

Podstawy analizy danych

gnuplot

Wykład 2

Janusz Andrzejewski

Plan

- Polecenia pomocnicze
- Obsługa plików
 - Opcja using
 - Pseudokolumna
 - Opcja skip
- Upiększanie wykresu
 - Ustalanie znaczników na osiach
 - Ustalanie nazw osi
 - Wykres „po polsku”
- Modyfikacje danych

Polecenia pomocnicze

- Wyświetlenie bieżącego katalogu

`pwd`

- Zmiana bieżącego katalogu

`cd ".." # do katalogu wyżej`

`cd 'c:\student' # przejdź do katalogu student
na dysku głównym C`

- Dla systemu Windows powinno używać się pojedynczego znaku cudzysłowu ! Jeśli chcemy podwójny cudzysłów to wówczas trzeba napisać:

`cd "c:\\student" #cd "c:\student" nie zadziała!!`

- Pomimo używania polskiej wersji językowej MS Windows, nazwy katalogów trzeba podawać po angielsku np. przejście do katalogu Pulpit to (jeśli jesteśmy zalogowani jako student)

`cd 'C:\Users\Student\Desktop'`

- Wywołanie konsoli systemowej (tzw. shell)

`shell #powrót do gnuplota exit`

Pliki

gnuplot może odczytać dane zarówno z plików tekstowych jak i binarnych. W dalszej części wykładu będziemy używać tylko plików tekstowych

- Pliki tekstowe składają się z kolumn, kolumny oddzielone są od siebie tzw. białymi znakami
- W skład pojedynczej kolumny wchodzi liczba (całkowita lub rzeczywista) albo ciąg znaków zawartych w cudzysłowie
- Linie rozpoczynające się od znaku # są uważane za linie komentarzu i są ignorowane.
- Znak oddzielający kolumny i znak od którego zaczyna się komentarz można dowolnie definiować

Plik tekstowy

Fragment pliku akcje.dat:

```
#akcje.dat
```

```
# Srednia cena akcji w dolarach za jedna akcje na rok
```

```
#ROK ABC XYZ
```

```
1975 49 1.6e3
```

```
1976 52 1.4e3
```

Układ danych w pliku jest następujący:

- wartość x jest w kolumnie 1
- wartość y jest w kolumnie 2
- dodatkowo wartości y odpowiadające każdej wartości x, są umieszczane w kolejnych kolumnach.
- nie zawsze tak być musi :) ...

Opcja using

```
plot 'akcje.dat'
```

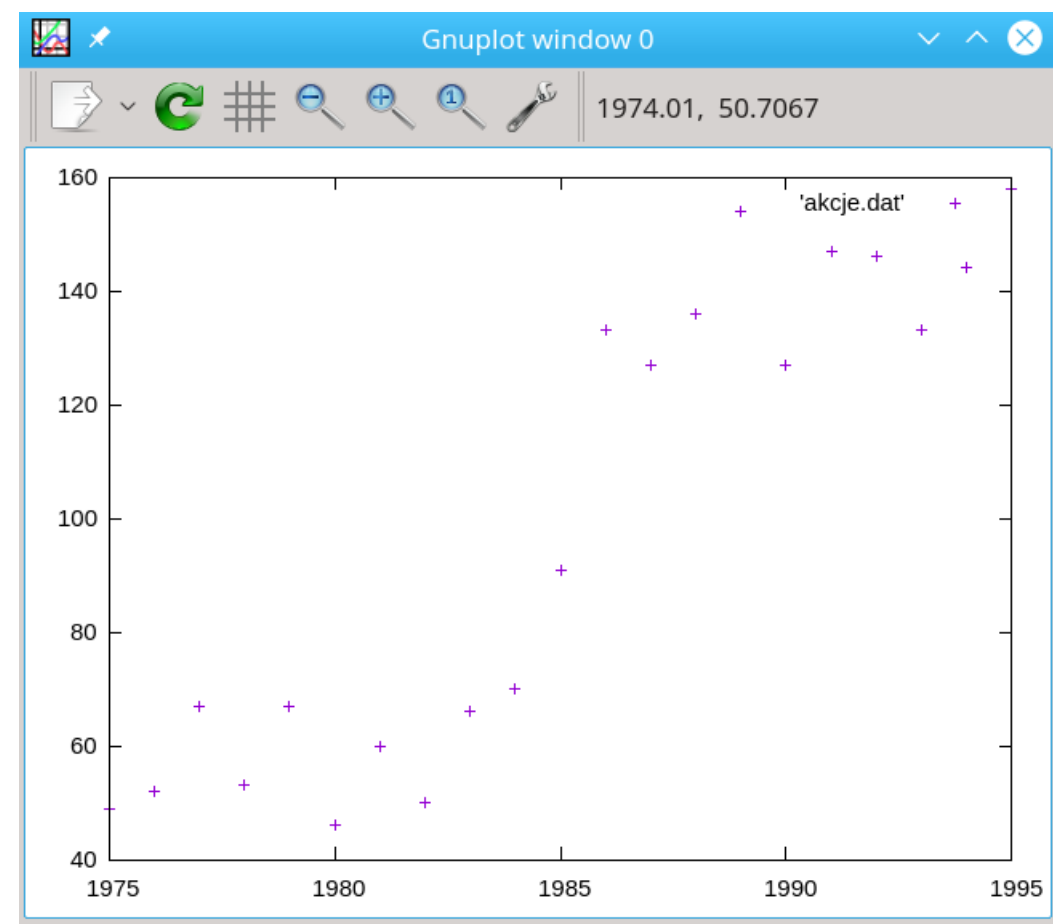
Komenda ta wykona tzw.
wykres punktowy -
zamieszczony obok.

