

pandas မှ date_range() အကြောင်း

Time Series Analysis တွင် မသိမဖြစ် လုံးဝ အခြေခံကြသည့် date_range()

ယခုလက်ရှိအချိန်(now)ကို ဖတ်ရန်

```
In [1]: import datetime
datetime.datetime.now()
```

```
Out[1]: datetime.datetime(2020, 7, 4, 13, 53, 23, 97455)
```

```
In [2]: delta = datetime.datetime(2020, 5, 30) - datetime.datetime(1938, 6, 3)
delta
```

```
Out[2]: datetime.timedelta(days=29947)
```

```
In [3]: import pandas as pd
import numpy as np
```

ရေးနိုင်သည့် ရက်စွဲပုံစံများ(formats)

```
In [4]: x = datetime.datetime(2020, 5, 17)
x
```

```
Out[4]: datetime.datetime(2020, 5, 17, 0, 0)
```

```
In [5]: '2016 Jul 1', '7/1/2016', '1/7/2016', 'July 1, 2016', '2016-07-01', '2016/07/01'
```

```
Out[5]: ('2016 Jul 1',
'7/1/2016',
'1/7/2016',
'July 1, 2016',
'2016-07-01',
'2016/07/01')
```

```
In [6]: # နာရီ မိနစ် စက္ကန့် ထည့်သွင်း ဖော်ပြသည်။
pd.Timestamp('2020-07-10')
```

```
Out[6]: Timestamp('2020-07-10 00:00:00')
```

```
In [7]: # နာရီ မိနစ် စက္ကန့် ထည့်သွင်း ဖော်ပြသည်။
pd.Timestamp('2016-07-10 10')
```

```
Out[7]: Timestamp('2016-07-10 10:00:00')
```

```
In [8]: # နာရီ မိနစ် စက္ကန့် ထည့်သွင်း ဖော်ပြသည်။
pd.Timestamp('2016-07-10 10:15')
```

```
Out[8]: Timestamp('2016-07-10 10:15:00')
```

properties of timestamps

hint: <http://pandas.pydata.org/pandas-docs/stable/timeseries.html#time-date-components> (<http://pandas.pydata.org/pandas-docs/stable/timeseries.html#time-date-components>)

```
In [9]: t = pd.Timestamp('2016-07-10 10:15')
```

```
In [10]: pd.Period('2016-01') # လကိုသာ ဆိုလိုသည်။
```

```
Out[10]: Period('2016-01', 'M')
```

```
In [11]: pd.Period('2016-01-02') # နေ့အထိ ဖော်ပြသည်။
```

```
Out[11]: Period('2016-01-02', 'D')
```

```
In [12]: pd.Period('2016-01-01') # နေ့အထိ ဖော်ပြသည်။
```

```
Out[12]: Period('2016-01-01', 'D')
```

```
In [13]: pd.Period('2016-01-01 10') # နာရီ အထိ ဖော်ပြသည်။
```

```
Out[13]: Period('2016-01-01 10:00', 'H')
```

```
In [14]: pd.Period('2016-01-01 10:10') # မိနစ်အထိ ဖော်ပြသည်။
```

```
Out[14]: Period('2016-01-01 10:10', 'T')
```

```
In [15]: pd.Period('2016-01-01 10:10:10') # စက္ကန့်အထိ ဖော်ပြသည်။
```

```
Out[15]: Period('2016-01-01 10:10:10', 'S')
```

```
In [16]: # TIME OFFSETS
pd.Timedelta('1 day') # ၁ရက်
```

```
Out[16]: Timedelta('1 days 00:00:00')
```

```
In [17]: pd.Period('2016-01-01 10:10') + pd.Timedelta('1 day') # '2016-01-01 10:10' မှ ၁ရက်တိတိကြာပြီးချိန်
```

```
Out[17]: Period('2016-01-02 10:10', 'T')
```

```
In [18]: pd.Timestamp('2016-01-01 10:10') + pd.Timedelta('1 day') # '2016-01-01 10:10' မှ ၁ရက်တိတိကြာပြီးချိန်
```

```
Out[18]: Timestamp('2016-01-02 10:10:00')
```

```
In [19]: pd.Timestamp('2016-01-01 10:10') + pd.Timedelta('15 ns') # '2016-01-01 10:10' မှ 15 နာနီ စက္ကန့်အကြာ
```

```
Out[19]: Timestamp('2016-01-01 10:10:00.000000015')
```

```
In [20]: #Please ignore below code
from IPython.core.interactiveshell import InteractiveShell # notebook setting အတွက်
InteractiveShell.ast_node_interactivity = "all" # notebook setting အတွက်
```

