ISSN: 2456-9909 (Online)

International Journal of Research in AYUSH and Pharmaceutical Sciences

Research Article

PREDICTION OF LINEARITY AND NON-LINEARITY IN PHARMACEUTICAL OPTIMIZATION STUDIES WITH PYTHON

T. Bhanuteja[1], A. Lakshmana Rao[2]*, T.E.G.K. Murthy[3], T. Pallavi[4]

[1]VIT University, Vellore, Tamil Nadu, India.

*[2]V. V. Institute of Pharmaceutical Sciences, Gudlavalleru, Andhra Pradesh, India.

[3]Bapatla College of Pharmacy, Bapatla, Andhra Pradesh, India.

[4]KL University, Guntur, Andhra Pradesh, India.

ARTICLE INFO

Article history:
Received: 15-04-2022
Revised: 27-04-2022
Accepted: 12-05-2022

Keywords:
Pharmaceutical formulations, Optimization, Variables, Experimental Design.

ABSTRACT

Novel simple user-friendly python programme was developed to predict linearity and non-linearity in pharmaceutical optimization. Optimization is the process of obtaining optimum formulation. There are independent and dependent variables in optimization techniques regarding pharmaceutical formulations. The number of levels of independent factor is usually selected based on the linear/ non-linear relationship existing between the dependent and independent variable. The programme is run after entering the independent and dependent variables. The program is used to detect the best fitted model based on the observed correlation between dependent and independent factors, to predict the outcome against the input (independent variables). The program output is the regression coefficients, regression equations, predicted dependent variable and standard error of point estimate. The model offering the low error of point estimate is assumed to be the best fitted model for the given data. The model is applied successfully for both linear and non-linear data.

INTRODUCTION

Optimization refers to obtaining resulting actions of our own interest by changing the independent variables one by one [1]. Orthogonal functions satisfying a second order differential equation, rotatable design and simplex lattice designs are commonly employed to optimise the composition of pharmaceutical formulations [2]. Evolutionary operations, Lagrangian, search and canonical analysis are commonly used for optimisation studies. The preferred optimization techniques are sequential optimization techniques, simultaneous optimization techniques and combination of both. A sequential model- based optimization (SMBO) study involves the performance of experiments repeatedly and the observations are fitted in to different models to identify the better choices about the configurations to be investigated. It allows interpolation of performance observed between parameter settings and facilitate for extrapolation to other regions of design space. Simultaneous methods involve (a) framing the experimental design (b) performing the experiments as per experimental design (c) insertion of the results in appropriate mathematical model (d) observing the maximum or minimum response through the best fitted model identified from a set of equations. To ascertain the system behaviour, a predictive model is required. Optimization algorithms are used in (a) experimental design, model development, parameter estimation, and statistical analysis; (b) process design, development, analysis, and retrofit; (c) model predictive control of risk factors and real-time optimization; and (d) identification, implementation and the coordination of a series of process operations related to the manufacturing and distribution of drug product. In the operation of pharmaceutical processes, there is huge interest in improving the scheduling and

Access this article online	
Quick Response Code	
	https://doi.org/10.47070/ijraps.v6i3.126
	Published by Mahadev Publications (Regd.) publication licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

planning of various unit operations. These tasks are usually formulated as linear programming. A simplex lattice design is one of the most commonly used techniques to optimize the formulation variables [3,4].

METHODS

The development of large-scale non-linear programming tools has led to the application of real-time optimization process and responds to changes in the production schedule and from other inputs. Linear equations yields straight lines and these equations are always represents straight lines. Non-linear equation forms a curve. Linear equations have one degree and non-linear equations have equal to or greater than 2 degree.

The general linear equation is expressed as;

$$y = mx + c$$

Where x and y are the independent and dependent variables respectively, m is the slope observed from the straight line and c is an intercept value. Quadratics or quadratic equations are polynomial equation of a second degree and comprises at least one term that is squared. The quadratic equation is expressed as;

$$Y = aX^2 + bx + c$$

Where x is an independent variable and a, b, c are numerical coefficients computed by solving nonlinear parabolic equation.

