

Risk and Performance Measurement, Scenario Generation and the Expected-Return Model

From log-returns to VaR/CVaR, from the covariance estimators to simulated horizon scenarios, and from market capitalisations to Black–Litterman expected returns with Idzorek confidence views, as implemented in `quantcore`

AISE502 – AI-Augmented Portfolio Intelligence Platform

Deterministic quant core, modules `quantcore.measurement` and `quantcore.models`

September 2026

Abstract

This note is the companion of *Mean–CVaR Portfolio Optimisation in Whole Shares* (the file `optimization_model.pdf` in `docs/`). It documents the three model layers that feed the optimiser and the platform’s risk and performance figures: (i) the *measurement* layer — log-returns in CHF, moving-window and exponentially weighted estimators of mean, volatility and covariance, Value-at-Risk and Conditional Value-at-Risk at 95 % and 99 % for a day, a week or a month, and the performance statistics; (ii) the *stock-price models* that turn the latest covariance into a set of horizon scenarios — geometric Brownian motion and the empirical-innovation model $\varepsilon_t = \Sigma_{t-1}^{-1/2} r_t$, $r(j|T) = \Sigma_T^{1/2} \varepsilon_j$; and (iii) the *expected-return model* — market-capitalisation weights, the broad-market average, reverse optimisation, and the Black–Litterman posterior with views whose uncertainty is calibrated from a degree of belief following Idzorek. Every formula is stated with its conventions and defaults, the numerical safeguards (covariance shrinkage, innovation re-standardisation) are motivated with measurements on the platform’s own data set, and the last section maps the mathematics to the functions, parameters and output fields of the code.

Contents

1	Data, conventions and notation	2
2	Returns	3
2.1	Log-returns and the currency conversion	3
2.2	Aggregation over time and over assets	3
2.3	Why log-returns are annualised additively	3
3	Moving-window and exponentially weighted estimators	3
3.1	Moving window	3
3.2	Exponential weighting	4
3.3	Sequences of matrices	4
3.4	Shrinkage towards the diagonal	4
4	Risk measures	5
4.1	Loss, VaR and CVaR	5

4.2	Gaussian (GBM) VaR and CVaR	5
4.3	Historical VaR and CVaR	6
4.4	Discrete scenario sets	6
4.5	Per-asset tables and time series	6
4.6	Portfolio risk in money	6
5	Performance measures	7
6	Stock-price models and scenario generation	7
6.1	Geometric Brownian motion	7
6.2	Empirical innovations (filtered historical simulation)	8
6.3	Sampling, moment matching, the drift convention and importance sampling	9
7	The expected-return model	10
7.1	Market capitalisation, weights and the broad-market average	10
7.2	Reverse optimisation: the equilibrium returns	10
7.3	The Black–Litterman posterior	11
7.4	Idzorek’s confidence: from a degree of belief to Ω	11
7.5	Views from the AI layer: sentiment with a degree of belief	12
7.6	Units and alternatives	12
8	Implementation map	12

1 Data, conventions and notation

The universe consists of N stocks with adjusted closing prices $P_i(t)$ in their listing currency $c(i)$, observed on the *union calendar* of all exchanges (a day on which at least one exchange in the universe trades). A holiday of one exchange is filled with that exchange’s previous close (at most five days), so every trading day of the union calendar has a price for every stock. Prices that are cleaned away (broken adjustments, spike ticks; see docs/METHODOLOGY.md, Section 1) are treated the same way.

FX rates $X_c(t)$ are quoted as CHF per one unit of currency c (so $X_{\text{CHF}} \equiv 1$), and the base currency of everything below is CHF.

Three frequencies are used: *daily* (union calendar, 252 periods per year), *weekly* (Friday closes, 52) and *monthly* (month-end closes, 12); π denotes the number of periods per year of the chosen frequency. A *horizon* is expressed as a number h of periods: at daily frequency a day is $h = 1$, a week $h = 5$, a month $h = 21$; at weekly frequency a week is $h = 1$ and a month $h = 4$.

Throughout, r denotes a log-return, $R = e^r - 1$ the corresponding simple return, and *losses are positive numbers*. Vectors are columns; $\mathbf{1}$ is the vector of ones.

2 Returns

2.1 Log-returns and the currency conversion

The one-period log-return of stock i in its listing currency is

$$r_i(t) = \ln P_i(t) - \ln P_i(t-1). \quad (1)$$

The price in CHF is $P_i^{\text{CHF}}(t) = P_i(t) X_{c(i)}(t)$, hence its log-return is the *sum* of the local return and the FX return,

$$r_i^{\text{CHF}}(t) = r_i(t) + (\ln X_{c(i)}(t) - \ln X_{c(i)}(t-1)), \quad (2)$$

exactly and without any cross term — the reason log-returns are the working unit of the whole platform. (With simple returns the conversion is $(1 + R_i)(1 + R_X) - 1$, which is neither additive nor symmetric.) The implementation converts prices with the FX frame and takes the log-return of the product; (2) is checked by a unit test.

2.2 Aggregation over time and over assets

Under the i.i.d. assumption of the price models (Section 6) log-returns add over time: the h -period return is $r^{(h)}(t) = \sum_{k=0}^{h-1} r(t-k)$. The historical risk measures use *overlapping* h -period sums (more observations, mildly autocorrelated); non-overlapping blocks are available as an option.

Over assets, log-returns do not add. The exact log-return of a portfolio whose value weights w_i are fixed at the start of the period is

$$r_p = \ln \left(1 + \sum_i w_i (e^{r_i} - 1) \right), \quad (3)$$

because simple returns aggregate linearly. A cash residual $1 - \mathbf{1}^\top w$ earns zero. This formula (`portfolio_log_returns`) is used wherever a portfolio return series is needed — the historical portfolio VaR and the portfolio performance statistics.

2.3 Why log-returns are annualised additively

Means of log-returns scale with time: $\mathbb{E}[r^{(h)}] = h\mu$, and under independence $\text{Var}[r^{(h)}] = h\sigma^2$. An annualised log-return is therefore $\pi\mu$, an annualised volatility $\sqrt{\pi}\sigma$; the geometric (compounded) annual return is $e^{\pi\mu} - 1$. Both are reported.

