Regresja

Statystyka biomedyczna

Piotr M. Szczypiński

Regresja

Regresja to metoda statystyczna pozwalająca na badanie związku pomiędzy zmiennymi i przewidywanie na tej podstawie nieznanych wartości jednej zmiennej (**zmiennej objaśnianej**) na podstawie znanych wartości innych zmiennych (**zmiennych objaśniających**).

Zastosowanie składa się z dwóch etapów:

1) **konstruowanie** modelu (uczenie) – znalezienie funkcji (w postaci wzoru matematycznego, algorytmu, drzewa decyzyjnego) opisującej zależność, oraz doboru współczynników tej funkcji tak aby jak najlepiej pasowała do danych z próby.
2) **zastosowanie** modelu – użycie skonstruowanego modelu do danych, w których znamy tylko zmienne objaśniające, w celu wyznaczenia wartości oczekiwanej zmiennej objaśnianej.

Regresja

Model (funkcja) regresji przybiera zwykle postać:

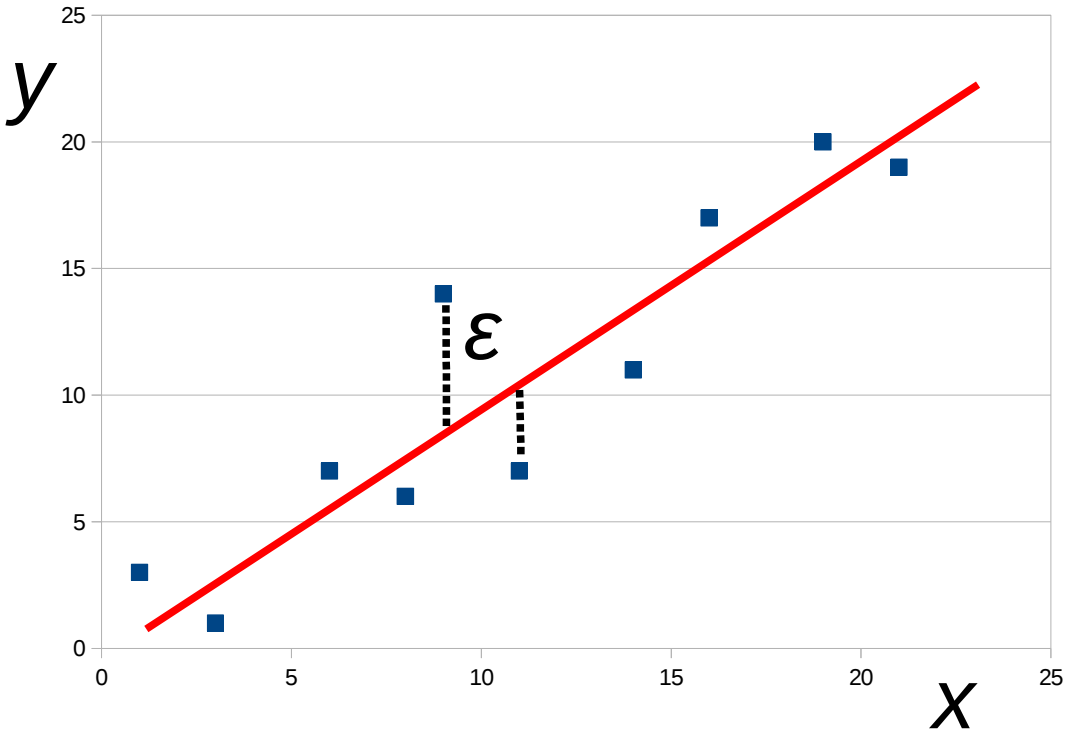
$$Y = f\left(\mathbf{X}, \boldsymbol{\beta}\right) + \varepsilon$$

Y - zmienna objaśniana
$\mathbf{X}$ - wektor zmiennych objaśniających (predyktorów)
$\boldsymbol{\beta}$ - wektor współczynników
ε - błąd losowy

