·论著·方法学研究·

广义线性模型在Python中的实现

罗晨曦，张清煜，李湘莹，李培政，王 静，马 露

武汉大学公共卫生学院（武汉 430071）

【摘要】**目的** 探索广义线性模型（generalized linear model，GLM）在 Python 软件中的实现方法，并比较其与其他常用统计软件在算法过程和结果方面的异同。**方法** 分别利用 Python 软件 statsmodles 库中的 GLM 函数、Logit 和 Poisson 函数，R 软件 GLM 函数，SAS 的 PROC GENMOD 过程步，对二项分布和泊松分布的数据集进行分析，比较三种软件的算法过程和分析结果。**结果** 三种软件构建 GLM 的逻辑相似，但在代码实现和模型拟合方法等方面稍有区别，各软件的结果基本相同。**结论** Python 软件可采用不同的算法构建广义线性模型，并且能提供与其他主流统计软件相同的统计分析结论。

【关键词】Python；SAS；R；广义线性模型；统计分析

Implementation of generalized linear model in Python

Chen-Xi LUO, Qing-Yu ZHANG, Xiang-Ying LI, Pei-Zheng LI, Jing WANG, Lu MA

School of Public Health, Wuhan University, Wuhan 430071, China
Corresponding author: Lu MA, Email: malu@whu.edu.cn

【Abstract】**Objective** To explore the implementation method of generalized linear model in Python software, and compare its process and results with other common statistical software. **Method** Using GLM function, Logit and Poisson function in statsmodles library of Python software, GLM function in R software and PROC GENMOD procedure step of SAS, the binomial distribution and Poisson distribution data sets were analyzed, and the process and analysis results of the three software algorithms were compared. **Results** The logic of the three kinds of software to construct the generalized linear model is similar, but the software is slightly different in code implementation, model fitting methods and other aspects, and the results of the software are basically the same. **Conclusion** Python softwares can use different algorithms to build generalized linear models, and can provide the same statistical analysis conclusions as other mainstream statistical softwares.

【Keywords】Python; SAS; R; Generalized linear model; Statistical analysis

广义线性模型（generalized linear model，GLM）是一般线性模型的扩展，通过连接函数建立因变量的数学期望值与线性组合的预测变量之间的关系，可实现非正态因变量以及指数分布族资料统计分析的一类方法。Fisher 最早在 1919 年曾采用广义线性模型的个别特例进行统计分析，20 世纪四五十年代 Berkson Dyke 和 Patterson 提出了 Logistic 回归模型并进行了应用，1972 年 Nelder 和 Wedderburn 在一篇论文中正式引入广义线性模型一词[1]。能够进行广义线性模型拟合的

DOI: 10.12173/j.issn.1004-4337.202302030
基金项目：湖北省卫生健康委 2021—2022 年度科研项目（WJ2021F103）
通信作者：马露，博士，副教授，硕士研究生导师，Email: malu@whu.edu.cn

统计软件很多，如常见的 SAS、R 和 SPSS 等。Python 语言作为当前最流行的计算机语言之一，在爬虫、大数据管理与分析、人工智能等方面展现出了巨大优势，然而其在统计分析中的应用并不多见[2–3]。为丰富 Python 在统计分析场景中的应用，本研究分别使用 Python 3.9.13、R 4.2.1 和 SAS 9.4 拟合广义线性模型，并对语言代码和参数估计结果进行比较。

1 资料与方法

1.1 资料来源

表 1 数据来源为 R 语言 AER 包自带数据集 Affairs，本研究选择数据集中 affairs（一年来婚外情频率）为因变量，gender（性别）、age（年龄）以及 rating（对婚姻的自我评分，5 分制，1 分表示非常不幸福、5 分表示非常幸福）为自变量。将因变量 affairs 转化为二分类变量（频率大于 0 为“1”、频率为 0 不变），将自变量 gender 中“male”转化为“0”、“female”转化为“1”，对此数据进行 Logistic 回归。

表 2 数据来源为 UCLA 统计学研究中心提供的公开数据集 poisson_sim，可通过访问链接 https://stats.idre.ucla.edu/stat/data/poisson_sim.csv 获取，本研究选择数据集中 num_awards（该学生在高中期间获得的奖项数量）为因变量，prog（该学生所属的课程类型，1 表示普通课程、2 表示学术课程、3 表示职业课程）和 math（该学生的数学成绩）为自变量。在分析数据之前，分别使用三种软件绘制因变量 num_awards 的频数分布图，观察是否服从泊松分布，并进行 Kolmogorov–Smirnov 检验，若得到 P 值均大于 0.05，可认为因变量服从泊松分布，对此数据进行 Poisson 回归。

