# Time Series - Lesson 1

In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
np.random.seed(12345)
np.set_printoptions(precision=4, suppress=True)
```

## Date and Time Data Types and Tools

In [3]:
```
from datetime import datetime, date

now = datetime.now()
print(now)

print(now.year, now.month, now.day)

print(date.today())
```

```
2019-11-03 23:30:35.324062
2019 11 3
2019-11-03
```

- datetime(year, month, day, hour=0, minute=0, second=0, microsecond=0)

In [4]:
```
datetime(2019, 1, 31, 8, 30)
```

Out[4]:
```
datetime.datetime(2019, 1, 31, 8, 30)
```

In [5]:
```
delta = datetime(2019, 1, 31, 8, 30) - datetime(2019, 1, 1, 8, 20)
print(delta.days, delta.seconds)
delta
```

```
30 600
```

Out[5]:
```
datetime.timedelta(days=30, seconds=600)
```

- timedelta(days=0, seconds=0, microseconds=0, milliseconds=0, minutes=0, hours=0, weeks=0)

In [6]:
```
from datetime import timedelta
```

In [7]:
```
timedelta(12, 30, 10)
```

Out[7]:
```
datetime.timedelta(days=12, seconds=30, microseconds=10)
```

In [8]:
```
start = datetime(2019, 1, 1)
start + timedelta(12, 30),  start + 2 * timedelta(12)
```

Out[8]:
```
(datetime.datetime(2019, 1, 13, 0, 0, 30),
 datetime.datetime(2019, 1, 25, 0, 0))
```

### Converting Between String and Datetime

In [9]:
```
# format

stamp = datetime(2019, 1, 15)

print(str(stamp))
print(stamp.strftime('%Y-%m-%d'))
print(stamp.strftime('%y-%m-%d'))
print(stamp.strftime('%A, %B %d, %Y'))
print(stamp.strftime('%a, %B %d, %Y'))
```

```
2019-01-15 00:00:00
2019-01-15
19-01-15
Tuesday, January 15, 2019
Tue, January 15, 2019
```

In [10]:
```
# parse

value = '2019-01-15'
datetime.strptime(value, '%Y-%m-%d')
```

Out[10]:
```
datetime.datetime(2019, 1, 15, 0, 0)
```

In [11]:
```
datestrs = ['1/15/2019', '1/30/2019']
[datetime.strptime(x, '%m/%d/%Y') for x in datestrs]
```

Out[11]:
```
[datetime.datetime(2019, 1, 15, 0, 0), datetime.datetime(2019, 1, 30, 0, 0)]
```

In [12]:
```
value = 'Tuesday, January 15, 2019'
datetime.strptime(value, '%A, %B %d, %Y')
```

Out[12]:
```
datetime.datetime(2019, 1, 15, 0, 0)
```

- dateutil.parser module offers a generic date/time string parser which is able to parse most known formats to represent a date and/or time

In [13]:
```
from dateutil.parser import parse
parse('2019-01-31')
```

Out[13]:
```
datetime.datetime(2019, 1, 31, 0, 0)
```

In [14]:
```
parse('Jan 31, 2019 10:45 AM')
```

Out[14]:
```
datetime.datetime(2019, 1, 31, 10, 45)
```

In [15]:
```
parse('31/1/2019', dayfirst=True)
```

Out[15]:
```
datetime.datetime(2019, 1, 31, 0, 0)
```

### pandas to_datetime method

In [16]:
```
datestrs = ['2019-01-06 12:00:00', '2019-02-06 00:00:00']
pd.to_datetime(datestrs)
```

Out[16]:
```
DatetimeIndex(['2019-01-06 12:00:00', '2019-02-06 00:00:00'], dtype='datetime64[ns]', freq=None)
```

In [ ]:
```

```

## Time Series Operations

In [17]:
```
np.random.seed(12345)

dates = [datetime(2018, 12, 31),
         datetime(2019, 1, 1), datetime(2019, 1, 2),
         datetime(2019, 1, 3), datetime(2019, 1, 4),
         datetime(2019, 1, 7), datetime(2019, 1, 8)]

