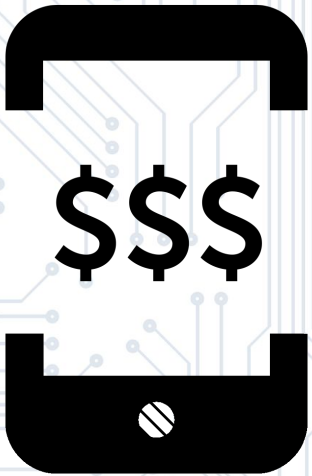


Algorithmic Trading

FinTech

Lesson 15.1

Legal Disclaimer

The content contained in the FinTech Boot Camp (the “Course”) is for informational purposes only. You should not construe any such information or other material provided pursuant to the Course (the “Materials”) as investment, financial, tax, legal or other advice. The Materials are not investment advice and any observations concerning any security, trading algorithm or investment strategy provided in the Course is not a recommendation to buy, sell or hold such investment or security or to make any other investment decisions. Trilogy Education Services, LLC, its parent, subsidiaries, and affiliates (“Trilogy”) do not provide any advice regarding the nature, potential value, risk or suitability of any particular investment strategy, trading algorithm, transaction, security or investment. Any use of the Materials, and any decisions made in reliance thereon, including any trading or investment decisions or strategies, are made at your own risk. You should seek the advice of a competent, licensed professional if you require investment, trading or other advice. No employee, agent or representative of Trilogy is authorized to provide any professional advice in connection with the Course and any such advice, if given, is in violation of Trilogy’s policies, is unauthorized and may not be relied upon. Nothing contained in the Materials constitutes a solicitation, recommendation, endorsement, or offer by Trilogy or any third party service provider to buy or sell any securities or other financial instruments in this or in in any other jurisdiction whether or not such solicitation or offer would be unlawful under the securities laws of such jurisdiction.

Intro to Algorithmic Trading

Intro to Algorithmic Trading

A typical day for traders involves:

- Tracking the transactional history of many stocks
- Identify the best opportunity to buy, sell, and hold
- Maintain knowledge about the highs and lows for each individual stock, as well as their overall portfolio value and profit/loss
- Keeping emotions in check

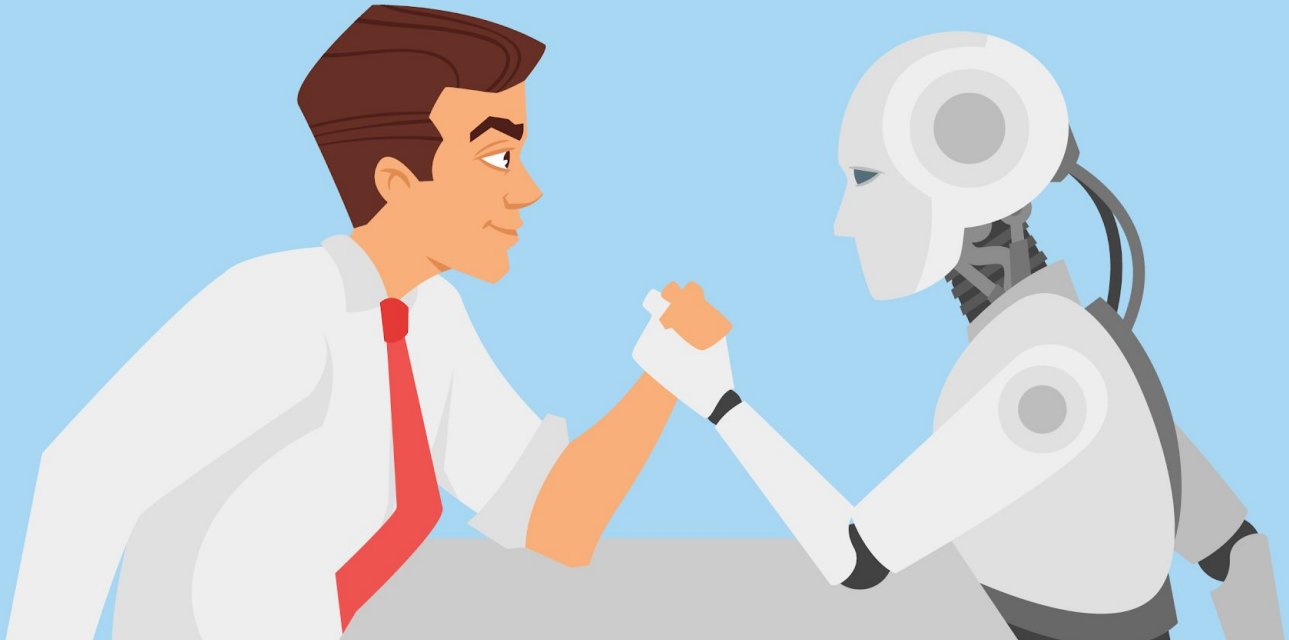
Intro to Algorithmic Trading

The sheer number of moving parts and details that need to be considered can make it difficult for the human mind to consistently make performant trades.

This is where **algorithmic trading** comes in.

Intro to Algorithmic Trading

The key difference between human traders and algorithmic trading is that computers can make decisions and predictions much more efficiently and effectively than humans, and they can do so without human emotions getting in the way.



Intro to Algorithmic Trading

It's common for FinTech firms to use algorithms to make trading decisions across all domains. This applies to trading in stocks, bonds, options, currencies (money transfer), carbon credits, mortgages, or even consumer loans.

What Is Algorithmic Trading?

Intro to Algorithmic Trading

Algo trading is the use of models to make trading decisions.

C

Intro to Algorithmic Trading

Algorithmic trading is simply using code to decide and execute a trade (buy/sell).

For example

A simple algorithm could be to sell 100 shares of a stock when the intraday percentage return of a stock falls below -3%.

```
# A pseudocode snippet for preventing further intraday loss of a stock
# by selling a stock when the intraday return falls below -3%. There is
# an assumption that a StockMarket account has 100 shares of MU.

# Initialize StockMarket account object
account = StockMarket()

# Set stock ticker and share size
stock = “MU”
shares = 100

# Continuous loop that checks intraday return of a particular stock
while True:
    # Check intraday return
    intraday_return = account.get_intraday_return(stock)

    # If intraday percentage return falls below -3%, sell the stock and shares
    if intraday_return < -0.03:
        account.sell(shares, stock)
```

Why Is Algorithmic Trading So Prevalent Today?