3 Moving-window and exponentially weighted estimators

Every statistic exists in two flavours (`quantcore.measurement.estimators`).

3.1 Moving window

With window length w (default 200 periods) the estimates at date t use the observations $t - w + 1, \dots, t$ with equal weight $1/w$: the sample mean m_t , the sample standard deviation (divisor $w - 1$), the sample covariance

$$\Sigma_t^{\text{roll}} = \frac{1}{w-1} \sum_{k=0}^{w-1} (r_{t-k} - m_t)(r_{t-k} - m_t)^\top. \quad (4)$$

Rows with a missing value are dropped for matrices; a series needs at least $w/2$ observations before a value is reported.

3.2 Exponential weighting

With decay $\lambda \in (0, 1)$ the observation k periods back receives the weight λ^k (RiskMetrics [1]); the weights are normalised over the available history,

$$\omega_k = \frac{\lambda^k}{\sum_{j=0}^{n-1} \lambda^j}, \quad k = 0, \dots, n-1, \quad (5)$$

(pandas `ewm(alpha=1-λ, adjust=True)`), and the estimators are the weighted moments

$$m_t^{\text{ewm}} = \sum_k \omega_k r_{t-k}, \quad \Sigma_t^{\text{ewm}} = \sum_k \omega_k (r_{t-k} - m_t^{\text{ewm}})(r_{t-k} - m_t^{\text{ewm}})^\top, \quad (6)$$

with the diagonal of (6) as the EWM variance (`var(bias=True)`). Both estimators are *centred*: the corresponding mean is subtracted. The effective window is $1/(1-\lambda)$ observations (half-life $\ln 0.5 / \ln \lambda$):

λ	0.94	0.97	0.99	0.997
effective window $1/(1-\lambda)$	17	33	100	333
half-life (periods)	11.2	22.8	69.0	230.7
weight of the observation 100 periods back	0.002	0.048	0.366	0.741

$\lambda = 0.94$ is the RiskMetrics value for daily data and the default of the measurement services (fast adaptation, noisy); $\lambda = 0.99$ is the default of the optimisation covariance (a smoother matrix that still adapts within a quarter). For matrices the history is truncated where the weights fall below 10^{-4} (917 observations for $\lambda = 0.99$).

3.3 Sequences of matrices

The innovation filter of Section 6.2 needs Σ_{t-1} for *every* date t , i.e. the covariance estimated from the data up to and including $t-1$. Storing T matrices of size $N \times N$ is impossible for a thousand stocks, so `covariance_sequence` streams them: the rolling estimator recomputes (4) for each date ($O(wN^2)$ per step), while the exponentially weighted estimator uses the exact recursion of the centred filter with $\alpha = 1 - \lambda$,

$$d_t = r_t - m_{t-1}, \quad m_t = m_{t-1} + \alpha d_t, \quad \Sigma_t = \lambda \Sigma_{t-1} + \alpha d_t (r_t - m_t)^\top, \quad (7)$$

which costs $O(N^2)$ per step and, after a warm-up of $\max(50, 2/(1-\lambda))$ observations initialised with (6), coincides with the normalised-weight estimate up to the factor $1/(1-\lambda^n) \rightarrow 1$ (verified by a unit test at 5 % for $\lambda = 0.97$).

3.4 Shrinkage towards the diagonal

A covariance estimated from n effective observations is ill-conditioned when N is not small against n and singular for $N > n$: the smallest eigenvalues are underestimated, the largest overestimated (the Marchenko–Pastur effect), and anything that inverts the matrix — the innovation filter, mean-variance weights — amplifies the error. The classical remedy is linear shrinkage [2] towards a structured target; the platform shrinks towards the diagonal (the target of Schäfer and Strimmer [3]), which keeps the variances and pulls the correlations towards zero:

$$\Sigma^{(s)} = (1-s) \Sigma + s \text{diag}(\Sigma), \quad s \in [0, 1]. \quad (8)$$

The intensity can be given, or estimated from the data (`shrinkage="auto"`): for centred observations x_k ($k = 1, \dots, n$), products $w_{kij} = x_{ki}x_{kj}$ and sample covariances $s_{ij} = \frac{n}{n-1}\bar{w}_{ij}$, the intensity that minimises the expected squared Frobenius loss of (8) is

$$s^* = \frac{\sum_{i \neq j} \widehat{\text{Var}}(s_{ij})}{\sum_{i \neq j} s_{ij}^2}, \quad \widehat{\text{Var}}(s_{ij}) = \frac{n}{(n-1)^3} \sum_{k=1}^n (w_{kij} - \bar{w}_{ij})^2, \quad (9)$$

clipped to $[0, 1]$. It is small for a long, strongly correlated sample and approaches one when N approaches n — exactly when the sample covariance becomes unreliable. For the EWM estimator the intensity is computed on the last $3/(1-\lambda)$ observations with equal weights. On the platform's data set (daily CHF returns, $\lambda = 0.99$) the automatic intensity is 0.047 for the 100 largest stocks, 0.069 for 300 and 0.114 for the full universe of 745 names. The sequence (7) uses one intensity for all its matrices.

4 Risk measures

4.1 Loss, VaR and CVaR

For a position of value V_0 and a horizon log-return r , the loss in return space is $\ell = -r$ and in money it is $V_0(1 - e^r) = V_0(1 - e^{-\ell})$. The Value-at-Risk at level β is the β -quantile of the loss distribution, the Conditional Value-at-Risk (expected shortfall) the mean of the loss beyond it,

$$\text{VaR}_\beta = q_\beta(\ell), \quad \text{CVaR}_\beta = \frac{1}{1-\beta} \int_\beta^1 q_\gamma(\ell) d\gamma \quad (= \mathbb{E}[\ell \mid \ell \geq \text{VaR}_\beta] \text{ for continuous } \ell), \quad (10)$$

see [4, 5]. The platform reports both at $\beta = 0.95$ and 0.99. Because $\ell \mapsto V_0(1 - e^{-\ell})$ is increasing, quantiles commute with it, and a return-space VaR translates into money exactly as $V_0(1 - e^{-\text{VaR}_\beta})$; the CVaR in money is computed the same way from the return-space CVaR (which is exact for VaR and a very good approximation for CVaR: for a weekly CVaR of 4.6 % the difference between the exact tail mean of $V_0(1 - e^{-\ell})$ and $V_0(1 - e^{-\text{CVaR}})$ is below 0.1 % of V_0 ; the scenario-based portfolio risk of Section 4.6 works in money directly and is exact).

