用统计分析对《红楼梦》进行文笔鉴赏

——以高频用字和句子长度为切入点

学生姓名：吴彬　学生学号：2017061033

目录

§1. 引言.....2
§2. 准备：数据收集与处理.....2
§3. 赏析：文笔之高频用字.....4
§4. 赏析：文笔之句子长度.....8
§5. 猜测：后 40 回是否出自曹雪芹之手.....11
§6. 参考文献.....16
§7. 附录一：《红楼梦》原文资料.....17
§8. 附录二：代码实现.....17
§9. 附录三：未在正文展示的一些结果.....33

摘要 《红楼梦》是中国古典的四大名著之一，其本身有许多值得研究的地方，甚至也留下诸如“后 40 回的作者是不是曹雪芹”的谜团。针对这个谜团，在攥写本文之前已有一批学者从统计分析的独特角度，运用了各种各样的统计方法进行了分析，得到的结果也不尽相同，有人支持后 40 回作者是曹雪芹，有人认为后 40 回作者另有其人。本文主要先从高频汉字和句子长度两方面入手，得到《红楼梦》文笔背后的一些统计学猜测。然后再单独从高频汉字的角度出发分析前 80 回与后 40 回的差异，认为后 40 回非曹雪芹所著。

第一部分，我们进行了数据的搜集和处理。在网络上获取了《红楼梦》全文的电子资源后，为了筛选得到其中曹雪芹写作的内容，我们先手动删除其前、后目录，再通过编程删除其中来自脂砚斋的回前墨、回后评，来自学者的注释部分和正文中的所有注释角标，最后剩下全 120 回的纯正文内容共 857248 字。

第二部分，我们从高频汉字和句子长度两方面对《红楼梦》的文笔进行了赏析。一方面，通过对 1-40 回和 41-80 回中所有单字在每一回的使用频数的统计和比较，发现曹雪芹在前 1-40 回和 41-80 回的高频字符种类使用习惯很相近，且前 100 高频使用字符占前 80 回的篇幅过半。再通过构建与高频汉字分布密度相关的统计量与单因素方差分析，发现至多每 5 回高频汉字的分布密度就存在显著差异。另一方面，通过对不同长度句子在 120 回中每一回出现频数的统计，发现《红楼梦》的攥写以四字骈句和六字俪句为主。通过多元线性回归分析和线性假设的显著性检验，发现一个回的总字数与长度介于 4 到 16 的句子在一个回出现的频数有很强烈的多元线性关系。

第三部分，我们单独从高频汉字的角度对《红楼梦》后 40 回的作者归属问题作了分析。我们将全书划分为 3 个板块：1-40 回，41-80 回，81-120 回，并先计算出每一回中经筛选的高频汉字出现的频率。然后分别独立地从单因素方差分析和秩和检验分析的视角，提出相应的用于作出推理的公理，对 3 个板块之间的“高频汉字在每回出现频率”的指标进行差异的显著性检验。结果 1-40 回与 41-80 回的对比结果是不存在显著差异，验证了前 80 回是曹雪芹所写的事实；1-40 回与 81-120 回、41-80 回与 81-120 回的对比结果是存在显著差异，由此从高频汉字频率的角度推测后 40 回作者并非曹雪芹。

关键字 红楼梦 高频汉字 句子长度 单因素方差分析 多元线性回归 正态拟合检验 秩和检验

§1. 引言

作为中国古典的四大名著之一，《红楼梦》是一部内涵丰富的作品，其中有诸多值得赏析的地方。而《红楼梦》所遗留下来的一些谜团更是给其蒙上一层神秘的面纱，其中就包括著名的争论：《红楼梦》后 40 回，即第 81 回到第 120 回究竟是出自曹雪芹之笔，还是另有人续之。对这个谜题，不同的人从不同的角度给出了截然相反的答案，其中就有一批人从统计分析的角度进行了考察。事实上，使用数学方法分析文学作品的事早有人为之。美国斯坦福大学的教授 Efron 和他的学生 Thisted 曾经就对莎士比亚的著作进行过相当深入的统计分析[1][2]，并指出 1985 年发现的一篇“无名氏”的诗稿（仅 9 节 429 字）确实为莎士比亚所著。而针对《红楼梦》，特别是分析前 80 回与后 40 回的差异方面，陈炳藻，陈大康等一批学者也进行了相关的统计分析。他们入手的角度不尽相同。陈炳藻专门分析名词，动词，形容词，副词，虚词的使用情况[3][4]，陈大康分析的是词，字，句的使用情况[5]，李贤平的“成书新说”中则专门分析虚字的使用情况[6]，韦博成则是选择花卉，树木，饮食，医药与诗词 5 个情景指标进行分析[7]，安鸿志则是对书中人物面对皇权态度的方面进行分析[8]。而各个学者使用的方法也不尽相同。陈炳藻使用的是相关性分析等方法[3][4]，李贤平使用的是主成分分析，典型相关分析，聚类分析等方法[6]，韦博成运用了 Fisher 精确条件检验和渐进正态检验，并且以统计学中“两个独立二项总体的等价性检验”为基本方法[7]。而各个学者得出的结论也存在迥异。陈炳藻认为前 80 回与后 40 回均为曹雪芹所作[3][4]，陈大康认为后 40 回非曹雪芹所作（但含有少量残稿）[5]，李贤平则提出《红楼梦》前 80 回是曹雪芹根据《石头记》增删而成，而后 40 回是曹家亲友搜集整理原稿加工补写而成[6]。

受以上研究的启发，本文先从《红楼梦》中的高频用字和句子长度两方面入手，简要分析《红楼梦》的文笔特点。然后从高频用字的角度简单分析前 80 回与后 40 回的差异，以此为根据作出关于后 40 回作者是否曹雪芹的推测。

§2. 准备：数据收集与处理

根据以下步骤进行数据收集与预处理。

1. 首先我们从子非网 https://www.88u4.com/469.html 下载《红楼梦》PDF 格式的电子书“《红楼梦》原.pdf”。
2. 为了方便后续的导入和处理，将电子书转换为 Docx 格式的 Word 文档“《红楼梦》原.docx”。
3. 为了方便后续的数据清洗，简单地将得到的文档中的第 1 回之前的目录和书籍出版信息，第 120 回之后的目录和鸣谢等无关正文内容的部分手动删除，并保存为“《红楼梦》删.docx”。
4. 将“《红楼梦》删.docx”导入到 python 中，使用到第三方库 docx 中的 Document( )函数。
5. 每一回一般都依次包含回前墨，正文，回后评，注释四大部分，注释又分为汉字角标的注释和数字角标的注释，且汉字角标注释位于数字角标注释之前。回前墨和回后评是脂砚斋对原小说作的批注，不属于曹雪芹所著文字；注释也是后人所注。因此每一回我们只取正文部分。具体方法是通过甄别关键字“回前墨”,“回后评”,角标依次舍去回前墨，回后评，数字角标注释，汉字角标注释部分，最后删去正文中零零星星出现的汉字角标。
6. 由于每一回前面都有一组对仗工整的 16 字标题，于是通过甄别段前关键字“第 X 回”，正式读入其后第 17 位（包括第 17 位）之后的内容。

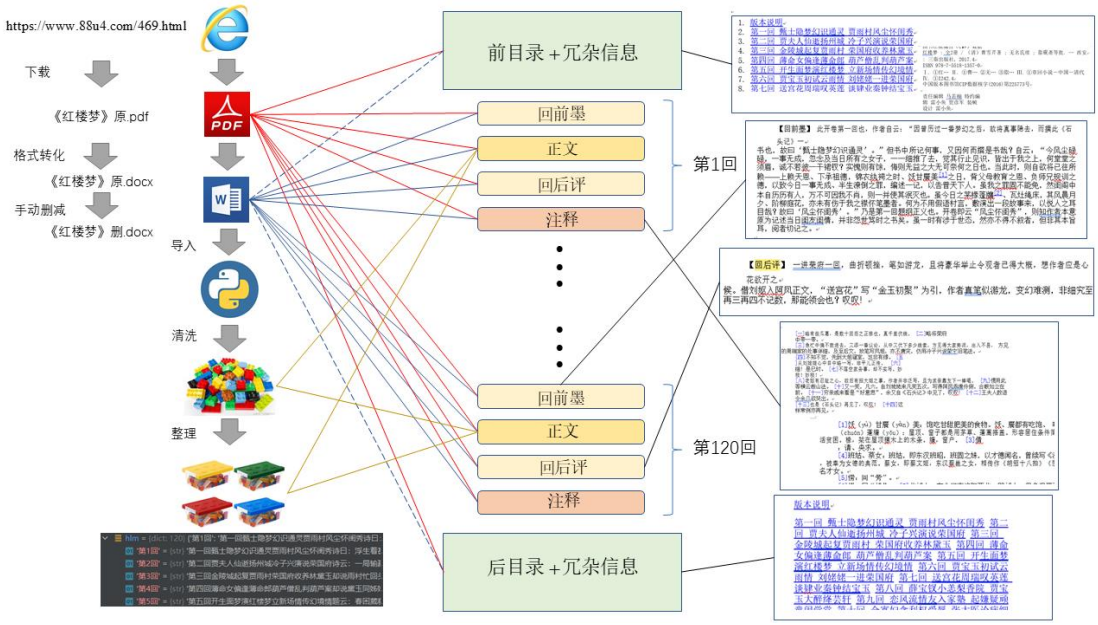

Fig 2.1 数据收集与处理的大致流程.

经统计，“《红楼梦》删.docx”中读入的总字符数为 963057，进行数据清洗后剩下的总字符数为 857248，被清洗掉的字符总数为 105809，超过 10 万字，占原总字符数的 10.99%。这一可观的比例说明了原文中“回前墨+回后评+注释”的部分不可忽视，进行数据清洗是必要的。而对清洗完后的数据分别整理成相应的 120 回，其中字符数最少的回数是第 8 回，共 3491 个字符，字符数最多的回数是第 62 回，共 11817 个字符。通过作直方图可以看到字符数在 120 回中的分配是不均匀的，但是若将 1-40 回，41-80 回，81-120 回视为三个组别，则字符数在这三个组之间的分布是比较均匀的。

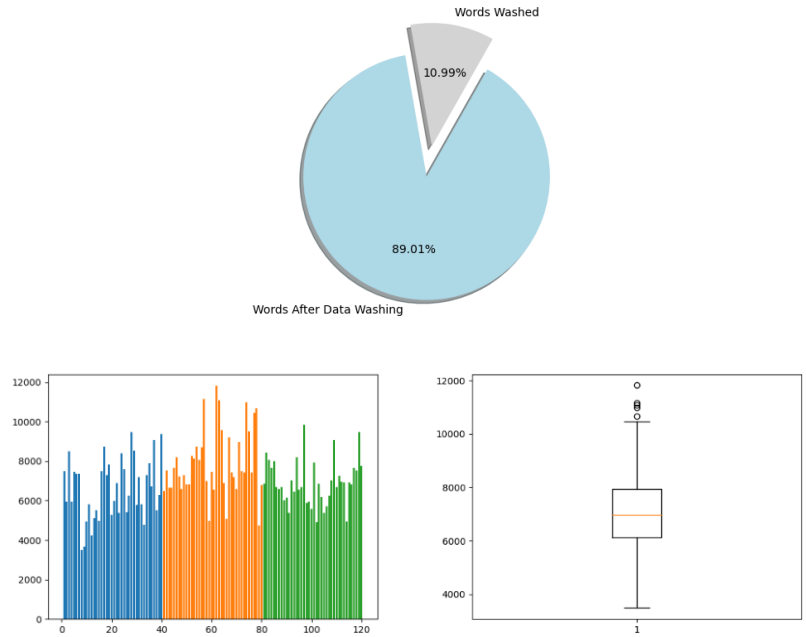

Fig 2.2 (上)清洗掉的数据 Words Washed 以及清洗完剩下的数据 Words After Data Washing 各占原数据总量的比例, (左下)对应于上图的 Words After Data Washing 部分分割成 120 回后每一回的数据总量分布直方图, (右下)左下图的箱线图展示.

§3. 赏析：文笔之高频用字

一般，基本可以肯定前 80 回是曹雪芹所著。分别统计 1 到 40 回，41 到 80 回中每个单字的使用频数，并按照频数从大到小的顺序排列，分别取出 1 到 40 回，41 到 80 回的前 100 个高频用字，得到两个集合 S_1，S_2，$|S_1|=|S_2|=100$. 经统计得到论据 3.1.

论据 3.1 $|S_1 \bigcap S_2|=93$.

$S_1 \bigcap S_2$ 的元素详见 Table 3.2 中不打星号*的 93 个字符。论据 3.1 揭示了曹雪芹在写 1 到 40 回和 41 到 80 回的过程中对部分汉字和标点符号保持着高频率使用的习惯，由此作出简单的结论 3.2.

结论 3.2 《红楼梦》中同一作者使用的高频用字的种类是比较稳定的。

本节我们想要研究《红楼梦》中出自曹雪芹之手的前 80 回中高频用字的一些规律。基于推测 3.2，我们直接在前 80 回中统计每个单字的使用频数，按照频数从大到小的顺序排列，得到前 100 个高频用字的清单。经统计，有以下的论据 3.3.

论据 3.3 前 80 回中的前 100 个高频使用字符(包括标点符号)占前 80 回篇幅的 60.05%。

基于论据 3.3，我们作出简单的结论 3.4.

结论 3.4 《红楼梦》中曹雪芹所著部分的高频用字所占的篇幅比例过半。

接下来我们想要对这 100 个高频字进行筛选，以进行下一步的研究。首先我们缩小研究对象为高频汉字，即筛去前 100 个高频字符中的所有标点符号。然后对前 100 个高频字作频数分布直方图和箱线图，发现少部分的高频汉字出现的频数是大于 5000，而绝大部分的高频汉字出现的频数介于 1000 到 5000 之间。考虑到有一些汉字在各种风格的写作，各种句式，各种场景的描写中都难免用到（例如频数大于 5000 的了，的，不，一），不是很能反映出文笔差异，可视为异常值。因此最终我们筛选得到了频数小于 5000 的那些高频汉字，共 79 个，详见 Table 3.2 中的 79 个不带标记的黑色汉字。经统计，筛选后得到的 79 个汉字占前 80 回篇幅的 18.93%.

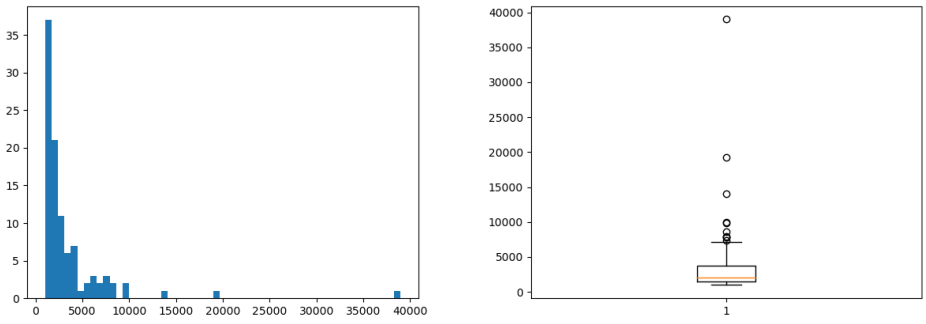

Fig 3.1 (左)前 80 回的前 100 高频字符的频数分布直方图, (右)左图对应的箱线图.

Table 3.2 前 80 回提取的前 100 个高频使用单字(标点符号),
标红色的标点符号和标绿色的单字表示弃用，打星号的单字表示非 1-40 回的前 100 高频字，
其余单字均同时是 1-40 回和 41-80 回的前 100 高频字.