If the polynomials have the degree three, they are known as cubic polynomials and represented as:

$$Y = ax^3 + bx^2 + cx + d$$

If the observed data follows linearity, then the factorial design can be used and data exhibits curvature and then response surface design is preferred. The objective of this study is identification of best fitted model, determination of predicted values and standard error of point estimate by using a suitable and easily operatable python programme. Python is a multi-purpose popular programming language currently used worldwide. Its versatile, can be operated on any computer, and have good productivity, work flow speed. So, a python program is developed and presented here.

Programme operation

The python programme is operated for the linear and non-linear data.

Python program

```
l_nl.py > ...
import numpy as np
x = np.array([2,4,6,8,10])
y = np.array([0.104,0.208,0.312,0.416,0.520])
#print(x)
#print(y)

p1 = np.polyfit(x,y,1)
p2 = np.polyfit(x,y,2)
p3 = np.polyfit(x,y,3)
print("Co efficients of the linear regression are shown below")
print(p1)
print("corresponding line equation is "+str(p1[0])+"x+ "+str(p1[1]))
print("Co efficients of the second degree parabola are shown below")
print(p2)
print("corresponding line equation is "+str(p2[0])+" x^2+ "+str(p2[1])+" x+ "+str(p2[2]))

print("Co efficients of the third degree cubic equation are shown below")
print(p3)
print("corresponding line equation is "+str(p3[0])+" x^3+ "+str(p3[1])+" x^2+ "+str(p3[2])+" x+ "+str(p3[3]))


y1=[]
for i in range(len(x)):
    y_1=p1[0]*x[i]+p1[1]
    y1.append(y_1)
print("corresponding y values when substituted in linear equation")
print(y1)

y2=[]
for i in range(len(x)):
    y_2=p2[0]*x[i]**2+p2[1]*x[i]+p2[2]
    y2.append(y_2)
print("corresponding y values when substituted in quadratic equation")
print(y2)

y3=[]
for i in range(len(x)):
    y_3=p3[0]*x[i]**3+p3[1]*x[i]**2+p3[2]*x[i]+p3[3]
    y3.append(y_3)
print("corresponding y values when substituted in cubic equation")
print(y3)
```

```
import cmath
d=0
for i in range(len(x)):
    temp=y[i]-y1[i]
    d+=temp*temp
d=cmath.sqrt(d)/cmath.sqrt(((len(x))-2))
print("Standard error of point regression related to linear equation is "+str(d))

e=0
for i in range(len(x)):
    temp=y[i]-y2[i]
 (variable) e: complex
e=cmath.sqrt(e)/cmath.sqrt(((len(x))-2))
print("Standard error of point estimate related to parabolic equation is "+str(e))

f=0
for i in range(len(x)):
    temp=y[i]-y3[i]
    f+=temp*temp
f=cmath.sqrt(f)/cmath.sqrt(((len(x))-2))
print("Standard error of point estimate related to  cubic equation is "+str(f))
```

Example I: The programme is applied to the following calibration data. The observed absorbance values against the known concentration are furnished below.

Concentration (mcg/ml)	Absorbance
2	0.104
4	0.208
6	0.312
8	0.416
10	0.520

Result: The coefficients generated from the programme and the observed linear, quadratic and cubic equations are furnished below. The numerical coefficients of the linear regression are shown below.

[0.052 -2.73]

Corresponding linear equation is 0.052x -2.73.

The numerical coefficients corresponding to the quadratic equation are given below.

[-3.55 0.052 -1.22]

Corresponding parabolic equation is -3.55 x^2+ 0.052 x -1.22

Coefficients of the third-degree cubic equation are shown below

[7.63 -5.14 0.052 2.01]

Corresponding cubic equation is 7.63 x^3 -5.14 x^2+ 0.052 x+ 2.01

Corresponding predicted y values when substituted in linear equation

[0.10, 0.20, 0.31, 0.41, 0.51]

Corresponding y values when substituted in quadratic equation

[0.10, 0.208, 0.312, 0.416, 0.52]

Corresponding y values when substituted in cubic equation

[0.10, 0.20, 0.31, 0.42, 0.52]

Standard error of point regression related to linear equation is (2.59)

Standard error of point estimate related to parabolic equation is (7.12)

Standard error of point estimate related to cubic equation is (5.39)

Example II: The blend uniformity observed when the mixing equipment is operated for different time periods is furnished below.