Intro to Algorithmic Trading

Advances in network technology and computing power along with authorization to use electronic exchanges have made it possible for anyone to write code and automate their trading strategy.

The result

Algorithmic trading can be orders of magnitude faster than traditional trading.

([The Economist])

Why Use Algorithmic Trading?

Intro to Algorithmic Trading

While there are many benefits to using algorithmic trading over traditional/manual trading, the two main benefits are:

01

Algorithmic trading can be backtested using historical and real-time data to see if it is a profitable trading strategy.

02

It reduces the possibility of human error in which traders mistime trades based on emotional and psychological factors.

Who Performs Algorithmic Trading?

Intro to Algorithmic Trading

Anyone who can code can create their own algorithmic trader bot.

Traditionally, Quantitative Analysts, or Quant Traders, create algorithmic trading models and are often required to have at least a master's or PhD degree level with a familiar background in statistics, but with human-friendly programming languages like Python and analytical tools and machine learning libraries, anyone can learn to build a very powerful algorithmic trader.

Intro to Algorithmic Trading

Some use algorithms to help inform, with the human making the final trade decision.

Intro to Algorithmic Trading

Others make algorithmic trading “full autopilot:” trade decisions, trade execution, rebalancing, and risk management, reporting are all automated.

Intro to Algorithmic Trading

Basic definitions you will need to know:

Signals

Information that is useful for predicting the future direction of an asset (e.g., a stock)

Entry Rules

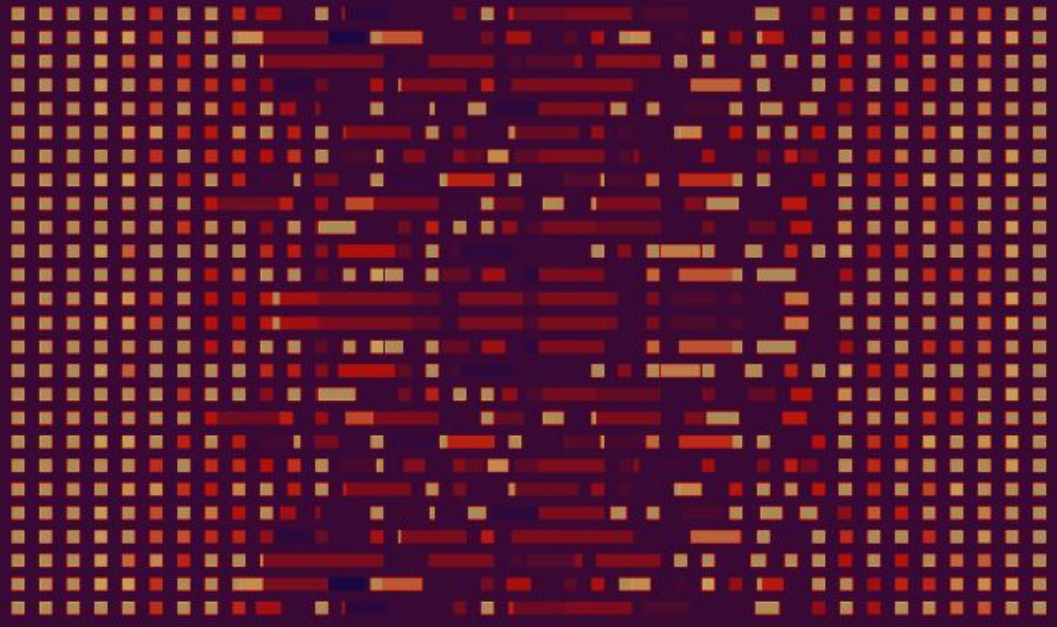
A decision rule telling you when to buy an asset (such as when a signal reaches a pre-specified, high enough level)

Exit Rules

A decision rule telling you when to sell or dispose of an asset (such as when a signal reaches a pre-specified, low enough level)

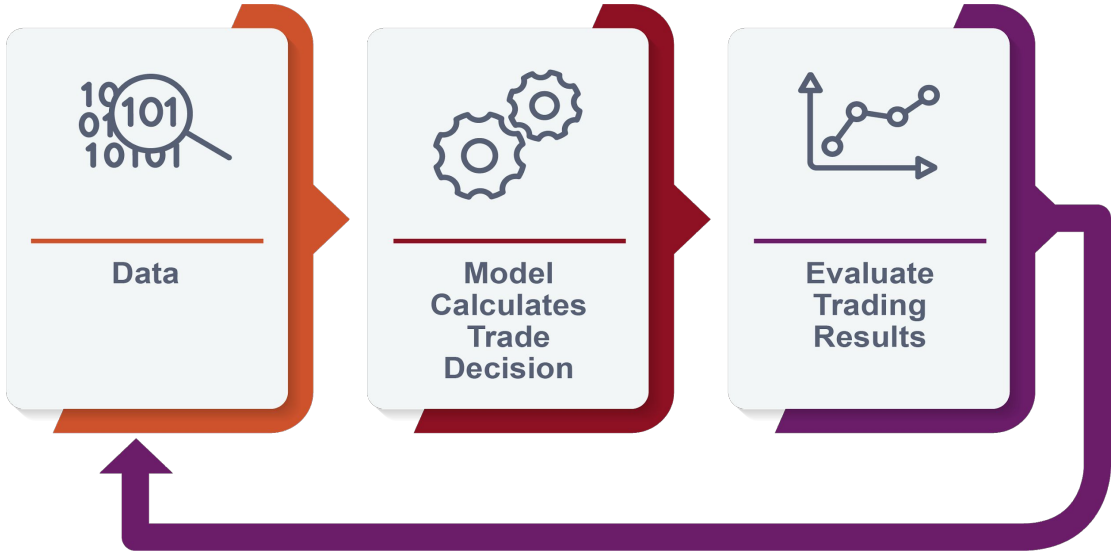
Intro to Algorithmic Trading

Good algo trading is largely about good signal generation. One of the ways to create a good signal is through building good Machine Learning (ML) models.



