

02-Live1 Building a Market Simulator

Compiled by Shipra De, Fall 2016

Okay, can someone volunteer please to monitor Piazza for me? So I request the following. Hello. I recently posted a notice to Piazza providing the details about this broadcast. Please advise the people on Piazza that if they have questions they should post their questions to that note. Does that make sense? And then whoever the volunteer is who I gratefully appreciate will occasionally interrupted me and say there's a question. I think you're volunteering, you're smiling. Ok. Cool. Awesome.

Ok. Hello there. We're here live in Atlanta with Georgia Tech. Ok everybody who's here live wave at TV Land. Awesome. And let me try to get my camera set up properly. So today I'm going to cover the next project which is to build a market simulator. Now because you're listening to this lecture and because you are here in person I am at the end going to reveal a secret to you that will garner you an extra point on this assignment and only those who watch this video or are here in person will know that secret information. I'm not going to mention it online in any way. You will discover when the autograder starts happening—well when the project is actually graded, not the buffet autograder—that some people will get an extra little bonus point because they paid attention and others won't. And Dave what do you predict will happen then? We'll say it's all Dave's fault.

IN: Orders.csv

OUT: History of value

Ok so the point of today's lecture is to go over this next project, with this create a market simulator. I'm going to call up the webpage for that so I can consult it myself. So I'm looking at the wiki for MC2-Project 1 and I'm going to talk about some of the aspects of it. The key things are for your assignment is what comes in is an orders file and what comes out is a history of value. So what that means is you're going to write a function and one of the arguments to that function is going to be the name of a file. You shouldn't hardcode anything about relative, direct reason, so on. You should just directly take that file name and read it in. You are allowed in this case to use the Pandas ReadCSV, a function, and we will provide you a template that uses that and reads it in correctly. So don't muck around, don't hardcode in the filename or whatever. Your function should directly take the name of the file. Don't worry about whether it's CSV or txt. Just take in the name, read the data in, then operate on it and I'm gonna talk about how to operate on it and your function should return a pandas data frame that has its index by dates and whose data column includes the value of the portfolio.

Date, Symbol, Order, Shares

Okay I'm going to cover picture how to conduct that computation. Let's look first for a moment at what's in this orders dot CSV file. And on the wiki there are specific examples. I'm going to move this

little bit so I can see it better. So this file is the set of rows where each row specifies a date, comma, symbol you're trading on, order (whether it's a buy or a sell) and the number of shares you want to trade on. So I got a specific example here I want to copy down so I get it correctly. So to avoid having to go back and forth so many times I'm gonna make a little bit of this up so may differ from what you see on the wiki. But the dates are all format a year, hyphen, month, day and it's trivial to read that in as a daytime object from a csv. Symbols are just our three or four letter symbols that we we've been using. The order can be a buy or sell. So here's an order. Now it's going to be your job as the simulating the market to essentially execute that order and take care of the accounting to track the value of portfolio once that order is executed. There will be other orders at other days so let's put in another order. I'm going to simplify this to be a hundred so my math will be easier later. Okay my second order has a date a day later, symbol's IBM, sell, 100.

Now does anybody have any questions about that? Does that seem weird? So the question was to zoom in on the whiteboard.

Date, Symbol, Order, Shares

2008-8-1, AAPL, BUY, 100

2008-8-2, IBM, SELL 100

What kind of order? Adj Close

Ok is there a problem for anybody?

>>How can you sell before you buy?

It's the magic of shorting. So if you issue a sell order and you don't own it, that's equivalent to shorting. And we'll talk about the underlying math for this project in the second. Good question. For those in TV land, the question is are they market orders or limit orders. They're a special sort of Balch order. Sorry I'm writing it small here. The answer is they are market on close orders, which means you get the adjusted close price for that day. So all the stuff that we're going to do for this assignment and actually for the others, are assuming adjusted close prices. Now if you look at the data that you have (the stock price data) you'll see that there are closing prices and adjusted closing prices and the difference between those two, we're actually going to talk about in a later section. For now the important thing to remember is use only adjusted close. But the other closing prices are the prices that on that date the market actually closed at. The adjusted ones are adjusted so that it makes our math easier at this time to calculate the increase or decrease in value of the stock if we held it and reinvested the dividends. But good question and we will always assume that we're using market on close prices. Any other questions? Any questions from Piazza land. Everybody happy?

