Python数据分析实战

第十八课 pandas字符串处理

本节课程目标

- 常见的字符串处理方式

```
#0.读取北京2018年天气的数据
import pandas as pd
file_path = './datas/beijing_tianqi_2018.csv'
df = pd.read_csv(file_path)
```

```
df.head()
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	ymd	bWendu	yWendu	tianqi	fengxiang	fengli	aqi	aqiInfo	aqiLevel
0	2018-01-01	3℃	-6℃	晴~多云	东北风	1-2级	59	良	2
1	2018-01-02	2℃	-5℃	阴~多云	东北风	1-2级	49	优	1
2	2018-01-03	2℃	-5℃	多云	北风	1-2级	28	优	1
3	2018-01-04	0℃	-8℃	阴	东北风	1-2级	28	优	1
4	2018-01-05	3℃	-6℃	多云~晴	西北风	1-2级	50	优	1

```
#查看下数据的类型
df.dtypes
```

```
ymd          object
bWendu       object
yWendu       object
tianqi       object
fengxiang    object
fengli       object
aqi           int64
aqiInfo      object
aqiLevel      int64
dtype: object
```

常见字符串处理函数

```
#1.获取Series的str属性，使用各种字符串处理函数
df['bWendu'].str
```

```
<pandas.core.strings.StringMethods at 0x120a8f0b8>
```

```
#字符串替换函数
df['bWendu'].str.replace('℃','')
```

```
0       3
1       2
2       2
3       0
4       3
5       2
6       2
7       2
8       1
9      -2
10     -1
11      2
12      3
13      6
14      2
15      4
```

```
16      6
17      5
18      7
19      3
20      0
21     -3
22     -4
23     -4
24     -3
25     -3
26     -1
27     -1
28      1
29      4
       ..
335     9
336     8
337     4
338     1
339    -2
340    -4
341    -2
342    -1
343     1
344    -1
345     1
346     3
347     4
348     2
349     7
350     7
351     9
352     9
353     6
354    10
355     8
356     1
357     2
358     2
359    -2
360    -5
361    -3
362    -3
363    -2
364    -2
Name: bWendu, Length: 365, dtype: object
```

```
#判断是否是数字
df['bWendu'].str.isnumeric()
```

```
0      False
1      False
2      False
3      False
4      False
5      False
6      False
7      False
8      False
9      False
10     False
11     False
12     False
13     False
14     False
15     False
16     False
17     False
18     False
19     False
20     False
21     False
22     False
23     False
24     False
25     False
26     False
27     False
28     False
29     False
       ...
335    False
336    False
337    False
338    False
339    False
340    False
341    False
342    False
343    False
344    False
345    False
346    False
347    False
348    False
349    False
350    False
351    False
352    False
353    False
```

```
354    False
355    False
356    False
357    False
358    False
359    False
360    False
361    False
362    False
363    False
364    False
Name: bWendu, Length: 365, dtype: bool
```

```
#或者我们再来看看空气质量
df['aqi'].str.len()
```

```
---------------------------------------------------------------------------

AttributeError                            Traceback (most recent call last)

<ipython-input-37-354c3628e462> in <module>
      1 #或者我们再来看看空气质量
----> 2 df['aqi'].str.len()
```

```
/anaconda3/lib/python3.7/site-packages/pandas/core/generic.py in __getattr__(self, name)
   5061         if (name in self._internal_names_set or name in self._metadata or
   5062                 name in self._accessors):
-> 5063             return object.__getattribute__(self, name)
   5064         else:
   5065             if self._info_axis._can_hold_identifiers_and_holds_name(name):
```

```
/anaconda3/lib/python3.7/site-packages/pandas/core/accessor.py in __get__(self, obj, cls)
    169             # we're accessing the attribute of the class, i.e., Dataset.geo
    170             return self._accessor
--> 171         accessor_obj = self._accessor(obj)
    172         # Replace the property with the accessor object. Inspired by:
    173         # http://www.pydanny.com/cached-property.html
```

```
/anaconda3/lib/python3.7/site-packages/pandas/core/strings.py in __init__(self, data)
   1794
   1795     def __init__(self, data):
-> 1796         self._validate(data)
   1797         self._is_categorical = is_categorical_dtype(data)
   1798
```

```
/anaconda3/lib/python3.7/site-packages/pandas/core/strings.py in _validate(data)
   1816             # (instead of test for object dtype), but that isn't practical for
   1817             # performance reasons until we have a str dtype (GH 9343)
-> 1818             raise AttributeError("Can only use .str accessor with string "
   1819                                  "values, which use np.object_ dtype in "
   1820                                  "pandas")
```

```
AttributeError: Can only use .str accessor with string values, which use np.object_ dtype in
pandas
```

