Group #: _____ UID:_____

Final Project Part A

Please title your notebook file in the format SectionY_GroupX_Project_A (where "Y" is your section # and "X" is your group #). Then all the way at the bottom of your notebook, please write the following pledge with all your names:

"We pledge on our honor that we have not given nor received any unauthorized assistance on this assignment to/from other groups. ***Nor have we received any assistance from prior year students.****"*

- *Name 1, Name 2, Name 3, Name 4, Name 5*

Please use markdown cells to number your answers corresponding to the questions and explain your analysis throughout the notebook. The more comments you provide, the better. ChatGPT: If you're using any code-generation tools, please indicate that next to that specific answer. Also provide an explanation of the code generated and your own analysis.

Project Instructions

- Please implement the following analysis in Python and submit your Python Notebook showing your results. Your notebook should be nicely documented with markdown cells.
- Present your results in class using a format similar to the Market Moves presentations. All the graphs and tables in your presentation should be "pretty-printed." Your presentation, which should not exceed 10 minutes, counts for 4 points.
- *Each point is worth 0.5%*.
- After you submit Part A, there will be a small Part B which will involve some additional computations. No presentation – just submission of a second notebook.

1. [2 points] **Write** a "wrapper" function**,** `runMovingAverageAndBB`, which will run the 2 technical trading strategy functions you wrote in 9 HW and then `inner-join` the results. This function takes 5 inputs: (a) a `dataFrame` **with a single instrument's prices** (b) `fastWindow` (c) `slowWindow` (d) `bbWindow` and (e) `stdevBand`. This function then simply calls the two 9 HW functions `movingAverageCrossover` and `bollingerBands` with the supplied input parameters and **then returns a single** `dataFrame` which is an `inner-join` of the outputs of the two functions. The output `dataFrame` should have 5 columns (and the `BMK-MA` and `BMK-BB` columns should be identical). The main objective of this function is to `inner-join` the dates across the daily returns of the benchmark and the three strategies and present them in a single `dataFrame`. **Be sure to `.dropna()` before returning the result.**
2. **Upload** into Python the supplied data file `PricesForProject.csv` which contains (**alphabetically arranged according to their ticker symbols**) close price data from 12/31/1999 through 12/31/2018 for:
   a. 6 equity instruments:
      i. Apple (AAPL), a technology company

ii. Exelon Corp (EXC), a utilities company
iii. General Electric (GE), a diversified high-tech industrial company
iv. Intel (INTC), a technology company
v. Pfizer (PFE), a pharmaceutical company
vi. S&P 500 ETF (SPY), the broad market ETF

b. 2 fixed income instruments:
i. Fidelity Investment Grade Bond Index (FBNDX)
ii. Vanguard Total Bond Market Index (VBTIX)

c. 2 commodity instruments:
i. S&P GSCI Broad Commodity Index (SPGSCI)
ii. Platinum (XPT)

d. 2 currency pairs:
i. Canadian Dollar (CAD)
ii. British Pound (GBP)

3. **Choose**:
   a. Any 5 of the 6 equity instruments
   b. 1 of the 2 fixed income instruments
   c. 1 of the 2 commodity instruments
   d. 1 of the 2 currency pairs

You have to now construct a portfolio of <u>your chosen 8 instruments</u> in such a way so as to get a good **portfolio Sharpe greater than 1.0**. For each instrument, you can:

1. Use the instrument itself: `BMK-MA` (identical to `BMK-BB`) OR
2. Apply the MA crossover strategy which goes "flat" (`MAFlat`) OR
3. Apply the MA crossover strategy which goes "short" (`MAShort`) OR
4. Apply the Bollinger Band strategy (`BB`)

Your choices must comply with the following **portfolio constraints**:

<u>a.</u> For those instruments where you choose *either* MA crossover strategy (either `MAFlat` or `MAShort`), **all your instruments should use the same choice** of (`fastWindow`, `slowWindow`). *This is to avoid curve-fitting bias*.

<u>b.</u> Likewise, for the instruments where you choose the Bollinger Band strategy, **all your instruments** should use **the same choice** of (`bbWindow`, `stdevBand`).

<u>c.</u> For the Bollinger Band strategy, your **`bbWindow` should >= 20** (else the `stdev` estimate will get unstable with too few points).

<u>d.</u> You should **use each technical strategy at least once** – i.e., you cannot simply choose the base instrument for all your 8 choices. (In practice, you'll likely end up applying one of the 3 strategies to most, if not all, of your 8 instruments, because of the Sharpe improvements you'll get.) We'll refer to your 8 choices as your 8 "strategies" (even if you chose just the base-case instrument for some of them).

<u>e.</u> Your **Portfolio Sharpe** should exceed 1.0

<u>f.</u> Your **Portfolio Beta** (to SPY) should be <= 0.5. A negative beta would be even better, but will be difficult, if not impossible, to achieve.

Parameter Choice: You *might* want to run a sensitivity analysis (like you did in 9 HW) to determine a good set of parameters for your chosen instruments. But now you have to

impose the *parameter uniformity constraint* – which might force you to "compromise" on your final choice of parameters. You don't have to do a "brute-force" exhaustive search of the parameter-space to pick the best parameters (although you could!), but spending some time looking at and understanding the "blue regions" of the parameter study may yield a higher portfolio Sharpe.

