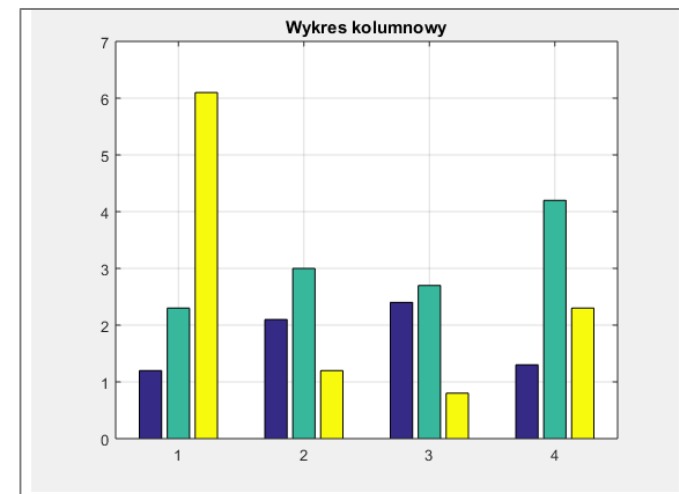
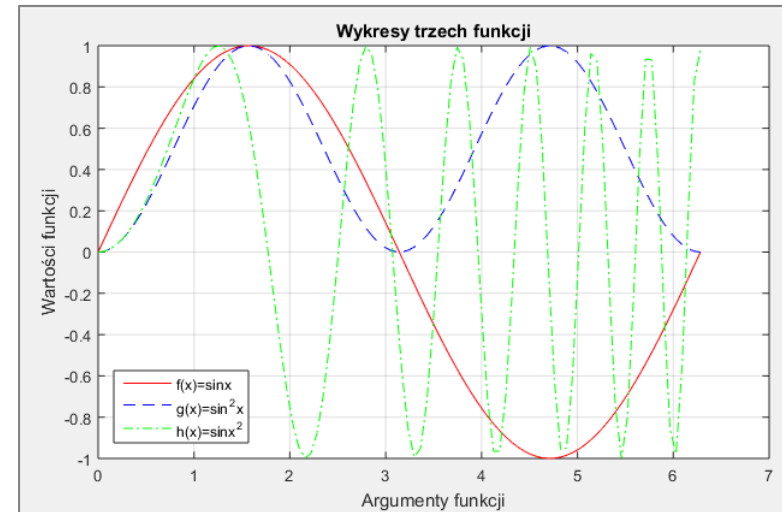
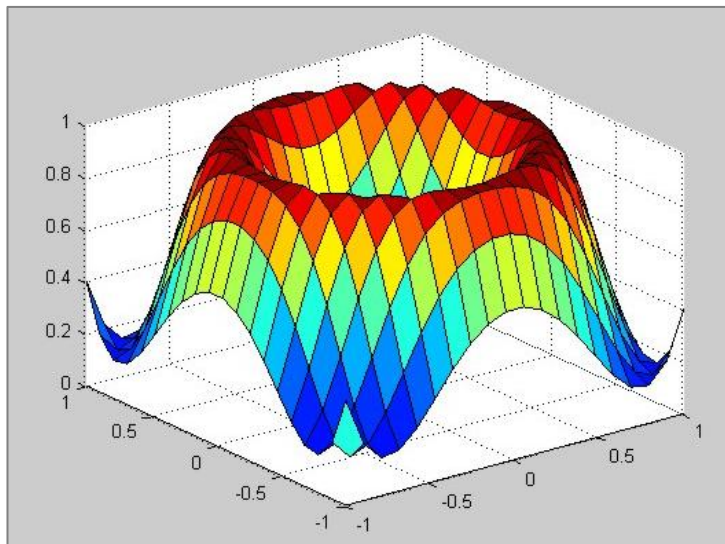


# *Grafika*

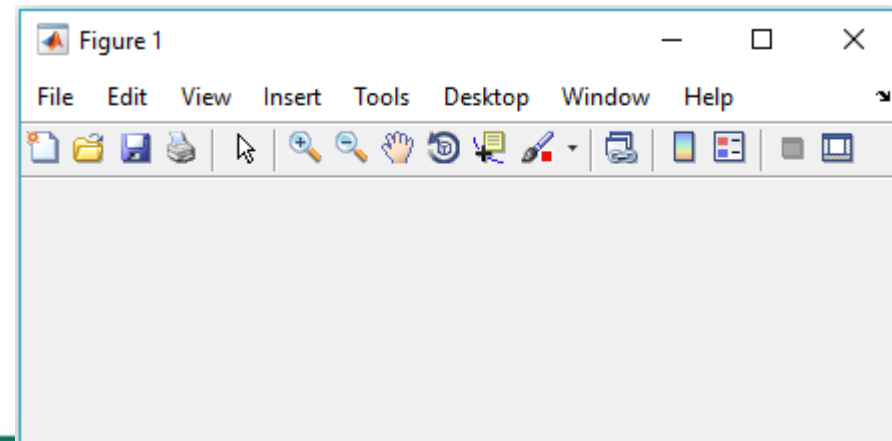
# Funkcje graficzne – podstawowe grupy

- ❖ wykresy 2D i 3D
- ❖ wykresy ciągłe i dyskretne
- ❖ grafika wektorowa i rastrowa
- ❖ wysokiego i niskiego poziomu



# Okno graficzne

- ❖ Wywołanie funkcji graficznej powoduje automatyczne otwarcie okna graficznego
- ❖ Funkcje obsługujące okna graficzne:
  - ***figure*** – tworzy nowe okno i uaktywnia je
  - ***figure(n)*** – uaktywnia okno o numerze n lub tworzy nowe
  - ***close*** – zamyka aktywne okno
  - ***close(n)*** – zamyka okno o numerze n
  - ***close all*** – zamyka wszystkie okna
  - ***clf*** – usuwa zawartość aktywnego okna
  - ***hold on*** – zachowuje bieżący widok okna graficznego
  - ***hold off***



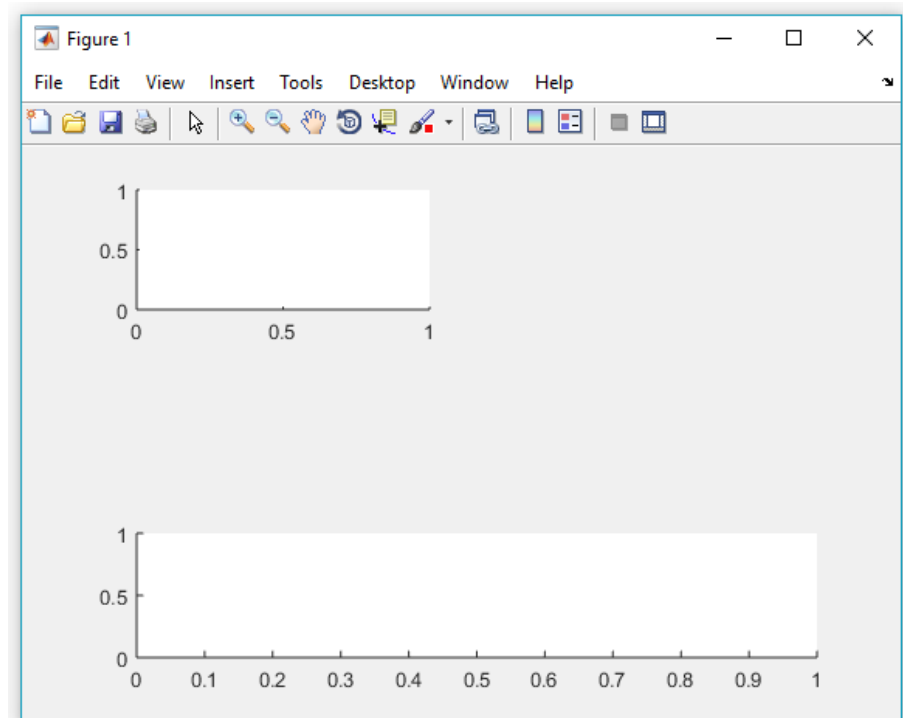
# Okno graficzne – podział

- ❖ ***subplot(w, k, nr)*** – dzieli okno graficzne na  $m \times n$  okienek (gdzie ***w***, ***k*** liczba wierszy i kolumn), oraz uaktywnia okienko o numerze ***nr*** (numeracja kolejno wierszami od lewej do prawej); tworzy wiele obszarów rysowania (np. dla kilku wykresów) w obrębie jednego okna graficznego.

