Ayoub TARSIMI

Aperture Investors

Deliverable Content:

- performance.ipynb : Jupyter notebook producing the required outputs
- excel_time_series_provider.py : class used for parsing time series from an excel workbook
- utils.py utility calc functions
- requirements.txt: list of required external dependencies

Before running the notebook make sure the files are within the same folder

Dependencies:

- Pandas : Data analysis and manipulation tool
- Openpyxl : Allows reading and modifying Excel files

Run **pip install -r requirements.txt** to install the dependencies in your python environment (if you don't already have them).

Outputs:

	nb obs	average geometric relative return	stdev geometric relative return	median geometric relative return	max geometric relative return	min geometric relative return	annualized average geometric relative return	hit ratio	average annualized sharpe ratio (market neutral)
1M	264	0.019742	0.073503	0.019852	0.269550	-0.142125	0.264397	0.602273	1.462362
3M	264	0.062617	0.132258	0.069211	0.419833	-0.326652	0.274990	0.655303	1.138985
6M	252	0.125665	0.185591	0.117268	0.601101	-0.278226	0.267121	0.714286	0.928831

Figure 1: Performance Summary Table

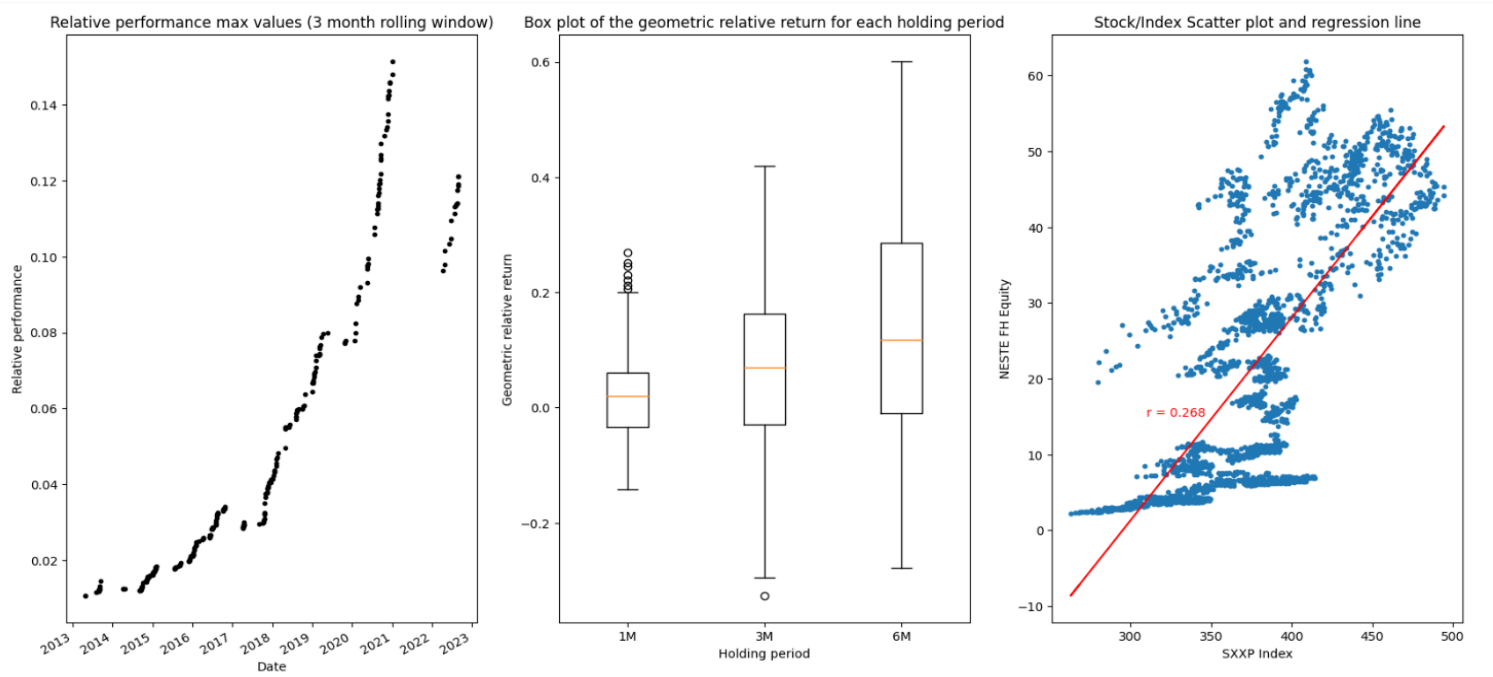

Figure 2: Occurrence chart, relative return box plot and stock/index scatter plot

Definitions:

- **Geometric relative return**: defines how well a stock performs relatively to a benchmark : $R = (Rp_2 - Rp_1)/Rp_1$

  Rp_i = relative performance at date t_i

  can also be defined as $R = (1 + R_s) / (1+R_i)$ with:

  R_s= return of the stock in $[T_1, T_2]$

  R_i = return of the index in $[T_1, T_2]$

- **Annualized Sharpe Ratio**: calculated on 1-day trading periods and assuming 252 trading periods per year (daily returns with 252 business days per year):

  $S = \sqrt{N} * E(R_s - R_i)/\sqrt{\mathrm{Var}(R_s - R_i)}$