

《用 Python 玩转数据》课程资料-

DataFrame 修改操作综合

by Dazhuang@NJU

DataFrame 对象（或 Series 对象）创建好以后常常需要修改包括添加、删除和直接修改行和列数据等，假设数据框 aDF 的值如下所示：

```
>>> aDF
   name  pay
0  Mayue 3000
1   Lilin 4500
2  Wuyun 8000
```

（1）添加列

添加列可直接赋值，例如给 aDF 中添加 tax 列的方法如下：

```
>>> aDF['tax'] = [0.05, 0.05, 0.1]
>>> aDF
```

```
   name  pay  tax
0  Mayue 3000 0.05
1   Lilin 4500 0.05
2  Wuyun 8000 0.1
```

（2）添加行

添加行可用对象的标签(loc)和位置(iloc)索引，也可通过 append()方法或 concat()函数等进行处理，以 loc 为例，例如要给 aDF 添加一个新行，可用如下方法：

```
>>> aDF.loc[5] = {'name': 'Liuxi', 'pay': 5000, 'tax': 0.05}
>>> aDF
```

```
   name  pay  tax
0  Mayue 3000 0.05
1   Lilin 4500 0.05
2  Wuyun 8000 0.1
5   Liuxi 5000 0.05
```

其中 5 为行标签。

（3）删除对象元素

删除数据可直接用“del 数据”的方式进行，但这种方式是直接对原始数据操作，不是很安全，pandas 中可利用 drop()方法删除指定轴上的数据，drop()方法返回一个新的对象，不会直接修改原始数据。例如：

删除行标签为 5 的行（aDF 没有变）：

```
>>> aDF.drop(5)
   name  pay  tax
0  Mayue 3000 0.05
1   Lilin 4500 0.05
```

```
2 Wuyun 8000 0.1
```

删除 tax 列:

```
>>> aDF.drop('tax', axis = 1)
```

```
   name  pay
0 Mayue 3000
1  Lilin 4500
2 Wuyun 8000
5  Liuxi 5000
```

可通过循环或类似 “aDF.drop(['pay', 'tax'], axis = 1)” 这样的方式删除多行或多列数据。此时 aDF 的值没有受影响:

```
>>> aDF
```

```
   name  pay  tax
0 Mayue 3000 0.05
1  Lilin 4500 0.05
2 Wuyun 8000 0.1
5  Liuxi 5000 0.05
```

可以通过设置 aDF.drop(..., inplace=True) 属性直接修改 aDF。

(4) 修改

除了添加和删除 DataFrame (或 Series) 中的元素外, 还可以直接对原始数据进行修改, 例如要将 tax 列统一修改成 0.03, 方法很简单:

```
>>> aDF['tax'] = 0.03
```

```
>>> aDF
```

```
   name  pay  tax
0 Mayue 3000 0.03
1  Lilin 4500 0.03
2 Wuyun 8000 0.03
5  Liuxi 5000 0.03
```

如果要修改某一行的数据也可用如下方法实现:

```
>>> aDF.loc[5] = ['Liuxi', 9800, 0.05]
```

```
   name  pay  tax
0 Mayue 3000 0.03
1  Lilin 4500 0.03
2 Wuyun 8000 0.03
5  Liuxi 9800 0.05
```

另外, 还可以通过例如 "aDF.loc[:, ['name', 'tax', 'pay']] 或 "aDF.iloc[:, [0,2,1]]", 或 reindex() 方法交换一个 DataFrame 对象的列或行。

从以上例子可以看到对 DataFrame 对象 (或 Series 对象) 进行各种修改操作的方法简单有效, 大家可以进行尝试和扩展达到熟练掌握。