Politechnika Wrocławska

Wykresy. . .

Wojciech Myszka

10 grudnia 2024

HR EXCELLENCE IN RESEARCH

Politechnika Wrocławska

1. Dygresja
2. Wykresy
3. Co jeszcze warto wiedzieć?

Dygresja

Liczenie słów I

Taki wydumany problem:

Mamy tekst, musimy wyznaczyć częstość występowania poszczególnych słów w tekście

1. Skąd wziąć tekst?

   Na przykład z serwisu Wolne lektury. Wybieram wiersz K.I. Gałczyńskiego *Bal u Salomona* i zapisuję jako `bal.txt`.

2. Trzeba go wczytać do MATLABa

```
wiersz = string(fileread('bal.txt'));
```

Liczenie słów II

3. Trzeba go podzielić na słowa (to znaczy wywalić wszystkie znaki przestankowe; „wywalić” czyli zamienić na odstępy).

   Robi to poniższa funkcja:

```
function words = extractWords(text)
    punctuationCharacters = ["." "?" "!" "," ";" ":" "–" "*"
                                        ...")" "(" "..." "„" """];
    text = replace(text,punctuationCharacters," ");
    words = split(join(text));
    words(strlength(words)<5) = [];
end
```

   - najpierw wskazuję wszystkie znaki przestankowe
   - każdy znak przestankowy zamieniam na odstęp

Liczenie słów III

- gdyby się okazało, że tekst, to kilka kolumn — scalam go używając funkcji `join()`, a następnie
- dzielę na wyrazy funkcją split...

```
text = ["ala" "na kota"];
split(join(text))
ans =
  3×1 string array
    "ala"
    "na"
    "kota"
```

- kolejne polecenie może być trochę wątpliwe: odrzucam wszystkie słowa krótsze niż 5 liter; ale można tego nie robić.

Liczenie słów IV

4. Teraz najtrudniejsze jak policzyć słowa (nie chodzi o liczbę słów, tylko o liczbę wystąpień każdego słowa)?

   Wspominałem o zmiennych kategorycznych (*categorical variables*)

5. Tak więc dokonamy konwersji

   ```
   slowa = extractwords(wiersz);
   C = categorical(slowa);
   ```

   `C` jest obiektem typu kategorycznego

6. Wyodrębnimy teraz kategorie (**czyli różne słowa**)

   ```
   c = categories(C)
   ```

   i policzymy słowa z każdej kategorii

Liczenie słów V

```
cc = countcats(C)
min(cc)
ans = 1
max(cc)
ans = 22
```

(powyższe przy założeniu, że biorę pod uwagę słowa dłuższe niż 5 znaków).

7. Które słowo występuje najczęściej?

```
[a b] = max(cc);
```

a zawiera wartość maksimum, a b numer elementu tablicy z maksimum

Liczenie słów VI

```
c(b)
ans = 1×1 cell array
    {'jeszcze'}
```

8. Kolejnym interesującym problemem może być posortowanie słów od najczęściej do najrzadziej występującego. To jest trochę zagmatwane.. Jest taka funkcja `sort()`, która potrafi tablicę posortować.

   Wywołana tak:

   ```
   Y = sort(X);
   ```

   posortowaną tablicę zapisuje do `Y` (od najmniejszego do największego); gdy dodać parametr `'descend'` — w kolejności odwrotnej,; natomiast użyta tak:

   ```
   [Y I] = sort(X,`descend`);
   ```

Liczenie słów VII

W tablicy `I` zapisany będzie oryginalny porządek (czyli coś typu *Teraz jest pierwsza, ale przedtem była na setnym miejscu w tablicy).

