

Wizualizacja danych - Gnuplot

dr hab. Bożena Woźna-Szcześniak

Akademia im. Jan Długosza

bwozna@gmail.com

Laboratorium 3



O czym dziś będzie mowa

- Kreślenia nieposortowanych plików danych
- Wygładzanie zaburzonych danych
- Matematyka z gnuplotem
- Zmienne i funkcje definiowane przez użytkownika
- Proste przekształcenia danych
- Pseudokolumny

Kreślenia nieposortowanych plików danych

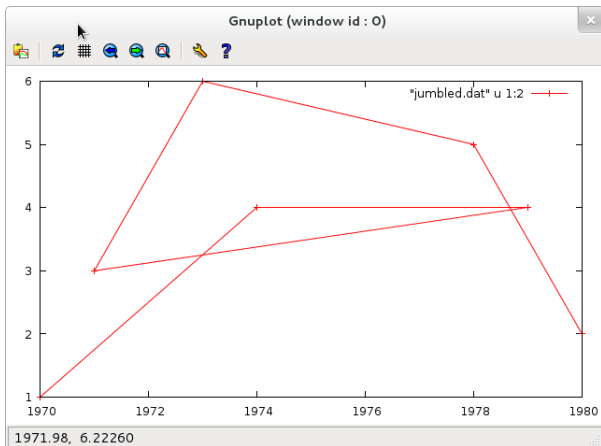
Założmy, że mamy taki plik z nieposortowanymi danymi

```
1970 1
1974 4
1979 4
1971 3      # out of order!
1973 6
1978 5
1980 2
```

Kreślenia nieposortowanych plików danych

Wykres technika znana nam do tej pory:

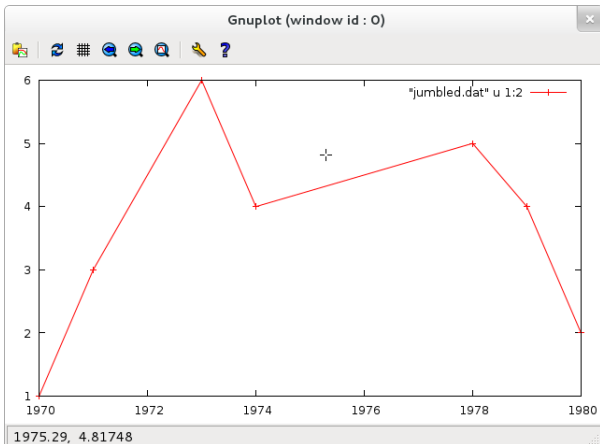
```
gnuplot> plot "jumbled.dat" u 1:2 w lp
```



Kreślenia nieposortowanych plików danych

Wykres z opcją `smooth unique`:

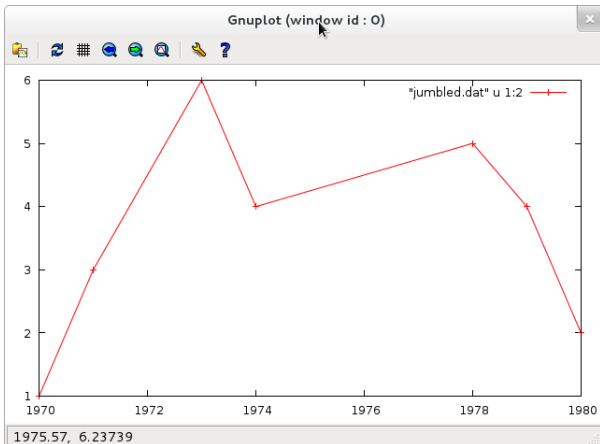
```
gnuplot> plot "jumbled.dat" u 1:2  
smooth unique w lp
```



Kreślenia nieposortowanych plików danych

Wykres z opcją `smooth frequency`:

```
gnuplot> plot "jumbled.dat" u 1:2  
smooth frequency w lp
```



Kreślenia nieposortowanych plików danych

- Dyrektywa `smooth` komendy `plot` jest prostym narzędziem do kreślenia wykresów na podstawie nieuporządkowanych plików danych.
- Dyrektywa `smooth` pobiera jeden z następujących parametrów: `unique`, `frequency`, `bezier`, `sbezier`, `csplines`, `acsplines`.
- Dyrektywa `unique` sortuje wartości wybrane przez dyrektywę `using` dla osi X.
- Jeśli dowolna wartość x występuje więcej niż jeden raz, to zostanie ona zastąpiona przez jeden punkt danych z średnia z odpowiadających wartości y .
- Dyrektywa `frequency` działa w ten sam sposób, ale jeśli dowolna wartość x występuje więcej niż jeden raz, to zostanie ona zastąpiona przez jeden punkt danych, który stanowi sumę wszystkich wartości y dla każdej wartości x .

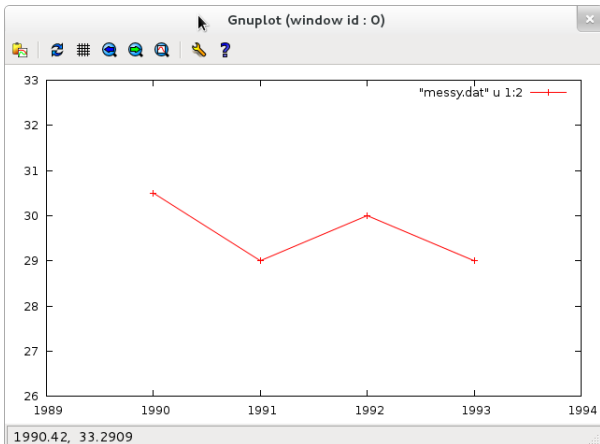
Kreślenia nieposortowanych plików danych

```
# Temperatura dla trzech miast.  
# Kody miast (kolumna 3):  
#Warszawa (=1), Katowice (=2), Czestochowa (=3)  
# Format: Rok Temperatura Miasto  
# ... Warszawa, Katowice  
1990 32 1  
1990 29 2  
1991 33 1  
1991 27 2  
1992 31 1  
1992 29 2  
1993 32 1  
1993 26 2  
# ... Czestochowa  
1991 27 3  
1993 29 3
```


Kreślenia nieposortowanych plików danych

Wykres z opcja `smooth unique`:

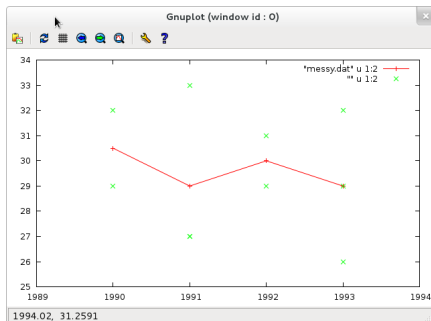
```
gnuplot> plot [1989:1994] "messy.dat" u 1:2  
smooth unique w lp
```



Kreślenia nieposortowanych plików danych

Wykres z opcją `smooth unique` i bez niej:

```
gnuplot> plot [1989:1994][25:34] "messy.dat" u 1:2  
smooth unique w lp, "" u 1:2 w p
```



Powyższa komenda kreśli zarówno oryginalne dane punktów (niepołączone), jaki i trend średniej temperatury.

Wyglądanie zaburzonych danych

- Pozostałe rodzaje dyrektywy `smooth` generują wykresy z wygładzonych danych, tzn. aby wygenerować wykres zastępuje się dane oryginalne danymi z opisu matematycznego.
 - `bezier` - oblicza i wykreśla przybliżenia Béziera rzędu n (gdzie n jest liczbą danych). Przybliżenie gwarantuje, że krzywa przechodzi przez punkt reprezentujący pierwszą i ostatnie daną oryginalną, natomiast nie gwarantuje że przechodzi przez każdy z wewnętrznych oryginalnych punktów. Powstała krzywa jest zazwyczaj bardzo gładka.
 - `sbezier` - najpierw stosuje nietypową transformację do danych, a następnie oblicza i wykreśla aproksymację Béziera dla przekształconych wyników.

Wyglądanie zaburzonych danych

- `csplines` - stosuje transformację `unique`, a następnie oblicza wartość pewnej funkcji *spline* trzeciego rzędu (będącą liczbą naturalną) z wyników transformacji i kreśli tak wygładzone dane. Powstała krzywa przechodzi przez dokładnie wszystkich dane. W przeciwieństwie do krzywej Béziera (która definiowana jest dla całego zbioru danych), funkcje *splines* definiowane są lokalnie (pomiędzy dwoma kolejnymi punktami danych) i łączone płynnie w tych punktach.
- `acsplines` - stosuje transformację `unique` do oryginalnych danych, a następnie oblicza ważoną przybliżoną funkcję *splines* trzeciego rzędu. Waga jest obowiązkowym trzecim argumentem dyrektywy `using`.

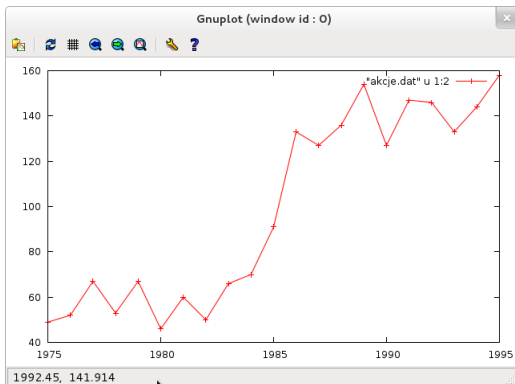
Wyglądanie zaburzonych danych

```
#akcje.dat
# Srednia cena akcji w dolarach za jedna akcje na rok
#ROK ABC XYZ
1975 49 162
1976 52 144
1977 67 140
1978 53 122
1979 67 125
1980 46 117
1981 60 116
1982 50 113
1983 66 96
1984 70 101
1985 91 93
1986 133 92
1987 127 95
1988 136 79
1989 154 78
1990 127 85
1991 147 71
1992 146 54
1993 133 51
1994 144 49
1995 158 43
```

Wyglądanie zaburzonych danych

Wykres danych oryginalnych bez wygładzenia

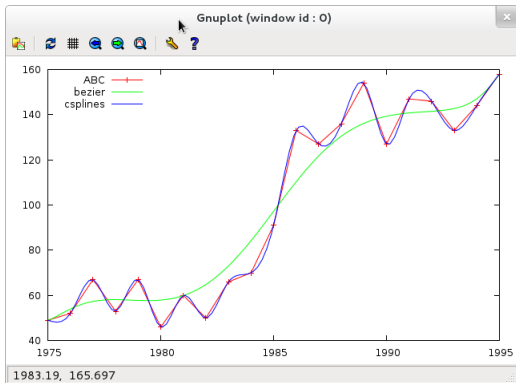
```
gnuplot> plot "akcje.dat" u 1:2 w lp
```



Wyglądanie zaburzonych danych

Wykres bez i z wygładzeniem

```
gnuplot> set key left top  
gnuplot> plot "akcje.dat" u 1:2 title "ABC" w lp,  
"" u 1:2 title "bezier" smooth bezier,  
"" u 1:2 title "csplines" smooth csplines
```



Wyglądanie zaburzonych danych

- Ważona aproksymacja `spline` (czyli `smooth acsplines`) modyfikuje lokalny charakter aproksymacji `spline`.
- Nie wymaga się już w niej, aby krzywa interpolacyjna przechodziła przez każdy punkt reprezentujący daną w rozważanym zbiorze. To czy punkt jest brany pod uwagę zależy od jego wagi.
- Waga jest pobierana z obowiązkowej trzeciej kolumny w specyfikacji dyrektywy `using` i może być albo taka sama dla wszystkich punktów z zestawu danych lub zmieniać się w zależności od punktu.