表1 Affairs数据集

Table 1. Affairs data set

ID	affairs	gender	age	rating
1	0	0	37	4
2	1	0	27	4
3	1	1	27	5
4	0	0	57	3
5	0	0	22	5
…	…	…	…	…
601	1	1	32	5

表2 poisson_sim数据集

Table 2. poisson_sim data set

ID	num_awards	prog	math
1	0	3	40
2	0	3	33
3	0	2	48
4	1	2	41
5	1	2	43
…	…	…	…
200	1	2	75

1.2 模型构建

广义线性模型作为一般线性模型的推广，其基本表示形式为：

$$g(\mu)=\beta_0+\beta_1X_1+\beta_2X_2+\cdots+\beta_nX_n+\varepsilon$$

利用极大似然估计（maximum likelihood estimation）来估计参数[4]，引入了连接函数，克服了一般线性模型的缺点，使得因变量不再仅适用于正态分布。

Logistic 回归适用于因变量为二分类变量的情况，模型假设因变量 Y 服从二项分布，线性模型的拟合形式为：

$$log_e[\frac{\pi}{1-\pi}]=\beta_0+\sum_{j=1}^{p}\beta_jX_j$$

其中 $\pi=\mu Y$ 为条件均值（即给定一系列 X 的值时 Y =1 的概率），$(\pi/1-\pi)$ 为 Y =1 时的优势比，$1\log(\pi/1-\pi)$ 为对数优势比。在本例中，$1\log(\pi/1-\pi)$ 为连接函数，概率分布为二项分布，因变量为是否发生出轨事件，假设模型整体是有意义的。

Poisson 回归适用于在给定时间内因变量为事件发生数目的情况，模型假设因变量 Y 服从泊松分布，线性模型的拟合形式为：

$$log_e(\lambda)=\beta_0+\sum_{j=1}^{p}\beta_jX_j$$

其中是 λ 是 Y 的均值（也等于方差）。在本例中，连接函数为 $1\log(\lambda)$，概率分布为泊松分布，因变量为在高中期间获得的奖项数量，假设模型整体是有意义的。

1.3 模型比较

Python 分别使用 statsmodels 库中的 GLM 函数、Logit 与 Poisson 函数，R 软件采用 GLM 函数，SAS 采用 PROC GENMOD 过程步进行 Logistic 回归和 Poisson 回归，并比较三种软件拟合的结果。

2 结果

在 Python 软件 statsmodels 库中可以通过使用 GLM 函数实现 Logistic 和 Poisson 回归，此时需要在 GLM 函数中指定分布族，另外，也可不指定分布族，而直接使用 Logit 和 Poisson 函数实现建模。在使用 Python 实现 Logistic 回归时，将 Affairs 数据集中因变量 affairs 赋值给 Y，再将自变量 gender、age、rating 赋值给 X，并给 X 加上一个截距项，目的是给自变量 X 加上一列常数项 1，从而在 y 轴上形成截距。否则，分析的结果中将没有截距项，使回归系数产生较大波动。

Logistic 回归命令如下：

```
data =pd.read_excel('D:/Affairs.xlsx')
Y=Affairs['affairs']
X= Affairs.loc[:,['gender','age', 'rating']]
X=sm.add_constant(X)
glm_binom = sm.GLM(Y,X, family=sm.families.Binomial())
res = glm_binom.fit()
print(res.summary())
```

在使用 Python 实现 Poisson 回归时，将 Poisson_sim 中因变量 num_awards 赋值给 Y，再将自变量 prog、math 赋值给 X，并给 X 加上一个截距项。

Poisson 回归具体命令如下：

```
data=pd.read_excel('D:/poisson_sim.xlsx')
Y=data['num_awards']
X=data.loc[:,['prog','math']]
X=sm.add_constant(X)
glm_poisson = sm.GLM(Y,X, family=sm.families.Poisson())
results = glm_poisson.fit()
print(results.summary())
```

Python 中也可直接使用 Logit 和 Poisson 函数进行分析，此时不需要在函数中指定分布族，但同样需要给 X 加上一列常数项。

Logistic 回归的命令为：

```
data =pd.read_excel('D:/Affairs.xlsx')
Y=Affairs['affairs']
X= Affairs.loc[:,['gender','age', 'rating']]
X=sm.add_constant(X)
logit = sm.Logit(Y,X)
result = logit.fit()
print(result.summary())
```

Poisson 回归的命令为：

```
data=pd.read_excel('D:/poisson_sim.xlsx')
Y=data['num_awards']
X=data.loc[:,['prog','math']]
X=sm.add_constant(X)
model_pos = sm.Poisson(Y,X)
results_pos = model_pos.fit()
print(results_pos.summary())
```

在 R 中，实现 Logistic 回归和 Poisson 回归使用的都是 GLM 函数，Logistic 回归对应的是二项分布，Poisson 回归对应的是泊松分布，在 GLM 函数中修改指定的分布族即可实现不同的回归。