*Przykład*

`subplot(3,2,1)`

`subplot(3,1,3)`



# *Grafika dwuwymiarowa*

## Polecenie *plot*

- ❖ ***plot(x,y)***, ***plot(x,y,s)*** – rysuje wykres elementów wektora  $x$  względem elementów wektora  $y$ 
  - $x$  i  $y$  są wektorami jednakowej długości, reprezentującymi współrzędne punktów na wykresie
  - $s$  jest łańcuchem znakowym definiującym parametry graficzne wykresu
  - jeżeli  $y$  jest macierzą, to wykreślonych zostanie na wykresie tyle serii, ile jest kolumn macierzy  $y$
  - jeżeli pominięty zostanie wektor  $x$ , to jako rzędne zostaną przyjęte indeksy elementów macierzy  $y$
- ❖ Wprowadzenie na wykres kilku serii niezależnych:  
***plot(x1,y1,s1,x2,y2,s2,...)***
- ❖ ***linspace(xp,xk,n)*** – generuje wierszowy wektor  $n$  liczb rozłożonych równomiernie w przedziale od  $xp$  do  $xk$

# Parametry graficzne

- ❖ Parametry graficzne podawane są w postaci łańcucha znakowego (w apostrofach), w którym określamy rodzaj linii, kolor oraz kształt znacznika punktu.
- ❖ Parametry te można dowolnie permutować w łańcuchu s.

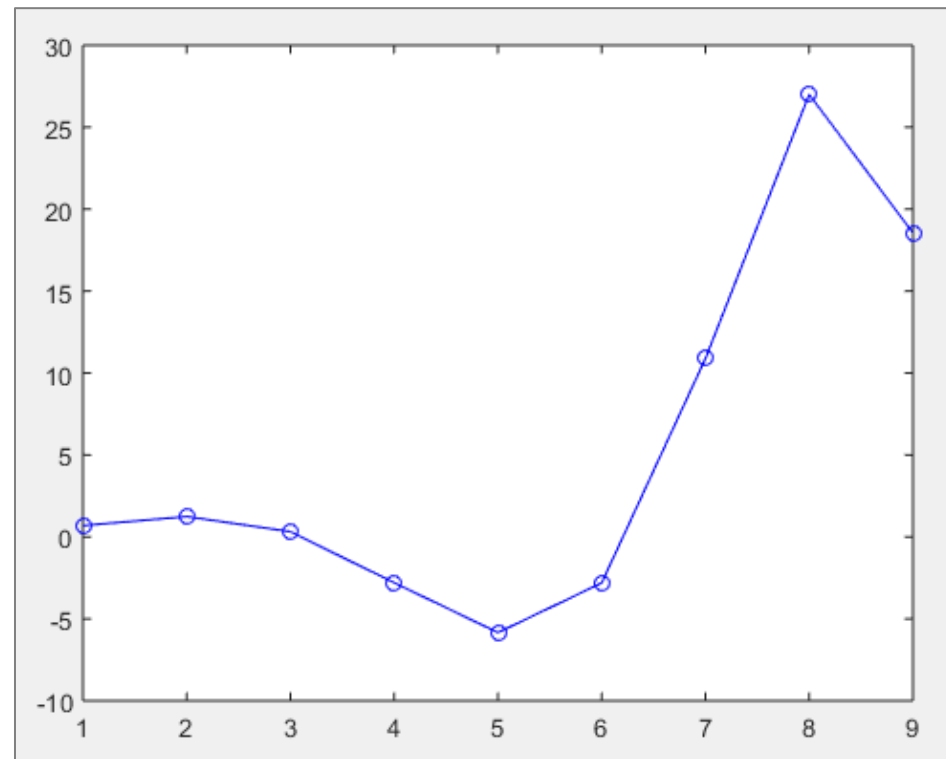
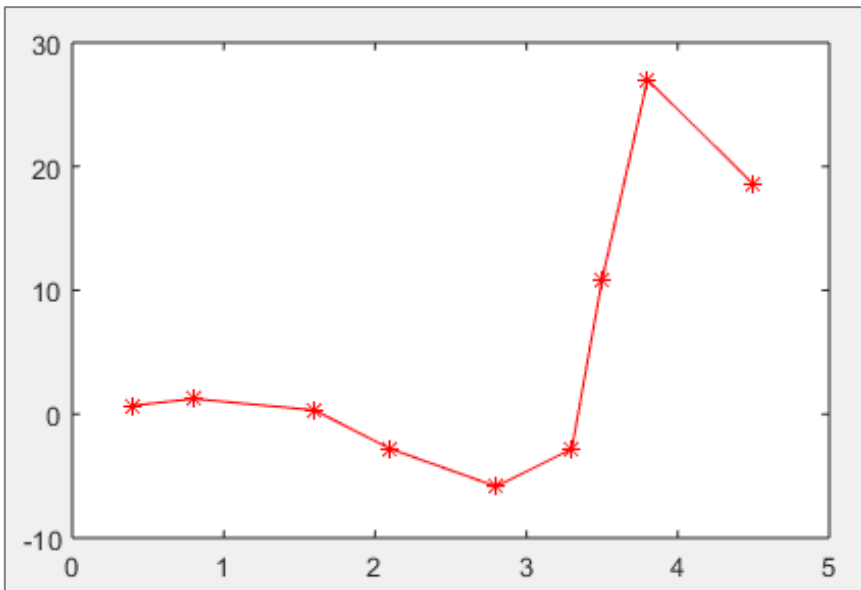
Znaki	Rodzaj linii
-	ciągła
--	kreskowana
:	kropkowana
-.	kreska-kropka

Znaki	Rodzaj linii
y	żółty
m	karmazynowy
c	turkusowy
r	czerwony
g	zielony
b	niebieski
w	biały
k	czarny

Znaki	Rodzaj linii
*	gwiazdka
+	krzyżyk
.	kropka
o	kółko
x	iks
s	kwadrat
d	romb
^	trójkąt

## Polecenie `plot` – przykład

```
s=[0.4 0.8 1.6 2.1 2.8 3.3 3.5 3.8 4.5];  
w=[0.69 1.24 0.32 -2.8 -5.84 -2.81 10.88 27 18.55];  
plot(s,w, 'r*-')  
figure  
plot(w, 'bo-')
```



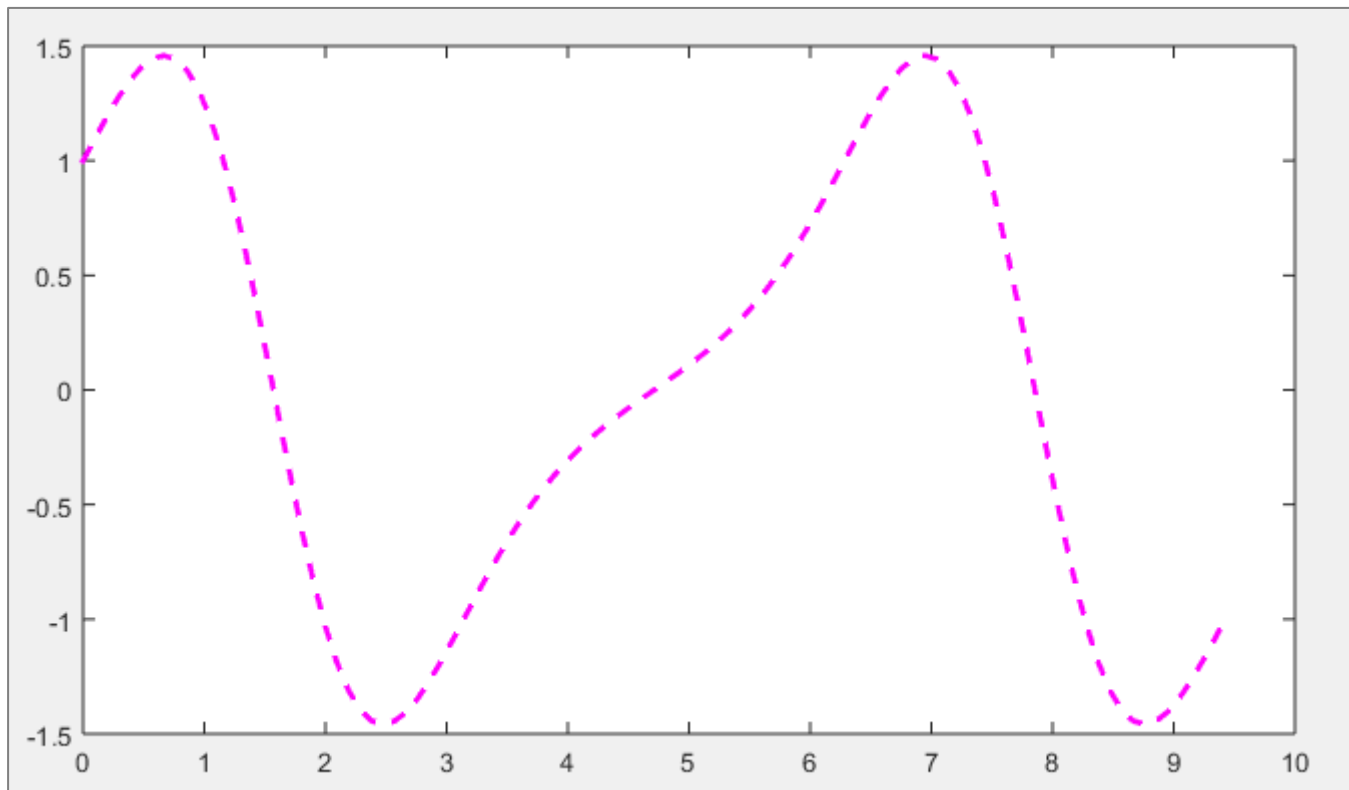


## Dodatkowe parametry graficzne

- ❖ Umieszczamy je za łańcuchem parametrów graficznych: `plot(x,y,s,par1, wart1,par2,war2,...)` – `par_n` podajemy jako łańcuch znakowy:
  - **'LineWidth', pt** – grubość linii, w punktach
  - **'MarkerSize', pt** – wielkość znaku, w punktach
  - **'MarkerFaceColor','zn'** – kolor wnętrza znaku; zn – symbol koloru jak w łańcuchu `s`
  - **'MarkerEdgeColor','zn'** – kolor krawędzi znaku; zn – symbol koloru jak w łańcuchu `s`
  - **'Color',[r g b]** – kolor podawany jako trójka liczb w przestrzeni RGB (czerwień, zieleń, niebieski) – wartości `r g b` w zakresie  $\langle 0,1 \rangle$