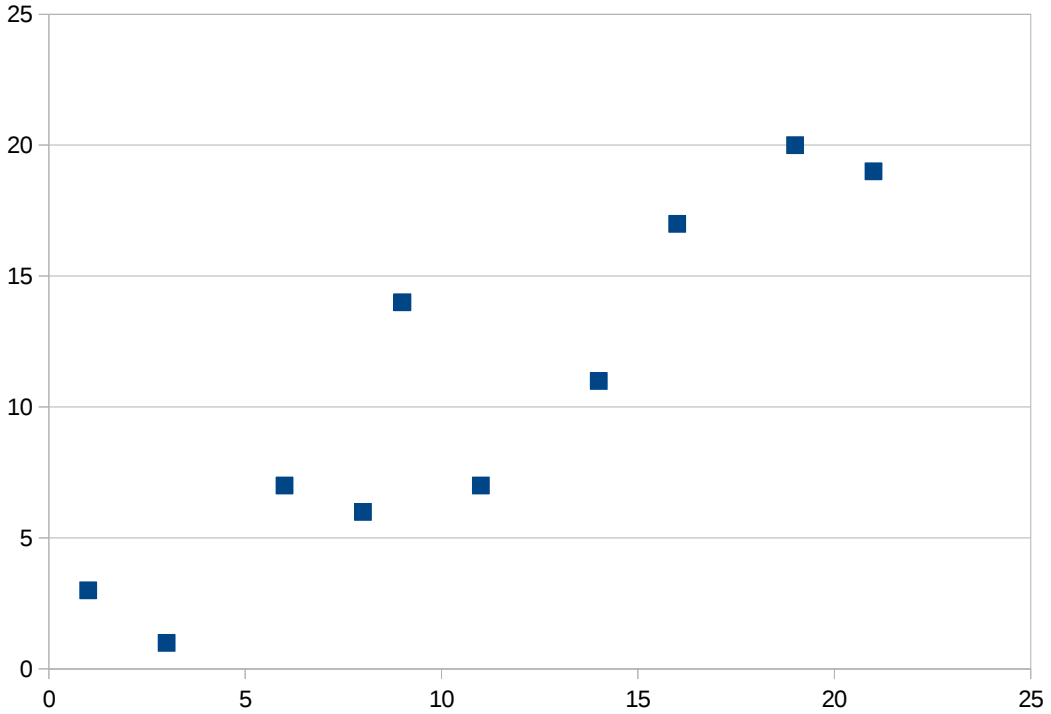
Regresja liniowa

Model ma postać funkcji liniowej:

$$Y = f(\mathbf{X}, \boldsymbol{\beta}) + \varepsilon \quad \Rightarrow \quad y = b_0 + b_1 x_1 + b_2 x_2 + \ldots + \varepsilon$$



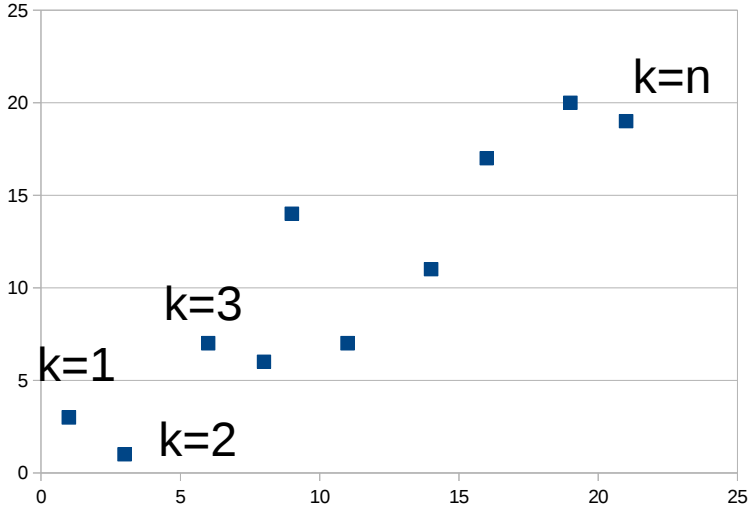
Regresja liniowa - uczenie



1) Zdecydowaliśmy o postaci funkcji $y = b_0 + b_1 x + \varepsilon$
2) Znamy pewien zbiór par zdarzeń zmiennych losowych *x*, *y*
3) Chcemy mieć jak najmniejsze błędy *ε*
4) **Nie znamy współczynników** b_0, b_1

