Portfolio backtesting in practice

Agenda

1 Returns: theory and facts

2 Portfolio backtesting

3 Portfolio performance metrics

4 Wrap-up

supervised
learning
factor
investing

©Guillaume Coqueret

Returns

Two ways to proceed

Arithmetic returns	**Logarithmic returns**
$r_{t-\Delta t,t} = \frac{P_t}{P_{t-\Delta t}} - 1$	$r_{t-\Delta t,t} = \log\left(\frac{P_t}{P_{t-\Delta t}}\right)$
$P_t = P_0 \prod_{s=1}^{t}(1 + r_{s-1,s})$	$P_t = P_0 \exp\left(\sum_{s=1}^{t} r_{s-1,s}\right)$
Good for cross-section aggregation	Good for time aggregation
Often used in discrete time	Often used in continuous time
ex: portfolio allocation	ex: option pricing

When Δt is sufficiently small so that returns are small too, both definitions are close because $\log(1 + x) = x + o(x)$.

Example of difference: aggregating AR in time: $(1 + r)(1 - r) \neq 1$!
Note: despite their practical importance, we do not include dividends in this lecture.

CS aggregation with arithm. returns

A simple proof

Consider 2 assets, A and B and a portfolio consisting of n^A and n^B units of each. The corresponding weights at time t are $w^A = \frac{n^A P_t^A}{n^A P_t^A + n^B P_t^B}$ and $w^B = \frac{n^B P_t^B}{n^A P_t^A + n^B P_t^B}$.

The total wealth is equal to $W_t = n^A P_t^A + n^B P_t^B$.

Its return is

$$
\begin{aligned}
\frac{W_{t+1}}{W_t} - 1 &= \frac{n^A P_{t+1}^A + n^B P_{t+1}^B}{n^A P_t^A + n^B P_t^B} - 1 \\
&= w^A \frac{P_{t+1}^A}{P_t^A} + w^B \frac{P_{t+1}^B}{P_t^B} - 1 \\
&= w^A \frac{P_{t+1}^A}{P_t^A} + w^B \frac{P_{t+1}^B}{P_t^B} - w^A - w^B \\
&= w^A r_{t,t+1}^A + w^B r_{t,t+1}^B
\end{aligned}
$$

supervised
learning
factor
investing

TS aggregation with log-returns

A simple proof

Take any three dates $s < t < u$,

$$r_{s,u} = \log\left(\frac{P_u}{P_s}\right) = \log\left(\frac{P_u}{P_t}\frac{P_t}{P_s}\right) = \log\left(\frac{P_u}{P_t}\right) + \log\left(\frac{P_t}{P_s}\right)$$
$$= r_{s,t} + r_{t,u}$$

This is why, in continuous time, prices are often exponentials of processes with independent increments (e.g., Brownian motion, Lévy processes)

supervised
learning
factor
investing

TS aggregation: link with CLT

Squeezing the tails

If log-returns are iid, then for a uniform sequence $0 = t_1 < t_2 < \cdots < t_n = t$, $r_t = \sum_{i=1}^{n} r_{t_i}$ and

$$\frac{r_t - n\bar{r}}{\sigma_r \sqrt{n}} \to N(0, 1),$$

so that when n is large,

$$r_t \stackrel{d}{\approx} \sigma_r \sqrt{n} \times N(0, 1) + n\bar{r},$$

i.e., Gaussian models become **good approximations**.

This is why long term returns (e.g., annual returns) are *close* to normally distributed.
NOTE: the CLT holds under quite general assumptions! (iid not necessarily required!).

supervised
learning
factor
investing

Returns in theory

Very often, iid Gaussian

IN THEORY, non-overlapping returns are often assumed independent and their distribution depends only on Δt, μ, σ.

$$r_{t-\Delta t,t} \overset{d}{=} \mathcal{N}\left(\mu\Delta t, \sigma^2\Delta t\right),$$

which (e.g.) matches the **Black-Scholes** model when $\mu = r - \frac{\sigma^2}{2}$.

$\rightarrow$ very tractable in computations!

$\rightarrow$ can be extended to Lévy processes to account for extreme events.