IN: Orders.csv, Initial cash

OUT: History of value

Date, Symbol, Order, Shares

2008-8-1, AAPL, BUY, 100

2008-8-2, IBM, SELL 100

2008-8-10, AAPL, SELL, 100

I'll put one more order up here just for discussion purposes. So an additional order where we sell the shares of Apple that we had bought earlier. Okay now this is a this is an example of your input, and what we want you to output is a history of the value of our portfolio over time. Now a piece of information that you're missing that we will provide is there's another piece of data coming in which is initial cash. So what will happen over time is the initial value of your portfolio is this initial cash, and then we'll go step-by-step and execute these orders and track the value of the portfolio over time. And it'll change plus or minus the initial cash value. Ok Dave pipe up if I get out of line. If I end up forgetting anything it's Dave's fault.

Oh, by the way, who watched the debate last night? Who thinks Hillary won? Who thinks Trump won? Ok so just for the record, most people thought Hillary won. A few people thought Trump won. I take no position. Not allowed. Neither can I comment on evolution or birth control.

Back to the subject at hand. In terms of this history you need to provide a history for every trading day between the first day and the last day of trading inclusive. Ok? So the way you can discover how big this history needs to be is you can scan this file, read it in, find the first day and the last day, and you know then, ok, I have to fill out a data frame that has all those days. Well there's actually a pretty easy way to do that and I'll show you in just a second.

Step 1:

Prices

Start



End

Okay, so these are my recommendations. It's totally okay to deviate from this structure. In fact I expect a number of you will. That's fine but make sure you understand sort of this basic idea before you go forward with that. Anyways, read in a data frame I'll call "prices." So we've provided you already a method that if you give it the symbols and a start date and an end date, it will read in this data for you from the set of csv files that we've provide. So if you do that, so you need to scan this orders file. And by the way you can read this orders file into a data frame as well and then conduct these operations fairly easily. Anyways you wanna get the list of what are the symbols in this file and what's the earliest date and what's the last date. Once you have that information you can call getdata and it will create a data frame for you that includes every single trading day between this and that. And it'll fill in the columns of what the closing prices were for those stockocks for those days.

Step 1:

| Prices | | | | |
|--------|-----------|------|-----|------|
| | Date | AAPL | IBM | CASH |
| Start | 2008-8-1 | 100 | 100 | |
| | 2008-8-2 | 100 | 110 | |
| | | 120 | 110 | |
| End | 2008-8-10 | 130 | 90 | |

So we'll assume that our first date here is 2008-8-1. And so we've got to date. That's our index. And then we've got values for Apple and IBM. And I want you to add, after you read this data in, I want you to add one more column here at the end. We'll call it cash. Ok so when you read it you'll get two columns. I want you to add this additional column cash and what's in these columns is the adjusted close for the relevant days for these stocks. And I will go ahead and fill in the last day here. So we've got rows and columns. And for simplicity I'm going to fill this in not with some real prices but with prices that are convenient for me to do math with. And I'll put in another reference date here which is a 2008-8-2. So let's say that Apple on this day closed at 100. On this day it again closed at 100 and somewhere along here let's say it went up to 120. Continued to be 120 and eventually closed at say 130 on that final day. And I'll give similar prices to IBM. Let's say it was at a hundred dollars here. Let's say 110 this day. We'll say it remained 110 and I'm just making these prices up. And this date eventually closed at 90.