Komenda:

```
plot 'akcje.dat' using 1:2 with lines, \  
'akcje.dat' using 1:3 with linespoints
```

Spowoduje narysowanie dwu zależności – pierwszej w postaci linii dla danych z kolumny 2 versus kolumna 1, drugiej w postaci linii z punktami dla kolumny 3 versus kolumna 1.

Opcja **using** służy do manipulowania danych w kolumnach



Opcja using

Komendę:

```
plot 'akcje.dat' using 1:2 with lines, \  
'akcje.dat' using 1:3 with linespoints
```

Można zastąpić przez:

```
p 'akcje.dat' u 1:2 w l, ' ' u 1:3 w lp
```

lub przez

```
p "akcje.dat" u 1:2 w l, "" u 1:3 w lp
```

Gdzie zamiast podawać drugi raz nazwę pliku użyto tzw. domyślnego pliku czyli poprzednio użytego pliku.

```
p 'akcje.dat' u 1:2 w l t 'ABC', ' ' u 1:3 w lp t 'XYZ'  
# ' ' - są to 2 pojedyncze cudzysłowy
```

Gdzie opcja **t** jest skrótem od **title**

PAMIETAJ !!! Opcja **title** występuje zawsze po opcji **using**. Ponadto można zamiennie używać pojedynczego lub podwójnego cudzysłowowa

Pseudokolumna, opcja skip

Nie zawsze x musi być w 1 kolumnie :

```
plot 'akcje.dat' using 2:3 title 'ABC vs XYZ' with lines
```

Można zrobić wykres kolumny 1 względem $x=0, 1, 2, \dots$ - czyli względem numeru wiersza

```
plot 'akcje.dat' u 0:1 title 'Kolejne lata'
```

Kolumna o numerze zero jest tzw. pseudokolumną i jest to numer linii z danymi w pliku z pominięciem linii będących komentarzem oraz linii pominiętych poprzez opcję `skip`

```
p 'akcje.dat' u 1:2 w l t 'ABC', '' u 1:3 skip 10 w lp t 'XYZ'
```

Gdzie dla drugiej zależności zostanie pominiętych pierwsze 10 linii z pliku

Dwie osie Y

```
plot 'akcje.dat' using 1:2 with lines, \  
' ' using 1:3 axis x1y2 with lines
```

Komenda ta spowoduje narysowanie 2 zależności, pierwszej używając kolumn 1 i 2 (opcja **using 1:2**) dla osi X na dole i osi Y z lewej strony (opcja **axis x1y1**), drugiej zależności używając kolumn 1 i 3 (**using 1:3**) dla osi X na dole i osi Y z prawej strony (**axis x1y2**)

Komenda **set** z różnymi opcjami zmienia domyślne ustawienia gnuplota.

```
set ytics nomirror      #Wylacza duplikowanie etykiet z osi Y1 na oś Y2  
set y2tics              #Wlacza etykiety na osi Y2  
set ylabel 'Cena akcji ABC [$]'      #ustawia opis osi Y1  
set y2label 'Cena akcji XYZ [Pln]'   #ustawia opis osi Y2  
set xlabel 'Lata'                  #ustawia opis osi X1  
p 'akcje.dat' u 1:2 ax x1y1 w l t 'Akcje ABC', \  
' ' u 1:3 ax x1y2 w l t 'Akcje XYZ'
```

