**How to Create Molly Price Volume**

1. Go to the directory :  
   C:\Sundeep\Stocks\_Automation\IBD\Meetup\_Groups\Molly\_Price\_Volume\   
   and open an already existing file in the directory, is of the name YYYY-mm-dd-GoogleFinance-Price-Volume.xlsx. For e.g. open the file, 2023-07-13-GoogleFinance-Price-Volume.xlsx
2. After opening, first thing to do is to “Save As” the file with the new date. For e.g. save the 2023-07-13-GoogleFinance-Price-Volume.xlsx to **2023-07-15-GoogleFinance-Price-Volume.xlsx**, or whatever date you want.
3. Remove all the existing data in the file from both the sheets in the xlsx (Price and Vol). Make sure you are on Cell A1 in both the sheets.
4. Go to googlesheets and copy the data from the respective sheets into the newly names xlsx respective sheets. Before doing this step, Sundeep assumes that you have the names of the tickers that you want to download the data and have already pasted them in the google sheet.
5. Before and once the data has been downloaded, make sure that the header row has the correct dates (this ensures the freshness of the data). Note : Google Finance only updates the data till the previous day. This works great, if we update the sheets on the weekend. If not, this is something to be aware of. Note 2: Google Finance only provides “Actual Close” for the Price and NOT the “Adjusted Close”. Again something that one needs to be aware of.
6. Once the data has been downloaded, if so desired, the data can be verified. Maybe go to Yahoo and verify a ticker for it’s price and volume. The prices should be accurate, but the volume might differ slightly.
7. Once the cursory verification of the data is complete, then go the python script **SC\_Molly\_Price\_Volume\_Analysis.py** , and modify the following line to update the **date** in the string.   
   price\_vol\_file = "**2023-07-13**-GoogleFinance-Price-Volume.xlsx" 🡨 Update that date to whatever is the date that you used while you downloaded the data in the previous steps.
8. One done, run the python script. Sometimes the script will give out this error when run:  
     
   Traceback (most recent call last):

File "C:/Sundeep/Stocks\_Automation/Scripts/SC\_Molly\_Price\_Volume\_Analysis.py", line 101, in <module>

col\_date\_list\_str = [d.strftime('%m/%d/%Y') for d in col\_date\_list\_dt]

File "C:/Sundeep/Stocks\_Automation/Scripts/SC\_Molly\_Price\_Volume\_Analysis.py", line 101, in <listcomp>

col\_date\_list\_str = [d.strftime('%m/%d/%Y') for d in col\_date\_list\_dt]

**AttributeError: 'str' object has no attribute 'strftime'**

1. Fear not, this is just an indication that script is reading more columns while reading the xlsx file and finding some of the dates to be NaN or some empty strings. To fix this, go to the end of the columns in both the sheets and delete say 5 columns. NOTE: These are the empty columns that need to be deleted when all data ends. In my specific example for the file, 2023-07-15-GoogleFinance-Price-Volume.xlsx, I deleted 5 columns after col EJ, which was the last populated column. Then run the script again.
2. Sometimes, but not always, the script can still fail with something like this:  
   root : ERROR

root : ERROR The numbers of Dates 137, is not b/w 132 and 136

root : ERROR Sundeep has, by design, made googlefinance sheet to download 134 columns

root : ERROR Please check and correct and run again. Exiting

root : ERROR

This means that the number of columns (or the number of days for which the data is available is 137 days). This varies from week to week, b/c google finance does not download weekend dates. Since Sundeep is downloading the data for last 200 days in Google Sheet, that 200 calendar days can translate to somewhat varying number of week/work days of the stock market. So, fear not again, just change that number in the script to 137 at line,   
if ((len(col\_date\_list\_str) <= 132) or (len(col\_date\_list\_str) >= **136**)): 🡨 Change to 137. Hopefully, we don’t need to change is many times as we can set a range after doing this a few times and that range should then hold good. Then run the script again

1. Hopefully the script should run fine now 😊 **The script will create two identical files in the same directory.** There is really no need for two identical files, but Sundeep is creating them just in case for now. The script should generate just one file after we have done it a few times. The files are :  
   2023-07-15-Molly-Price-Volume.xlsx and  
   2023-07-15-Molly-Price-Volume\_copy.xlsx
2. These files have the additional tabs for Price Change, 5 dma vol, 21 dma vol and 50 day dma vol.
3. You can open one file to sanity check.
4. Once satisfied, then open the file AAMSC\_Macros.xlsm (in the scripts directory) and edit the macro 🡪 SC\_update\_Molly\_Price\_Vol, this line  
   date\_str = "2023-07-13" to   
   date\_str = "**2023-07-13**" 🡨 To whatever date you have for the generate file above. In my e.g. it needs to be changed to   
   date\_str = "2023-07-15"
5. Save the script and run it. This will find all <> 1% price changes and compare that day’s volume with vol for 5 dma, 21 dma and 50 dma and color code them appropriately. It will also count the number of Accumulation and distribution days and add them a column for them at the front of the data in the Price-Chg sheet. All of this will be saved as the file   
   2023-07-15-Molly-Price-Volume\_final.xlsx  
   in the same directory.
6. Now, you can sort or whatever and enjoy