

Programming Assessment

Sergio Iommi - 08-Aug-2023

Video Demonstration

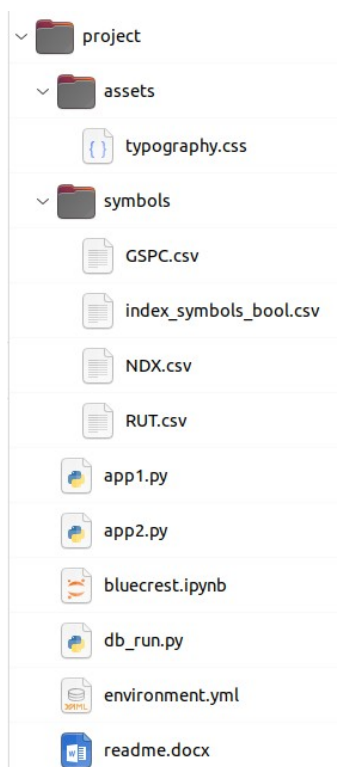
- <https://www.youtube.com/watch?v=nKMXSsmpTvA>

Setup/Install

- For the 2 apps I use MongoDB and Python (with few libraries) so before being able to run the apps in a Jupyter Notebook or as Python scripts, directly from the terminal to open them in a browser, we need to install and configure both.
 - The development and testing for the apps and the database backend have been done under Linux (Ubuntu 22.04.2 LTS) so this guide and steps are the ones I've run on such OS, but hopefully they will work on other OS with no/minor adjustments.
1. **Install MongoDB (locally) - MongoDB Community Edition**
 - I assume to run MongoDB locally with the Community Edition (no cloud) and with the database backend developed as a synchronous Python application (I use PyMongo, the official MongoDB driver for synchronous Python applications).
 - In case we need to run the database backend on the cloud we could use MongoDB Atlas or other cloud services like Amazon AWS, etc.
 - In case we want to develop the database as an asynchronous Python application, we could use the 'motor' official async driver for MongoDB:
 - <https://www.mongodb.com/docs/drivers/motor/>
 - The following commands and notes are for Ubuntu 22.04 and are based on:
 - <https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-ubuntu/>
 - For other OS please check the page:
 - <https://www.mongodb.com/docs/manual/administration/install-community/>
 - **Steps (commands in *italic*)**
 1. `sudo apt-get install gnupg curl`
 2. `curl -fsSL https://pgp.mongodb.com/server-6.0.asc | sudo gpg -o /usr/share/keyrings/mongodb-server-6.0.gpg --dearmor`
 3. Create a list file for MongoDB, i.e., file on local machine where the APT package manager looks for online sources of MongoDB to download and install.
 - Ubuntu 22.04 (Jammy):
 - <https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-ubuntu/#create-a-list-file-for-mongodb>
 - `echo "deb [arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-6.0.gpg] https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/6.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-6.0.list`
 4. Install the latest stable version
 - in case of problems:
 - <https://www.mongodb.com/docs/manual/reference/installation-ubuntu-community-troubleshooting/#std-label-install-ubuntu-troubleshooting>
 - `sudo apt update`
 - `sudo apt-get install -y mongodb-org`
 5. Start running the MongoDB Community Edition process

- <https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-ubuntu/#run-mongodb-community-edition>
 - check init system used by OS
 - `ps --no-headers -o comm 1`
 - `sudo systemctl daemon-reload`
 - `sudo systemctl start mongod`
6. Verify that MongoDB has started successfully
 - `sudo systemctl status mongod`
 7. type 'q' to exit the status log on terminal
 8. (optional) ensure mongodb service auto-start on system reboot
 - `sudo systemctl enable mongod`
 9. (optional) stop mongodb service once the application is no longer needed
 - `sudo systemctl stop mongod`
 10. (optional) check warnings and if default port is 27017
 - `mongosh`
 - exit by typing 'quit'
2. **Install MongoDB Compass (GUI for database)**
 1. This is not needed but it is useful to observe the database content from a GUI
 2. <https://www.mongodb.com/docs/compass/master/install/>
 3. **Python Setup**
 - For Python I assume that Anaconda is already installed on the target machine, otherwise this needs to be done beforehand. Please check the following page in that regard:
 - <https://docs.anaconda.com/free/anaconda/install/index.html>
 - To run the apps, a new conda environment is required, and this can be built using the environment.yml file provided and executing the following command from terminal
 - `conda env create --name py38_bluecrest --file environment.yml`

Jupyter Notebooks, Python Scripts and Folder Structure



The following is the structure of the project folder with the required files to run the apps.