BUT!

supervised
learning
factor
investing

©Guillaume Coqueret

Stylized Facts

Returns in **practice**! See Cont 2001[1]

Empirically, the following properties have been identified

- ▶ **distribution**: returns are heavy-tailed when Δt is small but converge to Gaussian laws when the horizon increases
- ▶ **lack of auto-correlation**: at daily frequencies (and above), returns are only mildly serially correlated
- ▶ **stochastic volatility**: returns exhibit clusters during which their dispersion increases
- ▶ **persisting absolute returns**: relatedly, the autocorrelogram of *absolute* returns is slowly decaying

$\rightarrow$ complex stochastic processes reflect these empirical properties (e.g., combining stochastic volatility to jump processes).
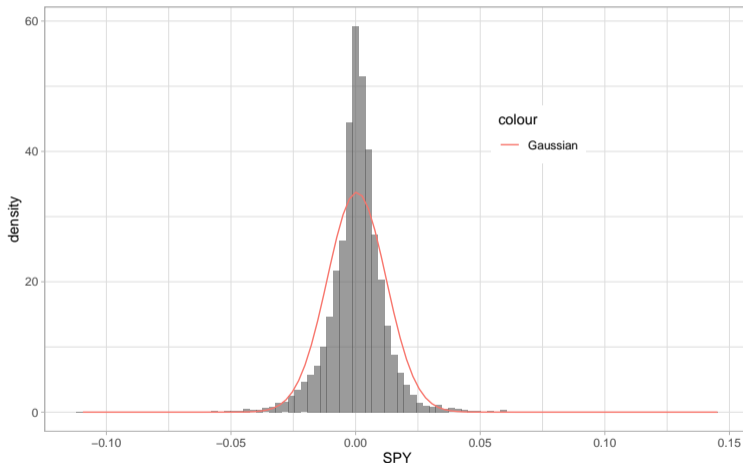
supervised
learning
factor
investing

[1] *Empirical properties of asset returns: stylized facts and statistical issues* - **Quant. Finance**
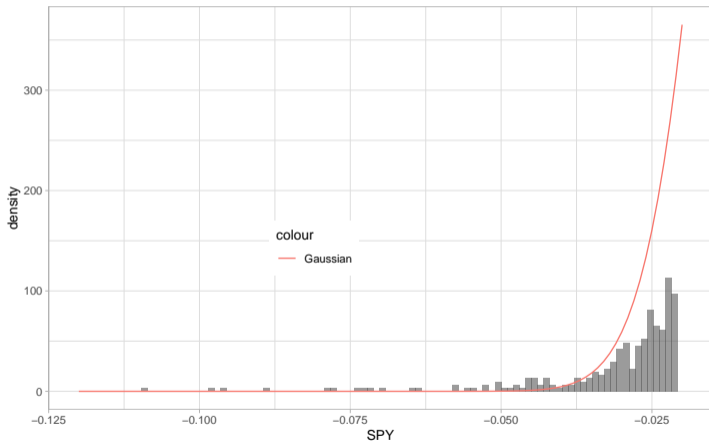
S&P 500 returns (1/3)

Distribution of daily returns Jan 1993 - May 2022 (SPY ETF):

The red curve is the fitted **Gaussian density**.

supervised
learning
factor
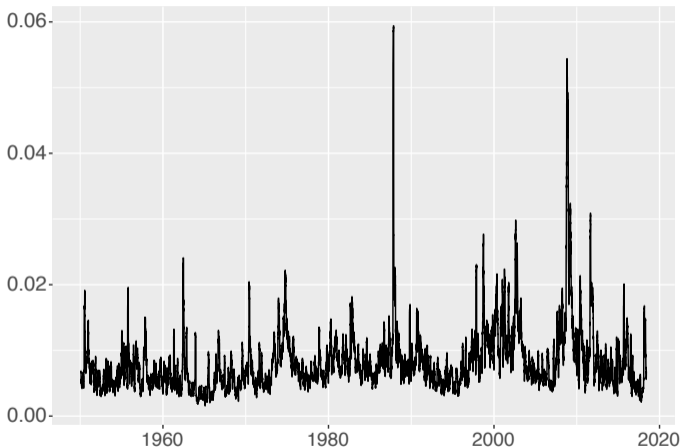investing

S&P 500 returns (2/3) - Tails!

The devil is in the 'de-tails'.

supervised
learning
factor
investing

The tails of empirical returns are much heavier than those of the corresponding Gaussian distribution. **Extreme events are not that rare**!

S&P 500 returns (3/3) - Volatility

Standard deviation over 21 daily returns ($\sim$ monthly).

