Assignment 2:

Installation

1. Open a command prompt in the folder, and type 'pip install –r requirements.txt' to make sure that all the dependencies are installed

2. To run the file, on a command prompt type 'python a2.py'

3. Please note: This script makes use of fix_yahoo_finance, which may bug out during execution and fetch blank dataframes. Please re-run the program

Explanation

1. **Write a program that prompts the user to enter any valid stock symbol available in an appropriate financial website such as Google Finance, Yahoo Finance, Quandl, CityFALCON, or another similar source for NYSE & NASDAQ. Ensure proper error handling for wrong user inputs**

   The following code downloads data from yahoo finance by prompting the user to enter a ticker to download data for, with error handling

```
yahoo_finance_bridge()
start = datetime.datetime(2007, 6, 1)
end = datetime.datetime(2018, 1, 1)
ticker = raw_input('Please Enter a Valid (single) Ticker to fetch the data for. Example \'AAPL\'')
data = pdr.get_data_yahoo(ticker, start=start, end=end, auto_adjust=True)
if data.empty:
    print('Please enter valid ticker')
    return 1
data = data['Open']
data.name = ticker
data = data.sort_index(axis=0, ascending=True)
```

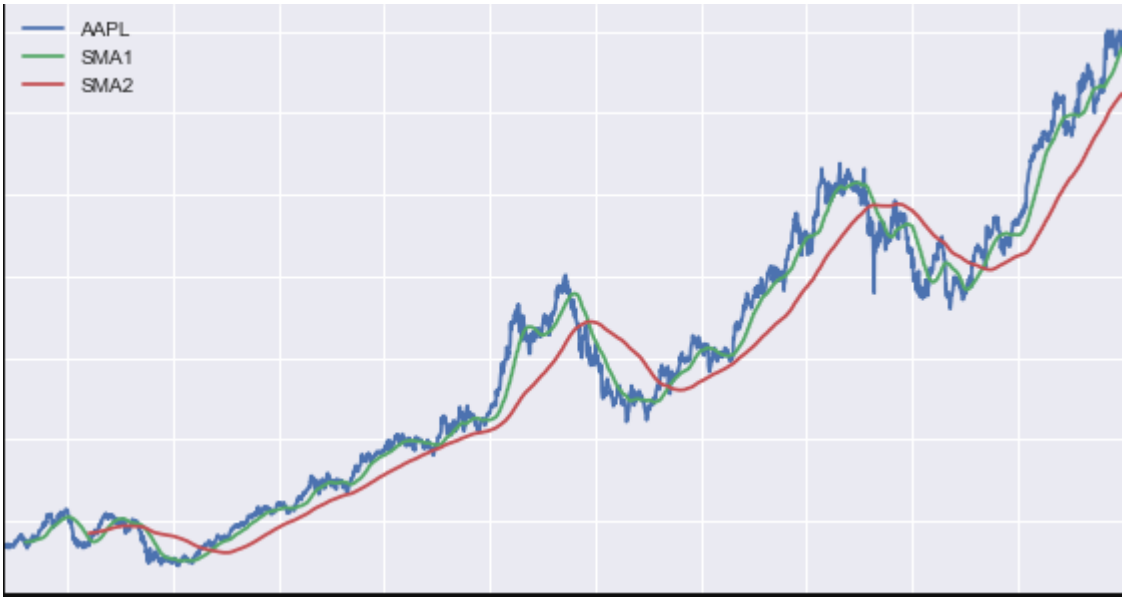
2. **Download/access End-of-day and Hourly data for last 10 years for user entered ticker from an appropriate financial website such as Google Finance, Yahoo Finance, Quandl, CityFALCON, or another similar source**

