

# Backtesting

---

Backtesting is the process of feeding historical data to an automated trading strategy and see how it would have performed. We will study various common backtest performance metrics. Backtest performance can easily be made unrealistic and un-predictive of future returns due to a long list of pitfalls, which will be examined in this course. The choice of a software platform for backtesting is also important, and criteria for this choice will be discussed. Illustrative examples are drawn from a futures strategy and a stock portfolio trading strategy.

This is a pre-recorded workshop conducted in Adobe Connect by Ernest Chan ([www.epchan.com](http://www.epchan.com)). This workshop focuses on the various practices and pitfalls of backtesting algorithmic trading strategies. Free MATLAB trial licenses will be arranged for extensive in-class exercises. No prior knowledge of MATLAB is assumed, but some programming experience is necessary. The math requirement assumed is basic college-level statistics.

- Total hours: 7 hours of recorded session.
- Registration: [ernest@epchan.com](mailto:ernest@epchan.com).

## Course outline:

### A. *Overview of Backtesting*

1. What is backtesting and how does it differ from “simulations”?
2. The importance of backtesting: Why is backtesting a necessary step for profitable automated trading?
3. The limitations of backtesting: Why is backtesting *not* a sufficient step to ensure profitability in automated trading?
4. What we can do to increase the predictive power of our backtest results: the avoidance of pitfalls.
5. How to identify good/bad strategies even before a backtest: a preview of various pitfalls through a series of examples.

### B. *Choosing a backtest platform*

1. Criteria for choosing a suitable backtest platform.
2. A list of backtesting platforms.
3. Discussion of pros and cons of each platform.
4. Special note: integrated backtesting and automated execution platforms.
5. Why do we choose MATLAB?

### C. *Tutorial to MATLAB*

1. Survey of syntax.

2. Advantage of array processing.
3. Exercises: building utility functions useful for backtesting.
4. Using toolboxes.

#### *D. Backtesting a single-instrument strategy*

1. Exercise: A Bollinger-band strategy for E-mini S&P500 futures (ES) as a prototype mean-reversion strategy.

#### *E. Performance measurement*

1. The equity curve.
2. Excess returns and the importance of the Sharpe ratio.
3. Tail risks and maximum drawdown and drawdown duration.
4. The importance of transaction costs estimates.

#### *F. Choosing a historical database*

1. Criteria for choosing a good historical database.
2. Equities data: split/dividend adjustments, survivorship bias.
3. Futures data: constructing continuous contracts, settlement vs closing prices.
4. Issues with synchronicity of data.
5. Issues with intraday/tick data.

#### *G. Backtesting a portfolio strategy*

1. Exercise: A long-short portfolio strategy of stocks in the S&P 500.
2. Relevance of strategy to 2007 quant funds meltdown.
3. The importance of universe selection: impact of market capitalization, liquidity, and transactions costs on strategies.
4. Strategy refinement: how small changes can make big differences in performance.

#### *H. Detection and elimination of backtesting pitfalls and bias*

1. How to detect look-ahead bias?
2. How to avoid look-ahead bias?
3. Data snooping bias: why out-of-sample testing is not a panacea.
4. Parameterless trading.
5. The use of linear models or “averaging-in”: pros and cons.
6. Exercise: linearization of the ES Bollinger band strategy.
7. Impact of noisy data on different types of strategies.
8. Impact of historical or current short-sale constraint.
9. The unavoidable limitation of backtesting: Regime change.
10. What to do when live performance is below expectations?