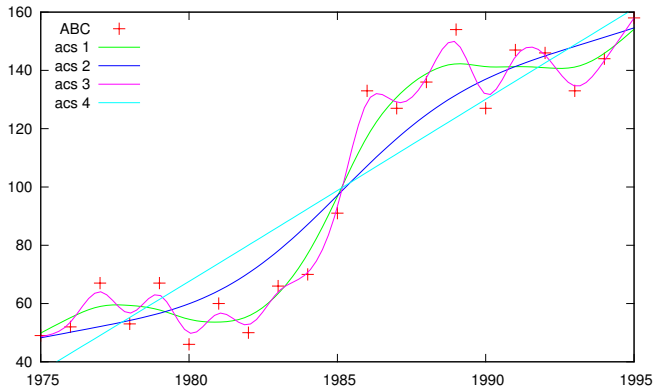
Wygładzanie zaburzonych danych

```
plot [1975:1995][40:160] "akcje.dat"  
  using 1:2 title "ABC" with points,  
  "" using 1:2:(1) title "acs 1" smooth  
  acsplines with lines,  
  "" using 1:2:(1/50.) title "acs 2" smooth  
  acsplines with lines,  
  "" using 1:2:(50) title "acs 3" smooth  
  acsplines with lines,  
  "" using 1:2:(1/10000.) title "acs 4" smooth  
  acsplines with lines
```

inny sposób

```
plot [1975:1995][40:160] "akcje.dat" u 1:2 t "ABC"  
  "" u 1:2:(1) t "acs 1" smooth acsplines w l,\  
  "" u 1:2:(1/50.) t "acs 2" smooth acsplines w l,\  
  "" u 1:2:(50) t "acs 3" smooth acsplines w l,\  
  "" u 1:2:(1/10000.) t "acs 4" smooth acsplines w l
```

Wyglądanie zaburzonych danych

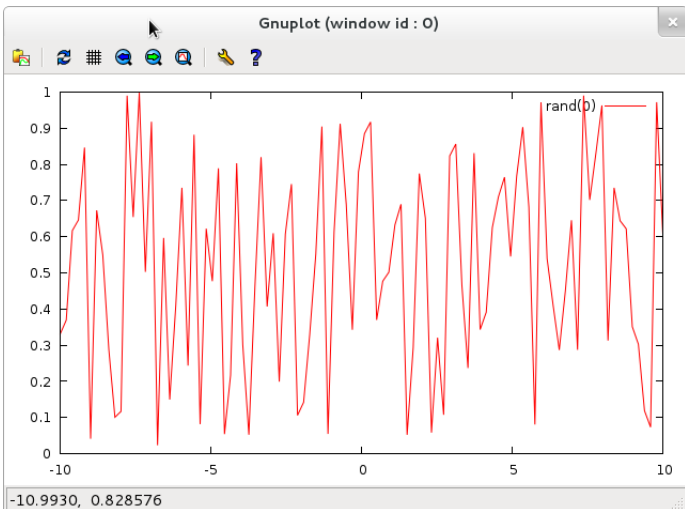


Wygładzanie zaburzonych danych

- W ogólności im większa jest waga punktu danych, tym bardziej przyczynia się ten punkt do wyglądu krzywej interpolacyjnej.
- Jeśli waga punktów danych dąży w granicy do nieskończoności, to aproksymacja `smooth acsplines` działa tak jak algorytm `smooth csplines`.
- Jeśli waga punktów danych dąży w granicy do zera, to indywidualny wkład każdego punktu zmniejsza się i krzywa staje się gładzsza.
- Jeśli waga punktów w granicy jest zero, to krzywa przekształca się w funkcję liniową.

Wbudowane funkcje matematyczne

```
gnuplot> plot rand(0)
```



Wbudowane funkcje matematyczne

- Funkcja `rand` wywołana z argumentem równym zero zwraca pseudo losową liczbę z zakresu od zera do jeden.

```
gnuplot> print rand(0)  
0.222190514600859
```

- Algorytm do generowania liczb pseudolosowych wymaga ustawienia dwóch *zarodków ciągu*.
- Te same wartości dla zarodków, generują dokładnie taki sam ciąg liczb pseudolosowych.
- Gnuplot nie posiada oddzielnej funkcji do ustawienia zarodków; uzyskuje się je poprzez dostarczenie specjalnych argumentów do funkcji `rand`.
- Komendę `print rand(-1)` stosuje się do ustawiania zarodków na wartości domyślne.

Wbudowane funkcje matematyczne

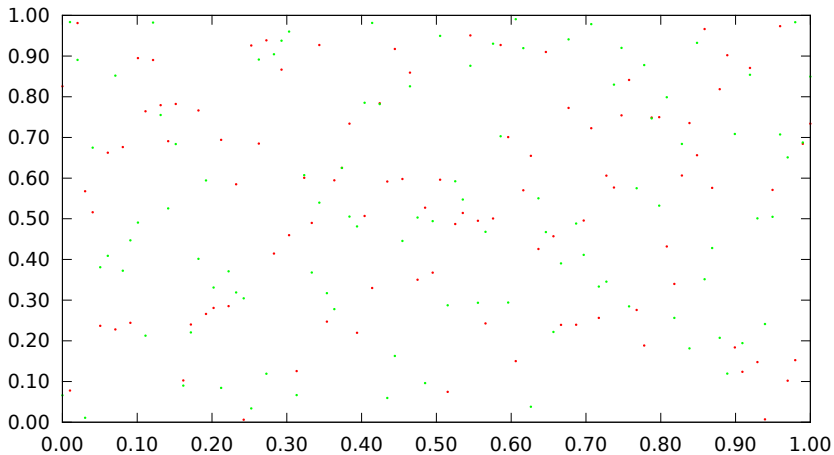
Wywołanie	Znaczenie
<code>rand(0)</code>	Zwraca kolejną pseudolosową liczbę, przy użyciu bieżących wartości obu ziaren
<code>rand(-1)</code>	Ustawia obydwa ziarna na wartości domyślne
<code>rand(x)</code>	Dla $x > 0$, ustawia obydwa ziarna na wartość x
<code>rand({x, y})</code>	Dla $x, y > 0$, ustawia pierwsze ziarno na x a drugie na y

Zabawa z random()

```
#Blokuję wyświetlanie legendy
unset key
#Ustawia format liczbowy opisu dla osi
set format x "%3.2f"
set format y "%3.2f"
#Tytuł wykresu
set title "Liczby pseudolosowe"
#Ustawia wartosci na osiach
set xtics 0.1
set ytics 0.1
set terminal pdf
set output "random.pdf"
plot [0:1][0:1] rand(0) with dots,\
      rand(0) with dots
```

Zabawa z random()