9. Można te dane wykorzystać do przearanżowania tablicy kategorii

```
[bb,neworder] = sort(cc,'descend');
CC = reordercats(C,neworder);
c = categories(CC)
ans = 1653×1 cell
'jeszcze'
'tylko'
'przez'
'wszystko'
'świat'
'potem'
```

Liczenie słów VIII

```
'które'
'muzyka'
'takie'
'wtedy'
```

10. A tak, poza tym jest funkcja która robi ładny wykres obrazujący częstość występowania słów

```
wc = wordcloud(C);
title("Bal u Salomona")
```

Liczenie słów IX

Politechnika Wrocławska

a na koniec... I

...można to jeszcze ładnie pokolorować

```
numWords = height(c);        % liczba słów
colors = rand(numWords,3); % generujemy losowo kolory
wc = wordcloud(c,cc,'Color',colors); % dodajemy kolory
```

(c zawiera słowa, cc częstości i dodajemy kolory)

Politechnika Wrocławska

a na koniec... II

Wykresy

Figure I

Z różnych względów warto używać polecenia `figure()` — pozwala to zmieniać właściwości obrazków i pozwala równocześnie operować wieloma wykresami.

Polecenie `figure()` tworzy nowe okno (używając domyślnych właściwości). Okno staje się bieżącym wykresem.

Polecenie może być użyte w następujące sposoby:

- `figure`
- `figure(Name,Value)`

  Definiujemy właściwości okna na zasadzie *nazwa* i *wartość*

  ```
  figure('Name', "Ala ma kota")
  ```

Figure II

`f = figure(___)`

`f` to „uchwyt” (*handle*, wskaźniki) okna

- `figure(f)`

  „przywołuje” okno związane ze zmienną `f`

- `figure(n)`

  „przywołuje” okno o numerze `n` (Figure *n*:)

Przywołanie okna powoduje, że powinno ono „wyjść na wierzch”

Politechnika Wrocławska

gcf

Funkcja `gcf` zwraca „uchwyt” bieżącego okna

Mając „uchwyt” można modyfikować różne parametry okna

```
f = gcf;
f.Name = "Bardzo ważny wykres"
```

Mając uchwyt można zawartość okna zapisać do pliku

Politechnika Wrocławska

saveas I

```
f = gcf;
% albo
f = figure
saveas(f,'filename')
```

Funkcja `saveas()` rozpoznaje typ pliku i zapisuje zawartość w odpowiednim formacie

- `.fig` specjalny plik zawierający wszystkie informacje o oknie i jego zawartości pozwalający na odtworzenie
- `.m` tworzy plik `.fig` oraz plik `.m` zawierający funkcję ładująca plik `.fig`.
- `.jpg`, `.png`, `.eps`, `.pdf`, `.tif`, `.svg` — tworzy plik graficzny z zawartością okna

saveas II

(Odradzam korzystanie z plików typu `.jpg`)

Jeżeli mamy drukarkę może się udać wydrukowanie na papierze poleceniem `print()`

Politechnika Wrocławska

„Liniowe”

`plot`	2-D
`plot3`	3-D
`stairs`	Schody
`errorbar`	Liniowy zaznaczonymi błędami
`area`	Wypełnione
`stackedplot`	kilka różnych wykresów o wspólnej osi x

Logarytmiccznee

`loglog`	Log-log
`semilogx`	Pół-logarytmiczny (x)
`semilogy`	Pół-logarytmiczny (y)

Funkcji

`fplot`	2-D
`fimplicit`	Funkcja określona równaniem
`fplot3`	3-D (parametryczna)

