异常值和缺失值的处理

作者：《数学建模学习交流》清风

在对数据进行预处理中，我们经常会遇到异常值和缺失值的情况，下面我们对这两种情况的常用技术进行介绍，希望能帮到大家。

在数据既有异常值又有缺失值时，先处理哪个并没有严格的顺序。我习惯先处理异常值，再处理缺失值。

异常值的识别方法

异常值，指的是样本中的一些数值明显偏离其余数值的样本点，所以也称为离群点。常见的异常值判断方法可以分为以下两种情况：

（1） 数据有一个给定范围

例如调查问卷中，需要对某个事物进行打分，满分为 0-10 分。如果填问卷的人填了一个 30 分，那么这个数据就是异常值。

这种情况比较简单，我们可以使用 MATLAB 的逻辑运算快速的找到这些异常值：

```
x = [8 9 10 7 6 3 30 4 13 9 2];
ind = find(x<0 | x>10)
```

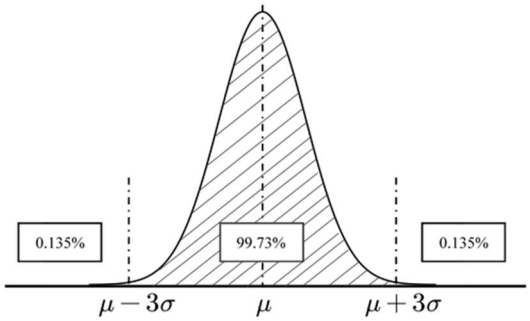
返回 7 和 9，意味着第 7 个位置和第 9 个位置的元素不在 0-10 的范围内。

（2） 数据没有给定的范围

这种情况下我们介绍两种最常用的判定方法：

第一：3σ原则识别异常值

学过概率论的同学应该知道，正态分布的概率密度函数图像是关于均值点处对称的，假设总体服从均值为μ，标准差为σ的正态分布，那么从该总体中随机抽取一个样本点，该点落在区间$[\mu-3\sigma,\mu+3\sigma]$上的概率约为 99.73%，而超出这个范围的可能性仅占不到 0.3%，是典型的小概率事件，所以这些超出该范围的数据可以认为是异常值。这就是3σ原则识别异常值的理论基础。



下面总结3σ原则识别异常值的步骤：（1）计算这组数据的均值μ和标准差σ（注意：我们得到的数据一般是样本数据，因此这里的标准差为样本标准差。如果总体的均值和标准差是已知的，那么就用总体的均值和标准差）。（2）判断这组数据中的每个值是否都位于$[\mu-3\sigma,\mu+3\sigma]$这个区间内，如果不在这个区间内就标记为异常值。

注意事项：使用3σ原则确定异常值时，样本数据要来自正态分布总体或者近似于正态分布总体，这一点需要根据历史经验或统计检验来进行判断。

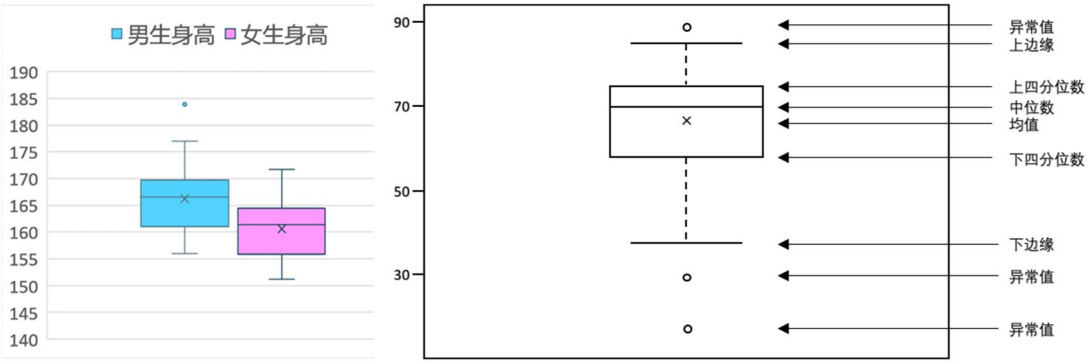
下面给出 MATLAB 的代码(MATLAB 版本 2017a，其他低版本请自行测试):

```
x = [48 51 57 57 49 86 48 53 59 50 48 47 53 56 60];
u = mean(x,'omitnan');  % 假设 x 是取自正态分布的样本
sigma = std(x,'omitnan');
lb = u - 3*sigma;    % 区间下界，low bound 的缩写
ub = u + 3*sigma;   % 区间上界，upper bound 的缩写
tmp = (x < lb) | (x > ub);
ind = find(tmp)
```

