# A Quick Start-Up Guide

To run basic back-tests, here are a few steps that should get you up and running

* Install standard modules using pip – *yfinance* (check Yahoo Finance website for updated installation procedures), *pandas, datetime, numpy,* etc
* Within the Production folder, it is imperative that the user create a folder called “*Indicator\_CSVs*” (case-sensitive)
* While running the tests, navigate to the production directory, and run “tests.py” from command line only (do not run using the “run” command on standard code editors)
* It is useful to run the command – *python test.py* ***--update*** (--update is used whenever the data needs to be updated, for new runs, or after tweaks to the features. For newer users, it’s a good heuristic to always run the file in --update mode)
* The arguments in the backtester class are the features in the model. These can be changed by the user. The argument - *current\_account\_size\_csv* allows the user to change the name of the generated file. This file can be formatted into excel, for further analysis.
* The trade logs are saved in a file called *backtest-results.txt*, which can be found in the Production folder.

# INTRODUCTION AND MOTIVATION

Investment/Rebalanced Portfolio type Funds and Trading Type Funds

The fund management sphere can be classified into two broad classifications based on the time period of holding positions.

1. **Rebalanced Portfolio funds** – where positions are held for a relatively long time and most changes in the portfolio may be triggered by a variety of factors from value/accounting metrics to macroeconomic trends in an endless spectrum of available financial data. Quant strategies in this sphere may include **sector-beta**, **relative value strategies**, or **smart–beta** type models that can be seen in ETFs.
2. **Trading Type Funds** – Position-hold times are generally shorter, and they tend to be opportunistic, as opposed to having a bias on the investment style. All trend following strategies are considered **“Trading Type” strategies**. Other quant strategies in the space may include **pairs trading, spread arbitrage, high-frequency trading**, etc. The rest of this document deals with quant strategies in the trend following space.

BACKGROUND

TREND FOLLOWING

Trend following is the strategy in which the system attempts to react after a price move rather than predicting future price movements while catching and holding on (“riding”) the trend until the trend has been proven to have “exhausted” itself. Entry and Exit logic may either be based on hard quantitative metrics or qualitative ones.

A large price dislocation or a protracted price move in certain direction indicates that the trend will continue to move in the same direction, enough for the systematic trader to profit from. This is the only prediction that is inbuilt into this methodology.

Trend following is suitable for investors who believe that the price of an asset is pricing all complex data. Therefore, trend followers rarely use any data other than historical price data. The implementation of these strategies in industries are often very simplistic and are rarely optimised for the best absolute returns or risk adjusted returns.

TURTLE TRADERS

The simplest trend trading strategy was devised by Richard Dennis and William Eckhardt in the 1980s. His class of traders are popularly known as “Turtle Traders”. Turtle trading is often referred to a trading strategy where the trader buys a stock during a breakout and immediately sells when a retracement in price is observed.

The caveat to this system is that it is optimised to fit all forms of return distribution. While it may be useful to train a machine learning model on changing the lookback periods on the system variables, the return distribution rarely shows too much deviation. Furthermore, the simplicity of the system causes it to be rigid, within the context of diversification.

Strongly enforced diversification is beneficial but it disallows variable position sizing and is far less opportunistic than an optimised model has the potential to be.

*More information about their system can be found here:* [*Turtle Trading System (definedge.com)*](https://www.definedge.com/turtle-trading-system/)

# GENESIS OF THE IDEA

Concentrated Trend Following

Robert Carver, author of the book *Systematic Trading:A Unique New Method For Designing Trading and Investing Systems* rightly explains the need for systematic trading, given in-built cognitive biases of humans. He explains the different types of systematic investing— asset allocating investing, semi-automatic trading, staunch systems trading. Our Model aims to follow a staunch systems trading pattern with no involvement of our team.

Trend systems in the industry use varied methods including volatility-based sizing and enforced market neutrality in order to increase exposure to the best possible opportunity. The major drawback of these implicit leveraging strategies, however, is that there is **no possible mechanism to quantify** the level of **expected outperformance**.

Our model (detailed in OptimisedModel.py) attempts to quantify, the best possible **“opportunities”** within the available basket of securities by constructing an **opportunity list** for every trading day. Any new position that the model takes is directly from this list of “opportunities”. This is in stark contrast to normal trend following systems, where the level diversification rarely changes.

This type of model as works better in relatively inefficient markets, like those in India. However, preventing diversification on the basket level opens the system to other forms of diversification:

* + *First Level of Diversification***:** **Diversifying across strategies** can be considered an equivalent to **diversifying across asset classes**, where they are equal to different return drivers

Considering that all models use trend following, there might be situations in excessively trending markets (assuming that separate strategies are trading the same basket of stocks) where the correlations between all the strategies point to one strategy. In corollary, a few securities may be observed to have a very large allocation within that part of the portfolio.