Logistic 回归命令如下：

```
Affairs <-read_xlsx("D:/ Affairs.xlsx")
result <-glm(affairs~gender+age+rating, data = Affairs , family = binomial())
summary(result)
```

Poission 回归命令如下：

```
poisson_sim <-read_xlsx("D:/poisson_sim.xlsx")
result <- glm(num_awards ~ prog + math, data = poisson_sim, family = poisson())
summary(result)
```

与之类似，在 SAS 软件中，使用 PROC GENMOD 过程步，修改分布函数以及连接函数即可实现不同的回归。

Logistic 回归命令如下：

```
Proc genmod data= Affairs;
model affairs=gender age rating/dist=binomial link=logit type1 type3;
Run;
```

Poission 回归命令如下：

```
Proc genmod data= poisson_sim;
model num_awards = prog math /dist=poisson link=log type1 type3;
Run;
```

无论是 Logistic 回归还是 Poisson 回归，三种软件拟合的回归模型系数的绝对值都是基本相等的。在 Logistic 回归中，Python 的 GLM 函数和 Logit 拟合的结果完全相同，但 Python 和 R 的系数符号与 SAS 正好相反，这是因为在 Logistic 回归中，Python 和 R 默认对因变量等于“1”的概

率建模，而 SAS 默认对因变量等于“0”的概率建模，导致 SAS 的参数估计结果的符号与 Python 和 R 的估计结果相反，在 PROC 语句中指定 DESCENDING 选项即可改为对因变量等于“1”的概率建模。我们还观察到在 Python 和 R 中，使用的是 z 检验，而在 SAS 中使用的是 Wald χ^2 检验，在数值上 χ^2 值等于 z 值的平方 [5–7]。三种软件在默认情况下未直接输出 OR 值。在模型评价方面，Python 在默认情况下没有给出 AIC 值，R 和 SAS 给出的 AIC 值一致，回归结果见表 3。Poisson 回归结果见表 4，在 Python 中 GLM 函数和 Poisson 函数拟合的结果完全相同。

表3 Logistic回归参数估计结果

Table 3. Results of parameter estimation in Logistic regression

参数	Python			R			SAS		
	估计值	z值	$P>\|z\|$	估计值	z值	$P>\|z\|$	估计值	χ^2 值	$P(>\chi^2)$
(Intercept)	0.97	1.75	0.08	0.97	1.75	0.08	−0.97	3.07	0.08
gender	−0.25	−1.29	0.20	−0.25	−1.29	0.20	0.25	1.66	0.20
age	−0.00	−0.03	0.98	−0.00	−0.03	0.98	0.00	<0.01	0.98
rating	−0.51	−5.90	<0.01	−0.51	−5.90	<0.01	0.51	34.81	<0.01

注：Python的结果为GLM函数拟合的结果

表4 Poisson回归参数估计结果

Table 4. Results of parameter estimation in Poisson regression

参数	Python			R			SAS		
	估计值	z 值	$P>\|z\|$	估计值	z 值	$P>\|z\|$	估计值	χ^2 值	$P(>\chi^2)$
(Intercept)	−5.58	−8.24	< 0.01	−5.58	−8.24	< 0.01	−5.58	67.92	< 0.01
prog	0.12	0.76	0.45	0.12	0.76	0.45	0.12	0.57	0.45
math	0.09	8.98	< 0.01	0.09	8.98	< 0.01	0.09	80.71	< 0.01

注：Python的结果为GLM函数拟合的结果

3 讨论

三种软件在语言方面，模型思想是相似的，都需要在函数内指定因变量和自变量以及分布族，Python 需要安装相应的库才能进行回归分析，R 语言则使用默认加载的 stats 包进行分析，SAS 可直接调用过程步进行分析。在 Python 和 R 语言中，有着相似的 GLM 函数可供直接使用来建立广义线性模型，但二者对参数估计的方法是不相同的，Python 使用的是迭代加权最小二乘法（Iterative Reweight Least Square，IRLS），R 语言使用的是极大似然估计法 [6–7]。对于广义线性模型的参数估计方法，在线性模型的假设下，最小二乘法和极大似然法都可用于参数的求解，但在广义线性模型中，由于无法写出二乘形式的优化函数，因此一般根据分布信息利用极大似然估计来估计参数。模型的极大似然估计过程中，某些变量的计算可能存在过于复杂的情况，在实际求解中可以根据大数定律以其期望代替这些变量，这样模型的极大似然估计可以近似得到估计量的求解迭代公式。本研究在 Python 实现广义线性模型的过程中，GLM 函数使用迭代加权最小二乘法进行参数估计，Logit 和 Poisson 函数则使用极大似然估计法来估计参数，虽然方法不同，但拟合出来的参数估计结果都是相同的。