pandas မှ date_range()ကို အသုံးပြု၍ ရက်စွဲများ၊ နေ့ရက်များ၊ အချိန်များကို လိုက်ဖြစ်အောင် ပြင်ဆင်နိုင်သည်။ pandas မှ date_range() က အောက်ပါ အတိုင်းဖြစ်သည်။

```
date_range(
    start=None,
    end=None,
    periods=None,
    freq=None,
    tz=None,
    normalize=False,
    name=None,
    closed=None,
    **kwargs,
)
```

၂၀၂၀ ခုနှစ် ဇူလိုင်လ ၁ နေ့(2020 Jul 1) နောက်ထပ် (၁၀)ရက်ကို ရက်အလိုက်ဖြစ်အောင် ပြင်ဆင်ရန်

စမည့်ရက်(start='2020 Jul 1')ထည့်ပေးရသည်။

ကြာမည့်အချိန်ကာလ(periods = 10)ထည့်ပေးရသည်။ စုစုပေါင်း ၁၀ ရက်

ပိုင်းခြားရမည့် ပမာဏ၊ စက္ကန့်တိုင်း၊ နာရီတိုင်း၊ နေ့စဉ်၊ လစဉ်၊ နှစ်စဉ်၊ ရှိဖွင့်ရက်တိုင်း စသည်ဖြင့် ထည့်ပေးရသည်။

freq = 'D' သည် frequency = daliy ဖြစ်သည်။ D= daliy

```
In [21]: rng = pd.date_range('2020 Jul 1', periods = 10, freq = 'D')
rng
```

```
Out[21]: DatetimeIndex(['2020-07-01', '2020-07-02', '2020-07-03', '2020-07-04',
                        '2020-07-05', '2020-07-06', '2020-07-07', '2020-07-08',
                        '2020-07-09', '2020-07-10'],
                        dtype='datetime64[ns]', freq='D')
```

၂၀၂၀ ခုနှစ် ဇူလိုင်လ ၁ နေ့(2020 Jul 1) နောက်ထပ် (၁၂)လကို လလိုက်ဖြစ်အောင် ပြင်ဆင်ရန်

စမည့်ရက်(start='2020 Jul 1')ထည့်ပေးရသည်။

ကြာမည့်အချိန်ကာလ(periods = 12)ထည့်ပေးရသည်။ စုစုပေါင်း ၁၂ လ

ပိုင်းခြားရမည့် ပမာဏ၊ စက္ကန့်တိုင်း၊ နာရီတိုင်း၊ နေ့စဉ်၊ လစဉ်၊ နှစ်စဉ်၊ ရှိဖွင့်ရက်တိုင်း စသည်ဖြင့် ထည့်ပေးရသည်။

freq = 'M' သည် frequency = daliy ဖြစ်သည်။ D= daliy

```
In [22]: m_range = pd.date_range('2020 Jul 1', periods = 12, freq = 'M')
m_range
```

```
Out[22]: DatetimeIndex(['2020-07-31', '2020-08-31', '2020-09-30', '2020-10-31',
                        '2020-11-30', '2020-12-31', '2021-01-31', '2021-02-28',
                        '2021-03-31', '2021-04-30', '2021-05-31', '2021-06-30'],
                        dtype='datetime64[ns]', freq='M')
```

စမည့်ရက် နှင့် နောက်ဆုံးရက် ထည့်ပေးလျှင်လည်းရသည်။ စမည့်ရက် နှင့် နောက်ဆုံးရက် ထည့်ပေးလျှင် freq = 'D' သည် daily frequency သည် default ဖြစ်သည်။ D= daily ဟု ဖော်ပြပေးသည်။

```
In [23]: pd.date_range(start='1/1/2018', end='1/08/2018')
```

```
Out[23]: DatetimeIndex(['2018-01-01', '2018-01-02', '2018-01-03', '2018-01-04',
                        '2018-01-05', '2018-01-06', '2018-01-07', '2018-01-08'],
                        dtype='datetime64[ns]', freq='D')
```