Minimalizacja błędu średniokwadratowego

Algorytm minimalizacji błędu średniokwadratowego umożliwia analityczne rozwiązanie problemu doboru współczynników funkcji liniowej.



$$y = b_0 + b_1 x_1 + \varepsilon$$

$$\varepsilon = y - \left(b_0 + b_1 x \right)$$

$$\sum_{k=1}^{n} \varepsilon_k^2 = \sum_{k=1}^{n} \left(y_k - \left(b_0 + b_1 x_k \right) \right)^2$$

Minimalizacja błędu średniokwadratowego (c.d.)

$$E(b_0, b_1) = \sum_{k=1}^{n} \varepsilon_k^2 = \sum_{k=1}^{n} \left(y_k - (b_0 + b_1 x_k)\right)^2 =$$

$$E(b_0, b_1) = \sum \left(b_1^2 x^2 + b_0^2 - 2\, y\, b_0 - 2\, x\, y\, b_1 + 2\, b_0 b_1\, x + y^2\right)$$

Dla ekstremum (minimum) pochodne muszą być zerowe:

$$\frac{\partial E}{\partial b_0} = \sum \left(2\, b_0 - 2\, y + 2\, b_1 x\right) = 0$$

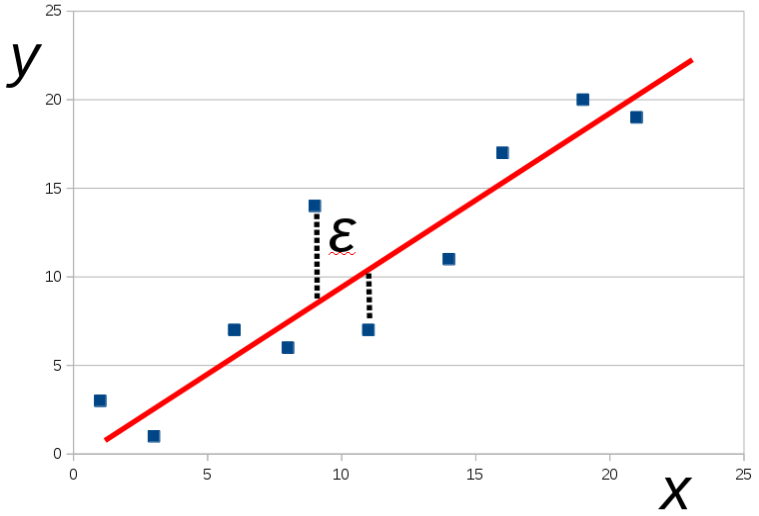
$$\frac{\partial E}{\partial b_1} = \sum \left(2\, b_1 x^2 - 2\, xy + 2\, b_0 x\right) = 0$$

$$\begin{cases} b_1 \sum x + b_0 K - \sum y = 0 \\ b_1 \sum x^2 + b_0 \sum x - \sum x\, y = 0 \end{cases}$$

Praca domowa: sprawdź czy w wywodzie nie ma błędu i wyprowadź wartości współczynników b1 i b0 z układu równań.

