Python数据分析实战

第十九课 pandas丢失数据处理

本节课程目标

- 认识缺失数据
- 过滤缺失数据
- 处理缺失数据

认识缺失数据

有两种丢失数据：

- None
- np.nan(NaN)

```
#None
10+None
```

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

TypeError                                 Traceback (most recent call last)

<ipython-input-1-e339f8b1cb56> in <module>
      1 #None
----> 2 10+None
```

```
TypeError: unsupported operand type(s) for +: 'int' and 'NoneType'
```

```
type(None)
```

```
NoneType
```

```
#NAN
```

```
type(np.nan)
```

```
float
```

```
10+np.nan
```

```
nan
```

```
import numpy as np
import pandas as pd
from pandas import Series,DataFrame
```

```
df = DataFrame({'Python':[np.nan,128,117,None],'Math':[119,88,116,np.NaN]},
               index=list('abcd'),columns=['Python','Math','En'])
```

```
df
```

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

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

	Python	Math	En
a	NaN	119.0	NaN
b	128.0	88.0	NaN
c	117.0	116.0	NaN
d	NaN	NaN	NaN

```
df.mean(axis=1)
```

```
a    119.0
b    108.0
c    116.5
d      NaN
dtype: float64
```

```
df.std()
```

```
Python     7.778175
Math      17.097758
En              NaN
dtype: float64
```

```
```

- `isnull()`
- `notnull()`
- `dropna()`：过滤丢失数据
- `fillna()`：填充丢失数据

```
df.isnull()
```

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

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

	Python	Math	En
a	True	False	True
b	False	False	True
c	False	False	True
d	True	True	True

```
df.notnull()
```

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

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

	Python	Math	En
a	False	True	False
b	True	True	False
c	True	True	False
d	False	False	False

```
#any/all
```

```
df.isnull().any(axis=1)
```

```
a    True
b    True
c    True
d    True
dtype: bool
```

```
df
```

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

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

	Python	Math	En
a	NaN	119.0	NaN
b	128.0	88.0	NaN
c	117.0	116.0	NaN
d	NaN	NaN	NaN

```
cond = df.isnull().all(axis=1)
cond
```

```
a    False
b    False
c    False
d     True
dtype: bool
```

```
df[cond]
```

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

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

	Python	Math	En
d	NaN	NaN	NaN

```
#logical_not
```

```
np.logical_not(cond)
```

```
a     True
b     True
c     True
d    False
dtype: bool
```

```
# dropna--删除
```

```
import random
df = DataFrame(np.random.randint(0,10,size=(5,3)),columns=
['python','math','en'],index=list('abcde'))
```

```
df
```

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

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

	python	math	en
a	6	6	0
b	0	5	5
c	7	6	8
d	6	5	7
e	0	2	5

```
df == 0
```

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

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

	python	math	en
a	False	False	True
b	True	False	False
c	False	False	False
d	False	False	False
e	True	False	False

```
df['en']['a'] = None
df['python']['b'] = None
df.loc['e','python'] = np.nan
```

```
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/indexing.html#indexing-view-versus-copy
```

```
df
```

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

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

	python	math	en
a	6.0	6	NaN
b	NaN	5	5.0
c	7.0	6	8.0
d	6.0	5	7.0
e	NaN	2	5.0

```
#drop
df.drop(['c','d'])
```

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

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

	python	math	en
a	6.0	6	NaN
b	NaN	5	5.0
e	NaN	2	5.0

```
df.drop('en',axis=1)
```

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

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

	python	math
a	6.0	6
b	NaN	5
c	7.0	6
d	6.0	5
e	NaN	2

```
df.dropna()
```

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

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

	python	math	en
c	7.0	6	8.0
d	6.0	5	7.0

```
#fillna
```

```
df
```

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

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

	python	math	en
a	6.0	6	NaN
b	NaN	5	5.0
c	7.0	6	8.0
d	6.0	5	7.0
e	NaN	2	5.0

```
df.fillna(df.mean())
```

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

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

	python	math	en
a	6.000000	6	6.25
b	6.333333	5	5.00
c	7.000000	6	8.00
d	6.000000	5	7.00
e	6.333333	2	5.00

```
a = df.fillna(10)
```

```
type(a)
```

```
pandas.core.frame.DataFrame
```

```
#axis
```

```

```

```

```

```

```

```

```