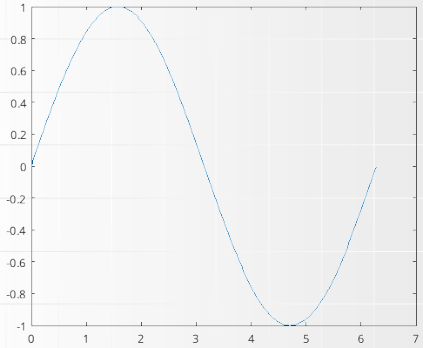
Parametry (właściwości) wykresu I

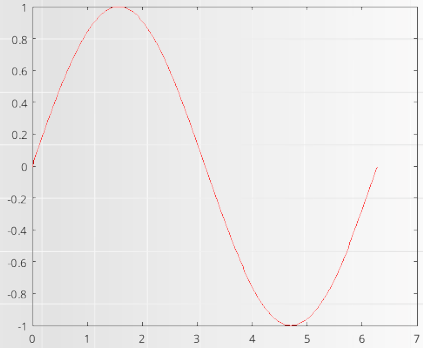
```
>> x=0:pi/100:2*pi;
>> y=sin(x);
>> p = plot(x,y)
p =
  Line with properties:
              Color: [0 0.4470 0.7410]
          LineStyle: '-'
          LineWidth: 0.5000
             Marker: 'none'
         MarkerSize: 6
    MarkerFaceColor: 'none'
              XData: [0 0.0314 0.0628 0.0942 0.1257 0.1571 0.1885
              YData: [0 0.0314 0.0628 0.0941 0.1253 0.1564 0.1874
  Show all properties
```

Parametry (właściwości) wykresu II

p jest strukturą opisującą dokładnie każdy wykres. Można ustalać wartości parametrów podczas wywołania funkcji lub po — modyfikując wartości struktury, na przykład

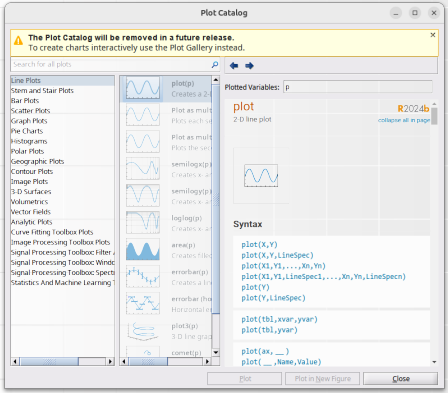
```
p.Color = 'red';
```





Plot catalog

Klikając prawym klawiszem myszy na zmienną związaną z wykresem (`p` w naszym przypadku) można tam znaleźć na samym dole menu `Plot catalog`. Wybranie tej pozycji otwiera okienko zawierające dokumentację wszystkich funkcji. Niestety funkcjonalność ta zniknie w przyszłych wydaniach MATLABa



Politechnika Wrocławska

Plot Gallery

Już teraz jest pozycja w głównym menu nazwana **PLOTS**. Pozwala ona interaktywnie tworzyć wykresy

- Zaznaczamy w przestrzeni roboczej zmienną (albo zmienne) i podpowiada jakie wykresy można z nich stworzyć.

Politechnika Wrocławska

Rozkład danych

`histogram`	Histogram
`histogram2`	Histogram dwuwymiarowy
`morebins`	Zwiększenie liczby przedziałów
`fewerbins`	Zmniejszenie liczby przedziałów
`histcounts`	Dzieli dane na przedziały i podaje lizcbę próbek, które wpadły
`histcounts2`	To samo, tylko dwie zmienne
`boxchart`	Wykres pudełkowy
`violinplot`	Wykres „skrzypcowy”
`swarmchart`	Bardzo podobny do skrzypcowego - rozkład zmienności
`swarmchart3`	jak wyżej, ale dwa parametry

Wykresy bąbelkowe

`bubblechart`	Bubble chart *(Since R2020b)*
`bubblechart3`	3-D bubble chart *(Since R2020b)*
`bubblelim`	Map bubble sizes to data range *(Since R2020b)*
`bubblesize`	Set minimum and maximum bubble sizes in points *(Since R2020b)*
`bubblelegend`	Create legend for bubble chart *(Since R2020b)*

Pozwalają na pokazanie dodatkowej zmiennej: promień i/lub kolor bąbelka

Wykresy rozproszenia

`scatter`	Scatter plot
`scatter3`	3-D scatter plot
`binscatter`	Binned scatter plot
`scatterhistogram`	Create scatter plot with histograms
`spy`	Visualize sparsity pattern of matrix
`plotmatrix`	Scatter plot matrix
`parallelplot`	Create parallel coordinates plot