In particular:

- typography.css is a css file that contains some formatting for the Dash components of the 2 apps
- the folder 'assets' and the file 'typography.css' must be kept in this position relative to the other files (notebook bluecrest.ipynb and python scripts app1.py and app2.py)
- notebook bluecrest.ipynb contains the same python code that is split into the 3 python scripts app1.py, app2.py and db_run.py so that the apps can be run either, via the Jupyter Notebook or the *.py scripts via terminal
- environment.yml is the yml file required to create the conda environment
- the folder 'symbols' contains 4 CSV files:
 - GSCP.csv contains the stock symbols for the S&P 500 Stock Index and the related stocks
 - NDX.csv contains the stock symbols for the Nasdaq 100 Stock Index and the related stocks
 - RUT.csv contains the stock symbols for the Russell 2000 Stock Index and the related stocks

- index_symbols_bool.csv is instead a file that is automatically created at the time of creation of the database and it stores all the stock symbols and a column for each stock index to mark the inclusion of each single stock symbol to one or more stock indices (this information is also stored in the database to avoid storing duplicate data if a stock is part of multiple indices)
- regarding the stock symbols I've written a note at the end of this readme document to give details about how I collected the various symbols

How to Create the Database and Run the 2 Apps

1. Once MongoDB is installed in the conda environment *py38_bluecrest* created, we can build the database and run the applications
2. **Database Creation**
 1. We can create the database with either, the Jupyter Notebook *bluecrest.ipynb* or the *db_run.py* script. It must be noted that the first time the code is run (with one of the 2 possibilities, Jupyter Notebook or Python script), the file 'index_symbols_bool.csv' gets created in the folder */projects/symbols* as well as the database 'stock_db' and collection 'daily' in MongoDB, as we can see from the screenshot below. In case we need to re-run the database creation, the 'stock_db' database must be deleted, meanwhile the CSV file will get overwritten.

MongoDB Compass - localhost:27017/stocks_db.daily

Connect Edit View Collection Help

localhost:27017 Documents stocks_db.daily

My Queries Databases Search

admin config local stocks_db daily

stocks_db.daily 19.5k DOCUMENTS 1 INDEXES

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } Explain Reset Find Options

ADD DATA EXPORT DATA 1 - 20 of 19486

	_id ObjectId	Symbol String	Name String	Date Date	Open Double
1	ObjectId('64d241f4f1f42523f...')	"WVE"	"Wave Life Sciences Ltd."	2022-01-03T05:00:00.000+00:00	3.130000114440918
2	ObjectId('64d241f4f1f42523f...')	"WVE"	"Wave Life Sciences Ltd."	2022-01-04T05:00:00.000+00:00	3.2899999618530273
3	ObjectId('64d241f4f1f42523f...')	"WVE"	"Wave Life Sciences Ltd."	2022-01-05T05:00:00.000+00:00	3.25
4	ObjectId('64d241f4f1f42523f...')	"WVE"	"Wave Life Sciences Ltd."	2022-01-06T05:00:00.000+00:00	2.950000047683716
5	ObjectId('64d241f4f1f42523f...')	"WVE"	"Wave Life Sciences Ltd."	2022-01-07T05:00:00.000+00:00	2.9600000381469727
6	ObjectId('64d241f4f1f42523f...')	"WVE"	"Wave Life Sciences Ltd."	2022-01-10T05:00:00.000+00:00	2.880000114440918
7	ObjectId('64d241f4f1f42523f...')	"WVE"	"Wave Life Sciences Ltd."	2022-01-11T05:00:00.000+00:00	2.819999933242798
8	ObjectId('64d241f4f1f42523f...')	"WVE"	"Wave Life Sciences Ltd."	2022-01-12T05:00:00.000+00:00	2.7699999809265137
9	ObjectId('64d241f4f1f42523f...')	"WVE"	"Wave Life Sciences Ltd."	2022-01-13T05:00:00.000+00:00	2.690000057220459
10	ObjectId('64d241f4f1f42523f...')	"WVE"	"Wave Life Sciences Ltd."	2022-01-14T05:00:00.000+00:00	2.619999885559082
11	ObjectId('64d241f4f1f42523f...')	"WVE"	"Wave Life Sciences Ltd."	2022-01-18T05:00:00.000+00:00	2.569999933242798
12	ObjectId('64d241f4f1f42523f...')	"WVE"	"Wave Life Sciences Ltd."	2022-01-19T05:00:00.000+00:00	2.5199999809265137
13	ObjectId('64d241f4f1f42523f...')	"WVE"	"Wave Life Sciences Ltd."	2022-01-20T05:00:00.000+00:00	2.509999990463257
14	ObjectId('64d241f4f1f42523f...')	"WVE"	"Wave Life Sciences Ltd."	2022-01-21T05:00:00.000+00:00	2.3399999141693115
15	ObjectId('64d241f4f1f42523f...')	"WVE"	"Wave Life Sciences Ltd."	2022-01-24T05:00:00.000+00:00	2.069999933242798