စမည့်ရက်(start='1/1/2018') နှင့် ကြာမည့်အချိန်ကာလ(periods = 8)ထည့်ပေးရသည်။ စုစုပေါင်း 8 ရက်

```
In [24]: pd.date_range(start='1/1/2018', periods=8)
```

```
Out[24]: DatetimeIndex(['2018-01-01', '2018-01-02', '2018-01-03', '2018-01-04',
                        '2018-01-05', '2018-01-06', '2018-01-07', '2018-01-08'],
                        dtype='datetime64[ns]', freq='D')
```

နောက်ဆုံးရက်(end='2020 Jul 1')နှင့် ကြာမည့်အချိန်ကာလ(periods = 8)ထည့်ပေးရသည်။ စုစုပေါင်း 8 ရက်

```
In [25]: pd.date_range(end='2020 Jul 1', periods=8)
```

```
Out[25]: DatetimeIndex(['2020-06-24', '2020-06-25', '2020-06-26', '2020-06-27',
                        '2020-06-28', '2020-06-29', '2020-06-30', '2020-07-01'],
                        dtype='datetime64[ns]', freq='D')
```

'2018-04-24'မှ စ၍ (၁) ရက်ခြားစီ ဆေးခန်း (၃)ကြိမ် ပြုရမည်။

```
In [26]: pd.date_range(start='2018-04-24', periods=3, freq = '2D')
```

```
Out[26]: DatetimeIndex(['2018-04-24', '2018-04-26', '2018-04-28'], dtype='datetime64[ns]', freq='2D')
```

Multiples are allowed

ရက်တွေ လတွေ နှစ်တွေကို မြောက်ဖော်ကိန်းဖြင့် ရေးနိုင်သည်။

3M သည် ၃လ ဖြစ်သည်။ 7D သည် တစ်ပတ်ဖြစ်သည်။

1/7/2020 နေ့မှ စ၍ လကုန်ရက်တိုင်း (၃)လစာ ပင်စင်ထုတ်ရမည်။ တစ်နှစ်စာ ဖြစ်သည်။

```
In [27]: pd.date_range(start='1/7/2020', periods=4, freq='3M') # 3M = ၃လ
```

```
Out[27]: DatetimeIndex(['2020-01-31', '2020-04-30', '2020-07-31', '2020-10-31'],
                        dtype='datetime64[ns]', freq='3M')
```

```
In [28]: pd.date_range(start='1/1/2018', periods=5, freq='3M')
```

```
Out[28]: DatetimeIndex(['2018-01-31', '2018-04-30', '2018-07-31', '2018-10-31',
                        '2019-01-31'],
                        dtype='datetime64[ns]', freq='3M')
```

frequency ကိုလည်း Offset လုပ်နိုင်သည်။

freq=pd.offsets.MonthEnd(3)) သည် ၃လ တစ်ခါဖြစ် သည်။

```
In [29]: pd.date_range(start='1/1/2018', periods=5, freq=pd.offsets.MonthEnd(3))
```

```
Out[29]: DatetimeIndex(['2018-01-31', '2018-04-30', '2018-07-31', '2018-10-31',
                        '2019-01-31'],
                        dtype='datetime64[ns]', freq='3M')
```

B သည် business day frequency ဖြစ်သည်။ ရုံးဖွင့်ရက်တိုင်း (စနေ တနင်္ဂနွေ မပါ)
freq=pd.offsets.BDay())

```
In [30]: pd.date_range(start='1/1/2018', end='1/10/2018', freq=pd.offsets.BDay())
```

```
Out[30]: DatetimeIndex(['2018-01-01', '2018-01-02', '2018-01-03', '2018-01-04',
                        '2018-01-05', '2018-01-08', '2018-01-09', '2018-01-10'],
                        dtype='datetime64[ns]', freq='B')
```

