In [1]:

```
# Pandas
import numpy as np
import pandas as pd
```

In [6]:

```
# INTERPOLATE
df = pd.read_csv("sets/stu_results.csv")
df
```

Out[6]:

	Student ID	Class	Study hrs	Sleeping hrs	Social Media usage hrs	Mobile Games hrs	Percantege
0	1001	10	2	9	3	5	50
1	1002	10	6	8	2	0	80
2	1003	10	3	8	2	4	60
3	1004	11	0	10	1	5	45
4	1005	11	4	7	2	0	75
5	1006	11	10	7	0	0	96
6	1007	12	4	6	0	0	80
7	1008	12	10	6	2	0	90
8	1009	12	2	8	2	4	60
9	1010	12	6	9	1	0	85

In [ ]:

```

```

In [11]:

```
df1 = pd.read_csv("sets/stu_results.csv")
df1
```

Out[11]:

	Student ID	Class	Study hrs	Sleeping hrs	Social Media usage hrs	Mobile Games hrs	Percantege
0	1001	10	2.0	9.0	3.0	5.0	50.0
1	1002	10	6.0	8.0	2.0	0.0	40.0
2	1003	10	NaN	8.0	NaN	4.0	NaN
3	1004	11	NaN	10.0	1.0	NaN	NaN
4	1005	11	NaN	7.0	2.0	0.0	NaN
5	1006	11	10.0	7.0	0.0	0.0	NaN
6	1007	12	4.0	NaN	0.0	0.0	80.0
7	1008	12	10.0	NaN	2.0	0.0	90.0
8	1009	12	2.0	8.0	2.0	4.0	60.0
9	1010	12	6.0	9.0	1.0	0.0	85.0

In [ ]:

```
# INTERPOLATE => function is basically used to fill NaN values in dataframes or in series
# important point => Very powerful functon
```

In [ ]:

```
# df.interpolate()

'''
method='linear',
    axis=0,
    limit=None,
    inplace=False,
    limit_direction='forward',
    limit_area=None,
'''
```

In [12]:

```
df1
```

Out[12]:

	Student ID	Class	Study hrs	Sleeping hrs	Social Media usage hrs	Mobile Games hrs	Percantege
0	1001	10	2.0	9.0	3.0	5.0	50.0
1	1002	10	6.0	8.0	2.0	0.0	40.0
2	1003	10	NaN	8.0	NaN	4.0	NaN
3	1004	11	NaN	10.0	1.0	NaN	NaN
4	1005	11	NaN	7.0	2.0	0.0	NaN
5	1006	11	10.0	7.0	0.0	0.0	NaN
6	1007	12	4.0	NaN	0.0	0.0	80.0
7	1008	12	10.0	NaN	2.0	0.0	90.0
8	1009	12	2.0	8.0	2.0	4.0	60.0
9	1010	12	6.0	9.0	1.0	0.0	85.0

In [13]:

```
df1.interpolate()
```

Out[13]:

	Student ID	Class	Study hrs	Sleeping hrs	Social Media usage hrs	Mobile Games hrs	Percantege
0	1001	10	2.0	9.000000	3.0	5.0	50.0
1	1002	10	6.0	8.000000	2.0	0.0	40.0
2	1003	10	7.0	8.000000	1.5	4.0	48.0
3	1004	11	8.0	10.000000	1.0	2.0	56.0
4	1005	11	9.0	7.000000	2.0	0.0	64.0
5	1006	11	10.0	7.000000	0.0	0.0	72.0
6	1007	12	4.0	7.333333	0.0	0.0	80.0
7	1008	12	10.0	7.666667	2.0	0.0	90.0
8	1009	12	2.0	8.000000	2.0	4.0	60.0
9	1010	12	6.0	9.000000	1.0	0.0	85.0

In [ ]:

```

```

In [14]:

```
df1.interpolate(method='linear')
```

Out[14]:

	Student ID	Class	Study hrs	Sleeping hrs	Social Media usage hrs	Mobile Games hrs	Percantege
0	1001	10	2.0	9.000000	3.0	5.0	50.0
1	1002	10	6.0	8.000000	2.0	0.0	40.0
2	1003	10	7.0	8.000000	1.5	4.0	48.0
3	1004	11	8.0	10.000000	1.0	2.0	56.0
4	1005	11	9.0	7.000000	2.0	0.0	64.0
5	1006	11	10.0	7.000000	0.0	0.0	72.0
6	1007	12	4.0	7.333333	0.0	0.0	80.0
7	1008	12	10.0	7.666667	2.0	0.0	90.0
8	1009	12	2.0	8.000000	2.0	4.0	60.0
9	1010	12	6.0	9.000000	1.0	0.0	85.0

In [ ]:

```

```

In [15]:

```
# Adding Date
df2 = pd.read_csv("sets/stu_results.csv")
df2
```

Out[15]:

	Date	Student ID	Class	Study hrs	Sleeping hrs	Social Media usage hrs	Mobile Games hrs	Percantege
0	01-10-2020	1001	10	2.0	9.0	3.0	5.0	50.0
1	02-10-2020	1002	10	6.0	8.0	2.0	0.0	40.0
2	03-10-2020	1003	10	NaN	8.0	NaN	4.0	NaN
3	06-10-2020	1004	11	NaN	10.0	1.0	NaN	NaN
4	08-10-2020	1005	11	NaN	7.0	2.0	0.0	NaN
5	09-10-2020	1006	11	10.0	7.0	0.0	0.0	NaN
6	10-10-2020	1007	12	4.0	NaN	0.0	0.0	80.0
7	11-10-2020	1008	12	10.0	NaN	2.0	0.0	90.0
8	12-10-2020	1009	12	2.0	8.0	2.0	4.0	60.0
9	13-10-2020	1010	12	6.0	9.0	1.0	0.0	85.0

In [17]:

```
df2.interpolate(method='time')
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-17-faeba8f5de91> in <module>
----> 1 df2.interpolate(method='time')

~\Anaconda3\lib\site-packages\pandas\core\generic.py in interpolate(self, me
thod, axis, limit, inplace, limit_direction, limit_area, downcast, **kwargs)
   6833                                     limit_area=limit_area,
   6834                                     inplace=inplace, downcast=downca
st,
-> 6835                                     **kwargs)
   6836
   6837         if inplace:

~\Anaconda3\lib\site-packages\pandas\core\internals\managers.py in interpola
te(self, **kwargs)
    517
    518     def interpolate(self, **kwargs):
--> 519         return self.apply('interpolate', **kwargs)
    520
    521     def shift(self, **kwargs):

~\Anaconda3\lib\site-packages\pandas\core\internals\managers.py in apply(sel
f, f, axes, filter, do_integrity_check, consolidate, **kwargs)
    393                                                 copy=align_copy)
    394
--> 395             applied = getattr(b, f)(**kwargs)
    396             result_blocks = _extend_blocks(applied, result_blocks)
    397

~\Anaconda3\lib\site-packages\pandas\core\internals\blocks.py in interpolate
(self, method, axis, index, values, inplace, limit, limit_direction, limit_a
rea, fill_value, coerce, downcast, **kwargs)
   1144                                     limit_area=limit_area,
   1145                                     fill_value=fill_value, inplace=
inplace,
-> 1146                                     downcast=downcast, **kwargs)
   1147
   1148         raise ValueError("invalid method '{0}' to interpolate.".form
at(method))

~\Anaconda3\lib\site-packages\pandas\core\internals\blocks.py in _interpolat
e(self, method, index, values, fill_value, axis, limit, limit_direction, lim
it_area, inplace, downcast, **kwargs)
   1210
   1211         # interp each column independently
-> 1212         interp_values = np.apply_along_axis(func, axis, data)
   1213
   1214         blocks = [self.make_block_same_class(interp_values)]

<__array_function__ internals> in apply_along_axis(*args, **kwargs)

~\Anaconda3\lib\site-packages\numpy\lib\shape_base.py in apply_along_axis(fu
nc1d, axis, arr, *args, **kwargs)
    377     except StopIteration:
    378         raise ValueError('Cannot apply_along_axis when any iteration
dimensions are 0')
```

```
--> 379     res = asanyarray(func1d(inarr_view[ind0], *args, **kwargs))
    380
    381     # build a buffer for storing evaluations of func1d.

~\Anaconda3\lib\site-packages\pandas\core\internals\blocks.py in func(x)
   1207                                             limit_area=limit_area,
   1208                                             fill_value=fill_value,
-> 1209                                             bounds_error=False, **kwar
gs)
   1210
   1211         # interp each column independently

~\Anaconda3\lib\site-packages\pandas\core\missing.py in interpolate_1d(xvalu
es, yvalues, method, limit, limit_direction, limit_area, fill_value, bounds_
error, order, **kwargs)
    137         if not getattr(xvalues, 'is_all_dates', None):
    138             # if not issubclass(xvalues.dtype.type, np.datetime64):
--> 139             raise ValueError('time-weighted interpolation only works
'
    140                              'on Series or DataFrames with a '
    141                              'DatetimeIndex')

ValueError: time-weighted interpolation only works on Series or DataFrames w
ith a DatetimeIndex
```

In [18]:

```
type(df2.Date[0])
```

Out[18]:

```
str
```

In [ ]:

```
```

In [19]:

```
# Adding str date into DatetimeValue
df3 = pd.read_csv("sets/stu_results.csv",parse_dates=['Date'],)
df3
```

Out[19]:

	Date	Student ID	Class	Study hrs	Sleeping hrs	Social Media usage hrs	Mobile Games hrs	Percantege
0	2020-01-10	1001	10	2.0	9.0	3.0	5.0	50.0
1	2020-02-10	1002	10	6.0	8.0	2.0	0.0	40.0
2	2020-03-10	1003	10	NaN	8.0	NaN	4.0	NaN
3	2020-06-10	1004	11	NaN	10.0	1.0	NaN	NaN
4	2020-08-10	1005	11	NaN	7.0	2.0	0.0	NaN
5	2020-09-10	1006	11	10.0	7.0	0.0	0.0	NaN
6	2020-10-10	1007	12	4.0	NaN	0.0	0.0	80.0
7	2020-11-10	1008	12	10.0	NaN	2.0	0.0	90.0
8	2020-12-10	1009	12	2.0	8.0	2.0	4.0	60.0
9	2020-10-13	1010	12	6.0	9.0	1.0	0.0	85.0

In [21]:

```
type(df3.Date[0])
```

Out[21]:

```
pandas._libs.tslibs.timestamps.Timestamp
```

In [ ]:

```

```

In [22]:

```
df4 = pd.read_csv("sets/stu_results.csv",parse_dates=['Date'],index_col='Date')
df4
```

Out[22]:

Date	Student ID	Class	Study hrs	Sleeping hrs	Social Media usage hrs	Mobile Games hrs	Percantege
2020-01-10	1001	10	2.0	9.0	3.0	5.0	50.0
2020-02-10	1002	10	6.0	8.0	2.0	0.0	40.0
2020-03-10	1003	10	NaN	8.0	NaN	4.0	NaN
2020-06-10	1004	11	NaN	10.0	1.0	NaN	NaN
2020-08-10	1005	11	NaN	7.0	2.0	0.0	NaN
2020-09-10	1006	11	10.0	7.0	0.0	0.0	NaN
2020-10-10	1007	12	4.0	NaN	0.0	0.0	80.0
2020-11-10	1008	12	10.0	NaN	2.0	0.0	90.0
2020-12-10	1009	12	2.0	8.0	2.0	4.0	60.0
2020-10-13	1010	12	6.0	9.0	1.0	0.0	85.0

In [24]:

```
df4.interpolate(method='time')
```

Out[24]:

Date	Student ID	Class	Study hrs	Sleeping hrs	Social Media usage hrs	Mobile Games hrs	Percantege
2020-01-10	1001	10	2.000000	9.00000	3.000000	5.000000	50.000000
2020-02-10	1002	10	6.000000	8.00000	2.000000	0.000000	40.000000
2020-03-10	1003	10	6.544601	8.00000	1.760331	4.000000	44.773663
2020-06-10	1004	11	8.272300	10.00000	1.000000	1.594771	59.917695
2020-08-10	1005	11	9.417840	7.00000	2.000000	0.000000	69.958848
2020-09-10	1006	11	10.000000	7.00000	0.000000	0.000000	75.061728
2020-10-10	1007	12	4.000000	7.32967	0.000000	0.000000	80.000000
2020-11-10	1008	12	10.000000	9.00000	2.000000	0.000000	90.000000
2020-12-10	1009	12	2.000000	8.00000	2.000000	4.000000	60.000000
2020-10-13	1010	12	6.000000	9.00000	1.000000	0.000000	85.000000

In [ ]:

```

```

In [ ]:

```

```

In [26]:

```
df5 = pd.read_csv("sets/stu_results.csv",parse_dates=['Date'],index_col='Class')
df5
```

Out[26]:

Class	Date	Student ID	Study hrs	Sleeping hrs	Social Media usage hrs	Mobile Games hrs	Percantege
10	2020-01-10	1001	2.0	9.0	3.0	5.0	50.0
10	2020-02-10	1002	6.0	8.0	2.0	0.0	40.0
10	2020-03-10	1003	NaN	8.0	NaN	4.0	NaN
11	2020-06-10	1004	NaN	10.0	1.0	NaN	NaN
11	2020-08-10	1005	NaN	7.0	2.0	0.0	NaN
11	2020-09-10	1006	10.0	7.0	0.0	0.0	NaN
12	2020-10-10	1007	4.0	NaN	0.0	0.0	80.0
12	2020-11-10	1008	10.0	NaN	2.0	0.0	90.0
12	2020-12-10	1009	2.0	8.0	2.0	4.0	60.0
12	2020-10-13	1010	6.0	9.0	1.0	0.0	85.0

In [28]:

```
df5.interpolate(method='index')
```

Out[28]:

Class	Date	Student ID	Study hrs	Sleeping hrs	Social Media usage hrs	Mobile Games hrs	Percantege
10	2020-01-10	1001	2.0	9.0	3.0	5.0	50.0
10	2020-02-10	1002	6.0	8.0	2.0	0.0	40.0
10	2020-03-10	1003	6.0	8.0	2.0	4.0	40.0
11	2020-06-10	1004	10.0	10.0	1.0	0.0	60.0
11	2020-08-10	1005	10.0	7.0	2.0	0.0	60.0
11	2020-09-10	1006	10.0	7.0	0.0	0.0	60.0
12	2020-10-10	1007	4.0	9.0	0.0	0.0	80.0
12	2020-11-10	1008	10.0	9.0	2.0	0.0	90.0
12	2020-12-10	1009	2.0	8.0	2.0	4.0	60.0
12	2020-10-13	1010	6.0	9.0	1.0	0.0	85.0

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [32]:

```
df6 = pd.read_csv("sets/stu_results.csv",parse_dates=['Date'])
df6
```

Out[32]:

	Date	Student ID	Class	Study hrs	Sleeping hrs	Social Media usage hrs	Mobile Games hrs	Percantege
0	2020-01-10	1001	10	2.0	9.0	3.0	5.0	50.0
1	2020-02-10	1002	10	6.0	8.0	2.0	0.0	40.0
2	2020-03-10	1003	10	NaN	8.0	NaN	4.0	NaN
3	2020-06-10	1004	11	NaN	10.0	1.0	NaN	NaN
4	2020-08-10	1005	11	NaN	7.0	2.0	0.0	NaN
5	2020-09-10	1006	11	10.0	7.0	0.0	0.0	NaN
6	2020-10-10	1007	12	4.0	NaN	0.0	0.0	80.0
7	2020-11-10	1008	12	10.0	NaN	2.0	0.0	90.0
8	2020-12-10	1009	12	2.0	8.0	2.0	4.0	60.0
9	2020-10-13	1010	12	6.0	9.0	1.0	0.0	85.0

In [33]:

```
df6.interpolate(method='nearest')
```

Out[33]:

	Date	Student ID	Class	Study hrs	Sleeping hrs	Social Media usage hrs	Mobile Games hrs	Percantege
0	2020-01-10	1001	10	2.0	9.0	3.0	5.0	50.0
1	2020-02-10	1002	10	6.0	8.0	2.0	0.0	40.0
2	2020-03-10	1003	10	6.0	8.0	2.0	4.0	40.0
3	2020-06-10	1004	11	6.0	10.0	1.0	4.0	40.0
4	2020-08-10	1005	11	10.0	7.0	2.0	0.0	80.0
5	2020-09-10	1006	11	10.0	7.0	0.0	0.0	80.0
6	2020-10-10	1007	12	4.0	7.0	0.0	0.0	80.0
7	2020-11-10	1008	12	10.0	8.0	2.0	0.0	90.0
8	2020-12-10	1009	12	2.0	8.0	2.0	4.0	60.0
9	2020-10-13	1010	12	6.0	9.0	1.0	0.0	85.0

In [ ]:

```

```

In [ ]:

```

```

In [34]:

```
# df.interpolate()
# axis : {0 or 'index', 1 or 'columns', None}, default None
df6
```

Out[34]:

	Date	Student ID	Class	Study hrs	Sleeping hrs	Social Media usage hrs	Mobile Games hrs	Percantege
0	2020-01-10	1001	10	2.0	9.0	3.0	5.0	50.0
1	2020-02-10	1002	10	6.0	8.0	2.0	0.0	40.0
2	2020-03-10	1003	10	NaN	8.0	NaN	4.0	NaN
3	2020-06-10	1004	11	NaN	10.0	1.0	NaN	NaN
4	2020-08-10	1005	11	NaN	7.0	2.0	0.0	NaN
5	2020-09-10	1006	11	10.0	7.0	0.0	0.0	NaN
6	2020-10-10	1007	12	4.0	NaN	0.0	0.0	80.0
7	2020-11-10	1008	12	10.0	NaN	2.0	0.0	90.0
8	2020-12-10	1009	12	2.0	8.0	2.0	4.0	60.0
9	2020-10-13	1010	12	6.0	9.0	1.0	0.0	85.0

In [35]:

```
df6.interpolate(limit=1)
```

Out[35]:

	Date	Student ID	Class	Study hrs	Sleeping hrs	Social Media usage hrs	Mobile Games hrs	Percantege
0	2020-01-10	1001	10	2.0	9.000000	3.0	5.0	50.0
1	2020-02-10	1002	10	6.0	8.000000	2.0	0.0	40.0
2	2020-03-10	1003	10	7.0	8.000000	1.5	4.0	48.0
3	2020-06-10	1004	11	NaN	10.000000	1.0	2.0	NaN
4	2020-08-10	1005	11	NaN	7.000000	2.0	0.0	NaN
5	2020-09-10	1006	11	10.0	7.000000	0.0	0.0	NaN
6	2020-10-10	1007	12	4.0	7.333333	0.0	0.0	80.0
7	2020-11-10	1008	12	10.0	NaN	2.0	0.0	90.0
8	2020-12-10	1009	12	2.0	8.000000	2.0	4.0	60.0
9	2020-10-13	1010	12	6.0	9.000000	1.0	0.0	85.0

In [36]:

```
df6.interpolate(limit=2)
```

Out[36]:

	Date	Student ID	Class	Study hrs	Sleeping hrs	Social Media usage hrs	Mobile Games hrs	Percantege
0	2020-01-10	1001	10	2.0	9.000000	3.0	5.0	50.0
1	2020-02-10	1002	10	6.0	8.000000	2.0	0.0	40.0
2	2020-03-10	1003	10	7.0	8.000000	1.5	4.0	48.0
3	2020-06-10	1004	11	8.0	10.000000	1.0	2.0	56.0
4	2020-08-10	1005	11	NaN	7.000000	2.0	0.0	NaN
5	2020-09-10	1006	11	10.0	7.000000	0.0	0.0	NaN
6	2020-10-10	1007	12	4.0	7.333333	0.0	0.0	80.0
7	2020-11-10	1008	12	10.0	7.666667	2.0	0.0	90.0
8	2020-12-10	1009	12	2.0	8.000000	2.0	4.0	60.0
9	2020-10-13	1010	12	6.0	9.000000	1.0	0.0	85.0

In [ ]:

```

```

In [37]:

```
# df6.interpolate()
# limit_direction : {'forward', 'backward', 'both'}, default 'forward'
df6
```

Out[37]:

	Date	Student ID	Class	Study hrs	Sleeping hrs	Social Media usage hrs	Mobile Games hrs	Percantege
0	2020-01-10	1001	10	2.0	9.0	3.0	5.0	50.0
1	2020-02-10	1002	10	6.0	8.0	2.0	0.0	40.0
2	2020-03-10	1003	10	NaN	8.0	NaN	4.0	NaN
3	2020-06-10	1004	11	NaN	10.0	1.0	NaN	NaN
4	2020-08-10	1005	11	NaN	7.0	2.0	0.0	NaN
5	2020-09-10	1006	11	10.0	7.0	0.0	0.0	NaN
6	2020-10-10	1007	12	4.0	NaN	0.0	0.0	80.0
7	2020-11-10	1008	12	10.0	NaN	2.0	0.0	90.0
8	2020-12-10	1009	12	2.0	8.0	2.0	4.0	60.0
9	2020-10-13	1010	12	6.0	9.0	1.0	0.0	85.0

In [38]:

```
df6.interpolate(limit=1,limit_direction='backward')
```

Out[38]:

	Date	Student ID	Class	Study hrs	Sleeping hrs	Social Media usage hrs	Mobile Games hrs	Percantege
0	2020-01-10	1001	10	2.0	9.000000	3.0	5.0	50.0
1	2020-02-10	1002	10	6.0	8.000000	2.0	0.0	40.0
2	2020-03-10	1003	10	NaN	8.000000	1.5	4.0	NaN
3	2020-06-10	1004	11	NaN	10.000000	1.0	2.0	NaN
4	2020-08-10	1005	11	9.0	7.000000	2.0	0.0	NaN
5	2020-09-10	1006	11	10.0	7.000000	0.0	0.0	72.0
6	2020-10-10	1007	12	4.0	NaN	0.0	0.0	80.0
7	2020-11-10	1008	12	10.0	7.666667	2.0	0.0	90.0
8	2020-12-10	1009	12	2.0	8.000000	2.0	4.0	60.0
9	2020-10-13	1010	12	6.0	9.000000	1.0	0.0	85.0

In [39]:

```
df6.interpolate(limit=2,limit_direction='backward')
```

Out[39]:

	Date	Student ID	Class	Study hrs	Sleeping hrs	Social Media usage hrs	Mobile Games hrs	Percantege
0	2020-01-10	1001	10	2.0	9.000000	3.0	5.0	50.0
1	2020-02-10	1002	10	6.0	8.000000	2.0	0.0	40.0
2	2020-03-10	1003	10	NaN	8.000000	1.5	4.0	NaN
3	2020-06-10	1004	11	8.0	10.000000	1.0	2.0	NaN
4	2020-08-10	1005	11	9.0	7.000000	2.0	0.0	64.0
5	2020-09-10	1006	11	10.0	7.000000	0.0	0.0	72.0
6	2020-10-10	1007	12	4.0	7.333333	0.0	0.0	80.0
7	2020-11-10	1008	12	10.0	7.666667	2.0	0.0	90.0
8	2020-12-10	1009	12	2.0	8.000000	2.0	4.0	60.0
9	2020-10-13	1010	12	6.0	9.000000	1.0	0.0	85.0

In [40]:

```
df6.interpolate(limit_direction='both')
```

Out[40]:

	Date	Student ID	Class	Study hrs	Sleeping hrs	Social Media usage hrs	Mobile Games hrs	Percantege
0	2020-01-10	1001	10	2.0	9.000000	3.0	5.0	50.0
1	2020-02-10	1002	10	6.0	8.000000	2.0	0.0	40.0
2	2020-03-10	1003	10	7.0	8.000000	1.5	4.0	48.0
3	2020-06-10	1004	11	8.0	10.000000	1.0	2.0	56.0
4	2020-08-10	1005	11	9.0	7.000000	2.0	0.0	64.0
5	2020-09-10	1006	11	10.0	7.000000	0.0	0.0	72.0
6	2020-10-10	1007	12	4.0	7.333333	0.0	0.0	80.0
7	2020-11-10	1008	12	10.0	7.666667	2.0	0.0	90.0
8	2020-12-10	1009	12	2.0	8.000000	2.0	4.0	60.0
9	2020-10-13	1010	12	6.0	9.000000	1.0	0.0	85.0

In [ ]:

```

```

In [ ]:

```

```

In [41]:

```
#df.interpolate()
'''
limit_area : {`None`, 'inside', 'outside'}, default None
    If limit is specified, consecutive NaNs will be filled with this
    restriction.

    * ``None``: No fill restriction.
    * 'inside': Only fill NaNs surrounded by valid values
      (interpolate).
    * 'outside': Only fill NaNs outside valid values (extrapolate).
'''
df6
```

Out[41]:

	Date	Student ID	Class	Study hrs	Sleeping hrs	Social Media usage hrs	Mobile Games hrs	Percantege
0	2020-01-10	1001	10	2.0	9.0	3.0	5.0	50.0
1	2020-02-10	1002	10	6.0	8.0	2.0	0.0	40.0
2	2020-03-10	1003	10	NaN	8.0	NaN	4.0	NaN
3	2020-06-10	1004	11	NaN	10.0	1.0	NaN	NaN
4	2020-08-10	1005	11	NaN	7.0	2.0	0.0	NaN
5	2020-09-10	1006	11	10.0	7.0	0.0	0.0	NaN
6	2020-10-10	1007	12	4.0	NaN	0.0	0.0	80.0
7	2020-11-10	1008	12	10.0	NaN	2.0	0.0	90.0
8	2020-12-10	1009	12	2.0	8.0	2.0	4.0	60.0
9	2020-10-13	1010	12	6.0	9.0	1.0	0.0	85.0

In [ ]:

```
```

In [43]:

```
df6.interpolate(limit_area='inside')
```

Out[43]:

	Date	Student ID	Class	Study hrs	Sleeping hrs	Social Media usage hrs	Mobile Games hrs	Percantege
0	2020-01-10	1001	10	2.0	9.000000	3.0	5.0	50.0
1	2020-02-10	1002	10	6.0	8.000000	2.0	0.0	40.0
2	2020-03-10	1003	10	7.0	8.000000	1.5	4.0	48.0
3	2020-06-10	1004	11	8.0	10.000000	1.0	2.0	56.0
4	2020-08-10	1005	11	9.0	7.000000	2.0	0.0	64.0
5	2020-09-10	1006	11	10.0	7.000000	0.0	0.0	72.0
6	2020-10-10	1007	12	4.0	7.333333	0.0	0.0	80.0
7	2020-11-10	1008	12	10.0	7.666667	2.0	0.0	90.0
8	2020-12-10	1009	12	2.0	8.000000	2.0	4.0	60.0
9	2020-10-13	1010	12	6.0	9.000000	1.0	0.0	85.0

In [44]:

```
df6.interpolate(limit_area='outside')
```

Out[44]:

	Date	Student ID	Class	Study hrs	Sleeping hrs	Social Media usage hrs	Mobile Games hrs	Percantege
0	2020-01-10	1001	10	2.0	9.0	3.0	5.0	50.0
1	2020-02-10	1002	10	6.0	8.0	2.0	0.0	40.0
2	2020-03-10	1003	10	NaN	8.0	NaN	4.0	NaN
3	2020-06-10	1004	11	NaN	10.0	1.0	NaN	NaN
4	2020-08-10	1005	11	NaN	7.0	2.0	0.0	NaN
5	2020-09-10	1006	11	10.0	7.0	0.0	0.0	NaN
6	2020-10-10	1007	12	4.0	NaN	0.0	0.0	80.0
7	2020-11-10	1008	12	10.0	NaN	2.0	0.0	90.0
8	2020-12-10	1009	12	2.0	8.0	2.0	4.0	60.0
9	2020-10-13	1010	12	6.0	9.0	1.0	0.0	85.0

In [ ]:

```

```