Step 1:**Prices**

| | Date | AAPL | IBM | CASH |
|-------|-----------|------|-----|------|
| Start | 2008-8-1 | 100 | 100 | 1.00 |
| | 2008-8-2 | 100 | 110 | 1.00 |
| | | 120 | 110 | 1.00 |
| End | 2008-8-10 | 130 | 90 | 1.00 |

Ok now we have also this potential asset called cash. What do you think the value of one share of cash out to be? The answer is 1. So we're going to use this column of cash (you'll see in a moment why it's useful to have that column), but we're going to use that column of cash because it's convenient. But it's also convenient to always assume that the value of one share of cash is 1 dollar. So that's that answer. So this is just a data frame called Prices. Now note that once you've read this in, you conveniently now have a data frame that has a row for each date that you need to attend to, and it turns out it also has a column for each value you're going to eventually need to account for.

Step 1:**Copy****Prices**

| | Date | AAPL | IBM | CASH |
|-------|-----------|------|-----|------|
| Start | 2008-8-1 | 100 | 100 | 1.00 |
| | 2008-8-2 | 100 | 110 | 1.00 |
| | | 120 | 110 | 1.00 |
| End | 2008-8-10 | 130 | 90 | 1.00 |

Trades

| | AAPL |
|------|------|
| 2008 | +100 |
| | 0 |
| | 0 |
| | 0 |
| | 0 |
| | 0 |
| | -100 |

So let's take advantage of that and make a copy of it. And I'm going to call this now the "Trades" data frame. So we create this by copying. So what this data frame represents is anytime the number of shares of any of these assets changes, this is this should log that change. So if we buy a hundred shares of something (well was let's say we buy 100 shares of Apple), in the Apple column we should see a +100 on this date. And I'll skip filling in the details, but each time the number of shares that we're holding changes we log that change here.

Now let me finish talking about what the meaning of this data frame is and then I'll talk about how to build it. So we should only see numbers here when that actually changes. So that means we should only see a plus 100 here when we buy apple, and on that very last day we sell apple so we should see a minus 100 here. And all other rows should be 0. So again this is just indicating changes in those values. Now how to build that?

The first step is just fill the whole thing with zeroes right? So you copied it over. You don't want to fill the dates in with zeros. We want to fill in all these columns with zeros. Fill it with zeros and then populate it step-by-step. Well okay here's how to populate it. Come back to your orders file and just step through it one at a time. So first entry in our orders file is buy 100 shares of Apple. So I've already filled that part in here. I left out one part though. Can anybody guess what that is? Well, I lose cash if I buy shares, right? So don't forget our cash column here. So my cash column on that particular row should be negative to represent that I'm spending dollars to buy those shares. Now what should the value be here? Not just the number, but how should I calculate what number should be there?

Step 1:

Copy

| Prices | | Trades | | |
|--------|-----------|--------|-----|------|
| | Date | AAPL | IBM | CASH |
| Start | 2008-8-1 | 100 | 100 | 1.00 |
| | 2008-8-2 | 100 | 110 | 1.00 |
| | | | | |
| | | 120 | 110 | 1.00 |
| | | | | |
| End | 2008-8-10 | 130 | 90 | 1.00 |

| | AAPL | IBM | CASH |
|------|------|------|--------|
| 2008 | +100 | 0 | -10000 |
| | 0 | -100 | 11000 |
| | 0 | 0 | 0 |
| | 0 | 0 | 0 |
| | 0 | 0 | 0 |
| | 0 | 0 | 0 |
| | -100 | 0 | +13000 |

Let me come frame the question a different way. On August first 2008 how much did a hundred shares of Apple cost? It's this times this right? For the next one I'm going to ask a member of the audience that seems to be paying the least attention. So that is negative 10,000 right? My next order, well I just totally skip everything. So let's suppose we went for some number of days without trading. Anyways we'd look at the next row in our orders file and kaboom, it's a sell a hundred shares of IBM. So we have a negative 100 here. What is the value that goes here? So number of shares times the price on that day. So that's 11,000 but because you shorted, you get cash, right?