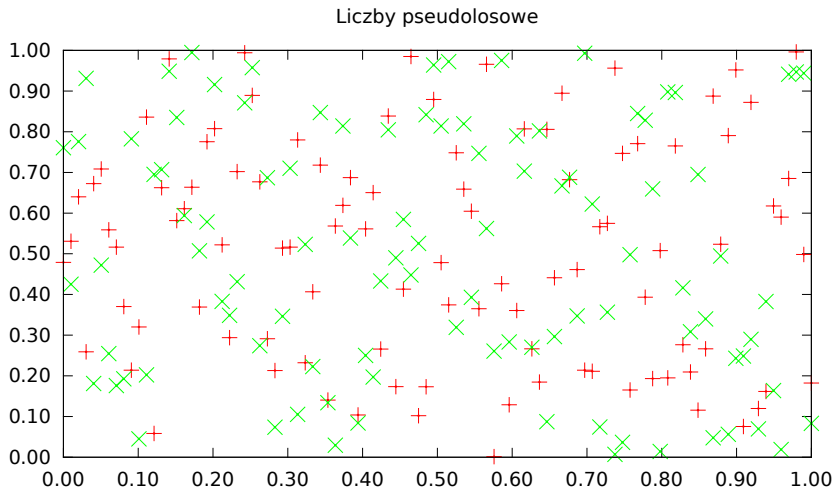
Liczby pseudolosowe



Zabawa z random()

```
unset key
set format x "%3.2f"
set format y "%3.2f"
set title "Liczby pseudolosowe"
set xtics 0.1
set ytics 0.1
set terminal pdf
set output "random-plus-krzyzyk.pdf"
set xrange [0:1]
set yrange [0:1]
set style function points
plot rand(0) pt 1, rand(0) pt 2
```

Zabawa z random()

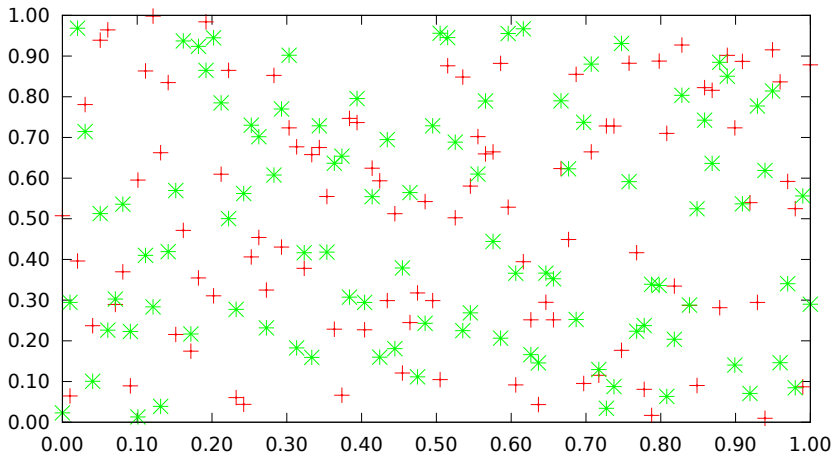


Zabawa z random()

```
unset key
set format x "%3.2f"
set format y "%3.2f"
set title "Liczby pseudolosowe"
set xtics 0.1
set ytics 0.1
set terminal pdf
set xrange [0:1]
set yrange [0:1]
set output "random-plus-gwiazdka.pdf"
plot rand(0) pt 1, rand(0) pt 3
```

Zabawa z random()

Liczby pseudolosowe

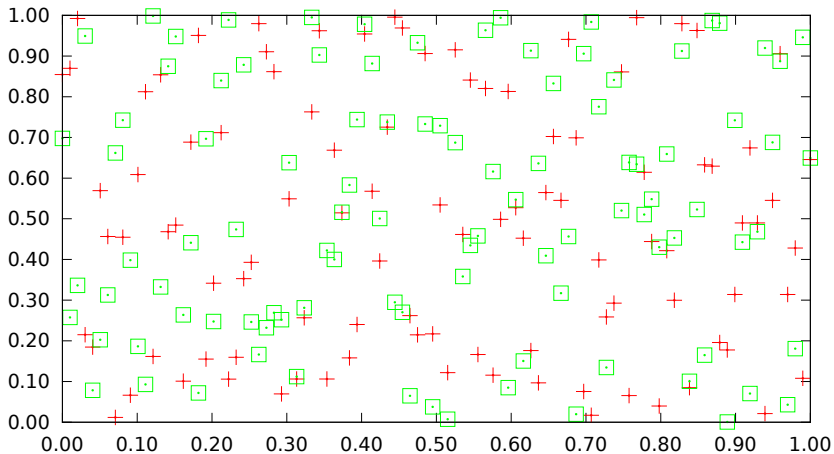


Zabawa z random()

```
unset key
set format x "%3.2f"
set format y "%3.2f"
set title "Liczby pseudolosowe"
set xtics 0.1
set ytics 0.1
set terminal pdf
set xrange [0:1]
set yrange [0:1]
set output "random-plus-kwadrat.pdf"
plot rand(0) pt 1, rand(0) pt 4
```

Zabawa z random()

Liczby pseudolosowe

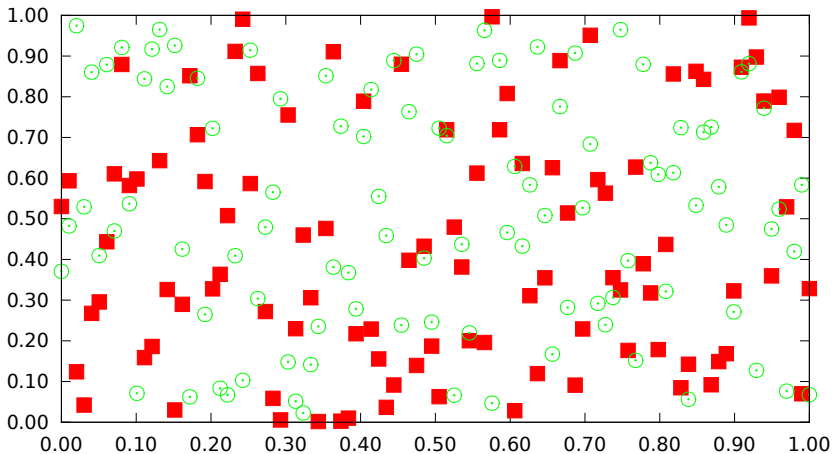


Zabawa z random()

```
unset key
set format x "%3.2f"
set format y "%3.2f"
set title "Liczby pseudolosowe"
set xtics 0.1
set ytics 0.1
set terminal pdf
set xrange [0:1]
set yrange [0:1]
set output "random-wypelnionykwadrat-okrag.pdf"
plot rand(0) pt 5, rand(0) pt 6
```