Gdy punkty nie układają się na linii

Politechnika Wrocławska

Wykresy typu kołowego

`donutchart`	Donut chart *(Since R2023b)*
`piechart`	Pie chart *(Since R2023b)*
`bubblecloud`	Create bubble cloud chart *(Since R2021a)*
`wordcloud`	Create word cloud chart from text data
`heatmap`	Create heatmap chart
`sortx`	Sort elements in heatmap row
`sorty`	Sort elements in heatmap column
`pie`	Legacy pie chart
`pie3`	3-D pie chart

Politechnika Wrocławska

Wykresy słupkowe

`bar`	Bar graph
`barh`	Horizontal bar graph
`bar3`	3-D bar graph
`bar3h`	Horizontal 3-D bar graph
`pareto`	Pareto chart

Politechnika Wrocławska

Wykres łodygowy

(*Stem plot*)

`stem`	Plot discrete sequence data
`stem3`	Plot 3-D discrete sequence data

Taki patyczek z bąbelkiem na końcu

Politechnika Wrocławska

Wykresy danych geograficznych

`geoplot`	Plot line in geographic coordinates
`geoscatter`	Scatter chart in geographic coordinates
`geobubble`	Standalone geographic bubble chart
`geodensityplot`	Density plot in geographic coordinates
`geoiconchart`	Icon chart in geographic coordinates *(Since R2024b)*

Wykresy biegunowe

Zamiast zmiennych kartezjańskich (x, y) używamy zmiennych biegunowych (θ, r)

`polarplot`	Plot line in polar coordinates
`polarscatter`	Scatter chart in polar coordinates
`polarbubblechart`	Polar bubble chart *(Since R2020b)*
`polarhistogram`	Histogram chart in polar coordinates
`fpolarplot`	Plot expression or function in polar coordinates *(Since R2024a)*
`compassplot`	Polar plot with arrows emanating from origin *(Since R2024b)*

Poziomice

`contour`	Contour plot of matrix
`contourf`	Filled 2-D contour plot
`contourc`	Low-level contour matrix computation
`contour3`	3-D contour plot
`contourslice`	Draw contours in volume slice planes
`clabel`	Label contour plot elevation
`fcontour`	Plot contours

contour() I

Użycie tej funkcji wymaga specjalnego przygotowania danych: wygenerowania prostokątnej siatki, w której rogach mamy podane wartości.

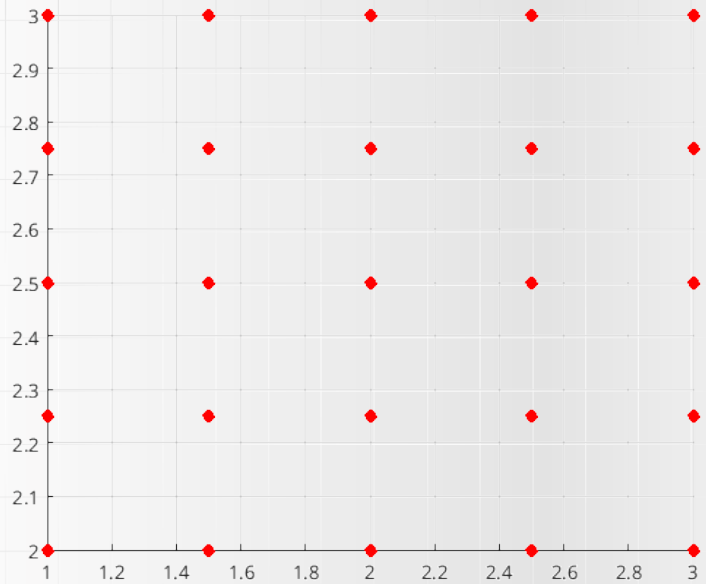
```
x = linspace(-2*pi,2*pi);
y = linspace(0,4*pi);
[X,Y] = meshgrid(x,y);
Z = sin(X)+cos(Y);
contour(X,Y,Z)
```

Służy do tego funkcja `meshgrid()`; funkcja `linspace()` generuje n (domyślnie 100) wartości z zadanego przedziału

Siatka wygląda tak:

Politechnika Wrocławska

contour() II



Politechnika Wrocławska

Pola wektorowe

`quiver`	Quiver or vector plot
`quiver3`	3-D quiver or vector plot
`compassplot`	Polar plot with arrows emanating from origin *(Since R2024b)*
`feather`	Arrows from x-axis

i pola przepływu

`streamline`	Plot streamlines from 2-D or 3-D vector data
`streamslice`	Plot streamlines in slice planes

Wykresy powierzchni I

`surf`	Surface plot
`surfc`	Contour plot under surface plot
`surface`	Primitive surface plot
`surfl`	Surface plot with colormap-based lighting
`surfnorm`	Surface normals
`mesh`	Mesh surface plot
`meshc`	Contour plot under mesh surface plot
`meshz`	Mesh surface plot with curtain
`fsurf`	Plot 3-D surface
`fmesh`	Plot 3-D mesh
`fimplicit3`	Plot 3-D implicit function

Politechnika Wrocławska

Wykresy powierzchni II

`pcolor`	Pseudocolor plot
`waterfall`	Waterfall plot
`ribbon`	Ribbon plot
`contour3`	3-D contour plot

surf() I

Podobnie, jak w przypadku poziomic, dane muszą być przygotowane na siatce

Ale wykorzystamy tu gotową funkcję `peaks()`, która generuje gotowy wykres 3D

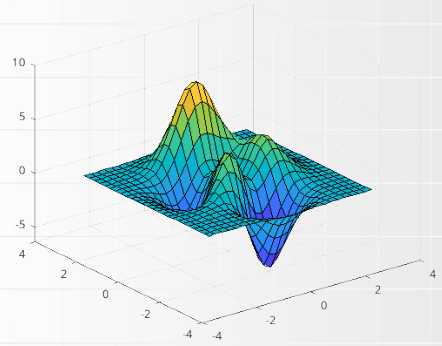
```
>> [X Y Z] = peaks(25);
>> surf(X, Y, Z)
>> s=surf(X, Y, Z)
s =
  Surface with properties:
        EdgeColor: [0 0 0]
        LineStyle: '-'
        FaceColor: 'flat'
    FaceLighting: 'flat'
        FaceAlpha: 1
```

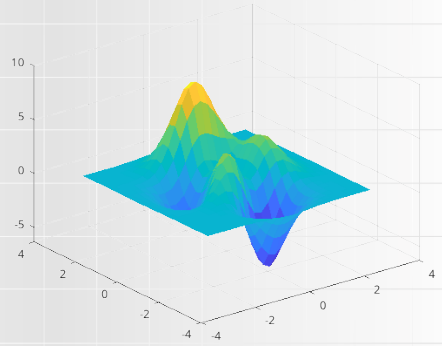
Politechnika Wrocławska

surf() II

```
XData: [25×25 double]
YData: [25×25 double]
ZData: [25×25 double]
CData: [25×25 double]
```

Show all properties

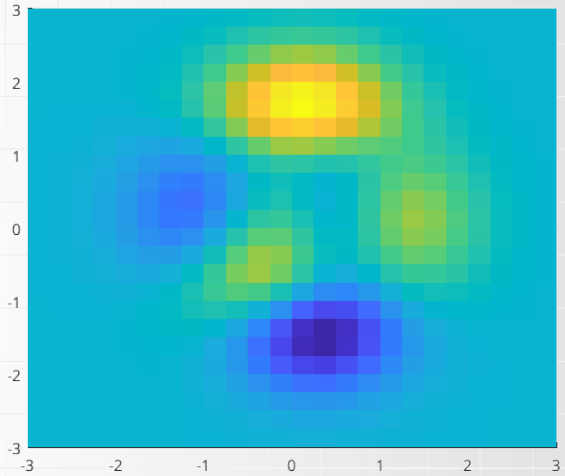




Politechnika Wrocławska

Mapa

Patrząc z góry na ten wykres dostajemy coś w rodzaju mapy



Dwie osie I

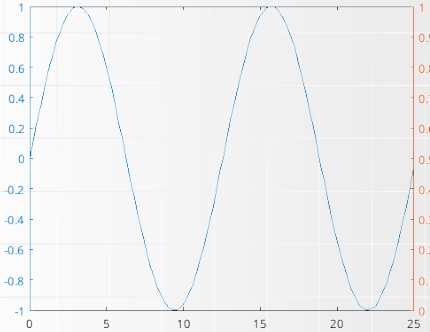
Czasami gdy trzeba na jednym wykresie przedstawić dwa przebiegi o zupełnie innym zakresie korzystamy z „dwu" osi y o innym zakresie po lewej i prawej stronie