On the following day what goes here for IBM? On the August 3rd, 2008, anybody hazard a guess? Zero, yeah? Was that a trick question? Did my cash change at all? How about on August 4th? Ok it's all 0. There's no nothing going on until August 10th. Ok so let's look at August tenth. So August tenth we're doing something with Apple. We're selling a hundred shares so what goes here? Yes, minus 100. And what goes here? How much does their cash change if we sell a hundred shares? Ok 13,000, and is it a plus thirteen thousand or minus 13 thousand? Plus, right. Ok so now we have what I'm calling the trades data frame that represents on each day how much did the value of that asset change? We haven't said what the total number is on any particular day. This is just the changes and we built that from our orders file.

Holdings

| | Date | AAPL | IBM | CASH |
|-------|------|------|-----|---------------|
| Start | | 0 | 0 | Starting Cash |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| End | | | | |

Ok so the next data frame that we're going to build, I'm going to call this holdings. And what that represents is on each day how much of each individual asset was I holding? Ok an important part of this data frame is the first row on the first day. So, essentially initialize this with all zeros except we're alerted as to how much cash we have on the first day. And that's part of the defined API. It tells you what your starting cash is. So this is starting cash.

And also on that first day you're not holding any stocks. All you've got is that cash. And this one again, you know the way we create that initially is just by a copy, but we initialize it by filling it with zeros and

adding the cash here at the top. Ok let's say the cash we start with is 15,000. Now we've got this initialized all zeros except for 15,000 there. What's the first step? What do I do next to begin filling this in?

What I want this to represent is, I want it to be filled in with values so that at the end of any particular day, I can see how many shares of each asset I own and how much cash I have.

>> For each day you combine the previous day and current day from trades.

Right. And this first day is a little bit special in that regard and there's different ways you can handle it. You can pretend that there's like one day before that's not part of your data frame. But let me restate what he said. For each day in your trades data frame you combine the day before plus the current day for the trades file.

Holdings

| | Date | AAPL | IBM | CASH |
|-------|------|------|------|-----------------------|
| Start | | 100 | 0 | 15000 5000 |
| | | 100 | -100 | 16000 |
| | | 100 | -100 | 16000 |
| | | 100 | -100 | 16000 |
| End | | 0 | -100 | 29000 |

So treating this first day little bit special, I will do the first day. So a first day for Apple is we get a hundred shares. So we add that to the day before which was zero, so a hundred shares. IBM nothing changes its still zero. Cash we had 15,000 the day before but we lost 10,000 because we invested that in IBM so that changes this to be only 5,000. So after processing this first row we've got August first 2008, 100 shares of Apple, 0 of IBM and 5000 cash.

Now I'm going to pick somebody to tell me what the second row should be? What should be here for Apple the second day? So the first day it was a hundred. We bought 100 shares so we own 100 shares. The second day there are no trades with Apple so how many shares of Apple would I be holding? 100. That's right. What about IBM? Minus a hundred because we shorted. Ok and what should our cash be? So we had on this day 5,000 (the day before). We sold a hundred shares of IBM and got a thousand dollars, so how much cash then do we have on this day? 16,000. Right. Good job. Anyways we just

repeat that. Each day we take the corresponding row in our trades data frame, we add it to the day before in our holdings and that is then the next row.

What should our row be here with them on the next day? Ok in fact it's going to be exactly the same all the way until this last day right? So just before the last day we'll have 100, negative 100, 16,000 and then the very last day we sell a hundred shares of the Apple and we get 13,000 dollars. So on this very last day of our holdings we had zero shares of Apple. We still have negative hundred shares of IBM and we get 13 thousand. So now we have 29,000.