Zabawa z random()

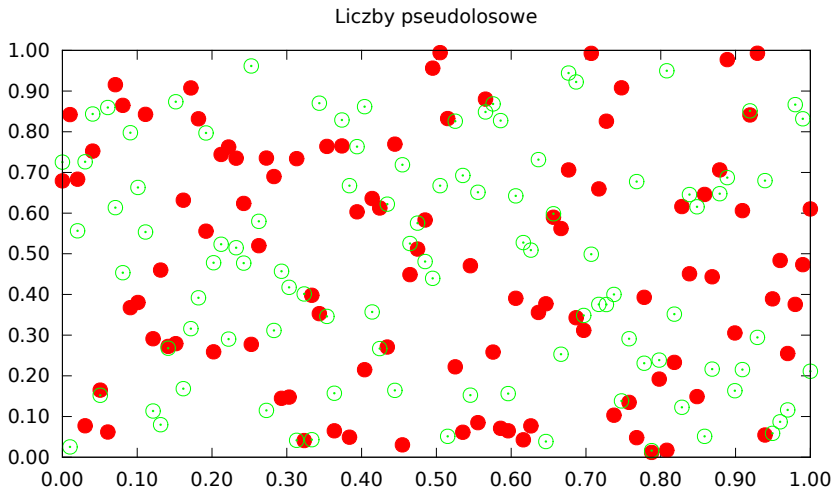
Liczby pseudolosowe



Zabawa z random()

```
unset key
set format x "%3.2f"
set format y "%3.2f"
set title "Liczby pseudolosowe"
set xtics 0.1
set ytics 0.1
set terminal pdf
set xrange [0:1]
set yrange [0:1]
set output "random-kolo-okrag.pdf"
plot rand(0) pt 7, rand(0) pt 6
```

Zabawa z random()

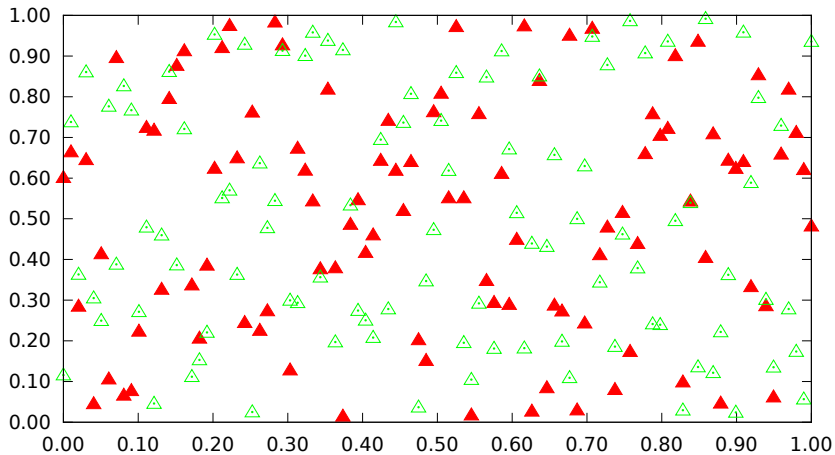


Zabawa z random()

```
unset key
set format x "%3.2f"
set format y "%3.2f"
set title "Liczby pseudolosowe"
set xtics 0.1
set ytics 0.1
set terminal pdf
set xrange [0:1]
set yrange [0:1]
set output "random-wypelnionytrojkat-trojkat.pdf"
plot rand(0) pt 9, rand(0) pt 8
```

Zabawa z random()

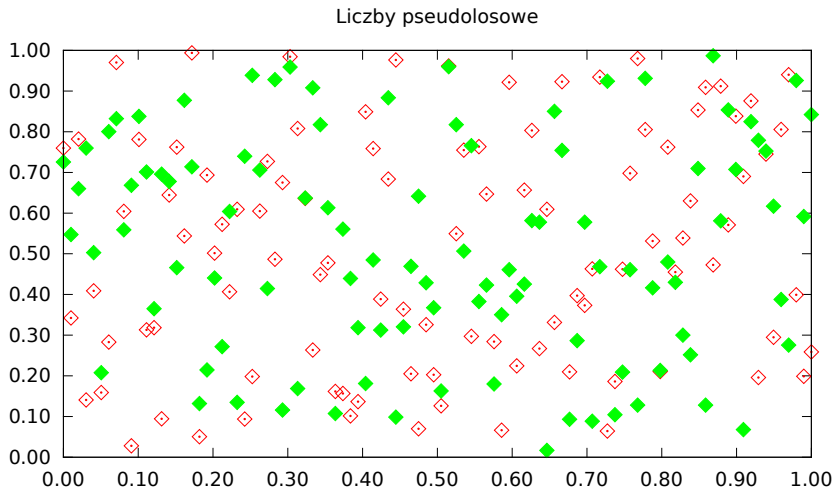
Liczby pseudolosowe



Zabawa z random()

```
unset key
set format x "%3.2f"
set format y "%3.2f"
set title "Liczby pseudolosowe"
set xtics 0.1
set ytics 0.1
set terminal pdf
set xrange [0:1]
set yrange [0:1]
set output "random-diamenty.pdf"
plot rand(0) pt 12, rand(0) pt 13
```

Zabawa z random()



Zmienne i funkcje definiowane przez użytkownika

Przykład definiowania zmiennych

```
# Podstawa logarytmu naturalnego
gnuplot> e = 2.71828182845905
gnuplot> print e
2.71828182845905
```

```
#pierwiastek kwadratowy z 2
gnuplot> sqrt2 = sqrt(2)
gnuplot> print sqrt2
1.4142135623731
```

```
# Stała Euler-Mascheroni
gnuplot> euler = 0.57721566490153
gnuplot> print euler
0.57721566490153
```

Zmienne i funkcje definiowane przez użytkownika

Przykład zmiennej już zdefiniowanej

```
# Stalej PI nie trzeba definiowac
gnuplot> print pi
3.14159265358979
```


Zmienne i funkcje definiowane przez użytkownika

- Funkcje mogą mieć do 10 zmiennych i mogą zawierać inne funkcje i operatory.

$$f(x) = -x * \log(x)$$

$$\min(a, b) = (a < b) ? a : b$$

$$\text{binom}(n, k) = n! / (k! * (n-k)!)$$

$$\text{step}(x) = (x < 0) ? 0 : 1$$

Zmienne i funkcje definiowane przez użytkownika

- Gnuplot domyślnie zakłada, że zmienna niezależna, tzw. dummy, oznaczona jest przez x .
- Można to zmienić za pomocą polecenia: `set dummy t`; teraz t przejmuje rolę zmiennej niezależnej x .

