In [ ]:

Introduction to Time Series with Pandas

A lot of our financial data will have a datatime index, so let's learn how to deal with this sort of data with pandas!

In [21]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

In [22]:
```
from datetime import datetime
```

In [3]:
```
# To illustrate the order of arguments
my_year = 2017
my_month = 1
my_day = 2
my_hour = 13
my_minute = 30
my_second = 15
```

In [23]:
```
# January 2nd, 2017
my_date = datetime(my_year,my_month,my_day)
```

In [24]:
```
# Defaults to 0:00
my_date
```

Out[24]:
```
datetime.datetime(2017, 1, 2, 0, 0)
```

In [25]:
```
# January 2nd, 2017 at 13:30:15
my_date_time = datetime(my_year,my_month,my_day,my_hour,my_minute,my_second)
```

In [26]:
```
my_date_time
```

Out[26]:
```
datetime.datetime(2017, 1, 2, 13, 30, 15)
```

You can grab any part of the datetime object you want

In [27]:
```
my_date.day
```

Out[27]:
```
2
```

In [28]:
```
my_date_time.hour
```

Out[28]:
```
13
```

Pandas with Datetime Index

You'll usually deal with time series as an index when working with pandas dataframes obtained from some sort of financial API. Fortunately pandas has a lot of functions and methods to work with time series!

In [29]:
```
# Create an example datetime list/array
first_two = [datetime(2016, 1, 1), datetime(2016, 1, 2)]
first_two
```

Out[29]:
```
[datetime.datetime(2016, 1, 1, 0, 0), datetime.datetime(2016, 1, 2, 0, 0)]
```

In [11]:
```
# Converted to an index
dt_ind = pd.DatetimeIndex(first_two)
dt_ind
```

Out[11]:
```
DatetimeIndex(['2016-01-01', '2016-01-02'], dtype='datetime64[ns]', freq=None)
```

In [30]:
```
# Attached to some random data
data = np.random.randn(2,2)
print(data)
cols = ['A','B']
```

```
[[ 0.96702648  0.97661239]
 [-0.66547711  0.00564531]]
```

In [13]:
```
df = pd.DataFrame(data,dt_ind,cols)
```

In [14]:
```
df
```

Out[14]:

	A	B
2016-01-01	0.020566	-1.179126
2016-01-02	0.148169	-1.549081

In [15]:
```
df.index
```

Out[15]:
```
DatetimeIndex(['2016-01-01', '2016-01-02'], dtype='datetime64[ns]', freq=None)
```

In [16]:
```
# Latest Date Location
df.index.argmax()
```

Out[16]:
```
1
```

In [17]:
```
df.index.max()
```

Out[17]:
```
Timestamp('2016-01-02 00:00:00')
```

In [18]:
```
# Earliest Date Index Location
df.index.argmin()
```

Out[18]:
```
0
```

In [19]:
```
df.index.min()
```

Out[19]:
```
Timestamp('2016-01-01 00:00:00')
```

Great, let's move on!