_MONGOSH

2. Jupyter Notebook
 1. change kernel to *py38_bluecrest*
 2. to create the database we can run the code in the cell below the one with title 'Create/Update Database (MongoDB)' by first changing some relevant parameters (e.g., start date for the data to get from Yahoo Finance)

3. Python Script (from terminal)

1. Another possibility is to run the script db_run.py from terminal
2. `conda activate py38_bluecrest`
3. `python db_run.py`

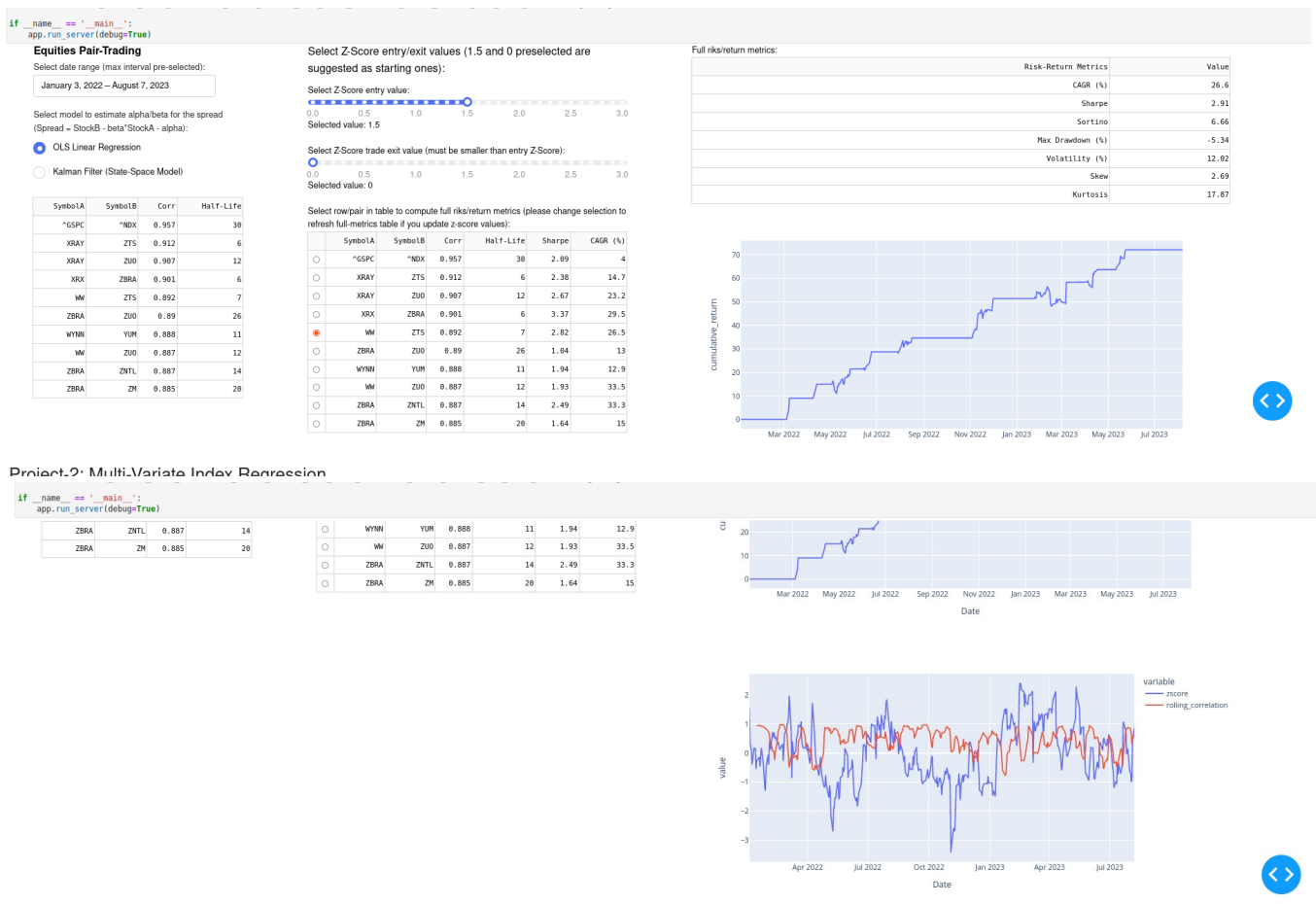
3. Database Update

1. To update the database with the most recent data on Yahoo Finance we can simply re-run the code