4.2 Gaussian (GBM) VaR and CVaR

If $r \sim N(\mu, \sigma^2)$ then $\ell = -r \sim N(-\mu, \sigma^2)$, its β -quantile is $-\mu + z_\beta \sigma$ with $z_\beta = \Phi^{-1}(\beta)$, and the tail mean follows from $\mathbb{E}[Z \mid Z \geq z_\beta] = \varphi(z_\beta)/(1 - \beta)$ for a standard normal Z :

$$\text{VaR}_\beta = -\mu + z_\beta \sigma, \quad \text{CVaR}_\beta = -\mu + \sigma \frac{\varphi(z_\beta)}{1 - \beta}. \quad (11)$$

The multipliers are $z_{0.95} = 1.645$, $\varphi(z_{0.95})/0.05 = 2.063$, $z_{0.99} = 2.326$, $\varphi(z_{0.99})/0.01 = 2.665$. Under the i.i.d. assumption the horizon parameters are $\mu_h = h\mu$ and $\sigma_h = \sqrt{h}\sigma$; μ and σ come from either estimator of Section 3. The mean is kept (a positive drift lowers the loss quantile one-for-one); for daily data it is negligible, for a month it is not.

Example 4.1 (Reference values). A stock with daily volatility $\sigma = 1\%$ and zero mean, one week ($h = 5$, $\sigma_h = 2.236\%$): $\text{VaR}_{0.95} = 3.678\%$, $\text{CVaR}_{0.95} = 4.612\%$, $\text{VaR}_{0.99} = 5.202\%$, $\text{CVaR}_{0.99} = 5.960\%$. For a position of CHF 100 000 the exact money figures are $\text{VaR}_{0.95} = 3\,611$, $\text{CVaR}_{0.95} = 4\,508$, $\text{VaR}_{0.99} = 5\,069$, $\text{CVaR}_{0.99} = 5\,786$ (the linear approximation $V_0 \cdot \text{CVaR}$ would give 4 612 and 5 960). The daily figures $\text{VaR}_{0.95} = 1.645\%$, $\text{CVaR}_{0.95} = 2.063\%$, $\text{VaR}_{0.99} = 2.326\%$, $\text{CVaR}_{0.99} = 2.665\%$ and the weekly ones are pinned by `tests/test_risk.py`.

4.3 Historical VaR and CVaR

The empirical counterparts replace the Gaussian by the empirical distribution of the realised h -period returns $r^{(h)}$ over a window of $\max(5w, 750)$ periods:

$$\text{VaR}_\beta^{\text{hist}} = -\hat{q}_{1-\beta}(r^{(h)}), \quad \text{CVaR}_\beta^{\text{hist}} = -\frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} r_t^{(h)}, \quad \mathcal{T} = \{t : r_t^{(h)} \leq \hat{q}_{1-\beta}\}, \quad (12)$$

with $\hat{q}$ the empirical quantile (linear interpolation). They capture fat tails and skewness that the Gaussian misses, at the price of noise: at $\beta = 0.99$ with 750 daily observations the tail contains eight points.

4.4 Discrete scenario sets

For S equiprobable losses (a simulated scenario set, Section 6) the definition (10) evaluates exactly: with $k = (1 - \beta)S$ and the losses sorted decreasingly $\ell_{(1)} \geq \ell_{(2)} \geq \dots$,

$$\text{CVaR}_\beta = \frac{1}{k} \left(\sum_{s=1}^{\lfloor k \rfloor} \ell_{(s)} + (k - \lfloor k \rfloor) \ell_{(\lfloor k \rfloor + 1)} \right), \quad \text{VaR}_\beta = \ell_{(\lceil k \rceil)}, \quad (13)$$

which is the value the Rockafellar–Uryasev linear programme of the optimisation returns (function `scenario_cvar`; see Remark 2.3 of the companion note). For $S = 2000$ and $\beta = 0.99$ the tail has $k = 20$ scenarios.

4.5 Per-asset tables and time series

`RiskService.asset_risk` returns, for a list of stocks, a frequency, a horizon, a window w and a decay λ , one row per stock with the annualised rolling and EWM volatilities and, for each confidence level $\beta \in \{0.95, 0.99\}$, the Gaussian VaR and CVaR from the rolling estimator, the same from the EWM estimator, and the historical VaR and CVaR — six figures per level, all in log-return units over the horizon (Table 1). `risk_series` gives the same quantities as time series: Gaussian figures from the rolling / EWM volatility and mean series, historical ones from rolling quantiles and rolling tail means of the one-period returns, scaled by $\sqrt{h}$ for longer horizons (an approximation used only in the series).

4.6 Portfolio risk in money

A portfolio is given in *numbers of shares* n_i plus cash C . With today's CHF prices P_i the value is $V_0 = C + \sum_i P_i n_i$ and the value weights are $w_i = P_i n_i / V_0$ (the cash weight C/V_0 is riskless). Three methods are offered (`portfolio_risk`):

- **normal**: $\sigma_p^2 = w^\top \Sigma w$ with the rolling or EWM covariance, $\mu_p = w^\top m$ with the window mean, then (11) with $h\mu_p$ and $\sqrt{h}\sigma_p$, converted to money;
- **historical**: the realised portfolio return series (3) over the last $\max(5w, 750)$ periods, aggregated to h periods, then (12);
- **scenarios**: money P&L $\Pi_s = V_0 \sum_i w_i (e^{r_{s,i}} - 1)$ for each scenario of a simulated set, then (13) on the losses $-\Pi_s$ — exact in money, and precisely the risk the optimiser constrains.

The result (`PortfolioRisk`) carries the value, the weights, the horizon, the method, the annualised portfolio volatility, and per confidence level the VaR and CVaR in return space and in money. A unit test checks that Gaussian scenarios with the same mean and covariance reproduce the normal figures within sampling error.

