In [ ]:

Time Resampling

Let's learn how to sample time series data! This will be useful later on in the course!

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

In [2]:
```
%matplotlib inline
import matplotlib.pyplot as plt
```

In [3]:
```
# Grab data
# Faster alternative
# df = pd.read_csv('time_data/walmart_stock.csv',index_col='Date')
df = pd.read_csv('time_data/walmart_stock.csv')
```

In [4]:
```
df.head()
```

Out[4]:

	Date	Open	High	Low	Close	Volume	Adj Close
0	2012-01-03	59.970001	61.060001	59.869999	60.330002	12668800	52.619235
1	2012-01-04	60.209999	60.349998	59.470001	59.709999	9593300	52.078475
2	2012-01-05	59.349998	59.619999	58.369999	59.419998	12768200	51.825539
3	2012-01-06	59.419998	59.450001	58.869999	59.000000	8069400	51.459220
4	2012-01-09	59.029999	59.549999	58.919998	59.180000	6679300	51.616215

Create a date index from the date column

In [5]:
```
df['Date'] = df['Date'].apply(pd.to_datetime)
```

In [6]:
```
df.head()
```

Out[6]:

	Date	Open	High	Low	Close	Volume	Adj Close
0	2012-01-03	59.970001	61.060001	59.869999	60.330002	12668800	52.619235
1	2012-01-04	60.209999	60.349998	59.470001	59.709999	9593300	52.078475
2	2012-01-05	59.349998	59.619999	58.369999	59.419998	12768200	51.825539
3	2012-01-06	59.419998	59.450001	58.869999	59.000000	8069400	51.459220
4	2012-01-09	59.029999	59.549999	58.919998	59.180000	6679300	51.616215

In [7]:
```
df.set_index('Date',inplace=True)
```

In [8]:
```
df.head()
```

Out[8]:

Date	Open	High	Low	Close	Volume	Adj Close
2012-01-03	59.970001	61.060001	59.869999	60.330002	12668800	52.619235
2012-01-04	60.209999	60.349998	59.470001	59.709999	9593300	52.078475
2012-01-05	59.349998	59.619999	58.369999	59.419998	12768200	51.825539
2012-01-06	59.419998	59.450001	58.869999	59.000000	8069400	51.459220
2012-01-09	59.029999	59.549999	58.919998	59.180000	6679300	51.616215

resample()

A common operation with time series data is resamplling based on the time series index. Let see how to use the resample() method.

All possible time series offfest strings

Alias	Description
B	business day frequency
C	custom business day frequency (experimental)
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 endfrequency
QS	quarter start frequency
BQS	business quarter start frequency
A	year end frequency
BA	business year end frequency
AS	year start frequency
BAS	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 [9]:
```
# Our index
df.index
```

Out[9]:
```
DatetimeIndex(['2012-01-03', '2012-01-04', '2012-01-05', '2012-01-06',
               '2012-01-09', '2012-01-10', '2012-01-11', '2012-01-12',
               '2012-01-13', '2012-01-17',
               ...
               '2016-12-16', '2016-12-19', '2016-12-20', '2016-12-21',
               '2016-12-22', '2016-12-23', '2016-12-27', '2016-12-28',
               '2016-12-29', '2016-12-30'],
              dtype='datetime64[ns]', name='Date', length=1258, freq=None)
```

You need to call resample with the rule parameter, then you need to call some sort of aggregation function. This is because due to resampling, we need some sort of mathematical rule to join the rows by (mean,sum,count,etc...)

In [10]:
```
# Yearly Means
df.resample(rule='A').mean()
```

Out[10]:

Date	Open	High	Low	Close	Volume	Adj Close
2012-12-31	67.158680	67.602120	66.786520	67.215120	9.239015e+06	59.389349
2013-12-31	75.264048	75.729405	74.843055	75.320516	6.951496e+06	68.147179
2014-12-31	77.274524	77.740040	76.864405	77.327381	6.515612e+06	71.709712
2015-12-31	72.569405	73.064167	72.034802	72.491111	9.040769e+06	68.831426
2016-12-31	69.481349	70.019643	69.023492	69.547063	9.371645e+06	68.054229

Custom Resampling

You could technically also create your own custom resampling function:

In [11]:
```
def first_day(entry):
    """
    Returns the first instance of the period, regardless of sampling rate.
    """
    return entry[0]
```

In [12]:
```
df.resample(rule='A').apply(first_day)
```

Out[12]:

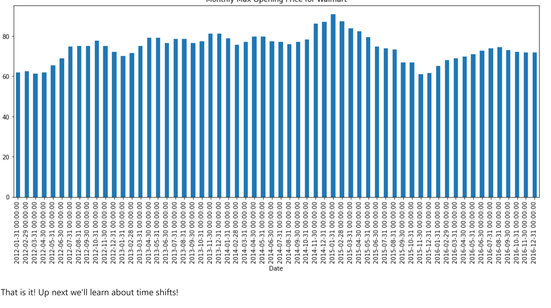
Date	Open	High	Low	Close	Volume	Adj Close
2012-12-31	59.970001	61.060001	59.869999	60.330002	12668800	52.619235
2013-12-31	68.930000	69.239998	68.449997	69.239998	10390800	61.879708
2014-12-31	78.720001	79.470001	78.500000	78.910004	6878000	72.254228
2015-12-31	86.269997	86.720001	85.550003	85.900002	4501800	80.624861
2016-12-31	60.500000	61.490002	60.360001	61.459999	11989200	59.289713

In [13]:
```
df['Close'].resample('A').mean().plot(kind='bar')
plt.title('Yearly Mean Close Price for Walmart')
```

Out[13]:
```
Text(0.5, 1.0, 'Yearly Mean Close Price for Walmart')
```

In [14]:
```
df['Open'].resample('M').max().plot(kind='bar',figsize=(16,6))
plt.title('Monthly Max Opening Price for Walmart')
```

Out[14]:
```
Text(0.5, 1.0, 'Monthly Max Opening Price for Walmart')
```



That is it! Up next we'll learn about time shifts!