Replacing the NULL values

Missing Values

In Pandas missing data is represented by two value,

- None
  - None is a Python singleton object that is often used for missing data in Python code.
- NaN
  - NaN (an acronym for Not a Number)
  - It is a special floating-point value recognized by all systems that use the standard IEEE floating-point representation

Methods to clean null values

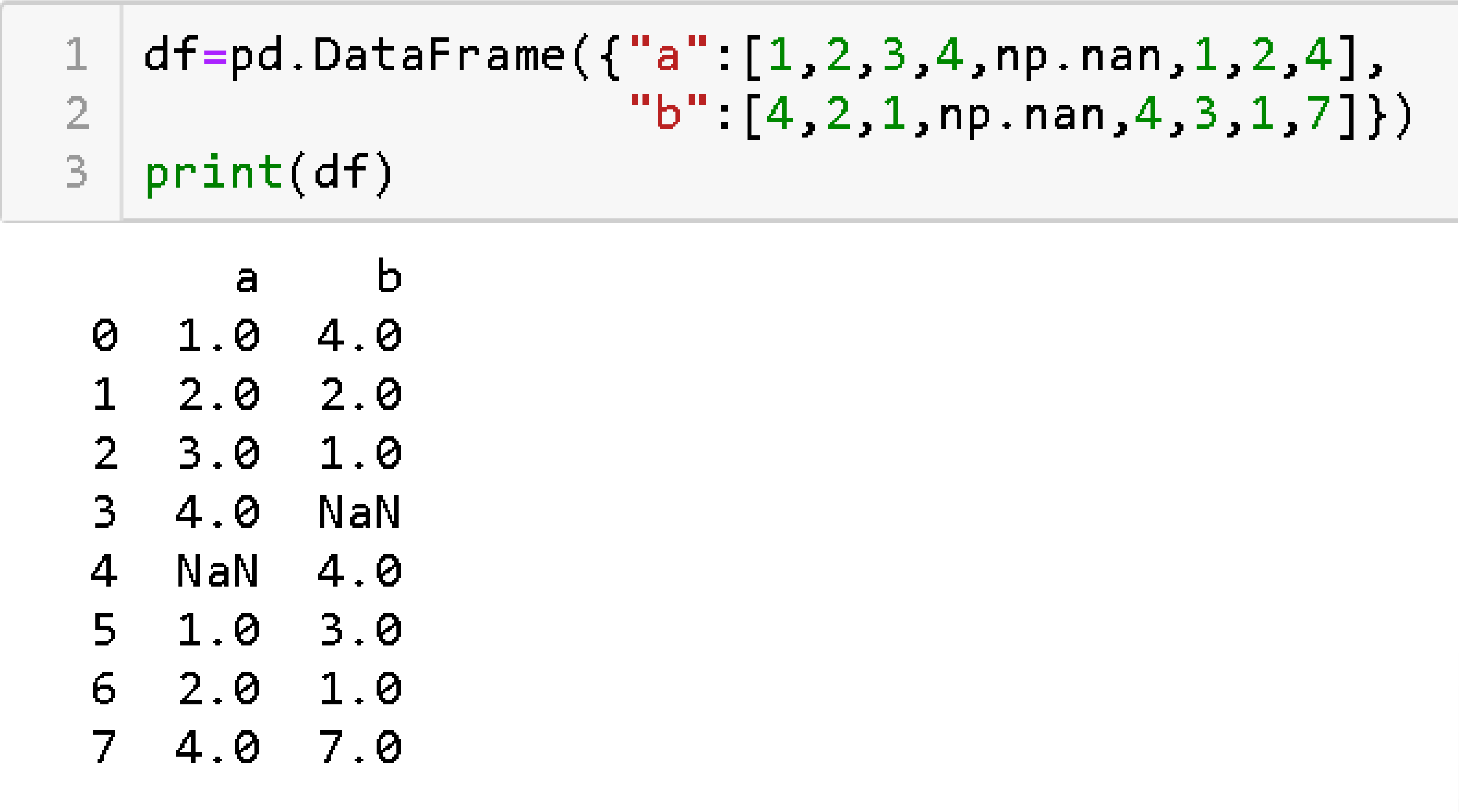
- Delete the rows (sometimes columns) having the null values
- Replace the null values with central tendency measures like mean,median or mode
- Fill with the the previous value or the future value
- Used Regression to interpolate the value

