Piotr M. Szczypiński

Biostatystyka
Ćwiczenia laboratoryjne

część 2

Pakiet SciPy.stats

SciPy.stats

Biblioteka SciPy została pomyślany jako zbiór narzędzi przydatnych w badaniach naukowych. Między innymi zawiera ona zbiór narzędzi (funkcji i klas) do obliczeń statystycznych. Dostępne są one w pakiecie o nazwie `scipy.stats`. Pakiet udostępnia zestawy klas do analizy zmiennych losowych o rozkładach ciągłych (ponad 80) oraz dyskretnych (10).

Dokumentacja narzędzi do obliczeń statystycznych dostępna jest na stronie http://docs.scipy.org/doc/scipy/reference/stats.html. Samouczek poświęcony bibliotece zamieszczono na stronie http://docs.scipy.org/doc/scipy/reference/tutorial/stats.html.

Rozkład normalny

> Rozkład normalny, zwany też rozkładem Gaussa (w literaturze francuskiej – rozkładem Gaussa-Laplace'a) – jeden z najważniejszych rozkładów prawdopodobieństwa. Odgrywa ważną rolę w statystycznym opisie zagadnień przyrodniczych, przemysłowych, medycznych, społecznych itp. Wykres funkcji prawdopodobieństwa tego rozkładu jest krzywą dzwonową (ang. bell curve).
>
> Przyczyną jego znaczenia jest częstość występowania w naturze. Jeśli jakaś wielkość jest sumą lub średnią bardzo wielu drobnych losowych czynników, to niezależnie od rozkładu każdego z tych czynników, jej rozkład będzie zbliżony do normalnego, stąd można go bardzo często zaobserwować w danych. Ponadto rozkład normalny ma interesujące właściwości matematyczne, dzięki którym oparte na nim metody statystyczne są proste obliczeniowo.
>
> Funkcja gęstości prawdopodobieństwa rozkładu normalnego ze średnią μ i odchyleniem standardowym σ (równoważnie: wariancją σ^2) jest przykładem funkcji Gaussa. Dana jest ona wzorem:
>
> $$\phi_{\mu,\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)$$
>
> (Źródło: Edytorzy Wikipedii, "Rozkład normalny," Wikipedia, wolna encyklopedia, //pl.wikipedia.org/w/index.php?title=Rozkład_normalny (dostęp październik 19, 2015)

Funkcję Gaussa można w pythonie zdefiniować i wizualizować następującym kodem:

```
from numpy import *
from matplotlib.pyplot import *
mu = 5.0
sig = 2.0
x = linspace(-5, 5, 101)
y = exp(-power(x - mu, 2.) / (2 * power(sig, 2.)))
plot(x, y)
show()
```

Zmienna `mu` określa tutaj wartość średnią rozkładu, natomiast `sig` odchylenie standardowe. Mówimy o standardowym rozkładzie normalnym gdy wartość `mu` = 0 oraz `sig` = 1.

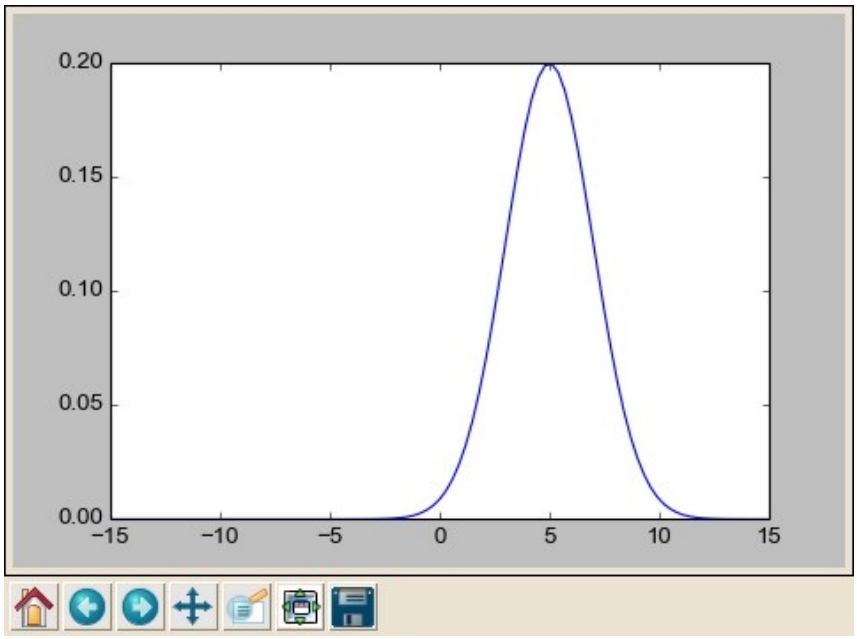
Pole powierzchni pod wykresem funkcji powinno mieć wartość równą jeden. Inaczej mówiąc, całka funkcji gęstości prawdopodobieństwa od -∞ do +∞ musi być równa jeden. Wynika to stąd, że prawdopodobieństwo zaistnienia zdarzenia losowego o dowolnej wartości jest równa jedności. Tego warunku nie spełnia jednak funkcja zdefiniowana powyższym kodem.