Regresja liniowa - rozwiązanie

$$y = b_0 + b_1 x_1 + \varepsilon$$



$$b_1 = \frac{S_{xy}}{S_{xx}}$$

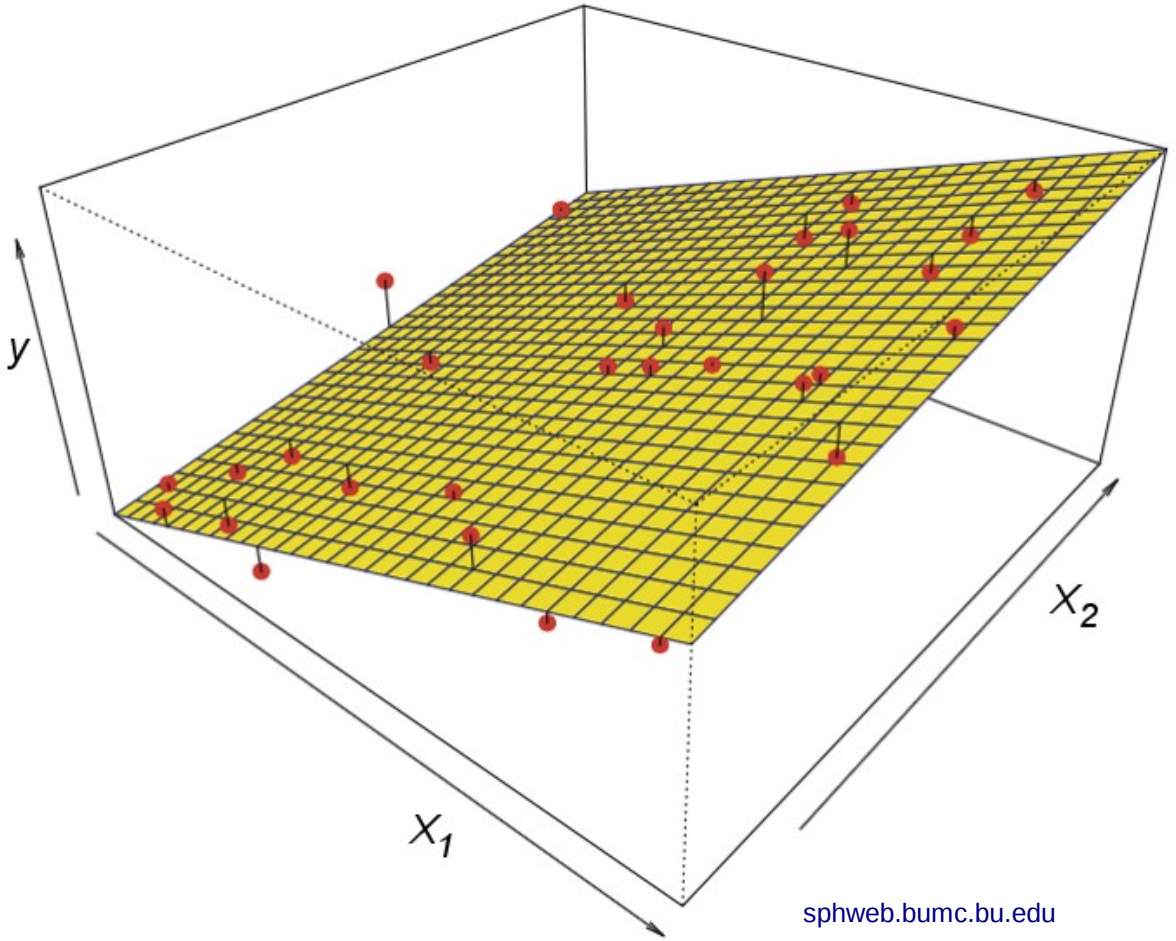
$$b_0 = \bar{y} - b_1 \bar{x} = \frac{1}{K}\sum_{k=0}^{n} y - b_1 \frac{1}{K}\sum_{k=0}^{n} x$$

$$\bar{y} = \frac{1}{K}\sum_{k=0}^{n} y \qquad \bar{x} = \frac{1}{K}\sum_{k=0}^{n} x$$

$$S_{xx} = \sum_{k=0}^{n} \left(x_k - \bar{x}\right)^2 \qquad S_{xy} = \sum_{k=0}^{n} x_k y_k + n\bar{x}\bar{y}$$

Wielowymiarowa regresja liniowa

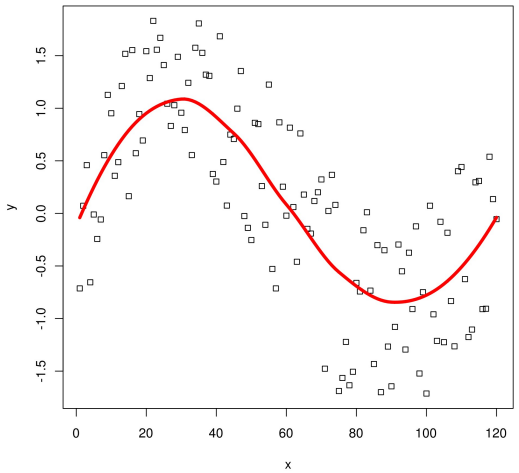
$$y = b_0 + b_1 x_1 + b_2 x_2 + \ldots + \varepsilon$$