# Dodatkowe parametry graficzne

```
x=linspace(0,3*pi);  
y=cos(x).*exp(sin(x));  
plot(x,y,'m--','LineWidth',2)
```

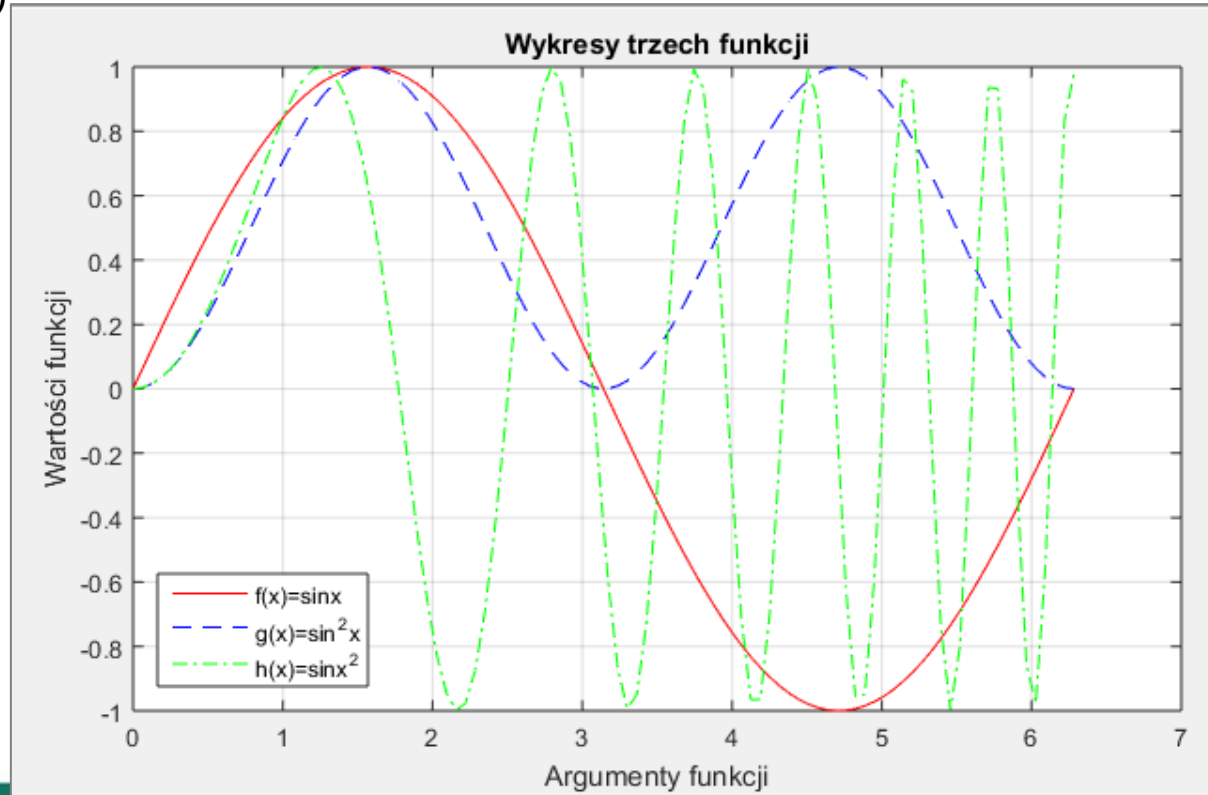


# Dodatkowe elementy wykresu

- ❖ Tytuł: ***title('To jest tytuł')***
- ❖ Opis osi: ***xlabel('opis osi X')***; podobnie: ***ylabel, xlabel***
- ❖ Tekst na wykresie: ***text(x,y,'Tekst na wykresie')*** – x, y oznaczają współrzędne tekstu na wykresie
- ❖ Legenda: ***legend('opis1','opis2',...)*** – opisy w kolejności kreślenia serii
- ❖ Siatka: ***grid on*** (włączona), ***grid off*** (wyłączona)

# Dodatkowe elementy wykresu

```
x=linspace(0,2*pi);
plot(x, sin(x),'r-', x, sin(x).^2,'b--', x, sin(x.^2),'g-.'.')
grid on
legend('f(x)=sinx', 'g(x)=sin^2x', 'h(x)=sinx^2', 'Location', 'best')
title('Wykresy trzech funkcji')
xlabel('Argumenty funkcji')
ylabel('Wartości funkcji')
```



## Skalowanie wykresów – osie

- ❖ ***axis([xmin, xmax, ymin, ymax, zmin, zmax])*** – ustawia zakres osi; trzeba podać wszystkie liczby (nawet te, których nie zmieniamy); dla wykresu 2D pomijamy *z*
- ❖ Zakres pojedynczej osi: ***xlim([xmin, xmax])***; podobnie: ***ylim, zlim***
- ❖ ***axis('equal')*** – ustawia jednakową skalę dla obu osi
- ❖ ***axis('square')*** – przekształca domyślną prostokątną ramkę w kwadratową
- ❖ ***axis('normal')*** – przywraca domyślne wartości na osiach
- ❖ ***axis('image')*** – zmienia rozmiary osi na takie same jak rozmiar wykresu
- ❖ ***axis('off/on')*** – usuwa/przywraca ramkę otaczającą wykres oraz znaczniki podziałki

# Skalowanie wykresów – osie

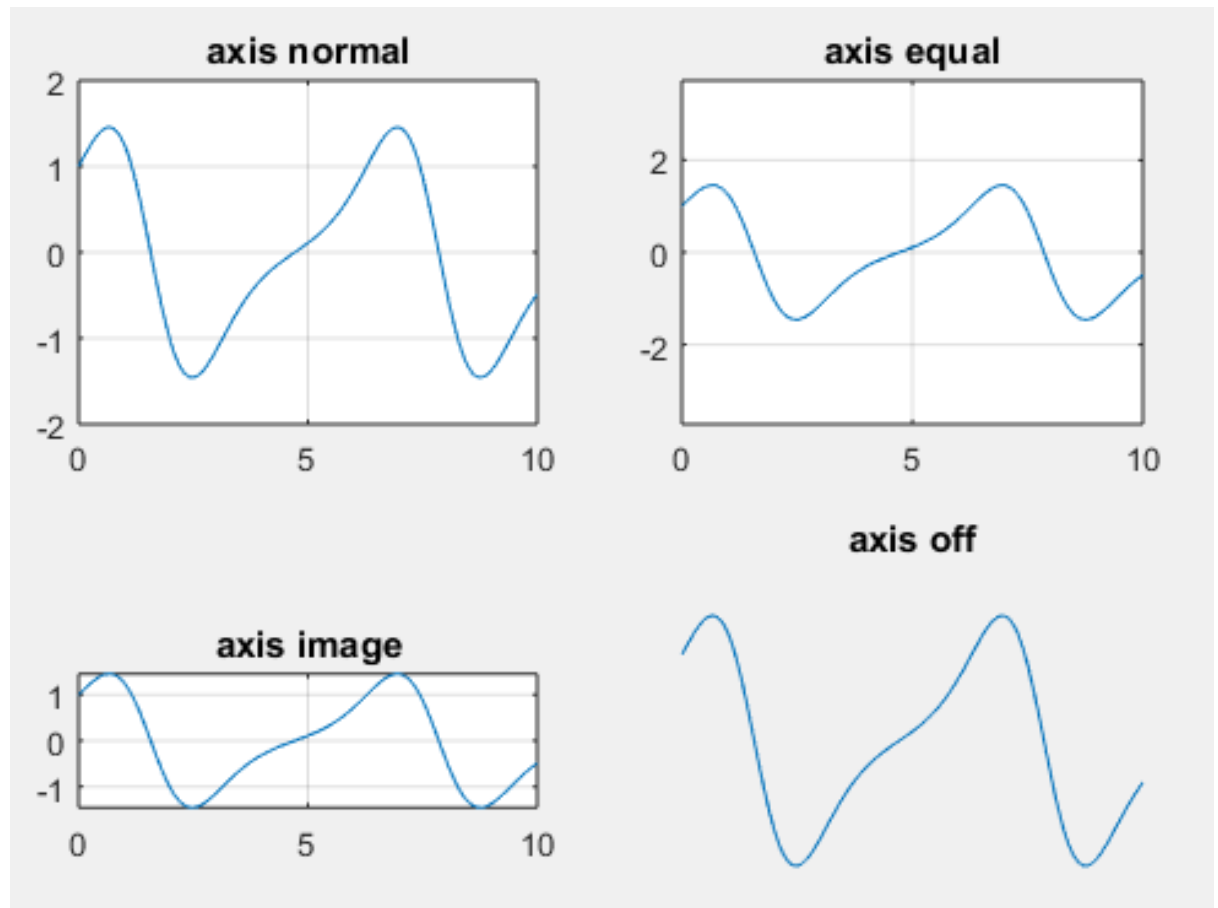
```
x=0:0.1:10;
y=cos(x).*exp(sin(x));
```

```
subplot(2,2,1)
plot(x,y); grid on
title('axis normal')
```

```
subplot(2,2,2)
plot(x,y); grid on
axis equal
title('axis equal')
```