   Downloads data from Yahoo Finance

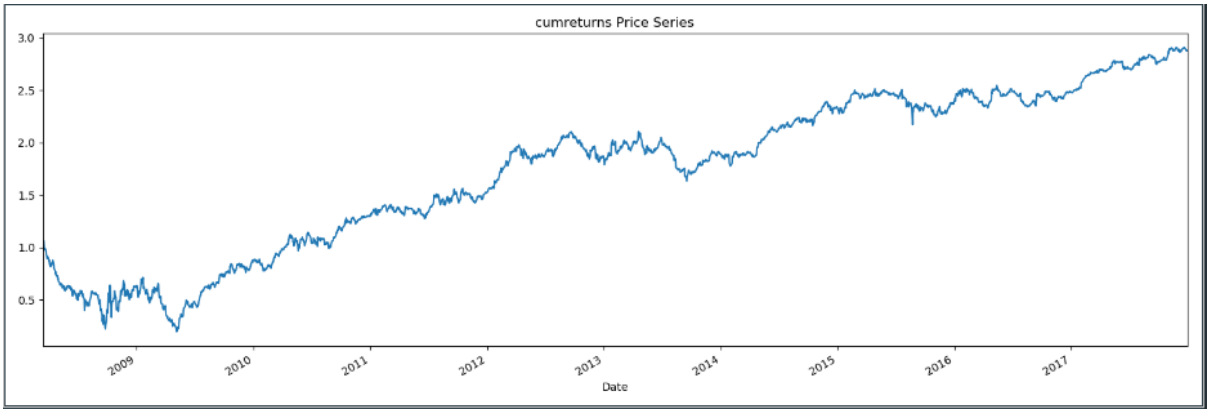
```
data = pdr.get_data_yahoo(ticker, start=start, end=end, auto_adjust=True)
if data.empty:
    print('Please enter valid ticker')
    return 1
```

3. **Constitute a Simple Moving Average crossover system for trend trading on the daily time frame**

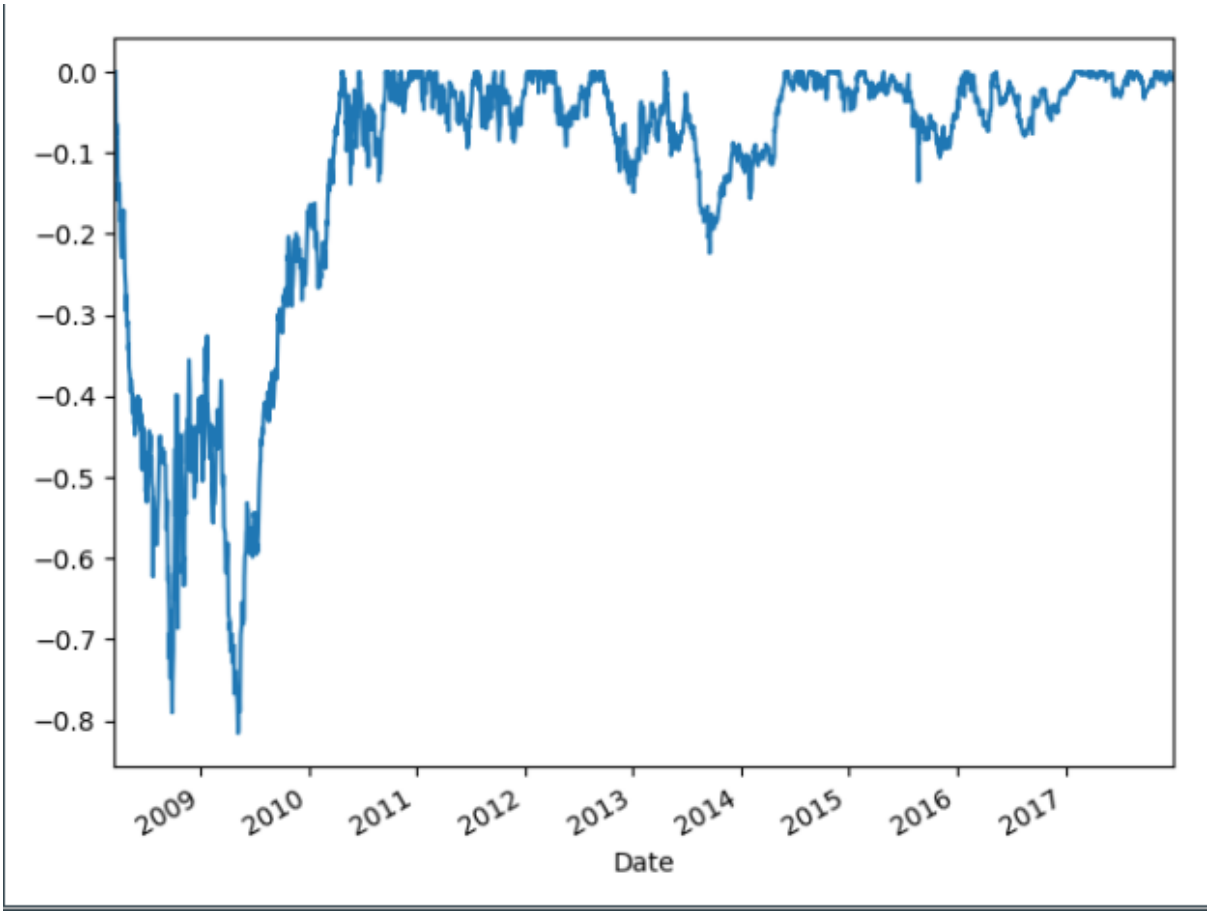
```
# ====== Step 2: Calc. SMA =====
data['SMA1'] = data.rolling(50).mean()
data['SMA2'] = data[ticker].rolling(200).mean()
data.plot(title=ticker + ' stock price | 50 & 200 days SMAs', figsize=(10, 6))
plt.show()
plt.close()
```