Intro to Algorithmic Trading

The model takes in data, and outputs a decision of whether to buy or sell.



In this unit, we'll start with some simple rule-based strategies, then move to an ML approach.

Demystifying Algo Trading

Demystifying Algo Trading

Trading algorithms are often misconstrued to be highly complex and esoteric, however, when breaking down the core components of a trading algorithm, there are three fundamental parts:

01

Obtaining the data

02

Making a trade decision

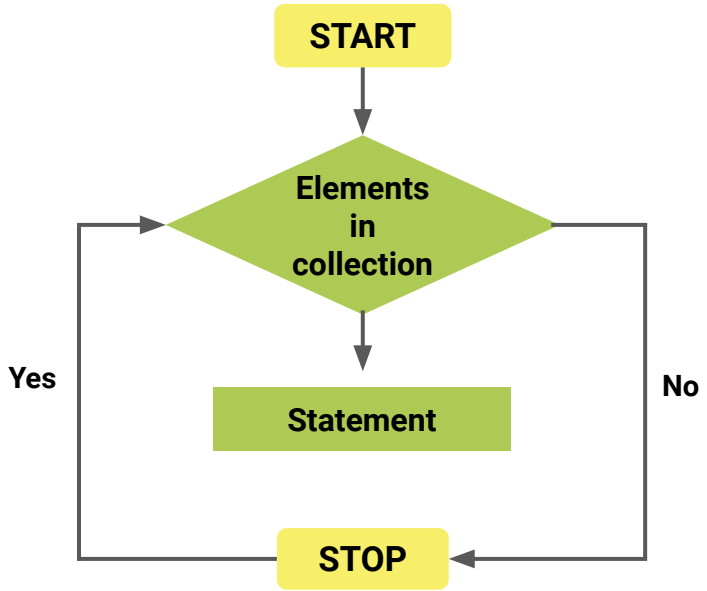
03

Evaluating the results

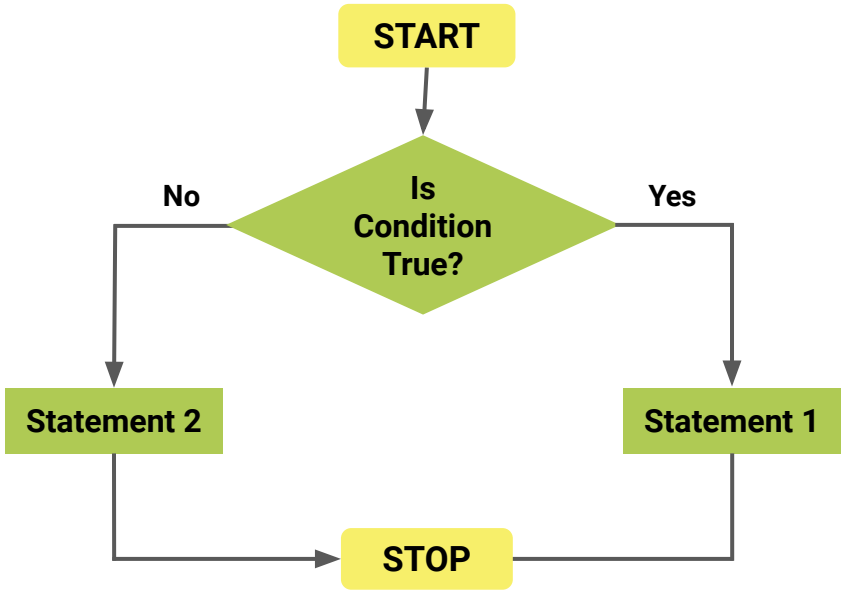
Demystifying Algo Trading

A trading algorithm can be as simple as a loop and a conditional.

Loop

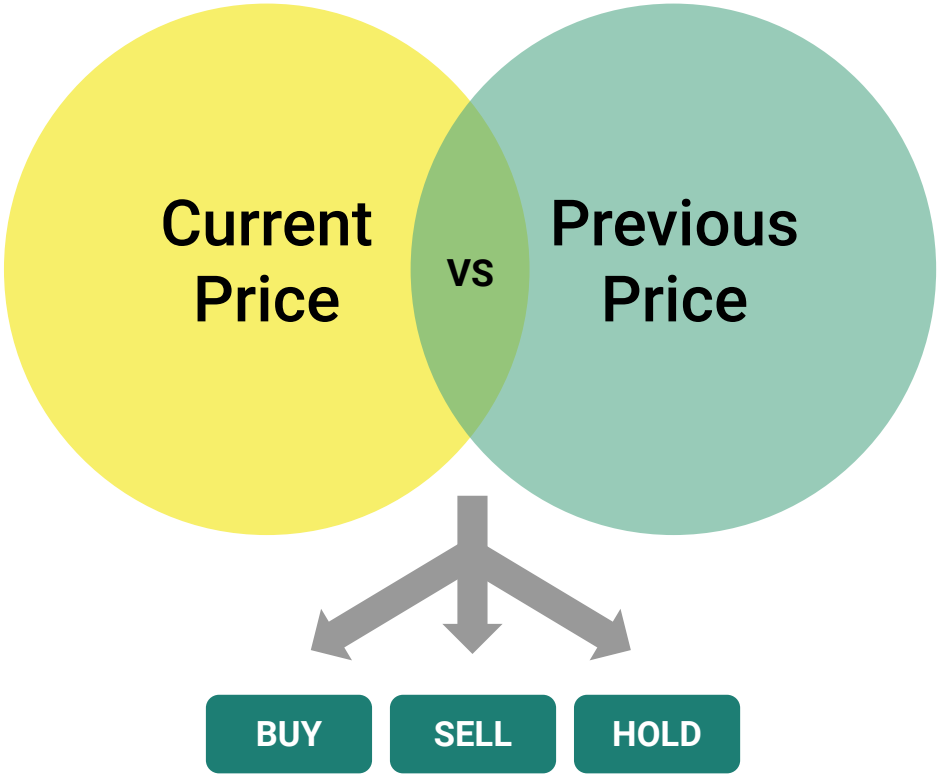


Conditional



Demystifying Algo Trading

This trading strategy iterates through the data and compares the **current price** to the **previous price** to make a buy, sell, or hold decision.