Functions in Pandas

Some of the functions for detecting, removing, and replacing null values in Pandas DataFrame,

- *isnull()*
- *notnull()*
- *dropna()*
- *fillna()*
- *replace()*
- *interpolate()*

Consider a DataFrame as shown below

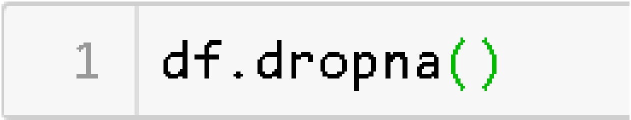


```python
df=pd.DataFrame({"a":[1,2,3,4,np.nan,1,2,4],
                 "b":[4,2,1,np.nan,4,3,1,7]})
print(df)
```

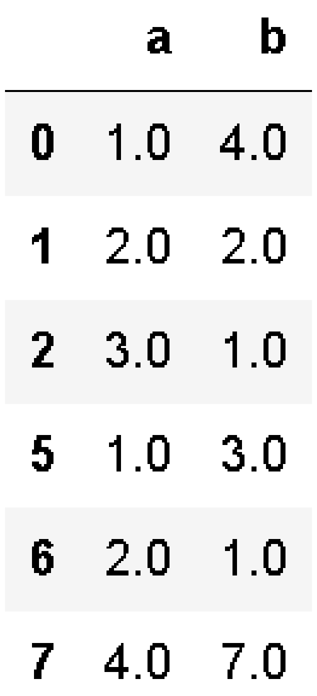
```
     a    b
0  1.0  4.0
1  2.0  2.0
2  3.0  1.0
3  4.0  NaN
4  NaN  4.0
5  1.0  3.0
6  2.0  1.0
7  4.0  7.0
```

Removing Null rows

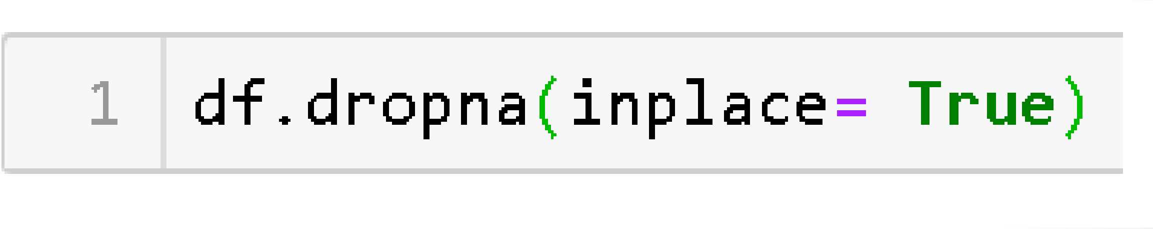
To make the changes permenant use the parameter inplace=True



```
df.dropna()
```

