

# Missing Data

Let's show a few convenient methods to deal with Missing Data in pandas:

```
In [1]: 1 import numpy as np
        2 import pandas as pd
```

```
In [2]: 1 df = pd.DataFrame({'A':[1,2,np.nan],
        2                     'B':[5,np.nan,np.nan],
        3                     'C':[1,2,3]})
```

```
In [3]: 1 df
```

Out[3]:

|   | A   | B   | C |
|---|-----|-----|---|
| 0 | 1.0 | 5.0 | 1 |
| 1 | 2.0 | NaN | 2 |
| 2 | NaN | NaN | 3 |

```
In [4]: 1 df.dropna()
```

Out[4]:

|   | A   | B   | C |
|---|-----|-----|---|
| 0 | 1.0 | 5.0 | 1 |

```
In [5]: 1 df.dropna(axis=1)
```

Out[5]:

|   | C |
|---|---|
| 0 | 1 |
| 1 | 2 |
| 2 | 3 |

```
In [6]: 1 df.dropna(thresh=2)
```

Out[6]:

|   | A   | B   | C |
|---|-----|-----|---|
| 0 | 1.0 | 5.0 | 1 |
| 1 | 2.0 | NaN | 2 |

```
In [7]: 1 df.fillna(value='FILL VALUE')
```

Out[7]:

|   | A                     | B | C |
|---|-----------------------|---|---|
| 0 | 1                     | 5 | 1 |
| 1 | 2 FILL VALUE          | 2 |   |
| 2 | FILL VALUE FILL VALUE | 3 |   |

```
In [8]: 1 df['A'].fillna(value=df['A'].mean())
```

Out[8]:

|   |     |
|---|-----|
| 0 | 1.0 |
| 1 | 2.0 |
| 2 | 1.5 |

Name: A, dtype: float64

## Great Job!

```
In [ ]:
```

```
1
```