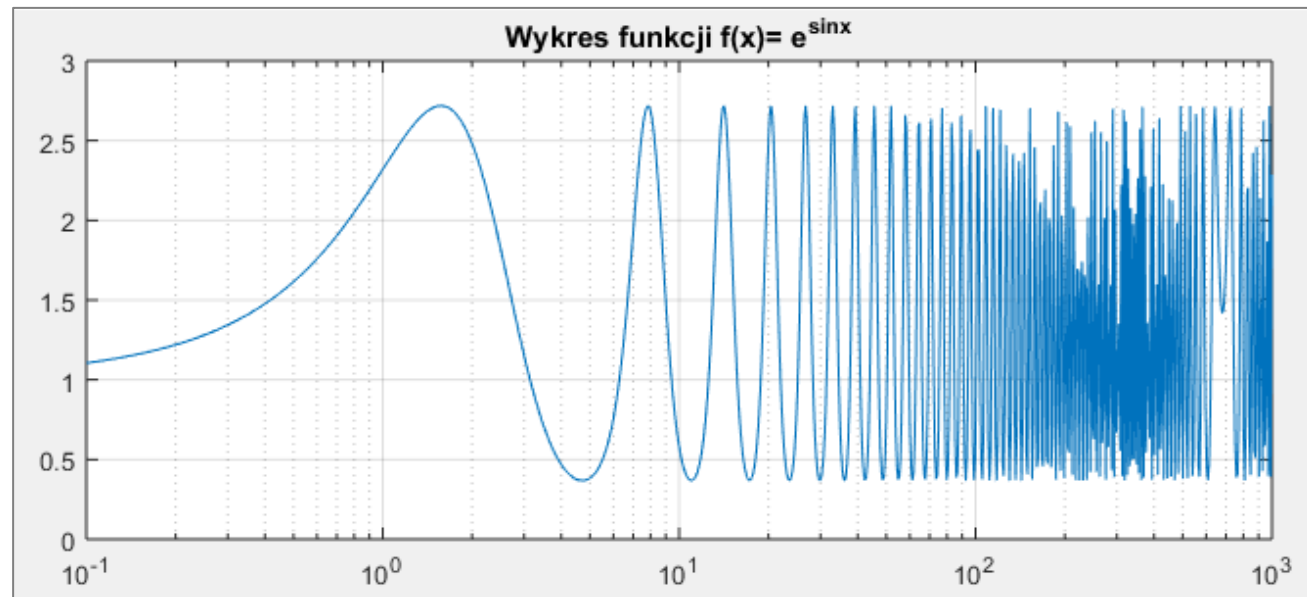
```
subplot(2,2,3)
plot(x,y); grid on
axis image
title('axis image')
```

```
subplot(2,2,4)
plot(x,y)
axis off
title('')
```



# Wykresy w skali logarytmicznej

- ❖ ***loglog(x,y)*** – rysuje wykres z użyciem skali logarytmicznych na obu osiach
- ❖ ***semilogx(x,y)*** – rysuje wykres z użyciem skali logarytmicznej na osi x
- ❖ ***semilogy(x,y)*** – rysuje wykres z użyciem skali logarytmicznej na osi y
- ❖ ***logspace(xp,xk,n)*** – generuje wektor wierszowy n liczb rozmieszczonych logarytmicznie między wartościami  $10^{x1}$  a  $10^{x2}$



Przykład

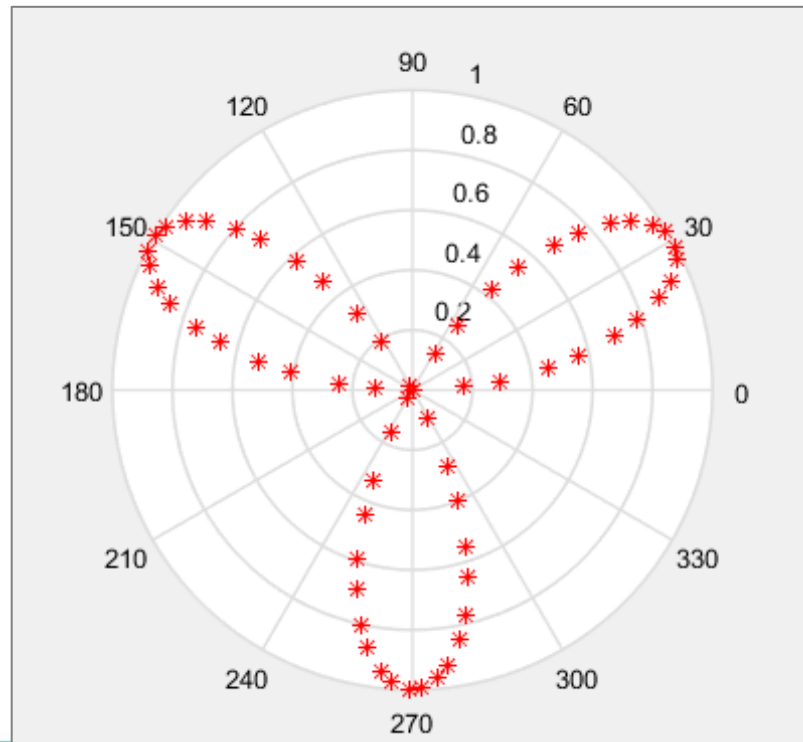
```
x=logspace(-1,3,1000);
semilogx(x,exp(sin(x)));
grid on
title('Wykres funkcji f(x)=e^{sinx}')
```

# Wykresy w biegunowym układzie współrzędnych

- ❖ ***polar(theta, r, s)*** – rysuje wykres w biegunowym układzie współrzędnych  $(\theta, r)$  z argumentami:
  - ***theta*** – wektor kątów (w radianach) dla poszczególnych punktów
  - ***r*** – wektor odległości poszczególnych punktów od początku układu współrzędnych

*Przykład*

```
t=-pi:0.1:pi;  
polar(t,sin(3*t),'r*')
```





# Wykresy funkcji podanych wzorem

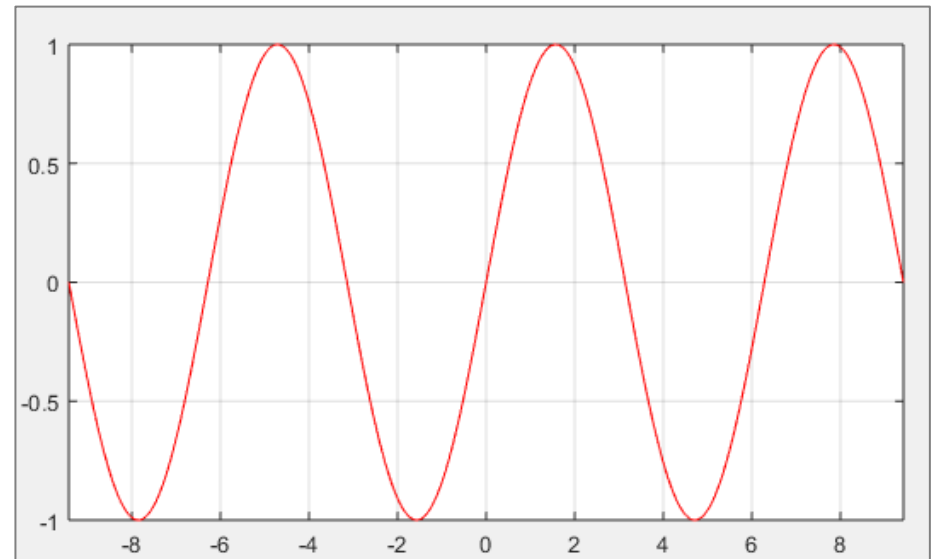
- ❖ `fplot('funkcja', [xmin xmax], s)` – wykres funkcji jednej zmiennej w przedziale od `xmin` do `xmax`
- ❖ `ezplot('funkcja', [xmin xmax])` – automatycznie dodaje tytuł wykresu i opis osi `x`
- ❖ `ezpolar('funkcja', [xmin xmax])` – wersja polecenia `ezplot` dla współrzędnych biegunowych