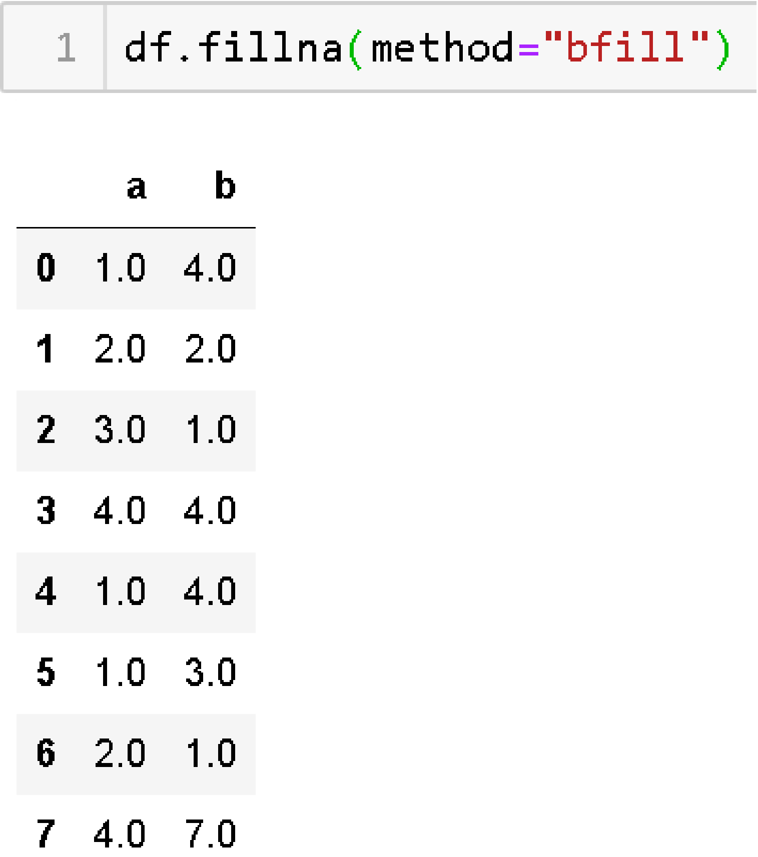


	a	b
0	1.0	4.0
1	2.0	2.0
2	3.0	1.0
5	1.0	3.0
6	2.0	1.0
7	4.0	7.0



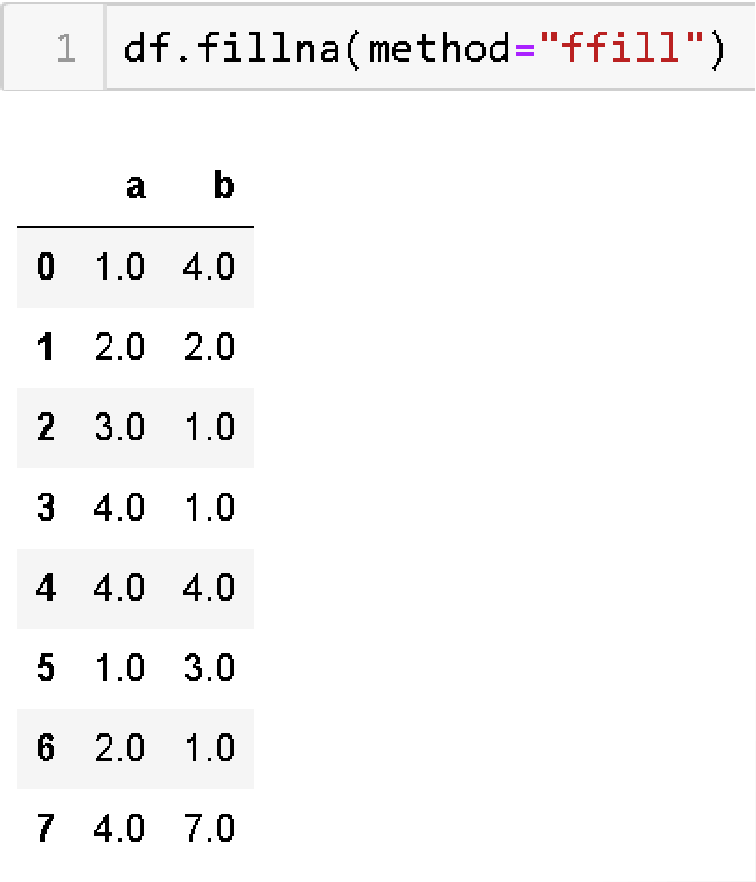
```
df.dropna(inplace= True)
```

Filling null values with Next value or with Previous value



```
df.fillna(method="bfill")
```