Blending Time (min)	Blend Uniformity (%)
2	64
4	86
6	94
8	98
10	95

Result: The numerical coefficients generated from the programme and the related equations are given below.

Coefficients of the linear regression are shown below

[3.7 65.2]

Corresponding linear equation is 3.7x+ 65.2

The numerical coefficients related to the quadratic equation are

[-0.96 15.27 38.2]

Corresponding parabolic equation is -0.96 x^2+ 15.27 x+ 38.2

Coefficients of the third degree cubic equation are shown below

[0.07 -2.27 22.1528.4]

Corresponding cubic equation is 0.07 x^3 -2.27 x^2+ 22.15 x+ 28.4

Corresponding y values when substituted in linear equation

[72.60, 80.0, 87.40, 94.80, 102.20]

Corresponding y values when substituted in quadratic equation

[64.89, 83.85, 95.11, 98.65, 94.48]

Corresponding y values when substituted in cubic equation

[64.19, 85.25, 95.11, 97.25, 95.19]

Standard error of point regression related to linear equation is (8.48)

Standard error of point estimate related to parabolic equation is (1.56)

Standard error of point estimate related to cubic equation is (0.90)

RESULTS & DISCUSSION

The observed data is analysed by the proposed python programme. It's able to solve different mathematical equations such as linear, cubic and quadratic models. The coefficients of various equations were noted from the python programme and the related equations were generated. The proposed programme will predict the response corresponding with the varied input parameters. This programme analyse the data by employing the linear, cubic and quadratic models. The statistical parameter standard error of point estimate is usually employed to check the accuracy of the model and it's calculated by using the following formulae.

$$\sigma est = \sqrt{\Sigma\left(y - \bar{y}\right)^2 / n}$$

Where:
Y = The observed value

$\bar{y}$ = The predicted value

n = The total number of observations

If the measured data represents an entire population, then the average is calculated by dividing with n, the number of data points. However, if the data is a true representative of a population sample, then the denominator is n-2. Most of the pharmaceutical operations handle samples and hence the python programme was written with the denominator n-2. The proposed python programme will provide the standard error of point estimate and the model offering the lowest standard error is considered as the best fitted model.

CONCLUSION

The standard error is an essential indicator of how precise an estimate of the sample statistic's population parameter is. The low standard error of point estimate indicates that the selected model is best fitted model. In first example, low standard error is observed for linear equation and cubic model for second example. So it's concluded that the proposed python programme is able to indicate the best fitted equation for the observed data and suitable for the analysis and extrapolation of the experimental data.

REFERENCES

1. R. Huisman, H.V. Van Kamp, J.W. Weyland, D.A. Doornbos, G.K. Bolhuis and C.F. Lerk. Development and optimization of pharmaceutical formulations using a simplex lattice design. Pharmaceutisch Weekblad. 984: 6(5): 185-194.
2. G.A. Gonzalez. Optimization of pharmaceutical formulations based on response-surface experimental designs. International Journal of Pharmaceutics. 1993; 97(1-3): 149-159.
3. T.P.Shailesh, D.P.Laxmanbhai and N.P. Chhaganbhai. Floating matrix tablets of domperidone: formulation and optimization using simplex lattice design. Thai Journal of Pharmaceutical Sciences. 2009; 33: 113-122.
4. P. Meenakshi, S. Farhatjahan, P. Vandana and S. Naazneen. Optimization of glipizide floating matrix tablet using simplex lattice design. Indian Journal of Pharmaceutical Sciences. 2021; 83(2): 297-306.

Cite this article as:
T. Bhanuteja, A. Lakshmana Rao, T.E.G.K. Murthy, T. Pallavi. Prediction of Linearity and Non-Linearity in Pharmaceutical Optimization Studies with Python. International Journal of Research in AYUSH and Pharmaceutical Sciences, 2022;6(3):610-613.
https://doi.org/10.47070/ijraps.v6i3.126
Source of support: Nil, Conflict of interest: None Declared

***Address for correspondence**
Prof. A. Lakshmana Rao
Principal,
V. V. Institute of Pharmaceutical Sciences,
Gudlavalleru Post, Krishna District, A.P.
Mobile: 9848779133
Email: dralrao@gmail.com