ts = pd.Series(np.random.randint(1, 100, 7), index=dates)
ts
```

Out[17]:
```
2018-12-31    99
2019-01-01    30
2019-01-02     2
2019-01-03    37
2019-01-04    42
2019-01-07    35
2019-01-08    30
dtype: int64
```

In [18]:
```
ts.index
```

Out[18]:
```
DatetimeIndex(['2018-12-31', '2019-01-01', '2019-01-02', '2019-01-03',
               '2019-01-04', '2019-01-07', '2019-01-08'],
              dtype='datetime64[ns]', freq=None)
```

In [19]:
```
ts[::2]
```

Out[19]:
```
2018-12-31    99
2019-01-02     2
2019-01-04    42
2019-01-08    30
dtype: int64
```

In [20]:
```
ts + ts[::2]
```

Out[20]:
```
2018-12-31    198.0
2019-01-01      NaN
2019-01-02      4.0
2019-01-03      NaN
2019-01-04     84.0
2019-01-07      NaN
2019-01-08     60.0
dtype: float64
```

In [ ]:

In [21]:
```
stamp = ts.index[0]
stamp
```

Out[21]:
```
Timestamp('2018-12-31 00:00:00')
```

## Indexing, Selection, Subsetting

In [22]:
```
ts
```

Out[22]:
```
2018-12-31    99
2019-01-01    30
2019-01-02     2
2019-01-03    37
2019-01-04    42
2019-01-07    35
2019-01-08    30
dtype: int64
```

In [23]:
```
stamp = ts.index[1]
ts[stamp]
```

Out[23]:
```
30
```

In [24]:
```
ts['1/7/2019'], ts['20190107']
```

Out[24]:
```
(35, 35)
```

In [25]:
```
ts['2018']
```

Out[25]:
```
2018-12-31    99
dtype: int64
```

In [26]:
```
ts['2019']
```

Out[26]:
```
2019-01-01    30
2019-01-02     2
2019-01-03    37
2019-01-04    42
2019-01-07    35
2019-01-08    30
dtype: int64
```

In [27]:
```
np.random.seed(12345)

longer_ts = pd.Series(np.random.randint(1, 100, 1000),
                      index=pd.date_range('1/1/2018', periods=1000))
longer_ts.head()
```

Out[27]:
```
2018-01-01    99
2018-01-02    30
2018-01-03     2
2018-01-04    37
2018-01-05    42
Freq: D, dtype: int64
```

In [28]:
```
longer_ts.tail()
```

Out[28]:
```
2020-09-22    37
2020-09-23    24
2020-09-24    89
2020-09-25    78
2020-09-26    44
Freq: D, dtype: int64
```

In [29]:
```
longer_ts['2019']
```

Out[29]:
```
2019-01-01    64
2019-01-02    96
2019-01-03    55
2019-01-04    90
2019-01-05    51
              ..
2019-12-27    38
2019-12-28    37
2019-12-29    96
2019-12-30    36
2019-12-31     7
Freq: D, Length: 365, dtype: int64
```

In [30]:
```
longer_ts['2019-09']
```

Out[30]:
```
2019-09-01    22
2019-09-02    78
2019-09-03    49
2019-09-04     4
2019-09-05    60
2019-09-06    77
2019-09-07    86
2019-09-08    21
2019-09-09    56
2019-09-10    23
2019-09-11    86
2019-09-12     7
2019-09-13    12
2019-09-14    87
2019-09-15    63
2019-09-16    79
2019-09-17    49
2019-09-18    63
2019-09-19    45
2019-09-20    34
2019-09-21    86
2019-09-22     2
2019-09-23    23
2019-09-24    21
2019-09-25    87
2019-09-26    61
2019-09-27    91
2019-09-28    26
2019-09-29    14
2019-09-30    32
Freq: D, dtype: int64
```

In [31]:
```
ts
```

Out[31]:
```
2018-12-31    99
2019-01-01    30
2019-01-02     2
2019-01-03    37
2019-01-04    42
2019-01-07    35
2019-01-08    30
dtype: int64
```

In [32]:
```
ts[datetime(2019, 1, 3):]
```

Out[32]:
```
2019-01-03    37
2019-01-04    42
2019-01-07    35
2019-01-08    30
dtype: int64
```

In [33]:
```
ts['1/3/2019':'1/10/2019']
```

Out[33]:
```
2019-01-03    37
2019-01-04    42
2019-01-07    35
2019-01-08    30
dtype: int64
```

In [34]:
```
ts['1/3/2019':'1/7/2019']
```

Out[34]:
```
2019-01-03    37
2019-01-04    42
2019-01-07    35
dtype: int64
```

In [35]:
```
ts.truncate(after='1/3/2019')
```

Out[35]:
```
2018-12-31    99
2019-01-01    30
2019-01-02     2
2019-01-03    37
dtype: int64
```

In [36]:
```
ts
```

Out[36]:
```
2018-12-31    99
2019-01-01    30
2019-01-02     2
2019-01-03    37
2019-01-04    42
2019-01-07    35
2019-01-08    30
dtype: int64
```

**Frequency offsets**

https://pandas.pydata.org/pandas-docs/stable/user_guide/timeseries.html#offset-aliases (https://pandas.pydata.org/pandas-docs/stable/user_guide/timeseries.html#offset-aliases)

In [37]:
```
# Dataframe

