Backtesting Portfolios

Prof. Daniel P. Palomar

MAFS5310 - Portfolio Optimization with R
MSc in Financial Mathematics
The Hong Kong University of Science and Technology (HKUST)
Fall 2020-21

Outline

Outline

Backtesting

- A backtest is a historical simulation of how a strategy would have performed should it have been run over a past period of time.
- Backtesting is one of the most essential, and yet least understood, techniques in the quant arsenal.
- But beware of backtesting!
- Some interesting quotes about backtesting (Lopez de Prado 2018)[1]:

  "Researching and backtesting is like drinking and driving. Do not research under the influence of a backtest."

  "Most backtests published in journals are flawed, as the result of selection bias on multiple tests."

  "A full book could be written listing all the different errors people make while backtesting."

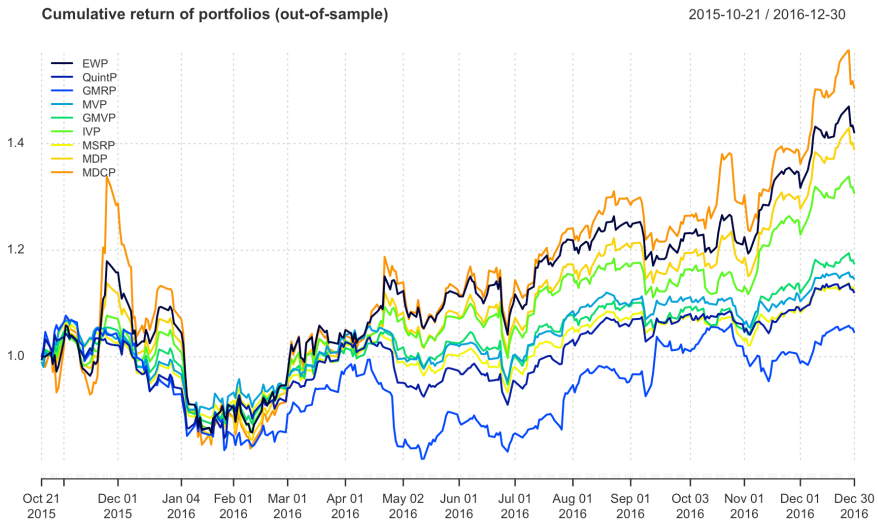
[1] M. Lopez de Prado, *Advances in Financial Machine Learning*. Wiley, 2018.

Backtesting vs. experiments

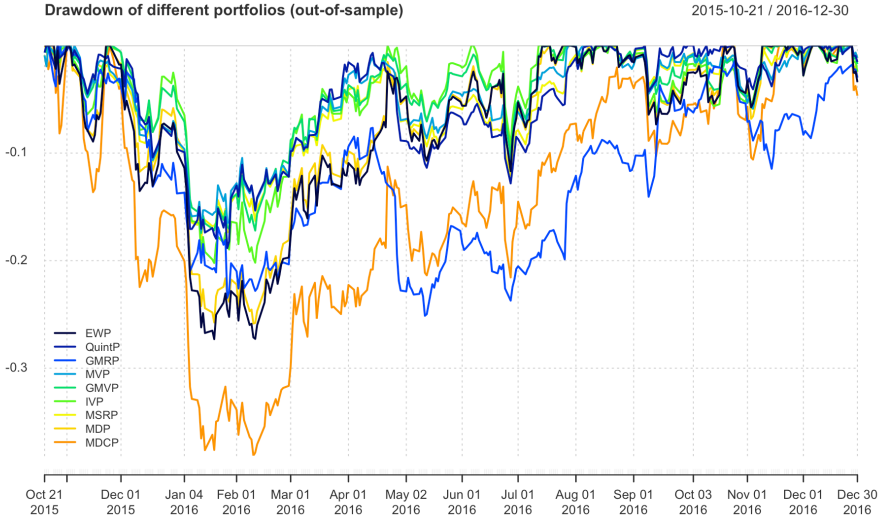
- Experiments, e.g., in physics, are conducted in a lab and can be repeated to control for different variables.
- In contrast, a backtest is a historical simulation of how a strategy would have performed in the past.
- Thus, a backtest is not an experiment, and it does not prove anything.
- A backtest guarantees nothing, not even achieving that Sharpe ratio if we could travel back in time. Random draws would have been different. The past would not repeat itself (Lopez de Prado 2018)[2].
- What is the point of a backtest then?
- It is a sanity check on a number of variables, including bet sizing, turnover, resilience to costs, and behavior under a given scenario. A good backtest can be extremely helpful, but backtesting well is extremely hard.

[2] M. Lopez de Prado, *Advances in Financial Machine Learning*. Wiley, 2018.

Example of a backtest: Cumulative P&L



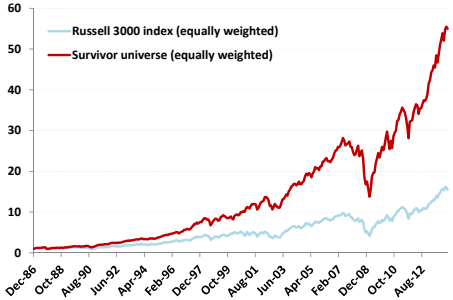
Example of a backtest: Drawdown



"The Seven Sins of Quantitative Investing"

In 2014 a team of quants at Deutsche Bank, led by Yin Luo, published a study under the title "Seven Sins of Quantitative Investing" (Luo et al. 2014)[3].