## Przykład

```
fplot('sin', [-3*pi, 3*pi], 'r')
```

```
fplot('sin(x)', [-3*pi, 3*pi], 'r')
```

```
fplot(@(x) sin(x), [-3*pi, 3*pi], 'r')
```



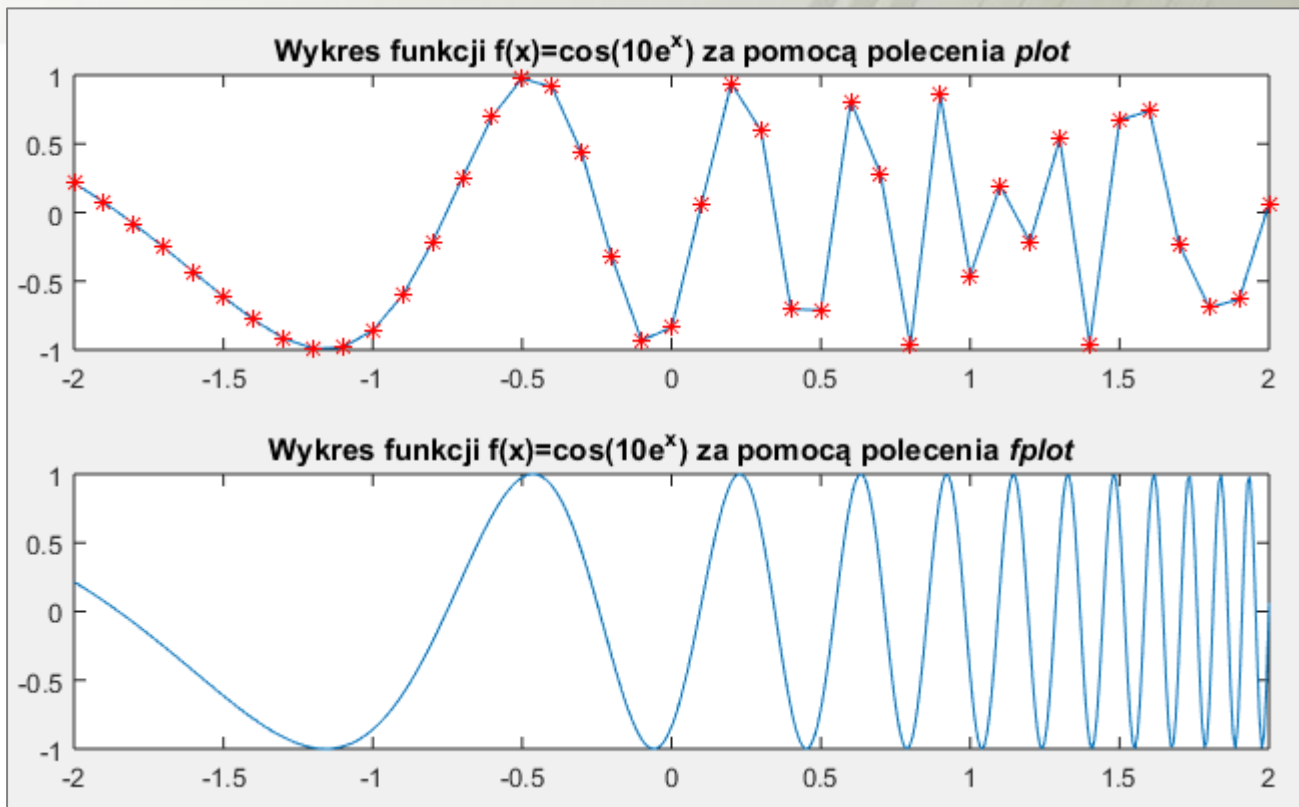
# Wykresy funkcji

```
function y = cos10e(x)
y=cos(10*exp(x));
end
```

```
x=-2:0.1:2;
y=cos10e(x);
```

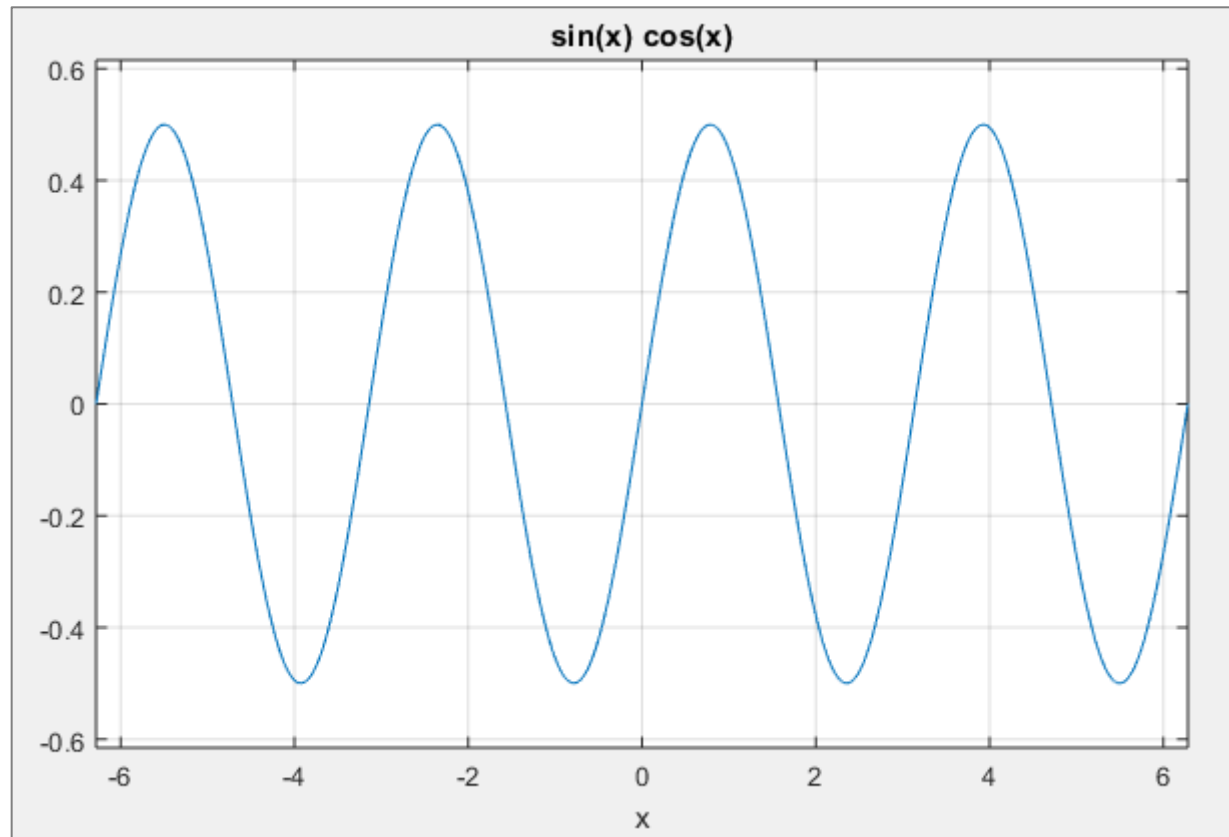
```
subplot(2,1,1)
plot(x,y,'*-', 'MarkerEdgeColor','r' )
title('Wykres funkcji f(x)=cos(10e^x) za pomocą polecenia \itplot')
```

```
subplot(2,1,2)
fplot('cos10e',[-2 2])
title('Wykres funkcji f(x)=cos(10e^x) za pomocą polecenia \itfplot')
```



