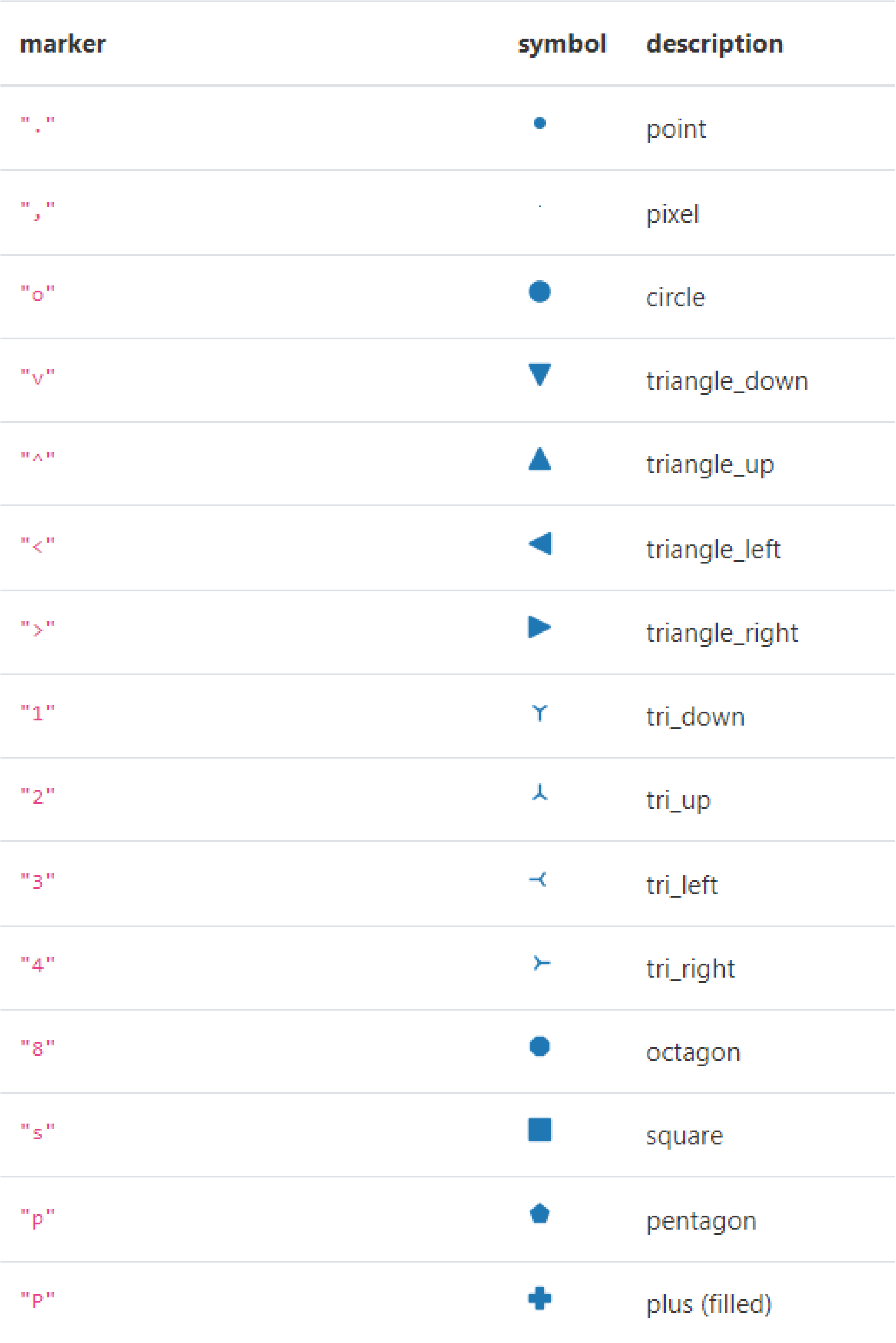
```
## (0.5, 4.5)
```



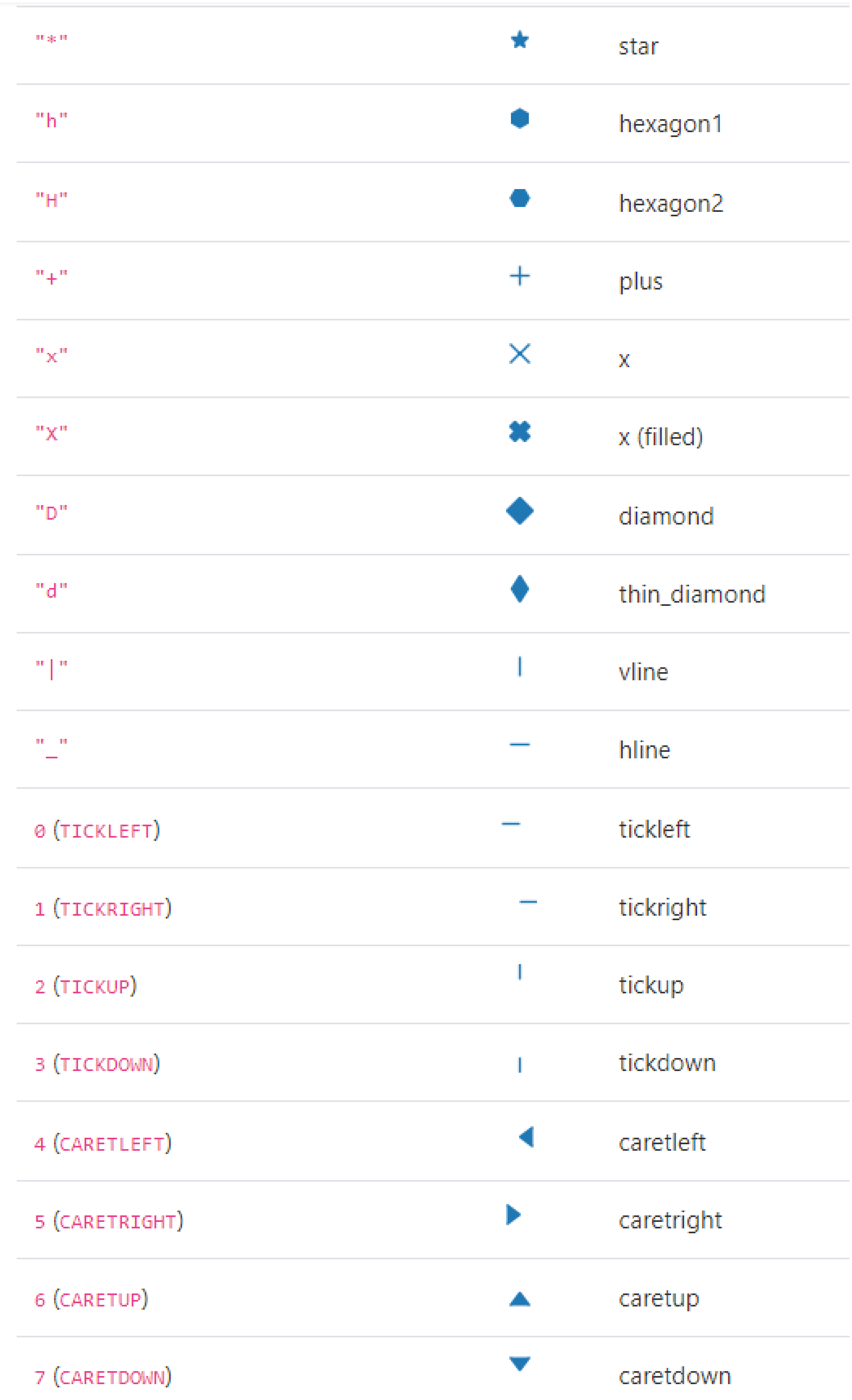
Kolory https://matplotlib.org/stable/gallery/color/named_colors.html



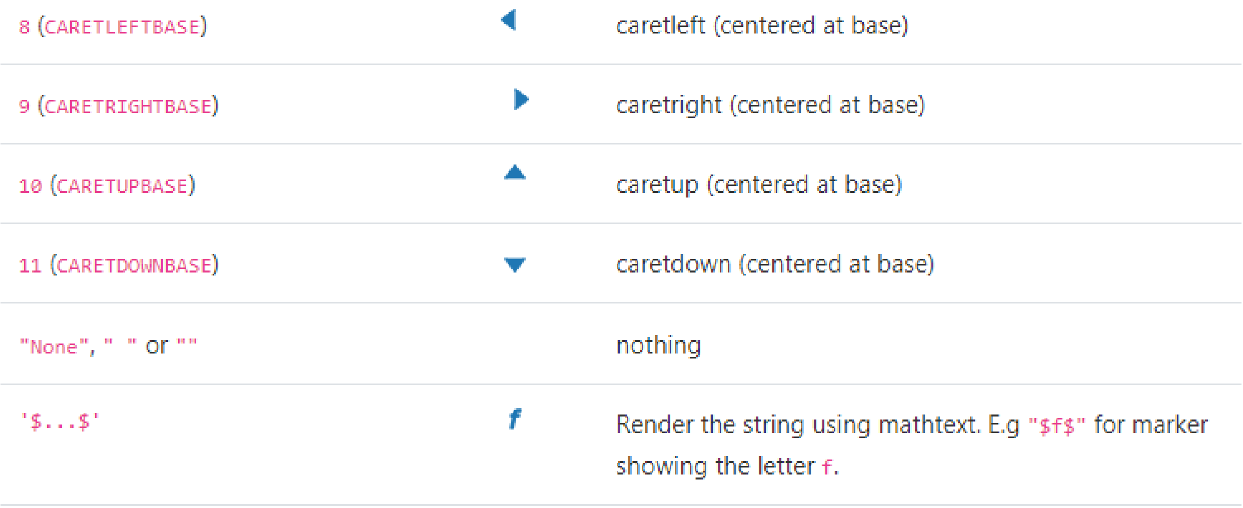
Markery https://matplotlib.org/stable/api/markers_api.html



marker	symbol	description
"."	•	point
","	·	pixel
"o"	●	circle
"v"	▼	triangle_down
"^"	▲	triangle_up
"<"	◀	triangle_left
">"	▶	triangle_right
"1"	Y	tri_down
"2"	⅄	tri_up
"3"	≺	tri_left
"4"	≻	tri_right
"8"	⯃	octagon
"s"	■	square
"p"	⬟	pentagon
"P"	✚	plus (filled)



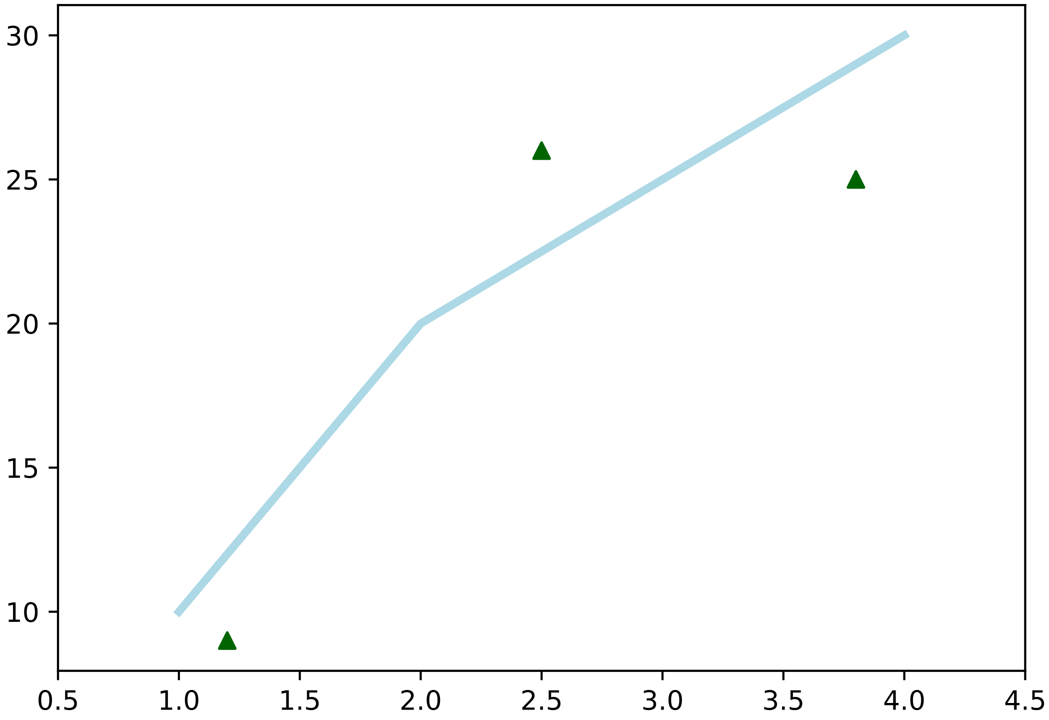
`"*"`	★	star
`"h"`	⬢	hexagon1
`"H"`	⬣	hexagon2
`"+"`	+	plus
`"x"`	×	x
`"X"`	✖	x (filled)
`"D"`	◆	diamond
`"d"`	♦	thin_diamond
`"|"`	|	vline
`"_"`	—	hline
`0` (`TICKLEFT`)	–	tickleft
`1` (`TICKRIGHT`)	–	tickright
`2` (`TICKUP`)	╵	tickup
`3` (`TICKDOWN`)	╷	tickdown
`4` (`CARETLEFT`)	◀	caretleft
`5` (`CARETRIGHT`)	▶	caretright
`6` (`CARETUP`)	▲	caretup
`7` (`CARETDOWN`)	▼	caretdown



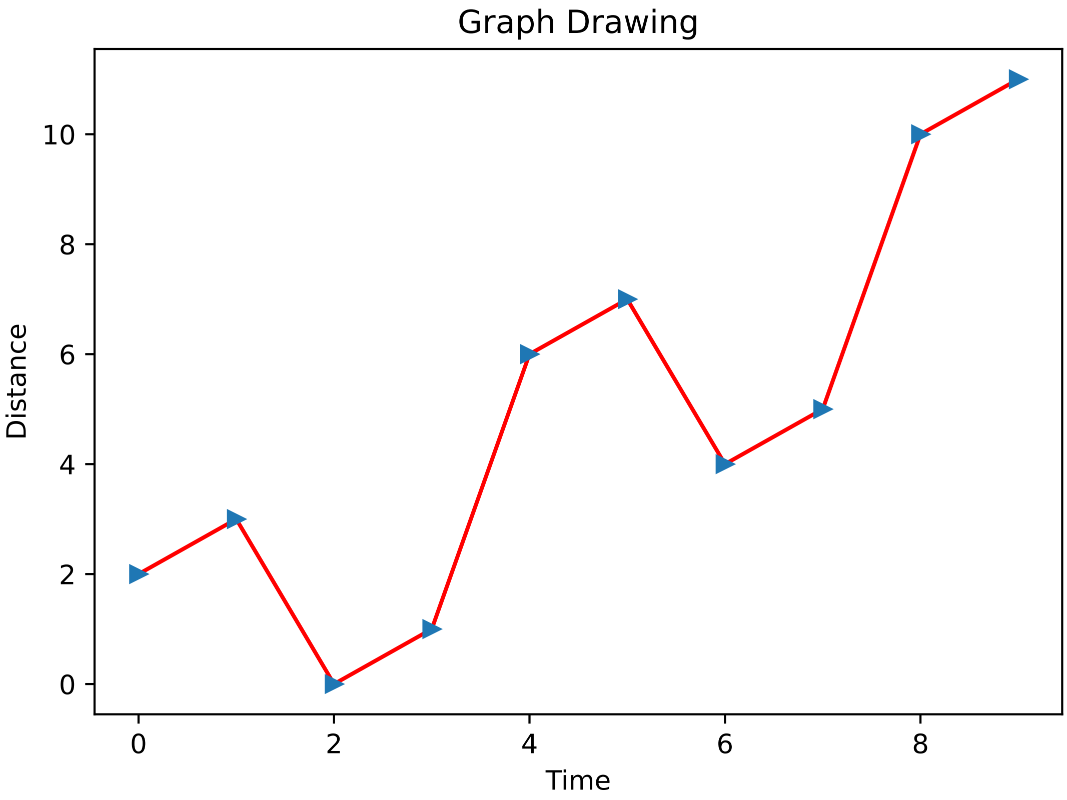
`8` (`CARETLEFTBASE`)	◀	caretleft (centered at base)
`9` (`CARETRIGHTBASE`)	▶	caretright (centered at base)
`10` (`CARETUPBASE`)	▲	caretup (centered at base)
`11` (`CARETDOWNBASE`)	▼	caretdown (centered at base)
`"None"`, `" "` or `""`		nothing
`'$...$'`	f	Render the string using mathtext. E.g `"$f$"` for marker showing the letter `f`.