1. **Survivorship bias**: Using as investment universe the current one, hence ignoring that some companies went bankrupt and securities were delisted along the way.[4]



[3] Y. Luo, M. Alvarez, S. Wang, J. Jussa, A. Wang, and G. Rohal, "Seven sins of quantitative investing," *White paper, Deutsche Bank Markets Research*, 2014.

[4] Source of plot: Luo et al. (2014)

"The Seven Sins of Quantitative Investing"

2. **Look-ahead bias**: Using information that was not public at the moment the simulated decision would have been made. Be certain about the timestamp for each data point. Take into account release dates, distribution delays, and backfill corrections.
   👉 One example is (Glabadanidis 2015)[5] as explained in (Zakamulin 2018)[6]: the amazing performance of a strategy based on MA indicators vanished completely. 😱

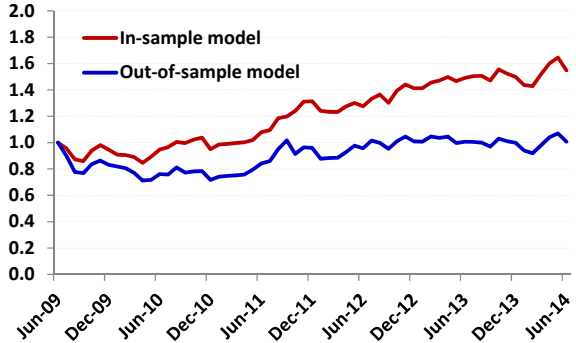
3. **Storytelling**: Making up a story ex-post to justify some random pattern.

[5] P. Glabadanidis, "Market timing with moving averages," *International Review of Finance*, vol. 15, no. 3, pp. 387–425, 2015.

[6] V. Zakamulin, "Revisiting the profitability of market timing with moving averages," *International Review of Finance*, vol. 18, no. 2, pp. 317–327, 2018.

"The Seven Sins of Quantitative Investing"

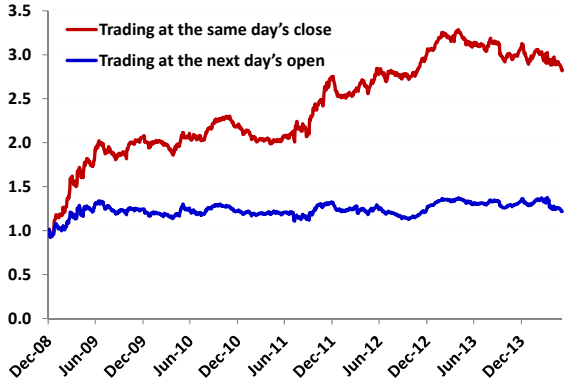
4. **Data mining and data snooping**: Training the model on the testing set.[7]



[7] Source of plot: Luo et al. (2014)

"The Seven Sins of Quantitative Investing"

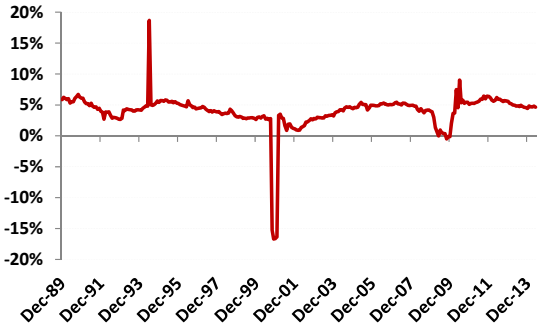
5. **Transaction costs**: Simulating transaction costs is hard because the only way to be certain about that cost would have been to interact with the trading book (i.e., to do the actual trade).[8]

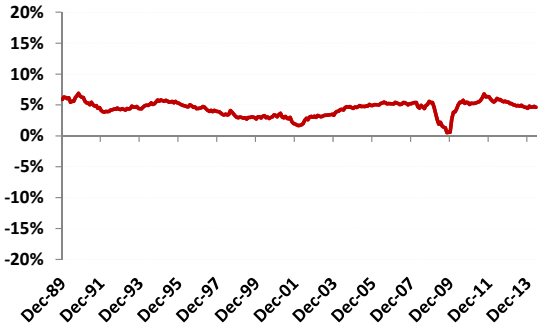


[8] Source of plot: Luo et al. (2014)

"The Seven Sins of Quantitative Investing"

6. **Outliers**: Basing a strategy on a few extreme outcomes that may never happen again as observed in the past.[9]

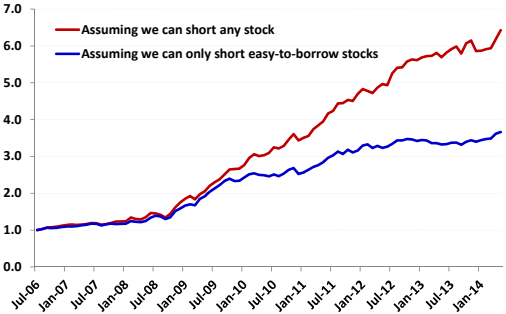




[9] Source of plot: Luo et al. (2014)

"The Seven Sins of Quantitative Investing"

7. **Shorting**: Taking a short position on cash products requires finding a lender. The cost of lending and the amount available is generally unknown, and depends on relations, inventory, relative demand. etc.[10]



These seven sins are a few basic errors that most papers published in journals make routinely.