Jeśli chcemy ustawić opis osi X2 to używamy **set x2label "opis osi x2"**

Jeśli do powyższego wykresu chcemy dodać jeszcze tytuł wykresu to można użyć komendy **replot** :

```
set title "Wykres cen akcji ABC i XYZ"  
replot
```

Upiększanie wykresu

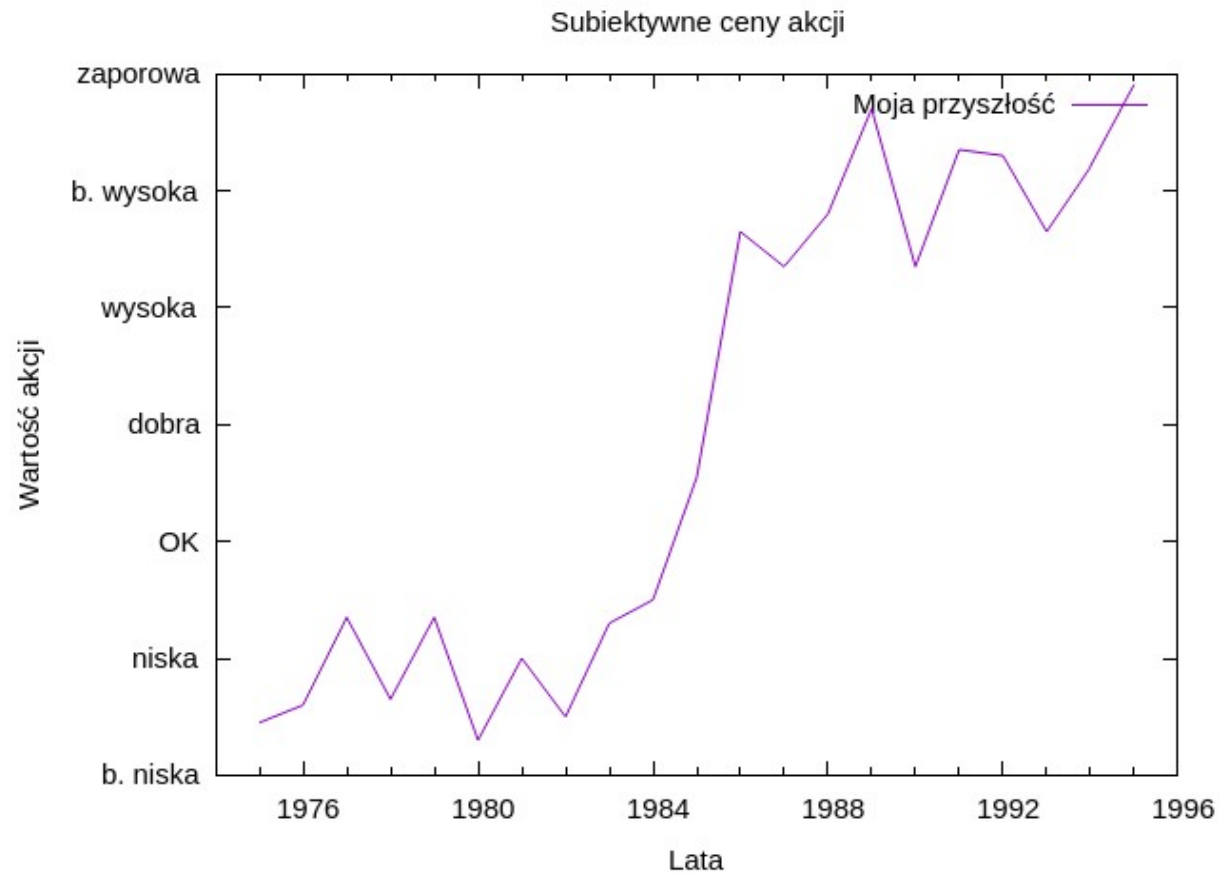
Do ustalania podziału i etykiet na osiach (tzw. główne znaczniki) służy opcja **_tics** komendy set, gdzie za **_** można podstawić x, y, x2, y2. Symbol x – oznacza dolną oś poziomą, x2 – górną oś poziomą, y – lewą oś pionową a y2 prawa os pionową. Komenda ta posiada argumenty (tylko wybrane):

- in/out - znaczniki są skierowane do wewnątrz/zewnątrz np. **set xtics out**
- liczba – oznacza inkrement pomiędzy znacznikami np.. **set x2tics 0.3**
- liczba1, liczba2 - znaczniki zaczynają się od liczba1 z inkrementem liczba2
- liczba1, liczba2, liczba3 – znaczniki zaczynają się od liczba1 z inkrementem liczba2 i są wyświetlane do liczba3 np. **set ytics 10, 1, 50**
- set ytics (opis liczba, ...) - ustala własny opis znacznika dla wartości liczba np.: **set ytics ("niska" 60, "OK" 80, "dobra" 100, \ "wysoka" 120, "b. wysoka" 140)**

