

# 10.pandas的替换和部分替换 (replace)



探索数据之美 (/u/efc989099b51)

+ 关注

 0.7

2017.12.12 13:42

字数 733

阅读 27465

评论 1

喜欢 9

/u/efc989099b51)

在处理数据的时候，很多时候会遇到批量替换的情况，如果一个一个去修改效率过低，也容易出错。replace()是很好的方法。

In [1]: import numpy as np  
import pandas as pd  
df=pd.read\_excel(r'C:\Users\Administrator\Desktop\变电站数据.xlsx')  
df

Out[1]:

	国家/地区	省/市/自治区	城市	变电站	Latitude	Longitude	表1变电站	表1Lat	表1Lon
0	中国	重庆	南岸	变电站-A	ab29.52de	NaN	变电站-B	29.54	106.57
1	中国	重庆	南岸	变电站-A	ab29.525d	106.56	变电站-C	29.57	106.55
2	中国	重庆	南岸	变电站-B	ab29.54ab	106.57	变电站-A	NaN	106.56
3	中国	NaN	南岸	变电站-B	ab29.545e	106.57	变电站-C	29.57	106.55
4	中国	重庆	南岸	变电站-C	ab29.57af	106.55	变电站-A	29.52	NaN
5	中国	重庆	南岸	变电站-C	ab29.57ae	106.55	变电站-B	29.54	106.57

源数据

## 1、替换全部或者某一行

replace的基本结构是：df.replace(to\_replace, value) 前面是需要替换的值，后面是替换后的值。

例如我们要将南岸改为城区：

In [3]: df.replace('南岸','城区')

Out[3]:

	国家/地区	省/市/自治区	城市	变电站	Latitude	Longitude	表1变电站	表1Lat	表1Lon
0	中国	重庆	城区	变电站-A	ab29.52de	NaN	变电站-B	29.54	106.57
1	中国	重庆	城区	变电站-A	ab29.525d	106.56	变电站-C	29.57	106.55
2	中国	重庆	城区	变电站-B	ab29.54ab	106.57	变电站-A	NaN	106.56
3	中国	NaN	城区	变电站-B	ab29.545e	106.57	变电站-C	29.57	106.55
4	中国	重庆	城区	变电站-C	ab29.57af	106.55	变电站-A	29.52	NaN
5	中国	重庆	城区	变电站-C	ab29.57ae	106.55	变电站-B	29.54	106.57

将南岸改为城区

这样Python就会搜索整个DataFrame并将文档中所有的南岸替换成了城区（要注意这样的操作并没有改变文档的源数据，要改变源数据需要使用inplace = True）。

(/apps/redi  
utm\_sourc  
banner-cl

```
In [6]: df.replace('南岸', '城区', inplace = True)
df

Out[6]:
```

	国家/地区	省/市/自治区	城市	变电站	Latitude	Longitude	表1变电站	表1Lat	表1Lon
0	中国	重庆	城区	变电站-A	ab29.52de	NaN	变电站-B	29.54	106.57
1	中国	重庆	城区	变电站-A	ab29.525d	106.56	变电站-C	29.57	106.55
2	中国	重庆	城区	变电站-B	ab29.54ab	106.57	变电站-A	NaN	106.56
3	中国	NaN	城区	变电站-B	ab29.545e	106.57	变电站-C	29.57	106.55
4	中国	重庆	城区	变电站-C	ab29.57af	106.55	变电站-A	29.52	NaN
5	中国	重庆	城区	变电站-C	ab29.57ae	106.55	变电站-B	29.54	106.57

使用inplace = True更改源数据

由于南岸只有城市一列具有相同值，使用起来比较方便。

但是如果我们要改变表1Lon里的某个数据，而不改变Longitude的数据要怎么做呢？

```
In [10]: df['表1Lon'].replace(106.55, 'A', inplace = True)
df

Out[10]:
```

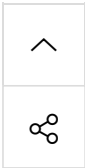
	国家/地区	省/市/自治区	城市	变电站	Latitude	Longitude	表1变电站	表1Lat	表1Lon
0	中国	重庆	城区	变电站-A	ab29.52de	NaN	变电站-B	29.54	106.57
1	中国	重庆	城区	变电站-A	ab29.525d	106.56	变电站-C	29.57	A
2	中国	重庆	城区	变电站-B	ab29.54ab	106.57	变电站-A	NaN	106.56
3	中国	NaN	城区	变电站-B	ab29.545e	106.57	变电站-C	29.57	A
4	中国	重庆	城区	变电站-C	ab29.57af	106.55	变电站-A	29.52	NaN
5	中国	重庆	城区	变电站-C	ab29.57ae	106.55	变电站-B	29.54	106.57

改变指定的列的数据

所以只想替换部分数据的时候并且要写入源数据就需要指定inplace。

在上面的操作只改变了表1Lon的数据，其它列的数据并没有被替换，而且在替换后的结果不需要我们再到源数据进行合并操作，可以直接体现在源数据中。

2、替换指定的某个或指定的多个数值（用字典的形式）



```
In [12]: df.replace({'A':'B',29.54:100})

Out[12]:
```