Group Activity:

A Simple Trading Algorithm

In this challenge, we will demystify algorithmic trading by showing a simplified trading strategy in code.

Suggested Time:
10 Minutes

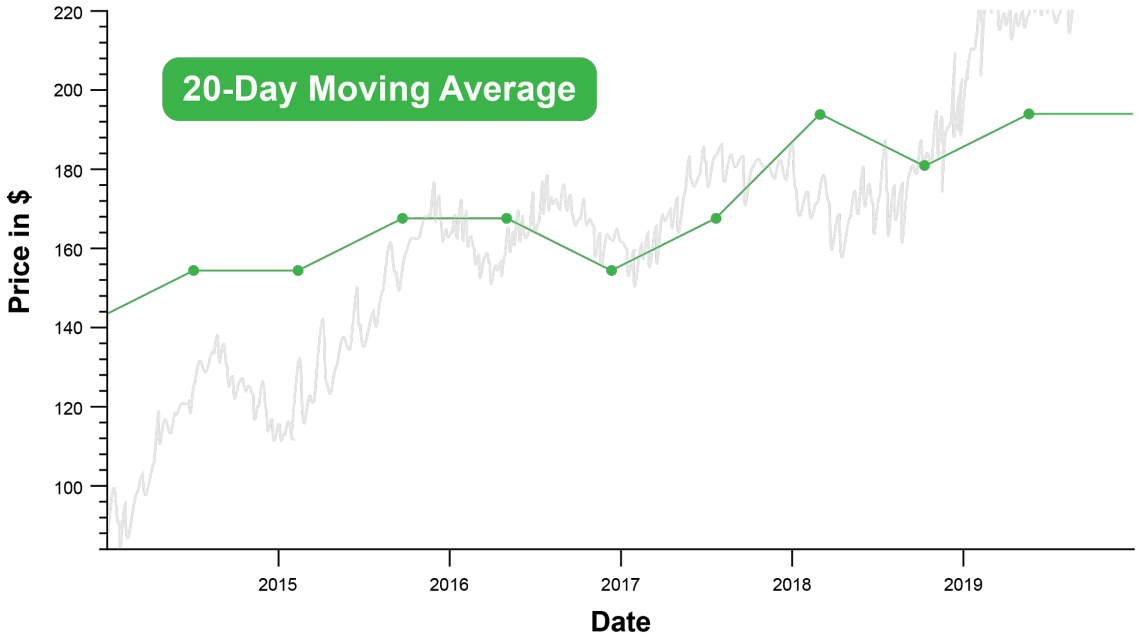
Trading Signals

What Is a Technical Indicator?

Trading Signals: Technical Indicator

Falling under the umbrella of technical analysis, a **technical indicator** is a data-driven metric that uses trading data such as closing price and volume to analyze the short or long-term price movements occurring over a specified period of time.

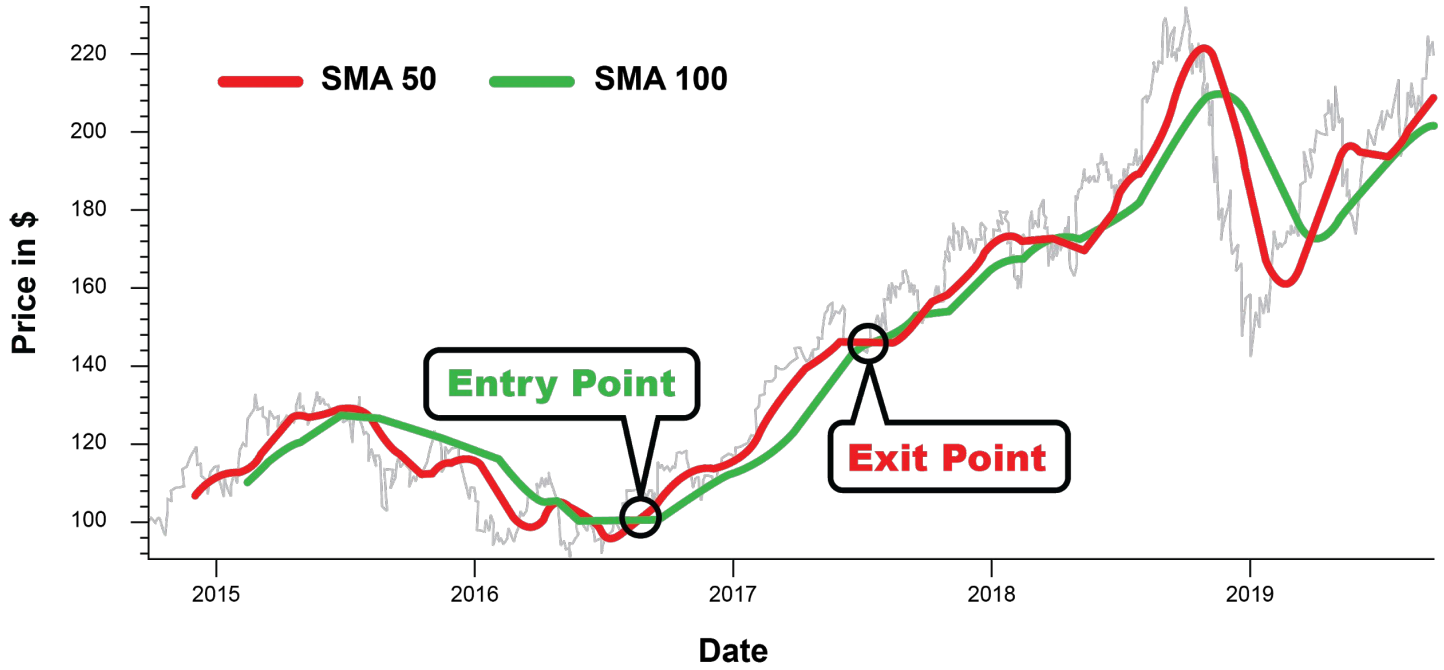
For example, a 20-day simple moving average is a technical indicator representing a rolling 20-day mean of a stock's closing prices.