dates = pd.date_range('1/1/2019', periods=100, freq='W-Tue')
dates[:20]
```

Out[37]:
```
DatetimeIndex(['2019-01-01', '2019-01-08', '2019-01-15', '2019-01-22',
               '2019-01-29', '2019-02-05', '2019-02-12', '2019-02-19',
               '2019-02-26', '2019-03-05', '2019-03-12', '2019-03-19',
               '2019-03-26', '2019-04-02', '2019-04-09', '2019-04-16',
               '2019-04-23', '2019-04-30', '2019-05-07', '2019-05-14'],
              dtype='datetime64[ns]', freq='W-TUE')
```

In [38]:
```
np.random.seed(12345)

long_df = pd.DataFrame(np.random.randint(1, 100, (100, 4)),
                       index=dates,
                       columns=['Colorado', 'Texas',
                                'New York', 'Ohio'])
long_df.head()
```

Out[38]:

| | Colorado | Texas | New York | Ohio |
|---|---|---|---|---|
| 2019-01-01 | 99 | 30 | 2 | 37 |
| 2019-01-08 | 42 | 35 | 30 | 2 |
| 2019-01-15 | 60 | 15 | 92 | 81 |
| 2019-01-22 | 74 | 12 | 78 | 11 |
| 2019-01-29 | 82 | 83 | 39 | 8 |

In [39]:
```
long_df.loc['3-2019']
```

Out[39]:

| | Colorado | Texas | New York | Ohio |
|---|---|---|---|---|
| 2019-03-05 | 15 | 82 | 6 | 35 |
| 2019-03-12 | 54 | 73 | 6 | 99 |
| 2019-03-19 | 6 | 60 | 84 | 13 |
| 2019-03-26 | 15 | 81 | 28 | 39 |

In [40]:
```
long_df.loc['3-2019':'5-2019']
```

Out[40]:

| | Colorado | Texas | New York | Ohio |
|---|---|---|---|---|
| 2019-03-05 | 15 | 82 | 6 | 35 |
| 2019-03-12 | 54 | 73 | 6 | 99 |
| 2019-03-19 | 6 | 60 | 84 | 13 |
| 2019-03-26 | 15 | 81 | 28 | 39 |
| 2019-04-02 | 89 | 97 | 6 | 75 |
| 2019-04-09 | 77 | 23 | 13 | 63 |
| 2019-04-16 | 42 | 19 | 51 | 83 |
| 2019-04-23 | 58 | 47 | 59 | 24 |
| 2019-04-30 | 22 | 8 | 82 | 64 |
| 2019-05-07 | 1 | 91 | 45 | 58 |
| 2019-05-14 | 15 | 27 | 44 | 31 |
| 2019-05-21 | 33 | 4 | 65 | 7 |
| 2019-05-28 | 67 | 50 | 38 | 73 |

In [ ]:
```

```

In [41]:
```
# Weekly default to Sundays

np.random.seed(12345)

dates = pd.date_range('1/1/2019', periods=100, freq='W')
long_df = pd.DataFrame(np.random.randint(1, 100, (100, 4)),
                       index=dates,
                       columns=['Colorado', 'Texas',
                                'New York', 'Ohio'])

long_df.loc['11-2019']
```

Out[41]:

| | Colorado | Texas | New York | Ohio |
|---|---|---|---|---|
| 2019-11-03 | 94 | 2 | 26 | 65 |
| 2019-11-10 | 60 | 27 | 57 | 13 |
| 2019-11-17 | 91 | 24 | 61 | 96 |
| 2019-11-24 | 78 | 70 | 49 | 65 |

## Date Ranges, Frequencies, and Shifting

In [42]:
```
ts
```

Out[42]:
```
2018-12-31    99
2019-01-01    30
2019-01-02     2
2019-01-03    37
2019-01-04    42
2019-01-07    35
2019-01-08    30
dtype: int64
```

In [43]:
```
# Resample at higher or lower frequencies