$\rightarrow$ Empirically, returns **do not seem stationary** over the long term.

supervised
learning
factor
investing

Agenda

1. Returns: theory and facts

2. **Portfolio backtesting**

3. Portfolio performance metrics
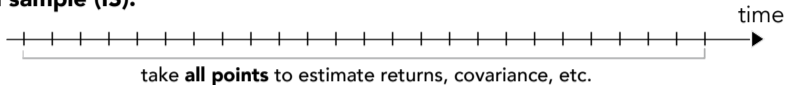
4. Wrap-up

©Guillaume Coqueret
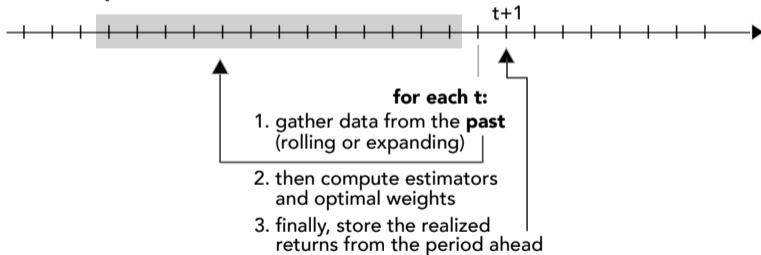
Protocol

User specified rules

- ▶ Investment universe (asset class, geography, sector)
- ▶ Portfolio strategy/strategies:
  - ▶ **strategy type**: agnostic, price-based (ex: chartist), characteristics-based, macro, etc.
  - ▶ **implementation constraints**: leverage, liquidity, factor/region/industry exposure, etc.
  - ▶ possibly, a benchmark for comparison purposes (EW-CW)
- ▶ Backtesting period (usually driven by data and investment universe) & rebalancing frequency (usually driven by policy and transaction costs)
- ▶ Performance metrics (covered below)

In-sample vs Out-of-sample

In sample (IS):

time

take **all points** to estimate returns, covariance, etc.

Out-of-sample (OOS):

t+1

for each t:
1. gather data from the **past** (rolling or expanding)

2. then compute estimators and optimal weights

3. finally, store the realized returns from the period ahead

Big difference between the two! Nobody uses IS anymore...

supervised
learning
factor
investing

Pseudo code form

initialize portfolio weights
initialize portfolio return

for t in (rebalancing_dates){
      1. filter data appropriately (retain **PAST** data)
      2. determine portfolio weights
      3. compute **FUTURE** portfolio returns (with present values)
}

Then, given portfolio weights and portfolio returns,
compute performance metrics

It's as simple as that. To keep things clear: **compartmentalise**!

supervised
learning
factor
investing

Four crucial steps

We must:

1. prepare data (session 4)
2. build ML engine (sessions 3, 5, 6, 7, 8)
3. convert signal (ML output) into investment decision
4. be attentive to perf metrics for both signal (ML) and portfolios (session 7)

The **meta-framing task** is even more important: does what you do make sense economically and technically/statistically?
It does require a bit of expertise (knowledge + experience) to figure this out.

supervised
learning
factor
investing

This stage is incredibly important.

A word on signal processing

Usually, the ML engine will yield forecasts/signals: future returns, profitability probability, confidence score, etc. This must be translated into **portfolio weights**. Several choices are possible:

1. immediate optimisation (Markowitz-like, the output can, e.g., serve as expected return vector)
2. heuristic two-stage construction:
   ▶ start by choosing the assets to invest in (those with positive score, those with score above a threshold (median), etc.)
   ▶ then weight: EW, CW, Inverse vol, Min variance, etc.

+ **constraints** (see next slide)

supervised
learning
factor
investing

Classical constraints

- ▶ (hard): $b_- < w_i < b^+$;
- ▶ turnover, liquidity: $(\boldsymbol{w}_t - \boldsymbol{w}_{t-1})'\boldsymbol{\Lambda}(\boldsymbol{w}_t - \boldsymbol{w}_{t-1}) < \delta$ with $\boldsymbol{\Lambda}$ diag;
- ▶ diversification, $\boldsymbol{w}_t'\boldsymbol{w}_t < \kappa$ (Herfindhal index);
- ▶ factor exposure (proxy), $v_k^- < [\boldsymbol{F}\boldsymbol{w}_t]_k < v_k^+ \rightarrow$ works for industry and geographical exposures as well;
- ▶ etc.

supervised
learning
factor
investing

Agenda

1 Returns: theory and facts

2 Portfolio backtesting

3 Portfolio performance metrics

4 Wrap-up

Notations

Weights and returns

- asset (stock) returns: r_t^n, t for time, n for asset index
- portfolio weights: w_t^n, proportion invested at time t in asset n
- **portfolio returns**: $r_t^P = \sum_{n=1}^{N} w_{t-1}^n r_t^n$ (to state the obvious): cross-sectional aggregation!