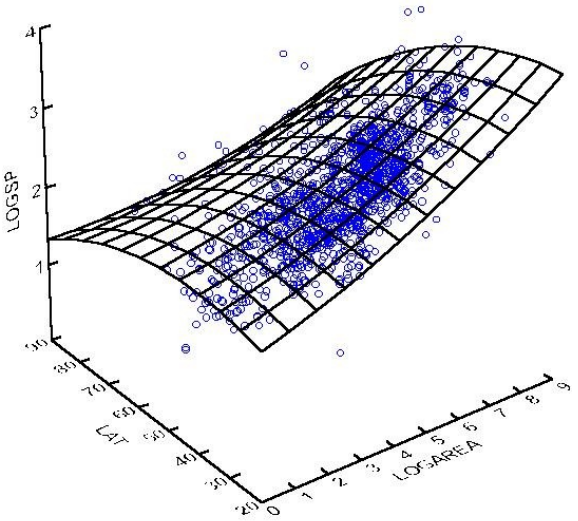
sphweb.bumc.bu.edu

Regresja funkcją wielomianu

$$y = b_0 + b_{11} x_1 + b_{12} x_1^2 + \ldots + b_{21} x_2 + b_{22} x_2^2 + \ldots + \varepsilon$$



en.wikipedia.org



ordination.okstate.edu

Funkcja jest nadal liniowa względem współczynników *b.*
Można więc rozwiązać problem analitycznie stosując minimalizację błędu średnikwadratowego!

Regresja liniowa i nieliniowa

$$y = b_0 + b_1 f_1(x_1, x_2 \ldots) + b_2 f_2(x_1, x_2 \ldots) + \ldots + \varepsilon$$

Funkcja jest nadal liniowa względem współczynników *b.* Ma rozwiązanie analityczne stosując minimalizację błędu średnikwadratowego.

$$y = b_1 e^{b_2 x} \varepsilon$$

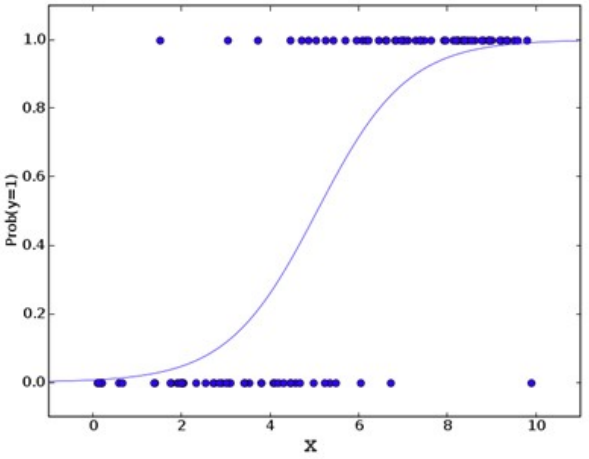
$$\ln(y) = \ln(b_1) + b_2 x + \ln(\varepsilon)$$

Przykład funkcji nieliniowej, którą można poddać linearyzacji. Można znaleźć rozwiązanie analityczne stosując minimalizację błędu średnikwadratowego logarytmów.

$$y = \frac{b_1 x}{b_2 + x} + \varepsilon$$

Przykład funkcji nieliniowej względem współczynników *b.* Rozwiązania poszukujemy metodami numerycznymi, z wykorzystaniem algorytmów optymalizacji numerycznej (minimalizacji błędu).

