zhuanlan.zhihu.com

Part27:基于pandas的数据清洗-处理空值、重复值和异常值

6-7 minutes

一、处理空值

1.有两种空值:

- None--对象类型
- np.nan(NaN)--浮点类型

- 两种空值的区别： 是否可以参与运算

```
import numpy as np
import pandas as pd
from pandas import DataFrame
```

为什么数据分析中需要用到的是浮点类型而不是对象类型?

- 数据分析中常常会使用某些形式的运算处理原始数据，如果原始数据中空值为NAN，可以参与运算，但是none不能参与运算。所以在pandas中如果遇到了none形式的空值，pandas会将其强行转化为NAN的形式。

2.pandas如何处理空值?

- isnull()+any()有空值就返回true
- notnull()+all()检测表中是否存在空值
- dropna
- fillna

- 处理方式：
  - 1.删除空所对应的行数据
  - 2.对空值进行赋值

```
#不管赋值的时候是none还是np.nan,都会转化为np.nan类型输出
df=DataFrame(data=np.random.randint(0,100,size=(8,7)))
df.iloc[2,3]=None
df.iloc[5,2]=np.nan
df.iloc[6,4]=None
df
```

	0	1	2	3	4	5	6
0	95	35	3.0	30.0	38.0	19	2
1	22	61	19.0	85.0	54.0	6	10
2	65	77	78.0	NaN	86.0	53	58
3	50	34	78.0	50.0	69.0	93	38
4	43	23	87.0	31.0	54.0	26	86
5	97	6	NaN	71.0	19.0	29	53
6	72	85	83.0	84.0	NaN	86	28
7	83	86	28.0	74.0	68.0	35	28

(1).对数据进行删除，3种方式

#方式一：df.loc[df.notnull().all(axis=1)]

```
#使用notnull()判断空值
df.notnull()
```

	0	1	2	3	4	5	6
0	True	True	True	True	True	True	True
1	True	True	True	True	True	True	True
2	True	True	True	False	True	True	True
3	True	True	True	True	True	True	True
4	True	True	True	True	True	True	True
5	True	True	False	True	True	True	True
6	True	True	True	True	False	True	True
7	True	True	True	True	True	True	True


```
#找到空值所对应的行，使用all()
df.notnull().all(axis=1)#axis=0表示列，=1表示行
```

```
0     True
1     True
2    False
3     True
4     True
5    False
6    False
7     True
dtype: bool
```


#将all()函数的结果作为索引取出符合要求的行

```
df.loc[df.notnull().all(axis=1)]
```

	0	1	2	3	4	5	6
0	95	35	3.0	30.0	38.0	19	2
1	22	61	19.0	85.0	54.0	6	10
3	50	34	78.0	50.0	69.0	93	38
4	43	23	87.0	31.0	54.0	26	86
7	83	86	28.0	74.0	68.0	35	28

#方式二：df.loc[~df.isnull().any(axis=1)]

	0	1	2	3	4	5	6
0	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False
2	False	False	False	True	False	False	False
3	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False
5	False	False	True	False	False	False	False
6	False	False	False	False	True	False	False
7	False	False	False	False	False	False	False


```
0    False
1    False
2     True
3    False
4    False
5     True
6     True
7    False
dtype: bool
```


~df.isnull().any(axis=1)#该步骤的目的也是准备用true值作为索引，所以

~df.isnull().any()和df.notnull().all(axis=1)等价

```
df.loc[~df.isnull().any(axis=1)]
```

	0	1	2	3	4	5	6
0	95	35	3.0	30.0	38.0	19	2
1	22	61	19.0	85.0	54.0	6	10
3	50	34	78.0	50.0	69.0	93	38
4	43	23	87.0	31.0	54.0	26	86
7	83	86	28.0	74.0	68.0	35	28