[10] Source of plot: Luo et al. (2014)

Even if your backtest is flawless, it is probably wrong

- Suppose you have implemented a flawless backtest (everyone can reproduce your results, you have considered more than the necessary slippage and transaction costs, etc.) and it still makes a lot of money.
- Yet, this flawless backtest is probably wrong. Why?
- Because only an expert can produce a flawless backtest. Becoming an expert means that you have run tens of thousands of backtests over the years. In conclusion, this is not the first backtest you produce, so we need to account for the possibility that this is a false discovery, a statistical fluke that inevitably comes up after you run multiple tests on the same dataset.
- The maddening thing about backtesting is that, the better you become at it, the more likely false discoveries will pop up (Lopez de Prado 2018).
  - Beginners fall for the seven sins of Luo et al. (Luo et al. 2014).
  - Professionals may produce flawless backtests, and will still fall for multiple testing, selection bias, or backtest overfitting.

Pessimistic views on backtesting

Some pesimistic views on backtesting from (Lopez de Prado 2018)[11]:

- Backtesting is not a research tool.
- It provides us with very little insight into the reason why a particular strategy would have made money. Just as a lottery winner may feel he has done something to deserve his luck, there is always some ex-post story (Luo's sin number three).
- Authors claim to have found hundreds of "alphas" and "factors," and there is always some convoluted explanation for them. Instead, what they have found are the lottery tickets that won the last game. The winner has cashed out, and those numbers are useless for the next round.
- If you would not pay extra for those lottery tickets, why would you care about those hundreds of alphas? Those authors never tell us about all the tickets that were sold, that is, the millions of simulations it took to find these "lucky" alphas.

[11] M. Lopez de Prado, *Advances in Financial Machine Learning*. Wiley, 2018.

So what's the point of backtesting?

- The purpose of a backtest is to discard bad models, not to improve them.
- Adjusting your model based on the backtest results is a waste of time... and it's dangerous.
- Invest your time and effort developing a sound strategy. By the time you are backtesting, it is too late. Never backtest until your model has been fully specified.

Backtesting overfitting

- Backtest overfitting can be defined as selection bias on multiple backtests (Bailey et al. 2016)[12].
- It takes place when a strategy is developed to perform well on a backtest, by monetizing random historical patterns. Because those random patterns are unlikely to occur again in the future, the strategy so developed will fail.
- The only backtests that most people share are those that portray supposedly winning investment strategies.
- How to address backtest overfitting is arguably the most fundamental question in quantitative finance.
- What makes backtest overfitting so hard to assess is that the probability of false positives changes with every new test conducted on the same dataset. That information is either unknown by the researcher or not shared with investors or referees.
- While there is no easy way to prevent overfitting, a number of steps can help reduce its presence.

[12] D. Bailey, J. Borwein, and M. L. de Prado, "Stock portfolio design and backtest overfitting," *Journal of Investment Management*, vol. 15, no. 1, pp. 1–13, 2016.

Backtesting recommendations

Some recommendations from (Lopez de Prado 2018):

- Develop models for entire asset classes or investment universes, rather than for specific securities (to reduce the prob. of false discoveries).
- Apply bagging (a machine learning technique based on ensembles) as a means to both prevent overfitting and reduce the variance of the forecasting error.
- Do not backtest until all your research is complete.
- Keep track of the number of backtests conducted on a dataset so that the probability of backtest overfitting may be estimated and the Sharpe ratio may be properly deflated.
- Simulate scenarios rather than history (e.g., stress testing). A standard backtest is a historical simulation, which can be easily overfit. Your strategy should be profitable under a wide range of scenarios, not just the anecdotal historical path.

Outline

Backtesting: Historical data vs synthetic data

- A backtest evaluates out-of-sample the performance of an investment strategy using past observations.
- These past observations can be used in two ways:
  1. to simulate the historical performance of an investment strategy, as if it had been run in the past;
  2. to simulate scenarios that did not happen in the past.
- The first (narrow) approach, also known as **walk-forward**, is so prevalent that, in fact, the term "backtest" has become a *de facto* synonym for "historical simulation."
- The second (broader) approach is less known. One example is the so-called **stress tests** (where different type of markets are recreated to test the strategy).
- Each approach has its pros and cons, and each should be given careful consideration.
- To perform a proper backtesting, we must find a different (true **out-of-sample**) validation procedure, i.e., using observations least likely to be correlated with the training data.

