



Politechnika  
Wrocławska

# Pakiety obliczeniowe

**INP001029WL**

**rok akademicki 2021/22**

**semestr zimowy**

## Wykład 4

**Karol Tarnowski**

**[karol.tarnowski@pwr.edu.pl](mailto:karol.tarnowski@pwr.edu.pl)**

**L-1 p. 220**



# Plan prezentacji (1)

- Skrypty, funkcje, funkcje anonimowe - kontynuacja
- Wykresy
  - różne typy wykresów
  - oznaczenia na wykresach
  - właściwości wykresów



# Plan prezentacji (2)

- Wykresy trójwymiarowe
- Edycja właściwości wykresów
  - ręczna
  - programowa



# Skrypty i funkcje

- Najprostszym typem programu Matlaba jest skrypt
- Fragment kodu wykonujący konkretne działanie można zamknąć w funkcji
- Szczególnym rodzajem funkcji są funkcje anonimowe



# Funkcja `eval`

- Funkcja `eval` wywołuje funkcję przekazaną jako uchwyt bądź napis
- Funkcja `eval` pozwala wykonać dowolny kod przekazany jako argument (w wielu przypadkach istnieją lepsze alternatywne rozwiązania)



# Rodzaje wykresów

```
%% przykłady różnych typów wykresów
```

```
clear  
clc
```

```
%% dane do przykładowych wykresów
```

```
x = [1:12];  
y = 20 + 10*cos(2*pi*x/12);
```

```
%% plot
```

```
figure()  
plot(x,y)  
title('plot(x,y)')
```

```
%% area
```

```
figure()  
area(x,y)  
title('area(x,y)')
```

```
%% stairs
```

```
figure()  
stairs(x,y)  
title('stairs(x,y)')
```

```
%% bar
```

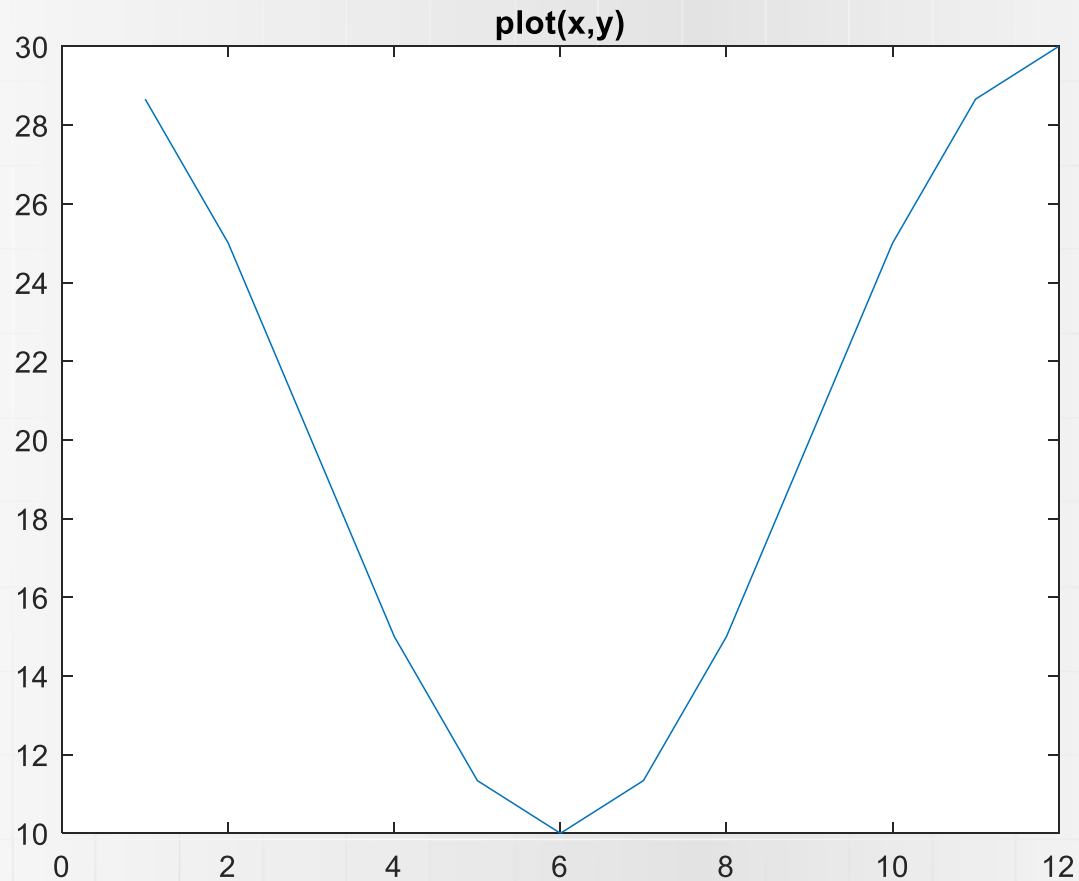
```
figure()  
bar(x,y)  
title('bar(x,y)')
```

```
%% stem
```

```
figure()  
stem(x,y)  
title('stem(x,y)')
```