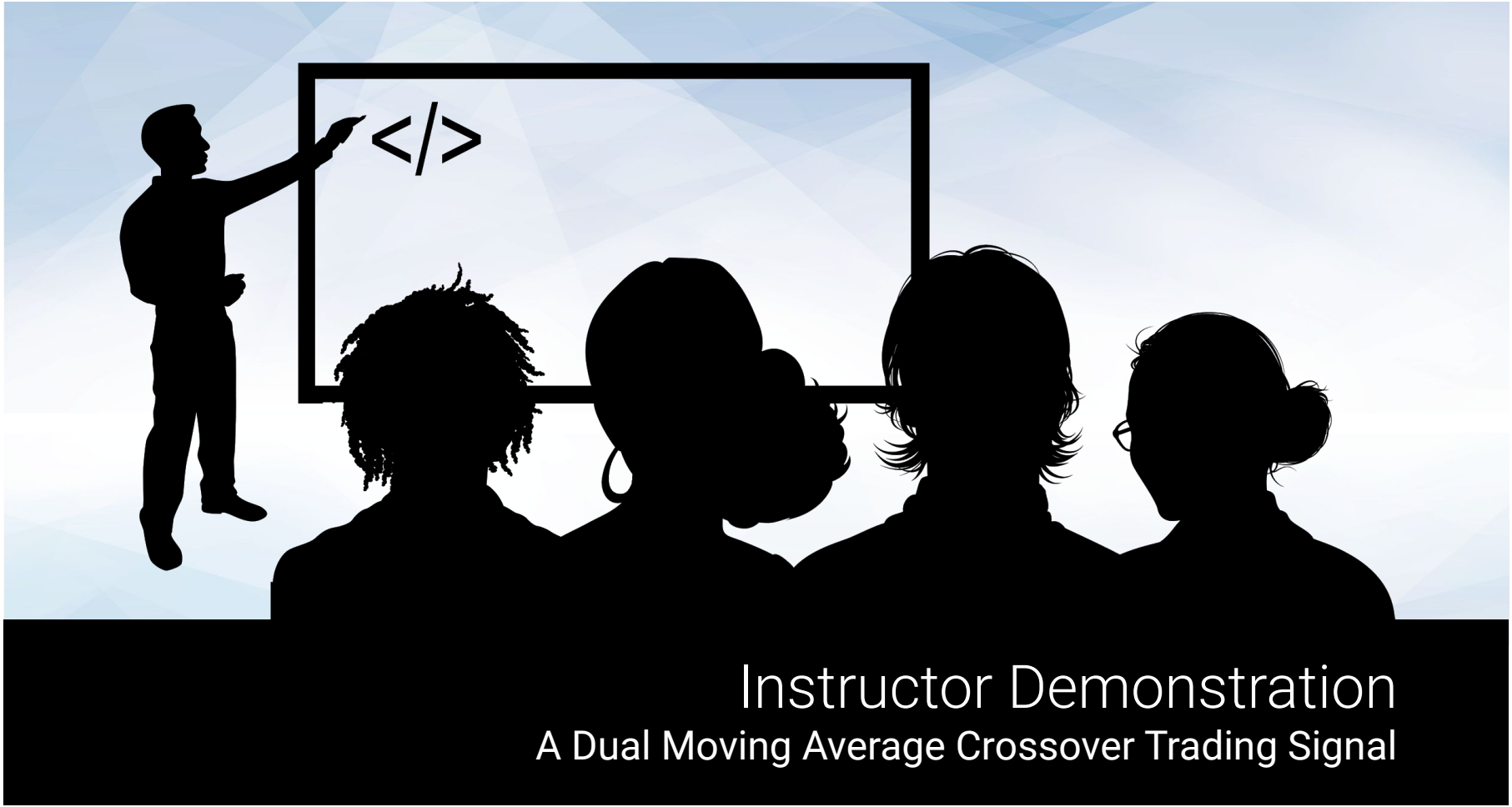


What Is a Trading Signal?

Trading Signals: Trading Signal

A **trading signal** is the point at which a technical indicator, such as the crossover of two moving averages (short MA and long MA), suggests an opportunity for action—namely whether an individual trader or algorithmic trading program should issue a buy or sell order for a security (such as a stock) at that point in time.

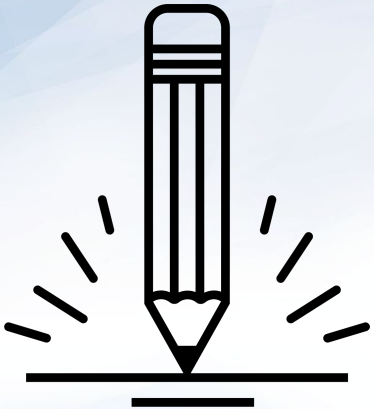




Instructor Demonstration

A Dual Moving Average Crossover Trading Signal


THE
BIG
SHORT



Activity: The Big Short

In this activity, you will take what you've learned about generating trading signals and implementing a corresponding trading strategy to perform the inverse:

Create a trading strategy that profits off of price declines by shorting (selling) and covering (buying) the stock.