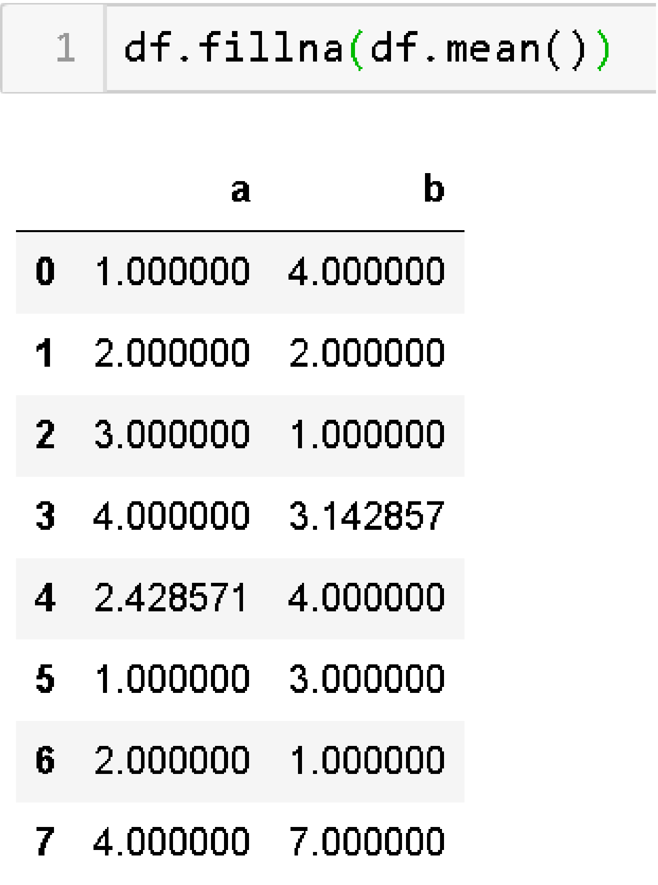
	a	b
0	1.0	4.0
1	2.0	2.0
2	3.0	1.0
3	4.0	4.0
4	1.0	4.0
5	1.0	3.0
6	2.0	1.0
7	4.0	7.0



```
df.fillna(method="ffill")
```

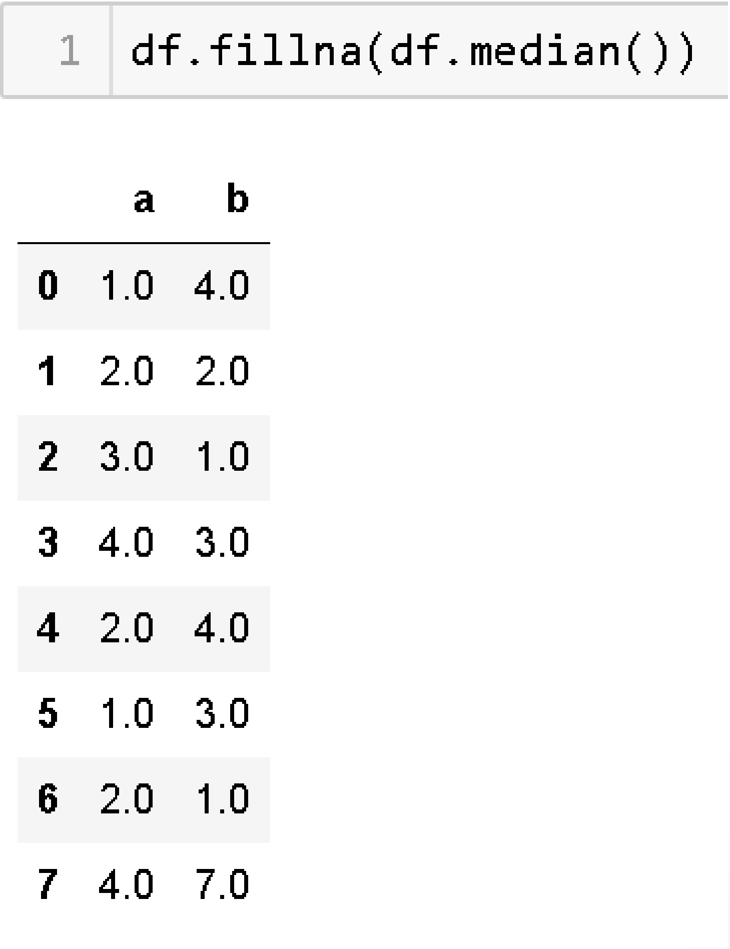
	a	b
0	1.0	4.0
1	2.0	2.0
2	3.0	1.0
3	4.0	1.0
4	4.0	4.0
5	1.0	3.0
6	2.0	1.0
7	4.0	7.0