5 Performance measures

All statistics (`performance_table`) are computed from one-period log-returns r_t , $t = 1, \dots, T$, at the chosen frequency with π periods per year and the annual CHF risk-free rate r_f (0 by default):

$$\begin{aligned} \text{cumulative return} &= e^{\sum_t r_t} - 1, \quad \text{ann. log-return} = \pi \bar{r}, \quad \text{ann. return} = e^{\pi \bar{r}} - 1, \\ \text{ann. volatility} &= \sqrt{\pi} \hat{\sigma}, \quad \text{Sharpe} = \frac{\pi \bar{r} - r_f}{\sqrt{\pi} \hat{\sigma}}, \quad \text{Calmar} = \frac{e^{\pi \bar{r}} - 1}{-\text{MDD}}, \\ \text{Sortino} &= \frac{\pi \bar{r} - r_f}{\sqrt{\pi \cdot \frac{1}{T} \sum_t \min(r_t, 0)^2}}, \end{aligned} \tag{14}$$

where $\hat{\sigma}$ is the sample standard deviation (divisor $T - 1$) and the maximum drawdown $\text{MDD} = \min_t (V_t / \max_{s \leq t} V_s - 1)$ is taken on the value path $V_t = \exp(\sum_{s \leq t} r_s)$. Skewness and excess kurtosis of r_t , the hit ratio (share of positive periods), the best and the worst period, the number of observations and the start and end dates complete the row.

Rolling versions use the window mean and standard deviation of Section 3 for the annualised return, volatility and Sharpe ratio, plus the cumulative return over the window and the running drawdown; *EWM* versions use the exponentially weighted mean and standard deviation.

For a portfolio in shares the return series is either the *constant-mix* series (3) with today's value weights (the natural benchmark of a rebalanced allocation) or the *buy-and-hold* series $\ln(V_t/V_{t-1})$ with $V_t = \sum_i P_i(t)n_i$ (the share numbers held over the whole history). Both are available; the API's `/performance/portfolio` defaults to constant mix.

6 Stock-price models and scenario generation

The optimiser and the scenario-based portfolio risk need a joint distribution of the horizon log-returns of the candidate stocks. Two models are implemented (`quantcore.models.simulation`); both assume that one-period log-returns are i.i.d. (no autocorrelation), so that an h -period return is the sum of h independent one-period draws, both use the *latest* covariance Σ_T of a `CovarianceModel` (rolling window or EWM, with shrinkage), and both take the expected returns μ from the expected-return model of Section 7 (converted to the log-drift m as described in Section 6.3). The output is an $S \times N$ matrix of horizon log-returns $r_{s,i}$, optionally with scenario probabilities, together with its provenance (model, covariance model and the shrinkage actually applied, sampling scheme, seed, horizon, date, a fingerprint of the matrix), the `ScenarioSet`.

6.1 Geometric Brownian motion

Under multivariate GBM the horizon log-return is multivariate normal,

$$r^{(h)} \sim N(hm, h\Sigma_T), \tag{15}$$

sampled as $r_s = hm + \sqrt{h} Az_s$ with $z_s \sim N(0, I)$ and $AA^\top = \Sigma_T$ (the eigen-root $V\Lambda^{1/2}$ with eigenvalues descending by default, so that the first coordinate of z_s drives the market factor; the symmetric and Cholesky roots give the same distribution). The draws z_s are scrambled Sobol' points passed through Φ^{-1} and moment-matched (Section 6.3), so mean and covariance of the set are exact. GBM is the natural partner of the Gaussian risk figures of Section 4 and of mean-variance thinking; it has no fat tails, no skewness and no tail dependence.

6.2 Empirical innovations (filtered historical simulation)

The second model keeps the shape of the observed return distribution and only rescales it to today's covariance. It is the multivariate form of filtered historical simulation [6].

Filtering. For every date t after the warm-up of the estimator, the return vector is standardised with the covariance known at the end of the previous day,

$$\varepsilon_t = \Sigma_{t-1}^{-1/2} r_t, \quad (16)$$

where Σ_{t-1} comes from the streamed sequence (7) (or the rolling recomputation) and $\Sigma^{-1/2}$ is the *symmetric* inverse root $V\Lambda^{-1/2}V^\top$ from the eigendecomposition $\Sigma = V\Lambda V^\top$ (eigenvalues floored at 10^{-10} of the largest; the Cholesky root is available and used automatically above 250 stocks, where the $O(TN^3)$ eigendecompositions become slow). If the model were exactly right, the innovations would be uncorrelated with unit variance — and, unlike Gaussian noise, they carry the empirical fat tails, skewness and non-linear co-movements of the data.

Resampling. With $A = \Sigma_T^{1/2}$ (symmetric root) and log-drift m , the s -th horizon scenario draws h dates $j_1, \dots, j_h$ with replacement from the innovation history and sets

$$r_s = hm + A \sum_{k=1}^h \tilde{\varepsilon}_{j_k}, \quad (17)$$

where $\tilde{\varepsilon}$ are the centred and re-standardised innovations described next. Since the h draws are independent with identity covariance, $\text{Cov}(r_s) = hAA^\top = h\Sigma_T$, consistent with (15), while the marginal shape of every asset and the joint tail behaviour come from the data. The dates are drawn by scrambled Sobol' points in h dimensions mapped to the ranks of the innovations sorted by their market-factor component (so that every day of the horizon covers the loss distribution evenly), alternatively uniformly ("mc") or balanced (every date equally often); the resampled sums are moment-matched to mean zero and covariance hI (Section 6.3).

