

薬物動態：経口投与 1-コンパートメントモデルのあてはめ計算

薬物動態、経口投与1-コンパートメントモデルのあてはめ計算 (prog02_02.py、prog02_03.py)

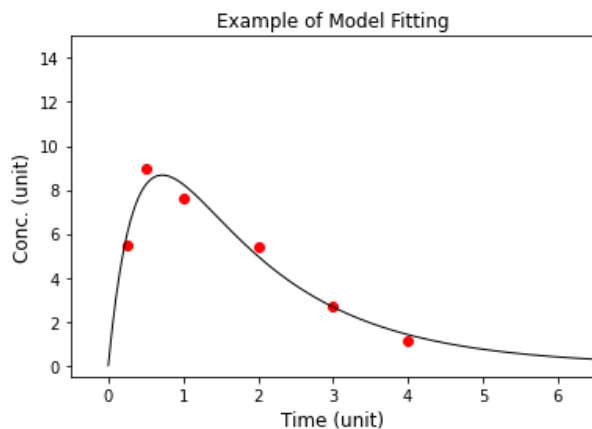
* 前回 (#001) の続きとなるので、前回の記事を確認した上で下記を参照することをお勧めします。

(1) データの入力：前回同様、プログラム中にデータを記載した。投与量 (DOSE)、薬物動態パラメータであるクリアランス (CL)、分布容積 (VD)、に加えて吸収速度定数 (KA)、吸収時のラグタイム (prog02_03.py) をグローバル変数として定義した。

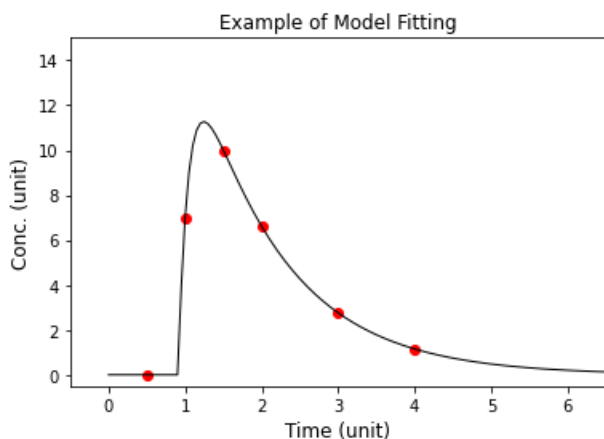
* prog02_03.py はラグタイムのあるモデルの場合の例。

(2) モデルあてはめ、結果のグラフ化、得られたパラメータ値の出力：前回同様。

以上の内容で作成した例を次ページ「prog02_02.py、prog02_03.py」にそれぞれテキストとして示した。出力結果は次のとおり。



```
-- Estimated Parameters --  
CL = 4.6679  
Vd = 7.2693  
Ke = 0.6421  
* t1/2 = 1.0794  
Ka = 2.5702
```



```
-- Estimated Parameters --  
CL = 5.7926  
Vd = 6.6187  
Ke = 0.8752  
* t1/2 = 0.7920  
Ka = 7.1404  
Lag T = 0.9072
```

以上

```

"""# Program 02-02 "prog02_02.py"""
# Fitting to One-Compartment po model
#
#import math
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
#
# Global variables
DOSE = 100.
CL = 4.0
VD = 5.
KA = 1.5
#
def set_data():
    # Data input in this program
    data_x = [0.25, 0.5, 1.0, 2.0, 3.0, 4.0]
    data_y = [5.5, 9.0, 7.6, 5.4, 2.7, 1.1]
    return data_x, data_y
#
def model_eq121(x, a, b, c):
    # 1-Comp. po model
    temp = np.exp(-a / b * x) - np.exp(-c * x)
    return DOSE * c / b / (c - a/b) * temp
#
def draw_plot(data_x, data_y):
    # Data plotting
    plt.title("Example of Model Fitting", fontsize = 12)
    plt.xlim(-0.5, 6.5)
    plt.ylim(-0.5, 15.0)
    plt.xlabel("Time (unit)", fontsize = 12)
    plt.ylabel("Conc. (unit)", fontsize = 12)
    plt.plot(data_x, data_y, "o", color = "red")
    # Simulation of predicted curve
    pred_x = np.arange(0., 10., 0.05)
    pred_y = model_eq121(pred_x, CL, VD, KA)
    plt.plot(pred_x, pred_y, linestyle = "solid", linewidth = "1", color = "black")
    plt.show()

```

```

#
def print_output():
# Output
    print ("-- Estimated Parameters --")
    print ('    CL = {:.4f}'.format(CL))
    print ('    Vd = {:.4f}'.format(VD))
    print ('    Ke = {:.4f}'.format(CL / VD))
    print (' * t1/2 = {:.4f}'.format(np.log(2) / (CL / VD)))
    print ('    Ka = {:.4f}'.format(KA))
#

if __name__ == '__main__':
    x, y = set_data()
    p_init = (4.0, 5.0, 1.5)
    param, cov = curve_fit(model_eq121, x, y, p0 = p_init)
    CL = param[0]
    VD = param[1]
    KA = param[2]
    draw_plot(x, y)
    print_output()

```

```

"""# Program 02-03 "prog02_03.py"""
# Fitting to One-Compartment po model
#
#import math
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
#
# Global variables
DOSE = 100.
CL = 4.0
VD = 5.
KA = 1.5
TLAG = 0.5
#
def set_data():
    # Data input in this program
    data_x = [0.5, 1.0, 1.5, 2.0, 3.0, 4.0]
    data_y = [0.0, 7.0, 10.0, 6.6, 2.8, 1.1]
    return data_x, data_y
#
def model_eq131(x, a, b, c, d):
    # 1-Comp. po model
    temp1 = 0.0
    temp2= np.exp(-a / b * (x - d)) - np.exp(-c * (x - d))
    return DOSE * c / b / (c - a/b) * (temp1 * (x <= d) + temp2 * (x > d))
#
def draw_plot(data_x, data_y):
    # Data plotting
    plt.title("Example of Model Fitting", fontsize = 12)
    plt.xlim(-0.5, 6.5)
    plt.ylim(-0.5, 15.0)
    plt.xlabel("Time (unit)", fontsize = 12)
    plt.ylabel("Conc. (unit)", fontsize = 12)
    plt.plot(data_x, data_y, "o", color = "red")
    # Simulation of predicted curve
    pred_x = np.arange(0., 10., 0.05)
    pred_y = model_eq131(pred_x, CL, VD, KA, TLAG)

```

```

plt.plot(pred_x, pred_y, linestyle = "solid", linewidth = "1", color = "black")
plt.show()
#
def print_output():
# Output
    print ("-- Estimated Parameters --")
    print ('    CL = {:.4f}'.format(CL))
    print ('    Vd = {:.4f}'.format(VD))
    print ('    Ke = {:.4f}'.format(CL / VD))
    print (' * t1/2 = {:.4f}'.format(np.log(2) / (CL / VD)))
    print ('    Ka = {:.4f}'.format(KA))
    print (' Lag T = {:.4f}'.format(TLAG))
#
if __name__ == '__main__':
    x, y = set_data()
    p_init = (4.0, 5.0, 1.5, 0.5)
    param, cov = curve_fit(model_eq131, x, y, p0 = p_init)
    CL = param[0]
    VD = param[1]
    KA = param[2]
    TLAG = param[3]
    draw_plot(x, y)
    print_output()

```