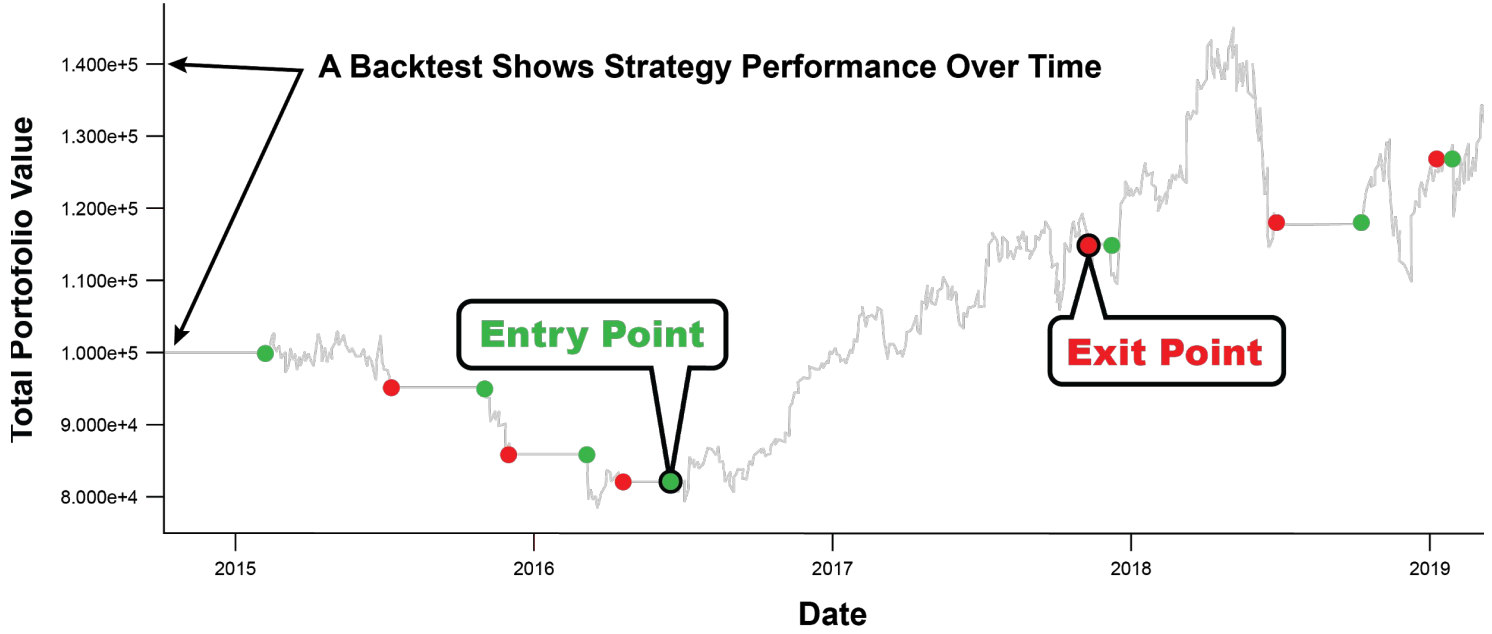
Suggested Time:
15 Minutes

Time’s Up! Let’s Review.

Backtesting

Backtesting

Backtesting is the process for measuring the overall performance of a trading strategy using historical stock prices to simulate executed trades dictated by the calculated trading signals and trade decision logic.



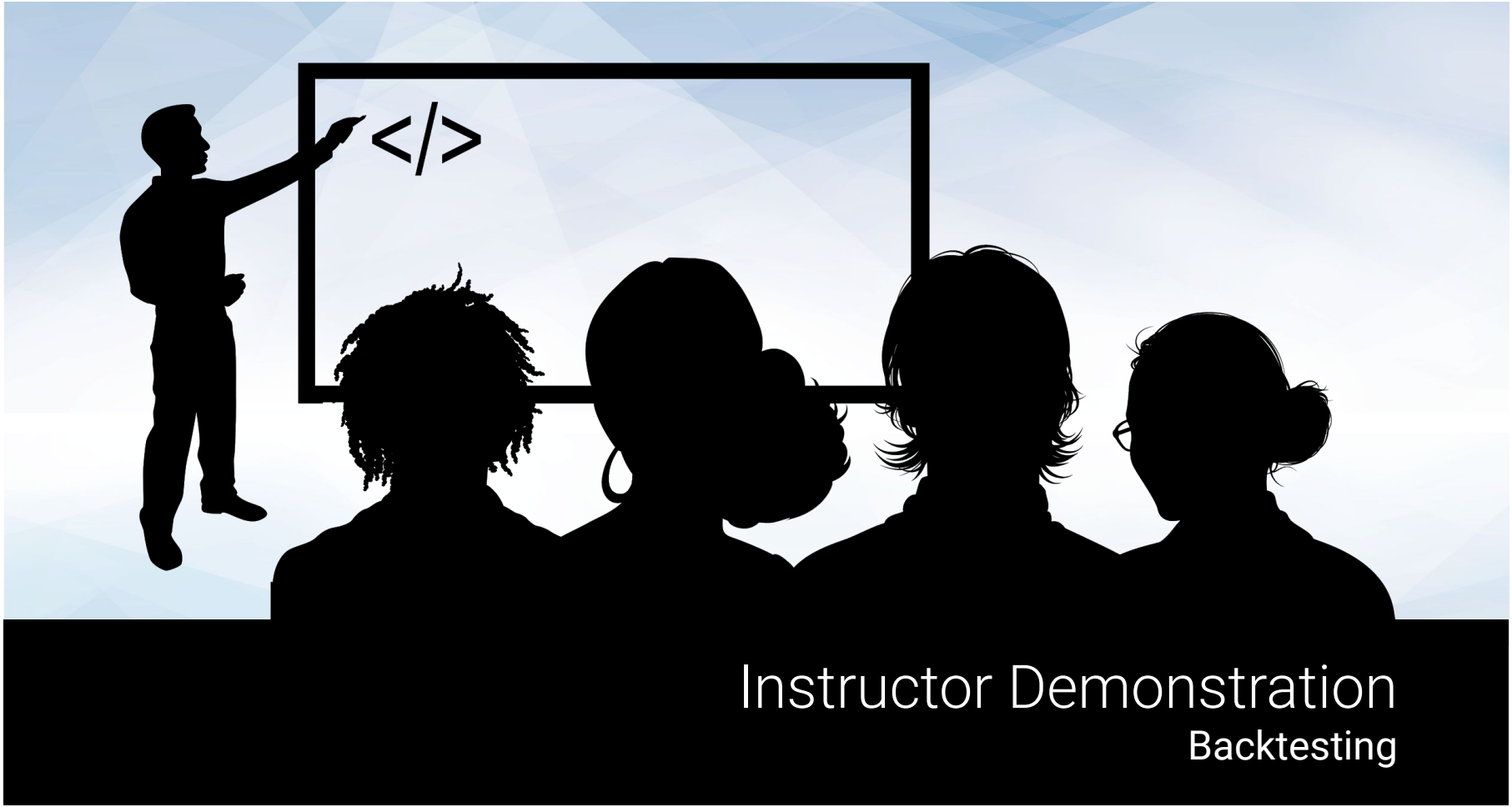
Why Do You Think Backtesting Is Important?

Backtesting

Backtesting helps to assess the validity or profitability of a trading strategy over time and provides a benchmark for how it may perform going forward.