That's pretty incredible we started with only \$15,000 we ended up with 29,000. That's pretty remarkable. Ah, we owe somebody something. What do we owe them? What's that worth? So the value of your portfolio isn't just how much cash you've got, it's affected also by the value of the shares that you hold. Everybody follow that?

Ok so now we have an account. What this project really is, is accounting right? You have to bear with a little bit of accounting here because that ends up being a big portion of what training is about. But anyways the value of your portfolio on any particular day is a combination of the value of the shares that you hold and the cash that you hold.

| Values | | | | |
|--------|------|-------|-------|-------|
| | Date | AAPL | IBM | CASH |
| Start | | 10000 | 0 | 5000 |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| End | | | -9000 | 29000 |

I'm going to call this last data frame "values." And again we initialize the data frame by copying from the previous one because it's already got the dimension that we want right? In the end we want to calculate the value for each day. So if we make the copy again we get a row for each day and a column for each asset. And what we want this data frame to represent is, instead of each column representing the number of shares of the asset that we hold, we wanted it to represent the value in dollars of the asset. Now who can complete the sentence? It says that values equals something times something.

Remember in pandas, asterisk is a cell by cell multiplication. Yes, prices times holdings. So what that means is on this day we're converting a hundred shares of Apple into a hundred times a hundred dollars, so we get to 10,000 here. Now conveniently, we've decided that a dollar's worth a dollar. So this column is prefilled with ones. So when we multiply, we do this multiplication, we're just multiplying 1 times what this column is, so that ends up being five thousand.

So we got Apple, IBM, and cash. And on this day, we've got zero IBM. But anyways, each row is filled in accordingly and we'll just skip to this last row. <Lose connection.>

We're back. Where we were. We wanted to calculate the value of each asset on each day and the simple answer is its prices times holding. That will fill in every single day, if we had shares it converts it into dollars, if we have dollars because they're already initialized as one's. It stays dollars and I'll skip down here to the end. On this last day we have minus a hundred shares of IBM and on that day IBM was valued at 90. So we have minus nine thousand in IBM on the this day. And 29,000 here.

The output, what we're supposed to finally provide is day-by-day what's the total value our portfolio right? So on this day how could we compute the value of our portfolio for that day? Add the columns so this plus this plus this. It's \$15,000. Now one thing to note is no matter what you buy or sell on the first day, this value should always be equal to your starting cash. Because you're exactly trading your cash for the value at the close of market that day so this always ends up being essentially what you're starting cash is.

Anyways each day is simply the sum of these. So this is essentially computed by "sum" and "axes" equals what? Is it zero? Is it one? Is it three? What would it mean if I were to say some axis equals 0? Ok so axis 0 is by row right? Axis one is across the columns. So the answer here is one. And then you will simply get a single column data frame where the value is the sum of those columns. And in this case its twenty thousand dollars on that last day. So through clever trading we made \$5,000.

Ok, that's it. There's more coming but any questions on that. So the question is can there be a situation where the cash column is not one? Not for this assignment. Where that might make sense is let's say you were working with foreign exchange, trading currencies or something. That might not necessarily be one. But for the purposes of this assignment assume it's always one.

How do you go from the orders file to the trades file without a for loop? The answer is I don't think you can. I think you have to do a for loop for this. So my mantra all semesters been no for loop. Except for that first step you can go without for loops, but that first step indeed you're going to need a for loop.

2011-6-15 +1 point

Here is the secret. On that date which is June 15, 2011, ignore all orders. We're going to give you a test case when we actually grade this with an order on that date. And if you execute that order, you don't get the secret point. If you don't execute that order, you do get the secret point. And that is your extra benefit for coming to class in person or bearing through my video.

There's one more little secret that I'll tell you, but I'm going to tell that at the very end. So you're gonna have to sit through the drudgery all the way to the very end to get the final secret. Any questions so far?

>>Why that date?

It's my wife's birthday but she wasn't born that year.

>>Will the autograder take that into account?