Wygodnym sposobem analizy zmiennych losowych o rozkładzie normalnym jest zastosowanie klasy `norm` biblioteki `scipy.stats`. Funkcja pdf (**p**robability **d**ensity **f**unction) tej klasy umożliwia wygenerowanie wartości gęstości standardowego rozkładu normalnego:

```
from numpy import *
from matplotlib.pyplot import *
from scipy.stats import *
x = linspace(-15, 15, 101)
y = norm.pdf(x)
plot(x, y)
show()
```

Na podstawie klasy norm definiującej standardowy rozkład normalny można wygenerować obiekt własnej klasy rozkładu normalnego, z określoną wartością średnią oraz odchyleniem standardowym. W poniższym przykładzie nadamy temu obiektowi nazwę `nowy_norm`. Do utworzenia obiektu służy konstruktor klasy norm z podanymi parametrami przesunięcia i skalowania. Proszę zwrócić uwagę, że wartość maksymalna funkcji jest odpowiednio skorygowana tak, aby pole powierzchni pod wykresem było równe jedności.

```
x = linspace(-15, 15, 101)
nowy_norm = norm(loc=5.0, scale=2.0)
y = nowy_norm.pdf(x)
plot(x, y)
show()
```



Rys. 1. Wykres funkcji gęstości prawdopodobieństwa rozkładu normalnego o średniej równej 5 i odchyleniu standardowym równym 2.

Symulacja procesu losowego

Przeprowadźmy teraz eksperyment. Naszym zadaniem będzie przycięcie listewek o długości 100 cm. Listewki będziemy przycinać "na oko" bez używania przymiaru. Przyjmijmy też, że błędy, które będziemy popełniać, będą opisane rozkładem normalnym o odchyleniu standardowym równym 4 cm. Naszym zadaniem będzie przycięcie dwudziestu takich listewek. Oczywiście, przycinanie będziemy symulować za pomocą programowania w pythonie.

Najpierw zdefiniujemy rozkład normalny o odpowiedniej wartości średniej i odchyleniu standardowym:

```
>>> ciecie = norm(loc=100.0, scale=4.0)
```

Wzorując się na wcześniejszych przykładach, należy wykreślić funkcję gęstości rozkładu prawdopodobieństwa i ocenić jego kształt.

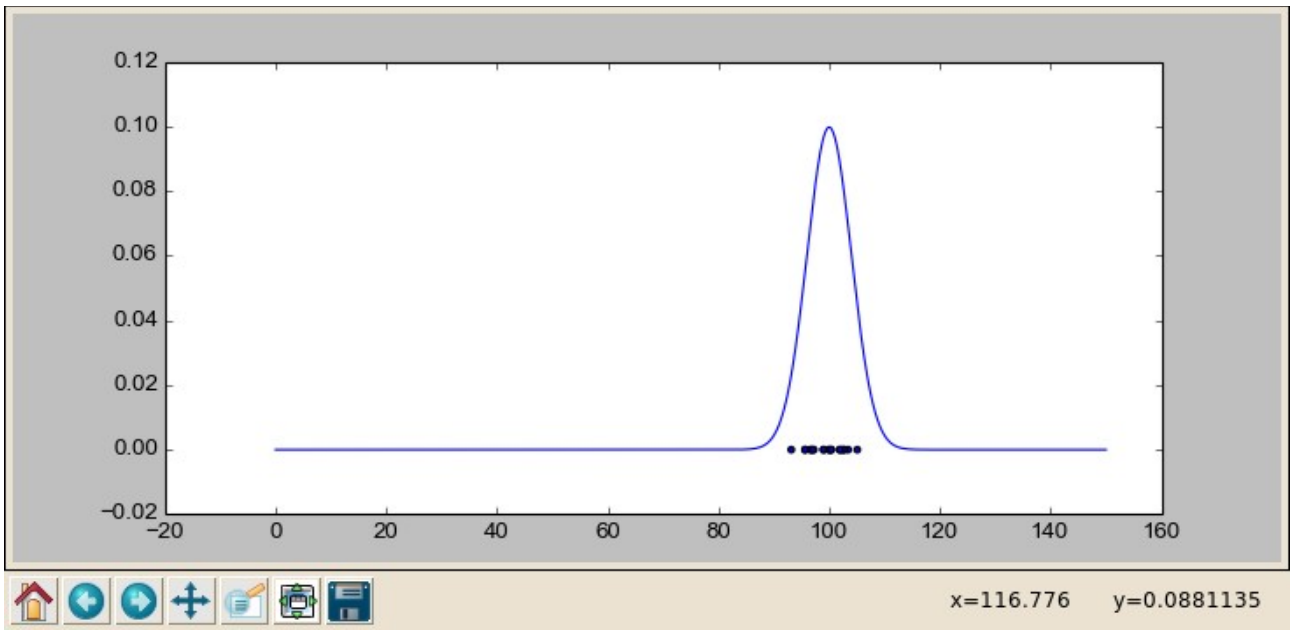
Teraz wygenerujemy dwadzieścia przykładowych długości listewek (zdarzeń losowych). Posłuży nam do tego funkcja `rvs` generująca zadaną liczbę przypadków, zgodnych ze zdefiniowanym rozkładem.

```
>>> listewki = ciecie.rvs(20)
>>> listewki
array([  97.66158494,   97.10440308,   99.68199941,  103.21015993,
        101.17013518,  105.58889144,  102.10077698,   99.88185083,
         99.46338206,  100.71656505,  105.14910646,  100.91474802,
        107.95611946,   98.49846886,   97.41393917,  100.11985839,
        101.20509861,  102.24460198,   96.09331436,   98.32039537])
```