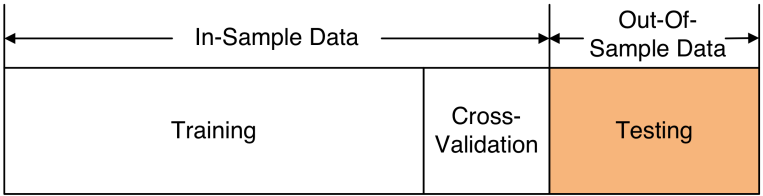
Cross-Validation (CV) backtesting

- The purpose of **cross-validation** (CV) is to determine the generalization error of an machine learning (ML) algorithm, so as to prevent overfitting.
- When we test an ML algorithm on the same dataset as was used for training, not surprisingly, we achieve spectacular results, but they have zero forecasting power.
- CV splits observations drawn from an i.i.d. process into two sets: the **training set** and the **testing set**.
- There are many alternative CV schemes that can be used with financial data for backtesting:
  - vanilla one-shot backtesting;
  - walk-forward backtesting;
  - k-fold CV backtesting;
  - combinatorial purged cross-validation (CPCV) backtesting;
  - multiple randomized backtesting;
  - etc.

Outline

Vanilla backtesting: in-sample and out-of-sample

- To perform a simple vanilla backtest, one divides the data into:
  - **in-sample data**, used to train and cross-validate the strategy (this is further divided into **training data** and **cross-validation data**); and
  - **out-of-sample** or **test data**, used to evaluate the strategy with new data.
- The **training data** is used to estimate the model parameters; in portfolio design, this typically amounts to estimating the sample mean of the returns $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$.
- The **cross-validation data** is used to choose a few hyper-parameters; in a mean-variance Markowitz portfolio design this could be the choice of the risk-aversion parameter.
- The **test data** is used to evaluate the performance of the strategy.



Outline

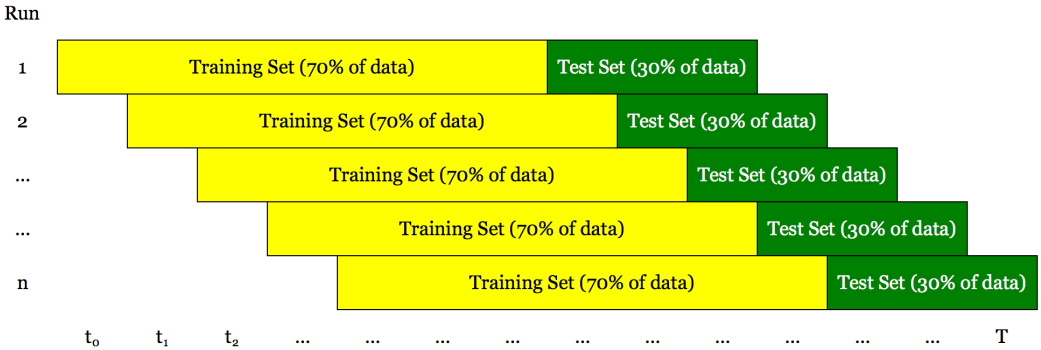
Walk-forward (WF) backtesting

- The most common backtest method in the literature is the walk-forward (WF) approach (Pardo 2008)[13].
- WF is a rolling-window version of the vanilla backtest. That is, the in-sample and out-of-sample windows are constantly shifted or slided.
- WF is a historical simulation of how the strategy would have performed in past.
- Each strategy decision is based on observations that predate that decision.
- Carrying out a flawless WF simulation is a daunting task.
- WF enjoys two key advantages:
  1. WF has a clear historical interpretation and its performance can be reconciled with paper trading.
  2. History is a filtration; hence, using trailing data guarantees that the testing set is out-of-sample (no leakage), as long as purging has been properly implemented

[13] Pardo, *The Evaluation and Optimization of Trading Strategies*, 2nd. John Wiley & Sons, 2008.

Walk-forward (WF) backtesting

- This figure illustrates the rolling-window approach of the training set and test set:



- The anchored WF is a variation where the training set grows as time progresses, i.e., it always starts at the very begining.

Pitfalls of WF backtesting

- A single scenario is tested (the historical path), which can easily lead to **overfitting**.
- So, WF is not necessarily representative of future performance, as results can be biased by the particular sequence of datapoints.
- It is a common mistake to find **leakage** in WF backtests.
  👉 One example is (Glabadanidis 2015)[14] as explained in (Zakamulin 2018)[15]: the amazing performance of a strategy based on MA indicators vanished completely. 😱
- The initial decisions are made on a smaller portion of the total sample. Even if a warm-up period is set, most of the information is used by only a small portion of the decisions.

[14] P. Glabadanidis, "Market timing with moving averages," *International Review of Finance*, vol. 15, no. 3, pp. 387–425, 2015.

[15] V. Zakamulin, "Revisiting the profitability of market timing with moving averages," *International Review of Finance*, vol. 18, no. 2, pp. 317–327, 2018.

