# Session 4 -Handling missing values

## Missing values

```
1 import numpy as np
2 import pandas as pd
```

```
1 missing_values=['--','na','n/a']
2 df = pd.read_csv('propertydata.csv', na_values=missing_values)
3 df
```

| | PID | ST_NUM | ST_NAME | OWN_OCCUPIED | NUM_BEDROOMS | NUM_BATH | SQ_FT |
|---|---|---|---|---|---|---|---|
| **0** | 100001000.0 | 104.0 | PUTNAM | Y | 3.0 | 1 | 1000.0 |
| **1** | 100002000.0 | 197.0 | LEXINGTON | N | 3.0 | 1.5 | NaN |
| **2** | 100003000.0 | NaN | LEXINGTON | N | NaN | 1 | 850.0 |
| **3** | 100004000.0 | 201.0 | BERKELEY | 12 | 1.0 | NaN | 700.0 |
| **4** | NaN | 203.0 | BERKELEY | Y | 3.0 | 2 | 1600.0 |
| **5** | 100006000.0 | 207.0 | BERKELEY | Y | NaN | 1 | 800.0 |
| **6** | 100007000.0 | NaN | WASHINGTON | NaN | 2.0 | HURLEY | 950.0 |
| **7** | 100008000.0 | 213.0 | TREMONT | Y | 1.0 | 1 | NaN |
| **8** | 100009000.0 | 215.0 | TREMONT | Y | NaN | 2 | 1800.0 |

```
1 df['OWN_OCCUPIED'] = df['OWN_OCCUPIED'].replace(r'[0-9]+',np.nan,regex=True)
2 df
```

| | PID | ST_NUM | ST_NAME | OWN_OCCUPIED | NUM_BEDROOMS | NUM_BATH | SQ_FT |
|---|---|---|---|---|---|---|---|
| **0** | 100001000.0 | 104.0 | PUTNAM | Y | 3.0 | 1 | 1000.0 |

```
1 df['NUM_BATH'] = df['NUM_BATH'].replace(r'[A-Z]+',np.nan,regex=True)
2 df
```

| | PID | ST_NUM | ST_NAME | OWN_OCCUPIED | NUM_BEDROOMS | NUM_BATH | SQ_FT |
|---|---|---|---|---|---|---|---|
| **0** | 100001000.0 | 104.0 | PUTNAM | Y | 3.0 | 1 | 1000.0 |
| **1** | 100002000.0 | 197.0 | LEXINGTON | N | 3.0 | 1.5 | NaN |
| **2** | 100003000.0 | NaN | LEXINGTON | N | NaN | 1 | 850.0 |
| **3** | 100004000.0 | 201.0 | BERKELEY | NaN | 1.0 | NaN | 700.0 |
| **4** | NaN | 203.0 | BERKELEY | Y | 3.0 | 2 | 1600.0 |
| **5** | 100006000.0 | 207.0 | BERKELEY | Y | NaN | 1 | 800.0 |
| **6** | 100007000.0 | NaN | WASHINGTON | NaN | 2.0 | NaN | 950.0 |
| **7** | 100008000.0 | 213.0 | TREMONT | Y | 1.0 | 1 | NaN |
| **8** | 100009000.0 | 215.0 | TREMONT | Y | NaN | 2 | 1800.0 |

# Find null values

```
1 df.isnull().sum()
```

```
PID             1
ST_NUM          2
ST_NAME         0
OWN_OCCUPIED    2
NUM_BEDROOMS    3
NUM_BATH        2
SQ_FT           2
dtype: int64
```

```
1 df.isnull().sum().sum()
```

```
12
```

```
1 #fillna() - you could give val, backward fill, forward fill
2 df.fillna(0)
```

| | PID | ST_NUM | ST_NAME | OWN_OCCUPIED | NUM_BEDROOMS | NUM_BATH | SQ_FT |
|---|---|---|---|---|---|---|---|
| **0** | 100001000.0 | 104.0 | PUTNAM | Y | 3.0 | 1 | 1000.0 |
| **1** | 100002000.0 | 197.0 | LEXINGTON | N | 3.0 | 1.5 | 0.0 |
| **2** | 100003000.0 | 0.0 | LEXINGTON | N | 0.0 | 1 | 850.0 |
| **3** | 100004000.0 | 201.0 | BERKELEY | 0 | 1.0 | 0 | 700.0 |
| **4** | 0.0 | 203.0 | BERKELEY | Y | 3.0 | 2 | 1600.0 |
| **5** | 100006000.0 | 207.0 | BERKELEY | Y | 0.0 | 1 | 800.0 |
| **6** | 100007000.0 | 0.0 | WASHINGTON | 0 | 2.0 | 0 | 950.0 |

```
1 df
```

