DS Essential part 2 Data Preparation Basics

August 14, 2022

0.0.1 Data Preparation Basics

Treating Missing Values

[2]:
```
import pandas as pd
import numpy as np
from pandas import Series, DataFrame
```

Figure out missing data

[4]:
```
missing = np.nan
series_obj = Series(['row 1','row 2',missing, 'row 4','row 5','row␣
 ↪6',missing,'row 8'])
series_obj
```

[4]:
```
0    row 1
1    row 2
2      NaN
3    row 4
4    row 5
5    row 6
6      NaN
7    row 8
dtype: object
```

[5]:
```
series_obj.isnull()
```

[5]:
```
0    False
1    False
2     True
3    False
4    False
5    False
6     True
7    False
dtype: bool
```

[7]:
```
### FIlling in Missing values
```

[8]:
```
np.random.seed(25)
DF_obj = DataFrame(np.random.rand(36).reshape(6,6))
DF_obj
```

[8]:

	0	1	2	3	4	5
0	0.870124	0.582277	0.278839	0.185911	0.411100	0.117376
1	0.684969	0.437611	0.556229	0.367080	0.402366	0.113041
2	0.447031	0.585445	0.161985	0.520719	0.326051	0.699186
3	0.366395	0.836375	0.481343	0.516502	0.383048	0.997541
4	0.514244	0.559053	0.034450	0.719930	0.421004	0.436935
5	0.281701	0.900274	0.669612	0.456069	0.289804	0.525819

[13]:
```
DF_obj.loc[3:5,0] = missing
DF_obj
```

[13]:

	0	1	2	3	4	5
0	0.870124	0.582277	0.278839	0.185911	0.411100	0.117376
1	0.684969	0.437611	0.556229	0.367080	0.402366	0.113041
2	0.447031	0.585445	0.161985	0.520719	0.326051	0.699186
3	NaN	0.836375	0.481343	0.516502	0.383048	0.997541
4	NaN	0.559053	0.034450	0.719930	0.421004	0.436935
5	NaN	0.900274	0.669612	0.456069	0.289804	0.525819

[28]:
```
#assingning nan to rows and columns
DF_obj.loc[1:4,5]= missing
DF_obj
```

[28]:

	0	1	2	3	4	5
0	0.870124	0.582277	0.278839	0.185911	0.411100	0.117376
1	0.684969	0.437611	0.556229	0.367080	0.402366	NaN
2	0.447031	0.585445	0.161985	0.520719	0.326051	NaN
3	NaN	0.836375	0.481343	0.516502	0.383048	NaN
4	NaN	0.559053	0.034450	0.719930	0.421004	NaN
5	NaN	0.900274	0.669612	0.456069	0.289804	0.525819

[27]:
```
#filling 0 to nan
filled_DF=DF_obj.fillna(0)
```

[18]:
```
filled_DF
```

[18]:

	0	1	2	3	4	5
0	0.870124	0.582277	0.278839	0.185911	0.411100	0.117376
1	0.684969	0.437611	0.556229	0.367080	0.402366	0.000000
2	0.447031	0.585445	0.161985	0.520719	0.326051	0.000000
3	0.000000	0.836375	0.481343	0.516502	0.383048	0.000000
4	0.000000	0.559053	0.034450	0.719930	0.421004	0.000000
5	0.000000	0.900274	0.669612	0.456069	0.289804	0.525819

[26]:
```
#filling and assign value
filled_DF = DF_obj.fillna({0:0.1, 5:1.25})
```

[21]:
```
filled_DF
```

[21]:

	0	1	2	3	4	5
0	0.870124	0.582277	0.278839	0.185911	0.411100	0.117376
1	0.684969	0.437611	0.556229	0.367080	0.402366	1.250000
2	0.447031	0.585445	0.161985	0.520719	0.326051	1.250000
3	0.100000	0.836375	0.481343	0.516502	0.383048	1.250000
4	0.100000	0.559053	0.034450	0.719930	0.421004	1.250000
5	0.100000	0.900274	0.669612	0.456069	0.289804	0.525819

[25]:
```
#forward fill method
filled_DF = DF_obj.fillna(method='ffill')
filled_DF
```

[25]:

	0	1	2	3	4	5
0	0.870124	0.582277	0.278839	0.185911	0.411100	0.117376
1	0.684969	0.437611	0.556229	0.367080	0.402366	0.117376
2	0.447031	0.585445	0.161985	0.520719	0.326051	0.117376
3	0.447031	0.836375	0.481343	0.516502	0.383048	0.117376
4	0.447031	0.559053	0.034450	0.719930	0.421004	0.117376
5	0.447031	0.900274	0.669612	0.456069	0.289804	0.525819

Counting missing values

[29]:
```
DF_obj.loc[1:4,5]= missing
DF_obj
```

[29]:

	0	1	2	3	4	5
0	0.870124	0.582277	0.278839	0.185911	0.411100	0.117376
1	0.684969	0.437611	0.556229	0.367080	0.402366	NaN
2	0.447031	0.585445	0.161985	0.520719	0.326051	NaN
3	NaN	0.836375	0.481343	0.516502	0.383048	NaN
4	NaN	0.559053	0.034450	0.719930	0.421004	NaN
5	NaN	0.900274	0.669612	0.456069	0.289804	0.525819

[30]:
```
DF_obj.isnull().sum()
```

[30]:
```
0    3
1    0
2    0
3    0
4    0
5    4
dtype: int64
```

[35]:
```
#droping all  rows and colums contrains nan values
DF_no_nan= DF_obj.dropna()
DF_no_nan
```

[35]:

	0	1	2	3	4	5
0	0.870124	0.582277	0.278839	0.185911	0.4111	0.117376

[34]:
```
#Droping column which have nan
DF_no_nan= DF_obj.dropna(axis = 1)
DF_no_nan
```

[34]:

	1	2	3	4
0	0.582277	0.278839	0.185911	0.411100
1	0.437611	0.556229	0.367080	0.402366
2	0.585445	0.161985	0.520719	0.326051
3	0.836375	0.481343	0.516502	0.383048
4	0.559053	0.034450	0.719930	0.421004
5	0.900274	0.669612	0.456069	0.289804