မိမိ လိုချင်သည့် timezone ကို လည်း tz ဖြင့် ရေးနိုင်(Specify လုပ်)သည်။

```
In [31]: pd.date_range(start='1/1/2018', periods=5, tz='Asia/Singapore')
```

```
Out[31]: DatetimeIndex(['2018-01-01 00:00:00+08:00', '2018-01-02 00:00:00+08:00',
                        '2018-01-03 00:00:00+08:00', '2018-01-04 00:00:00+08:00',
                        '2018-01-05 00:00:00+08:00'],
                        dtype='datetime64[ns, Asia/Singapore]', freq='D')
```

closed ဖြင့် အစဆုံးရက် နှင့် နောက်ဆုံးရက် ပါ မပါ control လုပ်နိုင်သည်။ ဘာမှ မပြောထားလျှင် (default) အစဆုံးရက် နှင့် နောက်ဆုံးရက်ကို ထည့်ထားသည်။ (The default includes boundary points on either end.)

```
In [32]: pd.date_range(start='2017-01-01', end='2017-01-04', closed=None)
```

```
Out[32]: DatetimeIndex(['2017-01-01', '2017-01-02', '2017-01-03', '2017-01-04'],
                        dtype='datetime64[ns]', freq='D')
```

closed='left' သည် နောက်ဆုံးရက်ကို ချန်ထားသည်။ ထည့်မ တွက်ပါ။ (to exclude end if it falls on the boundary.)

```
In [33]: pd.date_range(start='2017-01-01', end='2017-01-04', closed='left')
```

```
Out[33]: DatetimeIndex(['2017-01-01', '2017-01-02', '2017-01-03'], dtype='datetime64[ns]', freq='D')
```

closed='right' သည် အစဆုံးရက်ကို ချန်ထားသည်။ ထည့်မ တွက်ပါ။ (to exclude start if it falls on the boundary.)

```
In [34]: pd.date_range(start='2017-01-01', end='2017-01-04', closed='right')
```

```
Out[34]: DatetimeIndex(['2017-01-02', '2017-01-03', '2017-01-04'], dtype='datetime64[ns]', freq='D')
```

```
In [35]: # ရုံးဖွင့်ရက်များသာ (Only want business days)
pd.period_range('2016-01-01 10:10', freq = 'B', periods = 10)
```

```
Out[35]: PeriodIndex(['2016-01-01', '2016-01-04', '2016-01-05', '2016-01-06',
'2016-01-07', '2016-01-08', '2016-01-11', '2016-01-12',
'2016-01-13', '2016-01-14'],
dtype='period[B]', freq='B')
```

frequency များကို ပေါင်းရေးနိုင်သည်။ ဥပမာ ၂၅ နာရီကို ၁ ရက်နှင့် ၁နာရီ ဟု ရေးနိုင်သည်။ What if you want to advance by 25 hours each day. freq = '25H' သည် ၂၅ နာရီ

```
In [36]: p1 = pd.period_range('2016-01-01 10:10', freq = '25H', periods = 10)
p1
```

```
Out[36]: PeriodIndex(['2016-01-01 10:00', '2016-01-02 11:00', '2016-01-03 12:00',
'2016-01-04 13:00', '2016-01-05 14:00', '2016-01-06 15:00',
'2016-01-07 16:00', '2016-01-08 17:00', '2016-01-09 18:00',
'2016-01-10 19:00'],
dtype='period[25H]', freq='25H')
```

```
In [37]: p2 = pd.period_range('2016-01-01 10:10', freq = '1D1H', periods = 10)
p2
```

```
Out[37]: PeriodIndex(['2016-01-01 10:00', '2016-01-02 11:00', '2016-01-03 12:00',
'2016-01-04 13:00', '2016-01-05 14:00', '2016-01-06 15:00',
'2016-01-07 16:00', '2016-01-08 17:00', '2016-01-09 18:00',
'2016-01-10 19:00'],
dtype='period[25H]', freq='25H')
```

<http://pandas.pydata.org/pandas-docs/stable/timeseries.html#offset-aliases>
[\(http://pandas.pydata.org/pandas-docs/stable/timeseries.html#offset-aliases\)](http://pandas.pydata.org/pandas-docs/stable/timeseries.html#offset-aliases) Time objects ကိုလည်း Indexing လုပ်နိုင်သည်။ date range ကို သုံးထားသည်။

```
In [38]: rng = pd.date_range('2020 Jul 1', periods = 10, freq = 'D')
         rng
         pd.Series(range(len(rng)), index = rng)
```

```
Out[38]: DatetimeIndex(['2020-07-01', '2020-07-02', '2020-07-03', '2020-07-04',
                        '2020-07-05', '2020-07-06', '2020-07-07', '2020-07-08',
                        '2020-07-09', '2020-07-10'],
                        dtype='datetime64[ns]', freq='D')
```

```
Out[38]: 2020-07-01    0
         2020-07-02    1
         2020-07-03    2
         2020-07-04    3
         2020-07-05    4
         2020-07-06    5
         2020-07-07    6
         2020-07-08    7
         2020-07-09    8
         2020-07-10    9
         Freq: D, dtype: int64
```