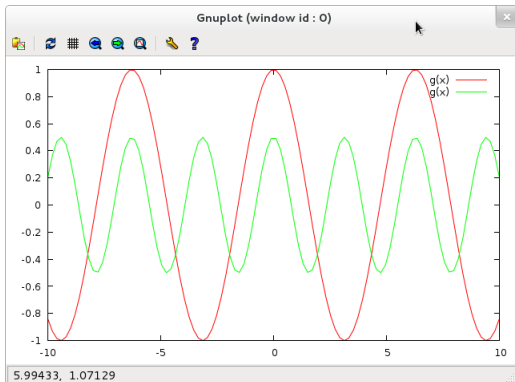
```
gnuplot> plot [-2*pi:2*pi] sin(x)
#zmiana zmiennej x na t
gnuplot> set dummy t
gnuplot> plot [-2*pi:2*pi] sin(t)
```

Zmienne i funkcje definiowane przez użytkownika

- Wszystkie inne zmienne występujące w definicji funkcji (parametry) muszą mieć przypisane wartości przed próbą ich kreślenie.

```
gnuplot> g(x) = cos(a*x) / a
```

```
gnuplot> plot a=1, g(x), a=2, g(x)
```



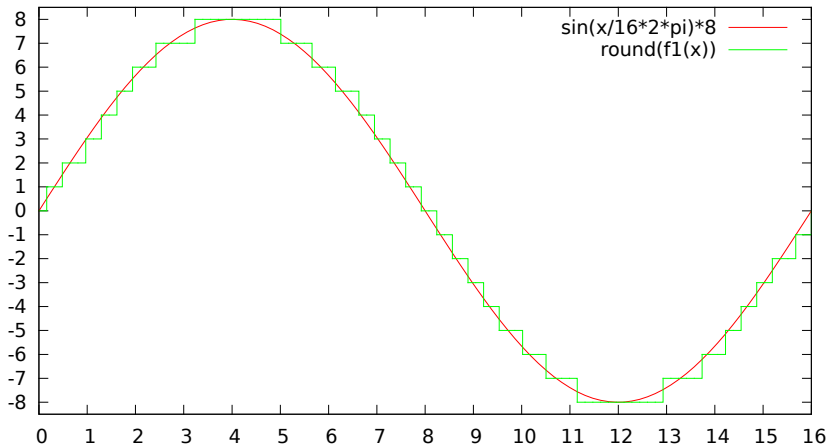
Zabawa z funkcjami

```
set encoding utf8
set title "Funkcje"
set xtic 1
set ytic 1
f1(x) = sin(x/16*2*pi)*8
round(x) = x - floor(x) < 0.5 ? floor(x) : ceil(x)

set terminal pdf
set output "funkcje.pdf"
plot [0:16] [-8.5:8.5] f1(x)
title "sin(x/16*2*pi)*8" with lines,
round(f1(x)) with steps
```

Zabawa z funkcjami

Funkcje



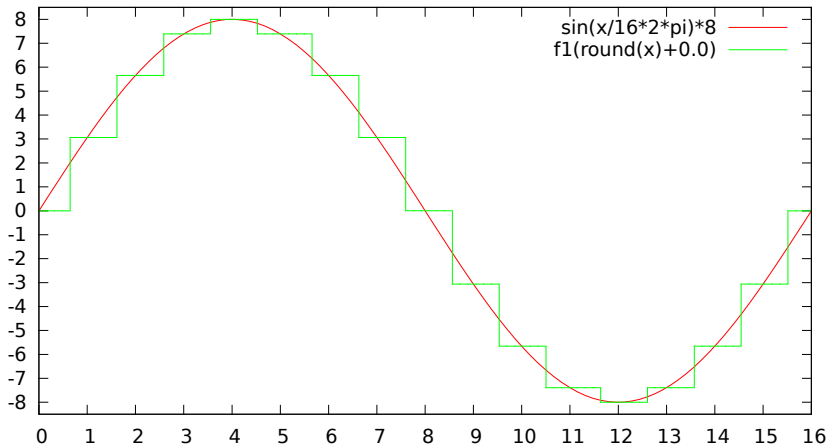
Zabawa z funkcjami

```
set encoding utf8
set title "Funkcje"
set xtic 1
set ytic 1
f1(x) = sin(x/16*2*pi)*8
round(x) = x - floor(x) < 0.5 ? floor(x) : ceil(x)

set terminal pdf
set output "funkcje2.pdf"
plot [0:16] [-8.5:8.5] f1(x)
title "sin(x/16*2*pi)*8" with lines,
f1(round(x)+0.0) with steps
```

