

The cell below allows the user to enter a comma-separated string of tickers that will be analyzed.

In addition to this, the user can decide which price should be analyzed. The options are: Open, High, Low, Close, Adj Close, Volume.

```
In [19]: import pandas.io.data as pweb
import datetime
import requests
import numpy
import seaborn
import matplotlib.pyplot as plt
%matplotlib auto

start = datetime.datetime(2016,1,1)
end = datetime.datetime(2016,12,16)

ticker = []
inputString = input("Input the Stocks, separate each stock by ',': ")
ticker = inputString.split(",")
price = input('\nEnter which price you want to analyze - Open/High/Low/C
lose/Adj Close/Volume: ')

data = pweb.get_data_yahoo(ticker,start,end)[(price)]
quote = pweb.get_quote_yahoo(ticker)

info = {}

for tick in ticker:
    info[tick] = {}
    info[tick]["PE"] = quote.loc[tick,"PE"]
    info[tick]["change_pct"] = quote.loc[tick,"change_pct"]
    info[tick]["last"] = quote.loc[tick,"last"]
    info[tick]["short_ratio"] = quote.loc[tick,"short_ratio"]
```

Using matplotlib backend: Qt4Agg

Input the Stocks, separate each stock by ',': AAPL,FB,MSFT

Enter which price you want to analyze - Open/High/Low/Close/Adj Close/V
olume: Adj Close

Shows the selected price of each stock over the past 5 days.

In [20]: `data.tail()`

Out[20]:

	AAPL	FB	MSFT
Date			
2016-12-12	113.300003	117.769997	62.169998
2016-12-13	115.190002	120.309998	62.980000
2016-12-14	115.190002	120.209999	62.680000
2016-12-15	115.820000	120.570000	62.580002
2016-12-16	115.970001	119.870003	62.299999

Additional information about each stock

In [21]: `quote`

Out[21]:

	PE	change_pct	last	short_ratio	time
AAPL	13.96	0.13	115.97	1.73	4:00pm
FB	46.28	-0.58	119.87	0.83	4:00pm
MSFT	29.53	-0.45	62.30	1.77	4:00pm

In [13]: `seaborn.factorplot(x=ticker, y='PE', data=quote, kind='bar', ci=None)
plt.xlabel('Stock Ticker')
plt.ylabel('PE Ratio')`

Out[13]: `<matplotlib.text.Text at 0x1637b10c898>`

Earnings per Share

```
In [23]: d = datetime.datetime(2016,12,16)
data.ix[d]

print("Earnings per Share \n")

for tick in ticker:
    print(tick,":",data.ix[d][tick]/info[tick]["PE"])
    info[tick]["EPS"] = data.ix[d][tick]/info[tick]["PE"]
```

Earnings per Share

AAPL : 8.30730666189

FB : 2.59010378133

MSFT : 2.10971889604

Historical price of stocks

```
In [24]: data.plot()  
plt.xlabel("Date")  
plt.ylabel("Price")  
plt.title("Historical Data of Stocks")
```

```
Out[24]: <matplotlib.text.Text at 0x1637c424160>
```

Allows the user to order the stocks by: PE, EPS, change_pct, last, short_ratio

```
In [26]: field = input("Choose a field to order by: ")  
  
print("\nTicker :", field, "\n")  
  
ordered = sorted(info.items(), key = lambda tup: (-tup[1][field]))  
  
for i in ordered:  
    print(i[0],":", i[1][field])
```

Choose a field to order by: short_ratio

Ticker : short_ratio

MSFT : 1.77

AAPL : 1.73

FB : 0.83

```
In [ ]:
```