Do ustalenia znaczników pobocznych służy opcja **m_tics** z argumentem oznaczającym ilość podprzedziałów np. **set mxtics 3**

Przykład

```
set title "Subiektywne ceny akcji"  
set ytics ("b. niska" 40, "niska" 60, "OK" 80, "dobra" 100,  
  "wysoka" 120, "b. wysoka" 140, "zaporowa" 160)  
set xtics 4  
set mxtics 4  
set ylabel "Wartość akcji"  
set xlabel "Lata"  
P [1974:1996][40:160] \  
  "akcje.dat" \  
  w l t "Moja przyszłość"
```



Po polsku

W przypadku gdy nie działają polskie litery można ustawić kodowanie znaków na UNICODE:

```
set encoding utf8
```

W Polsce a także w wielu krajach Europy, na oddzielenie części całkowitej od ułamkowej używa się przecinka a nie kropki. Można to zmienić komendą:

```
set decimalsign ', ' #w cudzysłowie  
#jest przecinek
```

Modyfikacje danych

Opcja **using** służy także do modyfikacji danych zawartych w pliku.

- Jeśli pierwszy lub drugi argument dyrektywy **using** jest ujęty w nawiasy, to nie jest on traktowany jako numer kolumny w pliku z danymi, ale jako wartość wyrażenia.
- Wewnątrz wspomnianych nawiasów, można uzyskać dostęp do wartości danej kolumny poprzez poprzedzenie numeru kolumny znakiem dolara (\$).
- Wyrażenia w nawiasach mogą być stałymi.

```
plot 'akcje.dat' using 1:($2*2)
```

```
plot 'akcje.dat' using 1:(sqrt($2)) with lines
```

```
plot 'akcje.dat' using ($1+35):2 w l \  
t 'Prognozowana cena akcji'
```

Przykład

```
a=30*120*4.4 #Ilosc akcji ABC * cena 1 akcji w $ *  
              razy cena $
```

```
b=10*30.3     #Ilosc akcji XYZ * cena 1 akcji w Pln
```

```
set ylabel 'Wartość portfela [Pln]'
```

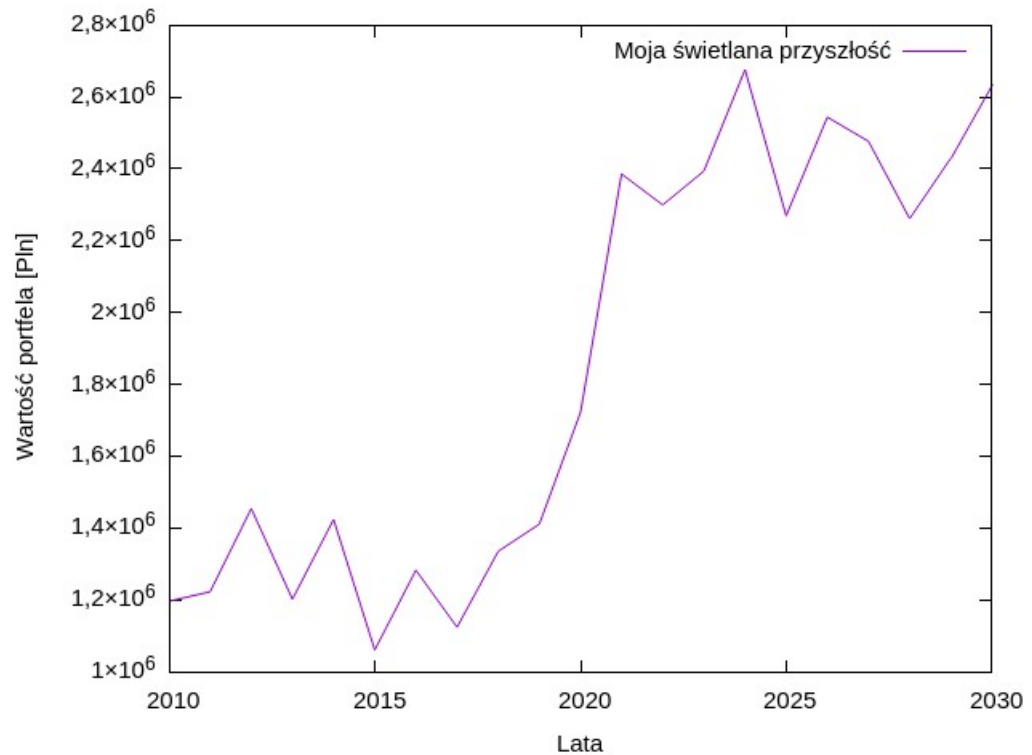
```
set decimalsign ','
```

```
set xlabel 'Lata'
```

```
p 'akcje.dat' \
```