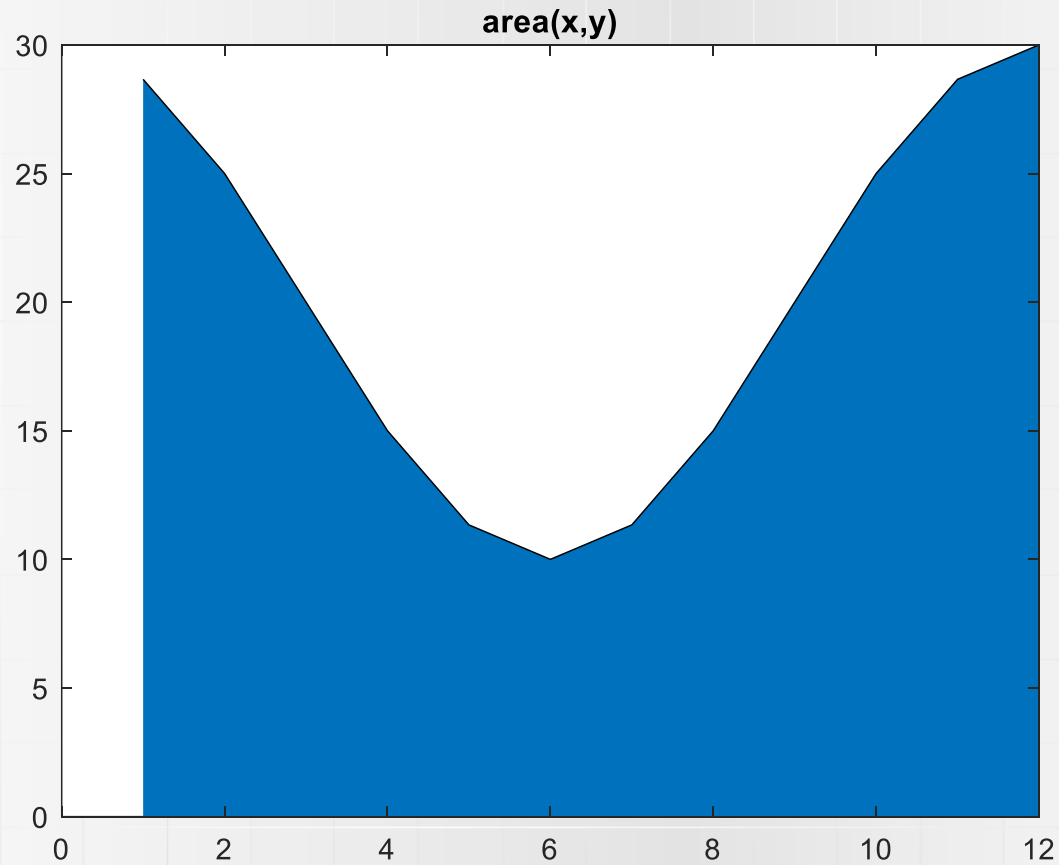


# Rodzaje wykresów





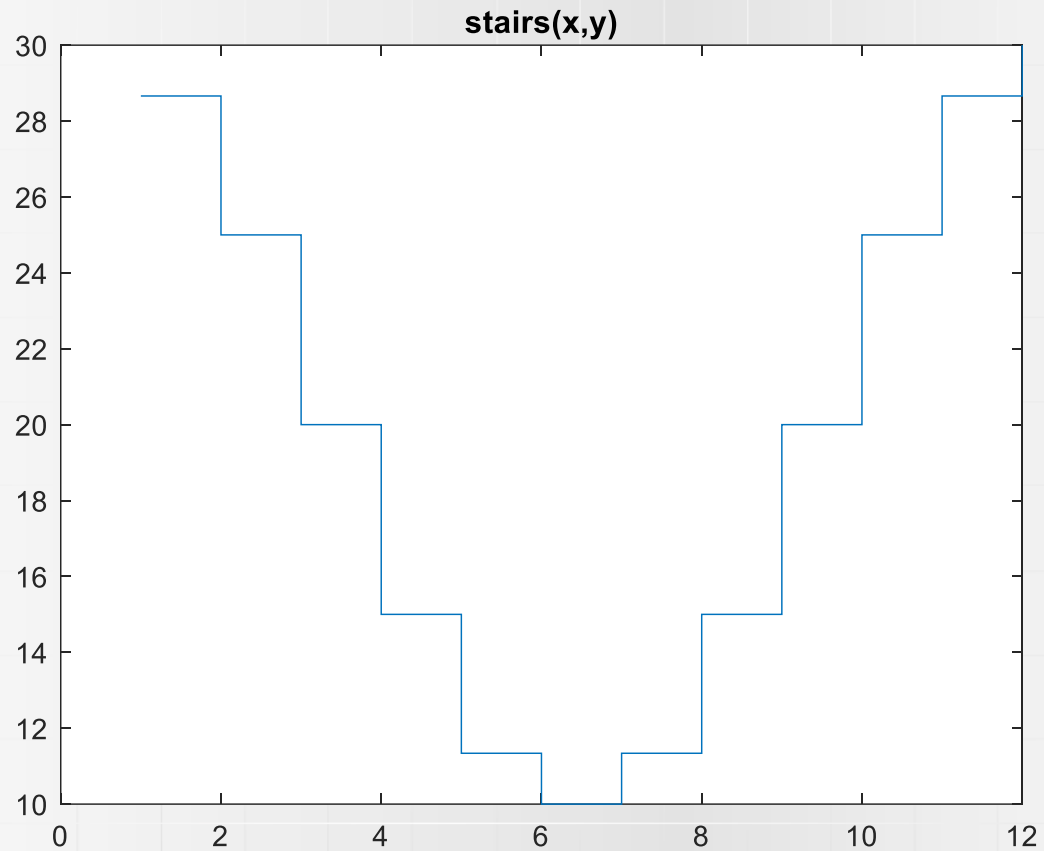
# Rodzaje wykresów





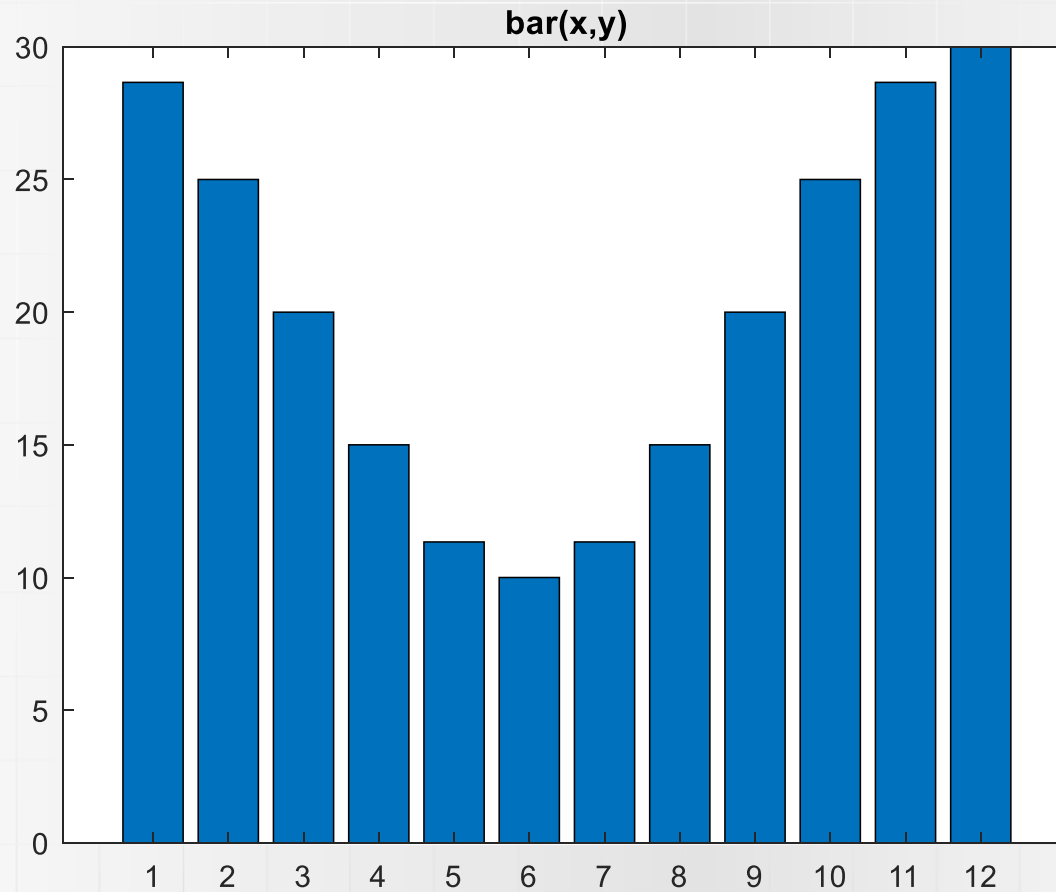


# Rodzaje wykresów



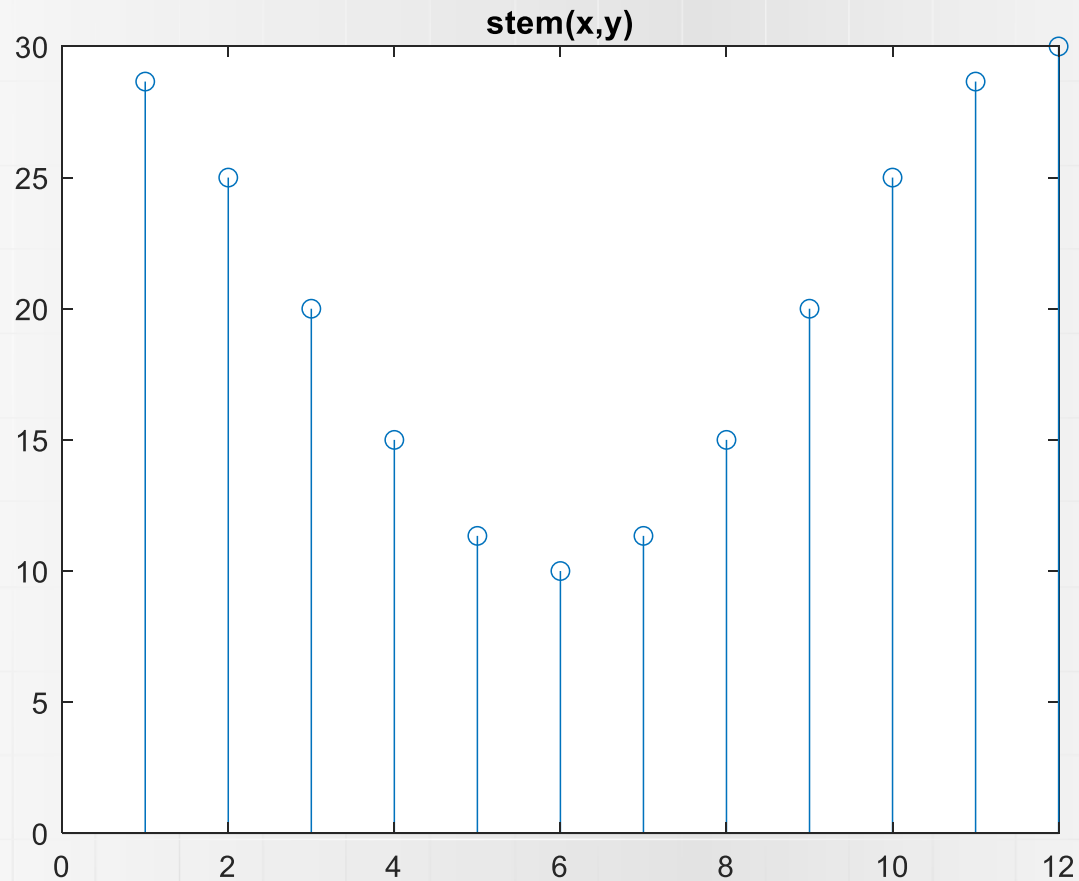


# Rodzaje wykresów





# Rodzaje wykresów





# Oznaczenia na wykresie

- Zestaw poleceń przydatnych do formatowania wykresów
  - `title` - dodaje tytuł do wykresu,
  - `xlabel`, `ylabel` - dodają opisy osi wykresu,
  - `xticks`, `yticks` - pozwalają ustawić znaczniki osi,
  - `xticklabel`, `yticklabel` - pozwalają modyfikować opisy znaczników osi



# Oznaczenia na wykresie

```
%% dane do przykładowych wykresów
```

```
x = [1:12];
```

```
y = 20 + 10*cos(2*pi*x/12);
```

```
%% bar
```

```
figure()
```

```
bar(x,y)
```

```
%% dodanie tytułu do wykresu
```

```