Outline

Cross-Validation (CV) backtesting

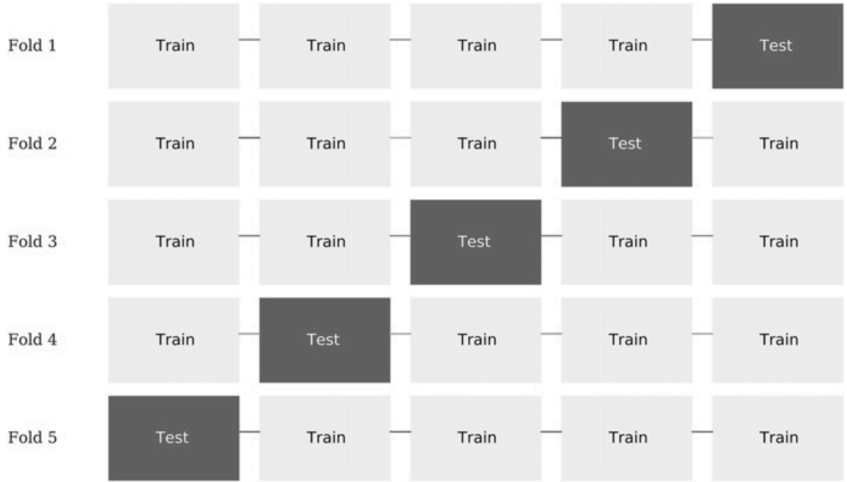
- A vanilla backtest would simply split the data into training and test data (in-sample and out-of-sample), but this is a single backtest!
- A WF backtest would do it in a rolling-window fashion, but it's still a single historical path.
- The idea in cross-validation backtesting is to test k alternative scenarios (of which only one corresponds with the historical sequence).
- Some issues:
  1. It is still using a single path of data.
  2. Cross-validation (CV) backtesting does not have a clear historical interpretation.
  3. Leakage is possible because the training data does not trail the test data. Extreme care must be taken to avoid leaking testing information into the training set.

k-fold Cross-Validation (CV) backtesting

- This is a common approach in machine learning (ML) applications:
  1. The dataset is partitioned into k subsets.
  2. For $i = 1, \ldots, k$
     a. The ML algorithm is trained on all subsets excluding i.
     b. The fitted ML algorithm is tested on i.
- In finance, k-fold CV is typically used in two settings: model development (like hyper-parameter tuning) and backtesting.

k-fold Cross-Validation (CV) backtesting

Train/test splits in a 5-fold CV scheme:



k-fold Cross-Validation (CV) backtesting

- One reason k-fold CV fails in finance is because observations cannot be assumed to be drawn from an i.i.d. process.
- Leakage takes place when the training set contains information that also appears in the testing set.
  - If X is a predictive feature, leakage will enhance the performance of an already valuable strategy.
  - The problem is leakage in the presence of irrelevant features, as this leads to false discoveries.
- There are at least two ways to reduce the likelihood of leakage (Lopez de Prado 2018)[16]:
  1. Drop from the training set any observation i where Y_i is a function of information used to determine Y_j, and j belongs to the testing set. For example, Y_i and Y_j should not span overlapping periods.
  2. Avoid overfitting the classifier. In this way, even if some leakage occurs, the classifier will not be able to profit from it. For example, one can use early stopping of the base estimators or bagging of classifiers.

[16] M. Lopez de Prado, *Advances in Financial Machine Learning*. Wiley, 2018.

Purged k-fold CV backtesting

"Purging" and "embargo" are described in (Lopez de Prado 2018)[17] as a way to fix the k-fold CV backtesting:

- Purging: One way to reduce leakage is to purge from the training set all observations whose labels overlapped in time with those labels included in the testing set.
- Embargo: In addition, since financial features often incorporate series that exhibit serial correlation (like ARMA processes), we should eliminate from the training set observations that immediately follow an observation in the testing set.
- There are other more sophisticated ways to split the data like the combinatorial purged cross-validation (CPCV) method in Section 12.4 of (Lopez de Prado 2018).

[17] M. Lopez de Prado, *Advances in Financial Machine Learning*. Wiley, 2018.

Purging

Avoiding leakage:

Outline

Multiple randomized backtesting

- The main drawback of the vanilla backtesting, the WF bactesting, and the k-fold CV backtesting is that they use a **single historical path**.
- The idea with multiple randomized backtesting is to use different paths.
- But how can we accomplish that if historical data is essentially a single path?
- One way is implemented in the R package portfolioBacktest: it performs multiple backtests of portfolios in an automated way on a rolling-window basis by taking data randomly from different markets, different time periods, and different stock universes.
- Details of the package can be found in this vignette.

Multiple randomized backtesting

- Multiple randomized backtesting generates multiple datasets from historical market data on a randomized fashion by randomly choosing different periods of time and randomly choosing a subset of the universe.
- For example, if the original data contains 500 stocks over a period of 10 years, one could choose at random 100 stocks over a random consecutive period of 2 years, and repeat this process a large number of times to get randomized datasets.
- This will introduce some randomness in each individual dataset and it will span different market regimes encountered over the 10 years.
- For each of the resampled datasets, a walk-forward backtesting can then be performed.

Outline

Backtesting with synthetic data

- The problem with backtesting on historical data is the danger of overfitting to the particular history path.
- **Monte Carlo simulations** offer a partial solution:
  - **resampling the existing history**: in its simplest version this means sampling the realized sequence of returns with a different order;
  - **creating a synthetic dataset**: characterize statistically the observed market historical data and then use those statistics to generate synthetic data.
- This will allow us to backtest a strategy on a large number of unseen, synthetic testing sets, hence reducing the likelihood that the strategy has been fit to a particular set of datapoints.
- However, the accuracy of such simulations will depend on how the new data is generated: Gaussian distribution vs heavy-tailed and skewed distributions.
- Time series modeling is key in order to generate valuable synthetic data.

