暑期培训作业1

缺失值和异常值的处理问题

队员1：鲁哲豪

队员2：姜铮

队员3：许展豪

摘 要

处理数据相关问题时，其精度对于最终求解结果有较大影响，所以应保证数据尽量精确。但是因为测量机械出现故障或者错误以及采集数据人员错误等因素，所获取的数据集都可能出现数据缺失或者数据异常等问题，本文研究了解决上述问题的方法，并选用了南京信息工程大学第十六届数学建模竞赛的B题-蘑菇有毒无毒的分类与识别的附件数据进行建模。

针对问题一，对于缺失值的处理，有删除和插补两种方法。删除法分为直接删除法和权重删除法，插补法则包含均值插补法、同类均值插补法、极大似然估计法。在附件中发现stem-hight和stem-width两列的最小值为0，我们认定这1059个数据为缺失值。由于缺失数据仅占总数据量的1.7%，我们分别使用简单删除法、均值插补法和同类均值插补法处理缺失值，使数据完整。

针对问题二，鉴别异常数据常用以下三种方法：3原则、简单统计量分析和箱线图法。先使用箱线图法处理数据，结果如下：cap-diameter异常数据的占比为3.93%，stem-heigh的异常数据占比为5.19%，stem-width的异常数据占比为3.32%。在进行Box-Cox变换后，cap-diameter异常数据占比为0.94%，stem-height异常数据占比为4.39%，stem-width异常数据占比为0.60%，异常数据占比明显下降。

针对问题三，处理异常数据的常用方法有：删除、视为缺失值、平均值修正、盖帽法和分箱法。由问题二的结果可得，异常数据占比非常小，可以采用直接删除法。去除掉异常数据后，观察概率分布，总体仍然近似服从正态分布。接着我们选用盖帽法处理异常数据，与直接删除法对比，盖帽法减少了对概率分布起到了一些不良影响。

关键词：箱线图法 Box-Cox变换 盖帽法

1. 问题重述

在处理与数据有关的问题的时候，数据的准确性对最终的求解结果有着很大的影响，因此要确保数据尽可能的准确。然而由于测量机械存在故障或误差和收集数据的人的失误等等原因，得到的数据集可能会有数据缺失或数据异常的情况，所以我们需要解决下列三个问题：

1. 给出缺失数据的算法
2. 给出异常数据的鉴别算法
3. 给出异常数据的修正算法
4. 问题分析

数据缺失和数据异常是数据处理时常常要面对的两大问题。数据缺失产生的主要原因分为机械原因和人为原因。机械原因是由于机械原因导致的[数据](https://link.zhihu.com/?target=http%3A//cda.pinggu.org/view/20393.html" \t "_blank)收集或保存的失败造成的数据缺失，比如数据存储的失败，存储器损坏，机械故障导致某段时间数据未能收集。人为原因是由于人的主观失误、历史局限或有意隐瞒造成的数据缺失，比如，在市场调查中被访人拒绝透露相关问题的答案，或者回答的问题是无效的，数据录入人员失误漏录了数据。

对于数据缺失可以采用以下的几种方法来进行处理[1]：1.删除法 2.可能值插补法 。删除法又包括简单删除法和权重法；可能值插补法包括了均值插补，利用同类均值插补，极大似然估计和多重插补等方法。

而数据异常，则是指数据集中存在的不合理的值，这些值是一些偏离正常范围的值，在大样本数据的收集过程中，难免会出现一些异常值，这些异常值虽然不多，但又会对实际问题的分析产生影响。因此，找出数据集中的异常值并对其进行处理是非常有必要的。对于异常数据的检测鉴别可以通过[2-3]：1.简单统计分析 2.3原则 3.箱线图等方法来进行处理。而对于检测出的异常值，都要对其进行一些处理，来避免其对于结果造成的不良影响。主要方法有[2-3]：1.直接删除 2.视作缺失数据，然后采用处理缺失数据的方法来处理 3.平均值修正 4.盖帽法 5.分箱法 6.回归插补法 7.多重插补法 等方法

1. 模型假设

1.选用的数据除缺失值与异常值外都是准确的。

1. 数据来源

我们选用了南京信息工程大学第十六届数学建模竞赛的B题-蘑菇有毒无毒的分类与识别。此题的附件secondary\_data.csv给出了61069个蘑菇的各种数据，这些数据中存在着缺失值与异常值，而这正是我们所需要的。

1. 模型建立与求解

5.1.1问题一模型准备

数据缺失在数据处理中是一类非常常见的问题。缺失值从缺失的分布来讲可以分为完全随机缺失，随机缺失和完全非随机缺失。完全随机缺失（MCAR）指的是数据的缺失是随机的，数据的缺失不依赖于任何不完全变量或完全变量。随机缺失(MAR)指的是数据的缺失不是完全随机的，即该类数据的缺失依赖于其他完全变量。完全非随机缺失(MNAR)指的是数据的缺失依赖于不完全变量自身。[4]