```
condtion = df['ymd'].str.startswith('2018-03')
condtion
```

```
0      False
1      False
2      False
3      False
4      False
5      False
6      False
7      False
8      False
9      False
10     False
11     False
12     False
13     False
14     False
15     False
16     False
17     False
18     False
19     False
20     False
21     False
22     False
23     False
24     False
25     False
26     False
27     False
28     False
29     False
       ...
335    False
336    False
337    False
```

```
338    False
339    False
340    False
341    False
342    False
343    False
344    False
345    False
346    False
347    False
348    False
349    False
350    False
351    False
352    False
353    False
354    False
355    False
356    False
357    False
358    False
359    False
360    False
361    False
362    False
363    False
364    False
Name: ymd, Length: 365, dtype: bool
```

```
condtion
```

```
df[condtion].head()
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	ymd	bWendu	yWendu	tianqi	fengxiang	fengli	aqi	aqiInfo	aqiLevel
59	2018-03-01	8°C	-3°C	多云	西南风	1-2级	46	优	1
60	2018-03-02	9°C	-1°C	晴~多云	北风	1-2级	95	良	2
61	2018-03-03	13°C	3°C	多云~阴	北风	1-2级	214	重度污染	5
62	2018-03-04	7°C	-2°C	阴~多云	东南风	1-2级	144	轻度污染	3
63	2018-03-05	8°C	-3°C	晴	南风	1-2级	94	良	2

```
#3.需要多次str处理的链式草操作
```

```
#获取ymd列的数据，进行替换
df['ymd'].str.replace('-','')
```

```
0     20180101
1     20180102
2     20180103
3     20180104
4     20180105
5     20180106
6     20180107
7     20180108
8     20180109
9     20180110
10    20180111
11    20180112
12    20180113
13    20180114
14    20180115
15    20180116
16    20180117
17    20180118
18    20180119
19    20180120
20    20180121
21    20180122
22    20180123
23    20180124
24    20180125
25    20180126
```

```
26     20180127
27     20180128
28     20180129
29     20180130
         ...
335    20181202
336    20181203
337    20181204
338    20181205
339    20181206
340    20181207
341    20181208
342    20181209
343    20181210
344    20181211
345    20181212
346    20181213
347    20181214
348    20181215
349    20181216
350    20181217
351    20181218
352    20181219
353    20181220
354    20181221
355    20181222
356    20181223
357    20181224
358    20181225
359    20181226
360    20181227
361    20181228
362    20181229
363    20181230
364    20181231
Name: ymd, Length: 365, dtype: object
```

```
df['ymd'].str.replace('-','').slice(0,6)
```

```
---------------------------------------------------------------------------

AttributeError                            Traceback (most recent call last)

<ipython-input-41-2a848a581c31> in <module>
----> 1 df['ymd'].str.replace('-','').slice(0,6)
```

```
/anaconda3/lib/python3.7/site-packages/pandas/core/generic.py in __getattr__(self, name)
   5065             if self._info_axis._can_hold_identifiers_and_holds_name(name):
   5066                 return self[name]
-> 5067             return object.__getattribute__(self, name)
   5068
   5069     def __setattr__(self, name, value):
```

```
AttributeError: 'Series' object has no attribute 'slice'
```

```
df['ymd'].str.replace('-','').str.slice(0,6)
```

```
0      201801
1      201801
2      201801
3      201801
4      201801
5      201801
6      201801
7      201801
8      201801
9      201801
10     201801
11     201801
12     201801
13     201801
14     201801
15     201801
16     201801
17     201801
18     201801
19     201801
20     201801
21     201801
22     201801
23     201801
24     201801
25     201801
26     201801
27     201801
28     201801
29     201801
        ...
335    201812
336    201812
337    201812
338    201812
339    201812
```

```
340    201812
341    201812
342    201812
343    201812
344    201812
345    201812
346    201812
347    201812
348    201812
349    201812
350    201812
351    201812
352    201812
353    201812
354    201812
355    201812
356    201812
357    201812
358    201812
359    201812
360    201812
361    201812
362    201812
363    201812
364    201812
Name: ymd, Length: 365, dtype: object
```

```
#其实上面的slice函数使用的就是str里面的切片方法，可以换种写法
df['ymd'].str.replace('-','').str[0:6]
#结果显示是一样的
```

```
0      201801
1      201801
2      201801
3      201801
4      201801
5      201801
6      201801
7      201801
8      201801
9      201801
10     201801
11     201801
12     201801
13     201801
14     201801
15     201801
16     201801
```

```
17     201801
18     201801
19     201801
20     201801
21     201801
22     201801
23     201801
24     201801
25     201801
26     201801
27     201801
28     201801
29     201801
        ...
335    201812
336    201812
337    201812
338    201812
339    201812
340    201812
341    201812
342    201812
343    201812
344    201812
345    201812
346    201812
347    201812
348    201812
349    201812
350    201812
351    201812
352    201812
353    201812
354    201812
355    201812
356    201812
357    201812
358    201812
359    201812
360    201812
361    201812
362    201812
363    201812
364    201812
Name: ymd, Length: 365, dtype: object
```

```
#4.使用正则表达式的处理
def get_ymd(x):
    year,month,day = x['ymd'].split('-')
    return f'{year}年{month}月{day}日'

df['中文日期'] = df.apply(get_ymd,axis = 1)
```

```
df['中文日期']
```

```
0       2018年01月01日
1       2018年01月02日
2       2018年01月03日
3       2018年01月04日
4       2018年01月05日
5       2018年01月06日
6       2018年01月07日
7       2018年01月08日
8       2018年01月09日
9       2018年01月10日
10      2018年01月11日
11      2018年01月12日
12      2018年01月13日
13      2018年01月14日
14      2018年01月15日
15      2018年01月16日
16      2018年01月17日
17      2018年01月18日
18      2018年01月19日
19      2018年01月20日
20      2018年01月21日
21      2018年01月22日
22      2018年01月23日
23      2018年01月24日
24      2018年01月25日
25      2018年01月26日
26      2018年01月27日
27      2018年01月28日
28      2018年01月29日
29      2018年01月30日
          ...
335     2018年12月02日
336     2018年12月03日
337     2018年12月04日
338     2018年12月05日
339     2018年12月06日
340     2018年12月07日
341     2018年12月08日
342     2018年12月09日
```

```
343    2018年12月10日
344    2018年12月11日
345    2018年12月12日
346    2018年12月13日
347    2018年12月14日
348    2018年12月15日
349    2018年12月16日
350    2018年12月17日
351    2018年12月18日
352    2018年12月19日
353    2018年12月20日
354    2018年12月21日
355    2018年12月22日
356    2018年12月23日
357    2018年12月24日
358    2018年12月25日
359    2018年12月26日
360    2018年12月27日
361    2018年12月28日
362    2018年12月29日
363    2018年12月30日
364    2018年12月31日
Name: 中文日期, Length: 365, dtype: object
```

```
df.head()
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	ymd	bWendu	yWendu	tianqi	fengxiang	fengli	aqi	aqiInfo	aqiLevel	中文日期
0	2018-01-01	3°C	-6°C	晴~多云	东北风	1-2级	59	良	2	2018年01月01日
1	2018-01-02	2°C	-5°C	阴~多云	东北风	1-2级	49	优	1	2018年01月02日
2	2018-01-03	2°C	-5°C	多云	北风	1-2级	28	优	1	2018年01月03日
3	2018-01-04	0°C	-8°C	阴	东北风	1-2级	28	优	1	2018年01月04日
4	2018-01-05	3°C	-6°C	多云~晴	西北风	1-2级	50	优	1	2018年01月05日