Stress testing

- Monte Carlo simulations based on the observed historical data are a significant improvement on a vanilla backtest directly on the historical data.
- However, those newly generated data will still follow the market trend corresponding to the original observed data.
- Stress testing generates synthetic data corresponding to **different market scenarios** such as bull markets, bear markets, side markets, crises, bubbles, etc.
- One can even consider specific periods of crises such as the stock market crash of October 1987, the Asian crisis of 1997, and the tech bubble that burst in 1999-2000.
- This way, the backtest is even more diverse by exploring different possible financial scenarios.
- In other words, stress testing tests the resilience of investment portfolios against possible future financial situations.
- It's the equivalent of exploring how the strategy might have performed over hundreds of years during a spectrum of market conditions.

Outline

Backtesting statistics

- There are many ways to perform a backtesting of a strategy (e.g., based on historical data, scenario based simulations, synthetic data).
- Regardless of the backtesting paradigm you choose, you need to report the results according to a series of statistics that investors will use to compare and judge your strategy against competitors.
- Some of these statistics are included in the Global Investment Performance Standards (GIPS): https://www.gipsstandards.org
- Backtest statistics comprise metrics used by investors to assess and compare various investment strategies.
- They should help us uncover potentially problematic aspects of the strategy, such as substantial asymmetric risks or low capacity.
- Overall, they can be categorized into general characteristics, performance, runs/drawdowns, implementation shortfall, return/risk efficiency, and attribution, cf. Chapter 14 in (Lopez de Prado 2018)[18].

[18] M. Lopez de Prado, *Advances in Financial Machine Learning*. Wiley, 2018.

General characteristics

The following statistics inform us about the general characteristics of the backtest:

- **Time range**: It specifies the start and end dates. The period used to test the strategy should be sufficiently long to include a comprehensive number of regimes.
- **Average AUM**: This is the average dollar value of the assets under management.
- **Capacity**: A strategy's capacity can be measured as the highest AUM that delivers a target risk-adjusted performance. A minimum AUM is needed to ensure proper bet sizing and risk diversification. Beyond that minimum AUM, performance will decay as AUM increases, due to higher transaction costs and lower turnover.
- **Leverage**: Leverage measures the amount of borrowing needed to achieve the reported performance. If leverage takes place, costs must be assigned to it. One way to measure leverage is as the ratio of average dollar position size to average AUM.

General characteristics

- **Maximum dollar position size**: This informs us whether the strategy at times took dollar positions that greatly exceeded the average AUM. In general we will prefer strategies that take maximum dollar positions close to the average AUM, indicating that they do not rely on the occurrence of extreme events or outliers.
- **Ratio of longs**: This indicates what proportion of the bets involved long positions. In long-short, market neutral strategies, ideally this value is close to 0.5. If not, the strategy may have a position bias, or the backtested period may be too short and unrepresentative of future market conditions.
- **Frequency of bets**: The number of bets per year in the backtest. A sequence of positions on the same side is considered part of the same bet. A bet ends when the position is flattened or flipped to the opposite side. The number of bets is always smaller than the number of trades. A trade count would overestimate the number of independent opportunities discovered by the strategy.

General characteristics

- **Average holding period**: The average number of days a bet is held. High-frequency strategies may hold a position for a fraction of seconds, whereas low frequency strategies may hold a position for months or even years. Short holding periods may limit the capacity of the strategy. The holding period is related but different to the frequency of bets.
- **Annualized turnover**: It measures the ratio of the average dollar amount traded per year to the average annual AUM. High turnover may occur even with a low number of bets, as the strategy may require constant tuning of the position. High turnover may also occur with a low number of trades, if every trade involves flipping the position between maximum long and maximum short.
- **Correlation to underlying**: This is the correlation between strategy returns and the returns of the underlying investment universe. When the correlation is significantly positive or negative, the strategy is essentially holding or short-selling the investment universe, without adding much value.

Performance

Performance statistics are dollar and returns numbers without risk adjustments. Some useful performance measurements include:

- **PnL**: Total amount of dollars generated over the entirety of the backtest.
- **PnL from long positions**: Portion of the PnL generated by long positions (interesting value to assess the bias of long-short, market neutral strategies).
- **Annualized return**: The time-weighted average annual rate of total return, including dividends, coupons, costs, etc.
- **Hit ratio**: The fraction of bets that resulted in a positive PnL.
- **Average return from hits/misses**: The average return from bets that generated a profit/loss.

The total returns is the rate of return from realized and unrealized gains and losses, including accrued interest, paid coupons, and dividends for the measurement period. GIPS rules calculate time-weighted rate of returns (TWRR), adjusted for external cash flows.

Runs statistics

Investment strategies rarely generate returns drawn from an i.i.d. process. Instead, the returns series exhibit frequent runs (uninterrupted sequences of returns of the same sign). We need proper metrics to assess runs.

- **Returns concentration**: the concentration of positive returns can be defined (inspired by the Herfindahl-Hirschman Index (HHI)) as

$$h^+ = \frac{\sum_{t=1}^{T^+}(w_t^+)^2 - 1/T^+}{1 - 1/T^+} = \left(\frac{E[(r_t^+)^2]}{E[r_t^+]^2} - 1\right)\left(\frac{1}{T^+ - 1}\right)$$

  where w_t^+ denotes the normalized positive returns r_t^+, $w_t^+ = \frac{r_t^+}{\sum_{t'} r_{t'}^+}$, and T^+ is the number of such positive returns. The same can be done with the negative returns.