2. Another possibility is to setup an automatic process to do that, as an example by using a *cron job* to run daily or even use *celery* (<https://github.com/celery/celery>). I have skipped this step.

4. Running of the Apps

1. Jupyter Notebook

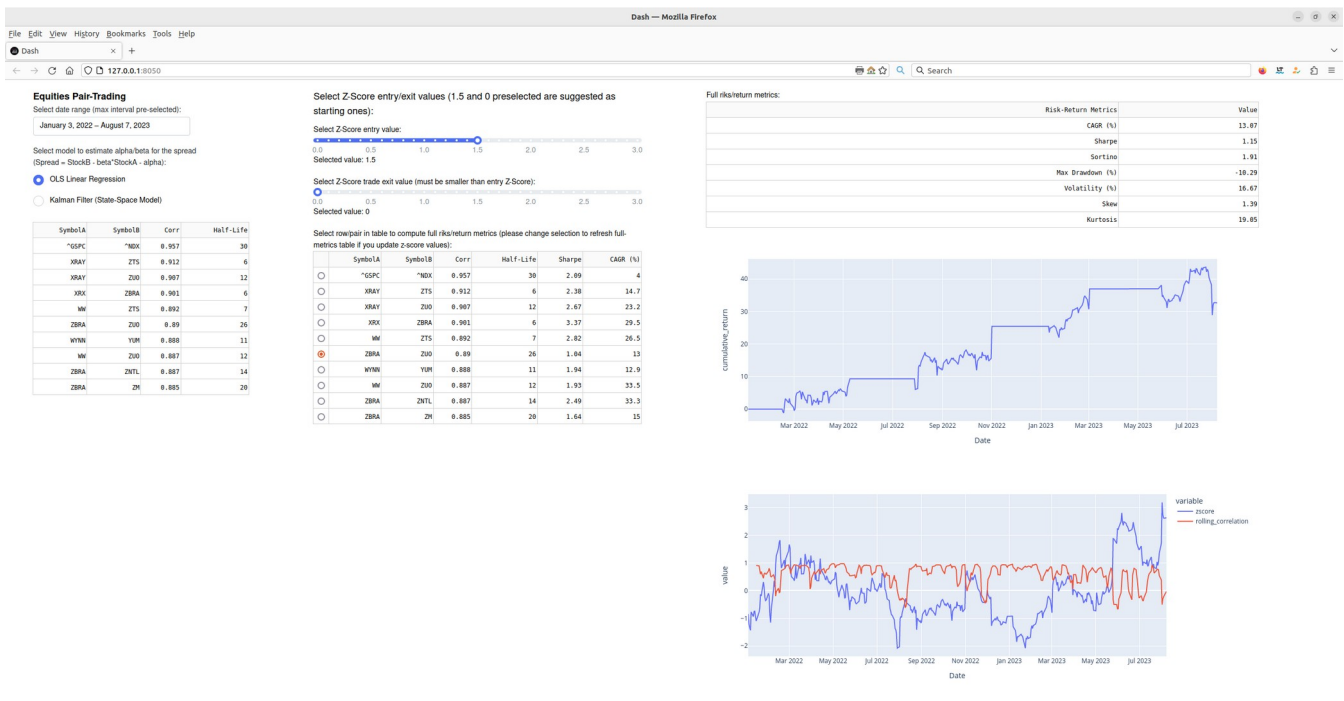
1. Shut down Dash server on 127.0.0.1:8050 (e.g., shut down other kernels using Dash)
2. change kernel to `py38_bluecrest`
3. The 2 apps are stored under the cells:
 1. Project-1: Equities Pairs Trading
 2. Project-2: Multi-Variate Index Regression
4. Once the code is executed the interactive GUI (built with Dash) should appear right below the executed cell
5. The following is how the 1st application (equities pair-trading) should appear