Zmienna `listewki` jest wektorem przechowującym dwadzieścia liczb o wartościach zbliżonych do 100. Napiszmy teraz program, który na jednym wykresie pokaże przebieg funkcji gęstości prawdopodobieństwa oraz za pomocą kropek oznaczy długości listewek z tablicy `listewki`. Cały program może wyglądać następująco:

```
from numpy import *
from matplotlib.pyplot import *
from scipy.stats import *
x = linspace(0, 150, 1001)
ciecie = norm(loc=100.0, scale=4.0)
listewki = ciecie.rvs(20)
y = ciecie.pdf(x)
plot(x, y)
scatter(listewki, 0*listewki, 10)
show()
```

Wynik działania programu przedstawiono na rysunku 2. Można zauważyć, że największe zagęszczenie kropek rzeczywiście występuje w miejscu, w którym wartość funkcji gęstości prawdopodobieństwa jest największe.



Rys. 2. Wykres funkcji gęstości prawdopodobieństwa oraz wartości zmiennej losowej będące wynikiem symulacji.

Zadanie 1
Wygeneruj 100 losowych wartości o rozkładzie normalnym, którego średnia jest równa 10 a odchylenie standardowe 20. Wynik przedstaw i porównaj z wykresem funkcji gęstości prawdopodobieństwa. Czy możliwa jest wizualna ocena zgodności uzyskanych wartości z zadanym rozkładem?

Histogram

Do wizualnej oceny tego czy zbiór wartości liczbowych jest zgodny z określonym rozkładem można wykorzystać histogram. Histogram jest wykresem przybliżającym funkcję gęstości prawdopodobieństwa na podstawie pewnego zbioru wartości liczbowych - zdarzeń losowych. W tym celu obszar zmienności mierzonej wielkości dzielimy na przedziały (zwane przedziałami klasowymi), a następnie dla każdego przedziału określamy ile wartości (zdarzeń losowych) mieści się w tym przedziale. Wynik przedstawiany jest w postaci wykresu słupkowego.

Histogram można w pythonie wygenerować funkcją `hist`, której pierwszy argument to nazwa wektora zmiennych natomiast druga to liczba przedziałów.

Przykładowy program generujący pewną liczbę zdarzeń losowych i przedstawiający wykres ich histogramu może wyglądać tak:

```
ciecie = norm(loc=100.0, scale=4.0)
listewki = ciecie.rvs(200)
hist(listewki, 15)
show()
```

Zadanie 2
Program z zadania 1 uzupełnij o wizualizację histogramu. Przeskaluj (przemnóż) wartości funkcji gęstości prawdopodobieństwa, aby kształt jej wykresu łatwiej można było porównać z histogramem.

Zadanie 3
Ze strony z dokumentacją `scipy.stats` wybierz inny rodzaj rozkładu losowego i dla niego napisz program o podobnym działaniu jak ten z zadania 2.