# Wykresy funkcji podanych wzorem

```
ezplot('sin(x)*cos(x)')  
grid on
```

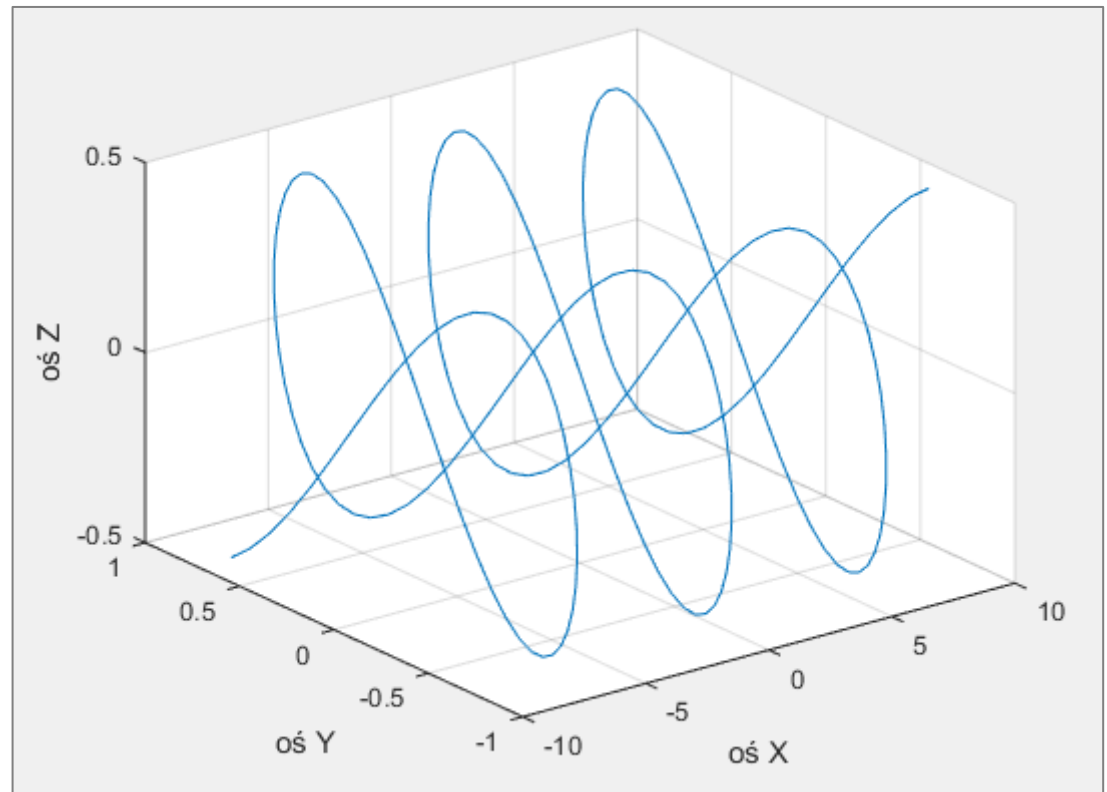


# *Grafika 3D*

# Funkcje kreślące w 3D

- ❖ Odmiiany funkcji kreślących w 2D: ***plot3(x,y,z)***

```
x=-10:0.1:10;  
y=sin(x);  
z=cos(x).*sin(y);  
plot3(x,y,z)  
grid on  
xlabel('oś X')  
ylabel('oś Y')  
zlabel('oś Z')
```



# Funkcje kreślące w 3D

## ❖ Wykresy powierzchniowe:

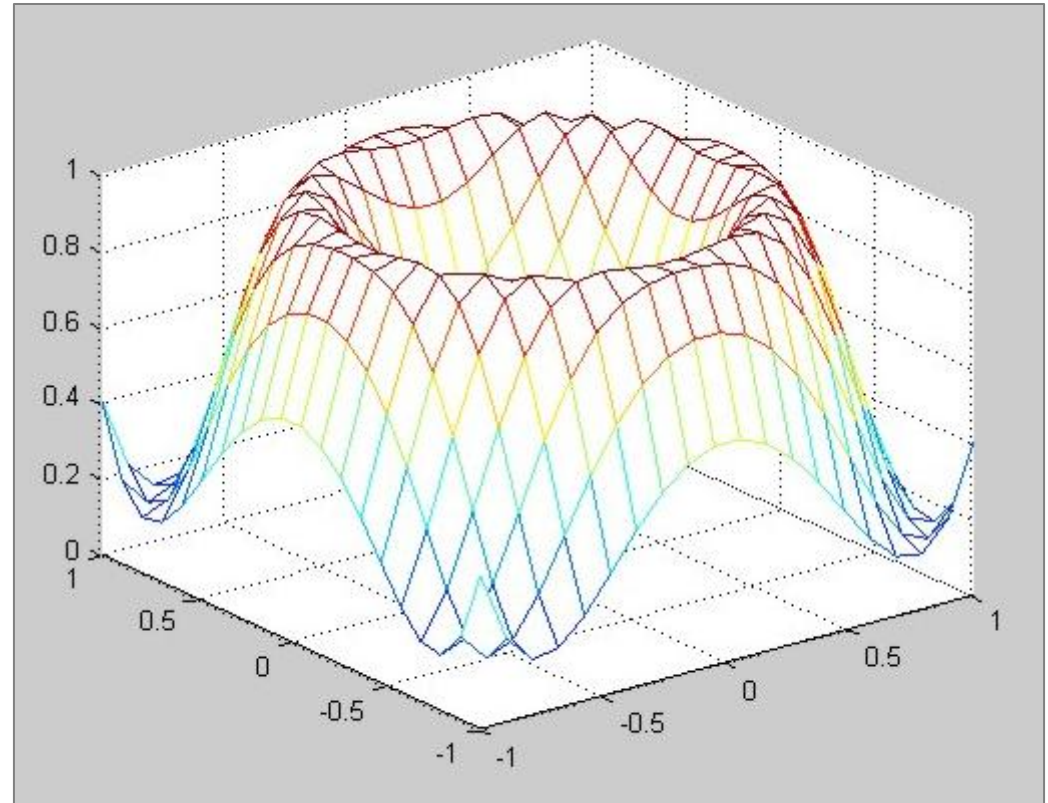
- są to wykresy  $x$ ,  $y$ ,  $z$ , gdzie  $z$  opisuje kształt powierzchni rozpiętej nad płaszczyzną  $XY$
- aby móc je wykonać, należy przygotować siatkę współrzędnych  $XY$  za pomocą funkcji ***meshgrid***

## ❖ $[X,Y]=\text{meshgrid}(x,y)$

- $x$  i  $y$  są wektorami współrzędnych odpowiednio o długościach  $n$  i  $m$
- $X$  i  $Y$  są tablicami współrzędnych odpowiednio o wymiarach  $m \times n$  i  $n \times m$
- $X$  jest wypełnione wierszami  $x$ ,  $Y$  jest wypełnione kolumnami  $y$
- $Z$  określamy jako funkcję na  $X$  i  $Y$  (a nie na  $x$  i  $y$  !)
- $\text{meshgrid}(x)$  jest równoważne  $\text{meshgrid}(x,x)$