Fill null value with the central tendency



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

	a	b
0	1.000000	4.000000
1	2.000000	2.000000
2	3.000000	1.000000
3	4.000000	3.142857
4	2.428571	4.000000
5	1.000000	3.000000
6	2.000000	1.000000
7	4.000000	7.000000

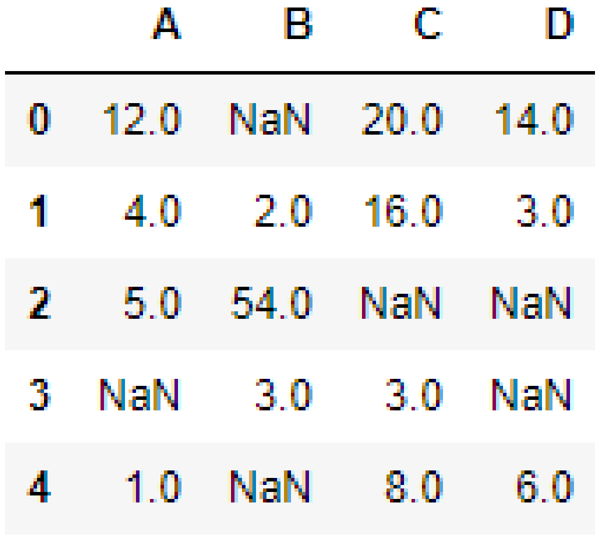


```python
df.fillna(df.median())
```

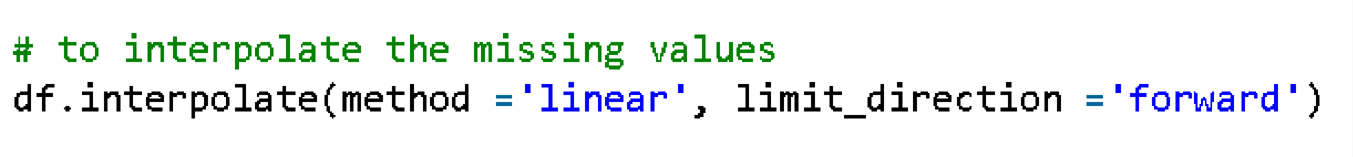
	a	b
0	1.0	4.0
1	2.0	2.0
2	3.0	1.0
3	4.0	3.0
4	2.0	4.0
5	1.0	3.0
6	2.0	1.0
7	4.0	7.0

Interpolate Method

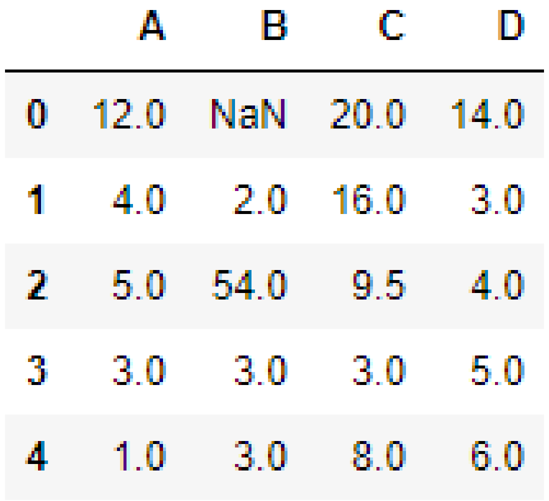
- Given Data,



	A	B	C	D
0	12.0	NaN	20.0	14.0
1	4.0	2.0	16.0	3.0
2	5.0	54.0	NaN	NaN
3	NaN	3.0	3.0	NaN
4	1.0	NaN	8.0	6.0



```
# to interpolate the missing values
df.interpolate(method ='linear', limit_direction ='forward')
```



	A	B	C	D
0	12.0	NaN	20.0	14.0
1	4.0	2.0	16.0	3.0
2	5.0	54.0	9.5	4.0
3	3.0	3.0	3.0	5.0
4	1.0	3.0	8.0	6.0

- *First row could not get filled as the direction of filling of values is forward and there is no previous value which could have been used in interpolation.*