'07-10-16 8:00' မှ စ၍ (၁)နာရီခြား (၁၀) ကြိမ်

```
In [39]: ts = pd.Series(range(10), pd.date_range('07-10-16 8:00', periods = 10, freq = 'H'))
         ts
```

```
Out[39]: 2016-07-10 08:00:00    0
         2016-07-10 09:00:00    1
         2016-07-10 10:00:00    2
         2016-07-10 11:00:00    3
         2016-07-10 12:00:00    4
         2016-07-10 13:00:00    5
         2016-07-10 14:00:00    6
         2016-07-10 15:00:00    7
         2016-07-10 16:00:00    8
         2016-07-10 17:00:00    9
         Freq: H, dtype: int64
```

```
In [40]: ts_period = ts.to_period()
         ts_period
```

```
Out[40]: 2016-07-10 08:00    0
         2016-07-10 09:00    1
         2016-07-10 10:00    2
         2016-07-10 11:00    3
         2016-07-10 12:00    4
         2016-07-10 13:00    5
         2016-07-10 14:00    6
         2016-07-10 15:00    7
         2016-07-10 16:00    8
         2016-07-10 17:00    9
         Freq: H, dtype: int64
```

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.date_range.html
[\(https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.date_range.html\)](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.date_range.html)

Offset aliases

A number of string aliases are given to useful common time series frequencies. We will refer to these aliases as offset aliases.

Alias Description

B = business day frequency

C = custom business day frequency

D = calendar day frequency

W =weekly frequency

M = month end frequency

SM =semi-month end frequency (15th and end of month)

BM =business month end frequency

CBM= custom business month end frequency

MS= month start frequency

SMS= semi-month start frequency (1st and 15th)

BMS= business month start frequency

CBMS= custom business month start frequency

Q = quarter end frequency

BQ = business quarter end frequency

QS = quarter start frequency

BQS =business quarter start frequency

A, Y = year end frequency

BA, BY = business year end frequency

AS, YS = year start frequency

BAS, BYS = business year start frequency

BH= business hour frequency

H =hourly frequency

T, min = minutely frequency

S = secondly frequency

L, ms =milliseconds

U, us = microseconds

N = nanoseconds

In [41]: `import datetime`

```

dti1 = pd.to_datetime(['1/1/2018'])
dti2 = pd.to_datetime([np.datetime64('2018-01-01')])
dti3 = pd.to_datetime([datetime.datetime(2018, 1, 1)])

dti = pd.to_datetime(['1/1/2018', np.datetime64('2018-01-01'),
                      datetime.datetime(2018, 1, 1)])

dti1
dti2
dti3
dti

```

Out[41]: DatetimeIndex(['2018-01-01'], dtype='datetime64[ns]', freq=None)

Out[41]: DatetimeIndex(['2018-01-01'], dtype='datetime64[ns]', freq=None)

Out[41]: DatetimeIndex(['2018-01-01'], dtype='datetime64[ns]', freq=None)

Out[41]: DatetimeIndex(['2018-01-01', '2018-01-01', '2018-01-01'], dtype='datetime64[ns]', freq=None)

In [42]: `import calendar as cal`

```

pd.date_range(start='7/1/2012', end='7/10/2012',
              freq=pd.offsets.CDay(calendar=cal)).to_pydatetime()

```

```

Out[42]: array([datetime.datetime(2012, 7, 2, 0, 0),
               datetime.datetime(2012, 7, 3, 0, 0),
               datetime.datetime(2012, 7, 4, 0, 0),
               datetime.datetime(2012, 7, 5, 0, 0),
               datetime.datetime(2012, 7, 6, 0, 0),
               datetime.datetime(2012, 7, 9, 0, 0),
               datetime.datetime(2012, 7, 10, 0, 0)], dtype=object)

```

In []: