82.04 用参数填充缺失数据DataFrame.fillna()

```
Signature:
X_data.fillna(
    value=None,
    method=None,
    axis=None,
    inplace=False,
    limit=None,
    downcast=None,
) -> Union[_ForwardRef('DataFrame'), NoneType]
Docstring:
Fill NA/NaN values using the specified method.
```

```
Parameters
----------
value : scalar, dict, Series, or DataFrame
    Value to use to fill holes (e.g. 0), alternately a
    dict/Series/DataFrame of values specifying which value to use for
    each index (for a Series) or column (for a DataFrame).  Values not
    in the dict/Series/DataFrame will not be filled. This value cannot
    be a list.
```

填充缺失数据

fillna()是最主要的处理方式了。

```
df1=pd.DataFrame([[1,2,3],[NaN,NaN,2],[NaN,NaN,NaN],[8,8,NaN]])
df1
```

代码结果：

	0	1	2
0	1.0	2.0	3.0
1	NaN	NaN	2.0
2	NaN	NaN	NaN

3	8.0	8.0	NaN

用常数填充：

```
df1.fillna(100)
```

代码结果：

	0	**1**	**2**
0	1.0	2.0	3.0
1	100.0	100.0	2.0
2	100.0	100.0	100.0
3	8.0	8.0	100.0

通过**字典**填充不同的常数：

```
df1.fillna({0:10,1:20,2:30})
```

代码结果：

	0	**1**	**2**
0	1.0	2.0	3.0
1	10.0	20.0	2.0

2	10.0	20.0	30.0
3	8.0	8.0	30.0

传入**inplace=True**直接修改原对象：

```
df1.fillna(0,inplace=True)
df1
```

代码结果：

	0	**1**	**2**
0	1.0	2.0	3.0
1	0.0	0.0	2.0
2	0.0	0.0	0.0
3	8.0	8.0	0.0

传入**method=” “**改变插值方式：

```
df2=pd.DataFrame(np.random.randint(0,10,(5,5)))
df2.iloc[1:4,3]=NaN # 1~3行的第3列赋为NaN
df2.iloc[2:4,4]=NaN # 2~3行的第4列赋为NaN
df2
```

代码结果：

	0	**1**	**2**	**3**	**4**
0	6	6	2	4.0	1.0
1	4	7	0	NaN	5.0
2	6	5	5	NaN	NaN
3	1	9	9	NaN	NaN
4	4	8	1	5.0	9.0

```
df2.fillna(method='ffill')#用前面的值来填充
```

代码结果：

	0	**1**	**2**	**3**	**4**
0	6	6	2	4.0	1.0
1	4	7	0	4.0	5.0
2	6	5	5	4.0	5.0
3	1	9	9	4.0	5.0
4	4	8	1	5.0	9.0

传入**limit=” “**限制填充个数：

```
df2.fillna(method='bfill',limit=2)
```

代码结果：

	0	1	2	3	4
0	6	6	2	4.0	1.0
1	4	7	0	NaN	5.0
2	6	5	5	5.0	9.0
3	1	9	9	5.0	9.0
4	4	8	1	5.0	9.0

传入**axis=” “**修改填充方向：

```
df2.fillna(method="ffill",limit=1,axis=1)
```

代码结果：

	0	1	2	3	4
0	6.0	6.0	2.0	4.0	1.0
1	4.0	7.0	0.0	0.0	5.0
2	6.0	5.0	5.0	5.0	NaN
3	1.0	9.0	9.0	9.0	NaN
4	4.0	8.0	1.0	5.0	9.0