Zabawa z funkcjami

Funkcje

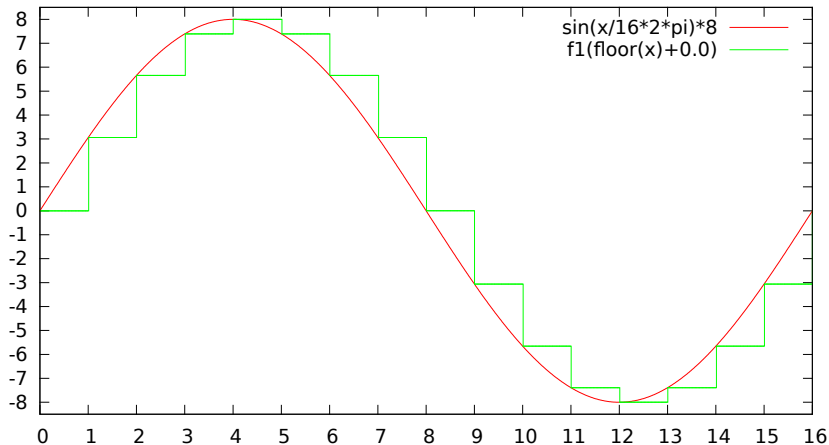


Zabawa z funkcjami

```
set encoding utf8
set title "Funkcje"
set xtic 1
set ytic 1
f1(x) = sin(x/16*2*pi)*8
round(x) = x - floor(x) < 0.5 ? floor(x) : ceil(x)
set terminal pdf
set output "funkcje3.pdf"
plot [0:16] [-8.5:8.5] f1(x)
title "sin(x/16*2*pi)*8" with lines,
f1(floor(x)+0.0) with steps
```


Zabawa z funkcjami

Funkcje



Zmienne i funkcje definiowane przez użytkownika

- Wszystkie funkcje i zmienne mają zasięg globalny.
- Wykaz wszystkich zmiennych i funkcji zdefiniowanych przez użytkownika można zobaczyć przy użyciu następujące dwa polecenia:

```
show variables
```

```
show functions
```

Liczby zespolone

- Gnuplot ma ograniczone wsparcie dla liczb zespolonych
- **Liczby zespolone** to pary liczb złożone z tzw. części rzeczywistej i urojonej.
- W gnuplocie, liczba zespolona jest przekazywana przy użyciu nawiasów klamrowych (`{}`), tj. liczba zespolona $z = 1 + i$, gdzie i jest jednostką urojoną w gnuplocie zapisywana jest jako: $z = \{1, 1\}$.
- Ważnym ograniczeniem liczby zespolonej w Gnuplocie jest to, że obie części muszą być stałymi numerycznymi a nie zmiennymi.
- Gnuplot może wykonywać proste działania arytmetyczne na liczbach zespolonych, np. $\{1, 1\} + \{-1, 0\}$.

```
gnuplot> z2 = {0,0} + {1,1}
```

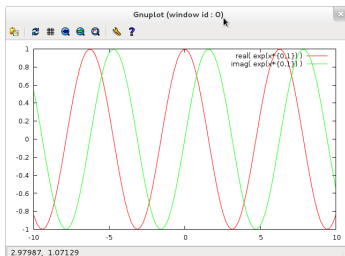
```
gnuplot> print z2
```

```
{1.0, 1.0}
```

Liczby zespolone

- Wiele wbudowanych funkcji matematycznych (takich jak $\sin(x)$, $\exp(x)$, itd.) może przyjąć argumenty zespolone i zwrócić liczby zespolone jako wyniki.
- Można również używać funkcji specjalnych $\mathit{real}(x)$ i $\mathit{imag}(x)$, aby wyłuskać odpowiednio część rzeczywistą i część urojoną liczby zespolonej x .

```
gnuplot> plot real( exp(x*{0,1}) ),  
imag( exp(x*{0,1}) )
```

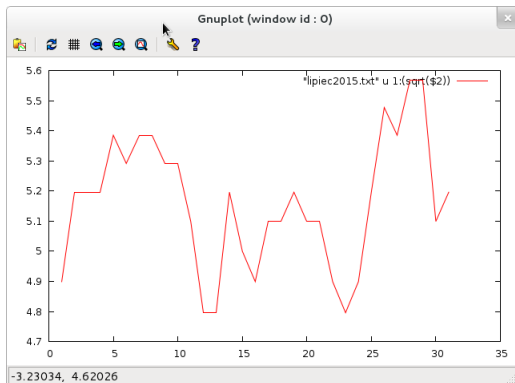


Proste przekształcenia danych

- W dyrektywie `using` komendy `plot` do każdego punktu danych można zastosować dowolną funkcję.
- 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 (`$`).

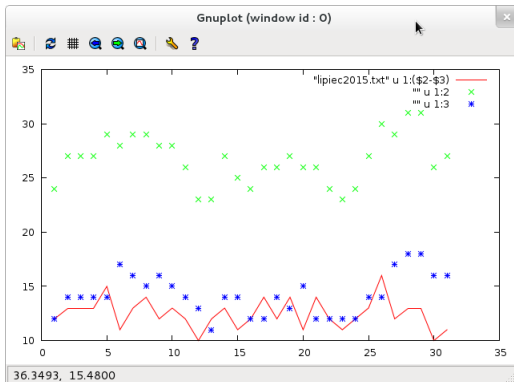
Proste przekształcenia danych

```
gnuplot> plot "lipiec2015.txt" u 1:(sqrt($2)) w l
```



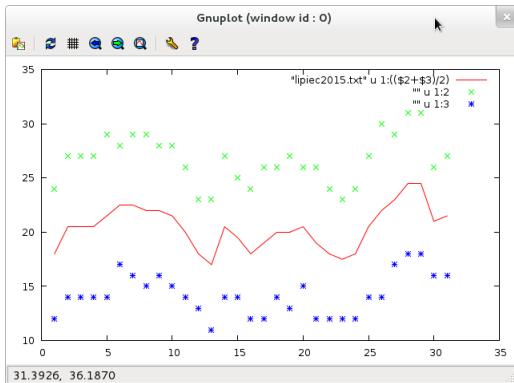
Proste przekształcenia danych

```
gnuplot> plot "lipiec2015.txt" u 1:($2-$3) w l,  
"" u 1:2, "" u 1:3
```



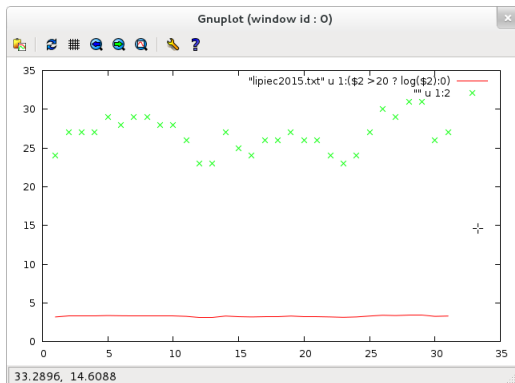
Proste przekształcenia danych

```
gnuplot> plot "lipiec2015.txt" u 1:((($2+$3)/2) w l,  
"" u 1:2, "" u 1:3
```