Centring and re-standardisation. The innovations are centred so that the scenario mean equals the supplied drift exactly (the drift is the expected-return model's job, not the sample mean's). Their variance, however, is systematically *above one*: any lagged volatility filter is too low right after a shock, so the standardised returns of the following days are too large — the classical reason why filtered historical simulation rescales the residuals before resampling [6]. Measured on the platform's data set (daily CHF returns 2016–2026, EWM $\lambda = 0.99$, automatic shrinkage, symmetric root):

number of stocks N	10	30	100
median variance of the raw innovations	1.11	1.25	1.61
scenario volatility / model volatility without re-standardisation	1.06	1.13	1.28

The growth with N is the conditioning effect of Section 3.4: with an effective window of 100 days and 100 stocks, Σ_{t-1} is nearly singular and its inverse root inflates the directions it underestimates. The implementation therefore rescales the centred innovation sample before resampling: `standardize="diagonal"` (default) divides each asset's innovations by their sample standard deviation, `"full"` whitens them with the inverse root of their sample covariance (only when the history has at least $3N$ dates). After rescaling, the per-asset scenario volatility matches the model volatility within $\pm 2\%$ (diagonal) or $\pm 0.5\%$ (full), and the scenario correlations match the model's with a mean absolute error of 0.02 (diagonal) or 0.004 (full). The diagonal option leaves every

asset's marginal distribution intact and works for any N , which is why it is the default. Finally, innovations beyond ± 8 are winsorised as a numerical guard against the same conditioning effect (about 0.2 % of the values on the data set); this is a safeguard, not a model choice, and it is reported.

6.3 Sampling, moment matching, the drift convention and importance sampling

Why not plain Monte Carlo. The optimiser is a sample-average approximation: it sees the model only through the scenario set, and the sample mean of a plain Monte-Carlo set of S scenarios is off by $\sigma_i/\sqrt{S}$ per asset — for $S = 2000$ and weekly volatilities of 3–11 % that is 3–13 % per year, the size of the expected returns themselves. Measured on the 30 largest stocks, two seeds then give portfolios that differ by 100 % of wealth and a true expected profit 10 % below the optimum. The remedies (all in `quantcore.models.sampling`, derived and measured in the optimisation note) are: scrambled Sobol' points instead of pseudo-random draws (S is rounded up to a power of two); *moment matching*, the affine transform $\tilde{z}_s = C^{-1/2}(z_s - \bar{z})$ of the sample that makes its (probability-weighted) mean and covariance exactly 0 and I , so that every scenario set has mean hm and covariance $h\Sigma_T$ exactly and only the shape of the tail is sampled; and the model expected returns in the optimiser's objective instead of the scenario mean. With these, the turnover between two seeds falls to 4–6 %, the flatness of the optimum.

Drift convention. The expected-return model delivers *arithmetic* expected returns μ (Section 7): reverse optimisation and Black–Litterman are statements about simple returns. The scenarios are generated in log-returns and turned into money with $R = e^r - 1$, whose expectation under normality is $e^{hm+h\sigma^2/2} - 1$. The log-drift is therefore the Itô-corrected

$$m_i = \frac{\ln(1 + h\mu_i)}{h} - \frac{\sigma_i^2}{2}, \quad \text{so that} \quad \mathbb{E}[e^{r_i^{(h)}} - 1] = h\mu_i \quad (18)$$

exactly (`drift_convention="arithmetic"`, the default). Using μ_i itself as the log-drift ("`log`") adds $\sigma_i^2/2$ to the expected simple return: on the data set the median bonus over the 30 largest stocks is 5.4 % per year against a median μ of 8.9 %, and for the most volatile names it is 20–34 % (Micron: $\mu = 16.9\%$ would become 65.6 %) — a strong and unintended tilt towards volatile stocks. The scenario set reports the expected simple return per asset (`expected_simple`) that the optimiser uses.

Importance sampling. Only the $(1 - \beta)S$ scenarios in the tail matter for the CVaR. By default (`importance_sampling="auto"`) the sampling distribution is therefore tilted towards the losses of the *cap-weighted portfolio of the candidate stocks*: its loss direction is $d = -A^\top v / \|A^\top v\|$, and two blocks of a quarter of the scenarios each are shifted along d by $z_{0.95}$ and $z_{0.99}$ (GBM: $z_s \mapsto z_s + c_k d$; empirical model: the dates are drawn with exponentially tilted probabilities $q_t^k \propto \exp(\vartheta_k d^\top \tilde{\varepsilon}_t)$), so that one set covers both usual confidence levels. The shift is a covariance statement: in return space it is $-c_k \Sigma v / \sqrt{v^\top \Sigma v}$, so each asset moves by its covariance with the reference portfolio — a high-beta stock is stressed hard, a negatively correlated one moves up — and, an exponential tilt of a Gaussian being a mean shift, the weighted set still has exactly the model covariance and correlations. Every scenario carries the self-normalised likelihood ratio of the mixture proposal as its probability p_s , bounded by $1/(1 - \sum_k \varepsilon_k)$; the weighted set is moment-matched with these probabilities, the optimiser uses p_s in its CVaR rows and the risk figures of Section 4 use the weighted tail. At equal S the tail noise falls by a factor 2–4; at $\beta = 0.99$ 2048 tilted scenarios are as good as 8192 plain ones. Because the direction comes from the market portfolio rather than from a solution of the optimisation, the scenario set does not depend on the risk limit, the constraints or the confidence level — one set serves a whole family of runs ("`adaptive`" tilts towards a first-pass solution instead, which is sharper at $\beta = 0.99$ but gives up that property).

Which model when. GBM is the transparent baseline; the empirical model is the default of the optimisation because CVaR is a tail measure and the empirical tails are heavier than Gaussian ones (the innovation kurtosis on the data set is well above zero). Both models produce identical second moments by construction, so differences in the optimised portfolios are entirely due to the tail shape.

Cost. The filter needs one eigendecomposition per date: for 100 stocks and 2 500 dates about two seconds, for 745 stocks with the Cholesky root and 1 500 dates about six seconds (four BLAS threads). The innovations depend only on the data and the covariance model, so they can be cached between optimisations with different views or constraints.

7 The expected-return model

The scenario drift μ — and therefore the mean side of the mean-CVaR trade-off — comes from a Black-Litterman model (`quantcore.models.black_litterman`) on *weekly* CHF log-returns over the last ten years. Weekly data reduce the asynchronous-trading noise of the daily union calendar, and ten years give ≈ 520 observations for the long-term covariance. Everything in this section is per week; annual figures are $52\times$ the means and $\sqrt{52}\times$ the volatilities.

