Working with missing data

module 8

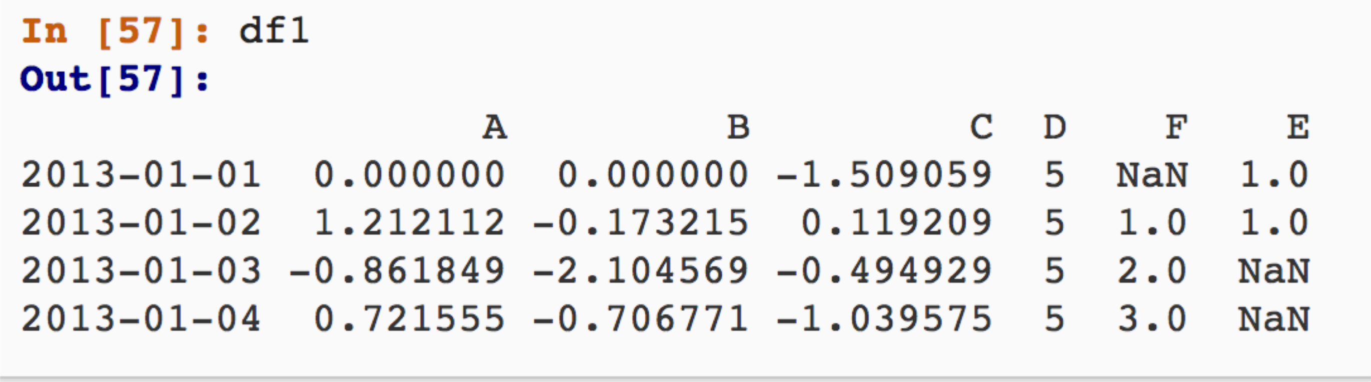
Missing Data

- Missing = not available = NA = NaN = np.nan (in Numpy)

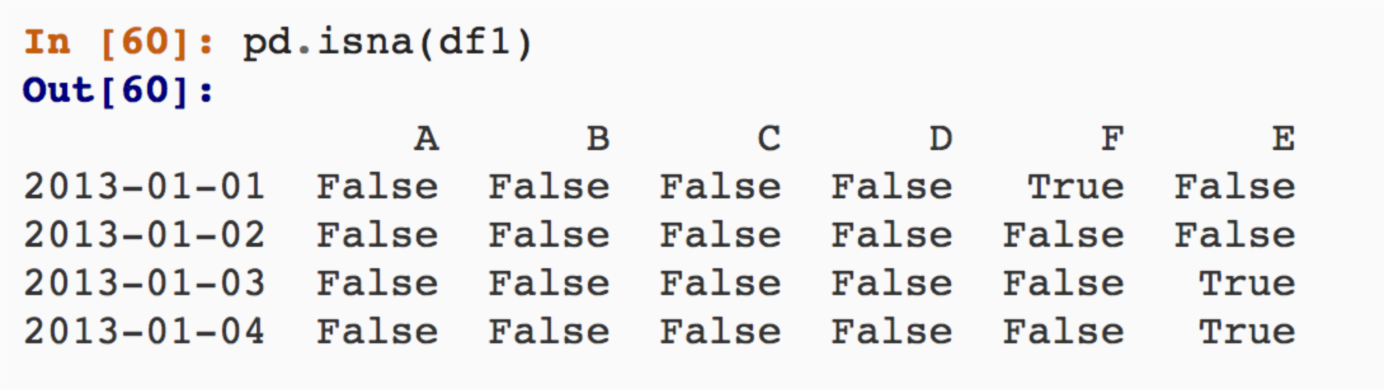
- How to deal with it ?
  - Generate the boolean mask where values are nan (or not nan)
    - pandas provides the isna() and notna() functions, which are also methods on Series and DataFrame objects:
  - Detect nan
    - df.isnull() /df.isnull().any() / df.isnull().sum() / df.isnull().sum().sum()
  - Drop any rows has missing data:
    - df.dropna()
  - Filling missing data:
    - df.fillna()

How to Deal with Missing Data

- To get the boolean mask where values are nan.
  - df.isna() /df.isnull() /df.isnull().any() / df.isnull().sum() / df.isnull.sum().sum()
  - df.notna()