，	39002	他	5401	只	3074	得	2161	要	1739	母	1455	叫	1182
。	19271	这	5367	贾	3021	日	2138	两	1711	凤	1419	什	1119
了	14039	你	5036	见	2936	出	2137	过	1697	时	1414	才	1111
的	9976	去	4144	里	2895	听	2115	事	1680	没	1393	呢*	1056
不	9894	个	4086	那	2886	头	2046	自	1622	可	1387	夫	1024
一	8586	也	4026	上	2598	、	1992	还	1581	！	1383	打*	1019
：	7949	儿	3966	便	2579	下	1986	话	1550	面	1353	罢	1018
“	7870	有	3918	好	2532	到	1974	心	1546	等	1352	无*	1012
”	7863	子	3916	家	2478	么	1960	因	1534	些	1333	样*	999
来	7393	玉	3867	太	2450	都	1878	小	1513	问	1262	想*	994
道	7202	又	3734	在	2448	知	1807	老	1485	娘*	1241		
人	6773	着	3701	姐	2394	之	1782	看	1470	奶*	1231		
我	6340	宝	3683	？	2344	回	1778	起	1463	中	1216		
是	6316	笑	3336	大	2299	二	1773	忙	1459	此	1195		
说	6181	们	3336	就	2235	如	1741	今	1458	吃	1187		

基于上面的论述，我们想要研究上面筛选得到的 79 个高频汉字在前 80 回不同回数中出现的密集程度是否有显著差异，使用的分析方法是单因素方差分析。设这 79 个高频汉字构成的集合为 W .

首先，确定因素为回数，有 80 个水平：第 1 回，第 2 回，……，第 80 回。其次，我们要确定考察的试验指标。由于应用单因素方差分析需要假设各水平下样本均来自正态总体[9]，因此我们要设计一个尽可能符合正态分布的试验指标。只根据一个高频汉字来构造指标，若该高频汉字的出现密度本身不是近似正态分布，那么这种指标的构造是困难的或带有运气的。而受启发于[9]中关于中心极限定理客观背景的一段描述

“在客观实际中有许多随机变量，它们是由大量的相互独立的随机因素的综合影响所形成的。而其中每一个别因素在总的影响中所起的作用都是微小的。这种随机变量往往近似服从正态分布”,

我们欲尝试根据全部 79 个高频汉字来构造试验指标。我们选取以下最直接简单的构造方法。在每一回均进行以下 $n(n\geq 2)$ 次独立重复实验：随机抽取一个长度为 L 的连续句段（包含标点符号，一个汉字的长度 = 一个标点符号的长度 = 1），并设该句段为序列 $S=\{s_1,s_2,\cdots,s_L\}$. 确立试验指标为

$$X:=\left|\{s_j\in W:j=1,2,\cdots,L\}\right|.$$

X 的观察值 $x\in\{0,1,\cdots,L\}$，且 x 越大，代表 79 个高频汉字出现的密集程度越大。

基于上面确定的因素和指标，我们抽象出一个单因素方差分析的模型。单因素回数 A 有 80 个水平 $A_1,A_2,\cdots,A_{80}$，A_j 对应的水平为“第 j 回”，在水平 $A_j\,(j=1,2,\cdots,80)$ 下进行 n 次独立重复试验得到样本 $X_{j1},X_{j2},\cdots X_{jn}$。设置 $L=100$，$n=50$，某一次试验得到的完整数据见附录三中的 Result 1.

视样本观察值为连续的定量数据。使用 Kolmogorov-Smirnov 检验[11]（以下简称 K-S 检验），借助 python 第三方库 scipy 中的 scipy.stats.kstest( )函数对 80 回中每一回抽样得到的结果进行正态分布拟合优度检验（该检验用的是 p 值法）。设定显著性水平为 0.05. 完整的检验结果见附录三中的 Result 2. 结果显示，只有第 21 回，第 26 回抽样得到的样本来自的总

体不能视为正态总体，且第 21 回计算得到的统计量 $D=0.158$ 与 p 值 $p=0.147$ 很相近。因此指标 X 在几乎所有的回数中都可视为来自正态分布。

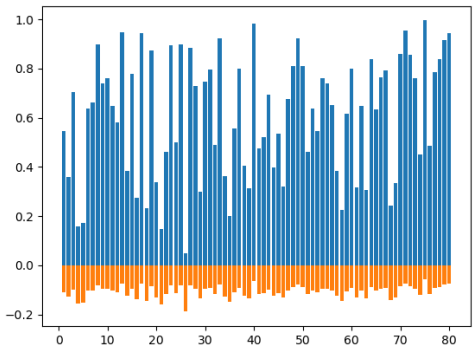

Fig 3.3 对 80 回抽样数据的正态分布 K-S 拟合优度检验结果图，

横轴对应 1 到 80 回，纵轴方向，蓝色部分表示统计量 D 的值，橙色部分表示 p 值.

基于此，我们删去水平 A_{21},A_{26}，并且假定剩下的各个水平 A_j 下的样本 $X_{1j},X_{2j},\cdots,X_{nj}$ 来自具有相同方差 σ^2，均值分别为 μ_j 的正态总体 $N\left(\mu_j,\sigma^2\right)$，$\mu_j$ 与 σ^2 未知。且设不同水平 A_j 下的样本之间相互独立。设置显著性水平为 0.05. 我们的任务是检验 78 个总体

$$\left\{N\left(\mu_j,\sigma^2\right): j=1,2,\cdots,80 \ \ \& \ \ j\neq 21,26\right\}$$

的均值是否相等，即检验假设

$$H_0:\mu_1=\cdots=\mu_{20}=\mu_{22}=\cdots=\mu_{25}=\mu_{27}=\cdots=\mu_{80},$$

$$H_1:\mu_j \text{ 不全相等，} \ j\neq 21,\ 26.$$

使用附录二 myfunctions.py 中的自定义函数 my_anova1( )得到如下的方差分析表。

Table 3.4 对 80 个回数高频字出现密集程度的方差分析表.

方差来源	平方和	自由度	均方	F比	F临界值(显著性水平0.05)
因素	35066.25769	77	455.4059441	13.0413	1.28293743
误差	133465.3	3822	34.92027734	—	—
总和	168531.5577	3899	—	—	—

由 python 第三方库 scipy 中的 scipy.stats.f.isf( )函数计算得

$$F_{0.05}(77,3822)=1.283<13.041,$$

故在显著性水平 0.05 下拒绝 H_0，认为 80 回中至少存在两回的高频字密集程度有显著差异。

更进一步，在因素和试验指标保持不变的基础上，将因素的水平数缩减为每相邻 5 回：1 到 5 回，6 到 10 回，……，76 到 80 回。遍历 $i \in \{0,2,\cdots,15\}$，假定水平

$$A_{5\times i+1},\ A_{5\times i+2},\ A_{5\times i+3},\ A_{5\times i+4},\ A_{5\times i+5}$$

下的样本

$$X_{1,5\times i+1},\ X_{2,5\times i+2},\ X_{3,5\times i+3},\ X_{4,5\times i+4},\ X_{5,5\times i+5}$$

来自具有相同方差 σ^2，均值分别为 $\mu_{5\times i+1},\ \mu_{5\times i+2},\ \mu_{5\times i+3},\ \mu_{5\times i+4},\ \mu_{5\times i+5}$ 的正态总体。且设不同水平下的样本之间相互独立。设置显著性水平为 0.05. 对某个特定的 i，我们的任务是检验 5 个总体

$$N\left(\mu_{5\times i+1},\sigma^2\right), N\left(\mu_{5\times i+2},\sigma^2\right), N\left(\mu_{5\times i+3},\sigma^2\right), N\left(\mu_{5\times i+4},\sigma^2\right), N\left(\mu_{5\times i+5},\sigma^2\right)$$

的均值是否相等，即检验假设

$$H_0: \mu_{5\times i+1} = \mu_{5\times i+2} = \mu_{5\times i+3} = \mu_{5\times i+4} = \mu_{5\times i+5},$$

$$H_1: \mu_{5\times i+1},\ \mu_{5\times i+2},\ \mu_{5\times i+3},\ \mu_{5\times i+4},\ \mu_{5\times i+5} \text{ 不全相等.}$$

使用附录二 myfunctions.py 中的自定义函数 my_anova1( )得到一系列方差分析表（见附录三中的 Result 3）。结果表明，只有水平为第 71 到 75 回的实验中的 F 统计量的值 1.245 小于显著性水平 $\alpha = 0.05$ 下的临界值 $F_{0.05}(4,245) = 2.408$，没有落入拒绝域；其他的水平数为 5 的实验中 F 统计量的值均大于 2.408，落入了拒绝域中。故在显著性水平 0.05 下，对除了第 71 到第 75 回之外的其他情况拒绝 H_0，认为对应的 5 个回数中至少存在两回的高频字密集程度有显著差异。

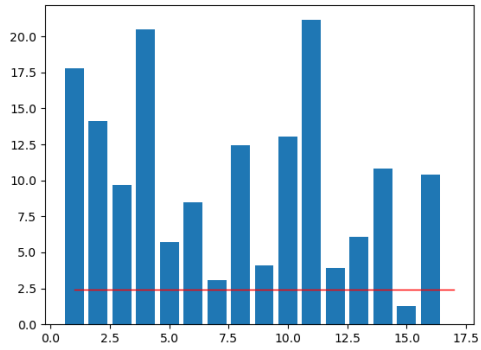

Fig 3.5 每相邻 5 回的单因素方差分析得到的 F 统计量的值与临界值 $F_{0.05}(4,245)$ 的关系图.

论据 3.5 曹雪芹所著的前 80 回中几乎每相邻 5 回的高频汉字密集程度存在显著差异。

结论 3.6 曹雪芹的文笔风格在 5 个回数之内可能会发生显著的变化，文学功底深厚；《红楼梦》前 80 回中不同回的内容刻画，情感表现等可能存在显著的差异。

§4. 赏析：文笔之句子长度

本节中，我们将研究对象转移到《红楼梦》中的句子长度。首先，统计全书 120 回每一回中各种长度的句子出现的频数（每一回的 16 字对仗标题不计入内）。句子长度的分割标准是：出现逗号“，”、句号“。”、问号“？”、感叹号“！”、冒号“：”、顿号“、”、分号“；”、省略号“……”就算作一句，后续内容算入另一句；其他的标点符号算作汉字。然后分别算出句长为 1，2，……，20 的句子在全书中出现的总频数并绘制成频数分布直方图。得到以下论据。

论据 4.1 句长为 4 的句子出现的频数最大，为 16316；句长为 6 的句子次之，出现的频数为 15505. 句长为 4 和 6 的句子的频数占句长为 1 到 20 的句子的频数总和的 30.12%。且频数分布直方图呈现右偏状态。

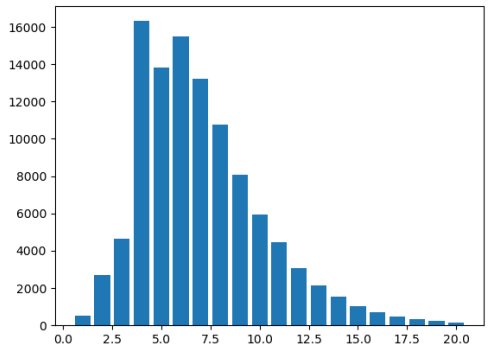

Fig 4.1 句子长度为 1 到 20 的句子在《红楼梦》全书中的频数分布直方图.

根据百度百科的词条[12]，“骈句”是
结构相似、内容相关、行文相邻、字数相等的两句话，跟对偶相似，只是不像对偶那样在音韵上有严格的要求。骈句也有工整和不工整之分，不工整的骈句在结构和字数上也可能不完全合乎要求。
“散句”则是相对于骈句而言。基于该词条和论据 4.1，可作出以下简单的结论。

结论 4.2 《红楼梦》全书的句子大多短促有力，读起来郎朗上口，很有可能是一部骈散结合的典籍。

此外，将全书 120 回划分为 3 个部分：1 到 40 回，41 到 80 回，81 到 120 回。在每一部分中分别统计句长为 1 到 20 的句子在当前部分出现的频数，并绘制成折线图。得到以下论据。

论据 4.3 1 到 40 回，41 到 80 回，81 到 120 回折线的位置和变化规律大体一致，折线间差距最明显的地方在句长为 4,5,6 的部分。

结论 4.4 全书上中下三个部分中不同长度句子的分布比例保持一个比较稳定的状态。三个部分对骈句和俪句[12]的使用强度在数据统计上有显著的差别。

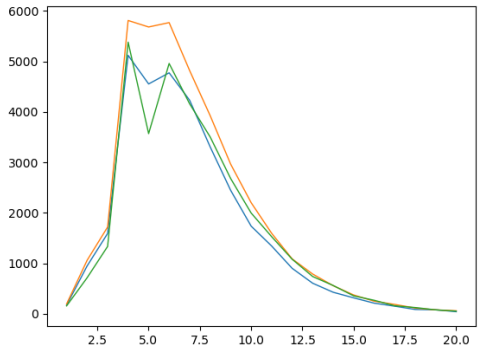

Fig 4.2　1 到 40 回，41 到 80 回，81 到 120 回各自的句子长度与频数的折线图.
蓝色对应 1 到 40 回，黄色对应 41 到 80 回，绿色对应 81 到 120 回.

进一步地，我们想要研究各种长度句子的频数与一个回（限制在前 80 回）的字数是否存在某种相关关系。基于此，我们拟采用多元线性回归分析[9]的方法。首先，由 Fig4.1 的直方图和 Fig 4.2 的折线图，发现长度特别短或者长度特别长的句子出现的频数都很低，由“句长×频数”的公式，它们对一个回的字数的贡献作用也微不足道，因此可以甚至是应该略去这部分长度的句子。对此我们进一步限制研究的句子长度的范围是不小于 4 且不大于 16 的整数。其次，统计前 80 回中每一回的字符总数（包括标点符号）。

将每一回的字符总数视为随机变量 Y，且设 Y 的数学期望 $E(Y)$ 存在。将不同长度句子的频数视为普通变量。共有 13 个普通变量

$$x_4, x_5, x_6, \cdots, x_{14}, x_{15}, x_{16},$$

其中 x_j 对应句子长度为 j. 受启发于论据 4.3，在这里我们直接尝试和讨论下述的多元线性回归模型：

$$Y = b_0 + b_4 x_4 + b_5 x_5 + \cdots + b_{16} x_{16} + \varepsilon,\ \varepsilon \sim N(0, \sigma^2),$$

其中 $b_0, b_4, b_5, \cdots, b_{16}, \sigma^2$ 都是与 $x_4, x_5, \cdots, x_{16}$ 无关的未知参数。前 80 回中共可以搜集到 80 个数据，可以表为集合

$$\left\{\left(x_{4,j}, x_{5,j}, \cdots, x_{16,j}, y_j\right) : j = 1, 2, \cdots, 80\right\}.$$

记

$$X = \begin{bmatrix} 1 & x_{4,1} & x_{5,1} & \cdots & x_{16,1} \\ 1 & x_{4,2} & x_{5,2} & \cdots & x_{16,2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{4,80} & x_{5,80} & \cdots & x_{16,80} \end{bmatrix},\ Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix},\ B = \begin{bmatrix} b_0 \\ b_4 \\ \vdots \\ b_{16} \end{bmatrix},$$

则由最大似然估计思想求得的参数 B 满足

$$X^T X B = X^T Y.$$

在 python 中，调用第三方库 statsmodels 中的 statsmodels.api.OLS( )函数，选择方法为“Least Squares”。函数输出的结果详见附录三中的 Result 4.