A strat is made, and a portfolio based on the strat is then constructed



Its drawdown is as follows:



4. **Calculate 2 period RSI (Relative Strength Indicator) values for the stock in question on the hourly time frame to identify overbought and oversold conditions**

```python
def RSI(series, period):
    delta = series.diff().dropna()
    u = delta * 0
    d = u.copy()
    u[delta > 0] = delta[delta > 0]
    d[delta < 0] = -delta[delta < 0]
    u[u.index[period-1]] = np.mean( u[:period] ) #first value is sum of avg gains
    u = u.drop(u.index[:(period-1)])
    d[d.index[period-1]] = np.mean( d[:period] ) #first value is sum of avg losses
    d = d.drop(d.index[:(period-1)])
    rs = pd.stats.moments.ewma(u, com=period-1, adjust=False) / \
         pd.stats.moments.ewma(d, com=period-1, adjust=False)
    return 100 - 100 / (1 + rs)
```

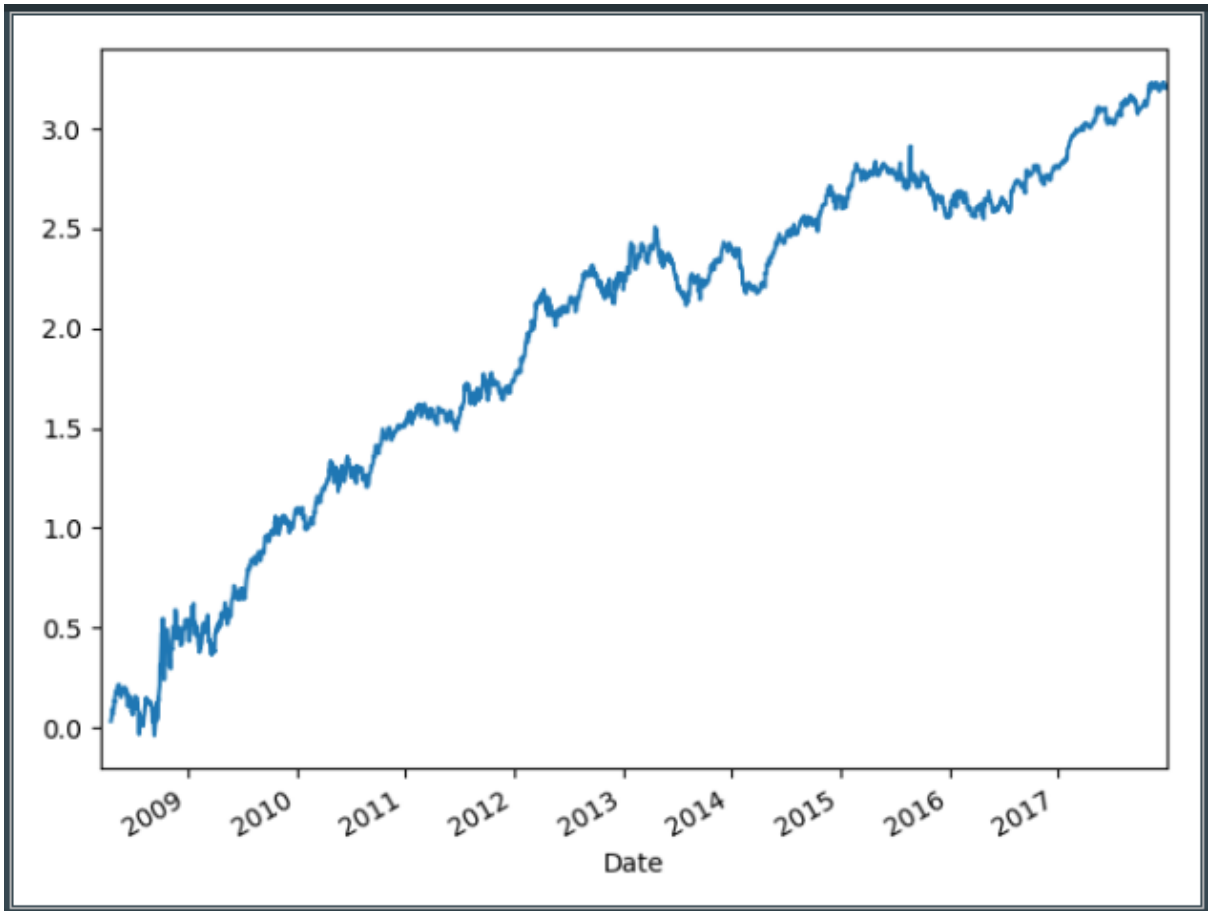
5. **Exit trade on MA Crossover in opposite direction AND favorable RSI indications on smaller time scale**

```python