Regresja logistyczna

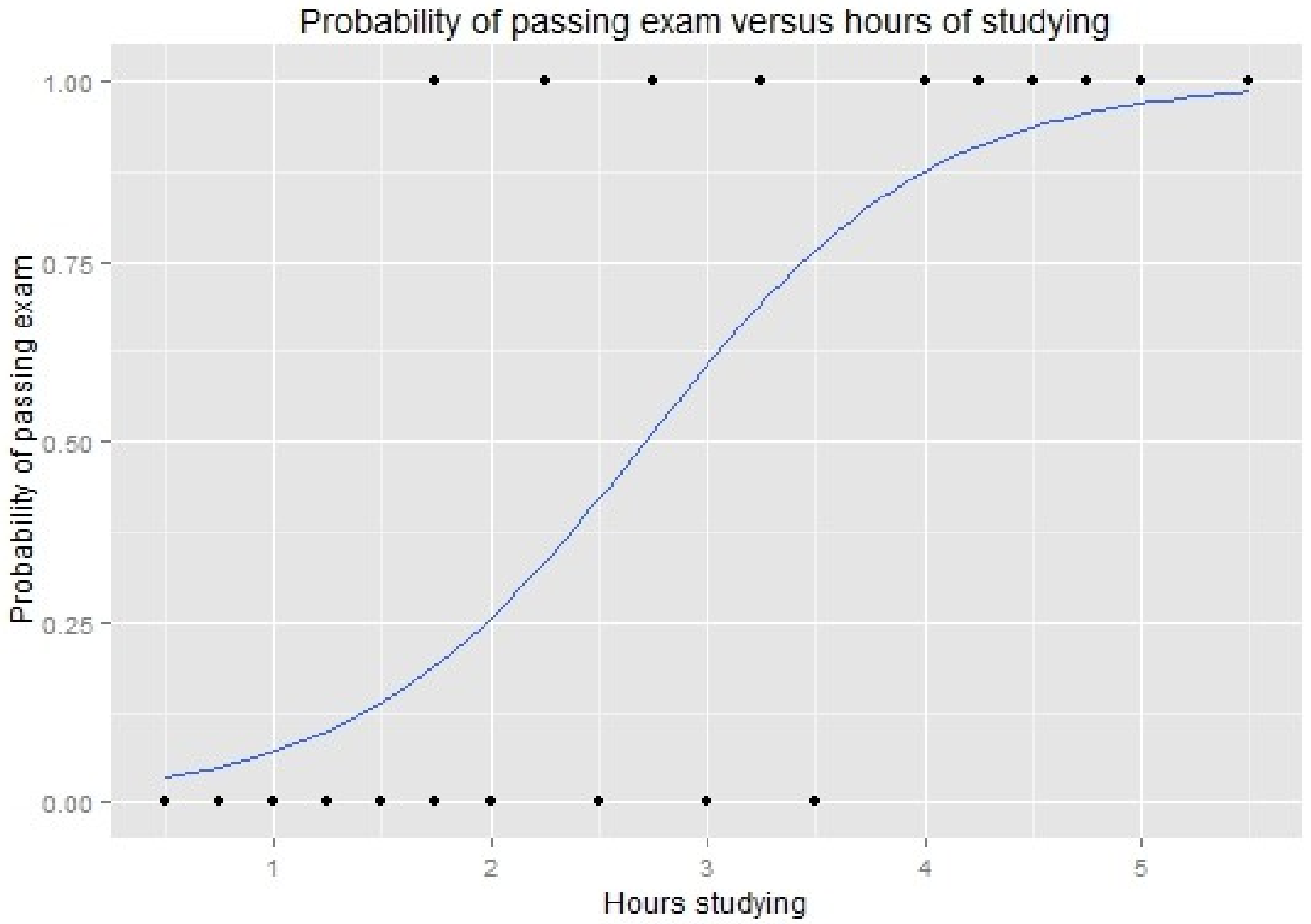


Regresja logistyczna – jedna z metod regresji używanych w statystyce w przypadku, gdy zmienna objaśniana jest na **skali dychotomicznej** (przyjmuje tylko dwie wartości).

Zwykle wartości zmiennej objaśnianej wskazują na wystąpienie, lub brak wystąpienia pewnego zdarzenia, które chcemy prognozować. Regresja logistyczna pozwala wówczas na obliczanie prawdopodobieństwa tego zdarzenia (prawdopodobieństwo sukcesu).