7.1 Market capitalisation, weights and the broad-market average

The market capitalisation of stock i in CHF is $M_i(t) = \text{shares outstanding}_i \times P_i(t) \times X_{c(i)}(t)$ with today's shares outstanding held constant back in time. The *cap-weighted broad market index* of the whole universe has the log-return

$$r_M(t) = \ln \left(1 + \sum_i \omega_i(t-1) (e^{r_i(t)} - 1) \right), \quad \omega_i(t-1) = \frac{M_i(t-1)}{\sum_j M_j(t-1)}, \quad (19)$$

i.e. (3) with weights that drift with prices between rebalancings, exactly like an index. Its average over the ten-year window, converted to an arithmetic expectation, $\mu_M = \bar{r}_M + \frac{1}{2} \widehat{\text{Var}}(r_M)$, is the platform's estimate of the expected market return ("the average of a broad market weighted by current capitalisation"); on the data set as of 2026-09-21 the mean log-return is 11.5 % per year with a volatility of 16.2 %, i.e. $\mu_M = 12.9\%$. The current weights $w_i = M_i(T) / \sum_j M_j(T)$, renormalised over the candidate stocks of an optimisation, are the *market weights* of the model.

7.2 Reverse optimisation: the equilibrium returns

A mean-variance investor with risk aversion δ solves $\max_w w^\top \mu - \frac{\delta}{2} w^\top \Sigma w$, whose first-order condition is $\mu = \delta \Sigma w$. Black and Litterman [7, 8] read this backwards: if the market portfolio w is the optimal portfolio of the representative investor, the excess returns that make it optimal are the *implied equilibrium returns*

$$\pi = \delta \Sigma w, \quad \delta = \frac{\mu_M - r_f}{w^\top \Sigma w}, \quad (20)$$

where δ is chosen such that the market portfolio itself earns exactly the market excess return: $w^\top \pi = \delta w^\top \Sigma w = \mu_M - r_f$. Here Σ is the ten-year weekly covariance of the candidate stocks and r_f the weekly CHF risk-free rate (0 by default). The equilibrium returns are proportional to each stock's covariance with the market portfolio — its beta — and are the neutral starting point: an investor without views holds the market. On the data set, with the 100 largest stocks as candidates (their market portfolio has a volatility of 19.7 %), $\delta \approx 3.3$; the equilibrium returns range from about 2.7 % (Unilever, Nestlé — low beta in CHF) to about 23 % (Nvidia) per year.

Remark 7.1. μ_M is the average of the *broad* index of all 745 stocks, whereas w and Σ refer to the candidate subset of an optimisation. This follows the specification (one market-wide average) and keeps δ stable across candidate sets; the alternative of using the candidate portfolio's own average is a one-line change.

7.3 The Black–Litterman posterior

The model treats the (unknown) vector of expected excess returns μ as a random variable with the equilibrium as prior, $\mu \sim N(\pi, \tau\Sigma)$, where the scalar τ (0.05 by default) expresses how uncertain the equilibrium is relative to the return covariance. K views are linear statements about μ with their own uncertainty,

$$P\mu = Q + \varepsilon, \quad \varepsilon \sim N(0, \Omega), \quad \Omega = \text{diag}(\omega_1, \dots, \omega_K), \quad (21)$$

where each row p_k of P ($K \times N$) is a portfolio: $p_k = e_i$ for an *absolute* view ("stock i will return q_k "), $p_k = e_i - e_j$ for a *relative* view ("i will outperform j by q_k "). Combining prior and views by Bayes' rule (equivalently, generalised least squares) gives the posterior distribution of μ with mean and covariance

$$\mu_{\text{BL}} = [(\tau\Sigma)^{-1} + P^\top \Omega^{-1} P]^{-1} [(\tau\Sigma)^{-1} \pi + P^\top \Omega^{-1} Q], \quad \Sigma_\mu = [(\tau\Sigma)^{-1} + P^\top \Omega^{-1} P]^{-1}. \quad (22)$$

The implementation uses the algebraically identical form that needs no inverse of $\tau\Sigma$ (a $K \times K$ system only), which is numerically preferable and also valid when Σ is singular:

$$\mu_{\text{BL}} = \pi + \tau\Sigma P^\top (P\tau\Sigma P^\top + \Omega)^{-1} (Q - P\pi), \quad \Sigma_{\text{BL}} = \Sigma + \tau\Sigma - \tau\Sigma P^\top (P\tau\Sigma P^\top + \Omega)^{-1} P\tau\Sigma. \quad (23)$$

The posterior mean is the prior corrected in the direction of the views by the amount by which they disagree with the prior ($Q - P\pi$), weighted by how confident they are relative to the prior; $\Sigma_{\text{BL}} = \Sigma + \Sigma_\mu$ is the covariance of the returns themselves (return noise plus the remaining uncertainty about the mean) and is reported for completeness. Without views, $\mu_{\text{BL}} = \pi$. The posterior mean is the drift handed to the scenario models: per day, $\mu = \mu_{\text{BL}}^{\text{ann}}/252$ (total return, i.e. excess plus r_f).

7.4 Idzorek's confidence: from a degree of belief to Ω

The weak point of the model is Ω : nobody can state the variance of a view. Idzorek [9] replaces it by a *confidence* $C_k \in (0, 1]$ with a portfolio meaning. Consider one view k alone. At 100 % confidence ($\omega_k = 0$) the posterior is $\mu_{100} = \pi + \tau\Sigma p_k^\top (p_k \tau\Sigma p_k^\top)^{-1} (q_k - p_k \pi)$ and the unconstrained mean-variance weights move from w to $w_{100} = (\delta\Sigma)^{-1} \mu_{100}$. Idzorek defines the confidence as the fraction of that move the investor wants to make: the target weights are $w + C_k (w_{100} - w)$, and ω_k is the value for which the posterior weights $w(\omega_k) = (\delta\Sigma)^{-1} \mu_{\text{BL}}(\omega_k)$ hit the target (in the paper by a one-dimensional least-squares search). For a single view this calibration has a closed form. From (23) with one view,