Raw performance

Realised returns

We start from the uniform time grid: $t = 1, \ldots, T$ with 1 being the first date of the backtest and T the last one.

We assume that ν periods make one year (e.g. $\nu = 12$ for monthly rebalancing and $\nu = 4$ for quarterly rebalancing)

The full portfolio trajectory is given by $P_t = \prod_{s=1}^{t}(1 + r_t^P)$. Initial value equals one (usually).

▶ Annualized arithmetic returns: $\bar{r}^P = \frac{\nu}{T} \sum_{t=1}^{T} r_t^P$

▶ Annualized geometric returns: $\left(\prod_{s=1}^{T}(1 + r_s^P) \right)^{\nu/T} - 1$

supervised
learning
factor
investing

©Guillaume Coqueret

21 / 27

Risk measures & relative perf.

A large palette of metrics

- Annualised volatility: $\sigma^P = \sqrt{\frac{\nu}{T} \sum_{t=1}^{T} (r_t^P - \bar{r}^P/\nu)^2}$

- VaR at q (often, $q = 0.05$): q quantile of r_t^P: $P[r_t^P \leq \text{VaR}_q^P] = q$

- Maximum Drawdown: maximum decline from a historical peak:

$$MDD^P = \max_{t \in [0,T]} \left(\left(\max_{\tau \in [0,t]} P_\tau \right) - P_t \right)$$

- See also: downwards vol, expected shortfall (CVaR).

Risk adjusted performance:

- **Sharpe ratio**: $\text{SR}^P = \bar{r}^P/\sigma^P$ (originally in excess of riskfree rate version)

- **MAR ratio**: $\text{MAR}^P = \bar{r}^P/MDD^P$ (originally used with geometric returns)

Relative performance: **Information ratio**: $\text{IR} = (\bar{r}^P - \bar{r}^B)/TE(P,B)$, where P=portfolio, B=benchmark and TE is the tracking error: $TE(P,B) = \sqrt{\mathbb{V}[r^P - r^B]}$.

Turnover

Asset rotation is costly

Turnover assesses the changes in the portfolio when rebalancing takes place. A simple proxy is the following:

$$\text{Turnover} \approx \frac{1}{T-1} \sum_{t=2}^{T} \sum_{n=1}^{N} |w_t^n - w_{t-1}^n|,$$

as it requires the portfolio weights only. The *true* turnover is

$$\text{Turnover} = \frac{1}{T-1} \sum_{t=2}^{T} \sum_{n=1}^{N} |w_t^n - w_{t-}^n|,$$

where $t-$ denotes the time **just before** rebalancing: the trajectory of the weight must be computed using the asset returns.

Cap-weighted strategies have very low turnover.

TC-adjusted performance: $SR_{TC} = \frac{\bar{r} - \delta \text{Turnover}}{\sigma}$, where δ (30-50 bps) is a TC constant. **NOTE**: TC could also be evaluated at the asset level: liquid assets have lower TC (that's preferable, but that's harder to do).

supervised
learning
factor
investing

Turnover and drifting weights

Illustration:

Let's consider a 1/N strategy with 2 assets:

time t:	time t+1:	upon rebanlancing
50% in stock A say 10$	stock A increased by 10% its weight is 55% (11/20)	we need to reduce the weight of stock A
50% in stock B say 10$	stock B decreased by 10% its weight is 45% (9/20)	and increase that of stock B

supervised
learning
factor
investing

Agenda

1 Returns: theory and facts

2 Portfolio backtesting

3 Portfolio performance metrics

4 Wrap-up

Portfolios in practice

It's all about returns

- ▶ empirically: heavy-tailed, volatility clusters, etc. (properties are time-varying, processes are only 'locally' stationary).
- ▶ backtesting requires many choices, i.e., many degrees of freedom, though some are dictated by data or policy contingencies.
- ▶ Each step counts and being careful at each stage can make a difference.

supervised
learning
factor
investing

Thank you for your attention

Any questions?

supervised
learning
factor
investing