**Software Functional Specification for construct.py**

Version: v1.0 Date: 2025-08-18

**1. Overview**

**1.1 Goal**

Provide a single-file CLI tool (construct.py) to backtest and compare six series—SPX, QQQ, Mean-Variance, Minimum-Variance, Equally Weighted, and a DRL-weighted portfolio—under a unified methodology and finance-standard KPIs. Default frequency is Monthly (M).

**1.2 Scope**

Inputs: index/equity prices and four weight tables. Outputs: KPI summary, equity/drawdown time series (csv+png), and charts at three granularities: monthly, quarterly, and yearly, plus optional return-distribution plots.

2. Functional Requirements (FRD)

FRD-1 Inputs & Configuration

- Indices: SPX.csv, QQQ.csv with column 'close'.

- Equities: gvkey, datadate, prccd, ajexdi; adjusted price adj\_close\_q = prccd / ajexdi.

- Weight tables: mean\_weighted\_rf.xlsx, minimum\_weighted\_rf.xlsx, equally\_weighted\_rf.xlsx, drl\_weight.csv.

- All input filenames can be overridden via CLI.

- Non-trading days are forward-filled.

FRD-2 Frequency & Annualization

- Unified annualization via freq: D=252, W=52, M=12, Q=4, Y=1.

- Default freq=M; all KPIs (Sharpe, Sortino, IR, annual return/vol, VaR/CVaR) use the same freq.

- Export charts for M/Q/Y granularities.

FRD-3 Weight Alignment & Constraints

- No short-selling (negative weights → error).

- For each trade\_date, weights must sum to 1; otherwise normalize proportionally and log a warning with original and normalized sums.

- unify\_quarterly\_weights(dict, anchor\_key) aligns trade\_date sets to the anchor strategy.

FRD-4 Portfolio Construction & Costs

- Trades occur only on rebalance dates. Cost = turnover × price × TRANSACTION\_COST.

- TRANSACTION\_COST is a module-level constant defaulting to 0.1% (=0.001) and can be edited at the top of the file.

FRD-5 Return Series & Resampling

- Compute daily strategy returns from holdings and prices; resample to target freq via geometric compounding (prod(1+r)-1).

FRD-6 KPIs (Finance-Standard)

- Absolute: cumulative return, annualized return (mean×n), annualized vol (std×√n).

- Risk-adjusted: Sharpe=(ann\_ret-ann\_rf)/ann\_vol; Sortino uses annualized downside vol (MAR given as annual and converted per period); Calmar=ann\_ret/|MaxDD|; IR=ann\_excess/ann\_tracking\_error based on strategy-minus-benchmark series.

- Risk control: Max Drawdown & Duration (from equity vs rolling max), VaR/CVaR (95/99) by historical simulation on losses.

- IR benchmark defaults to SPX; also support --ir-benchmark QQQ to export IR charts vs QQQ.

FRD-7 Charts & Exports

- For each object (six series), export equity.csv/png and drawdown.csv/png.

- Produce M/Q/Y chart sets; optional return-distribution hist/box plots.

- KPI summary: risk\_metrics\_summary.csv.

FRD-8 CLI

--freq {D,W,M,Q,Y} default M

--rf FLOAT annual risk-free rate, default 0.02

--mar FLOAT annual minimum acceptable return, default 0.0

--cost FLOAT trading cost (ignored; static TRANSACTION\_COST=0.001 in source)

--start/--end date range (YYYY-MM-DD)

--out PATH output dir, default test\_back

--ir-benchmark STR IR benchmark (SPX/QQQ), default SPX; if QQQ is chosen, also export charts vs QQQ

--price-file INDEX/SEC configurable input filenames (index/equities)

--weights-\* configurable filenames for each weight table

3. Non-Functional Requirements (NFR)

- Reproducibility, auditability (verbose logs: params, date range, freq, weight normalization, cost, IR benchmark, output files), performance considerations, code quality (type hints, tests, PEP8).

4. Data Specs

- Indices: columns include at least date, close.

- Equities: gvkey, datadate, prccd, ajexdi; adjusted price adj\_close\_q; pivot to wide by gvkey.

- Weights: gvkey, trade\_date, weight.

5. Methods & Formulae

- Resampling: geometric compounding (prod(1+r)-1).

- Annualization: n=periods[freq]; ann\_ret=mean\*n; ann\_vol=std\*sqrt(n); downside vol similarly annualized.

- Drawdown: equity/cummax - 1; duration is the longest consecutive negative span.

- VaR/CVaR: on loss series (-ret), take 95/99 quantiles and tail means.

6. Pipeline

1) Read & standardize prices (forward fill, align calendar, clip to range);

2) Read weights, validate non-negativity and sum=1; normalize and warn when needed;

3) Align rebalance dates (anchor strategy);

4) Build portfolio (apply cost at rebalance), get daily returns;

5) Resample to target freq for both strategies and benchmarks;

6) Compute KPIs;

7) Export summary, M/Q/Y charts, and time-series csv files;

8) Log run summary.

7. Key Modules / Functions

main(); load\_prices()/load\_index(); load\_weights(); unify\_quarterly\_weights();

normalize\_weights(); build\_portfolio(); resample\_returns(); compute\_drawdown();

compute\_metrics(); save\_timeseries()/save\_histograms(); summarize\_and\_dump().

8. Layout

project/

construct.py

data/

test\_back/

README.md

requirements.txt

9. Acceptance Criteria (Samples)

- risk\_metrics\_summary.csv includes all KPIs for six series;

- 6×3 equity/drawdown chart sets (M/Q/Y) exist;

- With IR benchmark=SPX, IR column present; with --ir-benchmark QQQ, additional charts vs QQQ;

- Logs include “normalize weights” when weight sum ≠ 1;

- TRANSACTION\_COST constant visible at top of source.

10. Configurables

- Input filenames & price columns, freq, rf/mar, IR benchmark, output dir, chart DPI/style.

(End)