resampler = ts.resample('D')
resampler
```

Out[43]:
```
<pandas.core.resample.DatetimeIndexResampler object at 0x11a1d4160>
```

In [44]:
```
resampler.sum()
```

Out[44]:
```
2018-12-31    99
2019-01-01    30
2019-01-02     2
2019-01-03    37
2019-01-04    42
2019-01-05     0
2019-01-06     0
2019-01-07    35
2019-01-08    30
Freq: D, dtype: int64
```

In [45]:
```
resampler = ts.resample('3D')
resampler.sum()
```

Out[45]:
```
2018-12-31    131
2019-01-03     79
2019-01-06     65
Freq: 3D, dtype: int64
```

In [46]:
```
resampler = ts.resample('M')
resampler.sum()
```

Out[46]:
```
2018-12-31     99
2019-01-31    176
Freq: M, dtype: int64
```

## Generating Date Ranges

- pandas.date_range(start=None, end=None, periods=None, freq='D', tz=None, normalize=False, name=None, closed=None)

In [47]:
```
# default frequency - Daily

index = pd.date_range('2019-11-01', '2019-12-01')
index
```

Out[47]:
```
DatetimeIndex(['2019-11-01', '2019-11-02', '2019-11-03', '2019-11-04',
               '2019-11-05', '2019-11-06', '2019-11-07', '2019-11-08',
               '2019-11-09', '2019-11-10', '2019-11-11', '2019-11-12',
               '2019-11-13', '2019-11-14', '2019-11-15', '2019-11-16',
               '2019-11-17', '2019-11-18', '2019-11-19', '2019-11-20',
               '2019-11-21', '2019-11-22', '2019-11-23', '2019-11-24',
               '2019-11-25', '2019-11-26', '2019-11-27', '2019-11-28',
               '2019-11-29', '2019-11-30', '2019-12-01'],
              dtype='datetime64[ns]', freq='D')
```

In [48]:
```
# Business days only

pd.date_range('2019-11-01', '2019-12-01', freq='B')
```

Out[48]:
```
DatetimeIndex(['2019-11-01', '2019-11-04', '2019-11-05', '2019-11-06',
               '2019-11-07', '2019-11-08', '2019-11-11', '2019-11-12',
               '2019-11-13', '2019-11-14', '2019-11-15', '2019-11-18',
               '2019-11-19', '2019-11-20', '2019-11-21', '2019-11-22',
               '2019-11-25', '2019-11-26', '2019-11-27', '2019-11-28',
               '2019-11-29'],
              dtype='datetime64[ns]', freq='B')
```

In [49]:
```
pd.date_range(start='2019-11-01', periods=20)
```

Out[49]:
```
DatetimeIndex(['2019-11-01', '2019-11-02', '2019-11-03', '2019-11-04',
               '2019-11-05', '2019-11-06', '2019-11-07', '2019-11-08',
               '2019-11-09', '2019-11-10', '2019-11-11', '2019-11-12',
               '2019-11-13', '2019-11-14', '2019-11-15', '2019-11-16',
               '2019-11-17', '2019-11-18', '2019-11-19', '2019-11-20'],
              dtype='datetime64[ns]', freq='D')
```

In [50]:
```
pd.date_range(start='2019-11-01', periods=20, freq='B')
```

Out[50]:
```
DatetimeIndex(['2019-11-01', '2019-11-04', '2019-11-05', '2019-11-06',
               '2019-11-07', '2019-11-08', '2019-11-11', '2019-11-12',
               '2019-11-13', '2019-11-14', '2019-11-15', '2019-11-18',
               '2019-11-19', '2019-11-20', '2019-11-21', '2019-11-22',
               '2019-11-25', '2019-11-26', '2019-11-27', '2019-11-28'],
              dtype='datetime64[ns]', freq='B')
```

In [51]:
```
pd.date_range(end='2019-11-01', periods=20)
```

Out[51]:
```
DatetimeIndex(['2019-10-13', '2019-10-14', '2019-10-15', '2019-10-16',
               '2019-10-17', '2019-10-18', '2019-10-19', '2019-10-20',
               '2019-10-21', '2019-10-22', '2019-10-23', '2019-10-24',
               '2019-10-25', '2019-10-26', '2019-10-27', '2019-10-28',
               '2019-10-29', '2019-10-30', '2019-10-31', '2019-11-01'],
              dtype='datetime64[ns]', freq='D')
```

In [52]:
```
# Month end frequency

