# 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 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 of the changes in the portfolio, may be triggered by a variety of factors, whether that be value/accounting metrics, macroeconomic trends, or anything else, 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 be pairs trading, spread arbitrage, high-frequency trading, etc. The rest of the document, deals with quant strategies in the trend following space.

Trend Following

Trend following is the strategy where, instead of predicting future price moves, the system attempts to react after a price move, and hold on for the rest of 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. It must be noted that, the only prediction that is inbuilt into this methodology, is that 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.

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 in the 1980s. His class of traders are popularly known as Turtle Traders.

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

The obvious caveats to this system, are that the system, cannot be optimised to fit any form 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 in terms 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.

# Feature Indicator Types

Type of Data

As stated before, most exogenous variables, that maybe used as features in other investing styles, such as valuation metrics, or macroeconomics metric are not generally used. 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 is generally the average price of the given trading day. In most programs, which draw OHLC data, the typical price is the average of the High, Low and Close. Open Prices are not included here, to prevent the idiosyncrasies of gap-ups and gap-downs from being baked in to the data.

Sourcing Indicators

Most of the indicators prototyped, are well known indicators, usable from most trading platforms. They were either trend following, or mean reversionary indicators. Initial tests of these indicators, were consistent with tests done by other market participants, which show that mean reversionary indicators are more effective in the shorter term, and the trend following indicators are more useful in the longer term.

However, the tests also exposed a lot of caveats of building systems with several indicators. The primary observation in this regard was – most of the trend indicators were highly correlated, and hence computing all of them wasn’t computationally viable. Using highly correlated indicators also fail 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 over complicated, primarily because the model was using indicators that were highly correlated, but importantly, the model itself had a lot of overfitting bias. Granular training, on a training period provided very noisy and misleading results.

All useful trading systems, use condensed data, and use as much uncorrelated data as possible.

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

# Genesis of the Idea

Concentrated Trend Following

* Trend systems in industry use several methods, including volatility based sizing, and enforced market neutrality in order to increase exposure to the best possible opportunity.
* A major drawback of these implicit leveraging strategies, is that, there is no way of quantifying the level of expected outperformance.
* The model (detailed in OptimisedModel.py) attempts to quantify, the best possible “opportunities” in the basket of securities, by constructing an opportunity list for every trading day. Any new position that the system takes, is 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, is equal to diversifying across asset classes, same they are equal to different return drivers. Since 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 go to one. The corollary of this, is that a few securities will be seen have a very large allocation within that part of the portfolio.
  + **Second Level of Diversification:** Diversifying across baskets of different securities. Models could be made, 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 same metrics and models, but with a long-time frame of rebalancing (e.g.: 1 year)
  + **Fourth Level of Diversification:** Besides moving across asset classes and strategies, miscellaneous systems of diversification such as intra day trading systems, and leveraged trading systems

# 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.

It 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

## 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 – For now, this defines what time period window 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

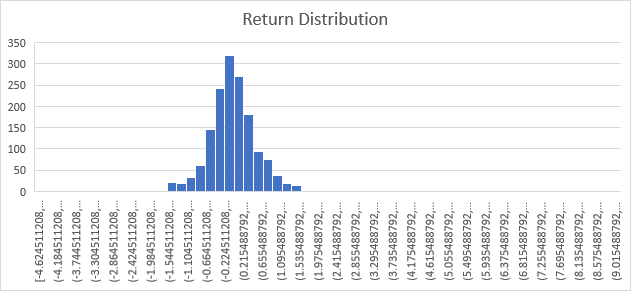
In the future, valuation metrics, and macroeconomic variables may be used to construct baskets of securities, to add positive carry characteristics to predictive factor based models.

Sentiment Data

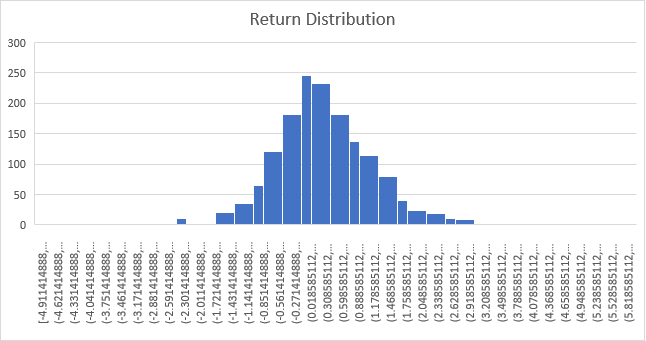
Sentiment Data, and relative correlation data may be used to inform the model, about how to vary the percent risk per trade, based on external market, or consumer sentiment.

# A Few Back-test Results

2015 – 01 – 01 to 2021 – 06 – 01 – S&P 500 – Long Short



2015 – 01 – 01 to 2021 – 06 – 01 – 700 Largest Indian Stocks – Long Short



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