Table 4.3 $b_0, b_4, b_5, \cdots, b_{16}$ **的最大似然估计值.**

b0	b4	b5	b6	b7	b8	b9
-78.7249	6.9737	6.6617	6.1522	8.3921	7.7706	10.9186
b10	b11	b12	b13	b14	b15	b16
10.7674	16.3793	17.9089	14.5182	15.9953	17.176	26.748

由 Table 4.3, 我们得到一个 13 元的经验回归方程

$$\begin{aligned} y = &-78.7249 + 6.9737x_4 + 6.6617x_5 + 6.1522x_6 + 8.3921x_7 + 7.7706x_8 \\ &+ 10.9186x_9 + 10.7674x_{10} + 16.3793x_{11} + 17.9089x_{12} \\ &+ 14.5182x_{13} + 15.9953x_{14} + 17.1760x_{15} + 26.7480x_{16}. \end{aligned}$$

现对上面的经验回归方程作线性假设的显著性检验。建立原假设 H_0 与备择假设 H_1。

$$H_0 : b_4 = b_5 = \cdots = b_{16} = 0,$$

$$H_1 : b_4, b_5, \cdots, b_{16} \text{ 不全为 } 0.$$

查询附录三中的 Result 4，可得到 F 统计量的值为 1536，大于 p 值 1.56×10^{-76}（$\alpha = 0.05$），落入了拒绝域中。于是在显著性水平 0.05 下，拒绝原假设 H_0，认为回归效果是显著的。

此外，附录三中的 Result 4 中修正后的 R 方值“Adj. R-squared”的值为 0.996，非常接近 1，表明整体回归方程拟合效果很好。附录三中的 Result 4 中的“P>|t|”一栏除了常数项外均为 0.000，表明 $x_4, x_5, \cdots, x_{16}$ 中的任意一个变量对因变量 y 的解释性很强。

综上所述，该多元回归分析的效果是比较成功的。根据得到的经验回归线性方程，我们得到结论 4.5.

结论 4.5 一个回的篇幅与介于 4 到 16 的各个长度的句子的频数呈现多元线性关系。

结合结论 4.4 和结论 4.5，我们还可以得到结论 4.6.

结论 4.6 给定一个回的篇幅，可以大致推测介于 4 到 16 的各个长度句子在本回的频数。反之，给定任意一种长度的句子在本回的频数，可以大致推测本回的篇幅。

此外，注意到 Table 4.3 中系数 b_4, b_5, b_6 较小而其他参数较大。事实上，在 Fig 4.1 和 Fig4.2 对应数值较大的句长 j，相应的在 Table 4.3 中的 b_j 较小；在 Fig 4.1 和 Fig4.2 对应数值较小的句长 j，相应的在 Table 4.3 中的 b_j 较大。这也是比较符合直观的认识。

§5. 猜测：后 40 回是否出自曹雪芹之手

引言提及到“《红楼梦》的后 40 回作者是否曹雪芹”也是“红学”中的一大研究热题。在本节中我们也将应用一些简单的，基础的方法对这个问题作出回答。

在本节中，我们还是将研究对象放在高频汉字上，不过我们重新提取了另一套数据。首先对 1 到 120 回中的每一回，计算§3 中筛选得到的 79 个高频字在该回中的频率（和）。然后，将 1 到 120 回依次划分为 1 到 40 回，41 到 80 回，81 到 120 回三部分。视每部分是一个抽样得来的样本且来自不同的，相互独立的总体，高频字在某一回的频率就是样本的一个观察值。于是我们得到了 3 个容量均为 40 的样本 X_1, X_2, X_3。

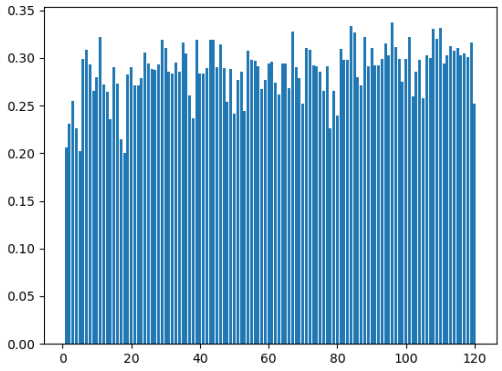

Fig 5.1　120 回中每一回的高频字频率分布直方图.

下面我们从两个视角来研究《红楼梦》后 40 回作者的归属问题。

视角一 单因素方差分析

视 1 到 40 回，41 到 80 回，81 到 120 回是同一个因素 K 下的 3 个不同的水平，分别记为 A_1，A_2，A_3. 此时三个水平下各有一个容量为 40 的样本 X_1，X_2，X_3。而样本来自对应水平的总体呈现的分布和态势也受到该水平对应的章节作者是谁的影响。基于这种思想，可设成立以下公理 5.1 和公理 5.2，并基于此和后面假设检验的结果作出后 40 回作者归属的推断。

公理 5.1 因素 K 下两个不同的水平的高频字频率之间有显著差异意味着这两个水平对应章节的作者很可能不是同一个人；

公理 5.2 因素 K 下两个不同的水平的高频字频率之间不存在显著差异意味着这两个水平对应章节的作者很可能是同一个人。

应用单因素方差分析之前，先对 X_1，X_2，X_3 作正态性检验。样本的观察值为连续的定量数据，因此使用 K-S 检验，借助 python 第三方库 scipy 中的 scipy.stats.kstest( )函数对

X_1，X_2，X_3 进行正态分布拟合优度检验（该检验使用的是 p 值法）。设定显著性水平为 0.05. 结果表明三个水平的统计量 D 的值均小于对应的 p 值。故在显著性水平 0.05 下认为 X_1，X_2，X_3 均来自正态分布总体。

Table 5.2 对 X_1，X_2，X_3 对应总体进行 K-S 检验得到的结果表格.

水平	统计量D	p值
1-40	0.154	0.273
41-80	0.128	0.499
81-120	0.142	0.366

样本总体的正态性分布拟合检验完毕后，先对 1-40 回和 41-80 回作单因素双水平方差分析。假设 A_1, A_2 下的样本 X_1, X_2 分别来自具有相同方差 σ^2，均值分别为 μ_1, μ_2 的正态总体 $N\left(\mu_1, \sigma^2\right)$，$N\left(\mu_2, \sigma^2\right)$，$\mu_1, \mu_2, \sigma^2$ 未知，且设 X_1, X_2 相互独立。设置显著性水平为 0.05，建立原假设 H_0 和备择假设 H_1。

$$H_0 : \mu_1 = \mu_2,$$

$$H_1 : \mu_1 \neq \mu_2.$$

得到方差分析表 Table 5.3.

Table 5.3 针对 1-40 回和 41-80 回的单因素双水平方差分析表.

方差来源	平方和	自由度	均方	F比	F临界值(显著性水平0.05)
因素	0.001419951	1	0.001419951	1.738152	3.963472051
误差	0.06372063	78	0.000816931	—	—
总和	0.065140581	79	—	—	—

$F = 1.738 < 3.963 = F_{0.05}(1, 78)$，$F$ 没有落在拒绝域，故在显著性水平 0.05 下接受原假设 H_0，认为 A_1, A_2 下高频字频率不存在显著差异。结合公理 5.2，这契合了前 80 回均出自曹雪芹之笔的事实。

再分别对 1-40 回与 81-120 回，41-80 回与 81-120 回，1-80 回与 81-120 回进行三组单因素双水平方差分析。假设检验过程的相关表述不再赘述。

Table 5.4 其他三个组别的单因素双水平方差分析表.

组别	方差来源	平方和	自由度	均方	F比	F临界值(显著性水平0.05)
1-40 & 81-120	因素	0.014335266	1	0.014335266	19.13387	3.963472051
	误差	0.058438299	78	0.000749209	—	—
	总和	0.072773565	79	—	—	—
41-80 & 81-120	因素	0.006731831	1	0.006731831	14.35627	3.963472051
	误差	0.036575154	78	0.000468912	—	—
	总和	0.043306985	79	—	—	—
1-80 & 81-120	因素	0.013571414	1	0.013571414	19.82283	3.921478181
	误差	0.080786993	118	0.000684636	—	—
	总和	0.094358406	119	—	—	—

三个组别的统计量 F 的值均大于对应的 F 临界值，落入了拒绝域中，分别代表水平 A_1 和 A_3，A_2 和 A_3，$A_1 \bigcup A_2$ 和 A_3 下高频字频率存在显著差异。

进一步地，根据蓝石提供的 Table 5.5[10]，计算效应尺度的大小。经计算，A_1 和 A_3 下两组样本的效应尺度为 1.007350521482138，A_2 和 A_3 下两组样本的效应尺度为 0.8494681628275679，$A_1 \bigcup A_2$ 和 A_3 下两组样本的效应尺度为 0.9256109576954272，均大于 0.8，因此三个组别中不同水平间的高频字频率存在的差异都是较大的。结合公理 5.1，这些结果都表明 1-80 回的作者和 81-120 回的作者很可能不是同一个人。因此我们作出结论 5.3.

Table 5.5 效应尺度大小与相应含义的表格.

效应尺度（ES）大小	含义/表述
小于0.2	差异无实际意义
0.2-0.5	差异较小
0.5-0.8	差异程度为中等
大于0.8	差异较大

结论 5.3 《红楼梦》的后 40 回非曹雪芹所著。

视角二 秩和双边检验[9]（Kruskal-Wallis 双边检验）

命题如果后 40 回的作者也是曹雪芹，那么 1-40 回，41-80 回，81-120 回中高频字频率来自的总体应该是同分布的。记 X_1, X_2, X_3 来自的连续型总体的概率密度函数分别是 $f_1(x)$，$f_2(x)$，$f_3(x)$. 可设成立以下公理 5.4，公理 5.5，公理 5.6，并基于此和后面假设检验的结果作出后 40 回作者归属的推断。

公理 5.4 $f_1(x) = f_2(x)$.

公理 5.5 若后 40 回作者是曹雪芹，则在显著性水平 α 下成立

$$f_1(x)=f_3(x),\ f_2(x)=f_3(x).$$

公理 5.6（5.5 的逆否命题） 若在显著性水平 α 下成立

$$f_1(x)\neq f_3(x),\ f_2(x)\neq f_3(x),$$

则后 40 回的作者非曹雪芹。

为了假设检验建模的需要，不妨假设 f_1, f_2, f_3 两两之间至多只差一个平移，即 $\exists a,b$，满足

$$f_1(x)=f_2(x-a)=f_3(x-b),\ a,b \text{ 为未知常数.}$$

现在再假设 X_1, X_2, X_3 来自的总体的均值存在，分别记作 μ_1, μ_2, μ_3。以 $f_1(x), f_2(x)$ 为例，**在假设 f_1, f_2 之间至多只差一个平移的前提下**，若 $\mu_1=\mu_2$，则有 $a=0$，此时有 $f_1(x)=f_2(x)$，表明 X_1, X_2 对应的总体同分布。而**抛开 f_1, f_2 之间至多只差一个平移的假设**，显然成立命题 5.7 及其逆否命题 5.8. 这两个命题是我们给出作者归属答案基于的思想。

命题 5.7 若（在显著性水平为 α 下）$f_1(x)=f_2(x)$，则在显著性水平为 α 下，假设检验的结果是不拒绝 $H_0:\mu_1=\mu_2$.

命题 5.8 若在显著性水平为 α 下假设检验的结果是拒绝 $H_0:\mu_1=\mu_2$，则（在显著性水平为 α 下）$f_1(x)\neq f_2(x)$.

以检验假设

$$H_0:\mu_1=\mu_2,\ H_1:\mu_1\neq\mu_2.$$

为例说明秩和检验的大致流程。将 X_1 中的 40 个观察值和 X_2 中的 40 个观察值合并起来，将合并后的 80 个数据按从小到大排好序，之后根据[9]中给出的计算公式和步骤确定每个观察值对应的秩，接着计算来自第一个样本 X_1 的秩和 R_1。由于 X_1 和 X_2 的样本容量均为 40，大于 10，因此根据[9]中的结论，成立

$$R_1 \sim N\left(E(R_1), D(R_1)\right),$$

即有

$$\frac{R_1 - E(R_1)}{D(R_1)} \sim N(0,1).$$

给定显著性水平α，计算得拒绝域的两个界限C_L, C_U.

$$C_L = E(R_1) - z_{\alpha/2} \bullet D(R_1),$$

$$C_U = E(R_1) + z_{\alpha/2} \bullet D(R_1).$$

拒绝域为$r_1 \le C_L$或$r_1 \ge C_U$.

Table 5.6 四个组别的秩和检验表（显著性水平 0.05）

	H_0	H_1	r_1	C_L	C_U
1-40 & 41-80	$\mu_1 = \mu_2$	$\mu_1 \ne \mu_2$	1524	1449.06	1790.94
1-40 & 81-120	$\mu_1 = \mu_3$	$\mu_1 \ne \mu_3$	1202	1449.06	1790.94
41-80 & 81-120	$\mu_2 = \mu_3$	$\mu_2 \ne \mu_3$	1212	1449.06	1790.94
1-80 & 81-120	$\mu_4 = \mu_3$	$\mu_4 \ne \mu_3$	3246	2124.54	2715.46

由 Table 5.6 第一行的结果，

$$C_L = 1449.06 < 1524 = r_1 < 1790.94 = C_U,$$

r_1没有落入拒绝域，故在显著性水平 0.05 下不拒绝H_0，认为 1-40 回和 41-80 回下的高频字在每回出现的频率不存在显著差异。这和 1-40 回及 41-80 回均出自曹雪芹之手的事实不矛盾，也验证了公理 5.4 的正确性。

由 Table 5.6 第二，三行的结果，

$$r_1 = 1202 < 1449.06 = C_L,$$
$$r_1 = 1212 < 1449.06 = C_L,$$

r_1均落入了拒绝域，故在显著性水平 0.05 下拒绝H_0，认为 1-40 回和 81-120 回，41-80 回和 81-120 回下的高频字在每回出现的频率存在显著差异。由命题 5.8，在显著性水平 0.05 下成立

$$f_1(x) \neq f_3(x),\ f_2(x) \neq f_3(x),$$

再由公理 5.6，可得出与结论 5.3 相同的结论。对 Table 5.6 第四行的结果的分析也可以导出和证实结论 5.3.

§ 6. 参考文献

[1] Efron, B. and Thisted, R., *Estimating the number of unseen species: How many words did Shakespeare know?* Biometrika, 63(1976), 435-437.

[2] Thisted, R. and Efron, B., *Did Shakespeare write a newly-discovered poem?* Biometrika, 74(1987), 445-455.

[3] 陈炳藻，*从词汇上的统计论《红楼梦》的作者问题*,“首届国际《红楼梦》研讨会”(1980，美国威斯康星大学).

[4] 贾洪卫，董坚，徐锐，*计算机与“红学”研究综论*(2003，可参见 https://www.Tlsoft.com，中国人民大学统计数据库研究室).

[5] 陈大康，*从数理语言看后四十回的作者*，红楼梦学刊, 1(1987), 293-318.

[6] 李贤平，*《红楼梦》成书新说*，复旦大学学报社科版, 5(1987), 3-16.

[7] 韦博成，*《红楼梦》前 80 回与后 40 回某些文风差异的统计分析*，应用概率统计, 4(2009).

[8] 安鸿志，*趣话概率—兼话《红楼梦》中的玄机*，科学出版社，北京, 2009.

[9] 盛骤，谢式千，潘承毅，*概率论与数理统计*，高等教育出版社，北京, 2018.

[10] 蓝石，*社会科学定量研究的变量类型、方法选择及范例解析*, 2011, 44-56.

[11] 维基百科，*Kolmogorov Smirnov test*.
https://en.wikipedia.org/wiki/Kolmogorov%E2%80%93Smirnov_test.

[12] 百度百科，*骈散结合*.
https://baike.baidu.com/item/%E9%AA%88%E6%95%A3%E7%BB%93%E5%90%88/10620401?fr=aladdin.

§7. 附录一：《红楼梦》原文资料

1. 从子非网 https://www.88u4.com/469.html 下载的《红楼梦》原版电子书资料

《红楼梦》原.pdf

2. 原版电子书资料转化为可供 word 文档打开和编辑的 docx 格式

《红楼梦》原.docx

3. 初步删去冗杂信息，剩下正文 + 回前墨 + 回后评 + 注释部分的《红楼梦》word 文档

《红楼梦》删.docx

§8. 附录二：代码实现

函数文件	功能说明
main.py	进行数据读取, 清洗, 整理, 计算, 分析
myfunctions.py	作为自定义的函数库，内置4个函数文件： 1. 进行数据读入与清洗整理的函数; 2. 进行单因素方差分析的函数; 3. 进行秩和检验的函数; 4. 计算效应尺度的函数.

main.py

main.py 文件:

myfunctions.py

myfunctions.py 文件:

main.py 代码

```
# coding: UTF-8
import docx
import myfunctions as my
import matplotlib.pyplot as plt
```

```
from prettytable import PrettyTable
import numpy as np
from scipy import stats
import statsmodels.api as sm

# 使用说明：该.py文件需要配合myfunctions.py和《红楼梦》删,docx
# 直接运行即可，所有结果将会输出

# 读入数据并清洗和整理
hlm = my.statistics_reading_cleaning_arranging()

# 读入红楼梦原版资料并统计原版资料总字数
file = docx.Document('A:\\shuli\\《红楼梦》删.docx')
word_symbol_original_count_all = 0
for para in file.paragraphs:
    if para.text != '':
        word_symbol_original_count_all += len(para.text)
del file, para

# 统计量：每一回的字数和标点符号总数，并可视化
word_symbol_count = {}
for k in range(1, 121):
    word_symbol_count['第' + str(k) + '回'] = len(hlm['第' + str(k) + '回'])
del k

plt.figure()
plt.bar(range(1, 41), list(word_symbol_count.values())[0:40])
plt.bar(range(41, 81), list(word_symbol_count.values())[40:80])
plt.bar(range(81, 121), list(word_symbol_count.values())[80:120])
plt.show()

plt.figure()
plt.boxplot(word_symbol_count.values())
plt.show()

t = list(word_symbol_count.values()).index(np.min(list(word_symbol_count.values())))
print('字符数最少的回数是', list(word_symbol_count.keys())[t],
      '字符数为', np.min(list(word_symbol_count.values())))
t = list(word_symbol_count.values()).index(np.max(list(word_symbol_count.values())))
print('字符数最多的回数是', list(word_symbol_count.keys())[t],
      '字符数为', np.max(list(word_symbol_count.values())))
print('----------------------------------------------------------------------\n')
del t

```

```
# 计算结果并可视化：计算清洗掉的数据占原版资料的比重，然后画成饼图
table = PrettyTable()
t = sum(list(word_symbol_count.values()))
table.add_column(' ', ['清洗前的数据共有字数', '清洗后的数据共有字数', '清洗掉的数据占原数据比重'])
table.add_column(' ', [word_symbol_original_count_all, t,
                       (word_symbol_original_count_all - t) / word_symbol_original_count_all])
print(table)
print('-------------------------------------------------------------------------\n')

plt.figure()
plt.pie([t, word_symbol_original_count_all - t],
        explode=[0.2, 0.05],
        labels=['Words After Data Washing', 'Words Washed'],
        colors=['lightblue', 'lightgray'],
        autopct='%1.2f%%',
        shadow=True,
        startangle=100
        )
plt.axis('equal')
plt.show()
del table, t

# 提取数据：给出前 1-40 回和 41-80 回单字使用次数的字典
hlm_words_1_40 = {}
hlm_words_41_80 = {}
for k in range(1, 41):
    for word in hlm['第' + str(k) + '回']:
        if word in hlm_words_1_40.keys():
            hlm_words_1_40[word] += 1
        else:
            hlm_words_1_40[word] = 1
    for word in hlm['第' + str(k + 40) + '回']:
        if word in hlm_words_41_80.keys():
            hlm_words_41_80[word] += 1
        else:
            hlm_words_41_80[word] = 1
del k, word

# 比对 1-40 回和 41-80 回各自的前 k 个高频字中有多少个重复的
hot_words_1_40 = []
hot_words_41_80 = []
k = 100
```

```
91.
92. t1 = sorted(list(hlm_words_1_40.values()), reverse=True)[0:k]
93. t2 = hlm_words_1_40.copy()
94. for value in t1:
95.     word = list(t2.keys())[list(t2.values()).index(value)]
96.     hot_words_1_40.append(word)
97.     del t2[word]
98.
99. t1 = sorted(list(hlm_words_41_80.values()), reverse=True)[0:k]
100.t2 = hlm_words_41_80.copy()
101.for value in t1:
102.    word = list(t2.keys())[list(t2.values()).index(value)]
103.    hot_words_41_80.append(word)
104.    del t2[word]
105.
106.del k, t1, t2, value, word
107.
108.num = 0
109.for item in hot_words_1_40:
110.    if item in hot_words_41_80:
111.        num += 1
112.        print('1-40 回和 41-80 回中均有高频字 ' + item)
113.print('1-40 回和 41-80 回中，各自的前', len(hot_words_1_40), '高频字重复的有', num, '个
    ')
114.print('----------------------------------------------------------------------\n')
115.del num, item
116.
117.# 提取数据：给出前 80 回的所有单字使用次数的字典，为后面提取高频用字提供基础
118.hlm_words_before80 = {}
119.for k in range(1, 81):
120.    for word in hlm['第' + str(k) + '回']:
121.        if word in hlm_words_before80.keys():
122.            hlm_words_before80[word] += 1
123.        else:
124.            hlm_words_before80[word] = 1
125.del k, word
126.
127.# 提取数据：考虑前 k1 个高频使用字符，并且列成表格
128.k1 = k2 = 100
129.t1 = sorted(list(hlm_words_before80.values()), reverse=True)[0:k1]
130.t2 = hlm_words_before80.copy()
131.hot_words_before80 = []
132.table = PrettyTable(['高频使用字符', '使用次数'])
133.for value in t1:
```

```
    word = list(t2.keys())[list(t2.values()).index(value)]
    hot_words_before80.append(word)
    table.add_row([word, value])
    k2 -= 1
    del t2[word]
print(table)
print('这' + str(k1) + '个高频使用字占前 80 回篇幅的百分比为: ' + str(sum(t1) / sum(list(hlm_words_before80.values()))))
print('----------------------------------------------------------------------\n')

# 计算: 将 1-40 回和 41-80 回整合在一起时, 其前 k 高频字既是 1-40 回前 k 高频字, 又是 41-80 回前 k 高频字的个数
num = 0
for item in hot_words_before80:
    if item in hot_words_1_40 and item in hot_words_41_80:
        num += 1
print('前 80 回前' + str(len(hot_words_before80)) + '个高频字中, 既是 1-40 回前' +
      str(len(hot_words_before80)) + '高频字, 又是 41-80 回中前' +
      str(len(hot_words_before80)) + '高频字的字有 ' + str(num) + '个')
print('----------------------------------------------------------------------\n')
del num, item

# 分析: 对前 k1 个高频使用字符作箱线图和直方图, 以决定要使用哪些字
plt.figure()
plt.hist(t1, bins=55)
plt.show()

plt.figure()
plt.boxplot(t1)
plt.show()
del k1, k2, t1, t2, value, word, table

# 数据提取: 经过箱线图和直方图的分析, 决定使用的高频字
hot_words_before80_picked = hot_words_before80[
                            hot_words_before80.index('去'):
                            ]
hot_words_before80_picked.remove('? ')
hot_words_before80_picked.remove('、')
hot_words_before80_picked.remove('! ')

# 计算: 决定使用的高频字占前 80 回的篇幅
hot_words_before80_picked_count = 0
for word in hot_words_before80_picked:
    hot_words_before80_picked_count += hlm_words_before80[word]
```

```
table = PrettyTable()
table.add_row(['筛选得高频字总字数', hot_words_before80_picked_count])
t = sum(list(word_symbol_count.values()))
table.add_row(['前 80 回总字数', t])
table.add_row((['筛选所得高频字占前 80 回比重',
                hot_words_before80_picked_count / t]))
print(table)

plt.figure()
plt.pie([hot_words_before80_picked_count, t - hot_words_before80_picked_count],
        explode=[0.2, 0.05],
        labels=['Frequently Used Picked', 'The Rest'],
        colors=['lightblue', 'lightgray'],
        autopct='%1.2f%%',
        shadow=True,
        startangle=100
        )
plt.axis('equal')
plt.show()
del table, t, word

# 提取数据：统计全 120 回中每一回各句长出现的频数（除去每一回的标题）
# 句子分割标准：逗号，句号，问号，感叹号，冒号，顿号，分号，省略号，其他标点符号算作字
sentence_len_120 = {}
t = 0  # 句长字数统计中间变量，当进入下一句时变为 0
for k in range(1, 121):
    sentence_len_120['第' + str(k) + '回'] = {}
    for word in hlm['第' + str(k) + '回'][19:]:
        if word in [',', '，', '.', '。', '?', '？', '!', '！', ':', '：', '、', '；', ';', '…']:
            if t == 0:
                continue
            elif t in sentence_len_120['第' + str(k) + '回'].keys():
                sentence_len_120['第' + str(k) + '回'][t] += 1
                t = 0
            else:
                sentence_len_120['第' + str(k) + '回'][t] = 1
                t = 0
        else:
            t += 1
del t, k, word

# 抽样：在 1 到 80 回的每一回随机抽取一个长度为 sampling_len 的句段
# 统计其中拥有筛选得高频字的个数，每一回抽取 iter_times 次
```

```
data1 = {}
sampling_len = 100
iter_times = 50
for k in range(1, 81):
    data1['第' + str(k) + '回'] = []
    for iter in range(iter_times):
        data1['第' + str(k) + '回'].append(0)
        pos = int(np.random.rand() *
                  (len(hlm['第' + str(k + 1) + '回']) - sampling_len))
        for item in hlm['第' + str(k + 1) + '回'][pos:pos + sampling_len]:
            if item in hot_words_before80_picked:
                data1['第' + str(k) + '回'][-1] += 1
del sampling_len, iter_times, k, iter, pos, item

# 打印出抽样的结果
for k in range(80):
    print(list(data1.keys())[k], list(data1.values())[k])
print('------------------------------------------------------------------------\n')

# 分析: 对 data1 中每一回对应的抽样来自的总体作 K-S 正态性检验, 然后可视化
not_norm = []
t1 = []
t2 = []
for k in range(1, 81):
    t = stats.kstest(data1['第' + str(k) + '回'], 'norm',
                     args=(
                         np.mean(data1['第' + str(k) + '回']),
                         np.std(data1['第' + str(k) + '回'], ddof=1)
                     ))
    t1.append(t[0])
    t2.append(t[1])
    if t1[-1] > t2[-1]:
        not_norm.append(k)

    print('第' + str(k) + '回: 统计量 D = ' + str(t1[-1]) + ', pvalue = ' + str(t2[-1]))
print('抽样不来自正态总体的章节是:', not_norm, '章节数为:', len(not_norm))
print('------------------------------------------------------------------------\n')

plt.figure()
plt.bar(range(1, 81), t2)
plt.bar(range(1, 81), np.array(t1) * (-1))
plt.show()
del t1, t2, t, k
```

```
262.
263.# 分析：使用单因素方差分析对于前 80 回，回数（80 个水平）对高频字的使用有无显著影响
264.t = data1.copy()
265.if not_norm != []:
266.    for k in not_norm:
267.        del t['第' + str(k) + '回']
268.table = my.anova1(t, alpha=0.05)
269.print('前 80 回对高频字使用密度的单因素方差分析')
270.print(table)
271.print('----------------------------------------------------------------------\n')
272.del table, not_norm, t
273.if 'k' in dir():
274.    del k
275.
276.# 分析：按顺序每 5 回作一个单因素方差分析，看看回数对高频字的使用有无显著影响
277.r = []
278.for k in range(16):
279.    t = {}
280.    for s in range(1, 6):
281.        t['第' + str(5 * k + s) + '回'] = data1['第' + str(5 * k + s) + '回']
282.    table = my.anova1(t, alpha=0.05)
283.    print('第' + str(5 * k + 1) + '到' + str(5 * k + 5) + '回对高频字使用密度的单因素方差
    分析')
284.    print(table)
285.    r.append(table._rows[0][4])
286.    print('----------------------------------------------------------------------\n')
287.plt.figure()
288.plt.bar(range(1, len(r) + 1), r)
289.plt.plot([1, len(r) + 1], [table._rows[0][5], table._rows[0][5]], color='red', linewid
    th=1)
290.plt.show()
291.del k, t, s, table, r
292.
293.# 数据提取：计算 120 回一每回中高频使用字的频率
294.hot_words_120_fre = {}
295.for k in range(1, 121):
296.    hot_words_120_fre['第' + str(k) + '回'] = 0
297.    for word in hlm['第' + str(k) + '回']:
298.        if word in hot_words_before80_picked:
299.            hot_words_120_fre['第' + str(k) + '回'] += 1
300.    hot_words_120_fre['第' + str(k) + '回'] /= word_symbol_count['第' + str(k) + '回
    ']
301.del k, word
302.
```

```
303.plt.figure()
304.plt.bar(range(1, 121), hot_words_120_fre.values())
305.plt.show()
306.
307.# 数据提取：将 120 回分成三大块：1-40 回，41-80 回，81-120 回，计算每一大块下的每一回的高频字频率作为样本观察值
308.data2 = {'1_40': [], '41_80': [], '81_120': []}
309.for k in range(1, 41):
310.    data2['1_40'].append(hot_words_120_fre['第' + str(k) + '回'])
311.    data2['41_80'].append(hot_words_120_fre['第' + str(k + 40) + '回'])
312.    data2['81_120'].append(hot_words_120_fre['第' + str(k + 80) + '回'])
313.del k
314.
315.# 分析：对 data2 中每一个水平对应的抽样来自的总体作 K-S 正态性检验，然后可视化
316.not_norm = []
317.t1 = []
318.t2 = []
319.for k in range(3):
320.    t = stats.kstest(data2[str(40 * k + 1) + '_' + str(40 * k + 1 + 39)], 'norm',
321.                     args=(
322.                         np.mean(data2[str(40 * k + 1) + '_' + str(40 * k + 1 + 39)]),
323.                         np.std(data2[str(40 * k + 1) + '_' + str(40 * k + 1 + 39)], ddof=1)
324.                     ))
325.    t1.append(t[0])
326.    t2.append(t[1])
327.    if t1[-1] > t2[-1]:
328.        not_norm.append(k)
329.print('抽样不来自正态总体的水平是:', not_norm, '个数为:', len(not_norm))
330.print('-------------------------------------------------------------------------\n')
331.
332.plt.figure()
333.plt.bar(range(1, 4), t2)
334.plt.bar(range(1, 4), np.array(t1) * (-1))
335.plt.show()
336.del t1, t2, t, k, not_norm
337.
338.# 分析：使用单因素分析四个水平对：1_40 与 41_80，1_40 与 81_120，41_80 与 81_120，1_80 与 81_120
339.# 目的：从高频字频率角度判断后 40 回(即 81-120 回)是否还是曹雪芹所著
340.print('通过 1-40 回与 41-80 回的单因素双水平方差分析验证前 80 回是曹雪芹所著')
341.t = data2.copy()
342.del t['81_120']
```

```
table = my.anova1(t, alpha=0.05)
print(table)
print('-------------------------------------------------------------------------\n')

print('通过 1-40 回与 81-120 回的单因素双水平方差分析验证后 40 回非曹雪芹所著')
t = data2.copy()
del t['41_80']
table = my.anova1(t, alpha=0.05)
print(table)
r = my.effect_scale(t)
print('效应尺度 = ' + str(r))
print('-------------------------------------------------------------------------\n')
del t, table, r

print('通过 41-80 回与 81-120 回的单因素双水平方差分析验证后 40 回非曹雪芹所著')
t = data2.copy()
del t['1_40']
table = my.anova1(t, alpha=0.05)
print(table)
r = my.effect_scale(t)
print('效应尺度 = ' + str(r))
print('-------------------------------------------------------------------------\n')
del t, table, r

print('通过 1-80 回与 81-120 回的单因素双水平方差分析验证后 40 回非曹雪芹所著')
t = data2.copy()
t['1_80'] = t['1_40'] + t['41_80']
del t['1_40'], t['41_80']
table = my.anova1(t, alpha=0.05)
print(table)
r = my.effect_scale(t)
print('效应尺度 = ' + str(r))
print('-------------------------------------------------------------------------\n')
del t, table, r

# 分析: 使用秩和检验分析四个总体对: 1_40 与 41_80, 1_40 与 81_120, 41_80 与 81_120, 1_80 与
  81_120
# 目的: 从高频字频率角度判断后 40 回(即 81-120 回)是否还是曹雪芹所著
print('通过 1-40 回与 41-80 回的秩和检验验证前 80 回是曹雪芹所著')
t = data2.copy()
del t['81_120']
my.kruskal_wallis_bilateral_test_over10(t, alpha=0.05)
print('-------------------------------------------------------------------------\n')

```

```
print('通过 1-40 回与 81-120 回的秩和检验验证后 80 回是曹雪芹所著')
t = data2.copy()
del t['41_80']
my.kruskal_wallis_bilateral_test_over10(t, alpha=0.05)
print('-------------------------------------------------------------------------\n')

print('通过 41-80 回与 81-120 回的秩和检验验证后 80 回是曹雪芹所著')
t = data2.copy()
del t['1_40']
my.kruskal_wallis_bilateral_test_over10(t, alpha=0.05)
print('-------------------------------------------------------------------------\n')

print('通过 1-80 回与 81-120 回的秩和检验验证后 80 回是曹雪芹所著')
t = data2.copy()
t['1_80'] = t['1_40'] + t['41_80']
del t['1_40'], t['41_80']
my.kruskal_wallis_bilateral_test_over10(t, alpha=0.05)
print('-------------------------------------------------------------------------\n')

del t

# 整理: 分别统计 1-40 回, 41-80 回, 81-120 回中句长为 1,2,...,20 的句子个数, 并且可视化
sentence_len_1_40 = {}
sentence_len_41_80 = {}
sentence_len_81_120 = {}
for k in range(1, 21):
    sentence_len_1_40[k] = 0
    sentence_len_41_80[k] = 0
    sentence_len_81_120[k] = 0

    for chapter in range(1, 41):
        if k in sentence_len_120['第' + str(chapter) + '回'].keys():
            sentence_len_1_40[k] += sentence_len_120['第' + str(chapter) + '回'][k]
        if k in sentence_len_120['第' + str(chapter + 40) + '回'].keys():
            sentence_len_41_80[k] += sentence_len_120['第' + str(chapter + 40) + '回'][k]
        if k in sentence_len_120['第' + str(chapter + 80) + '回'].keys():
            sentence_len_81_120[k] += sentence_len_120['第' + str(chapter + 80) + '回'][k]

del k, chapter

plt.figure()
plt.bar(
```

```
    np.array(range(1, 21)),
    np.array(list(sentence_len_1_40.values()))
    + np.array(list(sentence_len_41_80.values()))
    + np.array(list(sentence_len_81_120.values())),
)
plt.show()

plt.figure()
plt.plot(np.array(range(1, 21)), sentence_len_1_40.values(), linewidth=1)
plt.plot(np.array(range(1, 21)), sentence_len_41_80.values(), linewidth=1)
plt.plot(np.array(range(1, 21)), sentence_len_81_120.values(), linewidth=1)
plt.show()

# 分析：在前 80 回中使用多元线性回归分析句长在各回的规律
x = []
for chapter in range(1, 81):
    t = []
    for k in range(4, 17):
        t.append(sentence_len_120['第' + str(chapter) + '回'][k])
    x.append(t)
X = sm.add_constant(x)
model = sm.OLS(list(word_symbol_count.values())[0:80], X).fit()
print(model.summary())
del x, k, t, chapter, X, model

# # 清除所有的变量
# for var in dir():
#     if not var.startswith('__'):
#         globals().pop(var)
# del var
```

myfunctions.py 代码

```
# coding: UTF-8
import docx
from scipy import stats
from prettytable import PrettyTable
import numpy as np


# 函数功能：读入初步处理后的《红楼梦》，并进行数据清洗和整理
def statistics_reading_cleaning_arranging():
    # 读入原《红楼梦》Word 文档删去位于文档前后部分的目录和其他冗余信息后剩下的部分
    file = docx.Document('A:\\shuli\\《红楼梦》删.docx')
```

```
12.
13.     # 将《红楼梦》一百二十回的内容装入一个字典 hlm, 第 x 回对应的键为'第 x 回', 键的值为该回的正文内容（不分段落）
14.     hlm = {}
15.
16.     # 进行第一层处理: 1.检测关键字以划分一百二十回;
17.     # 2. 清洗掉每回前的回前墨部分， 每回之后的回后评部分.
18.
19.     switch = 1  # 控制是否将当前光标所指的文字收集起来, 1 表示收集, 0 表示不收集
20.     chapter = 1  # 定位当前的回
21.
22.     for para in file.paragraphs:
23.
24.         # 若本段为空段, 则忽略本段内容, 且若之前遇到'回前墨'或'回后评', 则将文字收集开关打开
25.         if para.text == '':
26.             if switch == 0:
27.                 switch = 1
28.             continue
29.
30.         # 若本段非空段的情况
31.         else:
32.             # 检测本段开头有无关键字'第', ’回'
33.             # 若检测到, 则在字典创建下一回的键, 初始化其值为空列表[]
34.             # 并设置文字收集开关为 1
35.             if para.text[0] == '第' and para.text[-1] == '回':
36.                 location = '第' + str(chapter) + '回'
37.                 chapter += 1
38.                 hlm[location] = ''
39.
40.             # 检测本段开头有无关键字'回前墨'或'回后评'
41.             # 若检测到, 则将文字收集开关调整为 0
42.             elif '回前墨' in para.text[0:min(5, len(para.text))] or \
43.                     '回后评' in para.text[0:min(5, len(para.text))]:
44.                 switch = 0
45.
46.         # 若文字收集开关 switch 是 1, 则将本段内容中除了空格外的内容收录到字典 hlm 对应的回中
47.         if switch == 1:
48.             for word in para.text:
49.                 if word != ' ':
50.                     hlm[location] += word
51.
52.     # 进行第二层处理: 去除每一回最后的注释
```

```
    # 先针对[数字]的注释
    for k in range(1, 121):
        t = []
        for pos in range(0, len(hlm['第' + str(k) + '回']) - 3):
            if hlm['第' + str(k) + '回'][pos: pos + 3] == '[1]' or \
                    hlm['第' + str(k) + '回'][pos: pos + 3] == '【1】':
                t.append(pos)
        if t != []:
            hlm['第' + str(k) + '回'] = hlm['第' + str(k) + '回'][0:t[-1]]

    # 再针对[汉字]的注释
    for k in range(1, 121):
        t = []
        for pos in range(0, len(hlm['第' + str(k) + '回']) - 3):
            if hlm['第' + str(k) + '回'][pos: pos + 3] == '[一]' or \
                    hlm['第' + str(k) + '回'][pos: pos + 3] == '【一】':
                t.append(pos)
        if t != []:
            hlm['第' + str(k) + '回'] = hlm['第' + str(k) + '回'][0:t[-1]]

    # 进行第三层处理：去除每一回正文中的注释角标[x]
    switch = 1  # 控制是否保留文字的开关，1 表示保留，0 表示不保留
    for k in range(1, 121):
        t = ''
        for word in hlm['第' + str(k) + '回']:
            if word == '[' or word == '【':
                switch = 0
            elif word == ']' or word == '】':
                switch = 1
                continue

            if switch == 1:
                t += word
        hlm['第' + str(k) + '回'] = t

    return hlm


# 函数功能：进行单因素方差分析
def anova1(data, alpha=0.05):
    # data 是单因素各水平抽样数据，alpha 是显著性水平
    s = len(data)  # 水平数
    n_j = []  # 各水平下的样本容量
    n = 0  # 样本总容量
```

```
    for item in data.values():
        n_j.append(len(item))
        n += n_j[-1]

    X_mean = 0
    for item in data.values():
        X_mean += sum(np.array(item))
    X_mean = X_mean / n

    SST = 0
    for item in data.values():
        SST += sum(np.array(item) ** 2)
    SST -= n * (X_mean ** 2)

    SSA = 0
    k = 0
    for item in data.values():
        SSA += n_j[k] * (np.mean(np.array(item)) ** 2)
        k += 1
    SSA -= n * (X_mean ** 2)

    SSE = SST - SSA

    table = PrettyTable()
    table.add_column('方差来源', ['因素', '误差', '总和'])
    table.add_column('平方和', [SSA, SSE, SST])
    table.add_column('自由度', [s - 1, n - s, n - 1])
    table.add_column('均方', [SSA / (s - 1), SSE / (n - s), '_'])
    table.add_column('F 比', [(SSA * (n - s)) / (SSE * (s - 1)), '_', '_'])
    table.add_column('F 临界值
', [stats.f.isf(alpha, dfn=s - 1, dfd=n - s), '_', '_'])

    return table


# 函数功能: 进行双边的秩和检验
def kruskal_wallis_bilateral_test_over10(data, alpha=0.05):
    # 要求检验的总体个数为 2, 且 n1, n2 均不小于 10
    if len(data) != 2:
        return

    all1 = list(data.values())[0]
    all2 = list(data.values())[1]
    n1 = len(all1)
```

```
    n2 = len(all2)
    if n1 < 10 or n2 < 10:
        return

    # 计算各个观察值的秩
    all = np.sort(np.array(all1 + all2))
    rank = np.array(range(1, n1 + n2 + 1))
    group = []
    backup = []
    for item in all:
        if item not in backup:
            pos = np.where(all == item)[0]
            if len(pos) > 1:
                backup.append(item)
                group.append(len(pos))
                t = np.sum(all[pos]) / len(pos)
                rank[pos] = t

    # 计算来自第 1 个总体的秩和
    R1 = 0
    for k in range(n1 + n2):
        if all[k] in all1:
            R1 += rank[k]

    # 计算正态总体的均值和方差
    mju = n1 * (n1 + n2 + 1) / 2
    if group == []:
        sigma = np.sqrt(n1 * n2 * (n1 + n2 + 1) / 12)
    else:
        n = n1 + n2
        t = np.sum((np.array(group, dtype='float') ** 3)) - \
            np.sum(np.array(group, dtype='float'))
        sigma = np.sqrt(
            n1 * n2 * (n * (n ** 2 - 1) - t) / (12 * n * (n - 1))
        )

    # 计算拒绝域
    t = stats.norm.isf(alpha)
    C_L = mju - t * sigma
    C_U = mju + t * sigma
    print('秩相同的组的组数为:', len(group))
    print('第一样本的秩和 R1 的观察值为 r1=', R1)
    print('拒绝域为: r1 < ' + str(C_L) + ', r1 > ' + str(C_U))
    if R1 < C_L or R1 > C_U:
```

```
        print('两个总体的数据有显著差异')
    else:
        print('两个总体的数据无显著差异')

    return R1, C_L, C_U, len(group)


# 函数功能：在单因素方差分析结果是有显著差异的情况下，计算效应尺度的大小
# 限制：要求单因素的水平数为 2，即只有两组
def effect_scale(data):
    if len(data) != 2:
        return
    t1 = np.array(list(data.values())[0])
    t2 = np.array(list(data.values())[1])
    t = 2 * np.abs(np.mean(t1) - np.mean(t2)) / (np.std(t1, ddof=1) + np.std(t2, ddof=
1))
    print('两组样本的效应尺度为', t)

    if t < 0.2:
        print('差异无实际意义')
    elif t < 0.5:
        print('差异较小')
    elif t < 0.8:
        print('差异程度为中等')
    else:
        print('差异较大')

     return t
```

§ 9. 附录三：未在正文展示的一些结果

Result 1

第 1 回 [23, 31, 16, 24, 22, 30, 30, 27, 14, 30, 20, 22, 28, 24, 30, 31, 26, 25, 24, 25, 13, 23, 5, 31, 20, 18, 23, 29, 23, 30, 15, 18, 6, 22, 24, 20, 25, 16, 25, 25, 21, 30, 15, 19, 8, 20, 24, 16, 31, 28]

第 2 回 [37, 28, 26, 27, 24, 22, 40, 27, 25, 23, 24, 28, 32, 27, 33, 29, 24, 26, 25, 24, 24, 27, 26, 16, 23, 22, 17, 23, 29, 26, 25, 28, 19, 28, 31, 22, 18, 29, 31, 24, 16, 32, 22, 23, 25, 13, 22, 36, 22, 30]

第 3 回 [21, 25, 26, 24, 12, 20, 19, 28, 26, 26, 30, 13, 22, 19, 19, 24, 29, 28, 23, 26, 20, 30, 27, 23, 27, 26, 19, 31, 20, 31, 32, 19, 21, 29, 24, 29, 32, 28, 22, 33, 20, 19, 24, 27, 16, 27, 25, 20, 19, 25]

第 4 回 [20, 17, 13, 14, 21, 10, 15, 27, 28, 15, 21, 22, 33, 16, 19, 16, 17, 16, 18, 31, 15, 10, 32, 19, 22, 13, 32, 19, 26, 34, 11, 21, 32, 13, 11, 22, 14, 16, 16, 19, 33, 19, 17, 20, 25, 17, 15, 19, 13, 16]

第 5 回 [33, 25, 25, 27, 24, 39, 26, 28, 34, 24, 32, 32, 38, 39, 26, 27, 25, 33, 36, 24, 20, 26, 19, 24, 25, 25, 23, 27, 22, 23, 29, 35, 32, 25, 29, 34, 43, 35, 35, 20, 25, 30, 31, 26, 32, 31, 36, 26, 31, 27]

第 6 回 [35, 31, 18, 35, 28, 22, 34, 33, 28, 33, 29, 29, 27, 31, 25, 35, 33, 28, 28, 27, 20, 32, 22, 32, 31, 38, 31, 45, 29, 24, 41, 28, 34, 32, 32, 36, 28, 28, 27, 39, 35, 31, 32, 24, 28, 27, 34, 41, 31, 33]

第 7 回 [34, 28, 25, 40, 35, 24, 34, 29, 27, 30, 28, 29, 33, 16, 30, 29, 27, 33, 35, 30, 28, 38, 33, 23, 35, 27, 25, 34, 16, 39, 29, 28, 17, 25, 18, 28, 29, 24, 23, 36, 25, 31, 33, 26, 31, 28, 23, 30, 30, 32]

第 8 回 [29, 26, 24, 26, 22, 29, 26, 31, 23, 30, 27, 23, 21, 26, 31, 22, 21, 29, 26, 21, 30, 29, 26, 21, 21, 29, 23, 24, 30, 27, 21, 28, 24, 26, 22, 27, 25, 30, 30, 25, 28, 23, 24, 28, 27, 27, 38, 27, 25, 24]

第 9 回 [38, 27, 23, 27, 35, 33, 34, 28, 35, 29, 30, 34, 33, 31, 34, 17, 25, 20, 22, 35, 32, 36, 34, 34, 26, 30, 34, 41, 24, 40, 26, 29, 36, 26, 30, 8, 31, 29, 32, 25, 23, 28, 28, 35, 28, 36, 29, 29, 33, 31]

第 10 回 [31, 41, 31, 30, 27, 33, 28, 36, 35, 29, 24, 31, 38, 38, 29, 33, 30, 43, 40, 30, 38, 37, 40, 40, 41, 34, 34, 41, 22, 35, 29, 34, 31, 27, 36, 23, 41, 38, 35, 38, 37, 40, 27, 41, 28, 26, 34, 29, 34, 32]

第 11 回 [32, 21, 19, 22, 24, 22, 29, 27, 28, 20, 36, 28, 31, 30, 23, 25, 22, 22, 28, 37, 35, 17, 30, 25, 25, 31, 21, 19, 30, 29, 22, 17, 23, 27, 23, 19, 34, 23, 17, 27, 28, 24, 23, 24, 31, 31, 31, 24, 17, 28]

第 12 回 [26, 27, 28, 37, 11, 14, 26, 30, 24, 29, 29, 24, 33, 18, 28, 31, 40, 31, 30, 23, 32, 16, 31, 25, 16, 32, 31, 28, 35, 25, 26, 25, 17, 16, 29, 22, 26, 29, 20, 21, 18, 42, 25, 26, 21, 10, 20, 17, 25, 33]

第 13 回 [28, 22, 22, 22, 20, 31, 18, 26, 10, 24, 28, 25, 25, 19, 6, 26, 15, 34, 31, 30, 24, 22, 27, 27, 21, 13, 23, 16, 20, 26, 25, 33, 22, 10, 16, 20, 22, 23, 13, 17, 12, 32, 21, 31, 19, 25, 17, 17, 34, 32]

第 14 回 [40, 40, 16, 22, 35, 20, 32, 29, 31, 30, 32, 38, 19, 36, 19, 26, 35, 27, 35, 21, 33, 26, 20, 31, 37, 33, 29, 35, 32, 21, 34, 39, 21, 30, 33, 24, 27, 34, 23, 34, 31, 26, 23, 35, 36, 35, 33, 21, 35, 28]

第 15 回 [27, 25, 26, 24, 31, 30, 29, 13, 27, 33, 29, 23, 18, 26, 28, 25, 19, 35, 32, 22, 21, 32, 24, 32, 31, 36, 32, 32, 32, 26, 27, 27, 21, 30, 26, 29, 28, 27, 23, 38, 35, 22, 34, 26, 34, 31, 27, 31, 33, 16]

第 16 回 [18, 20, 30, 25, 21, 15, 33, 18, 18, 19, 25, 24, 26, 13, 30, 20, 12, 9, 27, 19, 16, 25, 25, 31, 29, 26, 11, 14, 30, 19, 22, 20, 6, 15, 18, 27, 27, 27, 28, 29, 15, 28, 18, 22, 31, 25, 6, 25, 18, 22]

第 17 回 [29, 24, 21, 31, 18, 29, 22, 12, 20, 18, 32, 22, 18, 21, 34, 25, 23, 22, 24, 20, 21, 17, 26, 24, 9, 28, 29, 30, 28, 21, 17, 12, 14, 31, 20, 26, 30, 18, 12, 17, 19, 17, 25, 11, 21, 16, 13, 11, 22, 8]

第 18 回 [35, 30, 29, 21, 24, 30, 31, 26, 23, 30, 24, 32, 30, 26, 28, 29, 32, 25, 26, 25, 26, 24, 24, 33, 31, 34, 34, 24, 30, 36, 26, 29, 24, 24, 27, 23, 29, 25, 32, 28, 20, 25, 33, 35, 16, 32, 26, 24, 23, 26]

第 19 回 [32, 24, 21, 27, 34, 28, 33, 26, 33, 34, 18, 25, 29, 31, 33, 27, 25, 38, 35, 25, 29, 30, 27, 37, 33, 30, 23, 28, 34, 26, 39, 30, 29, 30, 30, 32, 31, 42, 27, 30, 28, 25, 29, 32, 32, 28, 25, 25, 24, 24]

第 20 回 [35, 21, 26, 22, 30, 27, 21, 35, 22, 26, 22, 27, 28, 31, 26, 29, 43, 22, 34, 22, 20, 21, 40, 31, 23, 28, 32, 24, 21, 29, 34, 30, 25, 19, 28, 21, 16, 28, 22, 36, 18, 27, 15, 33, 21, 20, 40, 27, 24, 31]

第 21 回 [27, 39, 33, 26, 20, 21, 41, 22, 41, 22, 26, 33, 29, 25, 23, 32, 31, 20, 32, 29, 18, 38, 13, 30, 24, 16, 29, 21, 22, 21, 30, 21, 21, 22, 20, 20, 33, 22, 33, 28, 30, 38, 14, 22, 34, 23, 32, 23, 17, 23]

第 22 回 [27, 23, 26, 29, 31, 16, 26, 24, 25, 26, 21, 25, 23, 23, 28, 41, 30, 25, 27, 34, 33, 24, 27, 21, 37, 23, 31, 21, 20, 22, 28, 30, 30, 25, 26, 40, 34, 25, 29, 24, 33, 11, 29, 6, 25, 31, 33, 27, 42, 26]

第 23 回 [32, 25, 42, 45, 33, 36, 30, 25, 29, 39, 26, 29, 28, 18, 41, 40, 21, 37, 35, 42, 33, 38, 25, 26, 32, 26, 31, 31, 16, 34, 36, 25, 36, 44, 29, 26, 38, 43, 36, 28, 37, 31, 39, 19, 24, 25, 33, 35, 30, 19]

第 24 回 [30, 30, 32, 21, 28, 29, 25, 33, 23, 28, 32, 31, 27, 35, 34, 25, 40, 34, 24, 34, 26, 27, 29, 29, 26, 27, 27, 31, 25, 29, 19, 33, 26, 25, 28, 32, 25, 23, 32, 29, 25, 33, 24, 29, 29, 37, 39, 29, 25, 34]

第 25 回 [39, 26, 32, 35, 28, 42, 19, 29, 25, 26, 33, 22, 34, 30, 27, 33, 29, 22, 30, 32, 30, 29, 34, 26, 29, 31, 28, 31, 22, 42, 25, 25, 36, 26, 37, 35, 27, 39, 36, 12, 37, 27, 20, 20, 27, 21, 30, 35, 30, 30]

第 26 回 [30, 22, 31, 28, 24, 31, 12, 28, 33, 30, 28, 18, 28, 30, 28, 30, 35, 28, 31, 29, 26, 14, 31, 30, 34, 25, 31, 36, 19, 37, 33, 21, 30, 27, 12, 36, 26, 27, 25, 31, 32, 17, 28, 28, 31, 30, 30, 15, 20, 27]

第 27 回 [25, 26, 20, 21, 33, 22, 28, 20, 27, 29, 32, 18, 31, 39, 34, 38, 31, 31, 40, 34, 27, 32, 36, 34, 18, 32, 30, 40, 39, 23, 43, 36, 26, 30, 25, 34, 46, 42, 45, 24, 40, 22, 20, 36, 38, 27, 26, 24, 34, 38]

第 28 回 [24, 32, 42, 31, 37, 27, 31, 24, 43, 37, 34, 25, 35, 28, 25, 29, 39, 36, 28, 42, 37, 32, 36, 24, 28, 38, 22, 23, 30, 33, 35, 32, 39, 36, 33, 36, 41, 41, 36, 29, 44, 31, 37, 34, 30, 40, 29, 31, 42, 38]

第 29 回 [30, 37, 30, 34, 34, 27, 35, 36, 37, 31, 28, 27, 29, 33, 30, 26, 39, 27, 35, 27, 30, 31, 30, 30, 32, 35, 22, 28, 25, 36, 30, 29, 36, 30, 29, 27, 35, 34, 30, 37, 28, 23, 31, 36, 37, 33, 35, 48, 35, 42]

第 30 回 [34, 26, 29, 20, 38, 25, 31, 28, 25, 34, 33, 23, 23, 23, 28, 23, 41, 31, 22, 39, 33, 28, 30, 26, 20, 24, 30, 18, 24, 29, 37, 35, 29, 35, 41, 35, 31, 21, 26, 24, 30, 23, 31, 29, 28, 23, 23, 31, 26, 32]

第 31 回 [26, 26, 31, 25, 34, 31, 24, 36, 22, 31, 29, 43, 29, 30, 24, 27, 27, 29, 18, 28, 26, 30, 34, 27, 27, 19, 30, 24, 28, 36, 27, 22, 31, 25, 19, 38, 38, 32, 33, 30, 34, 30, 38, 28, 25, 22, 26, 26, 19, 25]

第 32 回 [29, 32, 24, 27, 32, 34, 22, 22, 22, 24, 34, 28, 30, 35, 24, 24, 34, 31, 34, 29, 32, 36, 33, 31, 28, 28, 34, 20, 29, 30, 26, 39, 28, 24, 21, 34, 30, 28, 34, 28, 32, 24, 37, 28, 35, 32, 32, 30, 32, 21]

第 33 回 [24, 37, 26, 30, 20, 32, 34, 26, 26, 36, 33, 23, 26, 32, 37, 30, 35, 22, 31, 28, 28, 30, 33, 32, 25, 34, 27, 36, 28, 31, 29, 23, 28, 25, 36, 15, 37, 29, 25, 26, 19, 33, 28, 24, 29, 32, 27, 26, 32, 28]

第 34 回 [30, 23, 36, 18, 33, 29, 32, 21, 39, 26, 38, 37, 30, 41, 18, 26, 38, 32, 35, 35, 38, 28, 32, 36, 16, 29, 32, 22, 36, 36, 38, 32, 38, 33, 28, 23, 37, 30, 17, 32, 27, 36, 40, 41, 36, 31, 28, 39, 34, 38]

第 35 回 [35, 34, 34, 37, 34, 12, 30, 31, 41, 32, 29, 31, 29, 30, 32, 30, 28, 30, 28, 13, 31, 24, 37, 33, 34, 29, 41, 31, 31, 28, 31, 31, 36, 25, 36, 22, 26, 32, 35, 27, 28, 32, 37, 34, 34, 23, 29, 38, 35, 30]

第 36 回 [32, 28, 28, 21, 31, 17, 27, 28, 30, 24, 35, 21, 27, 30, 24, 19, 32, 41, 20, 28, 7, 36, 22, 33, 18, 23, 28, 23, 21, 28, 27, 29, 20, 20, 26, 35, 22, 11, 30, 21, 29, 31, 12, 22, 20, 24, 27, 21, 16, 32]

第 37 回 [12, 13, 27, 20, 31, 16, 20, 24, 19, 38, 19, 38, 25, 19, 29, 6, 22, 14, 30, 24, 29, 14, 27, 31, 32, 29, 33, 28, 29, 26, 22, 35, 27, 13, 24, 27, 31, 42, 41, 38, 16, 28, 14, 24, 17, 7, 18, 5, 14, 32]

第 38 回 [24, 26, 38, 31, 31, 25, 42, 38, 33, 37, 30, 33, 28, 38, 41, 21, 37, 30, 26, 40, 44, 39, 34, 32, 36, 29, 24, 33, 24, 21, 34, 37, 32, 37, 30, 38, 35, 37, 33, 39, 36, 26, 30, 28, 35, 39, 36, 38, 27, 37]

第 39 回 [30, 34, 33, 30, 29, 34, 39, 27, 30, 26, 33, 34, 17, 34, 36, 35, 19, 33, 31, 43, 13, 13, 16, 31, 26, 29, 25, 34, 33, 34, 31, 37, 19, 8, 37, 35, 21, 26, 13, 27, 24, 28, 30, 17, 31, 26, 15, 29, 25, 18]

第 40 回 [28, 27, 30, 19, 26, 22, 24, 29, 32, 22, 36, 22, 32, 26, 27, 30, 26, 34, 27, 32, 35, 30, 29, 27, 31, 22, 34, 31, 24, 39, 21, 23, 26, 33, 15, 30, 27, 25, 34, 31, 22, 33, 27, 32, 28, 33, 29, 37, 24, 29]

第 41 回 [21, 20, 28, 31, 24, 35, 41, 24, 18, 24, 22, 35, 26, 27, 21, 31, 29, 24, 32, 25, 33, 31, 22, 28, 24, 32, 36, 35, 29, 27, 38, 22, 32, 18, 21, 28, 28, 21, 29, 27, 21, 19, 36, 24, 41, 23, 31, 31, 36, 27]

第 42 回 [25, 28, 34, 41, 29, 43, 40, 37, 34, 33, 33, 34, 28, 24, 24, 27, 21, 22, 33, 30, 33, 36, 35, 34, 27, 29, 36, 34, 31, 24, 28, 36, 32, 33, 27, 19, 28, 28, 26, 37, 29, 28, 34, 30, 26, 24, 26, 27, 36, 22]

第 43 回 [37, 25, 38, 30, 20, 30, 29, 34, 31, 37, 24, 29, 40, 35, 27, 33, 26, 30, 25, 30, 34, 31, 35, 36, 32, 29, 39, 29, 32, 32, 34, 38, 26, 32, 30, 35, 33, 26, 33, 28, 30, 41, 24, 37, 29, 43, 28, 33, 27, 28]

第 44 回 [35, 38, 34, 26, 31, 33, 28, 32, 36, 30, 32, 28, 36, 31, 32, 18, 27, 24, 20, 31, 24, 36, 28, 32, 30, 28, 35, 23, 30, 33, 28, 32, 31, 33, 30, 31, 23, 35, 34, 29, 30, 28, 33, 23, 27, 28, 28, 31, 33, 30]

第 45 回 [32, 26, 34, 29, 30, 35, 24, 33, 28, 25, 34, 29, 40, 31, 42, 25, 31, 36, 29, 24, 36, 25, 28, 28, 31, 30, 28, 29, 22, 28, 34, 34, 27, 35, 29, 26, 41, 34, 32, 28, 36, 30, 35, 34, 31, 38, 27, 29, 36, 33]

第 46 回 [33, 38, 34, 32, 29, 23, 34, 37, 19, 33, 20, 32, 29, 22, 39, 21, 33, 28, 21, 21, 24, 18, 37, 24, 28, 30, 26, 19, 30, 27, 38, 36, 32, 29, 33, 29, 25, 35, 21, 33, 34, 28, 26, 19, 33, 32, 36, 27, 30, 35]

第 47 回 [8, 25, 16, 27, 36, 29, 35, 22, 22, 33, 28, 22, 32, 22, 20, 14, 22, 30, 15, 35, 35, 19, 22, 30, 20, 35, 32, 15, 27, 21, 27, 20, 23, 27, 22, 36, 25, 21, 26, 38, 34, 13, 29, 23, 31, 34, 28, 25, 18, 17]

第 48 回 [33, 23, 28, 35, 24, 27, 29, 28, 36, 31, 15, 30, 35, 27, 33, 42, 34, 34, 40, 29, 26, 23, 24, 24, 35, 30, 29, 24, 27, 28, 22, 27, 33, 34, 35, 32, 31, 25, 29, 37, 39, 34, 28, 24, 34, 26, 25, 23, 42, 37]

第 49 回 [29, 17, 11, 33, 26, 22, 23, 20, 14, 34, 23, 21, 24, 7, 21, 29, 16, 9, 27, 18, 28, 26, 21, 20, 16, 16, 12, 32, 28, 30, 19, 23, 20, 23, 16, 27, 24, 14, 28, 19, 29, 31, 17, 26, 36, 5, 32, 20, 21, 31]

第 50 回 [18, 28, 30, 27, 25, 29, 31, 34, 41, 27, 23, 33, 24, 32, 20, 32, 20, 27, 37, 33, 33, 32, 27, 33, 24, 29, 30, 26, 31, 27, 25, 30, 30, 32, 28, 21, 36, 28, 27, 27, 34, 35, 32, 31, 23, 29, 22, 24, 19, 23]