| | PID | ST_NUM | ST_NAME | OWN_OCCUPIED | NUM_BEDROOMS | NUM_BATH | SQ_FT |
|---|---|---|---|---|---|---|---|
| **0** | 100001000.0 | 104.0 | PUTNAM | Y | 3.0 | 1 | 1000.0 |
| **1** | 100002000.0 | 197.0 | LEXINGTON | N | 3.0 | 1.5 | NaN |
| **2** | 100003000.0 | NaN | LEXINGTON | N | NaN | 1 | 850.0 |
| **3** | 100004000.0 | 201.0 | BERKELEY | NaN | 1.0 | NaN | 700.0 |
| **4** | NaN | 203.0 | BERKELEY | Y | 3.0 | 2 | 1600.0 |
| **5** | 100006000.0 | 207.0 | BERKELEY | Y | NaN | 1 | 800.0 |
| **6** | 100007000.0 | NaN | WASHINGTON | NaN | 2.0 | NaN | 950.0 |
| **7** | 100008000.0 | 213.0 | TREMONT | Y | 1.0 | 1 | NaN |
| **8** | 100009000.0 | 215.0 | TREMONT | Y | NaN | 2 | 1800.0 |

```
1 #using backward fill, if last value is missing it wont get filled
2 df.fillna(method='bfill')
```

| | PID | ST_NUM | ST_NAME | OWN_OCCUPIED | NUM_BEDROOMS | NUM_BATH | SQ_FT |
|---|---|---|---|---|---|---|---|
| 0 | 100001000.0 | 104.0 | PUTNAM | Y | 3.0 | 1 | 1000.0 |

```
1 df.fillna(method='ffill')
```

| | PID | ST_NUM | ST_NAME | OWN_OCCUPIED | NUM_BEDROOMS | NUM_BATH | SQ_FT |
|---|---|---|---|---|---|---|---|
| **0** | 100001000.0 | 104.0 | PUTNAM | Y | 3.0 | 1 | 1000.0 |
| **1** | 100002000.0 | 197.0 | LEXINGTON | N | 3.0 | 1.5 | 1000.0 |
| **2** | 100003000.0 | 197.0 | LEXINGTON | N | 3.0 | 1 | 850.0 |
| **3** | 100004000.0 | 201.0 | BERKELEY | N | 1.0 | 1 | 700.0 |
| **4** | 100004000.0 | 203.0 | BERKELEY | Y | 3.0 | 2 | 1600.0 |
| **5** | 100006000.0 | 207.0 | BERKELEY | Y | 3.0 | 1 | 800.0 |
| **6** | 100007000.0 | 207.0 | WASHINGTON | Y | 2.0 | 1 | 950.0 |
| **7** | 100008000.0 | 213.0 | TREMONT | Y | 1.0 | 1 | 950.0 |
| **8** | 100009000.0 | 215.0 | TREMONT | Y | 1.0 | 2 | 1800.0 |

```
1 df.replace(to_replace=np.nan, value= -99) #same as fillna(-99)
```

| | PID | ST_NUM | ST_NAME | OWN_OCCUPIED | NUM_BEDROOMS | NUM_BATH | SQ_FT |
|---|---|---|---|---|---|---|---|
| **0** | 100001000.0 | 104.0 | PUTNAM | Y | 3.0 | 1 | 1000.0 |
| **1** | 100002000.0 | 197.0 | LEXINGTON | N | 3.0 | 1.5 | -99.0 |
| **2** | 100003000.0 | -99.0 | LEXINGTON | N | -99.0 | 1 | 850.0 |
| **3** | 100004000.0 | 201.0 | BERKELEY | -99 | 1.0 | -99 | 700.0 |
| **4** | -99.0 | 203.0 | BERKELEY | Y | 3.0 | 2 | 1600.0 |
| **5** | 100006000.0 | 207.0 | BERKELEY | Y | -99.0 | 1 | 800.0 |
| **6** | 100007000.0 | -99.0 | WASHINGTON | -99 | 2.0 | -99 | 950.0 |
| **7** | 100008000.0 | 213.0 | TREMONT | Y | 1.0 | 1 | -99.0 |
| **8** | 100009000.0 | 215.0 | TREMONT | Y | -99.0 | 2 | 1800.0 |

```
1 df.dropna() #no parameters - if there are nan values, the row will get deleted
```

| | PID | ST_NUM | ST_NAME | OWN_OCCUPIED | NUM_BEDROOMS | NUM_BATH | SQ_FT |
|---|---|---|---|---|---|---|---|
| **0** | 100001000.0 | 104.0 | PUTNAM | Y | 3.0 | 1 | 1000.0 |

```
1 df.dropna(how='all') #if the row fully consists of nan values, the row gets deleted
```

