# 处理缺失值

| 函数名 | 描述 |
| --- | --- |
| dropna | 根据每个标签的值是否缺失数据来筛选轴标签，并根据允许丢失的数据量来确定阈值 |
| fillna | 用某些值填充缺失的数据或使用插值方法 |
| isnull | 返回表明哪些值是缺失值的布尔值 |
| notnull | isnull的反函数 |

```
#定义数据data
data = pd.DataFrame([[1.0,6.5,3.0],[1.0,NA],[NA,NA,NA],[NA,6.5,3.0]])
data
Out[35]:
     0    1    2
0  1.0  6.5  3.0
1  1.0  NaN  NaN
2  NaN  NaN  NaN
3  NaN  6.5  3.0

#isnull(): 返回表明哪些值是缺失值的布尔值
data.isnull()
Out[32]:
       0      1      2
0  False  False  False
1  False   True   True
2   True   True   True
3   True  False  False

#notnull(): isnull()的反函数
data.notnull()
Out[33]:
       0      1      2
0   True   True   True
1   True  False  False
2  False  False  False
3  False   True   True
```

- **过滤缺失值 : dropna() ,根据每个标签的值是否缺失数据来筛选轴标签，并根据允许丢失的数据量来确定阈值**

```
函数签名 : dropna(axis=0, how='any', thresh=None, subset=None, inplace=False)
```

```
data
Out[35]:
     0    1    2
0  1.0  6.5  3.0
1  1.0  NaN  NaN
2  NaN  NaN  NaN
3  NaN  6.5  3.0
#默认情况
data.dropna()
Out[37]:
     0    1    2
0  1.0  6.5  3.0

#传入how = 'all'时，将删除所有值均为NA的行
data.dropna(how = 'all')
Out[38]:
     0    1    2
0  1.0  6.5  3.0
1  1.0  NaN  NaN
3  NaN  6.5  3.0

#若要对列操作，则传入axis = 1
data.dropna(axis = 1,how = 'all')
Out[39]:
     0    1    2
0  1.0  6.5  3.0
1  1.0  NaN  NaN
2  NaN  NaN  NaN
3  NaN  6.5  3.0

#thresh参数可以保留一定数量的观察值的行，如下为保留至少2个有效值的行
data.dropna(how = 'all',thresh=2)
Out[40]:
     0    1    2
0  1.0  6.5  3.0
3  NaN  6.5  3.0

#subset参数表示保留指定列不为null的行，如下保留第0列数据中不为null的行
data.dropna(subset = [0])
Out[44]:
     0    1    2
0  1.0  6.5  3.0
1  1.0  NaN  NaN
#保留第0，1列都不为null的行
data.dropna(subset = [0,1])
Out[43]:
     0    1    2
0  1.0  6.5  3.0

#inplace = True 则对数据原地操作，默认False
data.dropna(inplace = True)
data
Out[51]:
```

```
     0    1    2
0  1.0  6.5  3.0
```

- **补充缺失值：fillna()，用某些值填充缺失的数据或使用插值方法**

```
函数签名 : df.fillna( ['value=None', 'method=None', 'axis=None', 'inplace=False',
'limit=None', 'downcast=None', '**kwargs'])

#定义数据
df = pd.DataFrame(np.arange(25).reshape((5,5)))
df
Out[118]:
    0   1   2   3   4
0   0   1   2   3   4
1   5   6   7   8   9
2  10  11  12  13  14
3  15  16  17  18  19
4  20  21  22  23  24

df.iloc[2:,2:4] = NA
df
Out[120]:
    0   1    2    3   4
0   0   1  2.0  3.0   4
1   5   6  7.0  8.0   9
2  10  11  NaN  NaN  14
3  15  16  NaN  NaN  19
4  20  21  NaN  NaN  24

df.loc[4,4] = NA
df
Out[122]:
    0   1    2    3     4
0   0   1  2.0  3.0   4.0
1   5   6  7.0  8.0   9.0
2  10  11  NaN  NaN  14.0
3  15  16  NaN  NaN  19.0
4  20  21  NaN  NaN   NaN

fillna(['value=None', 'method=None', 'axis=None', 'inplace=False', 'limit=None',
'downcast=None', '**kwargs'])

#默认情况
df.fillna(0)
Out[123]:
    0   1    2    3     4
0   0   1  2.0  3.0   4.0
1   5   6  7.0  8.0   9.0
2  10  11  0.0  0.0  14.0
3  15  16  0.0  0.0  19.0
4  20  21  0.0  0.0   0.0
```

```
#method参数控制填充方向,ffill ->向前填充 , bfill ->向后填充
df.fillna(method = 'ffill')
Out[125]:
    0   1    2    3     4
0   0   1  2.0  3.0   4.0
1   5   6  7.0  8.0   9.0
2  10  11  7.0  8.0  14.0
3  15  16  7.0  8.0  19.0
4  20  21  7.0  8.0  19.0

#limit参数限制填充个数
df.fillna(method = 'ffill',limit = 2)
Out[129]:
    0   1    2    3     4
0   0   1  2.0  3.0   4.0
1   5   6  7.0  8.0   9.0
2  10  11  7.0  8.0  14.0
3  15  16  7.0  8.0  19.0
4  20  21  NaN  NaN  19.0

#axis参数控制填充轴
df.fillna(method = 'ffill',axis = 1)
Out[131]:
      0     1     2     3     4
0   0.0   1.0   2.0   3.0   4.0
1   5.0   6.0   7.0   8.0   9.0
2  10.0  11.0  11.0  11.0  14.0
3  15.0  16.0  16.0  16.0  19.0
4  20.0  21.0  21.0  21.0  21.0

#inplace参数表示原地操作
df.fillna(df.mean(),inplace=True)
df
Out[136]:
    0   1    2    3     4
0   0   1  2.0  3.0   4.0
1   5   6  7.0  8.0   9.0
2  10  11  4.5  5.5  14.0
3  15  16  4.5  5.5  19.0
4  20  21  4.5  5.5  11.5
```