pd.date_range('2019-01-01', freq='M', periods=12)
```

Out[52]:
```
DatetimeIndex(['2019-01-31', '2019-02-28', '2019-03-31', '2019-04-30',
               '2019-05-31', '2019-06-30', '2019-07-31', '2019-08-31',
               '2019-09-30', '2019-10-31', '2019-11-30', '2019-12-31'],
              dtype='datetime64[ns]', freq='M')
```

In [53]:
```
# Month Start frequency

pd.date_range('2019-01-01', freq='MS', periods=12)
```

Out[53]:
```
DatetimeIndex(['2019-01-01', '2019-02-01', '2019-03-01', '2019-04-01',
               '2019-05-01', '2019-06-01', '2019-07-01', '2019-08-01',
               '2019-09-01', '2019-10-01', '2019-11-01', '2019-12-01'],
              dtype='datetime64[ns]', freq='MS')
```

In [54]:
```
# Semi-Month frequency (15th and end of Month)

pd.date_range('2019-01-01', freq='SM', periods=12)
```

Out[54]:
```
DatetimeIndex(['2019-01-15', '2019-01-31', '2019-02-15', '2019-02-28',
               '2019-03-15', '2019-03-31', '2019-04-15', '2019-04-30',
               '2019-05-15', '2019-05-31', '2019-06-15', '2019-06-30'],
              dtype='datetime64[ns]', freq='SM-15')
```

In [55]:
```
# Hourly frequency

pd.date_range('2019-01-01', freq='H', periods=12)
```

Out[55]:
```
DatetimeIndex(['2019-01-01 00:00:00', '2019-01-01 01:00:00',
               '2019-01-01 02:00:00', '2019-01-01 03:00:00',
               '2019-01-01 04:00:00', '2019-01-01 05:00:00',
               '2019-01-01 06:00:00', '2019-01-01 07:00:00',
               '2019-01-01 08:00:00', '2019-01-01 09:00:00',
               '2019-01-01 10:00:00', '2019-01-01 11:00:00'],
              dtype='datetime64[ns]', freq='H')
```

In [56]:
```
pd.date_range('2019-11-01 12:56:31', periods=5)
```

Out[56]:
```
DatetimeIndex(['2019-11-01 12:56:31', '2019-11-02 12:56:31',
               '2019-11-03 12:56:31', '2019-11-04 12:56:31',
               '2019-11-05 12:56:31'],
              dtype='datetime64[ns]', freq='D')
```

In [57]:
```
pd.date_range('2019-11-01 12:56:31', freq='H', periods=5)
```

Out[57]:
```
DatetimeIndex(['2019-11-01 12:56:31', '2019-11-01 13:56:31',
               '2019-11-01 14:56:31', '2019-11-01 15:56:31',
               '2019-11-01 16:56:31'],
              dtype='datetime64[ns]', freq='H')
```

In [58]:
```
# Normalize start/end dates to midnight before generating date range

pd.date_range('2019-11-01 12:56:31', periods=5, normalize=True)
```

Out[58]:
```
DatetimeIndex(['2019-11-01', '2019-11-02', '2019-11-03', '2019-11-04',
               '2019-11-05'],
              dtype='datetime64[ns]', freq='D')
```

In [59]:
```
pd.date_range('2019-11-01 12:56:31', periods=5, freq='H', normalize=True)
```

Out[59]:
```
DatetimeIndex(['2019-11-01 00:00:00', '2019-11-01 01:00:00',
               '2019-11-01 02:00:00', '2019-11-01 03:00:00',
               '2019-11-01 04:00:00'],
              dtype='datetime64[ns]', freq='H')
```

In [60]:
```
pd.date_range('2019-01-01', '2019-01-02 23:59', freq='4h')
```

Out[60]:
```
DatetimeIndex(['2019-01-01 00:00:00', '2019-01-01 04:00:00',
               '2019-01-01 08:00:00', '2019-01-01 12:00:00',
               '2019-01-01 16:00:00', '2019-01-01 20:00:00',
               '2019-01-02 00:00:00', '2019-01-02 04:00:00',
               '2019-01-02 08:00:00', '2019-01-02 12:00:00',
               '2019-01-02 16:00:00', '2019-01-02 20:00:00'],
              dtype='datetime64[ns]', freq='4H')
```

In [61]:
```
pd.date_range('2019-01-01', periods=10, freq='2h30min')
```

Out[61]:
```
DatetimeIndex(['2019-01-01 00:00:00', '2019-01-01 02:30:00',
               '2019-01-01 05:00:00', '2019-01-01 07:30:00',
               '2019-01-01 10:00:00', '2019-01-01 12:30:00',
               '2019-01-01 15:00:00', '2019-01-01 17:30:00',
               '2019-01-01 20:00:00', '2019-01-01 22:30:00'],
              dtype='datetime64[ns]', freq='150T')
```

#### Week of month dates

In [62]:
```
# 3rd Tuesday