| | PID | ST_NUM | ST_NAME | OWN_OCCUPIED | NUM_BEDROOMS | NUM_BATH | SQ_FT |
|---|---|---|---|---|---|---|---|
| **0** | 100001000.0 | 104.0 | PUTNAM | Y | 3.0 | 1 | 1000.0 |
| **1** | 100002000.0 | 197.0 | LEXINGTON | N | 3.0 | 1.5 | NaN |
| **2** | 100003000.0 | NaN | LEXINGTON | N | NaN | 1 | 850.0 |
| **3** | 100004000.0 | 201.0 | BERKELEY | NaN | 1.0 | NaN | 700.0 |
| **4** | NaN | 203.0 | BERKELEY | Y | 3.0 | 2 | 1600.0 |
| **5** | 100006000.0 | 207.0 | BERKELEY | Y | NaN | 1 | 800.0 |
| **6** | 100007000.0 | NaN | WASHINGTON | NaN | 2.0 | NaN | 950.0 |
| **7** | 100008000.0 | 213.0 | TREMONT | Y | 1.0 | 1 | NaN |
| **8** | 100009000.0 | 215.0 | TREMONT | Y | NaN | 2 | 1800.0 |

```
1 df.dropna(axis=1) #drop columnwise, if nan value present
```

| | ST_NAME |
|---|---|
| **0** | PUTNAM |
| **1** | LEXINGTON |
| **2** | LEXINGTON |
| **3** | BERKELEY |
| **4** | BERKELEY |
| **5** | BERKELEY |
| **6** | WASHINGTON |
| **7** | TREMONT |
| **8** | TREMONT |

```
1 new_data = df.dropna(axis=0, how='any')
```

```
1 nt.com/ktisha/c21e73a1bd1700294ef790c56c8aec1f/raw/819b69b5736821ccee93d05b51de0510bea0029
```

```
1 data_set
```

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 6 | 148 | 72 | 35 | 0 | 33.6 | 0.627 | 50 | 1 |
| **1** | 1 | 85 | 66 | 29 | 0 | 26.6 | 0.351 | 31 | 0 |
| **2** | 8 | 183 | 64 | 0 | 0 | 23.3 | 0.672 | 32 | 1 |
| **3** | 1 | 89 | 66 | 23 | 94 | 28.1 | 0.167 | 21 | 0 |
| **4** | 0 | 137 | 40 | 35 | 168 | 43.1 | 2.288 | 33 | 1 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **763** | 10 | 101 | 76 | 48 | 180 | 32.9 | 0.171 | 63 | 0 |
| **764** | 2 | 122 | 70 | 27 | 0 | 36.8 | 0.340 | 27 | 0 |
| **765** | 5 | 121 | 72 | 23 | 112 | 26.2 | 0.245 | 30 | 0 |

```
1 data_set[[1,2,3,4,5]]=data_set[[1,2,3,4,5]].replace(0,np.nan)
2 print(data_set.isnull().sum())
```

```
0      0
1      5
2     35
3    227
4    374
5     11
6      0
7      0
8      0
dtype: int64
```

```
1 data_set.head()
```

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 6 | 148.0 | 72.0 | 35.0 | NaN | 33.6 | 0.627 | 50 | 1 |
| **1** | 1 | 85.0 | 66.0 | 29.0 | NaN | 26.6 | 0.351 | 31 | 0 |
| **2** | 8 | 183.0 | 64.0 | NaN | NaN | 23.3 | 0.672 | 32 | 1 |
| **3** | 1 | 89.0 | 66.0 | 23.0 | 94.0 | 28.1 | 0.167 | 21 | 0 |
| **4** | 0 | 137.0 | 40.0 | 35.0 | 168.0 | 43.1 | 2.288 | 33 | 1 |

```
1 data_set.fillna(data_set.mean(),inplace=True)
```

```
1 data_set
```

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 6 | 148.0 | 72.0 | 35.00000 | 155.548223 | 33.6 | 0.627 | 50 | 1 |
| **1** | 1 | 85.0 | 66.0 | 29.00000 | 155.548223 | 26.6 | 0.351 | 31 | 0 |
| **2** | 8 | 183.0 | 64.0 | 29.15342 | 155.548223 | 23.3 | 0.672 | 32 | 1 |
| **3** | 1 | 89.0 | 66.0 | 23.00000 | 94.000000 | 28.1 | 0.167 | 21 | 0 |
| **4** | 0 | 137.0 | 40.0 | 35.00000 | 168.000000 | 43.1 | 2.288 | 33 | 1 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **763** | 10 | 101.0 | 76.0 | 48.00000 | 180.000000 | 32.9 | 0.171 | 63 | 0 |
| **764** | 2 | 122.0 | 70.0 | 27.00000 | 155.548223 | 36.8 | 0.340 | 27 | 0 |
| **765** | 5 | 121.0 | 72.0 | 23.00000 | 112.000000 | 26.2 | 0.245 | 30 | 0 |
| **766** | 1 | 126.0 | 60.0 | 29.15342 | 155.548223 | 30.1 | 0.349 | 47 | 1 |
| **767** | 1 | 93.0 | 70.0 | 31.00000 | 155.548223 | 30.4 | 0.315 | 23 | 0 |

1

✓ 0s completed at 2:15 PM