# 公式的简单模拟以及数据集生成

## 身体免疫力

上述概率可以设计为100人中有37人身体免疫力是下降的；

但是最后我们针对的是个人的指导方案，的函数应该是白细胞、血小板和淋巴细胞的数值。

决定使用血检中各项的百分比作为变量。

中性粒细胞总数：1.8-6.3亿/升。

血小板总数：125-350亿/升 。

淋巴细胞总数：1.1-3.2亿/升。

N 是中性粒细胞总数，P 是血小板总数，L 是淋巴细胞总数

每个变量都除以其正常值的下限，然后相加，得到一个免疫力指数。如果所有变量都在正常范围内，那么免疫力指数应该在 3 到 9 之间。如果有任何一个变量低于正常范围，那么免疫力指数会降低；如果有任何一个变量高于正常范围，那么免疫力指数会增加。

## 2.治疗副反应（身体不适程度）

副反应这个函数应该是没什么问题了，使用了权值代表不同情况的严重程度。

定义治疗副反应的函数（）为一个介于0～1之间的指数，表示治疗后患者的不适程度，与各种副反应出现的概率成正比，初始值为0。

为每个副反应指定一个权重（W），表示该副反应对患者不适程度的影响大小，可以根据严重性、持续时间、可逆性等因素来确定，权重越大，影响越大。例如，肺炎的权重可能比皮疹的权重高。

使用加权平均法来计算治疗副反应的函数值，即F = (W1×P1 + W2×P2 + … + Wn×Pn) / (W1 + W2 + … + Wn)，其中P是各种副反应出现的概率，W是各种副反应的权重，n是副反应的种类数。

例如，根据标准组吉非替尼治疗后有以下副反应出现的概率和权重：

| 副反应 | 概率 | 权重 |
| --- | --- | --- |
| 白细胞下降 | 0.22 | 0.8 |
| 血小板下降 | 0.227 | 0.7 |
| 肝细胞损伤 | 0.82 | 0.9 |
| 肾脏损伤 | 0.28 | 0.9 |
| 呕吐不适 | 0.087 | 0.5 |
| 食欲匮乏 | 0.133 | 0.4 |
| 腹泻 | 0.387 | 0.5 |
| 皮疹 | 0.82 | 0.3 |
| 甲沟炎 | 0.387 | 0.2 |
| 心律失常 | 0 | 1 |
| 肺炎 | 0.013 | 1 |

那么：

治疗副反应的函数值 = (0.8×0.22 + 0.7×0.227 + … + 1×0.013) / (0.8 + 0.7 + … +1) = (1.8741) / (6.8) = 0.28

也就是说，治疗后患者的不适程度为28%。

### 2.1 8.1号

问题：治疗副反应的权重应该如何确定呢？

治疗副反应，出现某种不良反应，那么该变量的值就是1，再乘其对应的权值。

|  |  |
| --- | --- |
| 副反应 | 权重 |
| 中性粒细胞数目下降n | 0.2 |
| 血小板数目下降p | 0.2 |
| AST/ALT（[天冬氨酸转氨酶（AST）](https://www.cancertherapyadvisor.com/home/decision-support-in-medicine/critical-care-medicine/elevated-liver-enzymes/" \t "_blank)**[和](https://www.cancertherapyadvisor.com/home/decision-support-in-medicine/critical-care-medicine/elevated-liver-enzymes/" \t "_blank)**[丙氨酸转氨酶（ALT）](https://www.cancertherapyadvisor.com/home/decision-support-in-medicine/critical-care-medicine/elevated-liver-enzymes/" \t "_blank)）上升 肝脏损伤时会渗入血液 | 0.1  0.1 |
| 肌酐水平升高（肾脏损伤引起）c | 0.2 |
| 肺炎p | 0.2 |

## 3.肿瘤反馈治疗效果

老师回答我说：“肿瘤负荷等数据在这个数据集中缺失。这个只能模拟了，但是还是有一定的依据的，比如表格里面提到转移的问题，具有转移的患者肿瘤负荷应该比较大。”。

我看了一下脑转移和脊髓转移的数据，是转移的患者在总患者中的比例。

“肿瘤反馈治疗效果应该是指肿瘤最大直径的减少和肿瘤标志物的变化等等”。

### 8.1

设肿瘤标志物的变化为，肿瘤最大直径的变化为，肿瘤的体积变化

则有函数

## 4.肿瘤负荷

肿瘤负荷B可以由肿瘤总体积、离散程度（多个肿瘤之间的距离）、肿瘤纯度（是否有多种变异）构成的。总的来说，总体积越大，肿瘤负荷越大；等等。

肿瘤总体积，离散程度，肿瘤纯度。

肿瘤负荷：

## 5.G函数

“G函数最开始是根据一个一般化的参数，根据患者个性化数据不断迭代更新的”。

## 6.模型

### 8/30想法：

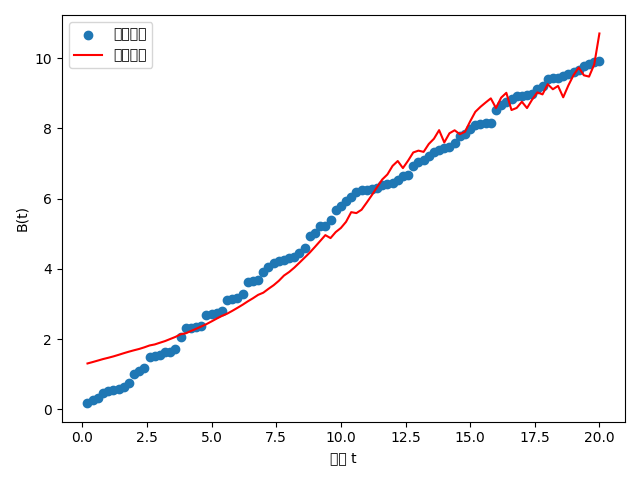
因为已经有模型了，不同的是每个患者的数值，根据每个患者的B0,ua,uf,ut的数值，来拟合出一条每个患者独自的肿瘤负荷预测曲线，然后肿瘤负荷根据指南有一个治疗临界点，这个临界点对应的时间为t1，t1-t0 = △t，△t就是患者单独地治疗间隔。

现在的问题，患者的肿瘤负荷肯定是逐渐增大的，至少比B0大，我自己胡乱生成的数据带进去可能会出现负数，如何调整G函数的形式和u的参数是个难题。

在提供的这个模型中，B(t)的变化是根据

这个因子来决定的，这个r如果过大的话，有可能会导致B0-r\*e^rt为负数。所以我需要一个在r的每个因子上添加参数，使得分母不会成为负数。

想要拟合B关于t的函数，就需要数据中含有时间点，若刚治疗完，此时对应的时间就是0。为此我想出的办法是将excle数据集排序，将肿瘤负荷B按照从小到大进行排序，那么对应的时间应该也是从小到大，选择了0到20的时间节点，实际上0对应着刚得癌症或者刚治疗完癌症，正常一个治疗周期是20天，所以得到的时间应该加20，ua身体免疫力，免疫力值越高对应肿瘤负荷越小，因此从大到小，ut副反应从小到大，uf反馈从小到大。



import numpy as np  
from scipy.optimize import curve\_fit  
import matplotlib.pyplot as plt  
import pandas as pd  
from functools import partial  
  
data1 = pd.read\_excel*(*"Ua\_dataset.xlsx"*)*data2 = pd.read\_excel*(*"Uf\_dataset.xlsx"*)*data3 = pd.read\_excel*(*"Ut\_dataset.xlsx"*)*data4 = pd.read\_excel*(*"B\_dataset.xlsx"*)*ua\_data = data1*[*'u\_a'*]*.values  
uf\_data = data2*[*'u\_f'*]*.values  
ut\_data = data3*[*'u\_t'*]*.values  
B\_data = data4*[*'肿瘤负荷'*]*.values  
t\_data = data4*[*'时间'*]*.values  
  
# 你的B(t)函数形式  
def B*(*t, a, b, c, d, C, ua, ut, uf*)*:  
 return np.exp*((*a \* ua + b \* ut + c \* uf*)* \* t + 0.5 \* d \* t\*\*2 + C*)*# 初始参数估计值（根据你的数据来设置）  
initial\_guess = *(*1.0, 1.0, 1.0, 1.0, 0.0*)*# 创建部分函数，将额外参数固定  
partial\_B = partial*(*B, ua=ua\_data, ut=ut\_data, uf=uf\_data*)*# 进行非线性拟合  
params, covariance = curve\_fit*(*partial\_B, t\_data, B\_data, p0=initial\_guess, maxfev=2000*)*# 拟合结果  
a\_fit, b\_fit, c\_fit, d\_fit, C\_fit = params  
  
# 输出拟合的参数  
print*(*f"a: *{*a\_fit*}*"*)*print*(*f"b: *{*b\_fit*}*"*)*print*(*f"c: *{*c\_fit*}*"*)*print*(*f"d: *{*d\_fit*}*"*)*print*(*f"C: *{*C\_fit*}*"*)*# 绘制原始数据和拟合曲线  
plt.scatter*(*t\_data, B\_data, label='原始数据'*)*plt.plot*(*t\_data, B*(*t\_data, a\_fit, b\_fit, c\_fit, d\_fit, C\_fit, ua\_data, ut\_data, uf\_data*)*, 'r', label='拟合曲线'*)*plt.xlabel*(*'时间 t'*)*plt.ylabel*(*'B(t)'*)*plt.legend*()*plt.show*()*

得到结果：

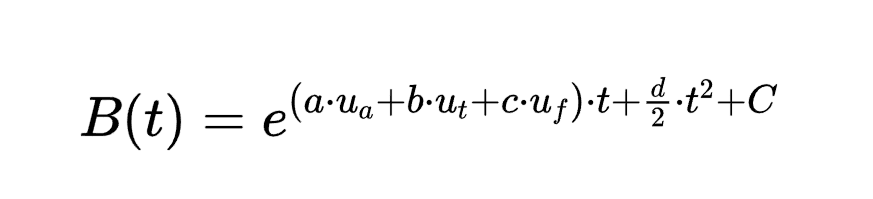
a: 0.016951704805181954

b: 0.013840630351633923

c: -0.0016788534894377886

d: 0.010161325444677469

C: 0.23916707589583788



目前想到的步骤是先把G函数形式想出来，比如线性、多项式，然后再进行参数拟合就是上述过程，再进行预测，将δt求出来。