pd.date_range('2019-01-01', '2020-01-01', freq='WOM-3TUE')
```

Out[62]:
```
DatetimeIndex(['2019-01-15', '2019-02-19', '2019-03-19', '2019-04-16',
               '2019-05-21', '2019-06-18', '2019-07-16', '2019-08-20',
               '2019-09-17', '2019-10-15', '2019-11-19', '2019-12-17'],
              dtype='datetime64[ns]', freq='WOM-3TUE')
```

## Shifting (Leading and Lagging) Data

In [63]:
```
np.random.seed(12345)

ts2 = pd.Series(np.random.randint(1, 100, 6),
                index=pd.date_range('1/1/2019', periods=6, freq='MS'))
ts2
```

Out[63]:
```
2019-01-01    99
2019-02-01    30
2019-03-01     2
2019-04-01    37
2019-05-01    42
2019-06-01    35
Freq: MS, dtype: int64
```

In [64]:
```
ts2.shift(3)
```

Out[64]:
```
2019-01-01     NaN
2019-02-01     NaN
2019-03-01     NaN
2019-04-01    99.0
2019-05-01    30.0
2019-06-01     2.0
Freq: MS, dtype: float64
```

In [65]:
```
ts2.shift(-3)
```

Out[65]:
```
2019-01-01    37.0
2019-02-01    42.0
2019-03-01    35.0
2019-04-01     NaN
2019-05-01     NaN
2019-06-01     NaN
Freq: MS, dtype: float64
```

In [66]:
```
ts2
```

Out[66]:
```
2019-01-01    99
2019-02-01    30
2019-03-01     2
2019-04-01    37
2019-05-01    42
2019-06-01    35
Freq: MS, dtype: int64
```

In [67]:
```
ts2.shift(3, freq='MS')
```

Out[67]:
```
2019-04-01    99
2019-05-01    30
2019-06-01     2
2019-07-01    37
2019-08-01    42
2019-09-01    35
Freq: MS, dtype: int64
```

In [68]:
```
ts2.shift(-3, freq='MS')
```

Out[68]:
```
2018-10-01    99
2018-11-01    30
2018-12-01     2
2019-01-01    37
2019-02-01    42
2019-03-01    35
Freq: MS, dtype: int64
```

In [69]:
```
ts2
```

Out[69]:
```
2019-01-01    99
2019-02-01    30
2019-03-01     2
2019-04-01    37
2019-05-01    42
2019-06-01    35
Freq: MS, dtype: int64
```

In [70]:
```
ts2.shift(3, freq='D')
```

Out[70]:
```
2019-01-04    99
2019-02-04    30
2019-03-04     2
2019-04-04    37
2019-05-04    42
2019-06-04    35
dtype: int64
```

In [71]:
```
ts2.shift(-3, freq='D')
```

Out[71]:
```
2018-12-29    99
2019-01-29    30
2019-02-26     2
2019-03-29    37
2019-04-28    42
2019-05-29    35
dtype: int64
```

In [72]:
```
ts2
```

Out[72]:
```
2019-01-01    99
2019-02-01    30
2019-03-01     2
2019-04-01    37
2019-05-01    42
2019-06-01    35
Freq: MS, dtype: int64
```

In [73]:
```
# 90 minutes