```
#方式1：链式replace
df['中文日期'].str.replace('年','').str.replace('月','').str.replace('日','')
```

```
0       20180101
1       20180102
2       20180103
3       20180104
4       20180105
5       20180106
6       20180107
7       20180108
8       20180109
9       20180110
10      20180111
11      20180112
12      20180113
13      20180114
14      20180115
15      20180116
16      20180117
```

```
17     20180118
18     20180119
19     20180120
20     20180121
21     20180122
22     20180123
23     20180124
24     20180125
25     20180126
26     20180127
27     20180128
28     20180129
29     20180130
         ...
335    20181202
336    20181203
337    20181204
338    20181205
339    20181206
340    20181207
341    20181208
342    20181209
343    20181210
344    20181211
345    20181212
346    20181213
347    20181214
348    20181215
349    20181216
350    20181217
351    20181218
352    20181219
353    20181220
354    20181221
355    20181222
356    20181223
357    20181224
358    20181225
359    20181226
360    20181227
361    20181228
362    20181229
363    20181230
364    20181231
Name: 中文日期, Length: 365, dtype: object
```

```
#方式2：正则表达式替换
df['中文日期'].str.replace('[年月日]','')
```

```
0      20180101
1      20180102
2      20180103
3      20180104
4      20180105
5      20180106
6      20180107
7      20180108
8      20180109
9      20180110
10     20180111
11     20180112
12     20180113
13     20180114
14     20180115
15     20180116
16     20180117
17     20180118
18     20180119
19     20180120
20     20180121
21     20180122
22     20180123
23     20180124
24     20180125
25     20180126
26     20180127
27     20180128
28     20180129
29     20180130
         ...
335    20181202
336    20181203
337    20181204
338    20181205
339    20181206
340    20181207
341    20181208
342    20181209
343    20181210
344    20181211
345    20181212
346    20181213
347    20181214
348    20181215
349    20181216
350    20181217
351    20181218
352    20181219
353    20181220
354    20181221
355    20181222
```

```
356    20181223
357    20181224
358    20181225
359    20181226
360    20181227
361    20181228
362    20181229
363    20181230
364    20181231
Name: 中文日期, Length: 365, dtype: object
```

```
```