- **Drawdown (DD) and Time under Water (TuW)**: DD is the maximum loss suffered by an investment between two consecutive high-watermarks (HWMs)[19] and TuW is the time elapsed inbetween.

[19] HWM: Rolling maximum of the cumulative PnL.

Implementation shortfall

Investment strategies often fail due to wrong assumptions regarding execution costs. Some important measurements of this include:

- **Broker fees per turnover**: fees paid to the broker for turning the portfolio over, including exchange fees.
- **Average slippage per turnover**: execution costs, excluding broker fees, involved in one portfolio turnover. For example, the loss caused by buying a security at a fill-price higher than the mid-price when the order was sent to the broker.
- **Return over turnover (ROT)**: ratio between dollar performance and portfolio turnover.
- **Return on execution costs**: ratio between dollar performance (including brokerage fees and slippage costs) and total execution costs. It should be a large multiple, to ensure that the strategy will survive worse-than-expected execution.

Efficiency statistics

- **Sharpe ratio**: suppose that a strategy's excess returns (in excess of the risk-free rate), $r_t,\ t = 1, \ldots, T$, are i.i.d. with mean μ and variance σ^2. The Sharpe ratio (SR) is defined as

$$\text{SR} = \frac{\mu}{\sigma}.$$

  It evaluates the skills of a particular strategy or investor. Since μ and σ are usually unknown, the true SR value cannot be known for certain and in practice the SR will contain substantial estimation errors.
- **Annualized SR**: SR value, annualized by a factor $\sqrt{a}$, where a is the average number of returns observed per year. This common annualization method relies on the assumption that returns are i.i.d.
- **Information ratio**: SR equivalent of a portfolio that measures its performance relative to a benchmark. The excess return is measured as the portfo lio's return in excess of the benchmark's return. The tracking error is estimated as the standard deviation of the excess returns.

Efficiency statistics

Some refinements of the SR, to account for limited observations and repeated trials, include (Lopez de Prado 2018)[20]:

- **Probabilistic Sharpe Ratio (PSR)**: it provides an adjusted estimate of SR, by removing the inflationary effect caused by short series with skewed and/or fat-tailed returns.
- **Deflated Sharpe Ratio (DSR)**: is a PSR where the rejection threshold is adjusted to reflect the multiplicity of trials.

[20] M. Lopez de Prado, *Advances in Financial Machine Learning*. Wiley, 2018.

Attribution

- The purpose of performance attribution is to decompose the PnL in terms of risk classes.
- For example, a corporate bond portfolio manager typically wants to understand how much of its performance comes from his exposure to the following risks classes: duration, credit, liquidity, economic sector, currency, sovereign, issuer, etc.
- Did his duration bets pay off? What credit segments does he excel at? Or should he focus on his issuer selection skills?
- These risks are not orthogonal, so there is an overlap between them. The sum of the attributed PnL's will not match the total PnL, but at least one is able to compute the Sharpe ratio (or information ratio) per risk class. Example: Barra's multi-factor method.
- Of equal interest is to attribute PnL across categories within each class. For example, the duration class could be split between short duration (less than 5 years), medium duration (between 5 and 10 years), and long duration (in excess of 10 years).

Outline

Backtesting and its dangers

- When a trader designs a portfolio strategy, the first thing to do is to backtest it.
- Backtesting is the process by which the portfolio strategy is put to test using the past historical market data available.
- A common approach is to do a single backtest against the existing historical data and then plot graphs and draw conclusions from that. One example is the so-called walk-forward backtest.
- This is a **big mistake**. Performing a single backtest is not representative as it is just one realization and one will definitely overfit the tested strategy if there is parameter tuning involved or portfolio comparisons involved. Section 1 of this book chapter on backtesting illustrates the dangers of backtesting.
- It is necessary to perform multiple backtests on different datasets, say, 500 datasets. Each dataset should contain a different period, with different market conditions, and different asset universe.

R package `portfolioBacktest`

- The R package portfolioBacktest performs multiple backtests of portfolios in an automated way on a rolling-window basis by taking data randomly from different markets, different time periods, and different stock universes.
- In more detail, it generates multiple datasets from historical market data on a randomized fashion by randomly choosing different periods of time and randomly choosing a subset of the universe.
- For example, if the original data contains 500 stocks over a period of 10 years, it could choose at random 100 stocks over a random consecutive period of 2 years, and repeat this process a large number of times to get randomized datasets.
- This will introduce some randomness in each individual dataset and it will span different market regimes encountered over the 10 years.
- For each of the resampled datasets, a walk-forward (aka rolling-window) backtesting is performed.
- Details of the package can be found in this vignette.