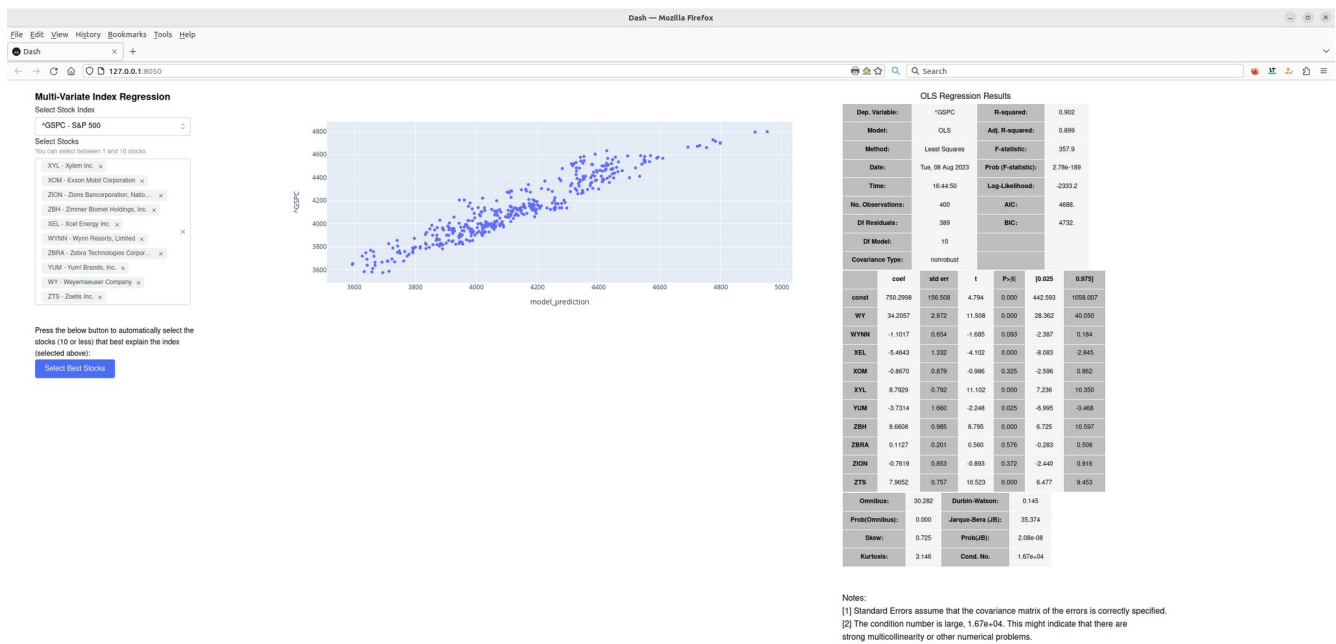


2. Python Script (from terminal, to open the WebApp in a browser)

1. Another possibility is to run the script app1.py (and app2.py) from terminal.
2. Shut down Dash server on 127.0.0.1:8050 (e.g., shut down other kernels using Dash)
3. from terminal move in the folder containing app1.py/app2.py python scripts