So the Buffett auto grader, the test cases we check on Buffett won't check that. It will only be the final auto grader when you hand it in. The test cases that we give you, we'll adjust them so there's no trades on that date. There may be holdings on that date but we'll make sure there's no trades on that date except for the final secret hold out test case from when you actually turn your code in. And this is the last it will be mentioned.

Ok. Important topic here for the last seven minutes. Leverage. remember this one more secret, so hang around.

What's leverage?

>>It's an account that lets you borrow from a broker.

Yes, so in order to use leverage, you need to be able to borrow. But you're not always using leverage just because you can. Anyways, here's a intuitive definition of leverage. How much you've got invested in the market divided by the liquidation value of your account. So put another way, brokers will allow you...you can deposit a hundred thousand dollars (if you have a hundred thousand dollars)...you can deposit a hundred thousand dollars interactive brokers and then proceeded to buy seven hundred thousand dollars worth of stock. And in fact if you are Goldman Sachs, back in 2008, you could buy three million dollars of stocks after having only deposited a hundred thousand dollars. Now that's crazy right? Ok but anyways, it is fairly common, in fact just standard, that you can deposit a hundred thousand dollars and then by two hundred thousand dollars worth of stock. Anybody can do that. The broker is taking a little bit of a risk on you in that case, but they consider it not to be that risky because they can look at your account and see, oh he's got two hundred thousand dollars worth of IBM. IBM is not going to go to 0 tomorrow. We know where he lives. We can collect those dollars that we lent him.

Here's something you could do and I'll show you why your broker won't let you do it. But let's suppose we have zero dollars in our account and we short a hundred thousand dollars of IBM. What do we have after that? Well let's say IBM is selling at a hundred dollars right? So if we short a hundred thousand dollars worth of IBM, we have negative 1,000 IBM right? But we have suddenly a hundred thousand dollars in cash. So poof, out of nothing we created a hundred thousand dollars. We could then proceed to buy, to spend our hundred thousand dollars on Google. And now we've got a hundred thousand dollars worth of Google, zero cash, but we are short IBM. It sounds a little bit sketchy right? Indeed it is sketchy, and here is the equation that your broker uses to prevent you from doing it.

$$\text{Leverage} = \frac{\sum |\text{investments}|}{\sum \text{investments} + \text{cash}}$$

Ok so I'm going to define leverage as the sum of the absolute value of your investments. So notice absolute value, which means if you're short, this turns out to be a positive number. Divided by the sum of your investments plus cash.

So this is the value of your portfolio if you liquidated it. This is how much long or short you've got in the market. And the typical broker requirement is that this value must be less than or equal to 2.0. If you go over that, they will, without telling you, liquidate some of your positions. They'll sell your IBM or force you to buy back your shorts or whatever. But they will not at the end of the market close, they won't let you exceed that value. But what that means is essentially you can deposit a hundred thousand dollars, borrow another hundred thousand dollars, and buy two hundred thousand dollars worth of stock.

Okay before I let you go I have two minutes. Here is the last secret. Previous versions of this project have stated that if a trade causes your leverage to accede 2.0, you have to discard that trade. We're changing it for this project to be 3.0 and that actually has been published on the wiki. But you know what, the people who copy last semester's solutions will still be checking for 2.0, and we're going to throw in a trade that causes 2.5 leverage that should go through. And because all of you know it should really be three, will sail through no problem. But the people who copy somebody else's code from last semester will explode and will cause us to apply additional scrutiny to their submission.

Any questions? So the question is, is there when you borrow this money, is there an interest for commission? Yes we're ignoring that for the purpose of this class but there are multiple ways in which you might incur interest. One is if short a stock, you actually have to pay interest on the value of the stock that you shorted and if you borrow money to buy more stock, you have to pay interest on that borrowed money. Anyways that's it for today. Thanks for paying attention. To those of you who I picked on, good job. Please come back again. I will cut you some slack next time, but you all were good sports. Ok see you next time. Bye-bye.