对于缺失值的处理，目前的技术已经相当成熟，主要分为删除有缺失值的样本和缺失值的插补。

删除含有缺失值的样本又主要包含直接删除法和权重删除法。直接删除法即简单粗暴地将存在缺失值的个案删除，当样本容量非常大，而含有缺失值的个案很小的时候，直接删除法是一种非常易于操作且准确度较高的方法。而当缺失值的类型为非完全随机缺失时，可以对完整的数据进行加权来减小误差，其主要思路为，首先对缺失数据的案例进行标记，然后对完整数据的案例赋予不同的权重，权重可以通过logistics或probit回归求得，最后依据所得到的权重对含有缺失值的个案进行删除。

第二种方法是对缺失值进行插补，一般而言，一个数据通常含有几十甚至是几百个不同的属性，如果仅仅因为某一属性的值缺失就删除了整个数据，是对信息的极大浪费，所以我们选择利用统计学中的原理来对缺失值进行插补。我们选择了下列三种方法：

1.均值插补。数据依据其属性可以分为定距型数据和非定距型数据，对于定距型数据，前后两组数据之差是恒定的，因此对于缺失的数据，很容易想到用其前后两组数据的均值来代替这个缺失的数据。对于非定距型数据，依据统计学中的众数原理，则可以使用该属性的众数来补齐缺失值。

2.利用同类均值插补。同均值插补的方法都属于单值插补，不同的是，它用层次聚类模型预测缺失变量的类型，再以该类型的均值插补。假设为信息完全的变量，Y为存在缺失值的变量，那么首先对X或其子集行聚类，然后按缺失个案所属类来插补不同类的均值。如果在以后统计分析中还需以引入的解释变量和Y做分析，那么这种插补方法将在模型中引入自相关，给分析造成障碍。

3.极大似然估计。在缺失值类型为随机缺失的条件下，可以对未知参数进行极大似然估计。依据总体的概率密度或者概率分布，建立极大似然函数，然后求出似然函数的极大值点，即可确定未知参数的值。而对于极大似然的参数估计在实际中通常采用的计算方法是期望值最大化，即EM算法。

5.1.2问题一模型建立

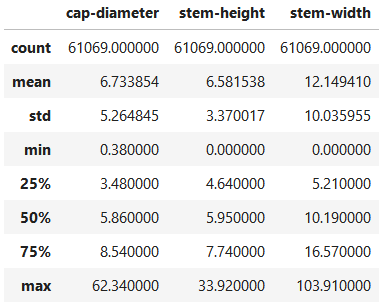
我们以secondary\_data.csv中蘑菇的cap-diameter（帽直径），stem-height（阀杆高度），stem-width（阀杆宽度）为例，来研究其中的缺失值，并尝试对其进行处理。首先利用Python读出secondary\_data.csv并对其进行统计，结果如图所示：

图1蘑菇属性的统计结果

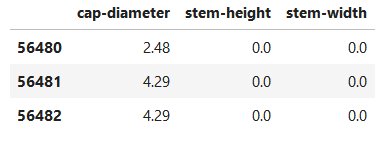
观察结果，不难发现stem-hight和stem-width两列的最小值为0，也就是意味着蘑菇的高度和宽度为0，这显然是不符合生活经验的，所以我们认为这两列存在着数据缺失的情况。随机抽取了56480-56482行发现stem-hight和stem-width两列确实有数据缺失的情况存在。

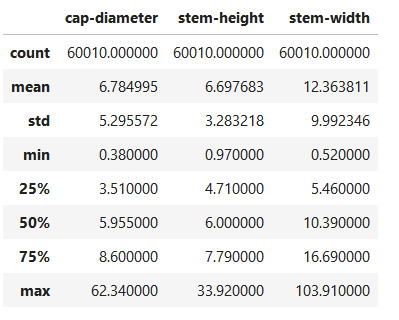
图2 随机抽样结果

下面我们需要对缺失的数据进行处理，我们采用了简单删除，均值插补和同类均值插补三种方法。

5.1.3问题一模型求解

1.简单删除法

根据统计结果，我们发现stem-hight和stem-width两列都有1059个数据缺失，且这1059个数据是同时缺失，由于总数据量有61069个，缺失数据仅占总数据量的1.7%，删除这部分数据对于总体的分布影响不大，因此，我们首先对这1059个数据采取简单删除，删除后剩余的60010个数据分布如下：