While backtesting is important, the results of the backtest correspond to historical prices **not** future prices.

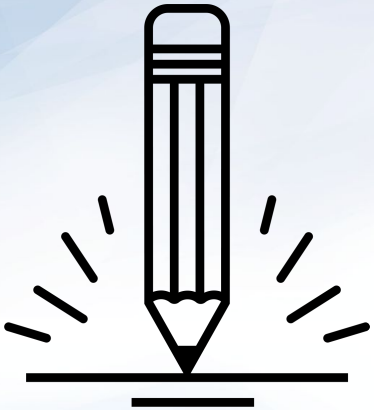
Therefore, backtesting may provide a reliable benchmark for the stock prices that have already occurred, but may prove to be less reliable as new stock data arises.



Instructor Demonstration

Backtesting


THE
BIG
SHORT
PART II



Activity: The Big Short Part II

In this challenge, you will take the **short position dual moving average crossover** trading strategy you made in the previous activity and run a backtest to quantify the performance of your trading strategy.

Suggested Time:
10 Minutes

Time’s Up! Let’s Review.


Break

Trading Evaluation

Trading Evaluation

Evaluating securities, trades, and portfolios is important regardless if it is a human or algorithm making the trade. The following metrics can be used to evaluate securities/trades:

01 Cumulative Return

02 Annual Return

03 Annual Volatility

04 Sharpe Ratio

05 Downside Deviation/Return

06 Sortino Ratio

These metrics can be calculated historically with backtesting, or they can be used to measure future trades and opportunities for portfolio growth.

Trading Evaluation

01

Cumulative Return

The total/aggregated amount of gains and losses for an investment. Cumulative return is measured across time and not for a given time period.

Trading Evaluation

02

Annual Return

A time weighted annual percentage representing the return on an investment over a period of time.

Trading Evaluation

03

Annual Volatility

The annualized degree of variation in trading prices over time.

Trading Evaluation

04

Sharpe Ratio

The return of investment compared to its risk, measured by the difference between the return on investment and the risk-free return.

Trading Evaluation

05

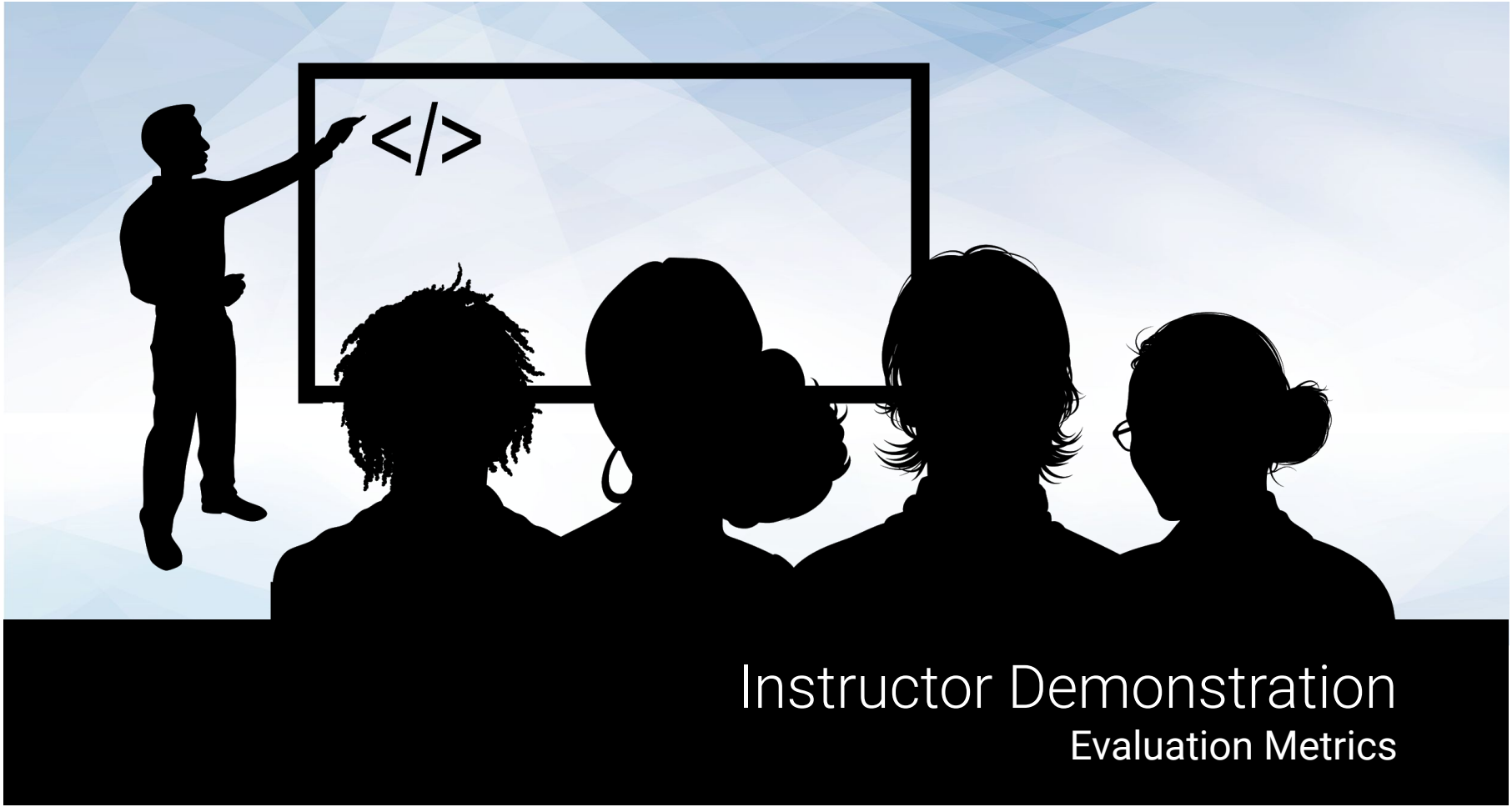
Downside Deviation/Return

The measure of risk for returns that are below the minimal acceptable return.