#方法三：dropna(axis=0),直接删除数据表中存在空值的行

	0	1	2	3	4	5	6
0	95	35	3.0	30.0	38.0	19	2
1	22	61	19.0	85.0	54.0	6	10
3	50	34	78.0	50.0	69.0	93	38
4	43	23	87.0	31.0	54.0	26	86
7	83	86	28.0	74.0	68.0	35	28

(2).对数据进行填充fillna()

- 使用空值的近邻值对数据进行填充fillna(method='ffill',axis=0)

```
df
```

	0	1	2	3	4	5	6
0	95	35	3.0	30.0	38.0	19	2
1	22	61	19.0	85.0	54.0	6	10
2	65	77	78.0	NaN	86.0	53	58
3	50	34	78.0	50.0	69.0	93	38
4	43	23	87.0	31.0	54.0	26	86
5	97	6	NaN	71.0	19.0	29	53
6	72	85	83.0	84.0	NaN	86	28
7	83	86	28.0	74.0	68.0	35	28

`df.fillna(method='ffill',axis=0)#axis=0`表示在列，`ffill`表示用前一个数填充，整体表示在列上用前一个数填充。

`#method`还可以使用`'bfill'`

	0	1	2	3	4	5	6
0	95	35	3.0	30.0	38.0	19	2
1	22	61	19.0	85.0	54.0	6	10
2	65	77	78.0	85.0	86.0	53	58
3	50	34	78.0	50.0	69.0	93	38
4	43	23	87.0	31.0	54.0	26	86
5	97	6	87.0	71.0	19.0	29	53
6	72	85	83.0	84.0	19.0	86	28
7	83	86	28.0	74.0	68.0	35	28

思考：什么时候删除空值行，什么时候填充行?

- 如果删除成本不高，那就填充行

二、处理重复值（行数据重复）

```
df=DataFrame(data=np.random.randint(0,100,size=(8,6)))
df
```

	0	1	2	3	4	5
0	4	25	87	61	18	92
1	81	17	86	0	63	98
2	10	36	78	20	76	7
3	31	2	67	72	67	77
4	90	56	53	49	39	22
5	99	32	2	3	60	95
6	44	14	33	23	21	56
7	89	32	4	14	49	72


```
df.iloc[2]=[2,3,4,4,4,3]
df.iloc[5]=[2,3,4,4,4,3]
df.iloc[7]=[2,3,4,4,4,3]
df
```

	0	1	2	3	4	5
0	4	25	87	61	18	92
1	81	17	86	0	63	98
2	2	3	4	4	4	3
3	31	2	67	72	67	77
4	90	56	53	49	39	22
5	2	3	4	4	4	3
6	44	14	33	23	21	56
7	2	3	4	4	4	3


```
df.drop_duplicates(keep='first')#keep值可以是first,也可以是last
```

	0	1	2	3	4	5
0	4	25	87	61	18	92
1	81	17	86	0	63	98
2	2	3	4	4	4	3
3	31	2	67	72	67	77
4	90	56	53	49	39	22
6	44	14	33	23	21	56

三、处理异常值

- 自定义一个1000行3列（A,B,C）取值范围是0-1的数据源，然后将C列中大于两倍标准差的异常值作为异常值

```
df=DataFrame(data=np.random.random(size=(1000,3)),columns=['A','B','C'])
df.head()
```

	A	B	C
0	0.839746	0.361429	0.724278
1	0.496363	0.601767	0.365269
2	0.504477	0.696346	0.356517
3	0.725737	0.995078	0.492420
4	0.073782	0.276550	0.139798


```
#定义异常值的标准：C列中大于两倍标准差的异常值
twice_std=2 * df['C'].std()
twice_std
```

```
0.5672911150134552
```

```
#判断C列的值是否是异常值,然后取反，利用索引去除异常值
df.loc[~(df['C'] > twice_std)]
df.head(5)
```

	A	B	C
0	0.839746	0.361429	0.724278
1	0.496363	0.601767	0.365269
2	0.504477	0.696346	0.356517
3	0.725737	0.995078	0.492420
4	0.073782	0.276550	0.139798