	国家/地区	省/市/自治区	城市	变电站	Latitude	Longitude	表1变电站	表1Lat	表1Lon
0	中国	重庆	城区	变电站-A	ab29.52de	NaN	变电站-B	100.00	106.57

只改变指定的值

这个很好理解，就是字典里的建作为原值，字典里的值作为替换的新值。

当然，我们也可使用列表的形式进行替换：df.replace(['A','29.54'],['B',100])

```
In [15]: df.replace(['A','29.54'], ['B',100])

Out[15]:
```

	国家/地区	省/市/自治区	城市	变电站	Latitude	Longitude	表1变电站	表1Lat	表1Lon
0	中国	重庆	城区	变电站-A	ab29.52de	NaN	变电站-B	29.54	106.57
1	中国	重庆	城区	变电站-A	ab29.525d	106.56	变电站-C	29.57	B
2	中国	重庆	城区	变电站-B	ab29.54ab	106.57	变电站-A	NaN	106.56
3	中国	NaN	城区	变电站-B	ab29.545e	106.57	变电站-C	29.57	B
4	中国	重庆	城区	变电站-C	ab29.57af	106.55	变电站-A	29.52	NaN
5	中国	重庆	城区	变电站-C	ab29.57ae	106.55	变电站-B	29.54	106.57

用列表的形式进行替换

还有如果想要替换的新值是一样的话，我们还可以这样做：

```
In [16]: df.replace(['A',29.54], 'B')

Out[16]:
```

	国家/地区	省/市/自治区	城市	变电站	Latitude	Longitude	表1变电站	表1Lat	表1Lon
0	中国	重庆	城区	变电站-A	ab29.52de	NaN	变电站-B	B	106.57
1	中国	重庆	城区	变电站-A	ab29.525d	106.56	变电站-C	29.57	B
2	中国	重庆	城区	变电站-B	ab29.54ab	106.57	变电站-A	NaN	106.56
3	中国	NaN	城区	变电站-B	ab29.545e	106.57	变电站-C	29.57	B
4	中国	重庆	城区	变电站-C	ab29.57af	106.55	变电站-A	29.52	NaN
5	中国	重庆	城区	变电站-C	ab29.57ae	106.55	变电站-B	B	106.57

替换的新值一样时

部分替换和替换某个值结合使用的话就可以替换单个列的数值：

(/apps/redi  
utm\_sourc  
banner-cl

替换单个列的数值

3、使用正则表达式替换

正则表达式很强大，能够让我们实现一次替换很多很多个不同的值：

(/apps/redi  
utm\_sourc  
banner-cli

```
In [43]: df['表1变电站'].replace(['变电站-B','变电站-A'],['A','E'],inplace = True)
df
Out[43]:
```

	国家/地区	省/市/自治区	城市	变电站	Latitude	Longitude	表1变电站	表1Lat	表1Lon
0	中国	重庆	城区	变电站-A	ab29.52de	NaN	A	B	106.57
1	中国	重庆	城区	变电站-A	ab29.525d	106.56	变电站-C	29.57	B
2	中国	重庆	城区	变电站-B	ab29.54ab	106.57	E	NaN	106.56
3	中国	NaN	城区	变电站-B	ab29.545e	106.57	变电站-C	29.57	B
4	中国	重庆	城区	变电站-C	ab29.57af	106.55	E	29.52	NaN
5	中国	重庆	城区	变电站-C	ab29.57ae	106.55	A	B	106.57

源数据

```
In [50]: df.replace('[A-Z]', '变电站')
Out[50]:
```

	国家/地区	省/市/自治区	城市	变电站	Latitude	Longitude	表1变电站	表1Lat	表1Lon
0	中国	重庆	城区	变电站-A	ab29.52de	NaN	A	B	106.57
1	中国	重庆	城区	变电站-A	ab29.525d	106.56	变电站-C	29.57	B
2	中国	重庆	城区	变电站-B	ab29.54ab	106.57	E	NaN	106.56
3	中国	NaN	城区	变电站-B	ab29.545e	106.57	变电站-C	29.57	B
4	中国	重庆	城区	变电站-C	ab29.57af	106.55	E	29.52	NaN
5	中国	重庆	城区	变电站-C	ab29.57ae	106.55	A	B	106.57

正则表达式没有指定regex =True

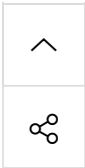
```
In [51]: df.replace('[A-Z]', '变电站',regex = True)
Out[51]:
```

	国家/地区	省/市/自治区	城市	变电站	Latitude	Longitude	表1变电站	表1Lat	表1Lon
0	中国	重庆	城区	变电站-变电站	ab29.52de	NaN	变电站	变电站	106.57
1	中国	重庆	城区	变电站-变电站	ab29.525d	106.56	变电站-变电站	29.57	变电站
2	中国	重庆	城区	变电站-变电站	ab29.54ab	106.57	变电站	NaN	106.56
3	中国	NaN	城区	变电站-变电站	ab29.545e	106.57	变电站-变电站	29.57	变电站
4	中国	重庆	城区	变电站-变电站	ab29.57af	106.55	变电站	29.52	NaN
5	中国	重庆	城区	变电站-变电站	ab29.57ae	106.55	变电站	变电站	106.57

正则表达式指定regex =True

使用正则表达式的时候记得后面加 regex=True参数。

有图中我们可以看到只要包含有大写的英文字母的数据都被替换了，如果我们要写入源数据还需要指定inpla = True。



```
In [55]: df['表1Lat'].replace('[A-Z]', '变电站', regex = True, inplace = True)
df

Out[55]:
```

	国家/地区	省/市/自治区	城市	变电站	Latitude	Longitude	表1变电站	表1Lat	表1Lon
0	中国	重庆	城区	变电站-A	ab29.52de	NaN	A	变电站	106.57
1	中国	重庆	城区	变电站-A	ab29.525d	106.56	变电站-C	29.57	B
2	中国	重庆	城区	变电站-B	ab29.54ab	106.57	E	NaN	106.56
3	中国	NaN	城区	变电站-B	ab29.545e	106.57	变电站-C	29.57	B
4	中国	重庆	城区	变电站-C	ab29.57af	106.55	E	29.52	NaN
5	中国	重庆	城区	变电站-C	ab29.57ae	106.55	A	变电站	106.57

(/apps/redi  
utm\_sourc  
banner-cl

指定列替换数据

当需要将缺失值替换掉的时候，我们可以考虑直接用fillna(), 功能更强大，这个前面已经说过了。

在某些情况下，如果我们只需要某个数据的部分内容，我们该怎么操作呢？

比如要把变电站都改为transformer\_substation ([https://link.jianshu.com?t=http://www.baidu.com/link?url=WzaUb93Y8KMc3IRj6e4fy5LYucGAvpOoTuQuC4I55wfKkeVRZCR6DLWjEC5bietbHa274\\_k5i27SDKhmY60\\_\\_-iTG6uQ\\_rGaz6uPrVUDNc3EspKMOUkXYRGafrXiFSGN](https://link.jianshu.com?t=http://www.baidu.com/link?url=WzaUb93Y8KMc3IRj6e4fy5LYucGAvpOoTuQuC4I55wfKkeVRZCR6DLWjEC5bietbHa274_k5i27SDKhmY60__-iTG6uQ_rGaz6uPrVUDNc3EspKMOUkXYRGafrXiFSGN)), 或者是把Latitude列的前面的ab改为AB:

```
In [82]: df['变电站'] = df['变电站'].str.replace('变电站', 'transformer_substation')
df

Out[82]:
```

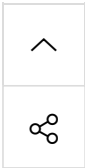
	国家/地区	省/市/自治区	城市	变电站	Latitude	Longitude	表1变电站	表1Lat	表1Lon
0	中国	重庆	城区	transformer_substation-A	ab29.52de	NaN	A	变电站	106.57
1	中国	重庆	城区	transformer_substation-A	ab29.525d	106.56	变电站-C	29.57	B
2	中国	重庆	城区	transformer_substation-B	ab29.54ab	106.57	E	NaN	106.56
3	中国	NaN	城区	transformer_substation-B	ab29.545e	106.57	变电站-C	29.57	B
4	中国	重庆	城区	transformer_substation-C	ab29.57af	106.55	E	29.52	NaN
5	中国	重庆	城区	transformer_substation-C	ab29.57ae	106.55	A	变电站	106.57

指定列更改替换部分字符

```
In [83]: df['Latitude'] = df['Latitude'].str.replace('ab', 'AB')
df

Out[83]:
```

	国家/地区	省/市/自治区	城市	变电站	Latitude	Longitude	表1变电站	表1Lat	表1Lon
0	中国	重庆	城区	transformer_substation-A	AB29.52de	NaN	A	变电站	106.57
1	中国	重庆	城区	transformer_substation-A	AB29.525d	106.56	变电站-C	29.57	B
2	中国	重庆	城区	transformer_substation-B	AB29.54AB	106.57	E	NaN	106.56
3	中国	NaN	城区	transformer_substation-B	AB29.545e	106.57	变电站-C	29.57	B
4	中国	重庆	城区	transformer_substation-C	AB29.57af	106.55	E	29.52	NaN
5	中国	重庆	城区	transformer_substation-C	AB29.57ae	106.55	A	变电站	106.57



指定列更改替换部分字符



赞赏支持

 Python基础知识之pandas库的应用 (/nb/19360819)

举报文章 © 著作权归作者所有



探索数据之美 (/u/efc989099b51)  
写了 24475 字，被 157 人关注，获得了 112 个喜欢  
(/u/efc989099b51)

+ 关注

喜欢 | 9



更多分享



下载简书 App ▶  
随时随地发现和创作内容




(/apps/redirect?utm\_source=note-bottom-click)



登录后发表评论 (/sign-in?utm\_source=desktop&utm\_medium=not-signed-in-comment)

1条评论 只看作者

按时间倒序 按时间正序



张特 (/u/c7e2ab5b2004)  
2楼 · 2019.04.14 22:38  
(/u/c7e2ab5b2004)  
虽然文章不错,但是代码错误好多~~~

赞 回复

^