Proste przekształcenia danych

```
gnuplot> plot "lipiec2015.txt"  
u 1:($2 >20 ? log($2):0) w l, "" u 1:2
```

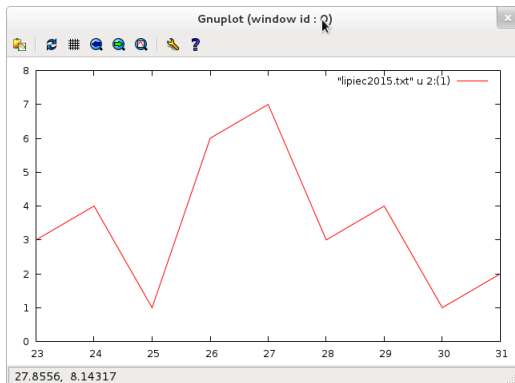


Proste przekształcenia danych

- Wyrażenia w nawiasach dyrektyw `using` mogą być stałymi.
- Przykład z kolejnego slajdu używa dyrektywy `smooth frequency`, aby policzyć ile razy każda wartość w drugiej kolumnie wystąpiła.
- **Pamięta:** dyrektywa `smooth frequency` sumuje dane dostarczane jako wartości `y` i kreśli ich sumę).

Proste przekształcenia danych

```
gnuplot> plot [][0:8]"lipiec2015.txt" u 2:(1)  
smooth frequency with lines
```

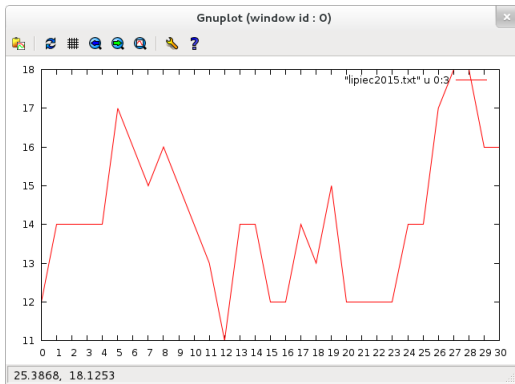


Pseudokolumny

- Gnuplot definiuje dwie **pseudokolumny**(0 i -2), które mogą być stosowane wraz z transformacjami danych.
- **Kolumna 0** zawiera numer linii w bieżącym zbiorze danych.
- **Kolumna -2** zawiera indeks aktualnych danych zawartych w pliku danych.
- Napotkana podwójna pusta linia w pliku powoduje, że numer linii jest ustawiany na zero, a indeks danych jest zwiększony o 1.

Pseudokolumnny

```
gnuplot> set xtics 1  
gnuplot> plot "lipiec2015.txt" u 0:3 with lines
```

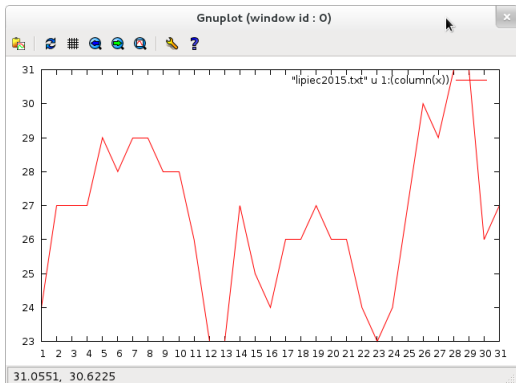


Pseudokolumny

- Innym sposobem wyboru kolumny jest zastosowanie funkcji `column(x)`.
- Funkcja `column(x)` oblicza swój argument i używa jego wartości (które powinny być liczbą całkowitą), aby wybrać kolumnę.
- Funkcja `column(x)` jest dostępna tylko wewnątrz nawiasów stosowanych jako część dyrektywy `using`.

Pseudokolumnny

```
gnuplot> set xtics 1  
gnuplot> x = 2  
gnuplot> plot "lipiec2015.txt" u 1:(column(x)) w l
```



Kreślenie funkcji i danych

- Dwie ważne opcje kreślenia funkcji: `set samples` i `set clip`.
- `set samples` kontroluje liczbę punktów, w których funkcja jest obliczana i wykreślana.
- Domyślna wartość dla `set samples` to 100 punktów.
- Sensowne wartości dla kreślenia bardziej wymagających krzywych to: 300-500.
- Ustawienie próbkowania większego niż liczba pikseli na urządzeniu wyjściowym nie ma sensu.
- Opcja `set clip` kontroluje jak gnuplot traktuje punkty danych, które są zbyt blisko lub poza zakresem wydruku.

Kreślenie funkcji i danych

- `set clip points` - ma znaczenie tylko w przypadku korzystania ze stylów, które pokazują dyskretne symbole (takie jak wykresy z punktów, z `linespoints`, i tak dalej).
- Jeśli `set clip points` zostało ustawione, to symbole, które dotykają lub nakładają się na granice wykresu nie są kreślone. Dokładnie ile punktów się obcina zależy od wielkości symbolu; dla większych symboli, więcej punktów jest odrzucane.
- `set clip [one | two]` - kontroluje jak gnuplot określi odcinki łączące punkty, jeśli co najmniej jeden z elementów znajduje się poza widzialnym zakresem tego wykresu.

Kreślenie funkcji i danych

- Jeśli ustawiony jest `set clip one`, to segmenty są rysowane w przypadku, gdy co najmniej jedno z punktów końcowych leży poza widzialnym zakresem tego wykresu.
- Jeśli ustawiony jest `set clip two`, to segmenty są rysowane jeśli oba punkty końcowe są poza bieżącym zakresem widzialnym wykresu.
- Domyślnie ustawiony jest `set clip one`, a `set clip two` jest wyłączony.

Wykresy logarytmiczne

```
# włączenie skali logarytmicznej na obu osiach
set logscale
# włączenie skali logarytmicznej na osi x
set logscale x
# włączenie skali logarytmicznej na osi y
set logscale y
# wyłączenie skali logarytmicznej na obu osiach
unset logscale
# wyłączenie skali logarytmicznej na osi x
unset logscale x
# wyłączenie skali logarytmicznej na osi y
unset logscale y
```