本研究发现 Python 软件可采用不同的算法构建广义线性模型，将结果与 R 软件和 SAS 进行比较之后，可以认为 Python 软件能提供客观正确的统计结论。虽然相较于专门的统计分析软件，Python 在构建统计模型时编程稍繁琐，但其实现回归的基本思想和 R 以及 SAS 是一致的。另外，Python 语言具有常用统计软件所不具备的优势：一方面，Python 通用性好，作为一种计算机通用语言，具有一定计算机语言基础的人员对 Python 代码比较容易理解；另一方面，Python 具有很好的可移植性和拓展性，由于其编程的底层语言是采用 C 语言编写的，而很多第三方库都是借助 C 语言进行编写，作为混合式语言编程开发而言，具有很强的适用性和可嵌入性 [8–10]。随着信息化快速发展，不断拓展 Python 的应用场景，必将有

益于从数据收集、多源数据融合和管理到人工智能化分析全过程的实现，从而促进在计算机与医学等相关领域的交叉融合。

参考文献

1 陈希孺 . 广义线性模型 (一)[J]. 数理统计与管理 , 2002, 21(5): 54–61. [Chen XR. Generalized linear model(I)[J]. Journal of Applied Statistics and Management, 2002, 21(5): 54–61.] DOI: 10.3969/j.issn.1002–1566.2002.05.013.

2 陈希孺 . 广义线性模型 (五)[J]. 数理统计与管理 , 2003(3): 56–63. [Chen XR. Generalized linear model (V)[J]. Journal of Applied Statistics and Management. 2003, 22(3): 56–63.] DOI: 10.3969/j.issn.1002–1566.2003.03.014.

3 Perkel JM. Programming: pick up python[J]. Nature, 2015, 518(7537): 125–126. DOI: 10.1038/518125a.

4 顾刘金 , 陈钢 . 广义线性模型及 SPSS 软件实现 [J]. 预防医学 , 2020, 32(7): 755–756. [Gu LJ, Chen G. Generalized linear model and SPSS software implementation[J]. Journal of Preventive Medicine, 2020, 32(7): 755–756.] DOI: 10.19485/j.cnki.issn2096–5087.2020.07.028.

5 Moore LM. Statistical modelling in GLIM[J]. Technometrics, 1991, 33: 109–110. DOI: 10.1080/00401706.1991.10484778.

6 Mccullagh P, Nelder JA. Generalized linear models. Monographs on statistics and applied probability No 37[M]. 1989. https://xueshu.baidu.com/usercenter/paper/show?paperid=4001f961ce14e854685ae652368244ad&site=xueshu_se&hitarticle=1.

7 Kyung M, Gill J, Ghosh M, et al. Jeff Gill. Generalized linear models: a unified approach. 2000, sage QASS series. Ken Meier and Jeff Gill. What works: a new approach to program and policy analysis. 2000. https://xueshu.baidu.com/usercenter/paper/show?paperid=ccda165d8e9e175e01b8241a32cf3c3b&site=xueshu_se.

8 焦奎壮 , 马煦晰 , 马小茜 , 等 . 广义估计方程与混合线性模型在 Python 中的实现 [J]. 医学新知 , 2022, 32(05): 333–338. [Jiao KZ, Ma XX, Ma XQ, et al. Implementation of generalized estimating equations and mixed linear models in Python[J]. New Medicine, 2022, 32(5): 333–338.] DOI: 10.12173/j.issn.1004–5511.202203007.

9 刘正 . 积木式 python 编程系统的研究与实现 [D]. 北京邮电大学 , 2020. [Liu Z. Research and implementation of Block–Python programming system[D]. Beijing University of Posts and Telecommunications, 2020.] DOI: 10.26969/d.cnki.gbydu.2020.002202.

10 卢绍兵 . 基于 Python 的混合语言编程及其实现研究 [J]. 科技资讯 , 2022, 20(14): 31–33. [Lu SB. Reseach on mixed language programming based on Python and its implementation[J]. Science & Technology Information, 2022, 20(14): 31–33.] DOI: 10.16661/j.cnki.1672–3791.2201–5042–7709.

收稿日期：2023 年 02 月 08 日 修回日期：2023 年 02 月 26 日
本文编辑：李 阳 黄 笛

引用本文：罗晨曦, 张清煜, 李湘莹, 等. 广义线性模型在Python中的实现[J]. 数理医药学杂志, 2023, 36(3): 161–165. DOI: 10.12173/j.issn.1004–4337.202302030
Luo CX, Zhang QY, Li XY, et al. Implementation of generalized linear model in Python[J]. Journal of Mathematical Medicine, 2023, 36(3): 161–165. DOI: 10.12173/j.issn.1004–4337.202302030