Notes:

Goals:

Have a streamlit application to drive optimization

Develop a case study to illustrate use of different assets, different portfolio techniques for optimization

See <https://colab.research.google.com/drive/1E4ZFiaKAkogsoezsB2reGFbfHsSraCjU?usp=sharing> for starter code

<https://www.mathworks.com/help/finance/asset-allocation-case-study.html>

<https://mhittesdorf.wordpress.com/2013/06/20/introducing-quantlib-portfolio-optimization/>

Workflow:

1. Explore individual stocks
2. Try different portfolio techniques
3. Compute various metrics
4. Build a report
5. Simulate portfolios
   1. Backtest
   2. VAR,CVAR

Notes (May 8th)

1. Bar graph (in percentages) Top 10 holdings
2. Annualized return to percentage
3. Risk also in percentage 18%
4. Var in another page (<https://docs.streamlit.io/library/get-started/multipage-apps>)
5. Page layout
   1. Home
   2. Exploration
      1. Index Benchmark: DOW
      2. Individual stocks
   3. Portfolio selection (Universe is only DOW - 30 stocks) ; Select a benchmark
   4. Portfolio optimization
      1. Long only
      2. Long-short
      3. Risk aversion
   5. Portfolio analysis
      1. Efficient frontier
      2. Whatifs for different gamma
      3. Benchmark returns vs Portfolio returns
   6. VAR analysis
      1. VAR
      2. CVAR

Todos (June 10)

Use the project structure something like this.. <https://github.com/dataprofessor/st-multipage>

See for additional references - <https://discuss.streamlit.io/t/question-about-project-structure/28867>

1. Home

In this application, we will illustrate these 5 concepts

1. Page Exploration

* Add a description at the beginning orienting users to choose or select options on the left of the screen.
* Provide a line chart for the table . Use plotly for that.. See <https://medium.com/codex/financial-charts-and-visuals-with-plotly-in-python-843ffa9341a9> for samples
* Add company name to the dropdown. Use a csv file something like this (<https://www.dogsofthedow.com/dowcomp.htm>) and read it and dynamically join the ticker and names..

**Outstanding bugs:**

What was done (06/10/2023- 06/11/2023):

* Moved all utils to the util file and import from there, now the “pages” code is much easier to read
* Created a vector of boolean values to check at the beginning if all previous pages have been executed. If yes , then run() the page, if not, a message appears that prompts you to execute the pages sequentially.
* Added plotly graph to exploration page for the ticker
* Added description on top of exploration page

Don’t know how to add full names to tickers but dynamically, so that if a ticker is replaced by another , the procedure automatically repeats.