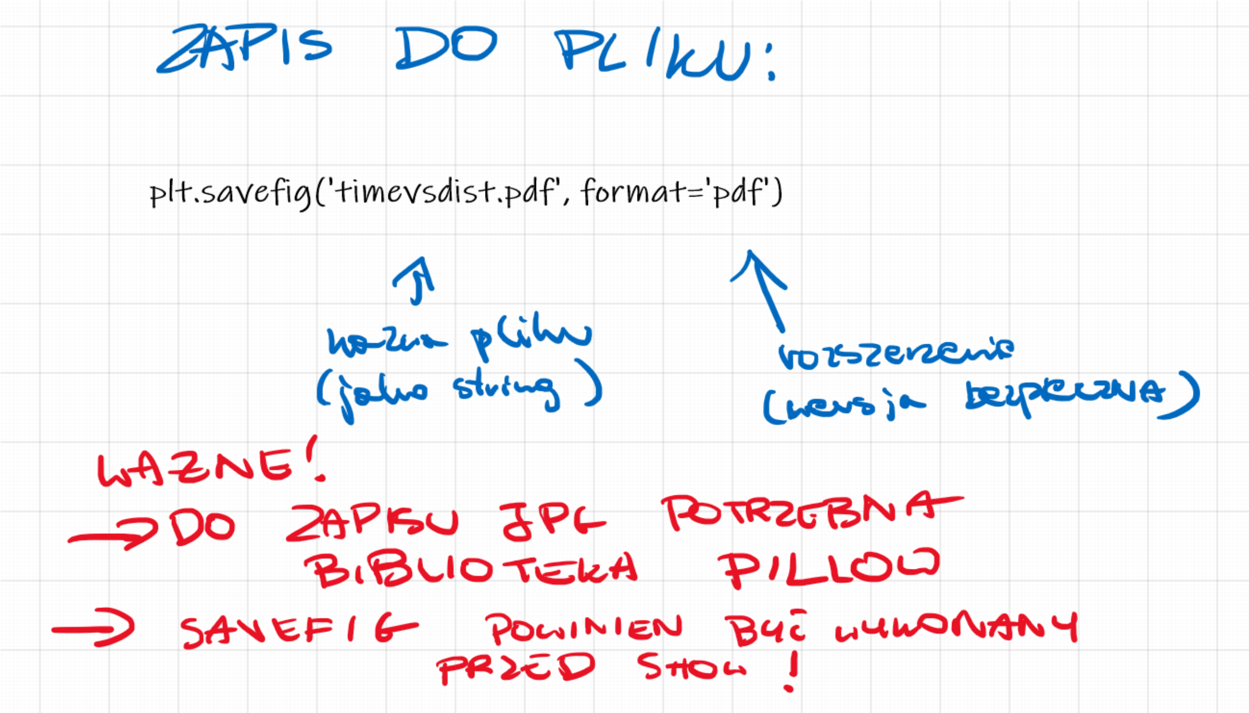
```
import matplotlib.pyplot as plt

plt.plot([1, 2, 3, 4], [10, 20, 25, 30], color='lightblue', linewidth=3)
plt.scatter([0.3, 3.8, 1.2, 2.5], [11, 25, 9, 26], color='darkgreen', marker='^')
plt.xlim(0.5, 4.5)
plt.show()
```

```
## (0.5, 4.5)
```



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

x = np.arange(0, 10)
y = x ^ 2
# Labeling the Axes and Title
plt.title("Graph Drawing")
plt.xlabel("Time")
plt.ylabel("Distance")

# Formatting the line colors
plt.plot(x, y, 'r')

# Formatting the line type
plt.plot(x, y, '>')

# save in pdf formats
plt.savefig('timevsdist.pdf', format='pdf')
plt.show()
```





ZAPIS DO PLIKU:

```
plt.savefig('timevsdist.pdf', format='pdf')
```

nazwa pliku (jako string)

rozszerzenie (wersja bezpieczna)

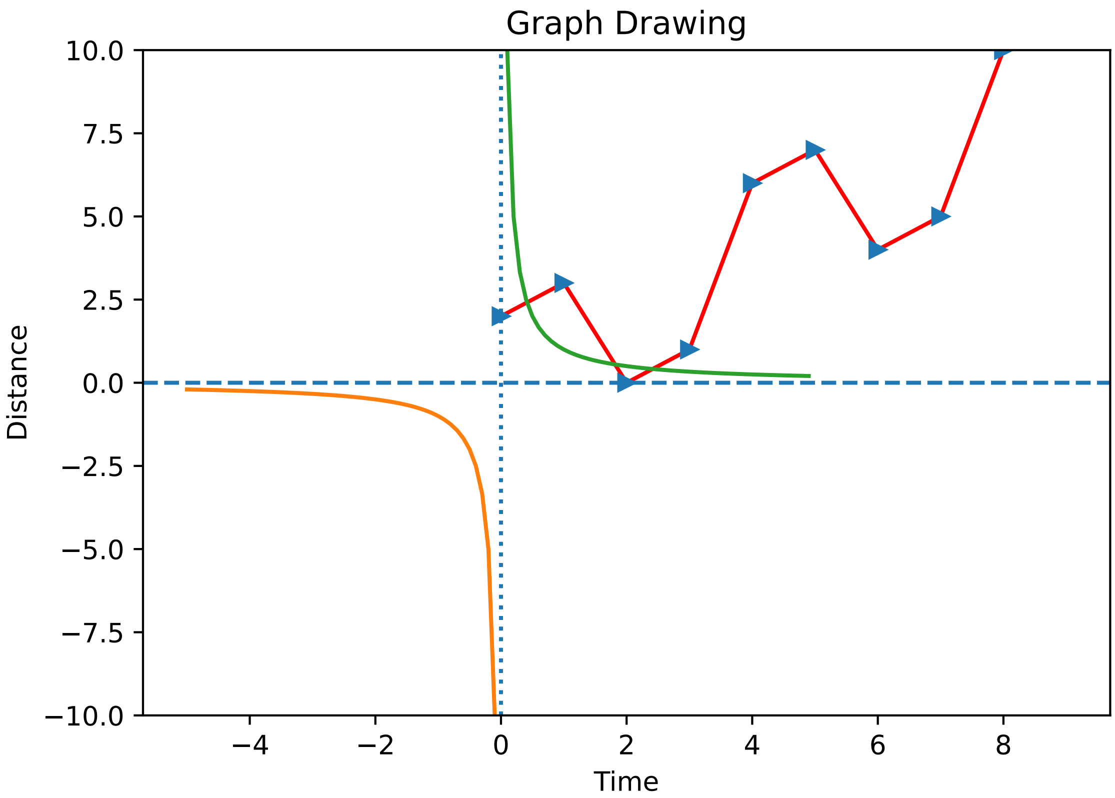
WAZNE!

→ DO ZAPISU JPG POTRZEBNA BIBLIOTEKA PILLOW

→ SAVEFIG POWINIEN BYĆ WYKONANY PRZED SHOW!

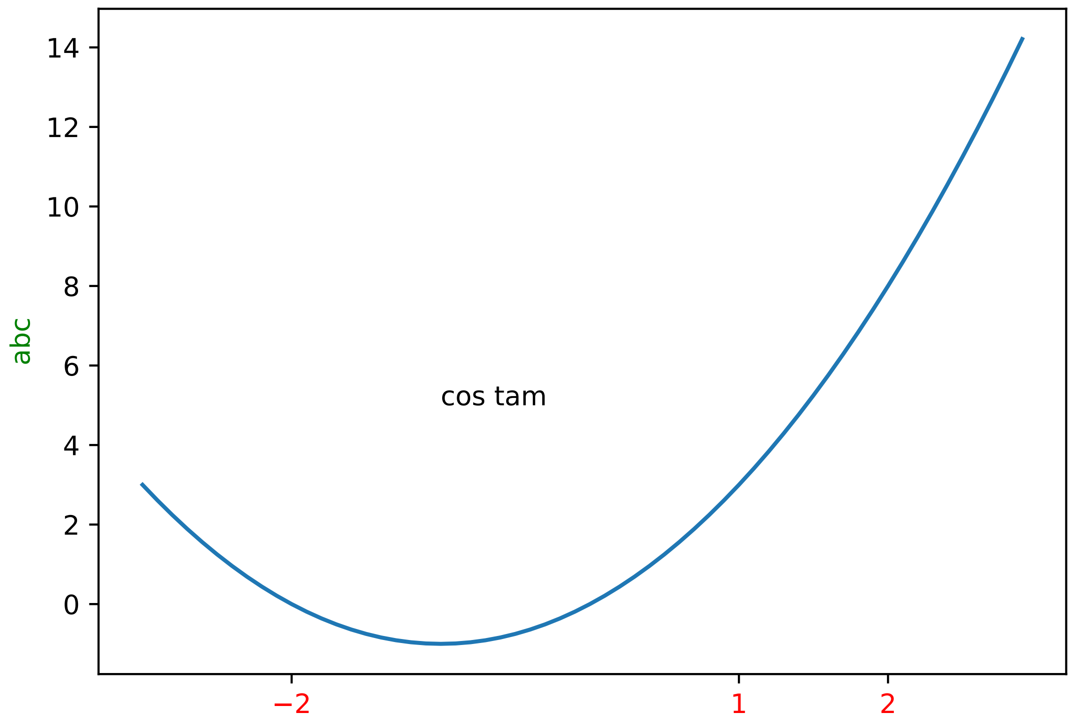
```
import numpy as np
import matplotlib.pyplot as plt

x = np.arange(-5, 5, 0.1)
x1 = x[x < 0]
y1 = 1 / x1
plt.plot(x1, y1)
x2 = x[x > 0]
y2 = 1 / x2
plt.plot(x2, y2)
plt.ylim(-10, 10)
plt.axhline(y=0, linestyle="--")
plt.axvline(x=0, linestyle=":")
plt.show()
```

