Missing Data

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

In [1]:

```
import numpy as np
import pandas as pd
```

In [3]:

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


})
```

In [4]:

```
df
```

Out[4]:

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

In [ ]:

```

```

In [9]:

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

In [5]:

```
df
```

Out[5]:

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

In [9]:

```
df.dropna(axis=0) #rows
```

Out[9]:

	A	B	C
0	1.0	5.0	1

In [8]:

```
df.dropna(axis=1)  #columns
```

Out[8]:

	C
0	1
1	2
2	3

In [12]:

```
df.dropna()
```

Out[12]:

	A	B	C
0	1.0	5.0	1

In [11]:

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

Out[11]:

	C
0	1
1	2
2	3

In [12]:

```
df
```

Out[12]:

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

In [13]:

```
df.dropna(thresh=2)
```

Out[13]:

	A	B	C
0	1.0	5.0	1
1	2.0	NaN	2

In [14]:

```
df.dropna(thresh=2)
```

Out[14]:

	A	B	C
0	1.0	5.0	1
1	2.0	NaN	2

In [16]:

```
df.dropna(axis=1,thresh=2)
```

Out[16]:

	A	C
0	1.0	1
1	2.0	2
2	NaN	3

In [18]:

```
df
```

Out[18]:

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

In [19]:

```
df.fillna(value='None')
```

Out[19]:

	A	B	C
0	1	5	1
1	2	None	2
2	None	None	3

In [21]:

```
df.fillna(value='FILL VALUE')
```

Out[21]:

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

In [ ]:

```

```

In [23]:

```
df
```

Out[23]:

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

In [24]:

```
df['A'].mean()
```

Out[24]:

1.5

In [ ]:

```
1,2,3,4,5,6

3+4/2

7/2

median 3.5




mode 1,2,3,4,5,6


mean 1,2,3,4,5,6/6

21/6 = 3.5

```

In [25]:

```
df.fillna(value=df['A'].mean())
```

Out[25]:

	A	B	C
0	1.0	5.0	1
1	2.0	1.5	2
2	1.5	1.5	3

In [26]:

```
df['A'].fillna(value=df['A'].mean())
```

Out[26]:

```
0    1.0
1    2.0
2    1.5
Name: A, dtype: float64
```

In [27]:

```
21/6
```

Out[27]:

```
3.5
```

Great Job!