图3 简单删除后的统计结果

不难发现，stem-height和stem-width两列的最小值都不为0，再取第56480-56482行看看，发现已被删除。

2.均值插补法

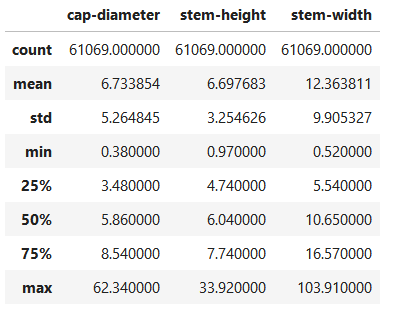
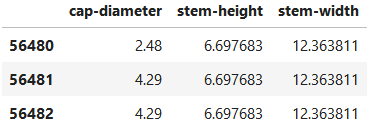
 现在我们采用均值插补法对这1059个数据进行处理，首先对数据进行排序，由于这些数据的分布是无序且非定距的，因此我们采用非定距的均值插补，统计结果如下：

图4 均值插补法后的统计结果

同样的，观察第56480-56482行，可以发现数据已经完成了补充。然而，这三行补充的数据是相似的，因为我们采用了非定距的均值插补法，利用的是这一列数据的众数对于缺失数据进行插补，直观来说，与真实值可能相差较大。

3.同类均值插补法

附件中给出了蘑菇的各种属性，除了stem-height和stem-width以外，还包括了cap-shape，cap-color，habitat等属性，这些定类变量可以作为同类均值插补的分类标准，帮助我们完成同类均值插补。首先利用cap-shape，cap-color，habitat三列对蘑菇进行分类，完成分类后再对每一类进行数据统计，

具体结果在附件中给出。

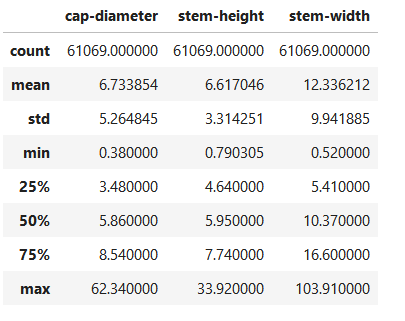
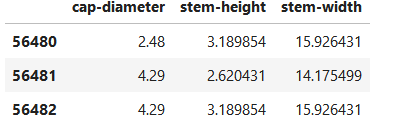
 完成分类后，在每一类中对stem-height和stem-width两列进行均值插补，得到的结果如下图所示：

图5 同类均值插补法的统计结果

完成插补后观察第56480-56482行发现其结果的确与均值插补法有所区别。

5.2.1问题二模型准备

数据异常也是数据分析和数据挖掘中经常会遇到的问题。异常数据在整个数据样本中不会出现的太多，但是这些样本数据会对最终的求解结果产生一定的偏差，所以，有必要找出这些异常数据并对其进行处理。

我们首先讨论单变量数据的异常识别，主要有以下三种方法：

1.简单统计量分析，这个方法非常简单。一般而言，我们所得到的数据都是有着具体的实际意义的，如果其中的某些数据不符合其现实意义，即可认为它们是异常数据。如时间和长度数据如果是负值，那么就可判定为异常数据，从而对数据的采集和分析做出改进。而对于其他类型的数据，可以考虑它们的最大值和最小值，超过最大值和最小值范围的即可认定为异常数据。

2. 3原则。如果在样本服从正态分布或者大样本近似服从正态分布的情况下，异常值可以定义为与平均值的偏差超过3倍标准差的值。正态分布下，偏离均值3倍标准差的值出现的概率小于0.003，这是一个小概率事件，统计学的知识告诉我们，小概率事件在一次独立实验中几乎不可能发生，因此我们有理由认为偏差超过3倍标准差的值是不合理的，是异常数据，应当予以处理。我们在拟合出样本分布后，得到其期望与方差，然后即可利用3原则来筛选出异常的数据。而在实际使用时，我们也可以根据数据的实际意义或者是数据处理的需要来选择不同倍数的标准差作为判定数据异常的依据。

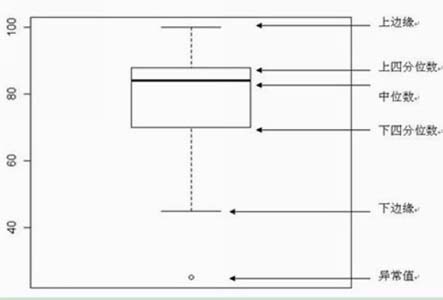
 3.箱线图法。箱线图是一种用作显示一组数据分散情况的统计图，它可以显示出一组数据的最大值、最小值、中位数及上下四分位数。如下图所示：

图6箱线图

其中上四分位数：1/4的样本值取值比他大，下四分位数：1/4的样本值取值比他小，上边缘=上四分位数+1.5\*（上四分位数-下四分位数），下边缘=下四分位数-1.5\*（上四分位数-下四分位数）。异常值则可以被定义为小于下边缘或者大于上边缘的数据。

5.2.2问题二模型建立

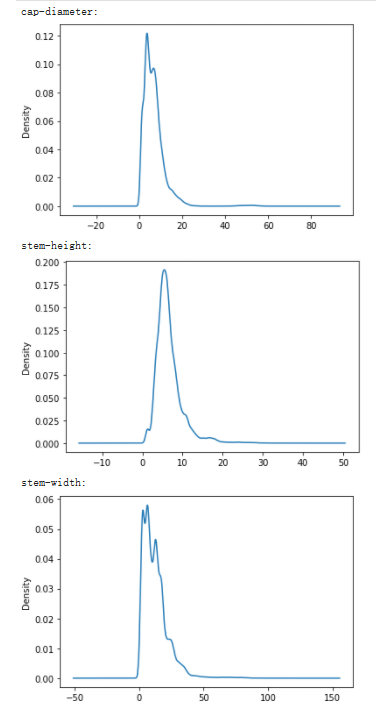
由问题一，我们完成了对于蘑菇数据中的stem-height和stem-width两列中缺失数据的补充工作，我们在同类均值插补的基础上，对于这两种属性以及cap-diameter的所占比例进行检测。由于我们已经完成了对缺失的1059个数据的补充，所以这61069个数据中不再存在缺失数据。首先我们先对这些数据进行统计处理，得到其概率分布：

图7 三种属性的概率分布

观察三种属性的概率分布图像，发现这三种属性都近似地服从正态分布，由大数定律和中心极限定理可知这是显然的，然而这三种属性的分布都存在着偏态分布的特征。下面我们将利用Box-Cox变换和箱线图法检测异常数据。

5.2.3问题二模型求解

1.箱线图法

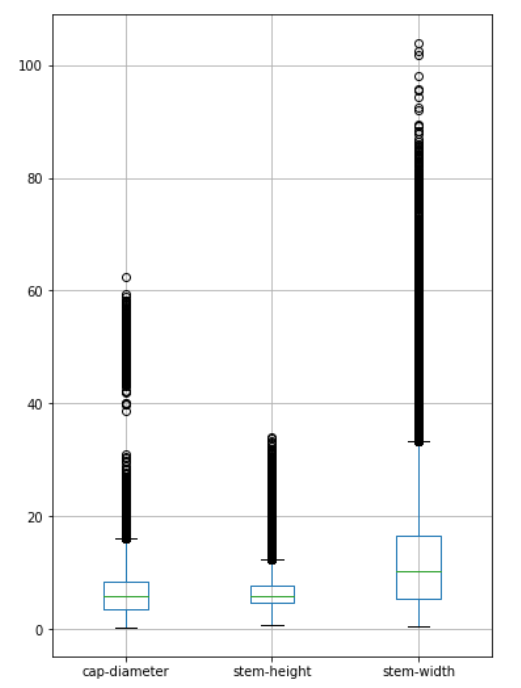
首先直接利用箱线图法对异常数据进行检测，箱线图法检测异常数据的原理及定义在6.2.1中已经阐释，这里就不再赘述。直接利用Python的numpy库和matplotlib库对数据进行处理，结果如下图所示：

图8 未经处理的箱线图

由图像及程序运行的结果可以得出如下结论：

cap-diameter异常数据： 2400 / 61069 占比：3.93%

stem-height异常数据： 3169 / 61069 占比：5.19%

stem-width异常数据： 2028 / 61069 占比：3.32%

可以发现，三种属性中都含有不同比例的异常数据。

2. Box-Cox变换

三种属性的概率分布虽然都近似服从正态分布，但都存在一些有偏的情形，下面我们利用Box-Cox变换对于数据进行预处理，在一定程度上修正分布的偏态，修正的结果如附录所示。

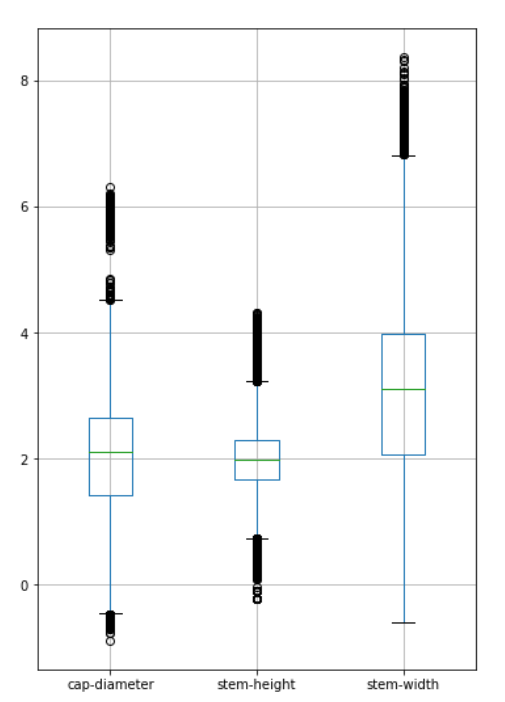
 修正后的概率分布更加接近正态分布，对于修正后的数据采取箱线图法，得到的图像与结果如下图所示：

图9 经过处理的箱线图

由图像及程序运行的结果可以得出如下结论：

cap-diameter异常数据： 573 / 61069 占比：0.94%

stem-height异常数据： 2680 / 61069 占比：4.39%

stem-width异常数据： 368 / 61069 占比：0.60%

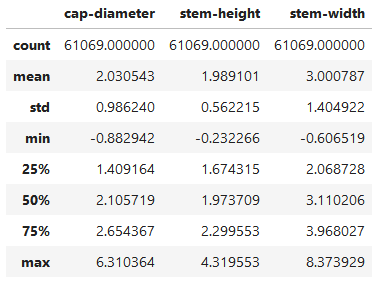
与未经过Box-Cox变换的数据相比，cap-diameter，stem-height和stem-width三列异常数据的占比都有着明显下降。

图10 经过Box-Cox变换的统计结果

而上图就是三列数据经过Box-Cox变换的统计结果，不难发现，虽然Box-Cox变换修正了偏态分布，但随之而来的就是数据的异常，所以下面我们需要对异常的数据进行处理。

5.3.1问题三模型准备

由第二问我们可以对数据集中的异常数据进行识别，但是仅仅将其检测识别出来是不够的，还需要对异常数据进行处理，处理的方法如下：

1.删除。直接将含有异常值的记录删除，通常有两种策略：整条删除和成对删除。这种方法最简单简单易行，但缺点也不容忽视，一是在观测值很少的情况下，这种删除操作会造成样本量不足;二是，直接删除、可能会对变量的原有分布造成影响，从而导致统计模型不稳定。

2.视为缺失值。视为缺失值，利用处理缺失值的方法来处理。这一方法的好处是能够利用现有变量的信息，来填补异常值。需要注意的是，将该异常值作为缺失值处理，需要根据该异常值(缺失值)的特点来进行，针对该异常值(缺失值)是完全随机缺失、随机缺失还是非随机缺失的不同情况进行不同处理。

3.平均值修正。如果数据的样本量很小的话，也可用前后两个观测值的平均值来修正该异常值。这其实是一种比较折中的方法，大部分的参数方法是针对均值来建模的，用平均值来修正，优点是能克服了丢失样本的缺陷，缺点是丢失了样本“特色”。

4.盖帽法。整行替换数据框里99%以上和1%以下的点，将99%以上的点值=99%的点值；小于1%的点值=1%的点值。

5.分箱法。分箱法通过考察数据的“近邻”来光滑有序数据的值。有序值分布到一些桶或箱中。分箱法包括等深分箱：每个分箱中的样本量一致；等宽分箱：每个分箱中的取值范围一致。

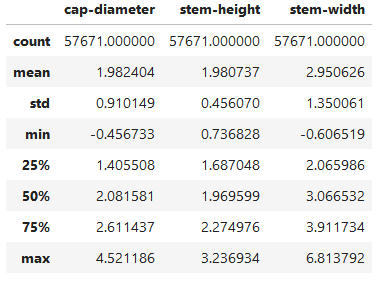
5.3.2问题三模型建立

在对三列数据进行Box-Cox变换后，异常数据所占的比例减少了很多，但与此同时，我们并不能放任这些异常数据不管，尤其是stem-height这一属性的异常值占比已经达到了4.39%，这肯定会对最终的结果产生不良影响，因此，第三问就要处理这些异常值。我们主要采用了直接删除法和盖帽法来对异常数据进行处理。

5.3.3问题三模型求解

1.直接删除法

同缺失数据的处理方法一样，在大样本中，我们同样可以对于有问题的小样本进行删除。删除后的三种属性的概率分布由附录给出，而统计结果如下图所示：

图11 经过直接删除后的统计结果

观察概率分布，去除掉异常数据的总体仍然近似服从正态分布。而对于结果有影响的异常数据已经被剔除，剩下了57671组数据。

2.盖帽法

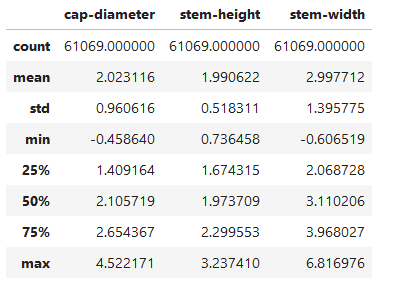
 盖帽法的基本原理为整行替换数据框里99%以上和1%以下的点，将99%以上的点值=99%的点值；小于1%的点值=1%的点值。即将99%以上的值划为异常值，然后用99%点的值对异常值进行替代，将1%以下的值划为异常值，然后用1%点的值对异常值进行替代。进行盖帽法处理后的概率分布同样在附录中给出，而统计结果如下图所示：

图12 经过盖帽法后的统计结果

1. 模型分析

6.1模型一分析

在问题一中，我们采用了简单删除法，均值插补法和同类均值插补法。对于大样本数据中的小部分数据缺失，采取简单删除法是行之有效的。在我们选择的数据中心，如果考虑那1059个缺失数据的个案，肯定会对总体的分布产生较大影响，而将其删除后，剩余的数据就可以更好地拟合出其概率分布。但是蘑菇的数据除了stem-height和stem-weight外还包括其他十余项，如果这样简单粗暴地将其删除，就是对其余属性的浪费。

比较均值插补法和同类均值插补法，均值插补法不太适用于多属性大样本的数据，大样本数据的众数可能仅占总体的很小部分，如我们选择的数据中，stem-width一列的众数为6.22，然而这个众数仅占总体的0.1%不到，如果将缺失的1059个数据全部取为6.22，无疑是十分荒谬的。而同类均值插补法则先依据其他属性对于数据进行分类，分类完成后再采用均值插补法，这样的做法，充分利用了数据的不同属性，并且在一定程度上避免了均值插补法的缺陷。

6.2 模型二分析

在问题二的解决过程中，我们主要采用了Box-Cox变换和箱线图法，首先直接采用箱线图法，直接剔除了大于上边缘和小于下边缘的值。而箱线图法对于异常数据的检测的优势主要体现在：箱线图根据数据的真实分布绘制，他对数据不做任何限制性的要求，比如要服从正态分布等。箱线图异常数据识别依赖于上四分数和下四分位数，因此异常值极其偏差不会影响异常值识别的上下边缘。这一点是优于3倍标准差方法的。

6.3 模型三分析

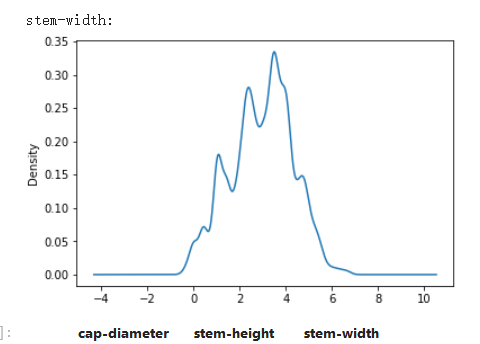
 对于问题三，我们采用了直接删除法和盖帽法对数据进行了处理。不同于问题一，问题一是对缺失值进行了删除，在此问中删除的全部为0值，对于整体的分布起到了良性结果，而在问题三中，删除的并不是0值，所以自然会对整体的概率分布产生偏差，如下图所示：

图13 经过直接删除后的概率分布

Stem-width一列的数据的概率分不变成了多峰分布，不太符合了正态分布的图像。所以，直接删除对概率分布起到了一些不良影响。

1. 参考文献

[1]熊中敏,郭怀宇,吴月欣.缺失数据处理方法研究综述[J].计算机工程与应用,2021,57(14):27-38.

[2]宋莹华,吴忠振.新型自动气象站异常数据的处理方法分析[J].安徽农学通报,2017,23(07):156-158.DOI:10.16377/j.cnki.issn1007-7731.2017.07.066.

[3]李茂洁.对环境监测数据中异常数据的处理方法研究[J].居舍,2019(16):184.

[4]佟昕,高强.统计学中的数据缺失及解决方法[J].辽宁经济管理干部学院(辽宁经济职业技术学院学报),2011(02):15-16.

1. 附录

8.1数据导入及数据统计

**import** numpy **as** np

**from** matplotlib **import** pyplot **as** plt

**import** pandas **as** pd

data**=**pd**.**read\_csv('data.csv')

data**.**info()

data**.**describe()

8.2寻找缺失值

num\_col**=**['cap-diameter','stem-height','stem-width']

data[data['cap-diameter']**==**0]**.**head(5)[num\_col]

data[data['stem-height']**==**0]**.**head(5)[num\_col]

随机抽样

data**.**loc[56480:56482,num\_col]

8.3简单删除法

data2**=**data[

(data['cap-diameter']**!=**0)**&**

(data['stem-height']**!=**0)**&**

(data['stem-width']**!=**0)]

data2p**=**data[

(data['cap-diameter']**==**0)**|**

(data['stem-height']**==**0)**|**

(data['stem-width']**==**0)]

print('%d - %d = %d'**%**(data**.**shape[0],data2p**.**shape[0],data2**.**shape[0]))

data2**.**describe()

8.4 均值插补法

data3**=**data**.**copy()

data3**.**loc[data3['stem-height']**==**0,'stem-height']**=**data3**.**loc[data3['stem-height']**!=**0,'stem-height']**.**mean()

data3**.**loc[data3['stem-width']**==**0,'stem-width']**=**data3**.**loc[data3['stem-width']**!=**0,'stem-width']**.**mean()

data3**.**describe()

8.5同类均值插补

norm\_col**=**['cap-shape','cap-color','habitat']

data4**=**data**.**copy()

gp**=**data4**.**groupby(norm\_col)

ccc**=**0

**for** i **in** gp**.**groups:

print(i)

ccc**+=**1

**if** ccc**>**10:

print('......')

**break**

gp['stem-height']**.**agg('mean')

数据统计

data4**.**loc[data4['stem-height']**==**0,'stem-height']**=**gp['stem-height']**.**transform('mean')

data4**.**loc[data4['stem-width']**==**0,'stem-width']**=**gp['stem-width']**.**transform('mean')

data4**.**describe()

8.6异常数据识别

data**=**data4**.**copy()

**for** i **in** num\_col:

print(i**+**':')

data[i]**.**plot(kind**=**"kde")

plt**.**show()

8.7箱线图法

box\_fig **=** data**.**boxplot(num\_col,return\_type **=** 'dict',figsize**=**(6,9))

plt**.**show()

**for** i **in** range(3):

y **=** box\_fig['fliers'][i]**.**get\_ydata()

print(num\_col[i])

print('异常数据：',len(y),'/',data**.**shape[0],'\t','占比：%.2f%%'**%**(len(y)**\***100**/**data**.**shape[0]))

8.8 Box-Cox变换

**from** scipy.stats **import** boxcox

**for** i **in** num\_col:

print(i)

y**=**data[i]

y**.**plot(kind**=**"kde")

print('变换前：')

plt**.**show()

y\_box,lambda\_**=** boxcox(y)

y\_box**=**pd**.**Series(y\_box)

data**.**loc[:,i]**=**y\_box

y\_box**.**plot(kind**=**"kde")

print('变换后：')

plt**.**show()

print('\n'**+**'-'**\***40**+**'\n')

box\_fig**=**data**.**boxplot(num\_col, return\_type**=**'dict', figsize**=**(6, 9))

plt**.**show()

**for** i **in** range(len(num\_col)):

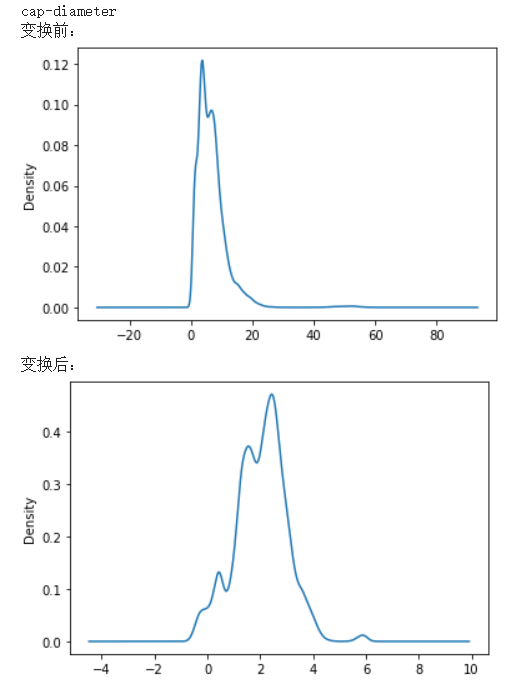
print(num\_col[i])

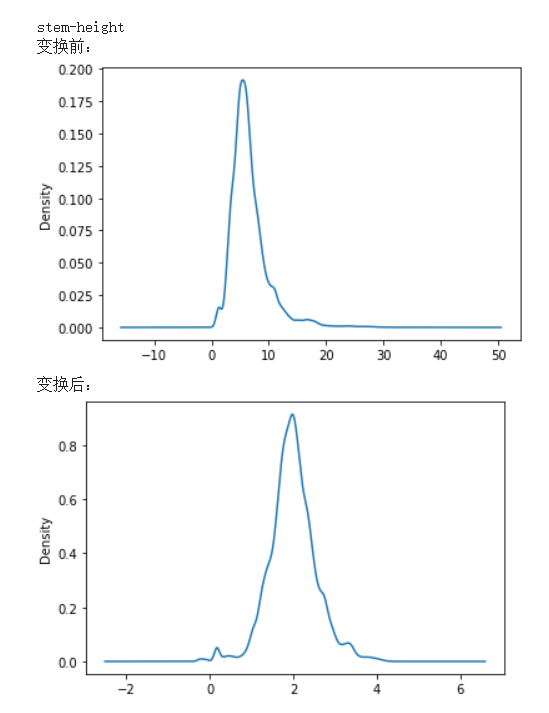
y**=**box\_fig['fliers'][i]**.**get\_ydata()

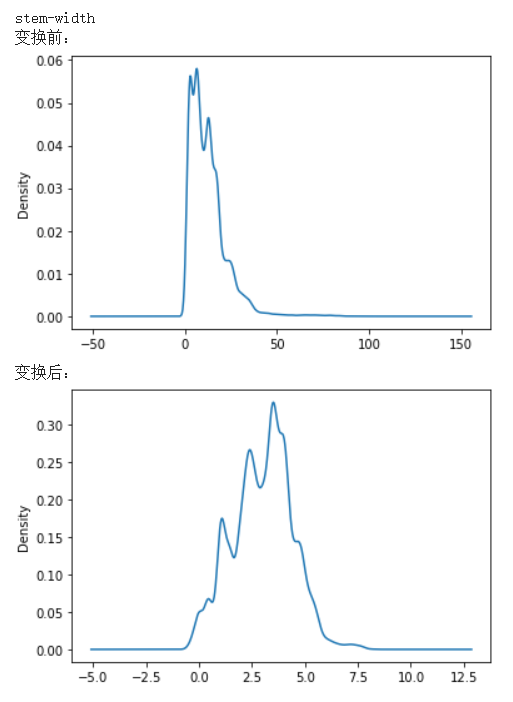
print('异常数据：', len(y), '/', data**.**shape[0], '\t',

'占比：%.2f%%' **%** (len(y) **\*** 100 **/** data**.**shape[0]))

Box-Cox变换结果







8.9 异常值处理

data**.**boxplot(return\_type**=**'dict', figsize**=**(6, 9))

plt**.**show()

L,R**=**{},{}

**for** i **in** num\_col:

l,r**=**data[i]**.**quantile((0.25,0.75))

m**=**r**-**l

l,r**=**l**-**1.5**\***m,r**+**1.5**\***m

L[i]**=**l

R[i]**=**r

print(i,l,r,sep**=**'\t')

删除

a,b,c**=**num\_col

data2**=**data[(data[a]**>**L[a])**&**(data[a]**<**R[a])**&**

(data[b]**>**L[b])**&**(data[b]**<**R[b])**&**

(data[c]**>**L[c])**&**(data[c]**<**R[c])]

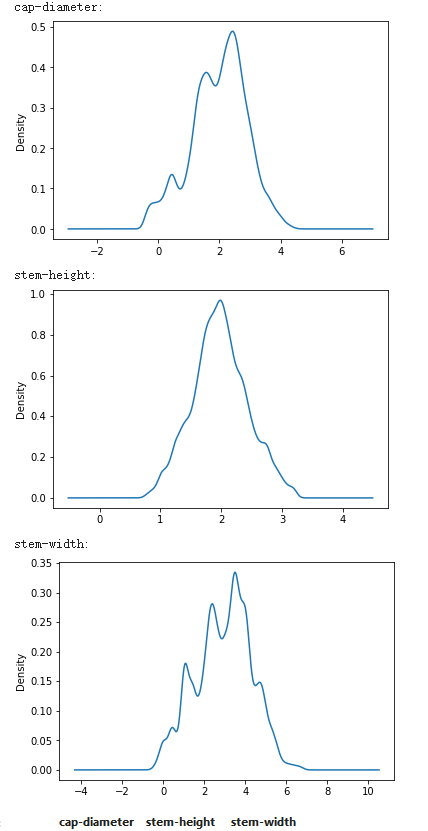
**for** i **in** num\_col:

print(i**+**':')

data2[i]**.**plot(kind**=**"kde")

plt**.**show()

data2**.**describe()

直接删除后的概率密度

盖帽法

data3**=**data**.**copy()

**for** i **in** num\_col:

data3**.**loc[data3[i]**<**L[i],i]**=**L[i]

data3**.**loc[data3[i]**>**R[i],i]**=**R[i]

**for** i **in** num\_col:

print(i**+**':')

data3[i]**.**plot(kind**=**"kde")

plt**.**show()

data3**.**describe()

采用盖帽法后的概率密度