- e.g., `cd /home/sergio/work/misc/bluecrest_2023-07/code/`
4. `conda activate py38_bluecrest`
 5. `python app1.py (or python app2.py)`
 1. press CTRL on keyboard and click on link shown on terminal (e.g., <http://127.0.0.1:8050>)
 2. if the browser doesn't open automatically and goes to that page, open a new browser instance and go to address <http://127.0.0.1:8050>
 1. the Dash server should start on the default port 8050, if that port is already used by another process (e.g., Jupyter Notebook that started already the Dash Server), please close such process (e.g., shut down the kernel of the Jupyter Notebook that is running the Dash server) and try to re-run the command in terminal
 3. test the WebApp by using the interactive interface
 4. Make sure to close the app and Dash server with CTRL+C on the terminal before running a different app (e.g., run app2.py after app1.py)
 5. The apps should look like the following if run in the browser:





Note on the Stock Symbols

The list of stocks included in each stock index aren't always the same but in developing the database backend and the 2 apps I assumed a fixed list.

In particular for the S&P 500 and the Nasdaq 100 stock indices I used Wikipedia to get the list of stocks. For the Russell 2000 stock index instead I haven't found a reliable source for the list of stocks so I opted to get the list of symbols by looking at the iShares Russell 2000 ETF (IWM).

Here are all the details for the 3 stock indices:

- **Nasdaq-100**
 - **Wikipedia**
 - <https://en.wikipedia.org/wiki/Nasdaq-100>
 - List of symbols taken from Wikipedia as of 17/07/2023
 - GOOGL and GOOG, two different class of stocks for the same company and it counts as a single company in the index
 - The Wikipedia page provides a detailed description of the changes happened to the index and given the publicity and all the references provided I consider it reliable
 - I saved a snapshot with the WaybackMachine of the Internet Archive:
 - <https://web.archive.org/web/20230730112436/https://en.wikipedia.org/wiki/Nasdaq-100>
 - https://web.archive.org/web/2023000000000000*/https://en.wikipedia.org/wiki/Nasdaq-100
 - **Nasdaq.com website**
 - <https://www.nasdaq.com/market-activity/quotes/nasdaq-ndx-index>
 - <https://archive.ph/jGWsh>
 - Checked on 20/07/2023 and it considers only 88 rows and it misses Google/Alphabet stocks so I don't consider it as a reliable list
- **S&P 500**
 - **Wikipedia**
 - https://en.wikipedia.org/wiki/List_of_S%26P_500_companies
 - I saved a snapshot with the WaybackMachine of the Internet Archive:

- https://web.archive.org/web/20230730114639/https://en.wikipedia.org/wiki/List_of_S&P_500_companies
- Index changes mentioned here:
 - https://en.wikipedia.org/wiki/List_of_S%26P_500_companies#Selected_changes_to_the_list_of_S&P_500_components
 - https://web.archive.org/web/202300000000000*/https://en.wikipedia.org/wiki/List_of_S&P_500_companies
- **Russell 2000**
 - iShares Russell 2000 ETF (IWM) as of 27/07/2023
 - <https://www.ishares.com/us/products/239710/ishares-russell-2000-etf#Holdings>
 - FTSERussell.com website
 - With the Google search “Russell 2000 Membership List site:ftserussell.com” I haven’t found an updated list of all the stocks/symbols but only old ones

Symbols removed from CSVs:

- ADRO
 - Aduro Biotech (present in iShares Russell 2000 ETF; merger with Chinook Therapeutics announced on 02-Jun-2020)
- GTXI
 - GTx, Inc. (present in iShares Russell 2000 ETF, Reverse Merger with Oncternal Therapeutics Completed on 10-Jun-2019)
- MSFUT
 - CASH COLLATERAL MSFUT USD (fund/not stock present in iShares Russell 2000 ETF)
- P5N994
 - Petrocorp Inc Escrow (stock present in iShares Russell 2000 ETF; Petrocorp Inc. changed name into Motomova Inc. in June 2023 but I haven’t been able to confirm if Motomova Inc. is still part of the Russell 2000 stock index)
- PDLI
 - PDL BioPharma, Inc. (stock present in iShares Russell 2000 ETF; delisted on 31-Dec-2020)
- XTSLA
 - BCF Treasury Fund (fund/not stock present in iShares Russell 2000 ETF)