CCU

Computer Programming and Applications - Fall 2023

Mid-Homework - Stock Market Analysis

Due Date: 2023/11/12 23:59 pm

Abstract

This project is meant to cap off the first half of our course, which mainly dealt with learning the libraries that we use in the late course(Numpy, pandas, matplot…), the second half of the project will deal more with quantitative trading techniques and platforms..

In this homework we'll be analyzing stock data related to a few car companies, from Jan 1 2012 to Jan 1 2022. Keep in mind that this project is mainly just to practice your skills with matplotlib, pandas, and numpy. Don't infer financial trading advice from the analysis we do here!

homework submission:

There are four parts (Part 0 is not included) of this homework, and you will gain 25 points for each completed part (total 100 points). Please follow the requirement in each part and response your answer (Your source code and program results) in a work or PowerPoint file, then send it to the homework area in E-course.

Part 0 – Import Package

Please follow the **Sample_code_0**, import the various libraries you will need-you can always just come back up here or import as you go alongChoose a tutorial that is suitable for your local environment.

Sample_code_0:

```
1. import numpy as np
2. import pandas as pd
3. import matplotlib.pyplot as plt
```

Part 1 - Getting the Data (25%)

Please follow the **sample_code_1**, use pandas_datareader to obtain the historical stock information for Tesla from Jan 1, 2013 to Jan 1, 2023.

Sample_code_1:

```
1. import pandas_datareader
2. import datetime
3. import pandas_datareader.data as web
4. start = datetime.datetime(2013, 1, 1)
5. end = datetime.datetime(2023, 1, 1)
6. tesla = web.DataReader("TSLA", 'yahoo', start, end)
7. print(tesla.head())
```

Part 1 Requirements:

Please repeat the step and add other two car companies – Ford and GM (code: 'F' and 'GM') , grab their data and use head() function to show the result, response the source code and results in your document.

Hint:

If you can't install pandas_datareader through pip, please install the followings:

1. Microsoft Visual C++ 14.0 (https://visualstudio.microsoft.com/visual-cpp-build-tools/)

2. lxml lib (https://www.lfd.uci.edu/~gohlke/pythonlibs/#lxml)

   Make sure you have download the correct file according to you python version and OS.

```
C:\Users\CCUDUCK\Desktop>pip install lxml-4.9.0-cp311-cp311-win_amd64.whl
```

Part 2 - Visualizing the Data (25%)

Please follow the **sample_code_2**, Create a CandleStick chart for Ford in January 2012 (too many dates won't look good for a candlestick chart)

Sample_code_2:

```
import pandas as pd
import datetime
import pandas_datareader.data as web
import mplfinance as mpf

start = datetime.datetime(2012, 1, 1)
end = datetime.datetime(2022, 1, 1)
ford = web.DataReader("F", 'yahoo', start, end)

# Rest the index to get a column of January Dates
ford_reset = ford.loc['2012-01':'2012-01']

mpf.plot(ford_reset, type='candle', style='charles',
         title='S&P 500, Jan 2012',
         ylabel='Price ($)',
         ylabel_lower='Shares \nTraded',
         volume=True,
         mav=(3,6,9))
```

Part 2 Requirements:

1. **(20%) Please repeat the step and add other two car companies, use mpf.plot() function to show the CandleStick chart, response the source code and results in your document**.

2. **(5%) Combine all of the three CandleStick charts in one figure (Hint: use add_subslot() function)**

Part 3 - Basic Financial Analysis (25%)

Daily Percentage Change

First, we will begin by calculating the daily percentage change. Daily percentage change is defined by the following formula:

$$r_t = \frac{p_t}{p_{t-1}} - 1$$

This defines r_t (return at time t) as equal to the price at time t divided by the price at time t-1 (the previous day) minus 1. Basically, this just informs you of your percent gain (or loss) if you bought the stock on day and then sold it the next day. While this isn't necessarily helpful for attempting to predict future values of the stock, it's very helpful in analyzing the volatility of the stock. If daily returns have a wide distribution, the stock is more volatile from one day to the next. Let's calculate the percent returns and then plot them with a histogram, and decide which stock is the most stable!

Please follow **sample_code_3**,

1. create a new column for each dataframe called returns. This column will be calculated from the Close price column. There are two ways to do this, either a simple calculation using the .shift() method that follows the formula above, or you can also use pandas' built in pct_change method.