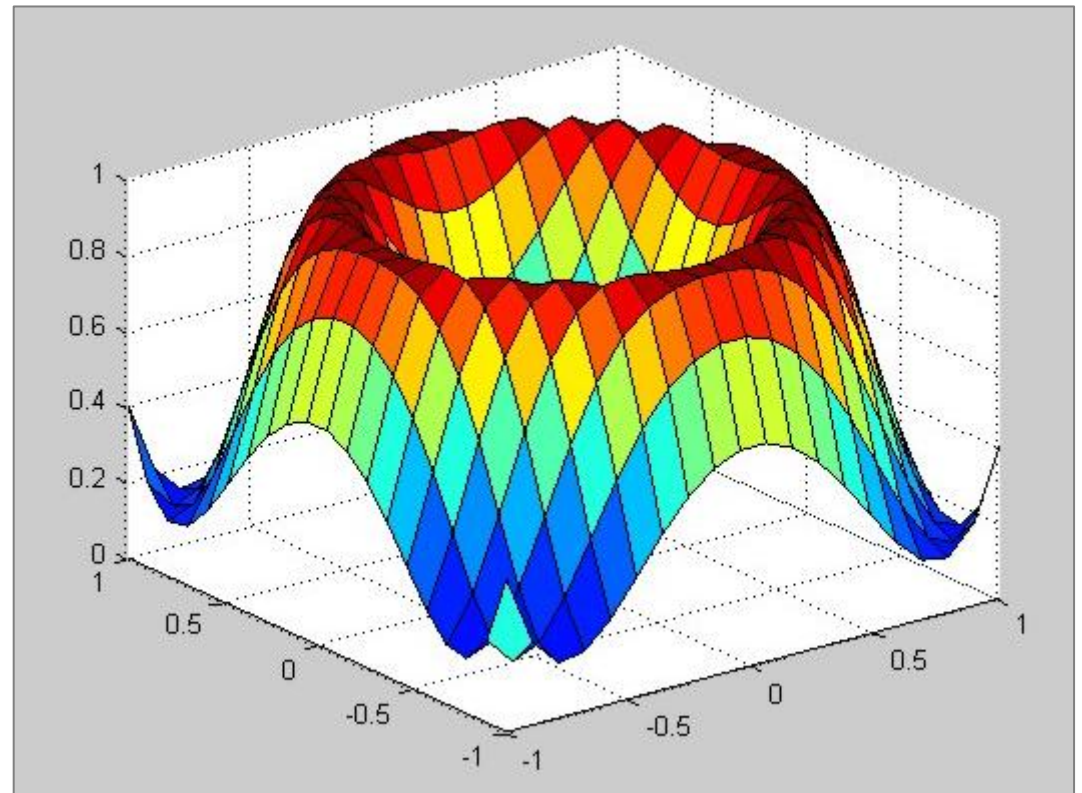
# Wykres siatkowy

❖ Składnia: *mesh(X,Y,Z)*



# Wykres powierzchniowy

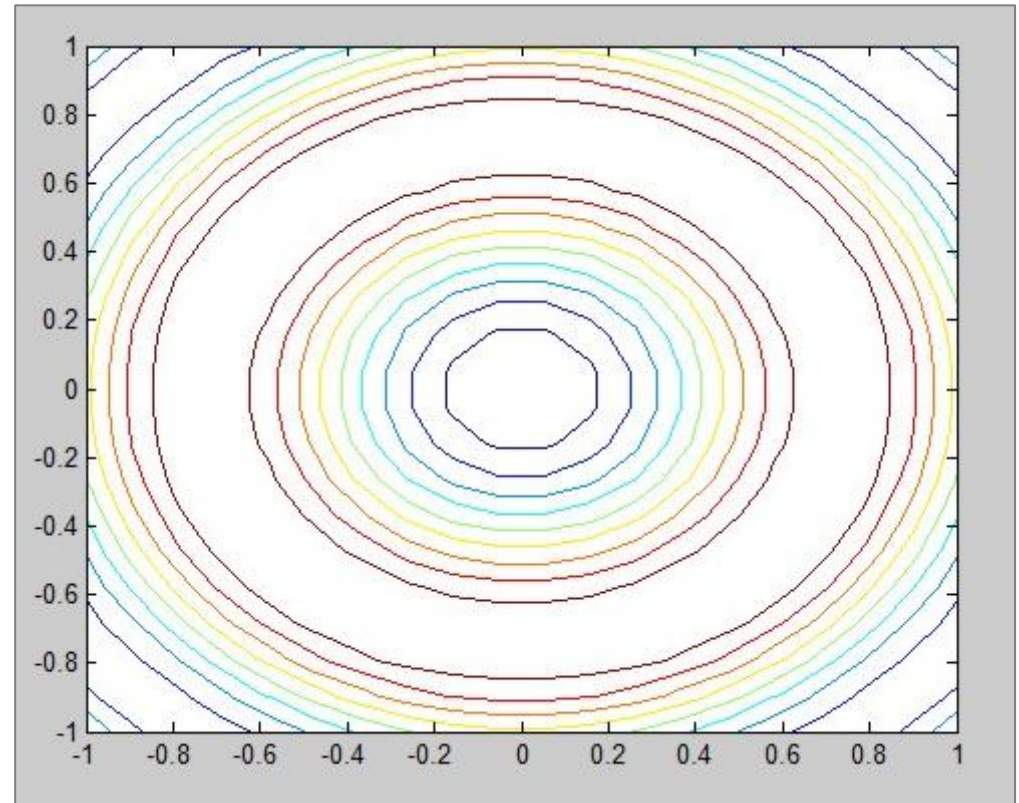
❖ Składnia: *surf(X,Y,Z)*





# Wykres poziomicowy

❖ Składnia: *contour(X,Y,Z)*



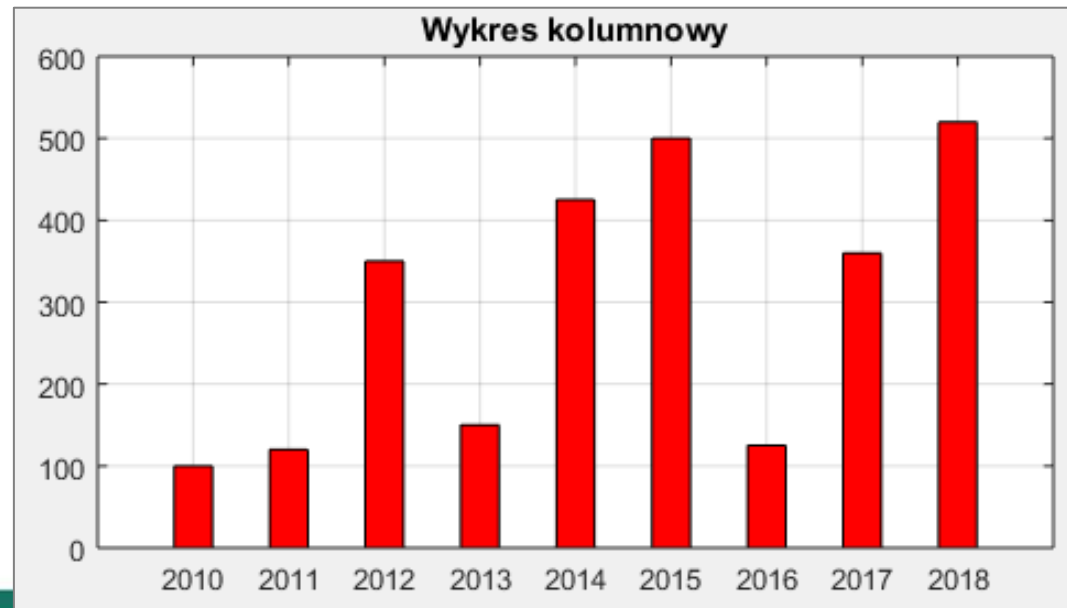
# *Prezentacja danych dyskretnych*

# Wykresy kolumnowe i słupkowe

- ❖ ***bar(x, y, w, s, styl)*** – rysuje kolumny o wysokości kolejnych elementów wektora *y* w punktach określonych przez elementy wektora *x*, *s* określa wygląd rysowanej linii, *w* określa szerokość kolumn, (domyślnie *w*=0,8)
- ❖ ***barh(x, y, w, s, styl)*** – rysuje wykres słupkowy
- ❖ ***bar3(x, y, w, s, styl)*** – rysuje trójwymiarowe kolumny
- ❖ ***bar3h(x, y, w, s, styl)*** – rysuje trójwymiarowe słupki

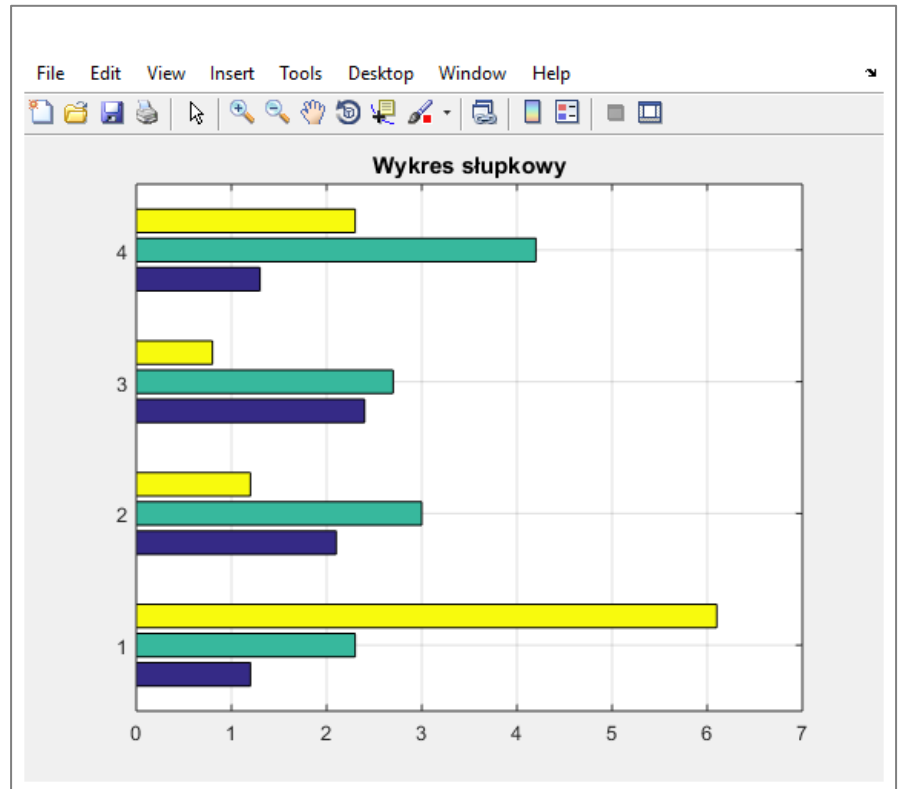
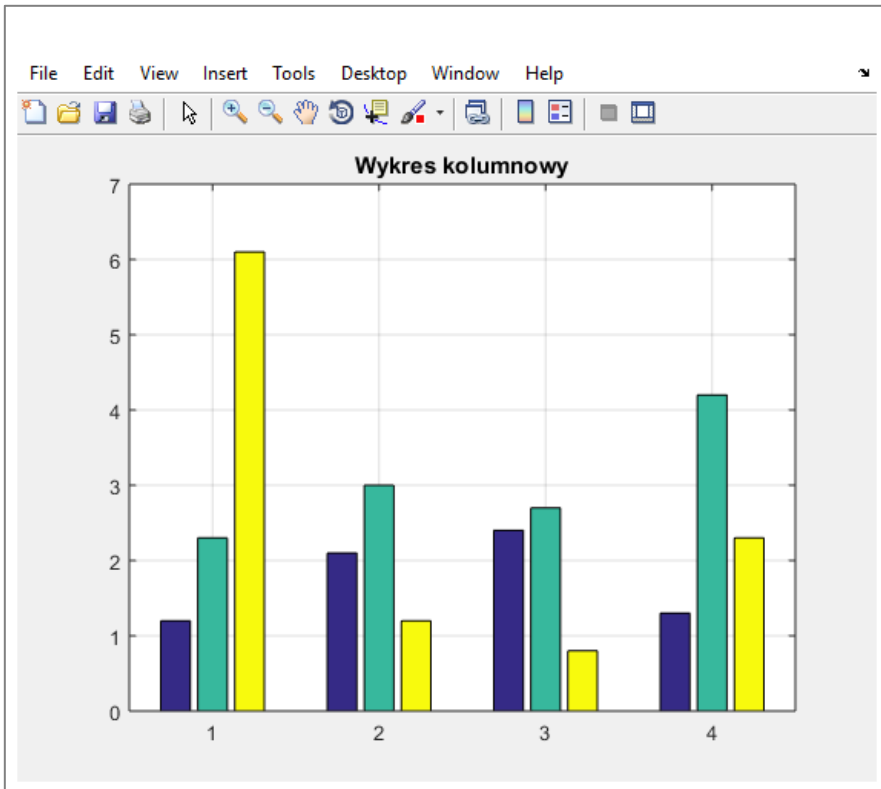
## Przykład

```
x=2010:2018;
y=[100 120 350 150 425 500 ...
   125 360 520];
bar(x,y,0.4,'r')
title('Wykres kolumnowy')
grid on
```