Usage of R package portfolioBacktest

Step 1 - load package & datasets:

```
library(portfolioBacktest)
data("dataset10")
```

The variable dataset10 constains 10 toy datasets; however, for a serious backtesting one should load more data and generate many more randomized datasets (see vignette for details):

```
# download data
data("SP500_symbols")
SP500 <- stockDataDownload(stock_symbols = SP500_symbols,
                           from = "2008-12-01",
                           to = "2018-12-01")
# resample
dataset500 <- stockDataResample(SP500, num_datasets = 500,
                                N_sample = 50,
                                T_sample = 252*2)
```

Usage of R package portfolioBacktest

Step 2 - define your own portfolio to backtest:

```
my_portfolio <- function(dataset) {
  prices <- dataset$adjusted
  N <- ncol(prices)
  w <- rep(1/N, N)
  return(w)
}
```

Step 3 - do backtest:

```
bt <- portfolioBacktest(my_portfolio, dataset500)
```

Step 4 - check your portfolio performance:

```
backtestSummary(bt)$performance
```

Example of R package `portfolioBacktest`

Example of **performance table** obtained with the R package `portfolioBacktest` over 500 resampled datasets:

Show 10 entries Search:

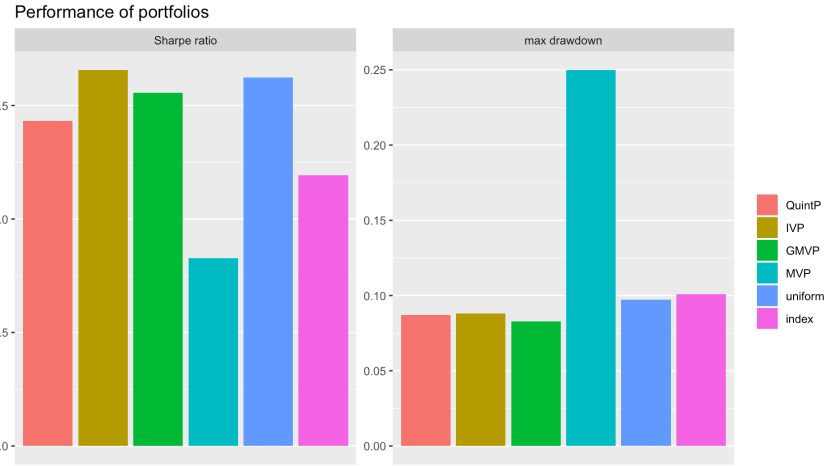
Leaderboard:

	cpu time	Sharpe ratio	max drawdown	annual return	annual volatility	Sterling ratio	Omega ratio	ROT (bps)
IVP	0.0019	1.819	8.6%	0.2056	11.8%	2.4372	1.3291	809.2451
uniform	0.0013	1.8102	9.3%	0.2238	12.6%	2.5081	1.3285	969.1074
GMVP	0.0598	1.6133	8.2%	0.1511	10.0%	1.8712	1.2934	234.0871
QuintP	0.0022	1.361	8.9%	0.1722	12.6%	1.9509	1.2551	210.2286
index	0	1.1989	10.1%	0.1452	11.7%	1.7212	1.2383	
MVP	0.1878	1.0117	23.1%	0.3067	33.9%	1.396	1.2124	302.5476

Showing 1 to 6 of 6 entries Previous 1 Next

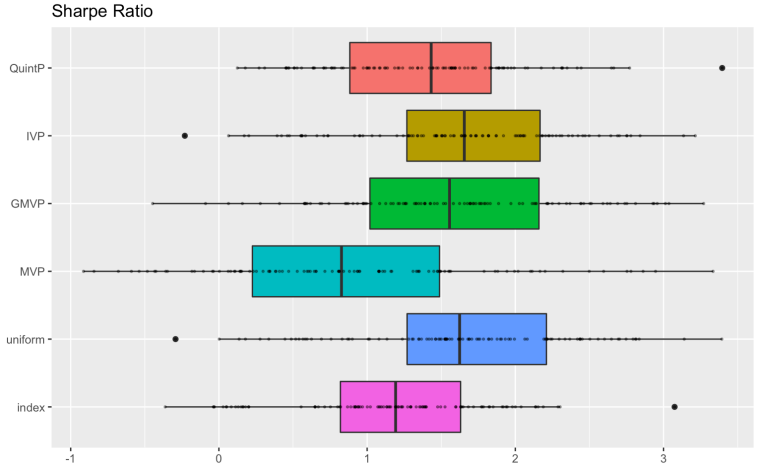
Example of R package `portfolioBacktest`

Example of **barplot** obtained with the R package `portfolioBacktest` over 500 resampled datasets:



Example of R package `portfolioBacktest`

Example of **boxplot** obtained with the R package `portfolioBacktest` over 500 resampled datasets:



Thanks

For more information visit:

https://www.danielppalomar.com

References I

Bailey, D., Borwein, J., & Prado, M. L. de. (2016). Stock portfolio design and backtest overfitting. *Journal of Investment Management*, *15*(1), 1–13.

Glabadanidis, P. (2015). Market timing with moving averages. *International Review of Finance*, *15*(3), 387–425.

Lopez de Prado, M. (2018). *Advances in financial machine learning*. Wiley.

Luo, Y., Alvarez, M., Wang, S., Jussa, J., Wang, A., & Rohal, G. (2014). Seven sins of quantitative investing. *White paper, Deutsche Bank Markets Research*.

Pardo. (2008). *The Evaluation and Optimization of Trading Strategies* (2nd ed.). John Wiley & Sons.

Zakamulin, V. (2018). Revisiting the profitability of market timing with moving averages. *International Review of Finance*, *18*(2), 317–327.