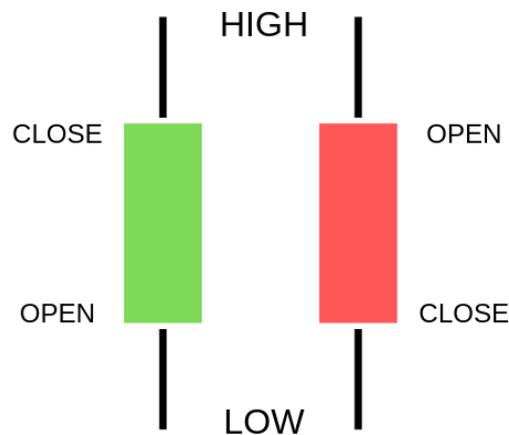


Plot Candlestick Charts in Python

Introduction

Candlestick Charts are one of the best ways to visualize stock data because they give us very detailed information about the evolution of share prices. The programming language Python is a great choice for financial analysis and the module `matplotlib`, additionally offers a lot of tools for advanced data visualization.

In this tutorial, we are going to write a little script that plots a professional candlestick chart for a specific company at the stock market. If you are not familiar with the concept of that diagram type, take a look at the following image.



As you can see, a candle stick can be either positive (green) or negative (red). The former means that the close price is higher than the open price and the latter means the opposite. We can gather four different values out of one candlestick:

- The highest share price of that day (top point of the black line)
- The lowest share price of that day (bottom point of the black line)
- The share price when the market opened (top point of the red area or bottom point of the green area)
- The share price when the market closed (top point of the green area or bottom point of the red area)

Additionally, we can also look at the price span of the respective day which is visualized in the colored area. Thus, this type of chart provides a lot of information.

Imports

For our script we will need to import a couple of libraries. Most of them are external and need to be installed.

```
import datetime as dt
import matplotlib.dates as mdates
```

```
import matplotlib.pyplot as plt
import pandas_datareader as web
from mpl_finance import candlestick_ohlc
```

Let us take a quick look at each of these:

- **datetime:** We will use this one to define our desired time span
- **matplotlib.dates:** This library will convert our dates into the necessary number format
- **matplotlib.pyplot:** Will be used for displaying our chart in the end
- **pandas_datareader:** The module that will load the desired stock data
- **candlestick_ohlc** from **mpl_finance:** Our main library for plotting

Except for the datetime module, none of those libraries is included in Core Python. This means that you will probably need to install them with pip.

Preparing The Data

In order to plot our data properly, we will first need to load it and to set it up. We will start by defining our desired time span.

```
start = dt.datetime(2010, 1, 1)
end = dt.datetime.now()
```

Our start date is January 1st, 2010 and our end date is defined as the current date. This means that we are looking at the data from our start date up until now. The next step is the definition of our data reader.

```
df = web.DataReader('AAPL', 'yahoo', start, end)
```

Here, we specify that we want to use the *Yahoo Finance API* in order to download the data for the ticker symbol *AAPL* (which is Apple). The time span we are looking at is defined by start and end. What we get is a data frame that contains our requested values. We can print out the first few rows to see the structure.

```
print(df.head())
```

	High	Low	Open	Close	Volume	Adj Close
Date						
2010-01-04	30.642857	30.340000	30.490000	30.572857	123432400.0	20.386072
2010-01-05	30.798571	30.464285	30.657143	30.625713	150476200.0	20.421322
2010-01-06	30.747143	30.107143	30.625713	30.138571	138040000.0	20.096491
2010-01-07	30.285715	29.864286	30.250000	30.082857	119282800.0	20.059338
2010-01-08	30.285715	29.865715	30.042856	30.282858	111902700.0	20.192701

Right now, we have two columns that we don't need for our chart. These are *Volume* and *Adj Close*. For our candlestick chart, we need the values *Open*, *High*, *Low*, and *Close* in that exact order. Also,

we will need *Date* for our x-axis. In pandas, it is quite simple to select and reorder columns in a data frame.

```
df = df[['Open', 'High', 'Low', 'Close']]
```

Basically, we are just selecting the four relevant columns in the right order and replacing our current data frame. Notice that we use double square brackets here.

Now, we have our columns in the right order but there is still a problem. Our date doesn't have the right format and since it is the index, we cannot manipulate it. Therefore, we need to reset the index and then convert our datetime to a number.

```
df.reset_index(inplace=True)
df['Date'] = df['Date'].map(mdates.date2num)
```

Here, we drop the index and map the *date2num* function onto our *Date* column. Now, our datetime values are converted into numbers that matplotlib can deal with. We can start with the plotting.

Plotting The Data

For the final step, we will define our plots and visualize the data we have prepared.

```
ax = plt.subplot()
candlestick_ohlc(ax, df.values, width=5, colorup='g', colordown='r')
ax.xaxis_date()
ax.grid(True)
plt.show()
```

First, we define a new subplot (also called axis) for our data. Then, we use the candlestick function, in order to plot our values. Also, we define the colors and the width of the sticks. Additionally, we put the dates on the x-axis and turn on the grid. Finally, we show our plot. The end result looks like this:



Of course, the more you zoom in, the more you will be able to see the individual candlesticks. This is a great way to visualize four different values in a single diagram.