$$w(\omega_k) - w = (\delta\Sigma)^{-1} \tau\Sigma p_k^\top \frac{q_k - p_k \pi}{p_k \tau\Sigma p_k^\top + \omega_k} = \frac{\tau}{\delta} p_k^\top \frac{q_k - p_k \pi}{p_k \tau\Sigma p_k^\top + \omega_k}, \quad (24)$$

so the tilt is always in the direction $p_k^\top$ and its size relative to the full tilt ($\omega_k = 0$) is $p_k \tau\Sigma p_k^\top / (p_k \tau\Sigma p_k^\top + \omega_k)$. Setting this equal to C_k gives

$$\boxed{\omega_k = \frac{1 - C_k}{C_k} p_k \tau\Sigma p_k^\top} \quad (25)$$

[10]: the view's variance is the prior variance of the view portfolio scaled by the odds against the view; $C_k = 1$ gives $\omega_k = 0$ (the view is imposed), $C_k = 0.5$ gives $\omega_k = p_k \tau \Sigma p_k^\top$ (the He-Litterman choice, available as `omega_method="he_litterman"`), $C_k \rightarrow 0$ removes the view. The implementation uses (25) for every view and then computes the joint posterior (23); the numerical search of the paper is kept as `idzorek_omega_numeric` and a unit test shows that both agree to 10^{-4} . With several views, each ω_k is calibrated for its view alone, as in the paper.

Example 7.2 (Four assets). Annual volatilities (20, 25, 15, 30) %, correlations $\rho_{12} = 0.5, \rho_{13} = 0.3, \rho_{14} = 0.2, \rho_{23} = 0.4, \rho_{24} = 0.1, \rho_{34} = 0.3$, market weights (0.4, 0.3, 0.2, 0.1), market return 7 %, $r_f = 1$ %, $\tau = 0.05$, weekly covariance $\Sigma = \Sigma^{\text{ann}}/52$. Then $w^\top \Sigma^{\text{ann}} w = 0.0250$ (market volatility 15.8 %), $\delta = 0.06/0.0250 = 2.40$, and the equilibrium total returns are $\pi + r_f = (7.36, 8.80, 4.35, 5.50)$ % — the market portfolio earns $0.4 \cdot 7.36 + 0.3 \cdot 8.80 + 0.2 \cdot 4.35 + 0.1 \cdot 5.50 = 7.00$ %. An absolute view "asset 2 returns 15 %" with confidence $C = 0.4$ gives $\omega = \frac{0.6}{0.4} p \tau \Sigma p^\top = 1.5 \times 6.01 \cdot 10^{-5} = 9.01 \cdot 10^{-5}$ (weekly units; the numerical search returns the same value) and the posterior total returns (8.35, 11.28, 4.94, 5.80) %: asset 2 moves 40 % of the way from 8.80 % to the 15 % it would reach at full confidence, and the correlated assets follow. The unconstrained weights move from 0.3 to 0.4656 for asset 2 — exactly $0.3 + 0.4(0.7139 - 0.3)$ — and stay unchanged for the others, as an absolute view on one asset only tilts that asset. The numbers are pinned by `tests/test_black_litterman.py`.

7.5 Views from the AI layer: sentiment with a degree of belief

The platform's news pipeline delivers, per stock, an assessment "positive / negative" with a degree of belief rather than a return forecast. `view_from_sentiment` turns (stock i , sentiment $s \in [-1, 1]$, confidence C) into the absolute view

$$q = \pi_i^{\text{ann}} + r_f^{\text{ann}} + s \cdot \gamma \cdot \sigma_i^{\text{ann}}, \quad \gamma = 0.25 \text{ by default}, \quad (26)$$

i.e. a fully positive assessment claims one quarter of the stock's annual volatility above its equilibrium return (about +5 % for a 20 %-volatility stock), and the degree of belief becomes the Idzorek confidence C . The strength γ is a modelling parameter; the view carries the provenance of the insight that produced it (`source`). On the data set, a sentiment of +0.8 with confidence 0.6 on Nestlé states $q = 6.1$ % against an equilibrium of 2.7 %, calibrates $\omega = 1.8 \cdot 10^{-5}$ and lifts the posterior to 4.7 % per year. Explicit views (`assets, q, confidence`), including relative ones, are accepted alongside.

7.6 Units and alternatives

Views state *annual total* returns; internally they are converted to weekly excess returns ($q/52$ minus the weekly r_f times the sum of the view weights, which vanishes for relative views). Results are reported per week and annualised; the optimiser receives the posterior total return per day, $\mu_{\text{BL}}^{\text{ann}}/252$, as the *arithmetic* expected return, converts it to the Itô-corrected log-drift (18) of (15) and (17), and uses the expected simple return over the horizon, $h\mu_{\text{BL}}^{\text{ann}}/252$, in its objective. Inputs that are log-return means (the index average, the historical means) are converted with $+\frac{1}{2}\sigma^2$. Two simpler models are available for comparison: `expected_returns="historical"` (the ten-year weekly means, converted) and `"zero"` (no drift, pure risk minimisation).

8 Implementation map

Determinism and provenance. Every figure is a deterministic function of the stored data and the stated parameters; simulations take an explicit seed. Every result carries the parameters that produced it (window, λ , shrinkage applied, model, seed, horizon, date, views with their calibrated ω), so that the advisor layer can cite them and a reviewer can recompute them.

References

- [1] J.P. Morgan/Reuters, *RiskMetrics — Technical Document*, 4th edition, New York, 1996.
- [2] O. Ledoit and M. Wolf, *A well-conditioned estimator for large-dimensional covariance matrices*, Journal of Multivariate Analysis 88(2), 365–411, 2004.
- [3] J. Schäfer and K. Strimmer, *A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics*, Statistical Applications in Genetics and Molecular Biology 4(1), Article 32, 2005.
- [4] C. Acerbi and D. Tasche, *On the coherence of expected shortfall*, Journal of Banking & Finance 26(7), 1487–1503, 2002.
- [5] A. J. McNeil, R. Frey and P. Embrechts, *Quantitative Risk Management: Concepts, Techniques and Tools*, revised edition, Princeton University Press, 2015.
- [6] G. Barone-Adesi, K. Giannopoulos and L. Vosper, *VaR without correlations for portfolios of derivative securities*, Journal of Futures Markets 19(5), 583–602, 1999.
- [7] F. Black and R. Litterman, *Global portfolio optimization*, Financial Analysts Journal 48(5), 28–43, 1992.
- [8] G. He and R. Litterman, *The intuition behind Black–Litterman model portfolios*, Goldman Sachs Investment Management Research, 1999.
- [9] T.M. Idzorek, *A step-by-step guide to the Black–Litterman model: incorporating user-specified confidence levels*, working paper, Ibbotson Associates, 2005 (also in S. Satchell (ed.), *Forecasting Expected Returns in the Financial Markets*, 2007).
- [10] J. Walters, *The Black–Litterman model in detail*, working paper (SSRN), 2014.