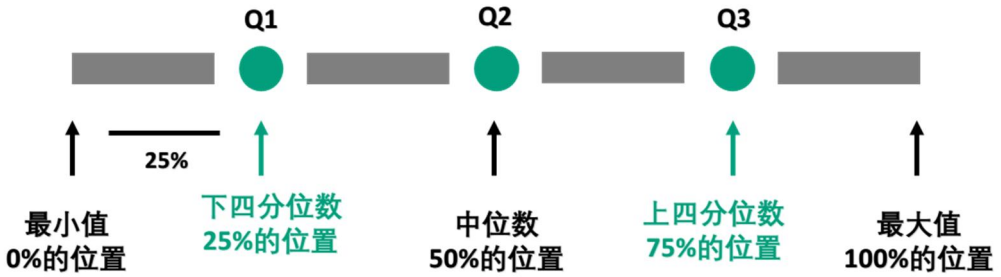
返回 6，意味着第 6 个位置是异常值

第二：箱线图识别异常值

箱线图又称为盒须图、盒式图或箱形图，是一种用作显示数据分散情况资料的统计图，因形状如箱子而得名。下方左侧给出了一个用来反映某班男女同学身高分布情况的箱线图，右侧是箱线图上各元素所代表的含义。可以看到，箱线图可以反映数据的许多统计信息，例如均值、中位数、上四分位数和下四分位数。另外，箱线图中规定了数据的异常值，因此我们可以借助箱线图来识别数据的异常值，下面我们来介绍箱线图中异常值的定义方法。(注意：箱线图的画法不唯一，下面给的是一种典型画法)



首先回顾下中位数的定义：我们将数据按从小到大的顺序排列，在排列后的数据中居于中间位置的数就是中位数，我们用 Q2 表示。
下四分位数则是位于排列后的数据 25%位置上的数值，我们用 Q1 表示；上四分位数则是处在排列后的数据 75%位置上的数值，我们用 Q3 表示。



然后我们要定义一个叫做四分位距（IQR: interquartile range）的指标，它是上四分位数(Q3, 即位于 75%)与下四分位数(Q1, 即位于 25%)的距离，因此 IQR=Q3-Q1。四分位距反映了中间 50%数据的离散程度，其数值越小，说明中间的数据越集中；其数值越大，说明中间的数据越分散。

接下来的工作和3σ原则识别异常值类似，我们需要给出一个合理的区间，位于该区间内的值是正常的数值，而在区间外的值就是我们定义异常值。在箱线图中，该区间一般为$[Q1 - k \times IQR, Q3 + k \times IQR]$，k是控制区间长度的一个正数，通常k取为 1.5。因此，我们只需要判断这组数据中的每个值是否都位于$[Q1 - 1.5 \times IQR,\ Q3 + 1.5 \times IQR]$这个区间内，如果不在这个区间内就标记为异常值。另外，如果我们将k取为 3，在这个区间外的异常值被称为极端异常值。

和3σ原则相比，箱线图并没有对数据服从的分布作任何限制性要求（3σ原则要求数据服从正态分布或近似服从正态分布），其判断异常值的标准主要以四分位数和四分位距为基础。在总体分布未知的情况下，使用箱线图识别异常值的结果更加客观。（通常，箱线图识别出来的异常值要多余3σ原则）

下面给出 MATLAB 的代码：

```
x = [48 51 57 57 49 86 48 53 59 50 48 47 53 56 60];
% 计算分位数的函数需要 MATLAB 安装了统计机器学习工具箱
Q1 = prctile(x,25); % 下四分位数
Q3 = prctile(x,75); % 上四分位数
IQR = Q3-Q1; % 四分位距
lb = Q1 - 1.5*IQR; % 下界
ub = Q3 + 1.5*IQR; % 上界
tmp = (x < lb) | (x > ub);
ind = find(tmp)
```

返回 6，意味着第 6 个位置是异常值

识别出异常值后，我们通常可以将异常值视为缺失值，然后交给缺失值处理方法来处理。

代码：x(ind) = nan　（注意：如果有多列数据都需要处理，可以写一个循环。）

缺失值的处理

如何处理数据的缺失值是一门很深的学问，事实上数据缺失在许多研究领域都是一个复杂的问题。下面我们介绍的只是一些比较简单的处理方法。

首先我们要计算异常值缺失的数量。举一个具体的例子，这是我随机生成的 20 个北京二手房价的数据，每一列是一个指标，每一行是一个样本：

	A	B	C	D	E	F	G	H	I
1	行政区域	卧室数	客厅数	房屋面积	楼层高低	是否是地铁房	是否是学区房	购买时价格	售价（万/m2）
2	丰台	2	2	109.4000	低	0	0		4.92
3	石景山	2	1	55.0000	高	1	0		3.37
4	石景山				高	1		2.85	3.77
5	西城	2	1	52.5000	高	1	1		8.32
6	东城	1	1	50.8000	低	1	1	3.50	8.08
7	西城	2	0	67.2000	中	1	1		8.13
8	朝阳	3	2	137.8000	高	1	0		5.7
9	东城	1	1	68.8000	低	1	1		7.69
10	西城	2	1	72.5000	中	1	1	5.55	7.03
11	朝阳	2	2	102.0000	高	1	0		6.17
12	东城	2	1	54.4000	高	1	0		9.07
13	西城	3	1	115.3000	中	1	0		7.3
14	东城	2	1	54.0000	高	1	0		9.92
15	丰台	3	1	66.3000	高	1	0		4.68
16	石景山	2	1	63.2000	低	1	0	2.61	3.9
17	海淀	2	2	109.4000	低	1	1		6.01
18	石景山	2	1	90.3000	低	1	0	2.95	3.97
19	西城	2	1	59.8000	高	0	1	4.78	7.8
20	海淀	1	1	65.7000	低	1	1		5.27
21	石景山	2	1	68.0000	中	1	0		3.26