4. [1 point] **Spell** out your choices of parameters and strategies in a Markdown cell.
5. [1 point] **Write** 5 lines of code clearly and at the top of your notebook **specifying your choices** so that we can `copy+paste` into our grading template. For example:
   a. `maFast = 45`
   b. `maSlow = 220`
   c. `bbWindow = 27`
   d. `bbStdevBand = 4`
   e. **(very important: in alphabetical order of ticker symbol)**
   ```
   myStrategies = ['AAPL-BB', 'EXC-MAFlat', 'FBNDX-BB',
   'GBP-BB', 'GE-BMK-MA', 'INTC-MAShort', 'SPGSCI-
   MAShort', 'SPY-BMK-MA']
   ```
6. [1 point] First, **run** your wrapper function `runMovingAverageAndBB` from #1 above **on all 12 instruments (in alphabetical order) and using your chosen parameters**. You should have 12 `dataFrame`s **with identical dates and 5 appropriately named columns** each. Combine these 12 to create a "master" `dataFrame` with 60 columns.
7. [2 points] **Explain** your rationale for your choices. The aim is to have as high a Sharpe as possible for your final portfolio, **and it should be no less than 1.0**. You can generally achieve that by choosing high-Sharpe components of your portfolio but also by combining strategies that are not highly correlated. Sometimes, a low-Sharpe strategy may improve the overall portfolio Sharpe because of its low correlation with other instruments (the diversification benefit). With the instruments at hand, all equities will likely be highly correlated (although the correlations could potentially drop after application of one of the technical strategies), but the non-equity instruments may exhibit low, or even negative correlations. Please bear in mind that there is no right answer here. I am not looking for a "super-optimized" portfolio; what I'd like to see is your reasoning for what you pick.
8. [1 point] **Index-slice** your master `dataFrame` from # 6 above using your `myStrategies` variable from 5e above to create your "`myStrategies`" `dataFrame` which has the daily returns of **just your 8 chosen strategies in alphabetical order (irrespective of asset class).**
9. [2 points] **Correlation structure**: Calculate the correlation matrix of the `myStrategies dataFrame` you created in #8 and comment on the correlation structure.
10. [1 point] **Weights**: The next step is to determine the instrument weighting scheme. The simple case is **equal weights**. Using the **Matrix formulation for portfolio return and risk**, calculate the annualized Sharpe of an **equal-weighted** portfolio of your chosen 8 strategies. Does the equal-weight portfolio outperform every individual strategy? **You**

get 2 extra credit points if you implement this Sharpe calculation **with a single line of code** by simply applying a bunch of methods to your `myStrategies dataFrame`.

11. [2 points] **MVO:** To see if you can do better than equal weights, run a Mean-Variance optimization with your chosen strategies in order to determine the optimal weights. This step will be identical to the efficient frontier generation you did in 7 HW. Use a **5,000-step Monte Carlo simulation with a seed of 64**. *Be sure to use the alphabetically ordered instruments*. **Assemble** all the different weight vectors so that you can later identify the weights of the minimum-vol and the maximum-Sharpe portfolios. (The extra credit question #12 in 7 HW.)
12. [2 points] **Plot** the risk/return scatter plot of all the 5,000 portfolios, ensuring that the axes are formatted correctly as percentages. **Include** a `colorbar` based on the Sharpe ratio, and use the `reverse colormap` as in 7 HW.
13. [2 points] **Mark** the minimum vol portfolio with a blue star. **Present** the weights of this portfolio and **rationalize** the weight vector, i.e., give a plausible explanation for that outcome of weights. **Construct** a legend for the star, and include in the legend the `x-value` (risk), `y-value` (return) and the Sharpe ratio.
14. [2 points] **Mark** the maximum Sharpe portfolio with a red star. **Present** the weights of this portfolio and **rationalize** the weight vector. **Construct** a legend for the star, and include in the legend the `x-value` (risk), `y-value` (return) and the Sharpe ratio.
15. [1 point] Is the maximum Sharpe higher than the equal weight Sharpe from #10 above?
16. [2 points] To the `myStrategies dataFrame` you created in #8, **add** a 9th column of daily returns which represent the maximum Sharpe portfolio (i.e., it's the weighted returns using the maximum Sharpe weights). Now **display** the performance statistics of this `dataframe` and explain the salient features of this table, i.e., how the final portfolio performance compares with the individual components. (Don't forget to look at the maxDD improvement.)
17. [2 points] **Beta:** Calculate the `beta` of your maximum Sharpe portfolio to the S&P 500. (Use the `SPY-BMK-MA` column from your master `dataFrame` from #6 above.) Is it <= 0.5, as one of the constraints specifies? What does your value of the `beta` imply?
18. [8 points] **Construct** 4 variations of the **equal weighted** portfolio as follows:
    a. All benchmarks from your master `dataFrame` from #6 above, i.e, just the base instruments themselves, without either technical strategy. Note down the Sharpe of this `BMK`-only portfolio.
    b. All `MAFlat` strategies from your master `dataFrame`. Note down the Sharpe of this `MAFlat` strategy portfolio. The lookback parameters should be the same as what you chose in #3.
    c. All `MAShort` strategies from your master `dataFrame`. Note down the Sharpe of this `MAShort` strategy portfolio. The lookback parameters should be the same as what you chose in #3.
    d. All `BollingerBand` strategies from your master `dataFrame`. Note down the Sharpe of this `BollingerBand` strategy portfolio. The lookback parameters should be the same as what you chose in #3.

    Do any of these 4 portfolios outperform your chosen `maxSharpe` portfolio from #14?

19. [4 points] **Present** your findings. Your PowerPoint presentation should answer all the above questions. The presentation itself should be well-formatted, clear and well laid out. Please also pay attention to public speaking aspects, i.e., address the whole class, make eye contact with all sides of the classroom, speak loudly and clearly, and try not to look at your notes too often, etc.
    Good luck!