第 51 回 [28, 28, 20, 37, 21, 25, 26, 35, 28, 28, 26, 31, 28, 31, 19, 32, 35, 27, 28, 18, 25, 28, 28, 30, 28, 25, 22, 24, 33, 32, 25, 27, 24, 16, 25, 24, 30, 23, 27, 29, 21, 22, 22, 26, 27, 28, 31, 36, 30, 33]

第 52 回 [16, 22, 11, 11, 22, 15, 38, 26, 21, 24, 23, 31, 21, 20, 13, 30, 20, 22, 34, 16, 29, 23, 19, 21, 30, 16, 33, 25, 25, 35, 20, 24, 24, 30, 21, 35, 30, 27, 10, 29, 21, 21, 8, 30, 13, 29, 29, 14, 13, 20]

第 53 回 [32, 33, 26, 28, 33, 32, 34, 36, 23, 32, 33, 23, 23, 30, 29, 32, 32, 39, 29, 31, 22, 31, 37, 25, 29, 35, 33, 33, 23, 30, 30, 34, 28, 38, 27, 37, 34, 27, 34, 32, 37, 30, 34, 36, 36, 30, 23, 31, 35, 39]

第 54 回 [23, 34, 29, 29, 39, 38, 27, 36, 32, 27, 34, 22, 39, 21, 41, 31, 31, 25, 33, 36, 38, 42, 42, 23, 26, 38, 28, 37, 26, 26, 33, 34, 33, 39, 23, 33, 24, 29, 30, 25, 27, 35, 29, 35, 26, 34, 35, 18, 23, 28]

第 55 回 [31, 26, 26, 26, 33, 34, 28, 39, 21, 30, 30, 36, 28, 40, 30, 33, 30, 28, 32, 42, 34, 34, 22, 33, 24, 28, 26, 27, 32, 29, 28, 33, 26, 28, 33, 32, 35, 27, 23, 37, 29, 25, 29, 35, 37, 30, 33, 27, 26, 33]

第 56 回 [38, 21, 29, 33, 25, 22, 29, 33, 29, 24, 24, 27, 25, 20, 26, 25, 26, 32, 32, 22, 28, 25, 34, 36, 35, 29, 44, 31, 20, 27, 26, 36, 24, 33, 37, 28, 34, 24, 31, 29, 29, 18, 36, 17, 26, 36, 31, 25, 42, 24]

第 57 回 [23, 30, 30, 32, 29, 26, 28, 26, 28, 27, 25, 44, 25, 32, 25, 26, 25, 31, 44, 28, 34, 20, 31, 18, 19, 24, 27, 15, 32, 33, 22, 28, 27, 21, 23, 22, 19, 23, 28, 26, 28, 26, 23, 30, 29, 43, 24, 26, 21, 17]

第 58 回 [19, 20, 20, 24, 31, 30, 33, 26, 30, 26, 36, 31, 25, 33, 37, 26, 24, 21, 24, 39, 25, 29, 28, 25, 27, 34, 25, 30, 25, 33, 28, 29, 25, 40, 25, 26, 26, 32, 18, 24, 23, 31, 38, 22, 30, 30, 26, 26, 30, 29]

第 59 回 [24, 34, 24, 28, 31, 30, 28, 38, 29, 26, 31, 37, 28, 23, 43, 35, 32, 34, 29, 32, 31, 19, 24, 29, 37, 23, 28, 33, 36, 26, 28, 28, 34, 34, 32, 33, 31, 36, 28, 29, 38, 32, 36, 29, 19, 24, 28, 27, 33, 31]

第 60 回 [24, 28, 24, 34, 23, 44, 25, 30, 27, 33, 25, 36, 24, 21, 29, 28, 24, 30, 30, 32, 31, 33, 29, 34, 32, 34, 30, 35, 27, 23, 28, 37, 27, 36, 31, 36, 25, 33, 33, 42, 33, 31, 25, 30, 42, 33, 22, 35, 32, 31]

第 61 回 [24, 25, 30, 22, 35, 30, 22, 24, 23, 25, 19, 27, 23, 17, 18, 30, 34, 23, 22, 26, 25, 22, 22, 45, 31, 20, 20, 34, 27, 26, 26, 25, 18, 14, 25, 25, 24, 29, 41, 24, 24, 27, 27, 29, 30, 22, 32, 29, 22, 36]

第 62 回 [23, 16, 23, 26, 29, 17, 30, 26, 24, 26, 17, 22, 23, 26, 35, 24, 25, 31, 32, 33, 34, 32, 25, 17, 27, 31, 15, 30, 33, 27, 24, 19, 33, 32, 28, 23, 28, 23, 21, 30, 26, 28, 36, 24, 31, 28, 32, 32, 30, 17]

第 63 回 [34, 30, 30, 29, 31, 29, 34, 34, 35, 15, 24, 39, 27, 31, 34, 41, 20, 30, 33, 31, 37, 33, 23, 33, 33, 26, 21, 27, 35, 29, 15, 24, 25, 30, 40, 12, 31, 37, 29, 40, 37, 31, 25, 23, 35, 29, 34, 38, 34, 27]

第 64 回 [31, 30, 37, 33, 42, 23, 42, 33, 28, 19, 32, 42, 20, 19, 28, 30, 41, 34, 38, 22, 30, 28, 38, 26, 18, 25, 35, 27, 24, 27, 27, 34, 36, 34, 36, 31, 41, 31, 21, 38, 27, 24, 27, 36, 33, 26, 30, 42, 25, 25]

第 65 回 [20, 30, 21, 29, 21, 34, 28, 34, 23, 25, 37, 22, 23, 29, 21, 24, 37, 20, 20, 21, 33, 23, 17, 31, 18, 35, 19, 27, 20, 31, 17, 24, 24, 29, 25, 27, 37, 36, 27, 32, 35, 30, 29, 26, 33, 17, 39, 21, 25, 36]

第 66 回 [32, 28, 32, 30, 39, 27, 35, 33, 33, 33, 29, 34, 31, 29, 43, 41, 36, 36, 30, 20, 33, 29, 36, 40, 32, 29, 42, 39, 23, 34, 33, 36, 42, 31, 29, 36, 22, 31, 32, 42, 37, 30, 19, 32, 25, 44, 37, 35, 40, 39]

第 67 回 [27, 32, 27, 23, 21, 29, 28, 41, 24, 30, 31, 35, 38, 25, 24, 26, 22, 27, 40, 31, 25, 34, 25, 24, 36, 26, 29, 39, 26, 32, 28, 19, 31, 33, 26, 27, 22, 40, 28, 31, 25, 20, 29, 29, 33, 34, 18, 30, 25, 40]

第 68 回 [32, 33, 22, 25, 29, 20, 29, 30, 41, 32, 32, 33, 22, 25, 32, 34, 26, 19, 32, 19, 32, 27, 22, 34, 23, 32, 34, 30, 34, 34, 29, 26, 26, 23, 29, 27, 25, 32, 35, 35, 22, 31, 29, 22, 21, 27, 34, 23, 24, 30]

第 69 回 [36, 38, 21, 32, 31, 16, 18, 29, 16, 25, 21, 30, 32, 17, 4, 22, 38, 27, 18, 22, 27, 31, 31, 15, 24, 38, 14, 20, 32, 29, 18, 23, 29, 29, 27, 23, 36, 27, 31, 22, 26, 29, 27, 29, 32, 30, 29, 35, 21, 28]

第 70 回 [35, 27, 41, 31, 21, 32, 34, 23, 36, 28, 42, 26, 27, 38, 40, 27, 29, 37, 47, 35, 36, 28, 39, 16, 37, 27, 31, 38, 29, 31, 38, 25, 26, 25, 32, 26, 31, 31, 30, 36, 33, 27, 30, 34, 20, 33, 45, 28, 37, 31]

第 71 回 [30, 30, 22, 37, 25, 23, 30, 38, 25, 19, 31, 37, 37, 18, 31, 23, 33, 35, 34, 19, 45, 28, 37, 45, 31, 24, 31, 26, 18, 38, 27, 35, 23, 25, 25, 27, 28, 33, 32, 34, 30, 32, 33, 34, 26, 24, 28, 29, 34, 31]

第 72 回 [32, 41, 29, 28, 29, 35, 28, 17, 30, 20, 25, 24, 32, 23, 34, 26, 22, 19, 36, 35, 30, 30, 31, 29, 33, 24, 30, 35, 32, 26, 30, 22, 27, 33, 22, 32, 32, 26, 22, 27, 27, 16, 25, 34, 32, 29, 23, 25, 19, 24]

第 73 回 [25, 26, 30, 25, 34, 31, 32, 34, 26, 35, 29, 30, 31, 28, 22, 26, 22, 29, 21, 32, 30, 35, 23, 24, 34, 20, 35, 30, 29, 28, 30, 30, 31, 22, 31, 27, 38, 24, 28, 33, 29, 30, 28, 25, 23, 30, 28, 34, 23, 27]

第 74 回 [35, 30, 20, 21, 22, 20, 30, 26, 29, 41, 30, 26, 24, 23, 32, 23, 32, 27, 24, 22, 25, 24, 30, 26, 20, 30, 22, 33, 50, 24, 22, 23, 27, 30, 27, 28, 32, 37, 16, 22, 26, 22, 32, 25, 31, 30, 23, 31, 42, 31]

第 75 回 [36, 19, 31, 38, 26, 34, 21, 29, 26, 28, 28, 22, 39, 23, 17, 32, 12, 28, 36, 22, 30, 30, 36, 33, 36, 28, 15, 32, 39, 34, 23, 24, 40, 23, 26, 32, 32, 23, 13, 22, 28, 25, 30, 26, 26, 25, 27, 33, 31, 18]

第 76 回 [30, 27, 26, 29, 28, 33, 31, 35, 24, 39, 21, 26, 30, 26, 25, 25, 30, 28, 17, 34, 32, 26, 28, 29, 31, 29, 35, 32, 36, 28, 30, 35, 29, 32, 26, 29, 27, 37, 15, 23, 25, 31, 27, 26, 35, 32, 36, 25, 27, 16]

第 77 回 [38, 22, 21, 16, 22, 19, 7, 12, 44, 6, 22, 22, 25, 38, 30, 10, 28, 34, 23, 22, 8, 33, 21, 33, 30, 18, 19, 21, 31, 22, 27, 32, 28, 29, 8, 31, 29, 26, 30, 19, 29, 24, 29, 5, 35, 14, 12, 27, 23, 18]

第 78 回 [25, 27, 17, 36, 28, 30, 16, 18, 32, 22, 17, 23, 35, 34, 33, 25, 25, 39, 16, 19, 32, 28, 18, 29, 9, 20, 23, 26, 26, 32, 24, 23, 30, 31, 28, 39, 20, 23, 30, 28, 38, 39, 17, 29, 20, 28, 24, 29, 18, 28]

第 79 回 [36, 21, 27, 34, 16, 25, 25, 28, 26, 34, 20, 24, 25, 29, 20, 18, 14, 26, 29, 30, 29, 13, 19, 24, 18, 16, 19, 19, 25, 25, 22, 16, 25, 33, 31, 25, 35, 17, 21, 16, 28, 18, 30, 37, 28, 27, 28, 23, 23, 24]

第 80 回 [23, 31, 33, 34, 33, 37, 39, 32, 34, 31, 24, 30, 29, 35, 26, 36, 37, 32, 29, 34, 23, 41, 37, 32, 24, 24, 23, 37, 24, 37, 27, 26, 29, 29, 39, 30, 32, 27, 34, 41, 21, 19, 28, 33, 25, 36, 34, 25, 31, 30]