```
## (-10.0, 10.0)
```

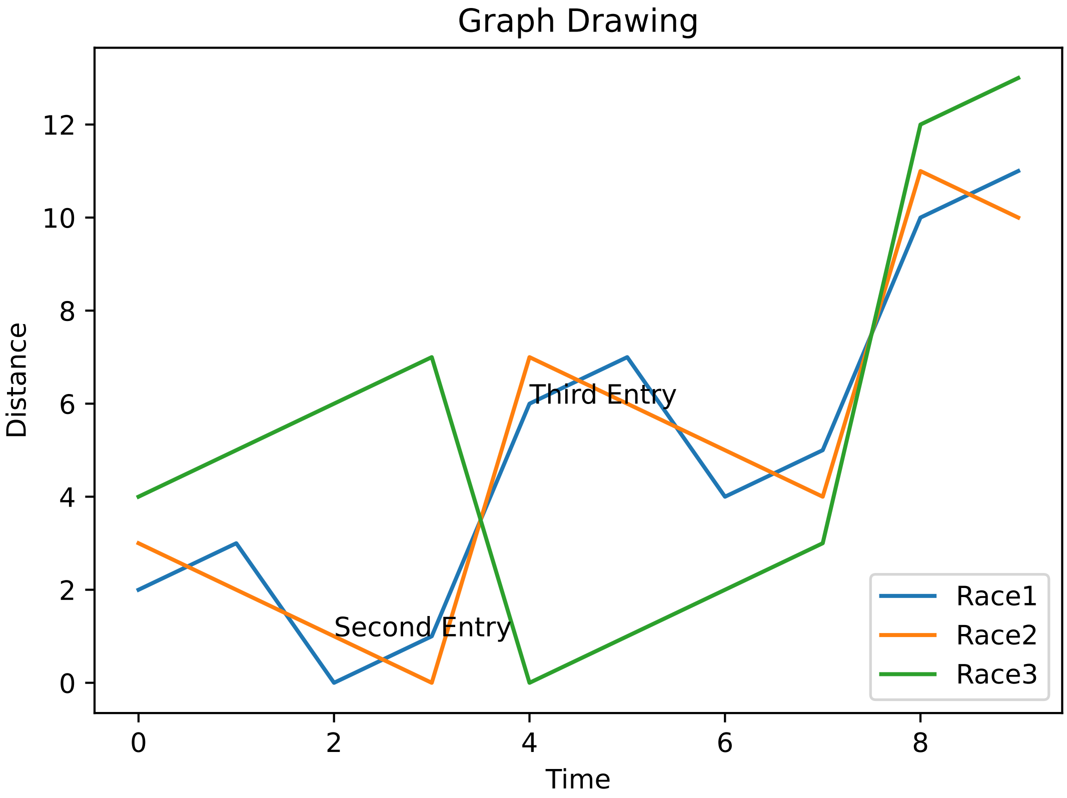


```
import numpy as np
import matplotlib.pyplot as plt

x = np.arange(-3, 3, 0.1)
y = x ** 2 + 2 * x
plt.plot(x, y)
plt.annotate(xy=[-1, 5], text="cos tam")
plt.xticks([-2, 1, 2], color="red")
plt.ylabel("abc", color="green")
plt.show()
```

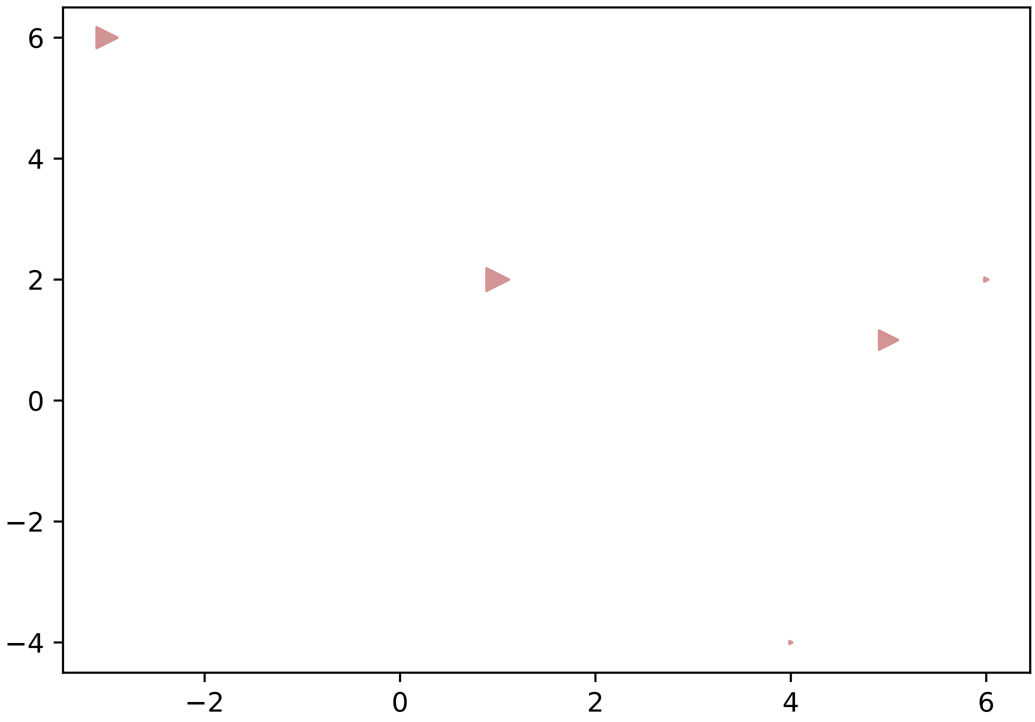
```
## ([<matplotlib.axis.XTick object at 0x0000000061243400>, <matplotlib.axis.XTick object at 0x000000006
```



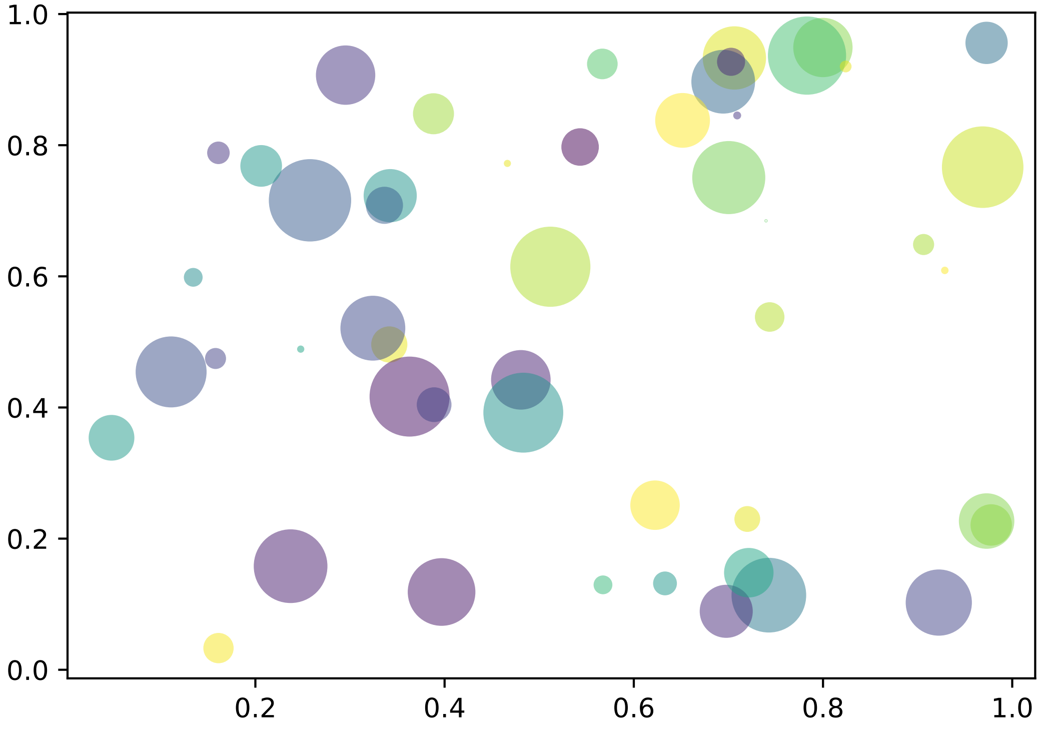
```
import numpy as np
from matplotlib import pyplot as plt

x = np.arange(0, 10)
y = x ^ 2
z = x ^ 3
t = x ^ 4
# Labeling the Axes and Title
plt.title("Graph Drawing")
plt.xlabel("Time")
plt.ylabel("Distance")
plt.plot(x, y)

# Annotate
plt.annotate(xy=[2, 1], text='Second Entry')
plt.annotate(xy=[4, 6], text='Third Entry')
# Adding Legends
plt.plot(x, z)
plt.plot(x, t)
plt.legend(['Race1', 'Race2', 'Race3'], loc=4)
plt.show()
```



```
from matplotlib import pyplot as plt

x = [1, -3, 4, 5, 6]
y = [2, 6, -4, 1, 2]
area = [70, 60, 1, 50, 2]
plt.scatter(x, y, marker=">", color="brown", alpha=0.5, s=area)
plt.show()
```

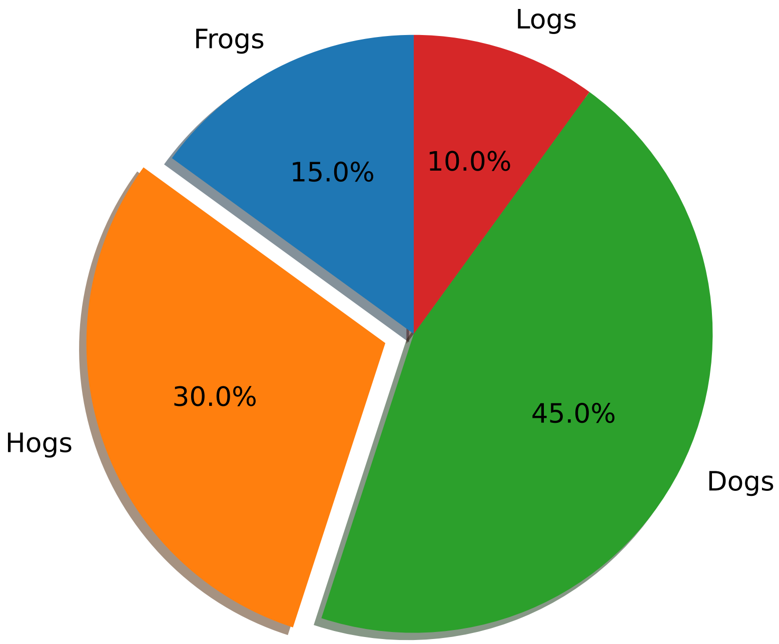


```
import numpy as np
import matplotlib.pyplot as plt

# Fixing random state for reproducibility
np.random.seed(19680801)

N = 50
x = np.random.rand(N)
y = np.random.rand(N)
colors = np.random.rand(N)
area = (30 * np.random.rand(N)) ** 2  # 0 to 15 point radii

plt.scatter(x, y, s=area, c=colors, alpha=0.5)
plt.show()
```



```python
import matplotlib.pyplot as plt

# Pie chart, where the slices will be ordered and plotted counter-clockwise:
labels = ['Frogs', 'Hogs', 'Dogs', 'Logs']
sizes = [15, 30, 45, 10]
explode = [0, 0.1, 0, 0]  # only "explode" the 2nd slice (i.e. 'Hogs')

fig1, ax1 = plt.subplots()
ax1.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax1.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

plt.show()
```

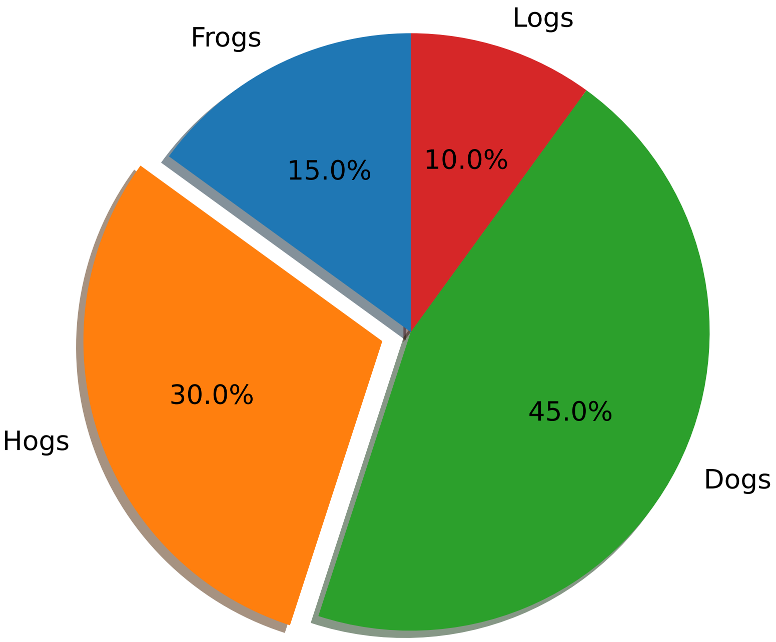
```
## ([<matplotlib.patches.Wedge object at 0x000000006184BCA0>, <matplotlib.patches.Wedge object at 0x0000
## (-1.2142507666786346, 1.134711077927507, -1.1258468317413044, 1.101230806657754)
```

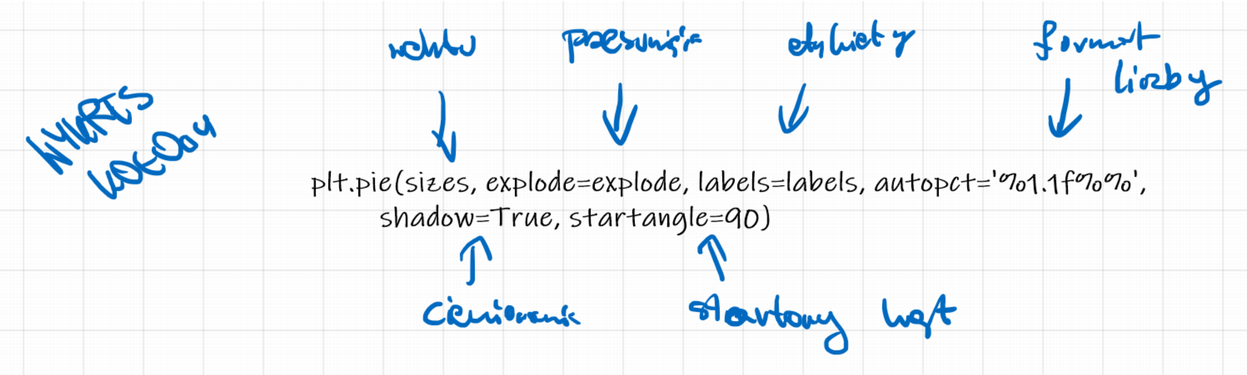


Wersja prostsza:

```python
import matplotlib.pyplot as plt

labels = ['Frogs', 'Hogs', 'Dogs', 'Logs']
sizes = [15, 30, 45, 10]
explode = [0, 0.1, 0, 0]  # only "explode" the 2nd slice (i.e. 'Hogs')
plt.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)
plt.axis('equal')
plt.show()
```

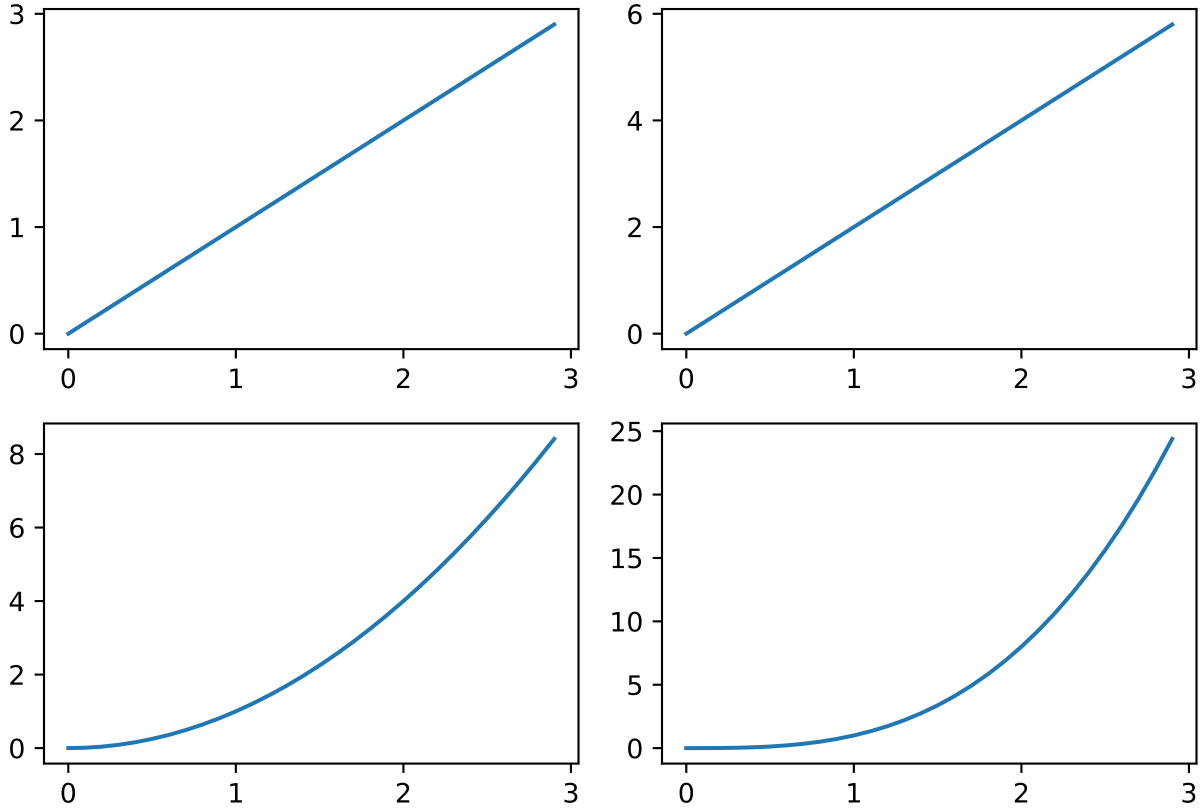
```
## ([<matplotlib.patches.Wedge object at 0x0000000060A7B100>, <matplotlib.patches.Wedge object at 0x0000
## (-1.2142507666786346, 1.134711077927507, -1.1258468317413044, 1.101230806657754)
```





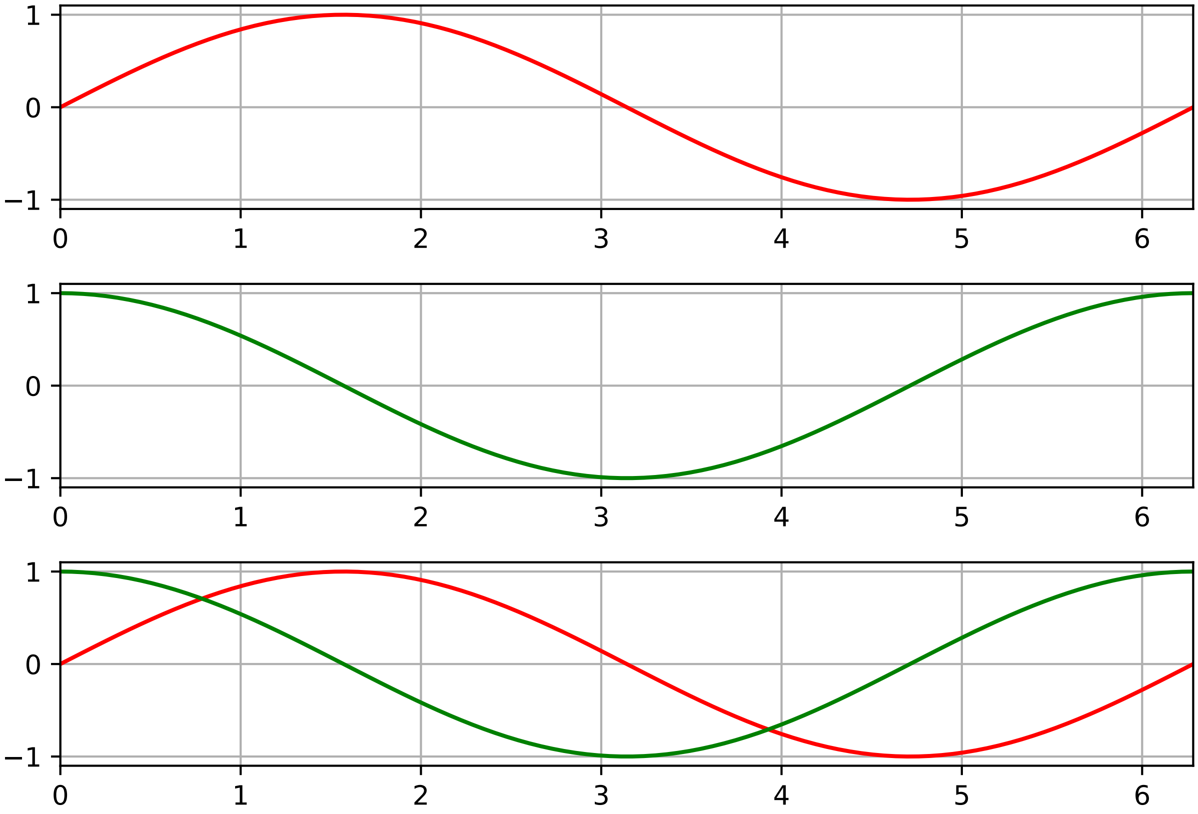
```
plt.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%',
    shadow=True, startangle=90)
```

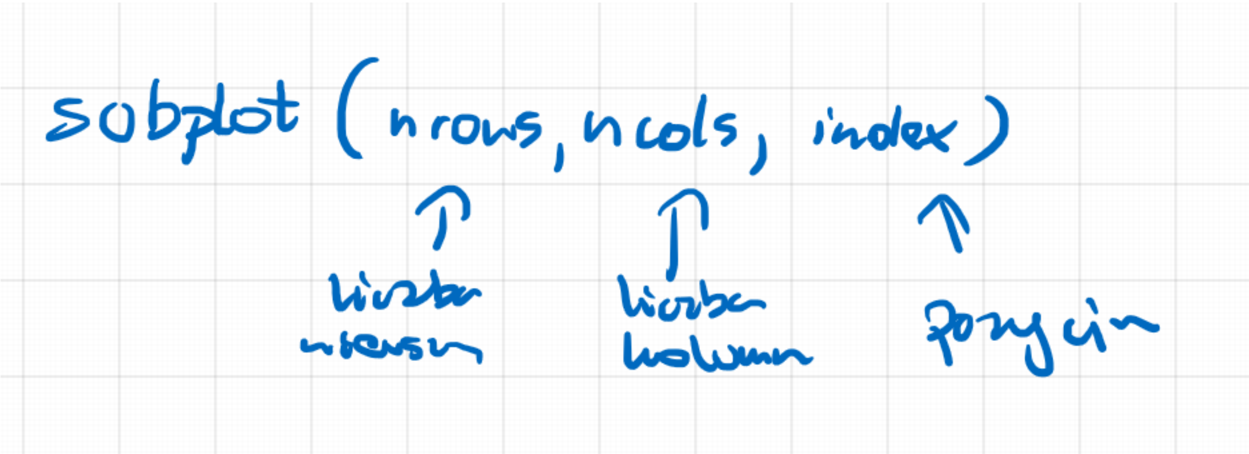
```
import numpy as np
import matplotlib.pyplot as plt