Formalnie model regresji logistycznej jest uogólnionym modelem liniowym, w którym użyto **logitu** jako funkcji wiążącej.

https://pl.wikipedia.org/wiki/Regresja_logistyczna



https://en.wikipedia.org/wiki/File:Exam_pass_logistic_curve.jpeg

Regresja logistyczna - logit

Szansa (*Odds*) jest definiowana jako:

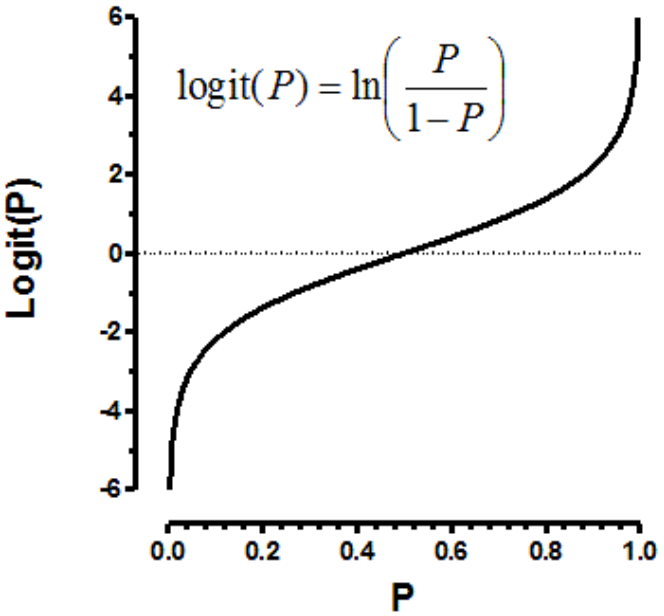
$$Odds(p) = \frac{p}{1-p}$$

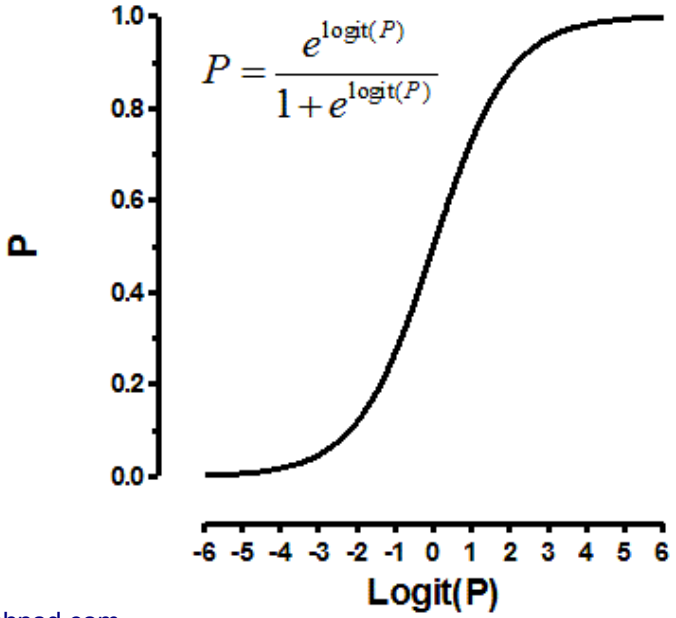
Logit to funkcja przekształcająca prawdopodobieństwo na logarytm szansy:

$$logit(p) = \ln\left(\frac{p}{1-p}\right)$$

Funkcja odwrotna logitu:

$$p = \frac{1}{1+e^{-logit(p)}}$$





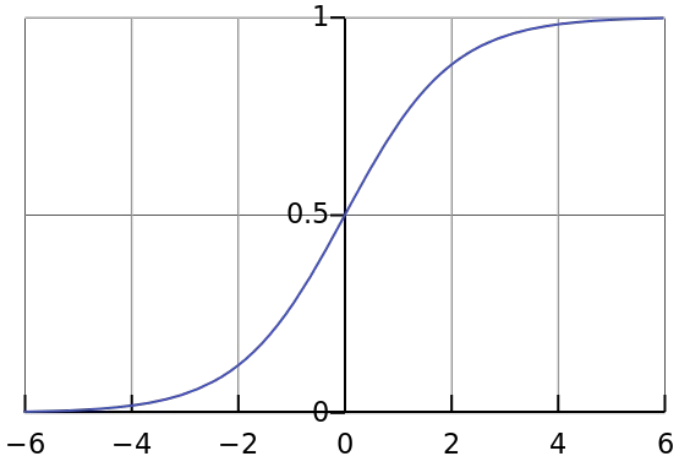
www.graphpad.com

Regresja logistyczna - model

Logit nieznanego prawdopodobieństwa sukcesu *p* jest modelowany jako liniowa funkcja zmiennej wilowymiarowej **X**:

$$logit(p) = \ln\left(\frac{p}{1-p}\right) = b_0 + b_1 x_1 + \ldots + b_n x_n$$

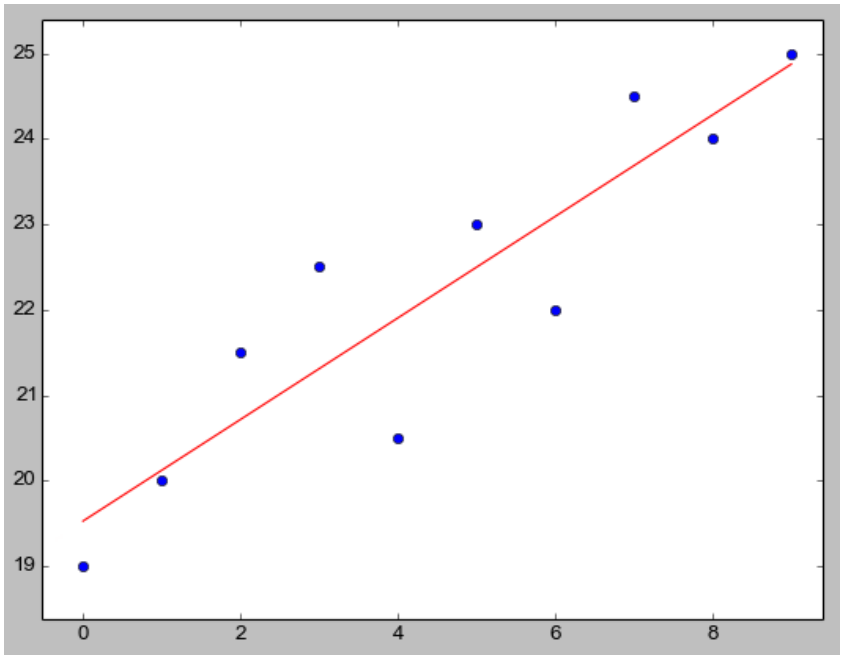
lub $$p = \frac{1}{1+e^{-(b_0 + b_1 x_1 + \ldots + b_n x_n)}}$$



W przypadku jednowymiarowym współczynnik b_0 decyduje o przesunięciu funkcji w poziomie natomiast b_1 o stromości zbocza.

Python: przykład regresji liniowej

```
# Przykład dopasowania funkcji liniowej
# author: Piotr Szczypiński, 2015

from numpy import arange, array, ones, linalg
from pylab import plot, show

# Przygotowanie danych przykładowych
x = arange(0,10)
A = array([ x, ones(10)])
y = [19, 20, 21.5, 22.5, 20.5, 23, 22, 24.5, 24, 25]

# Obliczenie współczynników b
b = linalg.lstsq(A.T, y)[0]

# Obliczenie punktów prostej
line = b[0]*x + b[1]

# Rysowanie punktów i prostej
plot(x, line,'r-', x, y, 'o')
show()
```



Python: regresja logistyczna

```
# Przykład dopasowania funkcji odwrotnej do logit
# author: Piotr Szczypiński, 2015

from numpy import arange
from scipy.optimize import curve_fit
from pylab import plot, show

# delogit - funkcja odwrotna do logitu
def delogit(x, b0, b1):
    e = 2.71828182846
    return 1.0 / (1.0 + e**(-(b0 + b1*x)))


# przygotowanie danych
x = arange(0, 14)
y = [0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1]

# uniwersalna funkcja dopasowania krzywej
b, pcov = curve_fit(delogit, x, y)

# wyrysowanie wyniku
line = delogit(x, b[0], b[1])
plot(x, line, 'r-', x, y, 'o')
Show()
```