concept	code	parameters (defaults) / output
log-returns, alignment, frequencies, horizons	<code>measurement.returns:</code> <code>log_returns,</code> <code>align_prices,</code> <code>resample_prices,</code> <code>horizon_periods,</code> <code>horizon_returns,</code> <code>portfolio_log_returns</code>	<code>freq</code> daily weekly monthly; <code>horizon</code> day week month integer
CHF conversion	<code>data.fx.FXConverter;</code> <code>MarketDataService.prices,</code> <code>.returns</code>	<code>currency="base"</code> (CHF) "local"
rolling / EWM series	<code>estimators.rolling_mean,</code> <code>rolling_vol,</code> <code>ewm_mean,</code> <code>ewm_vol</code>	<code>window=200,</code> <code>lam=0.94</code> ; annualised with the periods per year
covariance matrices	<code>estimators.rolling_cov,</code> <code>ewm_cov,</code> <code>covariance,</code> <code>cov_to_corr,</code> <code>covariance_sequence;</code> <code>models.covariance.CovarianceModel</code>	<code>method</code> rolling ewm, <code>window,</code> <code>lam,</code> <code>shrinkage</code> $\in [0, 1]$ or "auto"
VaR / CVaR	<code>risk.normal_var,</code> <code>normal_cvar,</code> <code>historical_var,</code> <code>historical_cvar,</code> <code>scenario_cvar,</code> <code>loss_to_money</code>	$\beta \in \{0.95, 0.99\}$; losses positive; money via $V_0(1 - e^{-\ell})$
per-asset risk table	<code>RiskService.asset_risk</code> → <code>asset_risk_table</code>	<code>columns</code> <code>mean_rolling_ann,</code> <code>vol_rolling_ann,</code> <code>vol_ewm_ann,</code> <code>var95_normal_rolling,</code> <code>cvar95_normal_rolling,</code> <code>var95_normal_ewm,</code> <code>cvar95_normal_ewm,</code> <code>var95_hist,</code> <code>cvar95_hist</code> (and the same for 99), <code>horizon_periods,</code> <code>n_obs,</code> <code>as_of</code> same names, one row per date
risk time series	<code>RiskService.risk_series</code>	holdings (shares), <code>cash,</code> <code>method</code> normal historical scenarios, <code>cov_method</code> ; per level <code>var,</code> <code>cvar,</code> <code>var_money,</code> <code>cvar_money</code>
portfolio risk	<code>RiskService.portfolio_risk</code> → <code>PortfolioRisk</code>	<code>risk_free</code> from the settings; <code>rebalanced=True</code> = constant mix
performance	<code>performance_table,</code> <code>rolling_performance,</code> <code>ewm_performance,</code> <code>PerformanceService</code>	
scenario models	<code>models.simulation.GBMSimulator,</code> <code>EmpiricalInnovationSimulator</code> → <code>ScenarioSet</code>	<code>n_scenarios=4096,</code> <code>seed=42,</code> <code>sampling</code> sobol mc antithetic lhs balanced, <code>moment_match=True,</code> <code>tilt</code> (<code>TiltSpec:</code> direction, shifts, mixtures), <code>drift_convention</code> arithmetic log, <code>root,</code> <code>standardize</code> diagonal full None, <code>clip=8,</code> <code>max_history=2500</code> ; output probabilities, <code>expected_simple,</code> hash Sobol' sizes are powers of two; likelihood ratios bounded by $1/(1 - \sum_k \text{mixture}_k)$
sampling schemes	<code>models.sampling:</code> <code>standard_normal_sample,</code> <code>moment_match,</code> <code>pca_root,</code> <code>tilt_normal_sample,</code> <code>tilted_resample,</code> <code>resample_indices,</code> <code>scenario_hash</code>	
one set per family of runs	<code>OptimizationService.scenario_key,</code> <code>.market_exposure,</code> <code>.clear_scenario_cache;</code> <code>ScenarioSet.with_drift,</code> <code>.shocks,</code> <code>.shock_hash</code>	the key excludes risk limit, confidence, constraints, costs, objective, cash, holdings and the expected-return model
market index and weights	<code>MarketDataService:</code> <code>cap_weighted_index,</code> <code>market_caps,</code> <code>market_weights</code>	<code>shares outstanding</code> $\times$ price $\times$ FX
Black-Litterman	<code>BlackLittermanModel,</code> <code>View,</code> <code>view_from_sentiment,</code> <code>idzorek_omega</code> → <code>BlackLittermanResult</code>	<code>tau=0.05,</code> <code>periods_per_year=52,</code> <code>omega_method</code> <code>idzorek</code> <code>he_litterman</code> ; <code>prior_ann,</code> <code>posterior_ann,</code> <code>omega,</code> <code>delta,</code> <code>market_return_ann</code>
pipeline in the optimiser	<code>OptimizationService:</code> <code>expected_returns,</code> <code>scenarios</code>	<code>bl_years=10,</code> <code>expected_returns</code> <code>black_litterman</code> <code>historical</code> <code>zero,</code> <code>views,</code> <code>covariance</code> = {ewm, 0.99, auto}, <code>scenario_model</code> empirical gbm gaussian_exact, <code>importance_sampling</code> auto market adaptive off, <code>row_generation</code> auto on off
HTTP	<code>/risk/*,</code> <code>/performance/*,</code> <code>/model/expected-returns,</code> <code>/model/scenarios,</code> <code>/market/*</code>	see /docs

Table 1: Where each concept lives in `quantcore`.