* + *Second Level of Diversification***: Diversifying across baskets of different securities**

A number of models could be constructed in order to first construct a basket of securities. Previously used linear trend following models can be added as an overlay. **Basket Construction** can be used as an analogue to **factor investing**; with a **trend following overlay**.

* + *Third Level of Diversification:* **Diversifying across Global Markets** using the same metrics and models but with a long-time frame of rebalancing (for example: a time period of one year as opposed to one week)
  + *Fourth Level of Diversification:*Besides moving across asset classes and strategies, **miscellaneous systems of diversification** such as intraday trading systems and leveraged trading systems may also be incorporated.

# FEATURE INDICATOR TYPES

Type of Data

As stated already, most exogenous variables that may be used as features in other investing styles— such as valuation metrics or macroeconomic metrics— are not generally used since these data sources are considered already baked into the price data. The format used for most trend following programs are **OHLC** or **OHLC + Volume Data Arrays** (*OHLC – Open, High, Low, Close Prices within a trading day*)

Often times, in the interest of encapsulation, the *typical price* of the day is used. The typical price of a day is most often the average price of the given trading day. The typical price is the average of the High, Low and Close in most programs that draw OHLC data. Open Prices are not included here to prevent the idiosyncrasies of gap-ups and gap-downs being baked into the data.

Sourcing Indicators

Most of the indicators prototyped are well known indicators and utilized by all general trading platforms. They are of 2 types— **trend following indicators** or **mean reversionary indicators**. Initial tests of these indicators were consistent with tests done by other market participants. These tests showed mean reversionary indicators were more effective in the shorter term while trend following indicators were more credible in the longer term. However, these tests also exposed many caveats of building systems with several varied indicators. The primary observation in this regard was that most trend indicators were highly correlated and hence not computationally viable. Using highly correlated indicators also failed to provide any new insights into the current state of the opportunity value of the stock/security.

Simplicity vs. Complexity

The best example for the simplest trend following system can be found in the turtle traders trading system. In this case, there aren’t enough degrees of freedom in terms of using the variables of the function, to optimise the system. However, initial tests of the system were overcomplicated, primarily because the model was using indicators that were highly correlated. Furthermore, the model compensated with overfitting biases. Granular training (over a training period) provided rather noisy and misleading results. All useful trading systems, use condensed data, and employ uncorrelated data as much as possible.

*For viewing the current model, navigate to the “OptimisedModel.py” file in the “Engines” sub-folder within the Production folder.*

# NATURE OF VARIABLES

The Family of Models

The Model that has been tested is just one among several other types of models (*see, Second Level of Diversification, above*) that can be constructed on similar data streams (OHLC, condensed down to typical price arrays).

Each model might have different methods of optimised for best possible return, or for other characteristics such as leveragability, best positive carry returns, etc.

One Instance of a Model

The Model, in its current state, attempts to thread the knife’s edge between simplicity and complexity. The Model attempts to construct an opportunity list from the given basket of securities by using three levels of filters:

1. **Breakout Filter** – Filters out all securities that do not show either a positive or negative price deviation, above or below a multiplied standard deviation value measured from the moving average value; and, storing the output by condensing down to a signum value
2. **Slope Filter** – Checks for contradictions, between the direction of price dislocation and actually direction
3. **Summing the Signum functions** of variable lookback periods of different normalisation indicators for historical price action within the defined base – lookback

OBJECTIVES

1. The Model aims to **profit from trends in markets** (mainly linear instruments like stocks, futures, bonds). Trend research, and model outputs may be used to inform options based investing, or dynamic hedging strategies. These could be useful frameworks to implement for risk management purposes of corporations, specifically in the commodities space, but also for investing purposes.
2. The absolute returns of the model might also be used as an ordinal proxy to quantify the level of efficiency in a market, or within baskets of securities.
3. Rate of change of absolute returns might also be used as proxies for market events, liquidity events or credit events.

## LIST OF FEATURES

**Base Lookback** – defines how far back the model should look in order to construct the state functions

**Multiplier1** – Value that is multiplied to Base Lookback, to define the second lookback level

**Multiplier2** – Value that is multiplied to Base Lookback, to define the third lookback level

**Linear Regression Filter Multiplier** – Defines the lookback of the slope filter

**Stop Loss Percentage** – Amount that a position has to depreciate in value before the model decides to exit

**Training Period** – Defines the time period window in which the data is grafted. This value has to be greater than base lookback multiplied by Multiplier2.

**Percent risk per Trade** (primary variable to increase or decrease risk taking behaviour) – This defines what percentage of the total portfolio is being risked at a time.

In general, increasing percent risk per trade, and lowering base lookback increases absolute returns (based on previous back-test results).

# PROJECT BRANCHES

Tweaking the Basket – the Factor based Model as Basket Builders

In the future, valuation metrics, and macroeconomic variables may be used to construct baskets of securities, to add positive carry characteristics to predictive factor-based models.