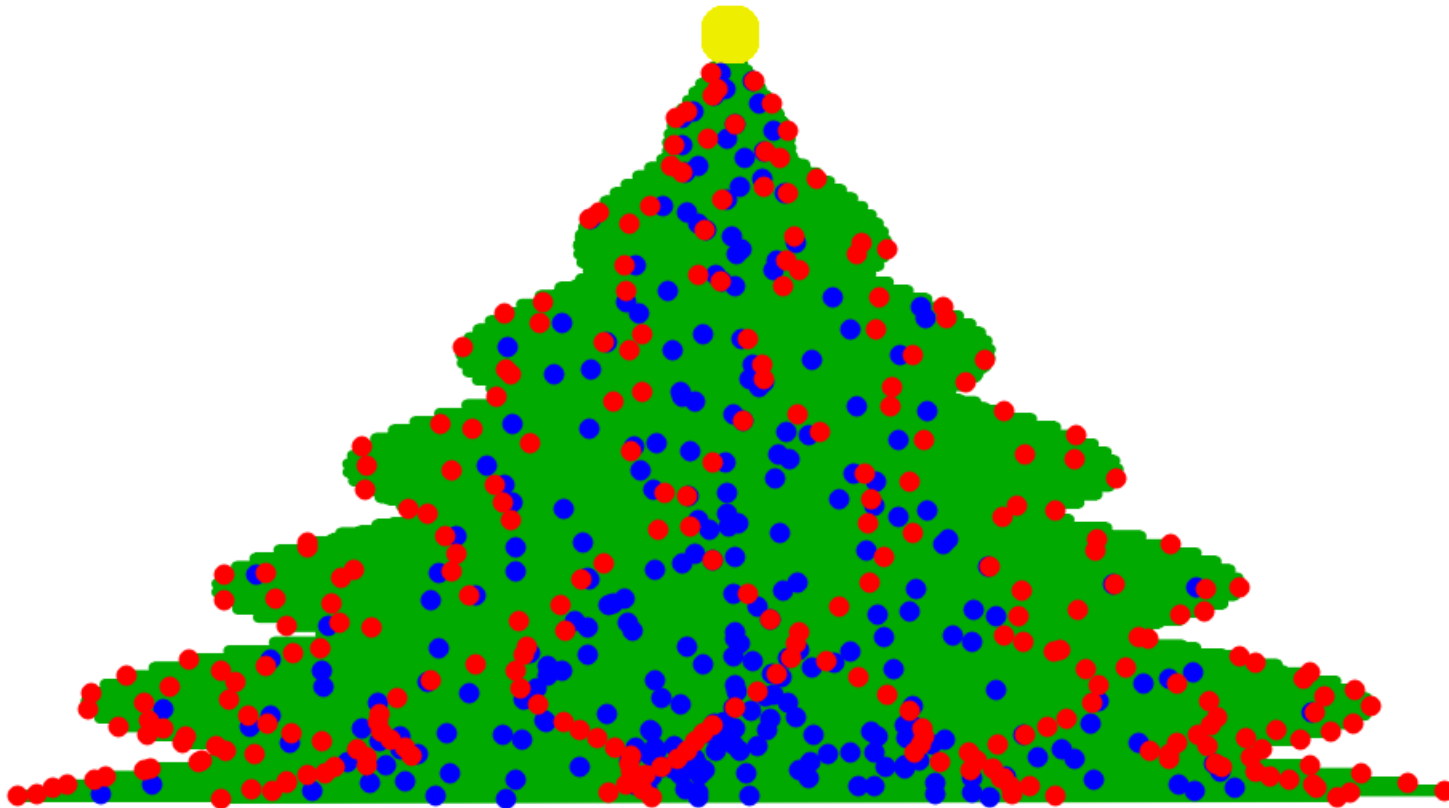
```
u ($1+35):($2*a+$3*b) \
```

```
w l t 'Moja świetlana przyszłość'
```



„Made by gnuplot”

Wesołych Świąt !!!



PS. Zachęcam do utworzenia własnej choinki przy pomocy gnuplot'a
PPS. Skrypt pojawi się na kolejnych zajęciach :)