Result 2

第 1 回: 统计量 D = 0.11061154864260975, pvalue = 0.5476782455850001

第 2 回: 统计量 D = 0.12767368022950915, pvalue = 0.3602677260827614

第 3 回: 统计量 D = 0.0990287845065273, pvalue = 0.7037362571480047

第 4 回: 统计量 D = 0.1561706363837247, pvalue = 0.15678869664644826

第 5 回: 统计量 D = 0.1529157165006741, pvalue = 0.17387696427290808

第 6 回: 统计量 D = 0.10361356111239739, pvalue = 0.6393197199860783

第 7 回: 统计量 D = 0.10193726862835323, pvalue = 0.6624838333939169

第 8 回: 统计量 D = 0.08104710769413481, pvalue = 0.8977860049052286

第 9 回: 统计量 D = 0.09657549966181861, pvalue = 0.7395399469810253

第 10 回: 统计量 D = 0.09517254588313118, pvalue = 0.760411972944893

第 11 回: 统计量 D = 0.10298828063181192, pvalue = 0.6479069807497692

第 12 回: 统计量 D = 0.10791340728564536, pvalue = 0.5820252102570843

第 13 回: 统计量 D = 0.07365592925023379, pvalue = 0.949042477483117

第 14 回: 统计量 D = 0.12528600120290068, pvalue = 0.38337746619255186

第 15 回: 统计量 D = 0.0938474250089239, pvalue = 0.7803817820241463

第 16 回: 统计量 D = 0.13759645153127154, pvalue = 0.27479287331873875

第 17 回: 统计量 D = 0.07434457624802304, pvalue = 0.9450978639838075

第 18 回: 统计量 D = 0.1433424209698841, pvalue = 0.23275625082330342

第 19 回: 统计量 D = 0.08366357012558712, pvalue = 0.8752177447739966

第 20 回: 统计量 D = 0.12999953594492825, pvalue = 0.338718087439033

第 21 回: 统计量 D = 0.15826951984996307, pvalue = 0.14650135471985692

第 22 回: 统计量 D = 0.11791011228587209, pvalue = 0.46116593505680775

第 23 回: 统计量 D = 0.08126986767073041, pvalue = 0.8959483083790726

第 24 回: 统计量 D = 0.11458900045331455, pvalue = 0.4993623783581191

第 25 回：统计量 D = 0.08102179708986412, pvalue = 0.89799379143308

第 26 回：统计量 D = 0.18856588105611016, pvalue = 0.04969207629783818

第 27 回：统计量 D = 0.0824540078342687, pvalue = 0.8859140004764948

第 28 回：统计量 D = 0.0971596202088898, pvalue = 0.7309339727956131

第 29 回：统计量 D = 0.13459846131801234, pvalue = 0.2988562002721789

第 30 回：统计量 D = 0.09601943341583935, pvalue = 0.7477788072587573

第 31 回：统计量 D = 0.09285903546576746, pvalue = 0.7954341639377761

第 32 回：统计量 D = 0.11528995611701431, pvalue = 0.4911371514656629

第 33 回：统计量 D = 0.07781649540189062, pvalue = 0.9225698031506534

第 34 回：统计量 D = 0.12752941600792556, pvalue = 0.36163552757018014

第 35 回：统计量 D = 0.1480407318177638, pvalue = 0.20219076659095972

第 36 回：统计量 D = 0.1099700673833306, pvalue = 0.5557301456931196

第 37 回：统计量 D = 0.0913017548950874, pvalue = 0.7987924126569736

第 38 回：统计量 D = 0.12310746488528168, pvalue = 0.40534342425901204

第 39 回：统计量 D = 0.13271606618167048, pvalue = 0.31473541771671637

第 40 回：统计量 D = 0.06465942828970284, pvalue = 0.985006473366436

第 41 回：统计量 D = 0.11649927891382661, pvalue = 0.4771519822318373

第 42 回：统计量 D = 0.1127277989419585, pvalue = 0.5216246081504285

第 43 回：统计量 D = 0.0997812815888392, pvalue = 0.6929366252907191

第 44 回：统计量 D = 0.12376565088108837, pvalue = 0.39861807903037405

第 45 回：统计量 D = 0.11157704029899657, pvalue = 0.5356945503395101

第 46 回：统计量 D = 0.13194366896403853, pvalue = 0.32142545745602974

第 47 回：统计量 D = 0.10082616074574058, pvalue = 0.6780864853326867

第 48 回：统计量 D = 0.09011386472786476, pvalue = 0.8115406052648687

第 49 回：统计量 D = 0.07773583854870636, pvalue = 0.9231424385177358

第 50 回：统计量 D = 0.09016625399462247, pvalue = 0.8109840804532484

第 51 回：统计量 D = 0.11777478432263977, pvalue = 0.46268391875878345

第 52 回：统计量 D = 0.1038439641430 12, pvalue = 0.6361716032078885

第 53 回：统计量 D = 0.11063053432568254, pvalue = 0.5474410271140888

第 54 回：统计量 D = 0.09507693386127647, pvalue = 0.7618446408176999

第 55 回：统计量 D = 0.09650150785031408, pvalue = 0.7406336457553946

第 56 回：统计量 D = 0.10257727796625371, pvalue = 0.6535860918538129

第 57 回：统计量 D = 0.1252801297059032, pvalue = 0.3834355358567662

第 58 回：统计量 D = 0.1442963272282206, pvalue = 0.22628043315798607

第 59 回：统计量 D = 0.10534325951617537, pvalue = 0.6158998667118634

第 60 回：统计量 D = 0.09107230979671194, pvalue = 0.8012753439388189

第 61 回：统计量 D = 0.1323963499944819, pvalue = 0.3174922367616954

第 62 回：统计量 D = 0.1030730792216944, pvalue = 0.6467386694164079

第 63 回：统计量 D = 0.133658653939491, pvalue = 0.30670905804420506

第 64 回：统计量 D = 0.08732655505695924, pvalue = 0.8403090203284765

第 65 回：统计量 D = 0.10403252976110133, pvalue = 0.6336016168503376

第 66 回：统计量 D = 0.0948699698765525, pvalue = 0.7649502411491769

第 67 回：统计量 D = 0.09301995685872155, pvalue = 0.7929744662061884

第 68 回：统计量 D = 0.14166847226273382, pvalue = 0.24446126703670215

第 69 回: 统计量 D = 0.1304919052821641, pvalue = 0.3342768798694518

第 70 回: 统计量 D = 0.08536366695814401, pvalue = 0.8594638173938903

第 71 回: 统计量 D = 0.07280437134060341, pvalue = 0.9536784952430427

第 72 回: 统计量 D = 0.08575991852190268, pvalue = 0.8556772876046093

第 73 回: 统计量 D = 0.09518344731453521, pvalue = 0.7602487059663332

第 74 回: 统计量 D = 0.11885782372000286, pvalue = 0.45062708725797845

第 75 回: 统计量 D = 0.057999632246624466, pvalue = 0.9960124218630027

第 76 回: 统计量 D = 0.11567899047541225, pvalue = 0.48660977819123974

第 77 回: 统计量 D = 0.09248089079595662, pvalue = 0.7858873113572697

第 78 回: 统计量 D = 0.08762884850765584, pvalue = 0.8372736888425902

第 79 回: 统计量 D = 0.07850248576442703, pvalue = 0.9176068646592311

第 80 回: 统计量 D = 0.07469303874909847, pvalue = 0.9430352876820735

Result 3

第 1 到 5 回对高频字使用密度的单因素方差分析

方差来源	平方和	自由度	均方	F 比	F 临界值
因素	2385.6240000000107	4	596.4060000000027	17.765241908226468	2.4084883700149953
误差	8225.01999999999	245	33.57151020408159	_	_
总和	10610.644	249	_	_	_

第 6 到 10 回对高频字使用密度的单因素方差分析

方差来源	平方和	自由度	均方	F 比	F 临界值
因素	1510.1839999999793	4	377.5459999999948	14.098273129095983	2.4084883700149953
误差	6561.0	245	26.779591836734692	_	_
总和	8071.183999999979	249	_	_	_

第 11 到 15 回对高频字使用密度的单因素方差分析

方差来源	平方和	自由度	均方	F 比	F 临界值
因素	1448.3440000000119	4	362.08600000000297	9.656740804925597	2.4084883700149953
误差	9186.440000000002	245	37.49567346938777	_	_
总和	10634.784000000014	249	_	_	_

第 16 到 20 回对高频字使用密度的单因素方差分析

方差来源	平方和	自由度	均方	F 比	F 临界值

+----------+--------------------+--------+--------------------+-------------------+--------------------+

| 因素 | 2731.8559999999998 | 4 | 682.9639999999999 | 20.48617736074515 | 2.4084883700149953 |

| 误差 | 8167.760000000009 | 245 | 33.337795918367384 | _ | _ |

| 总和 | 10899.616000000009 | 249 | _ | _ | _ |

+----------+--------------------+--------+--------------------+-------------------+--------------------+

第 21 到 25 回对高频字使用密度的单因素方差分析

+----------+--------------------+--------+--------------------+-------------------+--------------------+

| 方差来源 | 平方和 | 自由度 | 均方 | F 比 | F 临界值 |

+----------+--------------------+--------+--------------------+-------------------+--------------------+

| 因素 | 915.2240000000456 | 4 | 228.8060000000114 | 5.744087605900809 | 2.4084883700149953 |

| 误差 | 9759.159999999974 | 245 | 39.833306122448874 | _ | _ |

| 总和 | 10674.38400000002 | 249 | _ | _ | _ |

+----------+--------------------+--------+--------------------+-------------------+--------------------+

第 26 到 30 回对高频字使用密度的单因素方差分析

+----------+--------------------+--------+--------------------+-------------------+--------------------+

| 方差来源 | 平方和 | 自由度 | 均方 | F 比 | F 临界值 |

+----------+--------------------+--------+--------------------+-------------------+--------------------+

| 因素 | 1217.616000000009 | 4 | 304.40400000000227 | 8.461249940436957 | 2.4084883700149953 |

| 误差 | 8814.179999999993 | 245 | 35.976244897959155 | _ | _ |

| 总和 | 10031.796000000002 | 249 | _ | _ | _ |

+----------+--------------------+--------+--------------------+-------------------+--------------------+

第 31 到 35 回对高频字使用密度的单因素方差分析

+----------+--------------------+--------+--------------------+-------------------+--------------------+

| 方差来源 | 平方和 | 自由度 | 均方 | F 比 | F 临界值 |

+----------+--------------------+--------+--------------------+-------------------+--------------------+

| 因素 | 367.57600000000093 | 4 | 91.89400000000023 | 3.0677161255862497 | 2.4084883700149953 |

| 误差 | 7339.020000000019 | 245 | 29.955183673469463 | _ | _ |

| 总和 | 7706.59600000002 | 249 | _ | _ | _ |

+----------+--------------------+--------+--------------------+-------------------+--------------------+

第 36 到 40 回对高频字使用密度的单因素方差分析

+----------+--------------------+--------+--------------------+-------------------+--------------------+

| 方差来源 | 平方和 | 自由度 | 均方 | F 比 | F 临界值 |

+----------+--------------------+--------+--------------------+-------------------+--------------------+

| 因素 | 2425.4800000000105 | 4 | 606.3700000000026 | 12.449126067799547 | 2.4084883700149953 |

| 误差 | 11933.420000000013 | 245 | 48.70783673469393 | _ | _ |

| 总和 | 14358.900000000023 | 249 | _ | _ | _ |

+----------+--------------------+--------+--------------------+-------------------+--------------------+

第 41 到 45 回对高频字使用密度的单因素方差分析

+----------+--------------------+--------+--------------------+-------------------+--------------------+

方差来源	平方和	自由度	均方	F 比	F 临界值
因素	414.2960000000603	4	103.57400000001508	4.101443348958127	2.4084883700149953
误差	6186.999999999942	245	25.25306122448956	_	_
总和	6601.296000000002	249	_	_	_

第 46 到 50 回对高频字使用密度的单因素方差分析

方差来源	平方和	自由度	均方	F 比	F 临界值
因素	2008.3359999999811	4	502.0839999999953	13.016083532790176	2.4084883700149953
误差	9450.660000000003	245	38.57412244897961	_	_
总和	11458.995999999985	249	_	_	_

第 51 到 55 回对高频字使用密度的单因素方差分析

方差来源	平方和	自由度	均方	F 比	F 临界值
因素	2561.0560000000405	4	640.2640000000101	21.137601871415985	2.4084883700149953
误差	7421.119999999995	245	30.290285714285694	_	_
总和	9982.176000000036	249	_	_	_

第 56 到 60 回对高频字使用密度的单因素方差分析

方差来源	平方和	自由度	均方	F 比	F 临界值
因素	465.3839999999909	4	116.34599999999773	3.89291362455641	2.4084883700149953
误差	7322.220000000001	245	29.886612244897965	_	_
总和	7787.603999999992	249	_	_	_

第 61 到 65 回对高频字使用密度的单因素方差分析

方差来源	平方和	自由度	均方	F 比	F 临界值
因素	916.5759999999718	4	229.14399999999296	6.057850347887444	2.4084883700149953
误差	9267.360000000015	245	37.82595918367353	_	_
总和	10183.935999999987	249	_	_	_

第 66 到 70 回对高频字使用密度的单因素方差分析

+----------+--------------------+--------+--------------------+--------------------+--------------------+

| 方差来源 | 平方和 | 自由度 | 均方 | F 比 | F 临界值 |

+----------+--------------------+--------+--------------------+--------------------+--------------------+

| 因素 | 1583.2400000000198 | 4 | 395.81000000000495 | 10.843673053139751 | 2.4084883700149953 |

| 误差 | 8942.859999999986 | 245 | 36.501469387755044 | _ | _ |

| 总和 | 10526.100000000006 | 249 | _ | _ | _ |

+----------+--------------------+--------+--------------------+--------------------+--------------------+

第 71 到 75 回对高频字使用密度的单因素方差分析

+----------+--------------------+--------+--------------------+--------------------+--------------------+

| 方差来源 | 平方和 | 自由度 | 均方 | F 比 | F 临界值 |

+----------+--------------------+--------+--------------------+--------------------+--------------------+

| 因素 | 169.01600000000326 | 4 | 42.254000000000815 | 1.2446952536226872 | 2.4084883700149953 |

| 误差 | 8317.080000000016 | 245 | 33.94726530612252 | _ | _ |

| 总和 | 8486.09600000002 | 249 | _ | _ | _ |

+----------+--------------------+--------+--------------------+--------------------+--------------------+

第 76 到 80 回对高频字使用密度的单因素方差分析

+----------+--------------------+--------+--------------------+--------------------+--------------------+

| 方差来源 | 平方和 | 自由度 | 均方 | F 比 | F 临界值 |

+----------+--------------------+--------+--------------------+--------------------+--------------------+

| 因素 | 1816.1360000000277 | 4 | 454.0340000000069 | 10.381920217422714 | 2.4084883700149953 |

| 误差 | 10714.619999999995 | 245 | 43.73314285714284 | _ | _ |

| 总和 | 12530.756000000023 | 249 | _ | _ | _ |

+----------+--------------------+--------+--------------------+--------------------+--------------------+

Result 4

OLS Regression Results

==

Dep. Variable:	y	R-squared:	0.997
Model:	OLS	Adj. R-squared:	0.996
Method:	Least Squares	F-statistic:	1536.
Date:	Wed, 30 Dec 2020	Prob (F-statistic):	1.56e-76
Time:	18:29:02	Log-Likelihood:	-481.52
No. Observations:	80	AIC:	991.0
Df Residuals:	66	BIC:	1024.
Df Model:	13		
Covariance Type:	nonrobust		

==

	coef	std err	t	P>|t|	[0.025	0.975]
const	-78.7249	55.702	-1.413	0.162	-189.938	32.489
x1	6.9737	0.721	9.668	0.000	5.534	8.414

x2	6.6617	0.509	13.097	0.000	5.646	7.677
x3	6.1522	0.968	6.353	0.000	4.219	8.086
x4	8.3921	0.651	12.896	0.000	7.093	9.691
x5	7.7706	1.195	6.503	0.000	5.385	10.156
x6	10.9186	1.392	7.843	0.000	8.139	13.698
x7	10.7674	1.997	5.392	0.000	6.781	14.754
x8	16.3793	1.946	8.419	0.000	12.495	20.264
x9	17.9089	2.641	6.780	0.000	12.635	23.183
x10	14.5182	3.019	4.809	0.000	8.491	20.545
x11	15.9953	4.208	3.801	0.000	7.593	24.398
x12	17.1760	3.768	4.558	0.000	9.652	24.700
x13	26.7480	5.159	5.185	0.000	16.447	37.049

==========

Omnibus:	1.192	Durbin-Watson:	1.662
Prob(Omnibus):	0.551	Jarque-Bera (JB):	0.859
Skew:	0.252	Prob(JB):	0.651
Kurtosis:	3.068	Cond. No.	1.35e+03

==========

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The condition number is large, 1.35e+03. This might indicate that there are

strong multicollinearity or other numerical problems.

Result 5

通过 1-40 回与 41-80 回的单因素双水平方差分析验证前 80 回是曹雪芹所著

方差来源	平方和	自由度	均方	F 比	F 临界值
因素	0.001419950848292828	1	0.001419950848292828	1.7381524035983091	3.9634720513960966
误差	0.06372062998477812	78	0.0008169311536510016	_	_
总和	0.06514058083307095	79	_	_	_

通过 1-40 回与 81-120 回的单因素双水平方差分析验证后 40 回非曹雪芹所著

方差来源	平方和	自由度	均方	F 比	F 临界值
因素	0.014335265614828785	1	0.014335265614828785	19.1338681936802	3.9634720513960966
误差	0.05843829938820022	78	0.0007492089665153875	_	_
总和	0.072773565003029	79	_	_	_

两组样本的效应尺度为 1.007350521482138

差异较大

效应尺度 = 1.007350521482138

通过 41-80 回与 81-120 回的单因素双水平方差分析验证后 40 回非曹雪芹所著

方差来源	平方和	自由度	均方	F 比	F 临界值
因素	0.006731830719343357	1	0.006731830719343357	14.356270284431172	3.9634720513960966
误差	0.03657515397144717	78	0.0004689122304031688	_	_
总和	0.04330698469079053	79	_	_	_

两组样本的效应尺度为 0.8494681628275679

差异较大

效应尺度 =0.8494681628275679

通过 1-80 回与 81-120 回的单因素双水平方差分析验证后 40 回非曹雪芹所著

方差来源	平方和	自由度	均方	F 比	F 临界值
因素	0.013571413940018928	1	0.013571413940018928	19.82283032154969	3.921478181240644
误差	0.08078699252050292	118	0.0006846355298347705	_	_
总和	0.09435840646052185	119	_	_	_

两组样本的效应尺度为 0.9256109576954272

差异较大

效应尺度 =0.9256109576954272

Result 6

通过 1-40 回与 41-80 回的秩和检验验证前 80 回是曹雪芹所著

秩相同的组的组数为: 0

第一样本的秩和 R1 的观察值为 r1= 1524

拒绝域为: r1 < 1449.0617968263664, r1 > 1790.9382031736336

两个总体的数据无显著差异

通过 1-40 回与 81-120 回的秩和检验验证后 80 回是曹雪芹所著

秩相同的组的组数为: 0

第一样本的秩和 R1 的观察值为 r1= 1202

拒绝域为: r1 < 1449.0617968263664, r1 > 1790.9382031736336

两个总体的数据有显著差异

通过 41-80 回与 81-120 回的秩和检验验证后 80 回是曹雪芹所著

秩相同的组的组数为: 0

第一样本的秩和 R1 的观察值为 r1= 1212

拒绝域为: r1 < 1449.0617968263664, r1 > 1790.9382031736336

两个总体的数据有显著差异

通过 1-80 回与 81-120 回的秩和检验验证后 80 回是曹雪芹所著

秩相同的组的组数为: 0

第一样本的秩和 R1 的观察值为 r1= 3246

拒绝域为: r1 < 2124.5361802428615, r1 > 2715.4638197571385

两个总体的数据有显著差异
