handling_missing_data_fillna_dropna_interpolate(4.1)

January 12, 2020

##

Handling Missing Data - fillna, interpolate, dropna

[1]:
```
import pandas as pd
df = pd.read_csv("weather_data.csv",parse_dates=['day'])
type(df.day[0])
df
```

[1]:

	day	temperature	windspeed	event
0	2017-01-01	32.0	6.0	Rain
1	2017-01-04	NaN	9.0	Sunny
2	2017-01-05	28.0	NaN	Snow
3	2017-01-06	NaN	7.0	NaN
4	2017-01-07	32.0	NaN	Rain
5	2017-01-08	NaN	NaN	Sunny
6	2017-01-09	NaN	NaN	NaN
7	2017-01-10	34.0	8.0	Cloudy
8	2017-01-11	40.0	12.0	Sunny

[2]:
```
df.set_index('day',inplace=True)
df
```

[2]:

day	temperature	windspeed	event
2017-01-01	32.0	6.0	Rain
2017-01-04	NaN	9.0	Sunny
2017-01-05	28.0	NaN	Snow
2017-01-06	NaN	7.0	NaN
2017-01-07	32.0	NaN	Rain
2017-01-08	NaN	NaN	Sunny
2017-01-09	NaN	NaN	NaN
2017-01-10	34.0	8.0	Cloudy
2017-01-11	40.0	12.0	Sunny

0.1 fillna

Fill all NaN with one specific value

[3]:
```
new_df = df.fillna(0)
new_df
```

[3]:

day	temperature	windspeed	event
2017-01-01	32.0	6.0	Rain
2017-01-04	0.0	9.0	Sunny
2017-01-05	28.0	0.0	Snow
2017-01-06	0.0	7.0	0
2017-01-07	32.0	0.0	Rain
2017-01-08	0.0	0.0	Sunny
2017-01-09	0.0	0.0	0
2017-01-10	34.0	8.0	Cloudy
2017-01-11	40.0	12.0	Sunny

Fill na using column names and dict

[4]:
```
new_df = df.fillna({
        'temperature': 0,
        'windspeed': 0,
        'event': 'No Event'
    })
new_df
```

[4]:

day	temperature	windspeed	event
2017-01-01	32.0	6.0	Rain
2017-01-04	0.0	9.0	Sunny
2017-01-05	28.0	0.0	Snow
2017-01-06	0.0	7.0	No Event
2017-01-07	32.0	0.0	Rain
2017-01-08	0.0	0.0	Sunny
2017-01-09	0.0	0.0	No Event
2017-01-10	34.0	8.0	Cloudy
2017-01-11	40.0	12.0	Sunny

Use method to determine how to fill na values

[5]:
```
new_df = df.fillna(method="ffill")
new_df
```

[5]:

day	temperature	windspeed	event
2017-01-01	32.0	6.0	Rain
2017-01-04	32.0	9.0	Sunny
2017-01-05	28.0	9.0	Snow
2017-01-06	28.0	7.0	Snow
2017-01-07	32.0	7.0	Rain

```
2017-01-08         32.0        7.0   Sunny
2017-01-09         32.0        7.0   Sunny
2017-01-10         34.0        8.0  Cloudy
2017-01-11         40.0       12.0   Sunny
```

[6]:
```
new_df = df.fillna(method="bfill")
new_df
```

[6]:
```
            temperature  windspeed   event
day
2017-01-01         32.0        6.0    Rain
2017-01-04         28.0        9.0   Sunny
2017-01-05         28.0        7.0    Snow
2017-01-06         32.0        7.0    Rain
2017-01-07         32.0        8.0    Rain
2017-01-08         34.0        8.0   Sunny
2017-01-09         34.0        8.0  Cloudy
2017-01-10         34.0        8.0  Cloudy
2017-01-11         40.0       12.0   Sunny
```

Use of axis

[7]:
```
new_df = df.fillna(method="bfill", axis="columns") # axis is either "index" or␣
↪"columns"
new_df
```

[7]:
```
            temperature windspeed   event
day
2017-01-01           32         6    Rain
2017-01-04            9         9   Sunny
2017-01-05           28     Snow    Snow
2017-01-06            7         7     NaN
2017-01-07           32      Rain    Rain
2017-01-08       Sunny     Sunny   Sunny
2017-01-09          NaN       NaN     NaN
2017-01-10           34         8  Cloudy
2017-01-11           40        12   Sunny
```

limit parameter

[8]:
```
new_df = df.fillna(method="ffill",limit=1)
new_df
```

[8]:
```
            temperature  windspeed   event
day
2017-01-01         32.0        6.0    Rain
2017-01-04         32.0        9.0   Sunny
2017-01-05         28.0        9.0    Snow
```

```
2017-01-06         28.0        7.0    Snow
2017-01-07         32.0        7.0    Rain
2017-01-08         32.0        NaN    Sunny
2017-01-09          NaN        NaN    Sunny
2017-01-10         34.0        8.0   Cloudy
2017-01-11         40.0       12.0    Sunny
```

0.1.1 interpolate

[9]:
```
new_df = df.interpolate()
new_df
```

[9]:
```
            temperature  windspeed   event
day
2017-01-01    32.000000       6.00    Rain
2017-01-04    30.000000       9.00   Sunny
2017-01-05    28.000000       8.00    Snow
2017-01-06    30.000000       7.00     NaN
2017-01-07    32.000000       7.25    Rain
2017-01-08    32.666667       7.50   Sunny
2017-01-09    33.333333       7.75     NaN
2017-01-10    34.000000       8.00   Cloudy
2017-01-11    40.000000      12.00    Sunny
```

[10]:
```
new_df = df.interpolate(method="time")
new_df
```

[10]:
```
            temperature  windspeed   event
day
2017-01-01    32.000000       6.00    Rain
2017-01-04    29.000000       9.00   Sunny
2017-01-05    28.000000       8.00    Snow
2017-01-06    30.000000       7.00     NaN
2017-01-07    32.000000       7.25    Rain
2017-01-08    32.666667       7.50   Sunny
2017-01-09    33.333333       7.75     NaN
2017-01-10    34.000000       8.00   Cloudy
2017-01-11    40.000000      12.00    Sunny
```

Notice that in above temperature on 2017-01-04 is 29 instead of 30 (in plain linear interpolate)

There are many other methods for interpolation such as quadratic, piecewise_polynomial, cubic etc. Just google "dataframe interpolate" to see complete documentation

0.1.2 dropna

[11]:
```
new_df = df.dropna()
new_df
```

[11]:

day	temperature	windspeed	event
2017-01-01	32.0	6.0	Rain
2017-01-10	34.0	8.0	Cloudy
2017-01-11	40.0	12.0	Sunny

[12]:
```
new_df = df.dropna(how='all')
new_df

#any' : If any NA values are present, drop that row or column.
#'all' : If all values are NA, drop that row or column.
#thresh : int, optional
```

[12]:

day	temperature	windspeed	event
2017-01-01	32.0	6.0	Rain
2017-01-04	NaN	9.0	Sunny
2017-01-05	28.0	NaN	Snow
2017-01-06	NaN	7.0	NaN
2017-01-07	32.0	NaN	Rain
2017-01-08	NaN	NaN	Sunny
2017-01-10	34.0	8.0	Cloudy
2017-01-11	40.0	12.0	Sunny

[19]:
```
new_df = df.dropna(thresh=3)
new_df
```

[19]:

day	temperature	windspeed	event
2017-01-01	32.0	6.0	Rain
2017-01-10	34.0	8.0	Cloudy
2017-01-11	40.0	12.0	Sunny

0.1.3 Inserting Missing Dates

[14]:
```
dt = pd.date_range("01-01-2017","01-11-2017")
idx = pd.DatetimeIndex(dt)
df.reindex(idx)
```

[14]:

	temperature	windspeed	event
2017-01-01	32.0	6.0	Rain
2017-01-02	NaN	NaN	NaN

2017-01-03	NaN	NaN	NaN
2017-01-04	NaN	9.0	Sunny
2017-01-05	28.0	NaN	Snow
2017-01-06	NaN	7.0	NaN
2017-01-07	32.0	NaN	Rain
2017-01-08	NaN	NaN	Sunny
2017-01-09	NaN	NaN	NaN
2017-01-10	34.0	8.0	Cloudy
2017-01-11	40.0	12.0	Sunny