Rozkłady wielowymiarowe

W praktyce jedno zdarzenie losowe albo element zbiorowości nie musi być opisane wyłącznie jedną wartością. Jedno zdarzenie może być opisane kilkoma wartościami. Przykładowo w zbiorowości ludzi, każdy człowiek charakteryzuje się wzrostem, wagą oraz wiekiem. W tym sensie, element zbiorowości można opisać jako punkt (wektor) w trójwymiarowej przestrzeni trzech tych cech.

Poniższy przykład generuje dwadzieścia zdarzeń losowych o trzech niezależnych od siebie cechach każda o rozkładzie normalnym. Wektory cech są następnie prezentowane w przestrzeni trójwymiarowej (Rysunek 3.a).

```
from numpy import *
from matplotlib.pyplot import *
from scipy.stats import *
from mpl_toolkits.mplot3d import Axes3D
from mpl_toolkits.mplot3d import proj3d
nowy_rozklad = norm(loc=0.0, scale=4.0)
X = nowy_rozklad.rvs(50)
Y = nowy_rozklad.rvs(50)
Z = nowy_rozklad.rvs(50)
fig = figure(figsize=(8,8))
ax = fig.add_subplot(111, projection='3d')
ax.plot(X, Y, Z, 'o')
show()
```

Zmodyfikujemy teraz fragment kodu w taki sposób, aby zmienne losowe były od siebie proporcjonalnie zależne.

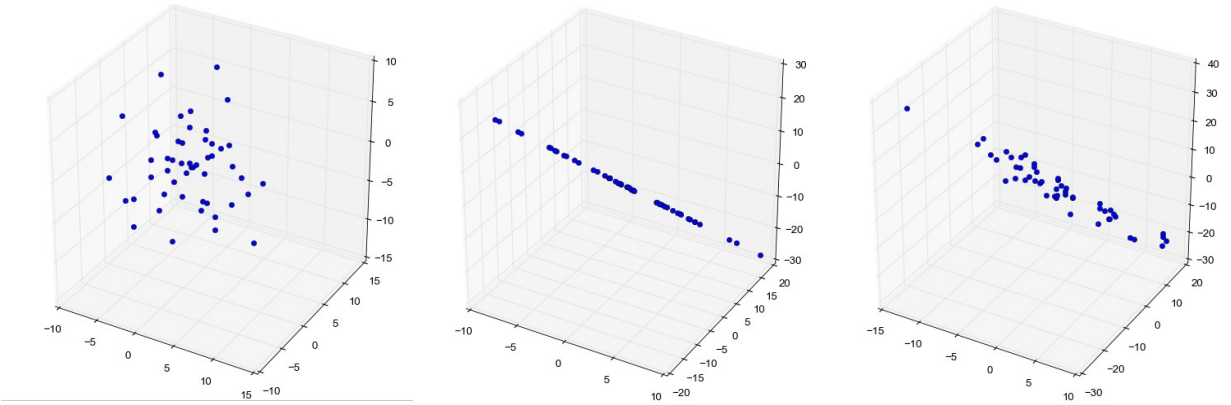
```
X = nowy_rozklad.rvs(50)
Y = 2*X
Z = -3*X
```

Wektory zmiennej losowej układają się teraz wzdłuż jednej linii (Rys. 3.b).

Zmieńmy teraz kod w taki sposób, aby zmienne były od siebie zależne, ale już nie w tak jaskrawy sposób.

```
X = nowy_rozklad.rvs(50)
W = nowy_rozklad.rvs(50)
Y = 2*X + 0.2*W
Z = -3*X + 0.5*W
```

Porównaj trzy uzyskane wykresy punktowe. Czy łatwo jest stwierdzić, że cechy zdarzenia losowego są od siebie niezależne?



Rys. 3. Wykresy trójwymiarowych rozkładów normalnych zmiennej losowej o różnym stopniu zależności pomiędzy cechami.

Format CSV

CSV (ang. comma-separated values, wartości rozdzielone przecinkiem) jest formatem przechowywania danych numerycznych w postaci tabel w plikach tekstowych.

Utwórz dowolną macierz zgodną z biblioteką numpy. Zapisz wartości z macierzy do pliku za pomocą instrukcji:

```
savetxt(open('macierz.csv', 'wb'), x, delimiter=',')
```

Zamiast nazwy zmiennej `x` podstaw nazwę utworzonej przez siebie macierzy. Otwórz zapisany plik w edytorze tekstu i sprawdź jego zawartość.

Załaduj dane z zapisanego uprzednio pliku do nowej macierzy. Użyj do tego instrukcji:

```
x = loadtxt(open('xyz.csv', 'rb'), delimiter=',', skiprows=1)
```

Zamiast nazwy zmiennej `x` użyj innej dowolnej nazwy. Sprawdź jakie wartości zawiera załadowana macierz.