

欠損値が含まれているかを判定

<元DataFrame>

	name	age	state	point	other
0	Alice	24	NY	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN
2	Charlie	NaN	CA	NaN	NaN
3	Dave	68	TX	70	NaN
4	Ellen	NaN	CA	88	NaN
5	Frank	30	NaN	NaN	NaN

isnull = isnanでもよい

要素ごとに欠損値かを判定

df.isnull()

	name	age	state	point	other
0	FALSE	FALSE	FALSE	TRUE	TRUE
1	TRUE	TRUE	TRUE	TRUE	TRUE
2	FALSE	TRUE	FALSE	TRUE	TRUE
3	FALSE	FALSE	FALSE	FALSE	TRUE
4	FALSE	TRUE	FALSE	FALSE	TRUE
5	FALSE	FALSE	TRUE	TRUE	TRUE

集約方向(axis)が= 0 or 1か？

集約方向が行なら、axis=0

集約方向が列なら、axis=1

列ごとにすべての要素が欠損値かを判定

df.isnull().all()

```
name      False
age        False
state      False
point      False
other      True
dtype: bool
```

行ごとにすべての要素が欠損値かを判定

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

```
0      False
1      True
2      False
3      False
4      False
5      False
dtype: bool
```

列ごとに1つでも欠損値を含むかを判定

df.isnull().any()

```
name      True
age        True
state      True
point      True
other      True
dtype: bool
```

行ごとに1つでも欠損値を含むかを判定

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

```
0      True
1      True
2      True
3      True
4      True
5      True
dtype: bool
```

列ごとに欠損値の個数をカウント

df.isnull().sum()

```
name      1
age        3
state      2
point      4
other      6      16
dtype: int64
```

行ごとに欠損値の個数をカウント

df.isnull().sum(axis=1)

```
0      2
1      5
2      3
3      1
4      2
5      3
dtype: int64
```

欠損値の総数をカウント

```
df.isnull().values
```

```
df.isnull().values.sum()
```

DataFrameのsum()と異なり、nd_aryのsum()は全体の合計を出せる

```
array([[False, False, False,  True,  True],
       [ True,  True,  True,  True,  True],
       [False,  True, False,  True,  True],
       [False, False, False, False,  True],
       [False,  True, False, False,  True],
       [False, False,  True,  True,  True]])
```

16

列ごとに欠損値でない要素をカウント

```
df.count()
```

```
name      5
age       3
state     4
point     2
other     0      14
dtype: int64
```

欠損値でない要素の総数をカウント

```
df.count().sum()
```

14