ts2.shift(1, freq='90T')
```

Out[73]:
```
2019-01-01 01:30:00    99
2019-02-01 01:30:00    30
2019-03-01 01:30:00     2
2019-04-01 01:30:00    37
2019-05-01 01:30:00    42
2019-06-01 01:30:00    35
Freq: MS, dtype: int64
```

In [74]:
```
ts2.shift(2, freq='90T')
```

Out[74]:
```
2019-01-01 03:00:00    99
2019-02-01 03:00:00    30
2019-03-01 03:00:00     2
2019-04-01 03:00:00    37
2019-05-01 03:00:00    42
2019-06-01 03:00:00    35
Freq: MS, dtype: int64
```

**Shifting dates with offsets**

In [75]:
```
from pandas.tseries.offsets import Day, MonthEnd
```

In [76]:
```
now = datetime(2019, 1, 15)
now + 3 * Day()
```

Out[76]:
```
Timestamp('2019-01-18 00:00:00')
```

In [77]:
```
now + MonthEnd(), now + MonthEnd(2)
```

Out[77]:
```
(Timestamp('2019-01-31 00:00:00'), Timestamp('2019-02-28 00:00:00'))
```

In [78]:
```
offset = MonthEnd()
offset.rollforward(now), offset.rollback(now)
```

Out[78]:
```
(Timestamp('2019-01-31 00:00:00'), Timestamp('2018-12-31 00:00:00'))
```

In [79]:
```
np.random.seed(12345)

ts3 = pd.Series(np.random.randint(1, 100, 20),
                index=pd.date_range('1/15/2019', periods=20, freq='4d'))
ts3
```

Out[79]:
```
2019-01-15    99
2019-01-19    30
2019-01-23     2
2019-01-27    37
2019-01-31    42
2019-02-04    35
2019-02-08    30
2019-02-12     2
2019-02-16    60
2019-02-20    15
2019-02-24    92
2019-02-28    81
2019-03-04    74
2019-03-08    12
2019-03-12    78
2019-03-16    11
2019-03-20    82
2019-03-24    83
2019-03-28    39
2019-04-01     8
Freq: 4D, dtype: int64
```

In [80]:
```
for (key, group) in ts3.groupby(offset.rollforward):
    print("\nKey: ", key)
    print(group)
```

```
Key:  2019-01-31 00:00:00
2019-01-15    99
2019-01-19    30
2019-01-23     2
2019-01-27    37
2019-01-31    42
Freq: 4D, dtype: int64

Key:  2019-02-28 00:00:00
2019-02-04    35
2019-02-08    30
2019-02-12     2
2019-02-16    60
2019-02-20    15
2019-02-24    92
2019-02-28    81
Freq: 4D, dtype: int64

Key:  2019-03-31 00:00:00
2019-03-04    74
2019-03-08    12
2019-03-12    78
2019-03-16    11
2019-03-20    82
2019-03-24    83
2019-03-28    39
Freq: 4D, dtype: int64

Key:  2019-04-30 00:00:00
2019-04-01    8
Freq: 4D, dtype: int64
```

In [81]:
```
ts3.groupby(offset.rollforward).sum()
```

Out[81]:
```
2019-01-31    210
2019-02-28    315
2019-03-31    379
2019-04-30      8
dtype: int64
```

In [82]:
```
ts3.resample('M').sum()
```

Out[82]:
```
2019-01-31    210
2019-02-28    315
2019-03-31    379
2019-04-30      8
Freq: M, dtype: int64
```

### Open-High-Low-Close (OHLC) resampling

In [83]:
```
pd.date_range('2019-01-01', periods=12, freq='T')
```

Out[83]:
```
DatetimeIndex(['2019-01-01 00:00:00', '2019-01-01 00:01:00',
               '2019-01-01 00:02:00', '2019-01-01 00:03:00',
               '2019-01-01 00:04:00', '2019-01-01 00:05:00',
               '2019-01-01 00:06:00', '2019-01-01 00:07:00',
               '2019-01-01 00:08:00', '2019-01-01 00:09:00',
               '2019-01-01 00:10:00', '2019-01-01 00:11:00'],
              dtype='datetime64[ns]', freq='T')
```

In [84]:
```
np.random.seed(12345)