titleTxt = ["bar(x,y)" "drugi wiersz tytuł"]
```

```
title(titleTxt)
```

```
%% opis osi y
```

```
ylabel('Wielkość zależna od miesiąca')
```

```
%% modyfikacja znaczników osi x
```

```
xticks([1 4 7 10])
```

```
xticklabels(["Styczeń" "Kwiecień" "Lipiec" "Październik"])
```

```
xlabel("miesiąc")
```



# Właściwości wykresu

- Właściwości wykresu można zadawać wykorzystując dodatkowe argumenty funkcji w układzie 'własność' - 'wartość'
- Często używane:
  - `LineWidth`
  - `MarkerSize`
  - `MarkerEdgeColor`
  - `MarkerFaceColor`



# Właściwości wykresu

---

```
%% wygenerowanie losowych danych
```

```
x1 = rand(1,10);
```

```
x2 = rand(1,10);
```

---

```
%% podstawowy wykres
```

```
plot(x1,x2,'o')
```

```
xlabel("x_1") % opisy osi wykorzystujące interpreter TeX
```

```
ylabel("x_2")
```

---

```
%% nadanie właściwości wykresu
```

```
% z wykorzystaniem argumentów 'właściwość'-'wartość'
```

```
plot(x1,x2,'o','MarkerSize',8)
```



# Kolory

- Kolory można zadawać przez oznaczenia literowe:  $r$ ,  $g$ ,  $b$ ,  $c$ ,  $m$ ,  $y$ ,  $k$
- Kolory można zadawać podając wektor o trzech elementach kodujący kolor w RGB





# Kolory

```
% z wykorzystaniem argumentów 'właściwość'-'wartość'  
plot(x1,x2,'o', ...  
      'MarkerSize',8, ...  
      'MarkerFaceColor', [0 .9 .7], ...  
      'MarkerEdgeColor', 'r')
```



# Osie

- Polecenia przydatne do odczytu i ustawienia zakresów osi:
  - `axis`
  - `xlim`
  - `ylim`



# Osie

```
%% wygenerowanie losowych danych
```

```
x1 = 2*rand(1,10) - 1;
```

```
x2 = 4*rand(1,10) - 2;
```

```
%% podstawowy wykres
```

```
plot(x1,x2, 'o')
```

```
%% odczytanie domyślnych zakresów osi
```

```
xyLimits = axis
```

```
%% ustawienie zakresu osi x
```

```
% od 0 do poprzednie ograniczenie górne
```

```
xlim([0 xyLimits(2)])
```



# Osie

```
%% ustawienie zakresów osi do I ćwiartki
```

```
xlim([0 1])
```

```
ylim([0 2])
```

```
%% dopasowanie zakresów do danych
```

```
axis tight
```

```
%% ustawienie szerokiego zakresu danych
```

```
xlim([-1 1])
```

```
ylim([-2 2])
```

```
%%
```

```
axis tight
```

```
%% ustawienie szerokiego zakresu danych
```

```
% inny sposób
```

```
axis([-1 1 -2 2])
```

```
%% równe skale obu osi
```

```
axis equal
```



# Wykresy wielu kolumn/wierszy

- Funkcja plot pozwala rysować wiele kolumn/wierszy macierzy danych razem



# Wykresy wielu kolumn/wierszy

---

```
%% generowanie danych macierzowych
```

```
a = [-2:.5:2]';
```

```
x = linspace(-2,2,11);
```

```
y = a*x.^2;
```

---

```
%% wykres kolumn macierzy y (od numeru kolumny)
```

```
plot(y)
```

---

```
%% wykres kolumn macierzy y (od parametru a)
```

```
plot(a,y)
```

---

```
%% wykres wierszy macierzy y (od parametru x)
```

```
plot(x,y) % domyślne kolory linii
```

---

```
%% wykres wierszy macierzy y (od parametru x)
```

```
plot(x,y,'r--') % wszystkie linie w takim samym formatowaniu
```



# Wykresy trójwymiarowe

- Funkcje pozwalające tworzyć wykresy trójwymiarowe:
  - `surf`
  - `mesh`
  - `contour`



# Wykresy trójwymiarowe

```
%% generowanie danych macierzowych
```

```
x = linspace(-2,2,11);
```

```
y = [-2:.5:2]';
```

```
z = y*x.^2;
```

```
%% wykres powierzchni (od indeksów)
```

```
surf(z)
```

```
%% wykres powierzchni (od parametrów)
```

```
surf(x,y,z)
```

```
%% wykres powierzchni typu mesh
```

```
mesh(x,y,z)
```

```
%% wykres typu contour
```

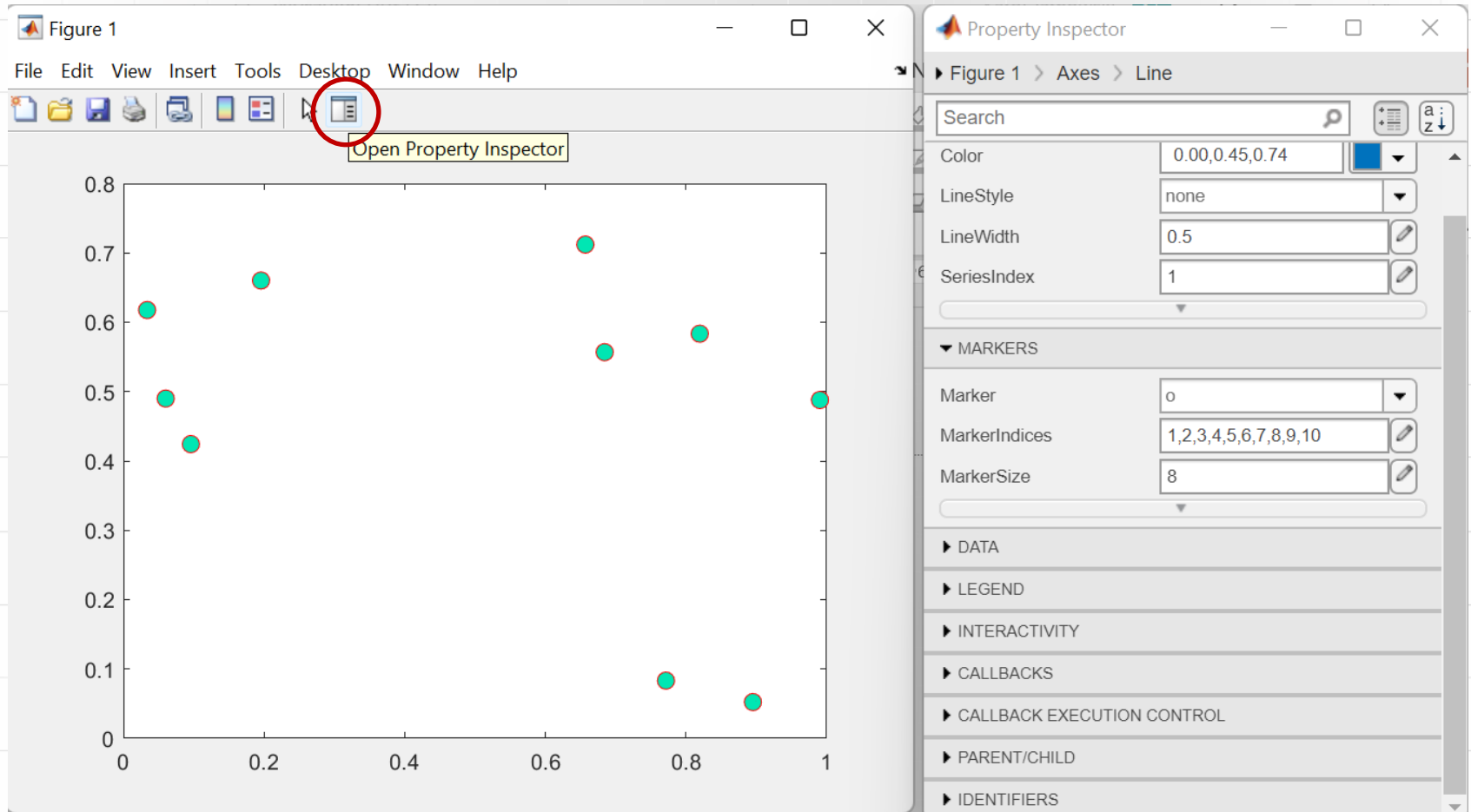
```
contour(x,y,z)
```

```
%% wykres pseudo color
```

```
pcolor(x,y,z)
```



# Edytor właściwości



The image shows a MATLAB window titled "Figure 1" with a scatter plot and a "Property Inspector" window. The scatter plot displays 10 data points with red circular markers. The x-axis ranges from 0 to 1, and the y-axis ranges from 0 to 0.8. The Property Inspector window is open to the "Line" properties of the selected series, showing settings for Color, LineStyle, LineWidth, SeriesIndex, MARKERS, and DATA.

**Figure 1**

File Edit View Insert Tools Desktop Window Help

Open Property Inspector

Property	Value
Color	0.00,0.45,0.74
LineStyle	none
LineWidth	0.5
SeriesIndex	1
MARKERS	
Marker	o
MarkerIndices	1,2,3,4,5,6,7,8,9,10
MarkerSize	8
DATA	
LEGEND	
INTERACTIVITY	
CALLBACKS	
CALLBACK EXECUTION CONTROL	
PARENT/CHILD	
IDENTIFIERS	

# Programowa edycja właściwości

- Właściwości obiektów graficznych można odczytywać/nadawać używając funkcji **get/set**

# Programowa edycja właściwości

```
%% wygenerowanie losowych danych
```

```
x1 = rand(1,10);
```

```
x2 = rand(1,10);
```

```
%% podstawowy wykres
```

```
p = plot(x1,x2,'o')
```

```
xlabel("x_1") % opisy osi wykorzystujące interpreter TeX
```

```
ylabel("x_2")
```

```
%% nadanie właściwości wykresu
```

```
set(p,'MarkerSize',8)
```

```
%% z wykorzystaniem argumentów 'właściwość'-'wartość'
```

```
set(p,'MarkerFaceColor','r')
```

```
%% z wykorzystaniem argumentów 'właściwość'-'wartość'
```

```
set(p, ...
```

```
    'MarkerFaceColor', [0 .9 .7], ...
```

```
    'MarkerEdgeColor', 'r')
```

# Programowa edycja właściwości

- Przydatne funkcje:
  - `meshgrid`
  - `subplot`
  - `figure`
  - `legend`
  - `gca`
  - `gcf`
  - `semilogy`
  - `semilogx`
  - `loglog`
  - `polar`
  - `plot3`
  - `plotyy`
  - `hist`
  - `colorbar`
  - `colormap`
  - `quiver`

# Programowa edycja właściwości

```
%% generowanie danych macierzowych
```

```
x = linspace(-1,1,11);  
y = [-2:.5:2]';  
[X Y] = meshgrid(x,y)  
Z = X.^2 + Y.^3;
```

```
%% wykorzystanie funkcji subplot do umieszczenia
```

```
% kilku wykresów jednym oknie  
subplot(3,3,[5 9])  
pcolor(X, Y, Z)
```

```
%% dodanie legendy do wykresu
```

```
subplot(3,3,[2 3])  
plot(x,sum(Z),x,mean(Z))  
l1 = legend("sum(Z)", "mean(Z)")  
set(l1,'Location','North') %nadanie właściwości legendy z wykorzystaniem  
% funkcji set
```

```
subplot(3,3,[4 7])  
plot(sum(Z'),y)  
l2 = legend("sum(Z')")  
set(l2,'Location','South')
```



# Podsumowanie

- Skrypty, funkcje, funkcje anonimowe
- Wykresy