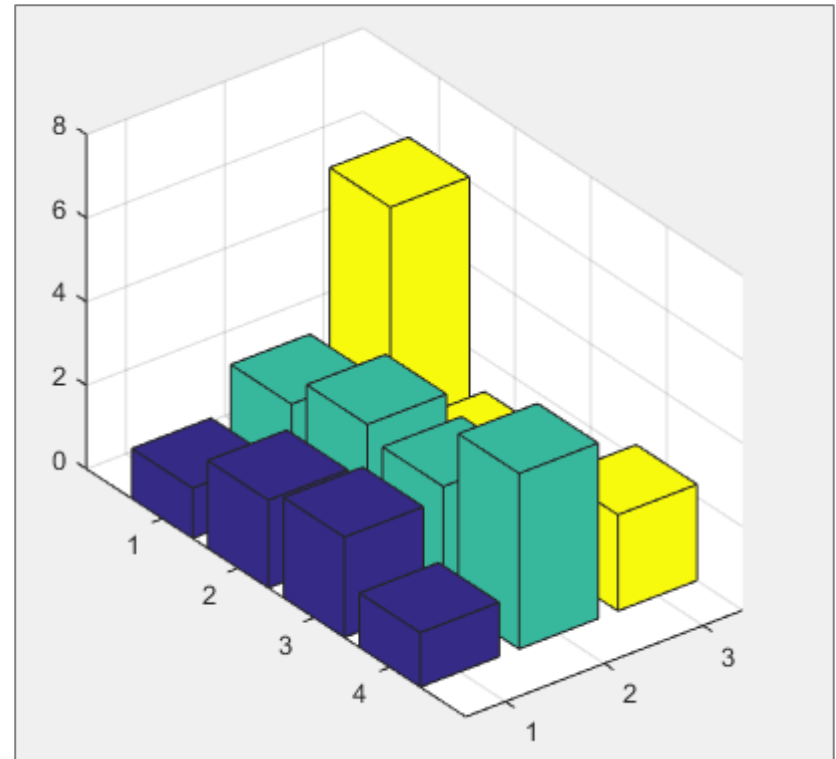
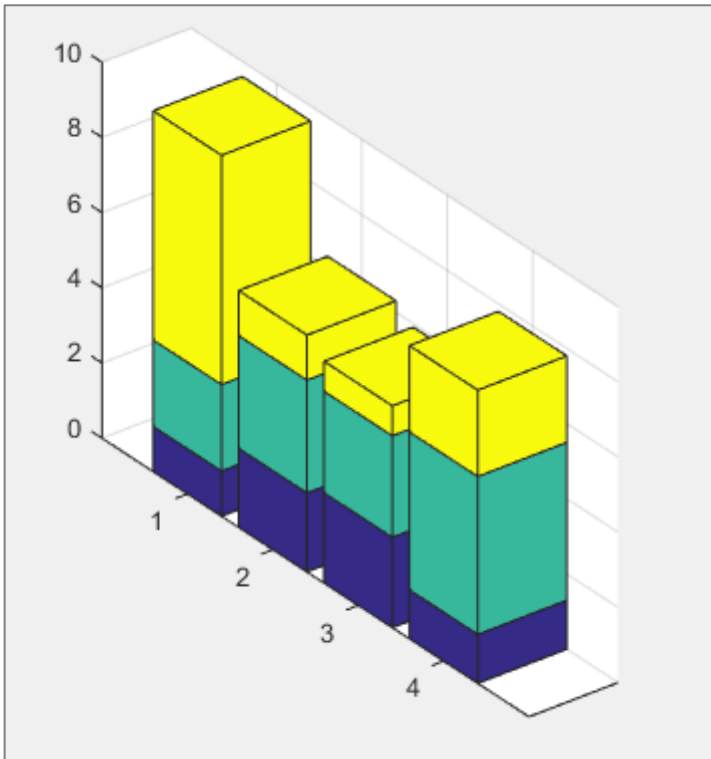
# Wykresy kolumnowe i słupkowe

```
y=[1.2 2.3 6.1;2.1 3.0 1.2; 2.4 2.7 0.8;1.3 4.2 2.3];  
figure; bar(y); title('Wykres kolumnowy'); grid on  
figure; barh(y); title('Wykres słupkowy'); grid on
```



# Wykresy kolumnowe i słupkowe

```
y=[1.2 2.3 6.1;2.1 3.0 1.2; 2.4 2.7 0.8;1.3 4.2 2.3];  
bar3(y,'stacked')  
figure  
bar3(y)
```

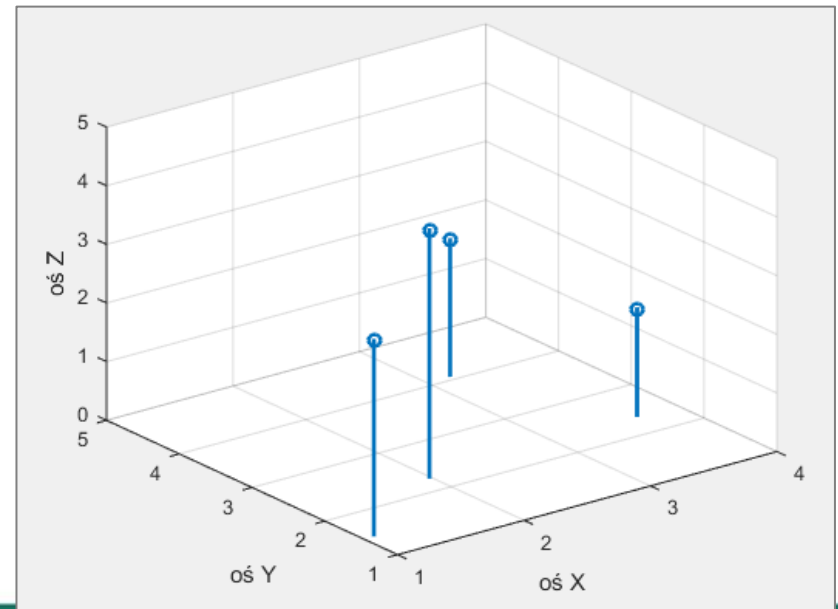


# Wykresy schodkowe i odcinkowe

- ❖ ***stairs(x, y, s)*** – rysuje schodki o wysokości kolejnych elementów wektora **y** w punktach określonych przez elementy wektora **x**; **s** określa wygląd rysowanej linii
- ❖ ***stem(x, y, 'fill', s)*** – na wykresie dwuwymiarowym rysuje odcinki o wysokości kolejnych elementów wektora **y** w punktach określonych przez elementy wektora **x**; **s** określa wygląd rysowanej linii
- ❖ ***stem3(x, y, z, 'fill', s)*** – na wykresie trójwymiarowym rysuje odcinki o wysokości kolejnych elementów wektora **z** w punktach określonych wektorami **x** i **y**; **s** określa wygląd rysowanej linii

## Przykład

```
x=[1.1 2.0 3.2 3.7];
y=[1.5 2.3 4.1 2.4];
z=[3.3 4.2 2.3 1.8];
stem3(x,y,z,'LineWidth',2)
xlabel('oś X')
ylabel('oś Y')
zlabel('oś Z')
```



# Wykresy schodkowe i odcinkowe

```
x=1:10;  
y=[0 2.2 6.5 3.5 4.1 6 1 2.5 3.4 7.1];  
subplot(1,2,1)  
stairs(x,y,'r','LineWidth',2)  
title('Wykres uzyskany za pomocą funkcji \itstairs');  
subplot(1,2,2)  
stem(x,y,'b','fill','LineWidth',1)  
title('Wykres uzyskany za pomocą funkcji \itstem');
```



# Wykresy kołowe

- ❖ `pie(x, etykiety)` – rysuje wykres kołowy wektora  $x$
- ❖ `pie3(x, etykiety)` – rysuje wykres kołowy z efektem trójwymiarowym
- ❖ `pie(x, r, etykiety)` – rysuje wykres kołowy wektora  $x$  w wyróżnionych wycinkach;  $r$  to zero-jedynkowy wektor tej samej długości co  $x$
- ❖ `pie3(x, r, etykiety)`

## Przykład

```
x = [807 3701 331 461 349];
kontynent = {'Afryka', 'Azja', 'Europa',
             'Ameryka Pn.', 'Ameryka Pd.'};
r = [0 0 1 0 0];
pie(x,r, kontynent)
colormap('hot')
```