Sentiment Data

Sentiment Data, and relative correlation data may be used to inform the model, about how to vary the percent risk per trade, based on external market, or consumer sentiment.

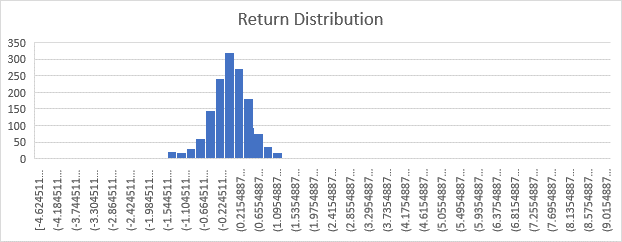
# A FEW BACKTEST RESULTS

1 January 2015 to 6 June 2021 – S&P 500

**ACCOUNT SIZE GRAPH**

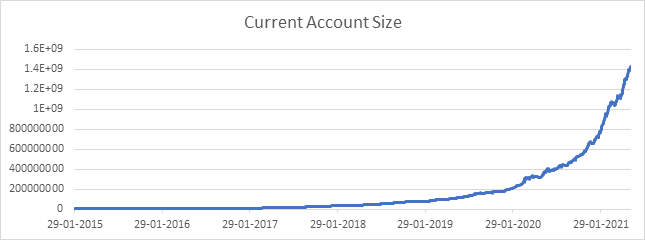
**DAILY CHANGE**

**RETURN DISTRIBUTION**

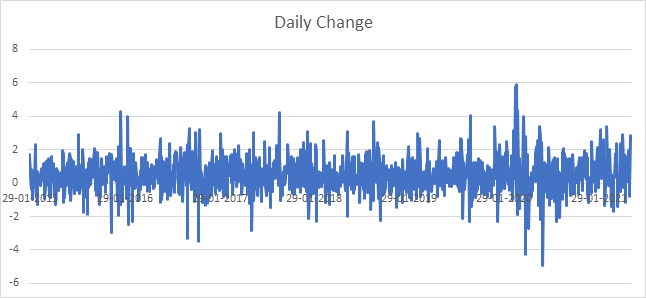


1 January 2015 to 1 June 2021– 700 Largest Indian Stocks – Long Short

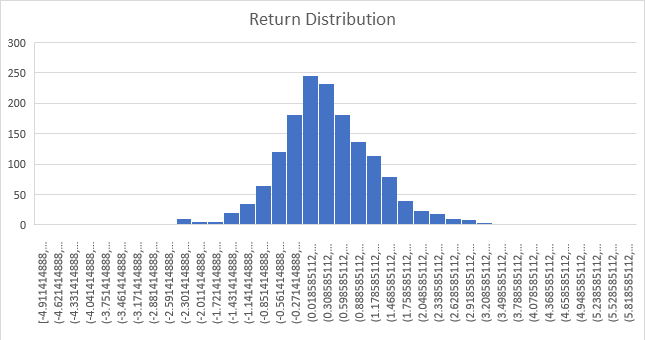
**CURRENT ACCOUNT SIZE**



**DAILY CHANGE**

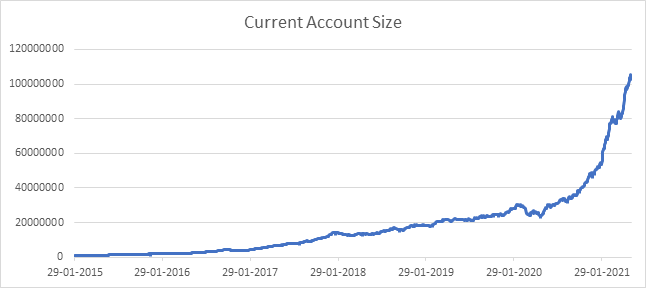


**RETURN DISTRIBUTION**

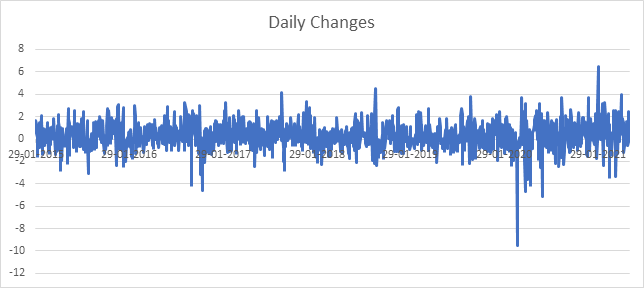


2015 – 01 – 01 to 2021 – 06 – 01 – 700 Largest Indian Stocks – Long Only After Transaction Costs (Realistic Absolute Returns)

**CURRENT ACCOUNT SIZE**



**DAILY CHANGE**



**RETURN DISTRIBUTION**

