Resample function of Pandas

Use of resample function of pandas in time series data

Resampling is used in time series data. This is a convenience method for frequency conversion and resampling of time series data. Although it works on the condition that objects must have a datetime-like index for example, DatetimeIndex, PeriodIndex, or TimedeltaIndex. In simpler words, if one wants to arrange the time series data in patterns like monthly, weekly, daily, etc., this function is very useful. This function is available in Pandas library. For the demonstration purpose, UCI dataset is used, i.e., https://archive.ics.uci.edu/ml/datasets/Parking+Birmingham.

Reading Data

In time series data, date variables' data type is objects when we read data from a .csv file. Therefore to read the date column in datetime format, we use **parse_dates** argument. In the study data, LastUpdated is the date variable and parse_dates=

["LastUpdated"] argument reading the date format properly, whereas when parse_dates argument doesn't use "LastUpdated" variable type is object.

```
dm=pd.read_csv("data.csv")
```

```
dm.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35717 entries, 0 to 35716
Data columns (total 4 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   SystemCodeNumber  35717 non-null  object
 1   Capacity          35717 non-null  int64 
 2   Occupancy         35717 non-null  int64 
 3   LastUpdated       35717 non-null  object
dtypes: int64(2), object(2)
memory usage: 1.1+ MB
```

```
df=pd.read_csv("data.csv", parse_dates=["LastUpdated"])
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35717 entries, 0 to 35716
Data columns (total 4 columns):
 #   Column            Non-Null Count  Dtype         
---  ------            --------------  -----         
 0   SystemCodeNumber  35717 non-null  object        
 1   Capacity          35717 non-null  int64         
 2   Occupancy         35717 non-null  int64         
 3   LastUpdated       35717 non-null  datetime64[ns]
dtypes: datetime64[ns](1), int64(2), object(1)
memory usage: 1.1+ MB
```

DateTimeIndex

As resample function uses DatetimeIndex, PeriodIndex, or TimedeltaIndex, therefore, now we need to change variable "LastUpdated" into datetimeindex as follows:

```
df2=df.set_index(pd.DatetimeIndex(df1["LastUpdated"])).drop("LastUpdated", axis=1)
```

```
df2.head()
```

LastUpdated	SystemCodeNumber	Capacity	Occupancy
2016-10-04 07:59:00	BHMBCCMKT01	577	61
2016-10-04 08:25:00	BHMBCCMKT01	577	64
2016-10-04 08:59:00	BHMBCCMKT01	577	80
2016-10-04 09:32:00	BHMBCCMKT01	577	107
2016-10-04 09:59:00	BHMBCCMKT01	577	150

Resampling

Resampling is for frequency conversion and resampling of time series. So, if one needs to change the data instead of daily to monthly or weekly etc. or vice versa. For this, we have resample option in pandas library[2]. In the resampling function, if we need to change the date to datetimeindex there is also an option of parameter "on" but the column must be datetime-like.

```
df.resample('W', on='LastUpdated').mean()
```

LastUpdated	Capacity	Occupancy
2016-10-09	1363.275862	546.699234
2016-10-16	1395.311828	612.520908
2016-10-23	1406.956522	597.105878
2016-10-30	1391.326531	628.676871
2016-11-06	1405.492228	600.865285
2016-11-13	1396.000000	609.621368
2016-11-20	1391.530612	627.790533
2016-11-27	1402.783505	678.044674
2016-12-04	1392.357143	713.621825
2016-12-11	1436.475410	712.380996
2016-12-18	1383.288645	696.309159
2016-12-25	1420.153846	844.256410

Below from resampling with option “D”, the data got changed into daily data, i.e., all the dates will be taken into account. 375717 records downsampled to 77 records.

```
df3.resample("D").mean() # daily option
```

LastUpdated	Occupancy
2016-10-04	655.543651
2016-10-05	655.185185
2016-10-06	636.942130
2016-10-07	576.282407
2016-10-08	428.036232
...	...
2016-12-15	736.445110
2016-12-16	675.021073
2016-12-17	726.115385
2016-12-18	613.589583
2016-12-19	844.256410

77 rows × 1 columns

Other Rule Options

The most used options for rule (representing target conversion) are as below and other options can also be found in the reference [1]:

Date Offset	Frequency String	Description
`DateOffset`	None	Generic offset class, defaults to absolute 24 hours
`BDay` or `BusinessDay`	`'B'`	business day (weekday)
`CDay` or `CustomBusinessDay`	`'C'`	custom business day
`Week`	`'W'`	one week, optionally anchored on a day of the week
`WeekOfMonth`	`'WOM'`	the x-th day of the y-th week of each month
`LastWeekOfMonth`	`'LWOM'`	the x-th day of the last week of each month
`MonthEnd`	`'M'`	calendar month end
`MonthBegin`	`'MS'`	calendar month begin
`BusinessHour`	`'BH'`	business hour
`CustomBusinessHour`	`'CBH'`	custom business hour
`Day`	`'D'`	one absolute day
`Hour`	`'H'`	one hour
`Minute`	`'T'` or `'min'`	one minute
`Second`	`'S'`	one second
`Milli`	`'L'` or `'ms'`	one millisecond
`Micro`	`'U'` or `'us'`	one microsecond
`Nano`	`'N'`	one nanosecond

A resample option is used for two options, i.e., upsampling and downsampling.

Upsampling: In this, we resample to the shorter time frame, for example monthly data to weekly/biweekly/daily etc. Because of this, many bins are created with NaN values and to fill these there are different methods that can be used as pad method and bfill method. For example, changing weekly data to daily data and using bfill method following results are obtained, so bfill filling backward the new missing values in the resampled data:

Other method is pad method, it forward fills the values as below:

We can also use asfreq() or fillna() methods in upsamling.

Downsampling: In this we resample to the wider time frame, for example resample daily data to weekly/biweekly/monthly etc. For this we have options like sum(), mean(), max() etc. For example, daily data got resampled to month start data and mean function is used as below:

Graphical representation of Resampling

After resampling data by four different rules, i.e., hourly, daily, weekly, and monthly, following graphs are obtained. We can clearly see the difference in shorter vs wider time frames. In the hourly plot, more noise is there and it is decreasing from daily to weekly to monthly. As per study objective, we can decide which option for rule would be best.

Thanks!

References:

■