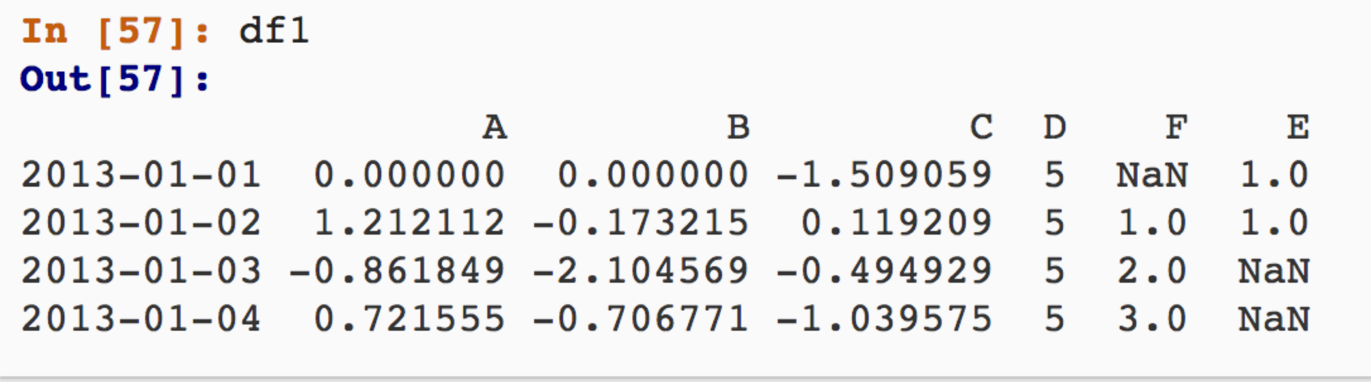
```
In [57]: df1
Out[57]:
                   A         B         C  D    F    E
2013-01-01  0.000000  0.000000 -1.509059  5  NaN  1.0
2013-01-02  1.212112 -0.173215  0.119209  5  1.0  1.0
2013-01-03 -0.861849 -2.104569 -0.494929  5  2.0  NaN
2013-01-04  0.721555 -0.706771 -1.039575  5  3.0  NaN
```



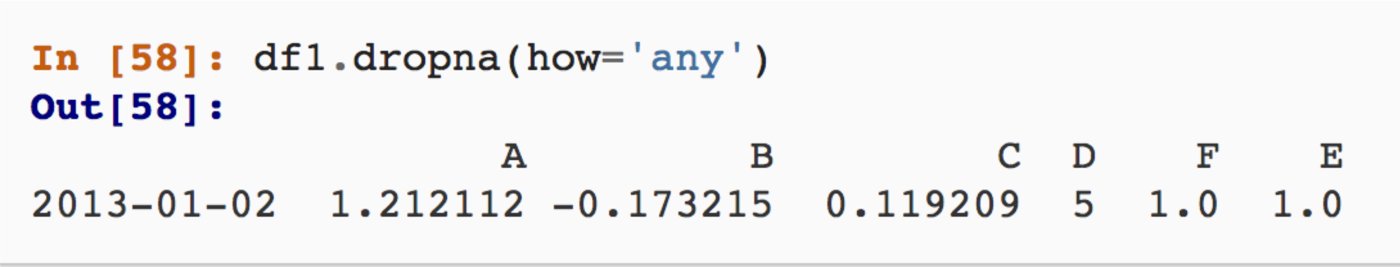
```
In [60]: pd.isna(df1)
Out[60]:
                A      B      C      D      F      E
2013-01-01  False  False  False  False   True  False
2013-01-02  False  False  False  False  False  False
2013-01-03  False  False  False  False  False   True
2013-01-04  False  False  False  False  False   True
```

How to Deal with Missing Data

- Drop any rows has missing data:
  - df.dropna(how='any')