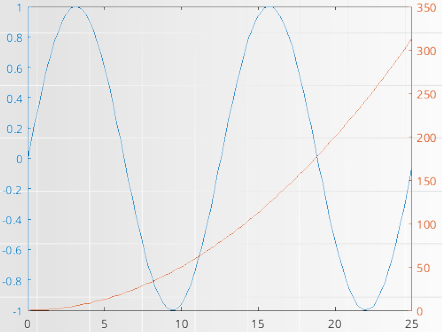
```
x = linspace(0,25);
y = sin(x/2);
yyaxis left
plot(x,y);
r = x.^2/2;
yyaxis right
plot(x,r);
```

Efekt będzie taki

Politechnika Wrocławska

Dwie osie II

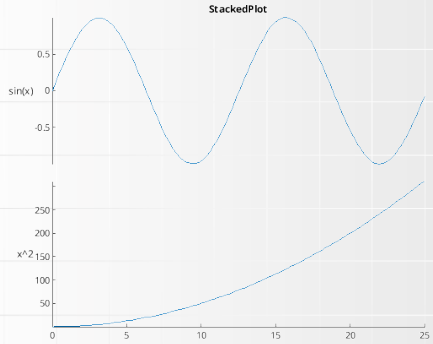




stackedplot()

Alterntywą dla dwu osi jest funkcja stackedplot

```
x = linspace(0,25);
y = sin(x/2);
r = x.^2/2;
stackedplot(x',[y ;r]','DisplayLabels',[{'sin(x)'};{'x^2'}],'Titl
```



Opisy wykresów I

1. Tytuł

```
title(titletext,subtitletext)
```

gdy korzystamy z `tiledlayout` można dodać tytuł do każdej części:

```
tiledlayout(2,1)
ax1 = nexttile;
plot(ax1,(1:10).^2)
ax2 = nexttile;
plot(ax2,(1:10).^3)
title(ax1,'Top Plot')
title(ax2,'Bottom Plot')
```

Opisy wykresów II

2. Opis osi

   Funkcje: `xlabel()`, `ylabel()` i `zlabel()`

3. Dowolny tekst na wykresie

   Funkcja `text()`

```
x = 0:pi/20:2*pi;
y = sin(x);
plot(x,y)
text(pi,0,'\leftarrow sin(\pi)')
```

Współrzędne są w jednostkach wykresu!

Opisy wykresów III

4. Annotacja

Funkcja tworzy linię (lub strzałkę albo jakiś obiekt) między wskazanymi punktami i dodaje do niej tekst;

Dostępne obiekty

- `line`
- `arrow`
- `doublearrow`
- `textarrow`
- `rectangle`
- `ellipse`
- `texbox`

Politechnika Wrocławska

Opisy wykresów IV

5. Legenda

   Funkcja `legend()` pozwala dodać opisy pozwalające identyfikować krzywe na wykresie (gdy jest ich więcej)

   ```
   legend(label1,...,labelN)
   ```

Co jeszcze warto wiedzieć?

Politechnika Wrocławska

properties I

Polecenia

- `plot` i wszystkie używane do tworzenia wykresów
- `figure`
- `title`
- legend
- `text`
- `annottion`
- i wiele, wiele innych

mogą być użyte w formacie

```
p = polecenie(...)
```

Politechnika Wrocławska

properties II

Zmienna `p` jest złożoną strukturą zawierającą **właściwości** obiektu.

Po zamknięciu okna, którego dotyczą własności zmienna staje się niedostępna

Polecenie `inspect` otwiera aplikację *Property inspector* pozwalającą na odczyt i modyfikację różnych właśiwości wykresu.