2. plot a histogram of each companies returns. Either do them separately, or stack them on top of each other.

Sample_code_3:

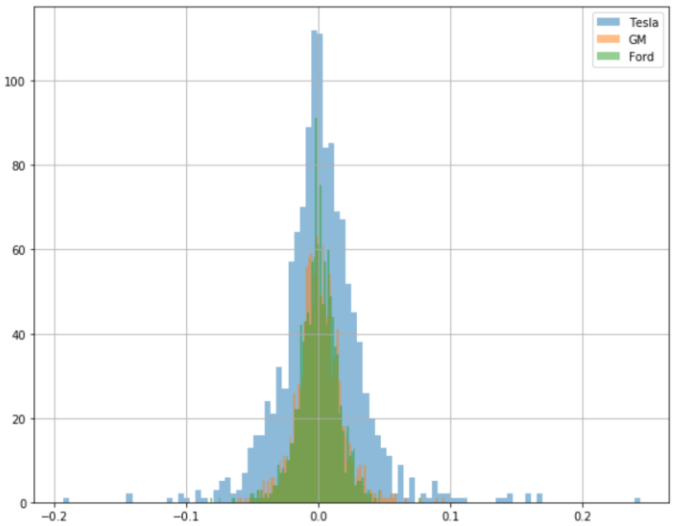
```
import pandas as pd
import datetime
import pandas_datareader.data as web
import mplfinance as mpf
import matplotlib.pyplot as plt

start = datetime.datetime(2012, 1, 1)
end = datetime.datetime(2022, 1, 1)
tesla = web.DataReader("TSLA", 'yahoo', start, end)

# Method 1: Using shift
tesla['returns'] = (tesla['Close'] / tesla['Close'].shift(1) ) - 1

# Method 2: Using pct_change
tesla['returns'] = tesla['Close'].pct_change(1)
print(tesla.head())

tesla['returns'].hist(bins=100,label='Tesla',figsize=(10,8),alpha=0.5)
#ford hist here
```

```
#CM hist here
plt.show()
```

Part 3 Requirements:

1. **(20%) Please repeat the step and add other two car companies, calculate the daily percentage change for each company, then plot a histogram of each companies returns, response the source code and results in your document**.

2. **(5%) Compare all of the three histograms in one figure:**



Part 4 - Cumulative Daily Returns (25%)

Lets us say there is a stock 'ABC' that is being actively traded on an exchange. ABC has the following prices corresponding to the dates given:

Date	Price
01/01/2018	10
01/02/2018	15
01/03/2018	20
01/04/2018	25

Daily Return : Daily return is the profit/loss made by the stock compared to the previous day. (This is what ew just calculated above). A value above one indicates profit, similarly a value below one indicates loss. It is also expressed in percentage to convey the information better. (When expressed as percentage, if the value is above 0, the stock had give you profit else loss).
So for the above example the daily returns would be:

Date	Daily Return	%Daily Return
01/01/2018	10/10 = 1	-
01/02/2018	15/10 = 3/2	50%
01/03/2018	20/15 = 4/3	33%
01/04/2018	25/20 = 5/4	20%

Cumulative Return: While daily returns are useful, it doesn't give the investor a immediate insight into the gains he had made till date, especially if the stock is very volatile. Cumulative return is computed relative to the day investment is made. If cumulative return is above one, you are making profits else you are in loss.
So for the above example cumulative gains are as follows:

Date	Cumulative Return	%Cumulative Return
01/01/2018	10/10 = 1	100 %
01/02/2018	15/10 = 3/2	150 %
01/03/2018	20/10 = 2	200 %
01/04/2018	25/10 = 5/2	250 %

The formula for a cumulative daily return is:

$$i_i = (1 + r_t) * i_{t-1}$$

Here we can see we are just multiplying our previous investment at i at t-1 by 1+our percent returns. Pandas makes this very simple to calculate with its cumprod() method.

Using something in the following manner:

df[daily_cumulative_return] = (1 + df[pct_daily_return]).cumprod()

Part 4 Requirements:

1. **(20%) Please Create a cumulative daily return column for each car company's dataframe, response the source code and results in your document**.

2. **(5%) Please plot three companies' Cumulative Return columns against the time series index in one figure. And answer which stock showed the highest return for a $1 invested? Which showed the lowest?**