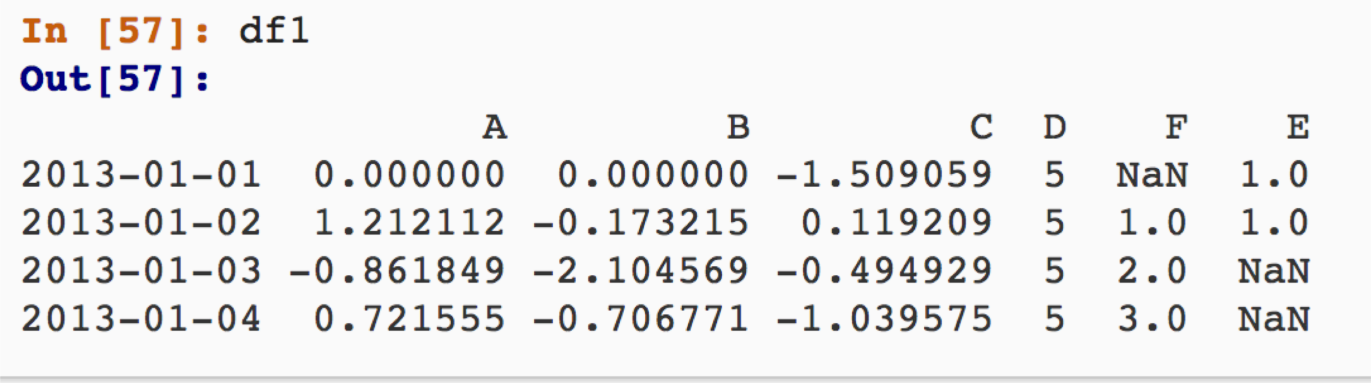
```
In [57]: df1
Out[57]:
                   A         B         C  D    F    E
2013-01-01  0.000000  0.000000 -1.509059  5  NaN  1.0
2013-01-02  1.212112 -0.173215  0.119209  5  1.0  1.0
2013-01-03 -0.861849 -2.104569 -0.494929  5  2.0  NaN
2013-01-04  0.721555 -0.706771 -1.039575  5  3.0  NaN
```



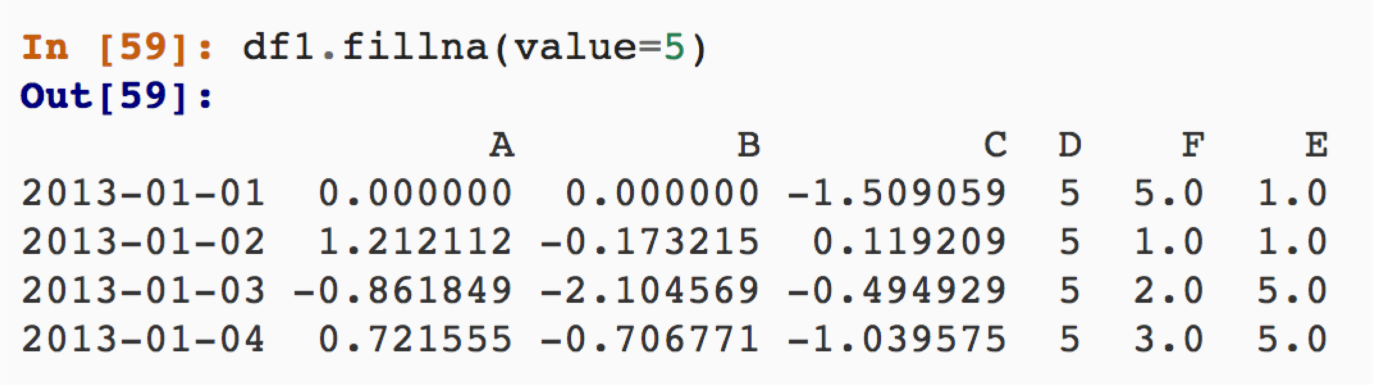
```
In [58]: df1.dropna(how='any')
Out[58]:
                   A         B         C  D    F    E
2013-01-02  1.212112 -0.173215  0.119209  5  1.0  1.0
```

How to Deal with Missing Data

- Filling missing data:
  - df.fillna(value=5)



```
In [57]: df1
Out[57]:
                   A         B         C  D    F    E
2013-01-01  0.000000  0.000000 -1.509059  5  NaN  1.0
2013-01-02  1.212112 -0.173215  0.119209  5  1.0  1.0
2013-01-03 -0.861849 -2.104569 -0.494929  5  2.0  NaN
2013-01-04  0.721555 -0.706771 -1.039575  5  3.0  NaN
```



```
In [59]: df1.fillna(value=5)
Out[59]:
                   A         B         C  D    F    E
2013-01-01  0.000000  0.000000 -1.509059  5  5.0  1.0
2013-01-02  1.212112 -0.173215  0.119209  5  1.0  1.0
2013-01-03 -0.861849 -2.104569 -0.494929  5  2.0  5.0
2013-01-04  0.721555 -0.706771 -1.039575  5  3.0  5.0
```