x = np.arange(0,3,0.1)
plt.subplot(2, 2, 1)
plt.plot(x, x)
plt.subplot(2, 2, 2)
plt.plot(x, x * 2)
plt.subplot(2, 2, 3)
plt.plot(x, x * x)
plt.subplot(2, 2, 4)
plt.plot(x, x ** 3)
plt.tight_layout()
plt.show()
```



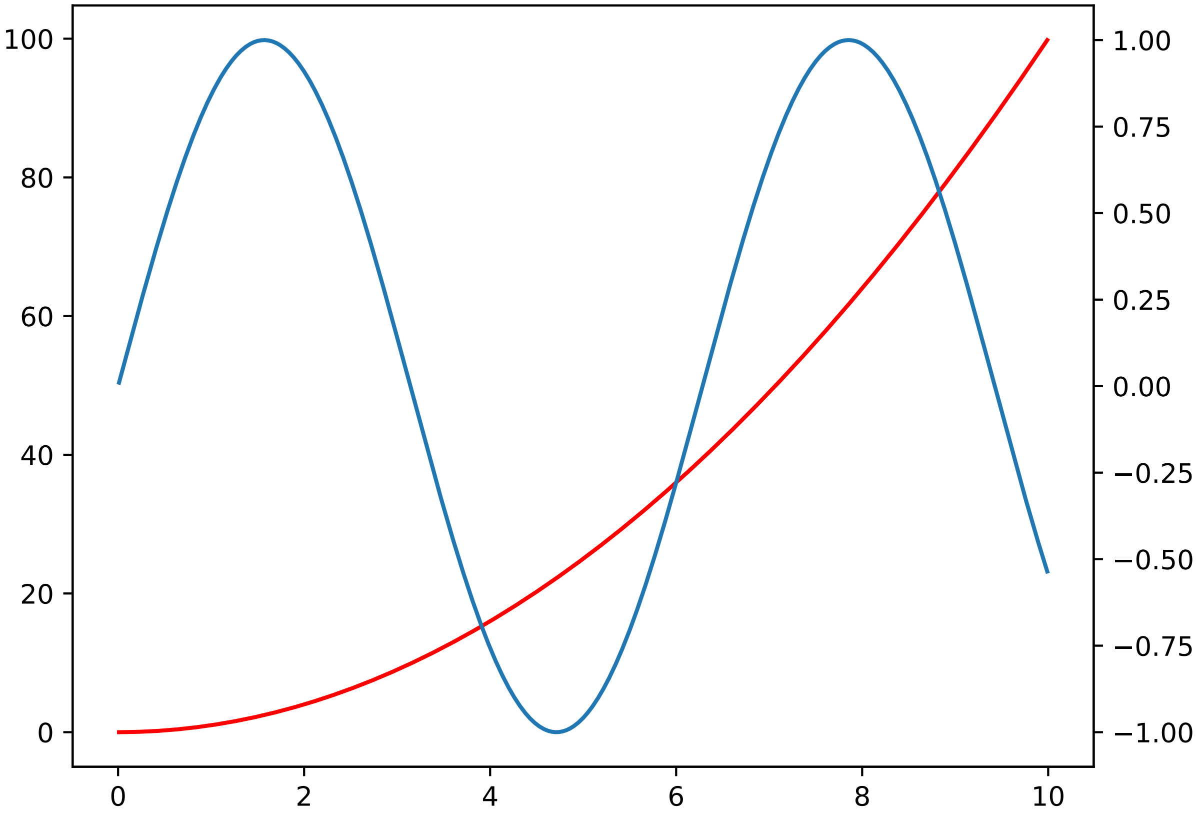
```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(0, np.pi * 2, 100)
plt.subplot(3, 1, 1)
plt.plot(x, np.sin(x), 'r')
plt.grid(True)
plt.xlim(0, np.pi * 2)
plt.subplot(3, 1, 2)
plt.plot(x, np.cos(x), 'g')
plt.grid(True)
plt.xlim(0, np.pi * 2)
plt.subplot(3, 1, 3)
plt.plot(x, np.sin(x), 'r', x, np.cos(x), 'g')
plt.grid(True)
plt.xlim(0, np.pi * 2)
plt.tight_layout()
plt.savefig("fig3.png", dpi=72)
plt.show()
```

```
## (0.0, 6.283185307179586)
## (0.0, 6.283185307179586)
## (0.0, 6.283185307179586)
```

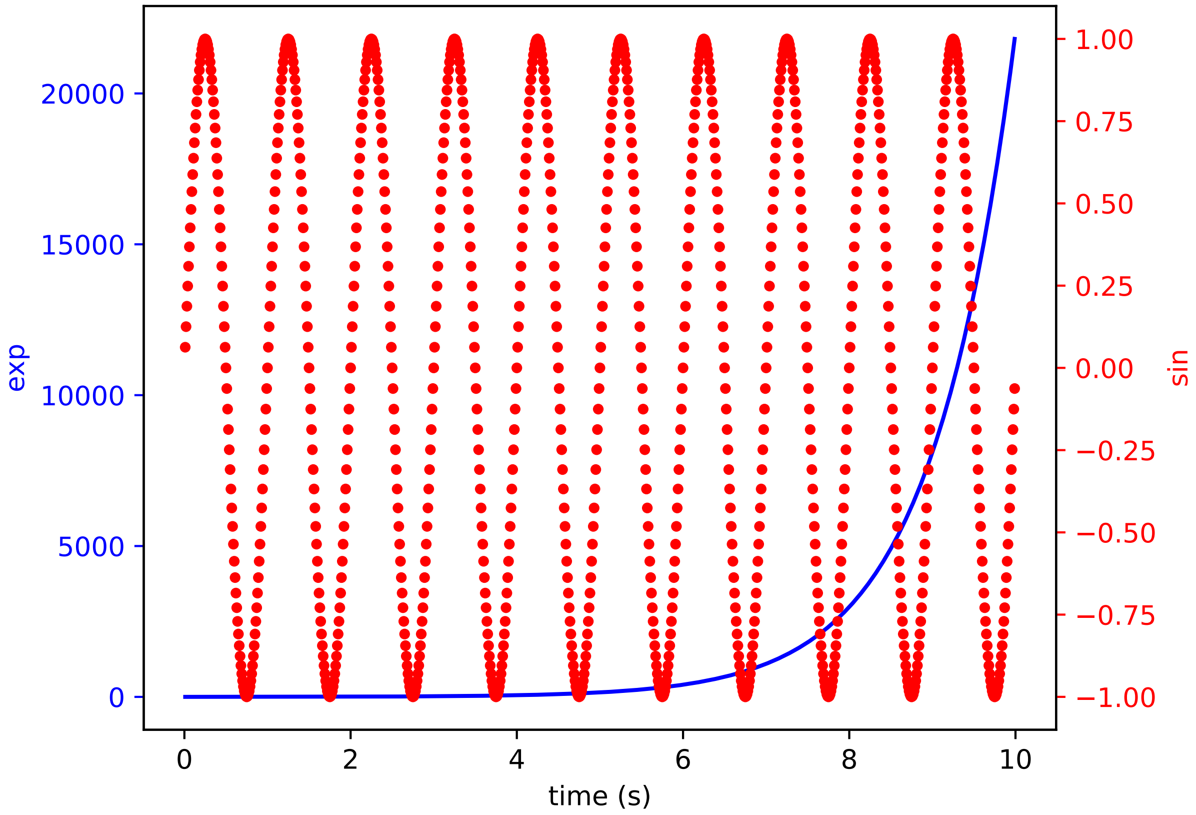



subplot (nrows, ncols, index)
liczba wierszy
liczba kolumn
pozycja

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

fig, ax1 = plt.subplots()
x = np.arange(0.01, 10.0, 0.01)
y = x ** 2
ax1.plot(x, y, 'r')
ax2 = ax1.twinx()
y2 = np.sin(x)
ax2.plot(x, y2)
fig.tight_layout()
plt.show()
```

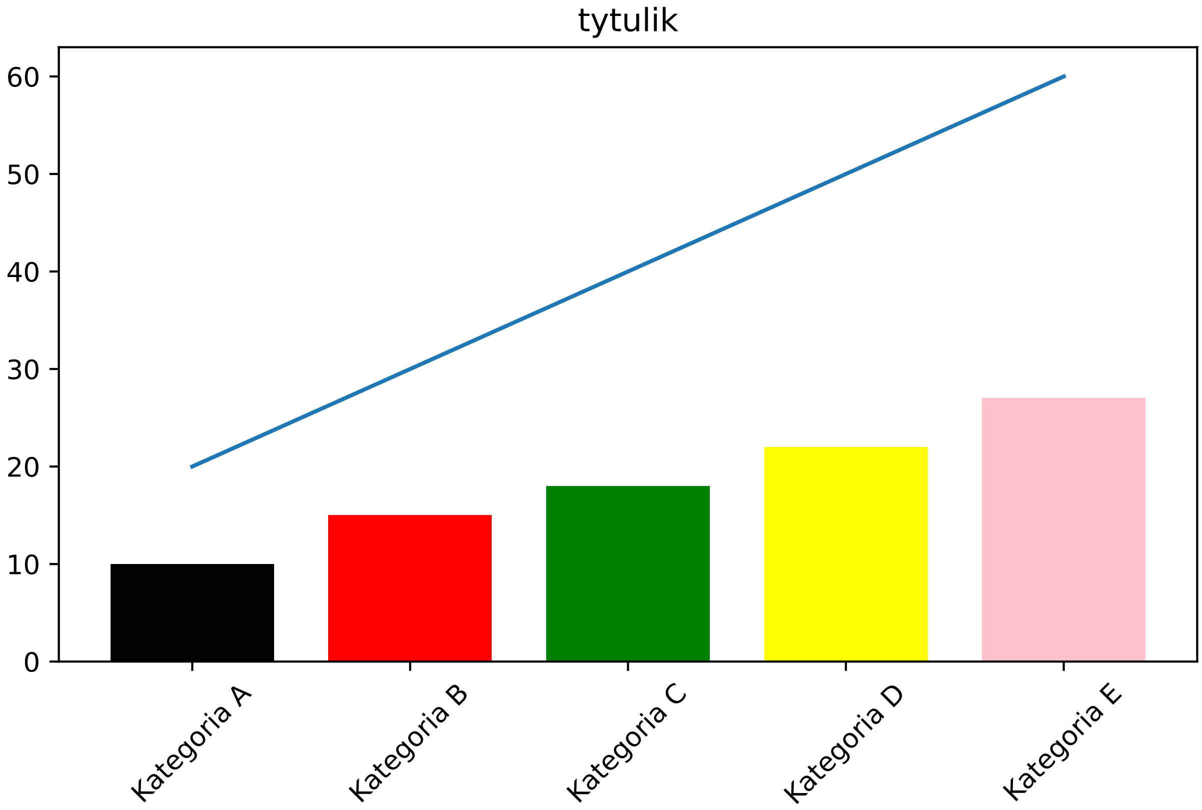


```python
import numpy as np
import matplotlib.pyplot as plt

fig, ax1 = plt.subplots()
t = np.arange(0.01, 10.0, 0.01)
s1 = np.exp(t)
ax1.plot(t, s1, 'b-')
ax1.set_xlabel('time (s)')

ax1.set_ylabel('exp', color='b')
ax1.tick_params('y', colors='b')

ax2 = ax1.twinx()
s2 = np.sin(2 * np.pi * t)
ax2.plot(t, s2, 'r.')
ax2.set_ylabel('sin', color='r')
ax2.tick_params('y', colors='r')

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



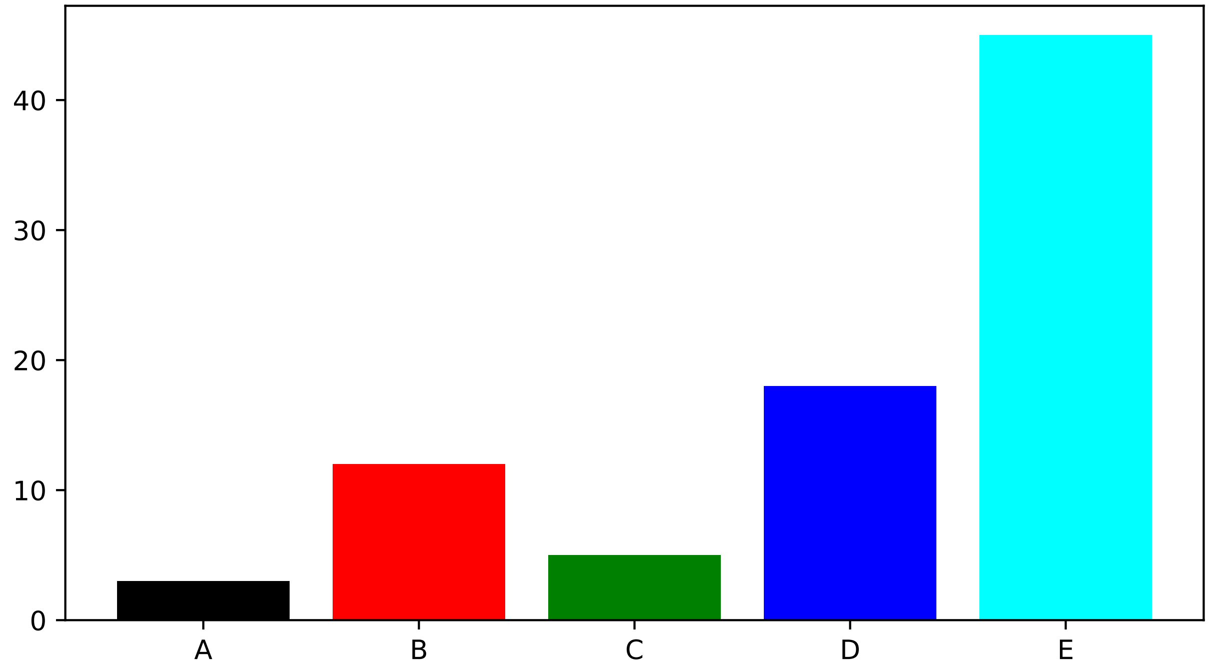
```
import numpy as np
import matplotlib.pyplot as plt

wys = [10, 15, 18, 22, 27]
x = np.arange(0, len(wys))
k = ["black", "red", "green", "yellow", "pink"]
plt.bar(x, wys, color=k, width=0.75)
etyk = ["Kategoria A", "Kategoria B", "Kategoria C", "Kategoria D", "Kategoria E"]
plt.xticks(x, etyk, rotation=45)
y2 = [20, 30, 40, 50, 60]
plt.plot(x, y2)
plt.title("tytulik")
plt.tight_layout()
plt.show()
```

```
## <BarContainer object of 5 artists>
## ([<matplotlib.axis.XTick object at 0x0000000062BA2DF0>, <matplotlib.axis.XTick object at 0x00000000062
```



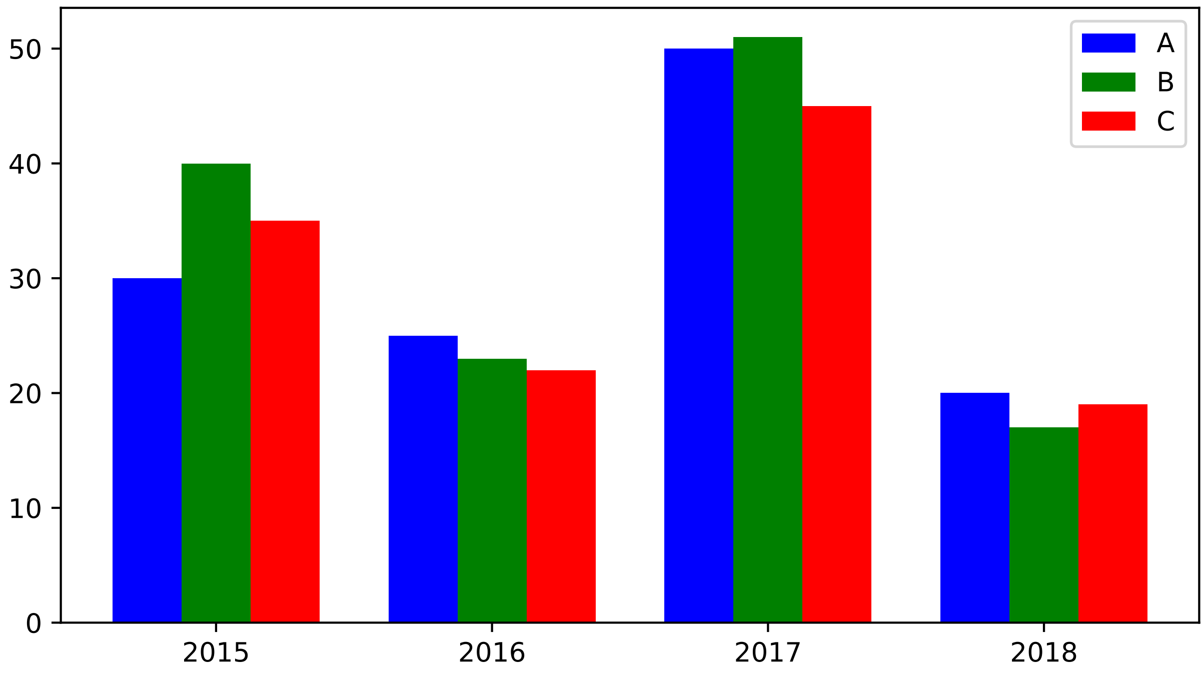
```
import numpy as np
import matplotlib.pyplot as plt

height = [3, 12, 5, 18, 45]
bars = ('A', 'B', 'C', 'D', 'E')
y_pos = np.arange(len(bars))
plt.bar(y_pos, height, color=['black', 'red', 'green', 'blue', 'cyan'])
plt.xticks(y_pos, bars)
plt.show()
```

```
## <BarContainer object of 5 artists>
## ([<matplotlib.axis.XTick object at 0x0000000062B44EE0>, <matplotlib.axis.XTick object at 0x000000006
```



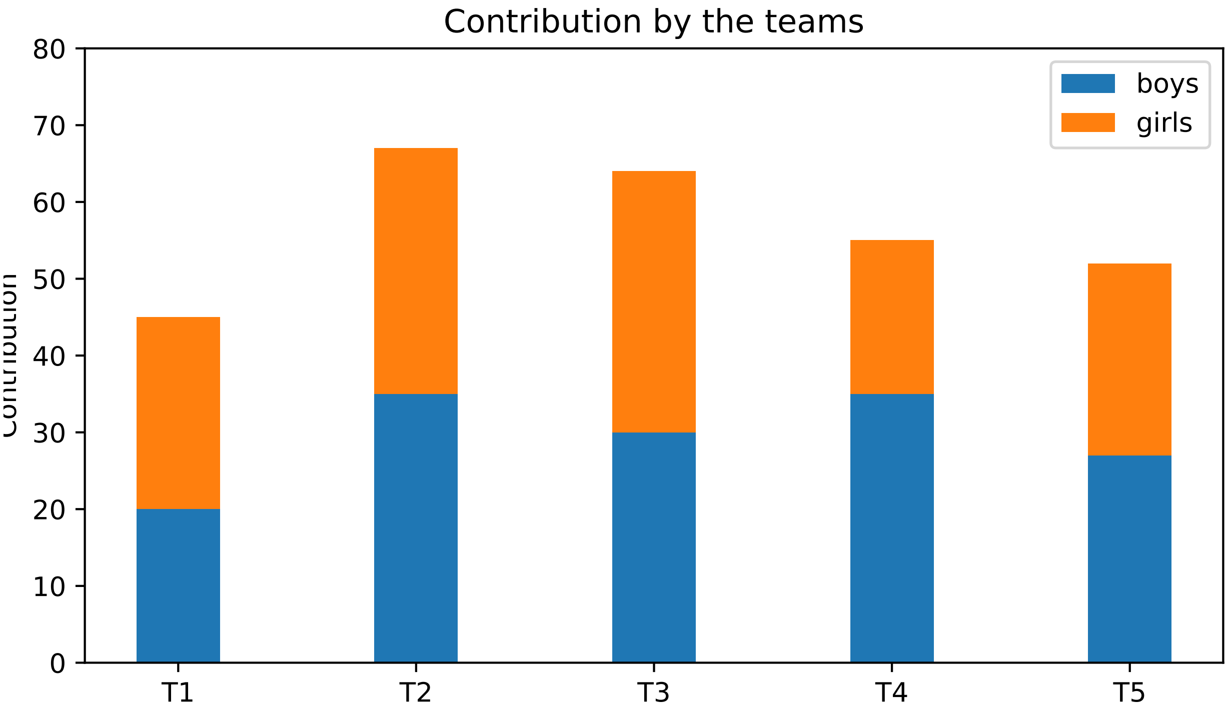
```
import numpy as np
import matplotlib.pyplot as plt

data = [[30, 25, 50, 20],
        [40, 23, 51, 17],
        [35, 22, 45, 19]]
X = np.arange(4)

plt.bar(X + 0.00, data[0], color='b', width=0.25, label="A")
plt.bar(X + 0.25, data[1], color='g', width=0.25, label="B")
plt.bar(X + 0.50, data[2], color='r', width=0.25, label="C")
labelsbar = np.arange(2015,2019)
plt.xticks(X+0.25,labelsbar)
plt.legend()
plt.show()
```

```
## <BarContainer object of 4 artists>
## <BarContainer object of 4 artists>
## <BarContainer object of 4 artists>
## ([<matplotlib.axis.XTick object at 0x0000000062CF96D0>, <matplotlib.axis.XTick object at 0x0000000062
```



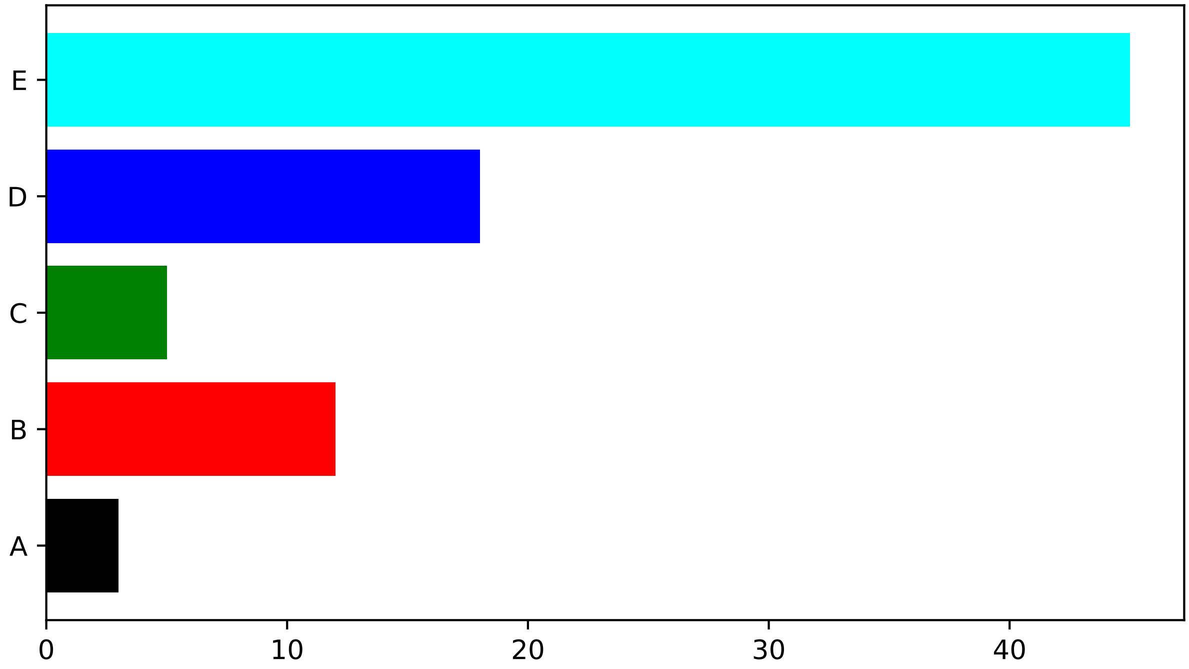
```
import numpy as np
import matplotlib.pyplot as plt

N = 5

boys = (20, 35, 30, 35, 27)
girls = (25, 32, 34, 20, 25)
ind = np.arange(N)
width = 0.35

plt.bar(ind, boys, width, label="boys")
plt.bar(ind, girls, width,bottom=boys, label="girls")

plt.ylabel('Contribution')
plt.title('Contribution by the teams')
plt.xticks(ind, ('T1', 'T2', 'T3', 'T4', 'T5'))
plt.yticks(np.arange(0, 81, 10))
plt.legend()
plt.show()
```

```
## <BarContainer object of 5 artists>
## <BarContainer object of 5 artists>
## ([<matplotlib.axis.XTick object at 0x0000000062CBF5B0>, <matplotlib.axis.XTick object at 0x000000006
## ([<matplotlib.axis.YTick object at 0x0000000062A946A0>, <matplotlib.axis.YTick object at 0x000000006
```



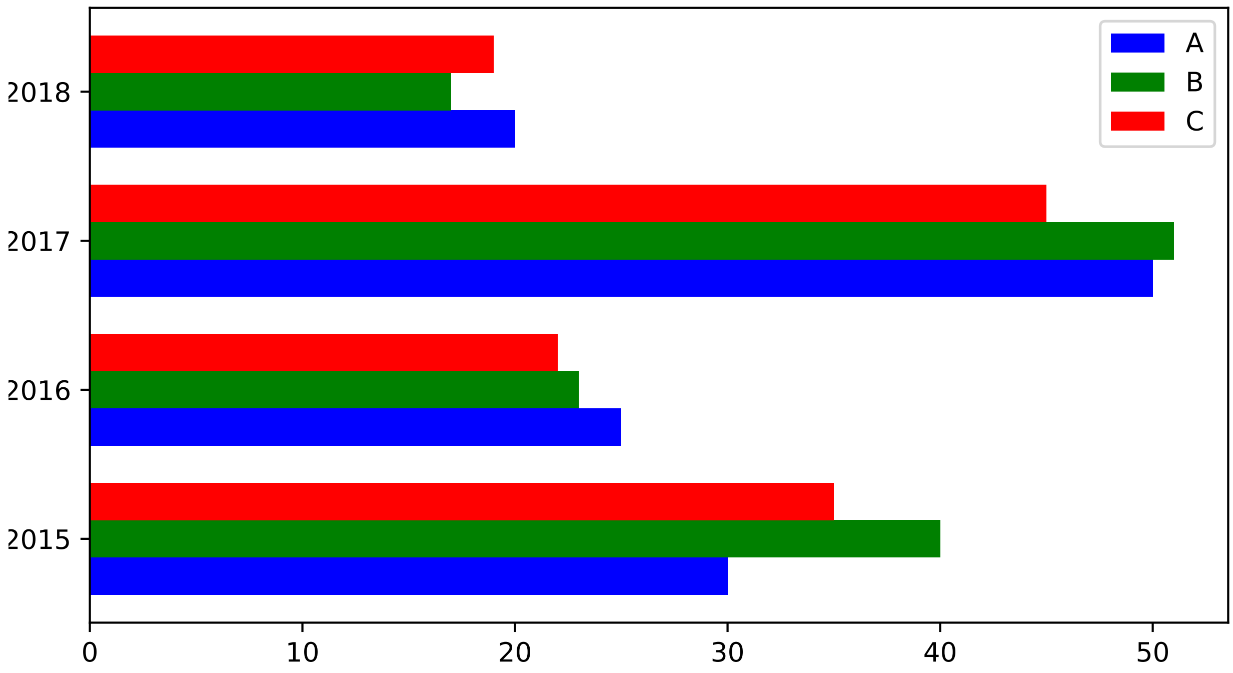
```
import numpy as np
import matplotlib.pyplot as plt

width = [3, 12, 5, 18, 45]
bars = ('A', 'B', 'C', 'D', 'E')
x_pos = np.arange(len(bars))
plt.barh(x_pos, width, color=['black', 'red', 'green', 'blue', 'cyan'])
plt.yticks(x_pos, bars)
plt.show()
```

```
## <BarContainer object of 5 artists>
## ([<matplotlib.axis.YTick object at 0x0000000062C23E50>, <matplotlib.axis.YTick object at 0x000000006
```



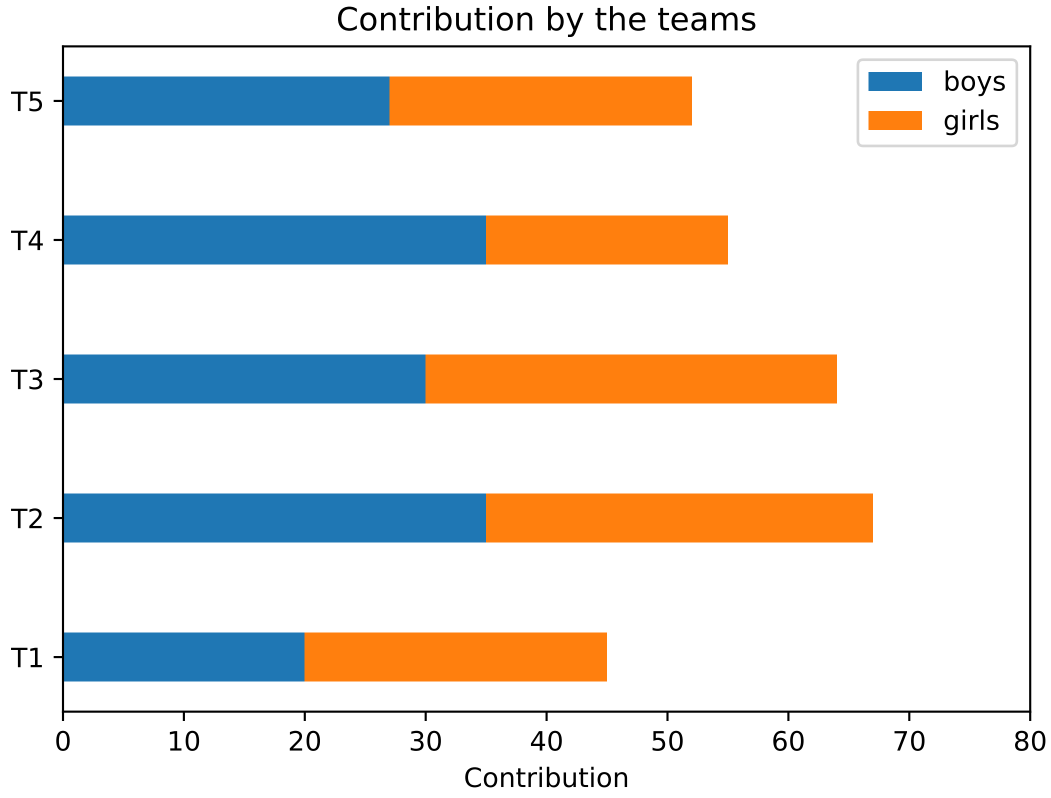
```
import numpy as np
import matplotlib.pyplot as plt

data = [[30, 25, 50, 20],
        [40, 23, 51, 17],
        [35, 22, 45, 19]]
Y = np.arange(4)

plt.barh(Y + 0.00, data[0], color='b', height=0.25, label="A")
plt.barh(Y + 0.25, data[1], color='g', height=0.25, label="B")
plt.barh(Y + 0.50, data[2], color='r', height=0.25, label="C")
labelsbar = np.arange(2015,2019)
plt.yticks(Y + 0.25, labelsbar)
plt.legend()
plt.show()
```

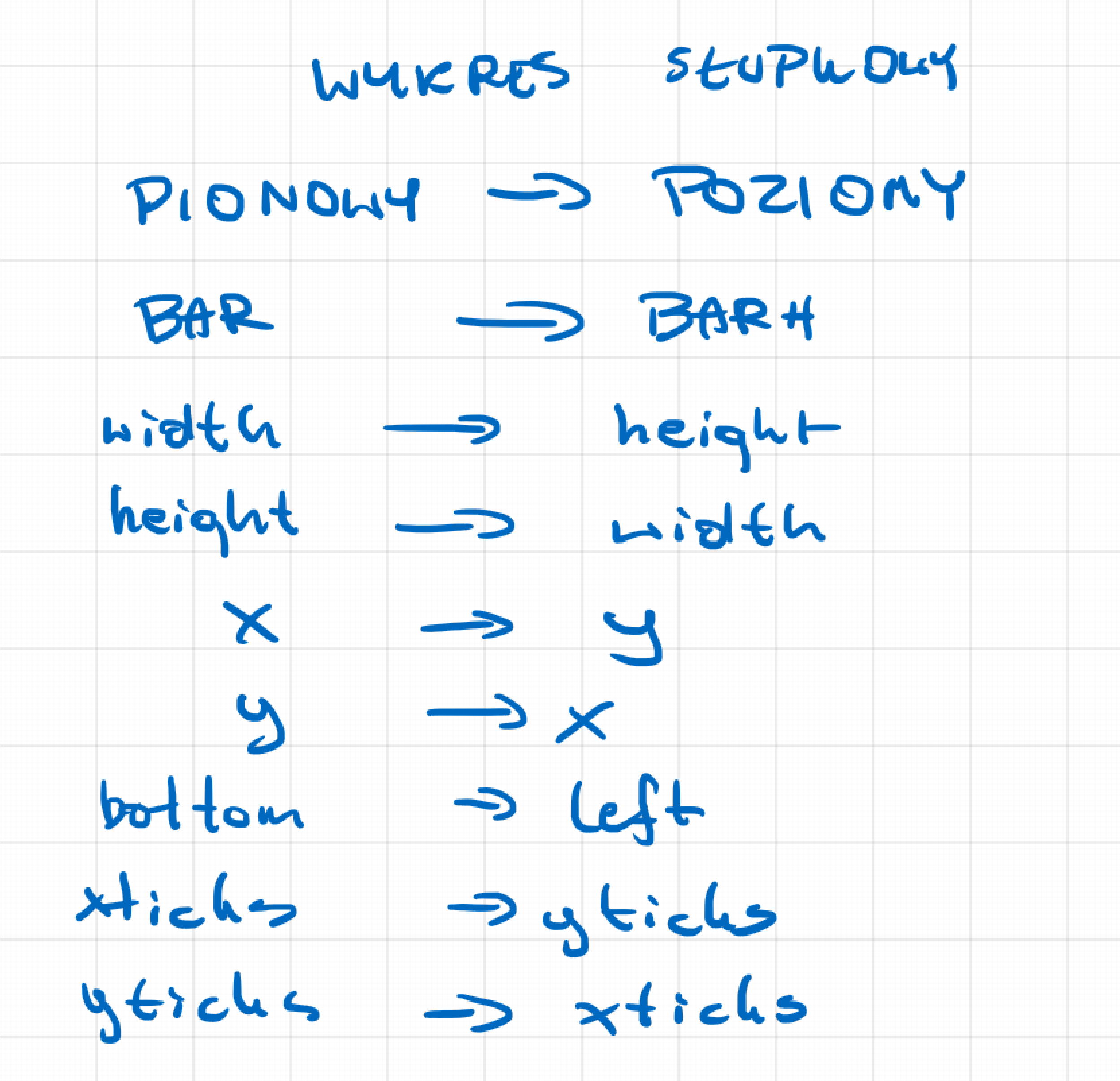
```
## <BarContainer object of 4 artists>
## <BarContainer object of 4 artists>
## <BarContainer object of 4 artists>
## ([<matplotlib.axis.YTick object at 0x0000000062A18490>, <matplotlib.axis.YTick object at 0x000000006
```



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

N = 5

boys = (20, 35, 30, 35, 27)
girls = (25, 32, 34, 20, 25)
ind = np.arange(N)
height = 0.35

plt.barh(ind, boys, height, label="boys")
plt.barh(ind, girls, height, left=boys, label="girls")

plt.xlabel('Contribution')
plt.title('Contribution by the teams')
plt.yticks(ind, ('T1', 'T2', 'T3', 'T4', 'T5'))
plt.xticks(np.arange(0, 81, 10))
plt.legend()
plt.show()
```

```
## <BarContainer object of 5 artists>
## <BarContainer object of 5 artists>
## ([<matplotlib.axis.YTick object at 0x0000000060A29790>, <matplotlib.axis.YTick object at 0x000000006
## ([<matplotlib.axis.XTick object at 0x0000000060A14970>, <matplotlib.axis.XTick object at 0x000000006
```





WYKRES SŁUPKOWY

PIONOWY → POZIOMY

BAR → BARH

width → height

height → width

x → y

y → x

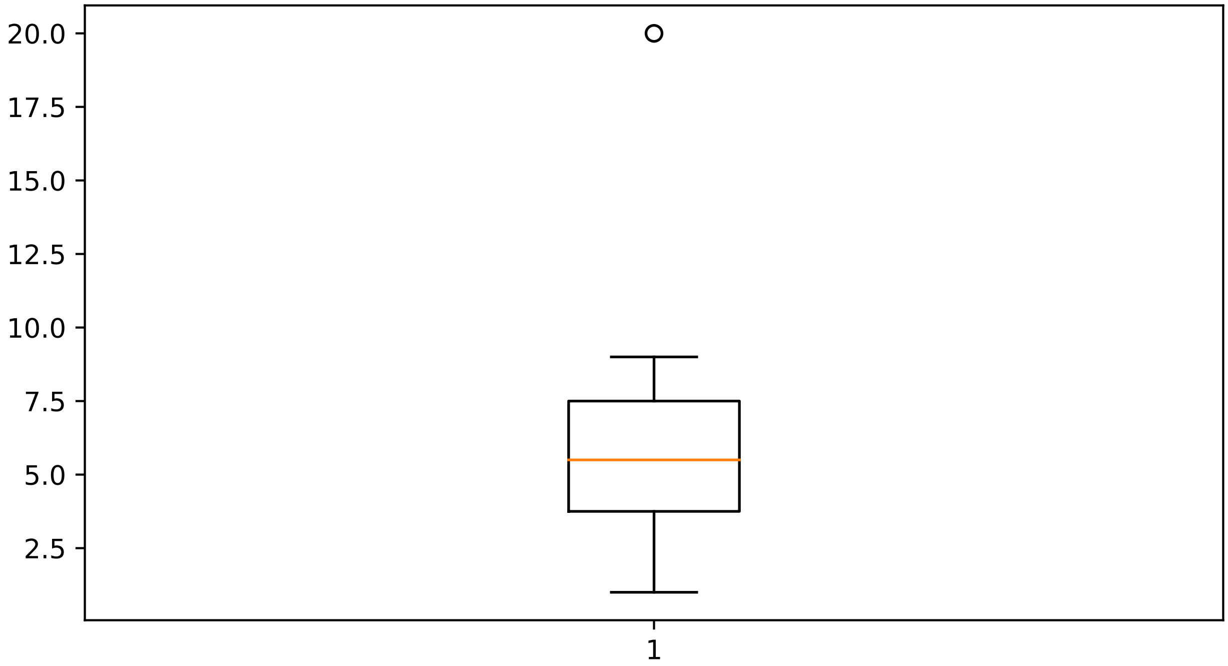
bottom → left

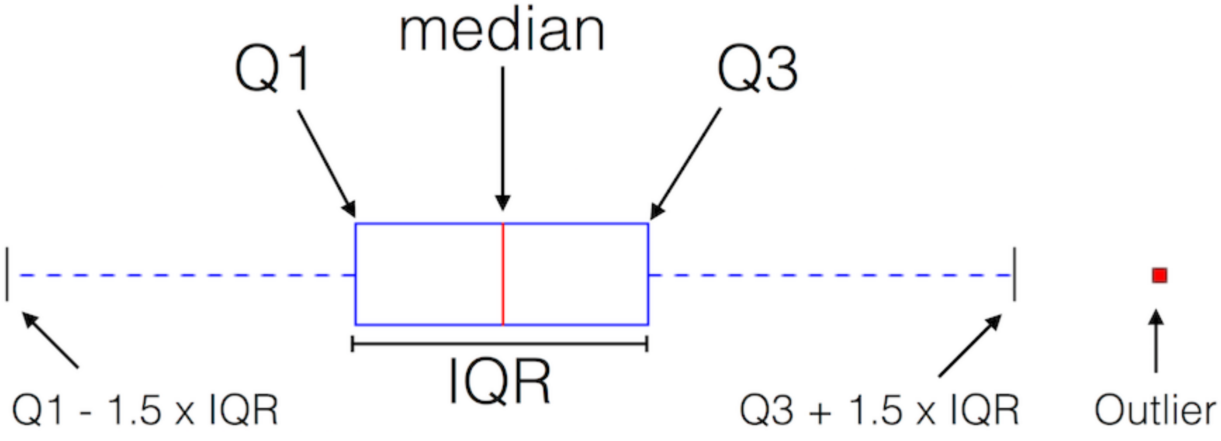
xticks → yticks

yticks → xticks

```
import matplotlib.pyplot as plt

dane = [1, 4, 5, 6, 3, 9, 7, 20]
plt.boxplot(dane)
plt.show()
```

```
## {'whiskers': [<matplotlib.lines.Line2D object at 0x0000000062A2BC10>, <matplotlib.lines.Line2D objec
```





Q1: *Quartile 1*, or median of the *left* data subset
after dividing the original data set into 2 subsets via the median
(25% of the data points fall below this threshold)

Q3: *Quartile 3*, median of the *right* data subset
(75% of the data points fall below this threshold)

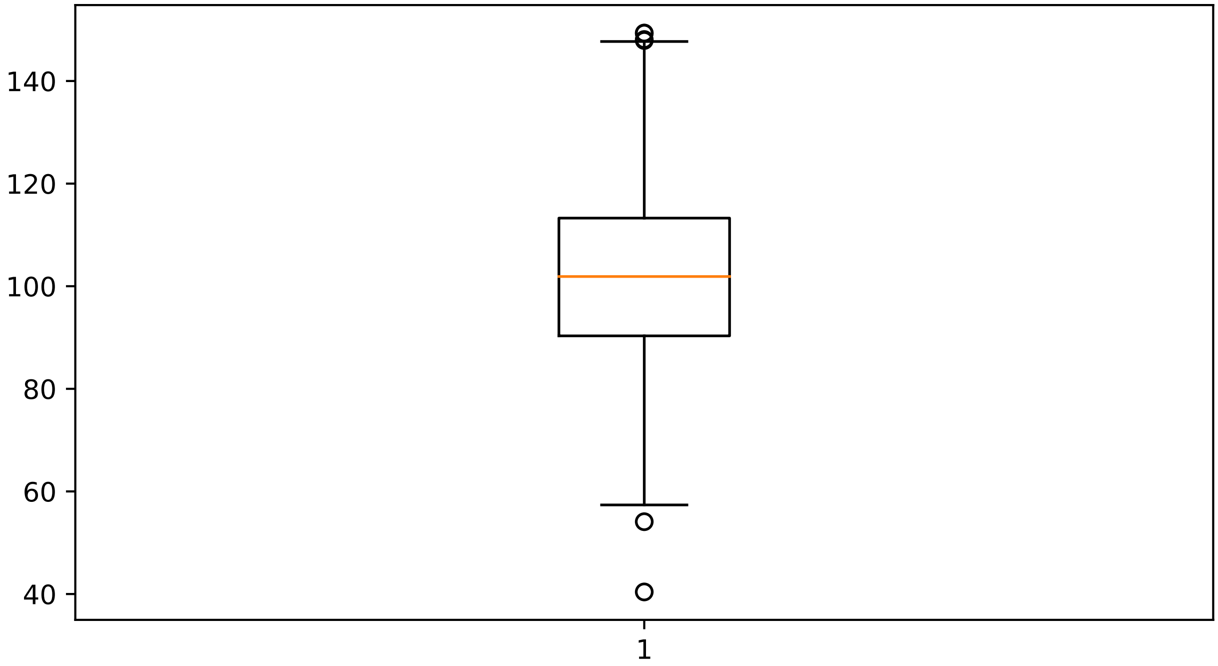
IQR: *Interquartile-range*, Q3 - Q1

Outliers: Data points are considered to be outliers if
value < *Q1 - 1.5 x IQR* or
value > Q3 + 1.5 x IQR

Sebastian Raschka, 2016

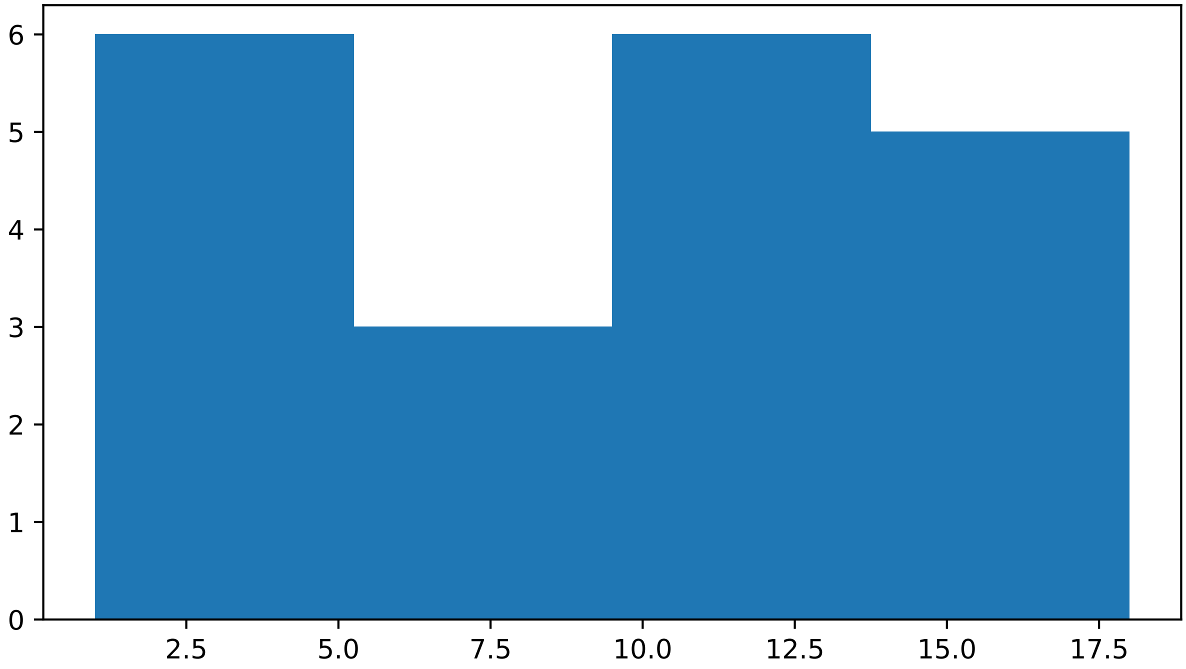
```
import matplotlib.pyplot as plt
import numpy as np

# Creating dataset
np.random.seed(10)
data = np.random.normal(100, 20, 200)

# Creating plot
plt.boxplot(data)

# show plot
plt.show()
```

```
## {'whiskers': [<matplotlib.lines.Line2D object at 0x0000000062A9C550>, <matplotlib.lines.Line2D objec
```



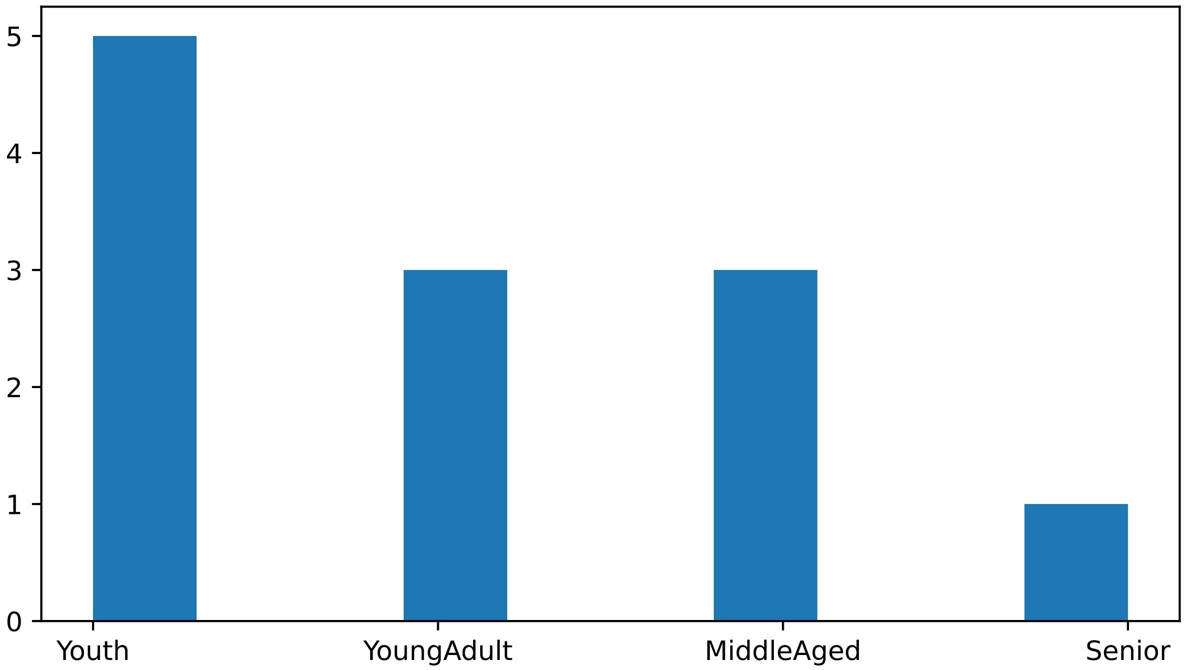
```
import matplotlib.pyplot as plt

x = [1, 1, 2, 3, 3, 5, 7, 8, 9, 10,
     10, 11, 11, 13, 13, 15, 16, 17, 18, 18]

plt.hist(x, bins=4)
plt.show()
```

```
## (array([6., 3., 6., 5.]), array([ 1.  ,  5.25,  9.5 , 13.75, 18.  ]), <BarContainer object of 4 arti
```



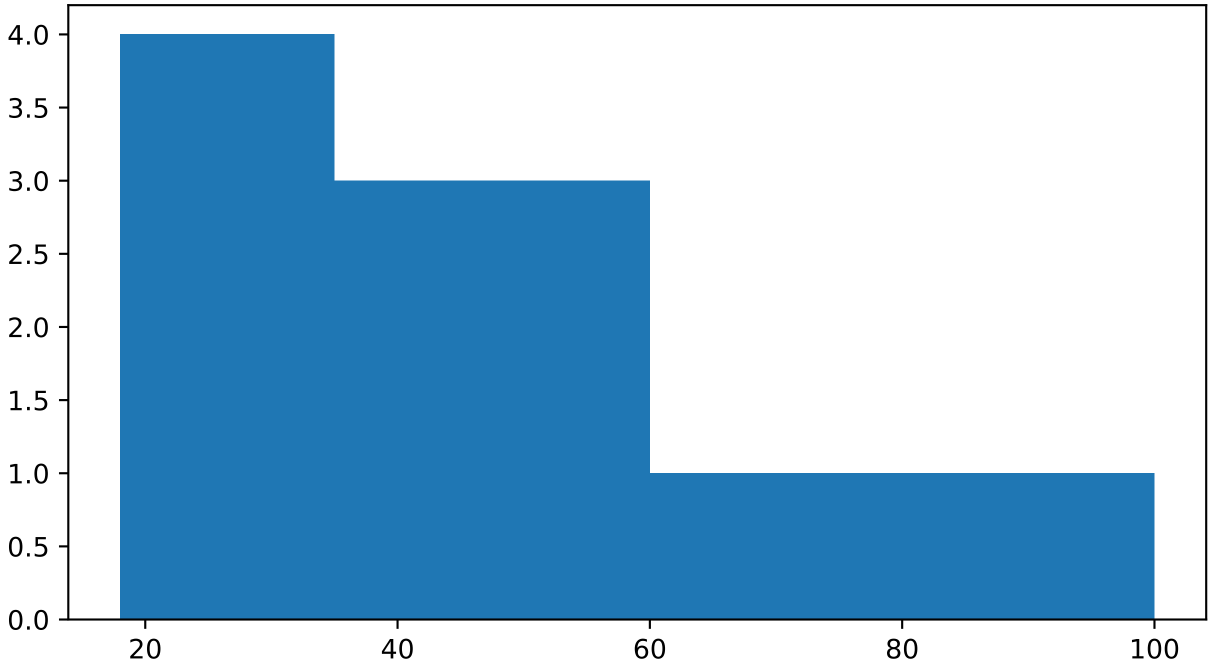
```
import pandas as pd
import matplotlib.pyplot as plt

ages = [20, 22, 25, 27, 21, 23, 37, 31, 61, 45, 41, 32]
bins = [18, 25, 35, 60, 100]
cats2 = pd.cut(ages, [18, 26, 36, 61, 100], right=False)
print(cats2)
group_names = ['Youth', 'YoungAdult',
               'MiddleAged', 'Senior']
data=pd.cut(ages, bins, labels=group_names)
plt.hist(data)
plt.show()
```

```
## [[18, 26), [18, 26), [18, 26), [26, 36), [18, 26), ..., [26, 36), [61, 100), [36, 61), [36, 61), [26
## Length: 12
## Categories (4, interval[int64, left]): [[18, 26) < [26, 36) < [36, 61) < [61, 100)]
## (array([5., 0., 0., 3., 0., 0., 3., 0., 0., 1.]), array([0.  , 0.3, 0.6, 0.9, 1.2, 1.5, 1.8, 2.1, 2.4
```



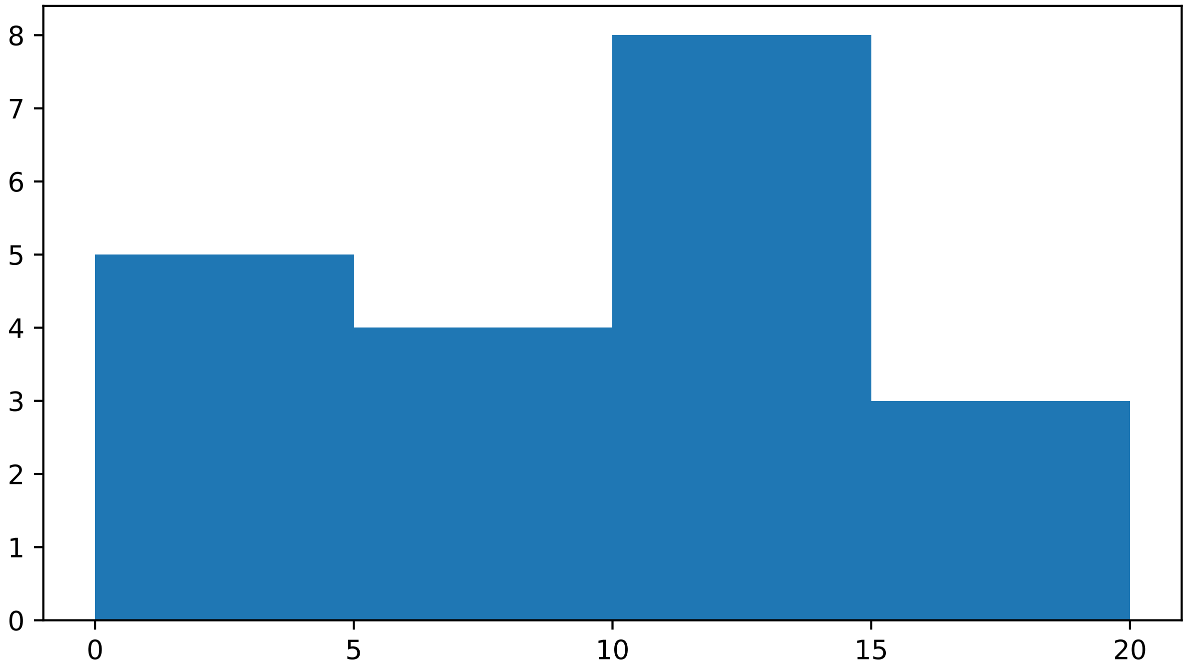
```
import pandas as pd
import matplotlib.pyplot as plt

ages = [20, 22, 25, 27, 21, 23, 37, 31, 61, 45, 41, 32]
bins = [18, 25, 35, 60, 100]
plt.hist(ages, bins=bins)
plt.show()
```

```
## (array([4., 4., 3., 1.]), array([ 18,  25,  35,  60, 100]), <BarContainer object of 4 artists>)
```



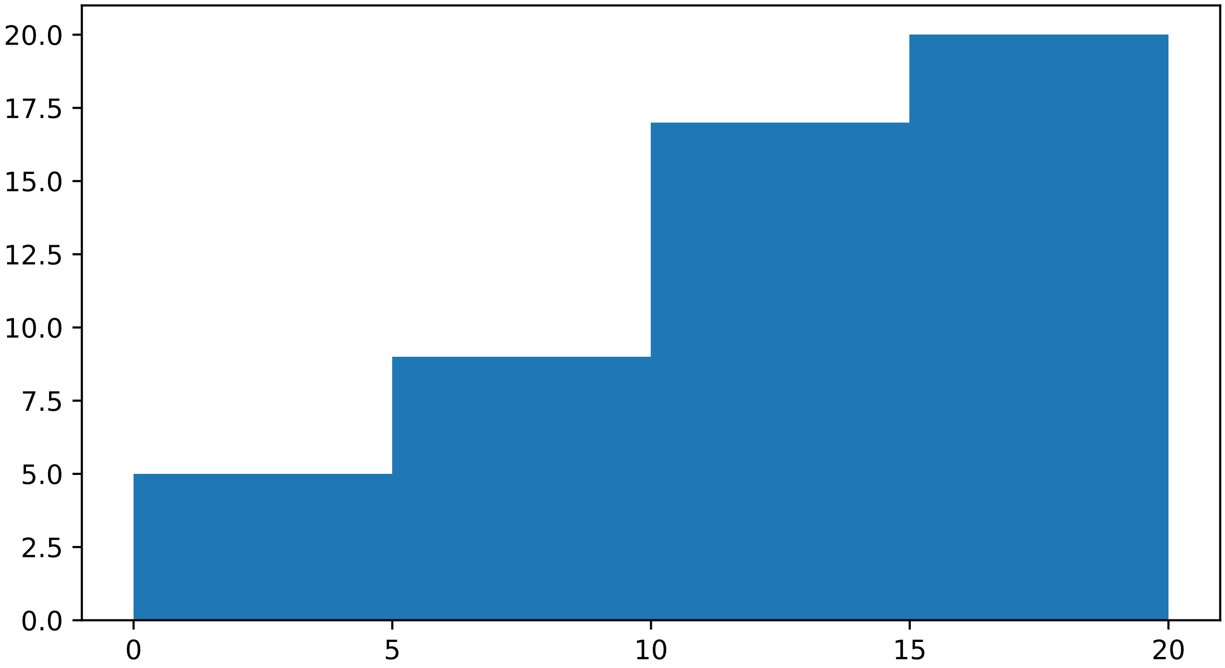
```
import matplotlib.pyplot as plt

x = [1, 1, 2, 3, 3, 5, 7, 8, 9, 10,
     10, 11, 11, 13, 13, 15, 14, 12, 18, 18]

plt.hist(x, bins=[0,5,10,15,20])
plt.xticks([0,5,10,15,20])
plt.show()
```

```
## (array([5., 4., 8., 3.]), array([ 0,  5, 10, 15, 20]), <BarContainer object of 4 artists>)
## ([<matplotlib.axis.XTick object at 0x000000006189BB20>, <matplotlib.axis.XTick object at 0x00000000006
```



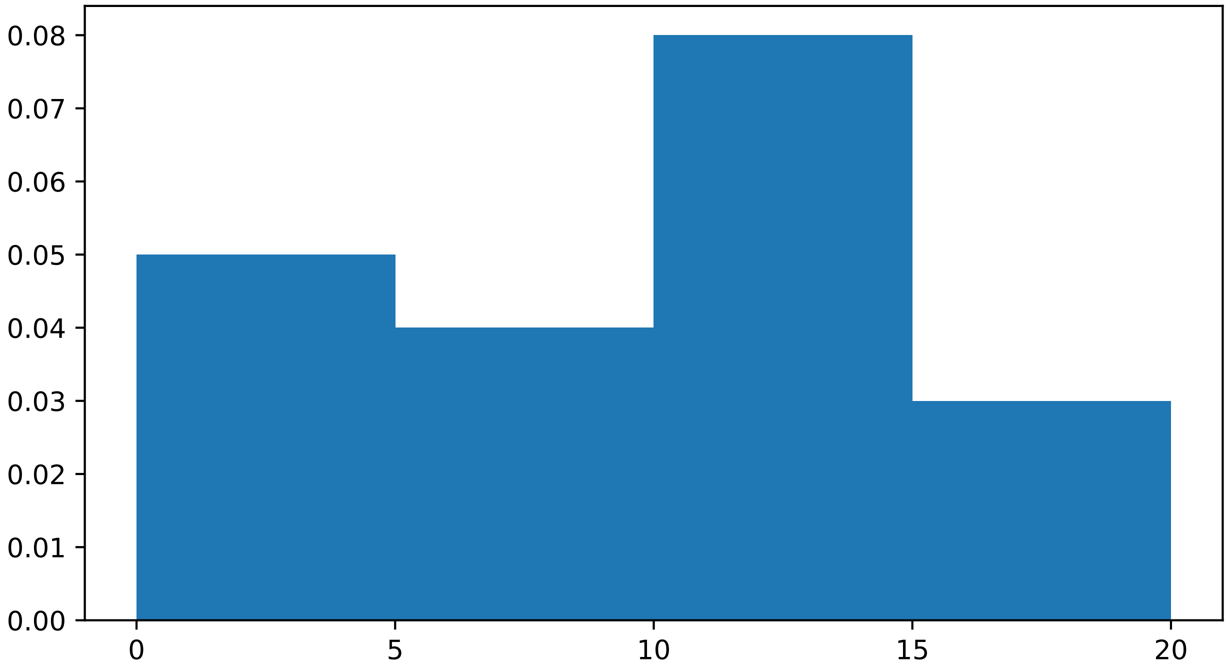
```
import matplotlib.pyplot as plt

x = [1, 1, 2, 3, 3, 5, 7, 8, 9, 10,
     10, 11, 11, 13, 13, 15, 14, 12, 18, 18]

plt.hist(x, bins=[0,5,10,15,20], cumulative=True)
plt.xticks([0,5,10,15,20])
plt.show()
```

```
## (array([ 5.,  9., 17., 20.]), array([ 0,  5, 10, 15, 20]), <BarContainer object of 4 artists>)
## ([<matplotlib.axis.XTick object at 0x000000006189B730>, <matplotlib.axis.XTick object at 0x000000006
```



```
import matplotlib.pyplot as plt

x = [1, 1, 2, 3, 3, 5, 7, 8, 9, 10,
     10, 11, 11, 13, 13, 15, 14, 12, 18, 18]

plt.hist(x, bins=[0,5,10,15,20], density=True)
plt.xticks([0,5,10,15,20])
plt.show()
```

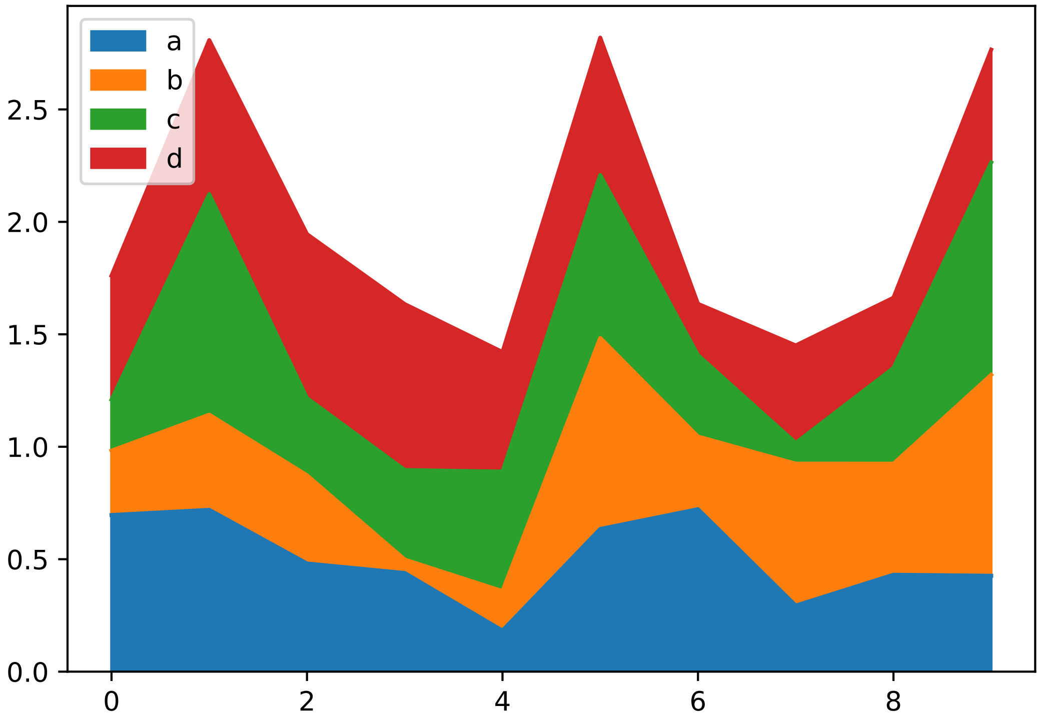
```
## (array([0.05, 0.04, 0.08, 0.03]), array([ 0,  5, 10, 15, 20]), <BarContainer object of 4 artists>)
## ([<matplotlib.axis.XTick object at 0x000000005EA30D30>, <matplotlib.axis.XTick object at 0x000000005
```



Wykres warstwowy:

https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.html

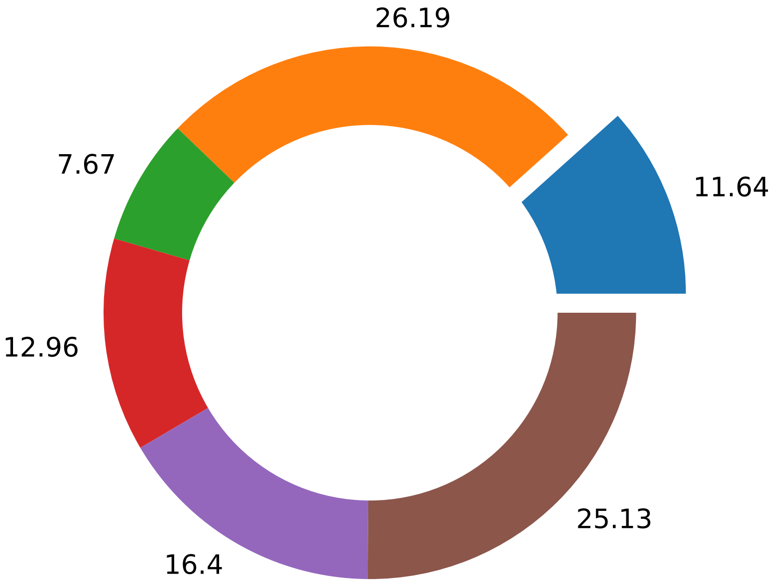
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

np.random.seed(123)

df = pd.DataFrame(np.random.rand(10, 4), columns=['a', 'b', 'c', 'd'])
df.plot.area()
plt.show()
```



Wykres pierścieniowy:

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

np.random.seed(345)
data = np.random.randint(20, 100, 6)
total = sum(data)
data_per = data/total*100
explode = (0.2, 0, 0, 0, 0, 0)
plt.pie(data_per, explode = explode, labels = [round(i,2) for i in (list(data_per))])
circle = plt.Circle( (0,0), 0.7, color='white')
p=plt.gcf()
p.gca().add_artist(circle)
plt.show()
```

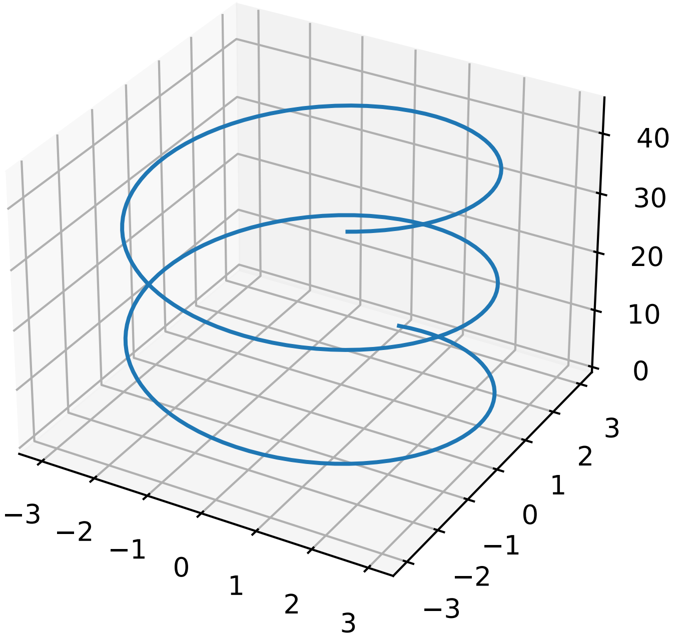
```
## ([<matplotlib.patches.Wedge object at 0x0000000061219940>, <matplotlib.patches.Wedge object at 0x0000
```



Helisa:

$$\begin{cases} x = a\cos(t) \\ y = a\sin(t) \\ z = at \end{cases}$$

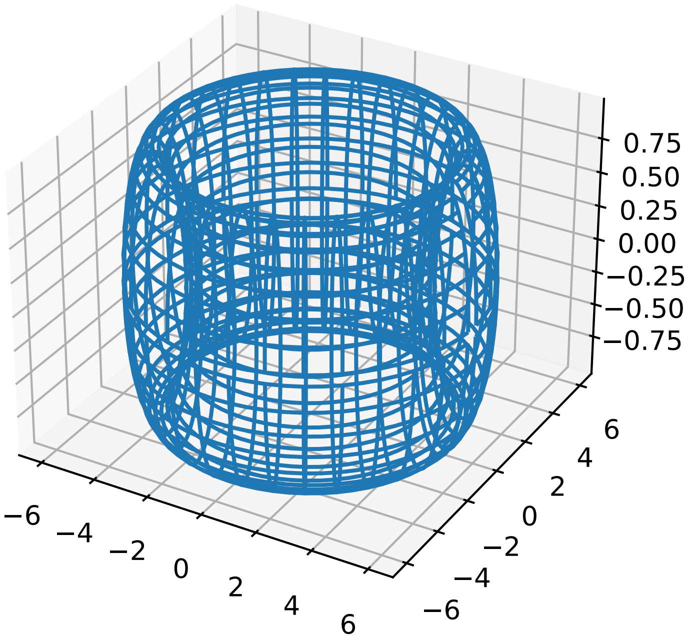
```
import numpy as np
import matplotlib.pyplot as plt

fig = plt.figure()
ax = plt.axes(projection='3d')
t = np.linspace(0, 15, 1000)
a = 3
xline = a * np.sin(t)
yline = a * np.cos(t)
zline = a * t
ax.plot3D(xline, yline, zline)
plt.show()
```



Torus

$$p(\alpha,\ \beta) = \Big((R + r\cos\alpha)\cos\beta,\ (R + r\cos\alpha)\sin\beta,\ r\sin\alpha\Big)$$

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

fig = plt.figure()
ax = plt.axes(projection='3d')
r = 1
R = 5
alpha = np.arange(0, 2 * np.pi, 0.1)
beta = np.arange(0, 2 * np.pi, 0.1)
alpha, beta = np.meshgrid(alpha, beta)
x = (R + r * np.cos(alpha)) * np.cos(beta)
y = (R + r * np.cos(alpha)) * np.sin(beta)
z = r * np.sin(alpha)
ax.plot_wireframe(x, y, z)
plt.show()
```



Źródło:

- https://www.geeksforgeeks.org/bar-plot-in-matplotlib/
- Dokumentacja https://matplotlib.org/
- https://datatofish.com/plot-histogram-python/
- https://jakevdp.github.io/PythonDataScienceHandbook/04.12-three-dimensional-plotting.html