```
data.loc[data.slug == 'bitcoin', 'close']['2018-01']
'Output':
date
2018-01-01    13657.2
2018-01-02    14982.1
2018-01-03    15201.0
2018-01-04    15599.2
2018-01-05    17429.5
2018-01-06    17527.0
2018-01-07    16477.6
2018-01-08    15170.1
2018-01-09    14595.4
2018-01-10    14973.3
```

Find the mean market value of Ethereum for the month of January.

```
data.loc[data.slug == 'ethereum', 'market']['2018-01'].mean()
'Output':
96739480000.0
```

Resampling Datetime Objects

A Pandas DataFrame with an index of **DatetimeIndex**, **PeriodIndex**, or **TimedeltaIndex** can be resampled to any of the date time frequencies from seconds, to minutes, to months. Let's see some examples.

Let's get the average monthly closing values for Litecoin.

```
data.loc[data.slug == 'bitcoin', 'close'].resample('M').mean().head()
'Output':
date
2013-04-30    139.250000
2013-05-31    119.993226
2013-06-30    107.761333
2013-07-31     90.512258
2013-08-31    113.905161
Freq: M, Name: close, dtype: float64
```

Get the average weekly market value of Bitcoin Cash.

```
data.loc[data.symbol == 'BCH', 'market'].resample('W').mean().head()
'Output':
date
2017-07-23    0.000000e+00
2017-07-30    0.000000e+00
2017-08-06    3.852961e+09
2017-08-13    4.982661e+09
2017-08-20    7.355117e+09
Freq: W-SUN, Name: market, dtype: float64
```

Convert to Datetime Datatype Using 'to_datetime'

Pandas uses the **to_datetime** method to convert strings to Pandas datetime datatype. The **to_datetime** method is smart enough to infer a **datetime** representation from a string of dates passed with different formats. The default output format of **to_datetime** is in the following order: **year, month, day, minute, second, millisecond, microsecond, nanosecond**.

The input to **to_datetime** is recognized as **month, day, year**. Although, it can easily be modified by setting the attributes **dayfirst** or **yearfirst** to **True**.

For example, if **dayfirst** is set to **True**, the input is recognized as **day, month, year**.

Let's see an example of this.

```
# create list of dates
my_dates = ['Friday, May 11, 2018', '11/5/2018', '11-5-2018', '5/11/2018',
            '2018.5.11']
pd.to_datetime(my_dates)
'Output':
DatetimeIndex(['2018-05-11', '2018-11-05', '2018-11-05', '2018-05-11',
               '2018-05-11'],
              dtype='datetime64[ns]', freq=None)
```