rng = pd.date_range('2019-01-01', periods=12, freq='T')
ts4 = pd.Series(np.random.randint(10, 50, 12), index=rng)
ts4
```

Out[84]:
```
2019-01-01 00:00:00    44
2019-01-01 00:01:00    47
2019-01-01 00:02:00    39
2019-01-01 00:03:00    11
2019-01-01 00:04:00    46
2019-01-01 00:05:00    47
2019-01-01 00:06:00    44
2019-01-01 00:07:00    39
2019-01-01 00:08:00    11
2019-01-01 00:09:00    24
2019-01-01 00:10:00    37
2019-01-01 00:11:00    26
Freq: T, dtype: int64
```

In [85]:
```
ts4.resample('5min').sum()
```

Out[85]:
```
2019-01-01 00:00:00    187
2019-01-01 00:05:00    165
2019-01-01 00:10:00     63
Freq: 5T, dtype: int64
```

In [86]:
```
ts4.resample('5min').ohlc()
```

Out[86]:

| | open | high | low | close |
|---|---|---|---|---|
| 2019-01-01 00:00:00 | 44 | 47 | 11 | 46 |
| 2019-01-01 00:05:00 | 47 | 47 | 11 | 24 |
| 2019-01-01 00:10:00 | 37 | 37 | 26 | 26 |

## Upsampling and Interpolation

In [87]:
```
np.random.seed(12345)

frame = pd.DataFrame(np.random.randint(20, 50, (2,4)),
                     index=pd.date_range('1/1/2019', periods=2,
                                         freq='W-MON'),
                     columns=['Colorado', 'Texas', 'New York', 'Ohio'])
frame
```

Out[87]:

| | Colorado | Texas | New York | Ohio |
|---|---|---|---|---|
| 2019-01-07 | 22 | 25 | 49 | 21 |
| 2019-01-14 | 24 | 29 | 25 | 22 |

In [88]:
```
df_daily = frame.resample('D').asfreq()
df_daily
```

Out[88]:

| | Colorado | Texas | New York | Ohio |
|---|---|---|---|---|
| 2019-01-07 | 22.0 | 25.0 | 49.0 | 21.0 |
| 2019-01-08 | NaN | NaN | NaN | NaN |
| 2019-01-09 | NaN | NaN | NaN | NaN |
| 2019-01-10 | NaN | NaN | NaN | NaN |
| 2019-01-11 | NaN | NaN | NaN | NaN |
| 2019-01-12 | NaN | NaN | NaN | NaN |
| 2019-01-13 | NaN | NaN | NaN | NaN |
| 2019-01-14 | 24.0 | 29.0 | 25.0 | 22.0 |

In [89]:
```
frame.resample('D').ffill()
```

Out[89]:

| | Colorado | Texas | New York | Ohio |
|---|---|---|---|---|
| 2019-01-07 | 22 | 25 | 49 | 21 |
| 2019-01-08 | 22 | 25 | 49 | 21 |
| 2019-01-09 | 22 | 25 | 49 | 21 |
| 2019-01-10 | 22 | 25 | 49 | 21 |
| 2019-01-11 | 22 | 25 | 49 | 21 |
| 2019-01-12 | 22 | 25 | 49 | 21 |
| 2019-01-13 | 22 | 25 | 49 | 21 |
| 2019-01-14 | 24 | 29 | 25 | 22 |

In [90]:
```
frame.resample('D').ffill(limit=2)
```

Out[90]:

| | Colorado | Texas | New York | Ohio |
|---|---|---|---|---|
| 2019-01-07 | 22.0 | 25.0 | 49.0 | 21.0 |
| 2019-01-08 | 22.0 | 25.0 | 49.0 | 21.0 |
| 2019-01-09 | 22.0 | 25.0 | 49.0 | 21.0 |
| 2019-01-10 | NaN | NaN | NaN | NaN |
| 2019-01-11 | NaN | NaN | NaN | NaN |
| 2019-01-12 | NaN | NaN | NaN | NaN |
| 2019-01-13 | NaN | NaN | NaN | NaN |
| 2019-01-14 | 24.0 | 29.0 | 25.0 | 22.0 |

In [91]:
```
frame
```

Out[91]:

| | Colorado | Texas | New York | Ohio |
|---|---|---|---|---|
| 2019-01-07 | 22 | 25 | 49 | 21 |
| 2019-01-14 | 24 | 29 | 25 | 22 |

In [92]:
```
frame.resample('W-THU').asfreq()
```

Out[92]:

| | Colorado | Texas | New York | Ohio |
|---|---|---|---|---|
| 2019-01-10 | NaN | NaN | NaN | NaN |
| 2019-01-17 | NaN | NaN | NaN | NaN |

In [93]:
```
frame.resample('W-THU').ffill()
```

Out[93]:

| | Colorado | Texas | New York | Ohio |
|---|---|---|---|---|
| 2019-01-10 | 22 | 25 | 49 | 21 |
| 2019-01-17 | 24 | 29 | 25 | 22 |

In [ ]:
```

```