可以看到，“购买时价格”这个指标的缺失值有 14 个，占到总样本数的 70%，缺失的有点太多了，所以这一个指标我们可以考虑删除。至于存在多大比例的缺失值我们可以接受，这个并没有一个标准，总之缺失值越少越好，缺的过多就要考虑删除。

另外，我们可以看到，位于 BCDG 四列的指标都有一个缺失值，但是这个缺失值都位于第 4 行的样本中，因此我们可以考虑直接删除这个样本。当然，如果你觉得样本搜集的成本过高或者样本量太少，你也考虑使用后面介绍的缺失值填补的方法。

MATLAB 中计算缺失值数量的函数非常简单，我们可以使用 ismissing 函数和 sum 函数，下面举个例子：

```
A = [3 NaN 5 6 7 NaN NaN 9];
TF = ismissing(A)
```

```
% TF = 1x8 的逻辑数组（为 1 的位置表示是缺失值）
%    0   1   0   0   0   1   1   0
```

```
sum(TF)
```

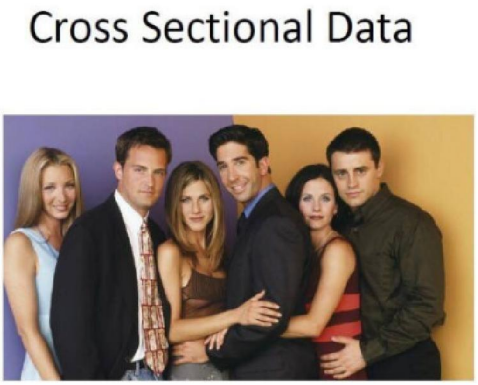
```
对 TF 向量求和，结果为 3，代表有 3 个缺失值
```

另外，ismissing 函数也可以对矩阵或者表格数据类型判断缺失值，有兴趣的同学可以查询 MATLAB 官网。https://www.mathworks.cn/help/matlab/ref/ismissing.html

（技巧：MATLAB 的 table 表格数据类型非常灵活好用，类似于 python 中的 pandas 包，想进阶学习 MATLAB 的同学一定要好好学习这方面的知识。目前市面上这方面的资料较少，大家可以在官网自学各种函数，官网帮助文档非常详细）

下面我们再来介绍缺失值填补，我们需要对缺失的数据类型进行区分：横截面数据和时间序列数据。这两种数据的缺失值处理方法有所不同。

横截面数据是指在某一时点收集的不同对象的数据，例如北京、上海、广州、深圳等 30 个城市今天的最高气温；时间序列数据是指对同一对象在不同时间连续观察所取得的数据，例如北京今年来每天的最高气温。



横截面数据

时间序列数据

对于横截面数据，我们通常使用某个具体的数值来代替缺失值，例如非缺失数据的平均值、中位数或者众数。

MATLAB 的 fillmissing 函数可以很方便的帮助我们实现这个功能。

语法：F = fillmissing(A,'constant',v) 使用常数 v 填充缺失的数组或表。

```
A = [2 3 nan 3 nan nan 8 4];
v = mean(A,'omitnan');  % 平均值
% v = median(A,'omitnan');  % 中位数
```

```
% v = mode(A);  % 众数，常用于离散变量的缺失值
F = fillmissing(A,'constant',v)
```

```
非缺失值数据的平均值是 4，所以将缺失值 nan 代替为 4
     2     3     4     3     4     4     8     4
```

对于数据序列数据，我们通常有下面几种策略：

使用上一个非缺失值	F = fillmissing(A, 'previous')
使用下一个非缺失值	F = fillmissing(A, 'next')
距离最近的非缺失值	F = fillmissing(A, 'nearest')
使用相邻非缺失值的线性插值	F = fillmissing(A, 'linear')
使用分段三次样条插值	F = fillmissing(A, 'spline')

更多介绍请看帮助文档：https://ww2.mathworks.cn/help/matlab/ref/fillmissing.html

```
A = [10 12 16 23 nan 49 68];
F = fillmissing(A, 'previous')
```

```
F =
    10    12    16    23    23    49    68
```

```
F = fillmissing(A, 'next')
```

```
F =
    10    12    16    23    49    49    68
```

```
F = fillmissing(A, 'nearest') % 距离同样近时选择右侧的
```

```
F =
    10    12    16    23    49    49    68
```

```
F = fillmissing(A, 'linear')
```

```
F =
    10    12    16    23    36    49    68
```

```
F = fillmissing(A, 'spline')
```

```
F =
   10.0000   12.0000   16.0000   23.0000   33.9626   49.0000   68.0000
```

当然，我们这里介绍的方法比较简单，如果你专门做数据挖掘，还可以使用一些其他的方法来填补缺失值，例如 KNN 填补、随机森林填补、多重插补等方法。这里我们就不介绍了，有兴趣的同学可以自己搜索相关的论文或者博客学习。