06

Sortino Ratio

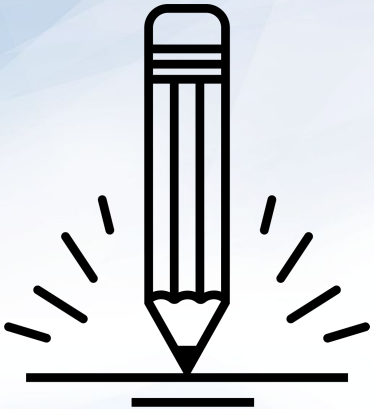
The quotient of harmful volatility and overall volatility. The Sortino ratio focuses on downside deviation rather than standard deviation.



Instructor Demonstration

Evaluation Metrics


THE
BIG
SHORT
PART III



Activity: The Big Short Part III

In this challenge, you will receive a walkthrough of the various evaluation metrics that can be used to evaluate their trading algorithms, namely portfolio and trade-related evaluation metrics.

Suggested Time:
10 Minutes

Time’s Up! Let’s Review.

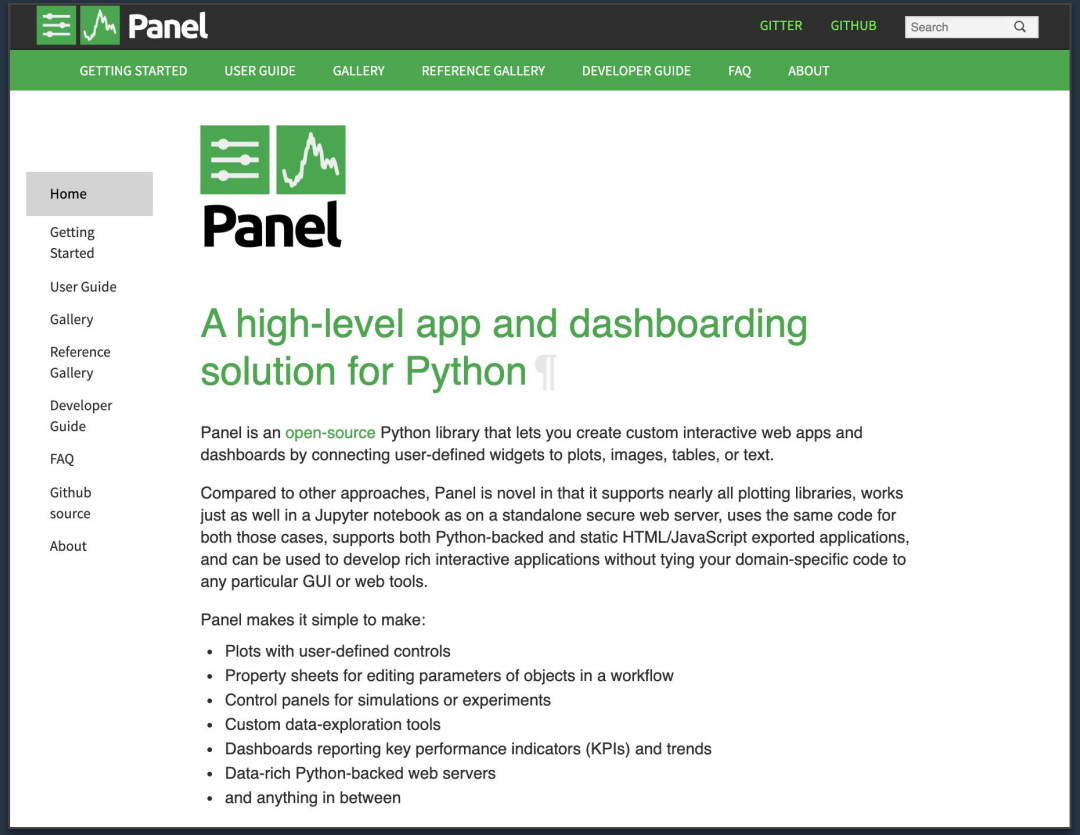
Trading Dashboards in Python

What Is Panel?

Trading Dashboards in Python

Panel is a high-level dashboarding library that allows a user to create custom interactive web apps and dashboards by connecting user-defined widgets, plots, images, tables, or text.

In addition, Panel works with other visualization libraries such as Plotly Express or Hvplot via extensions.



(Panel)

What Is a Dashboard?

Trading Dashboards in Python

A **dashboard** is an information management tool that organizes, stores, and displays important information from multiple data sources into one, easy-to-access place.

Why Use a Dashboard?

Trading Dashboards in Python

Having a **single interactive interface** for key information and visualizations allows a user to monitor the health of an existing process, business function, or application, and therefore aids in measuring performance and identifying any discrepancies that may arise.

Trading Dashboards in Python

Volunteers! Who'd like to summarize one of the following concepts:

What is algorithmic trading? What does the process entail?

What are trading signals?

What is backtesting? Why is it important?

What kinds of portfolio evaluation metrics exist?

What kinds of trade evaluation metrics exist?

Give yourselves a round of applause.

You’re on your way to becoming
your own investment fund managers!

AdobeStock_237395461

Trading Dashboards in Python

By learning how to algorithmically trade, you're already way ahead of those who only know how to manually invest. You now have the tools to do both, and can choose which method you prefer:

01

Algorithmic Trading

02

Manual Trading

03

A Hybrid of Algorithmic Trading and Manual Trading

Group Activity:
Trading Dashboard

In this challenge, you'll create a trading dashboard with Panel using the evaluation metrics generated from previous activities.

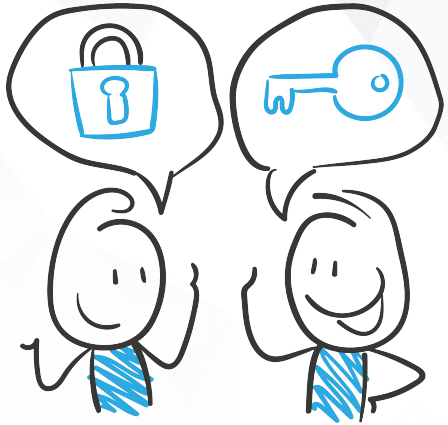
Suggested Time:
15 minutes

Reflect

Next Steps

Create an Algorithmic Trading Framework

Take what you've learned today to create a full-fledged trading application that you can use in real-world scenarios.



Take a moment to reflect on what you just learned.

Questions?


The End