# Fill our newly created position column - set to sell (-1) when the price hits the upper banc
for row in range(len(data)):

    if (data['RSI'].iloc[row] > 69) and (data['RSI'].iloc[row - 1] < 69):
        data['Position'].iloc[row] = 1

    if (data['RSI'].iloc[row] < 30) and (data['RSI'].iloc[row - 1] > 30):
        data['Position'].iloc[row] = -1
```

Cumulative Strat Return



6. **Estimate and compare historic performance**

Stats for Strategy Return from 2008-03-14 00:00:00 - 2017-12-29 00:00:00

Annual risk-free rate considered: 0.00%

Summary:

Total Return	Sharpe	CAGR	Max Drawdown
-	0.32	-	-118.40%

Annualized Returns:

mtd	3m	6m	ytd	1y	3y	5y	10y	incep.
0.02%	3.58%	5.66%	14.08%	14.15%	6.34%	7.11%	-	-

Periodic:

	daily	monthly	yearly
sharpe	0.32	0.53	0.78
mean	80.42%	81.42%	24.72%
vol	249.41%	154.34%	31.63%
skew	1.55	8.05	2.16
kurt	248.83	74.65	5.22
best	357.34%	431.86%	101.99%
worst	-298.85%	-78.13%	-3.83%

Drawdowns:

max	avg	# days
-118.40%	-5.15%	27.34

Misc:

avg. up month	14.11%
avg. down month	-5.65%
up year %	88.89%
12m up %	83.18%