Quant Finance 101

CHICAGO QUANT CLUB

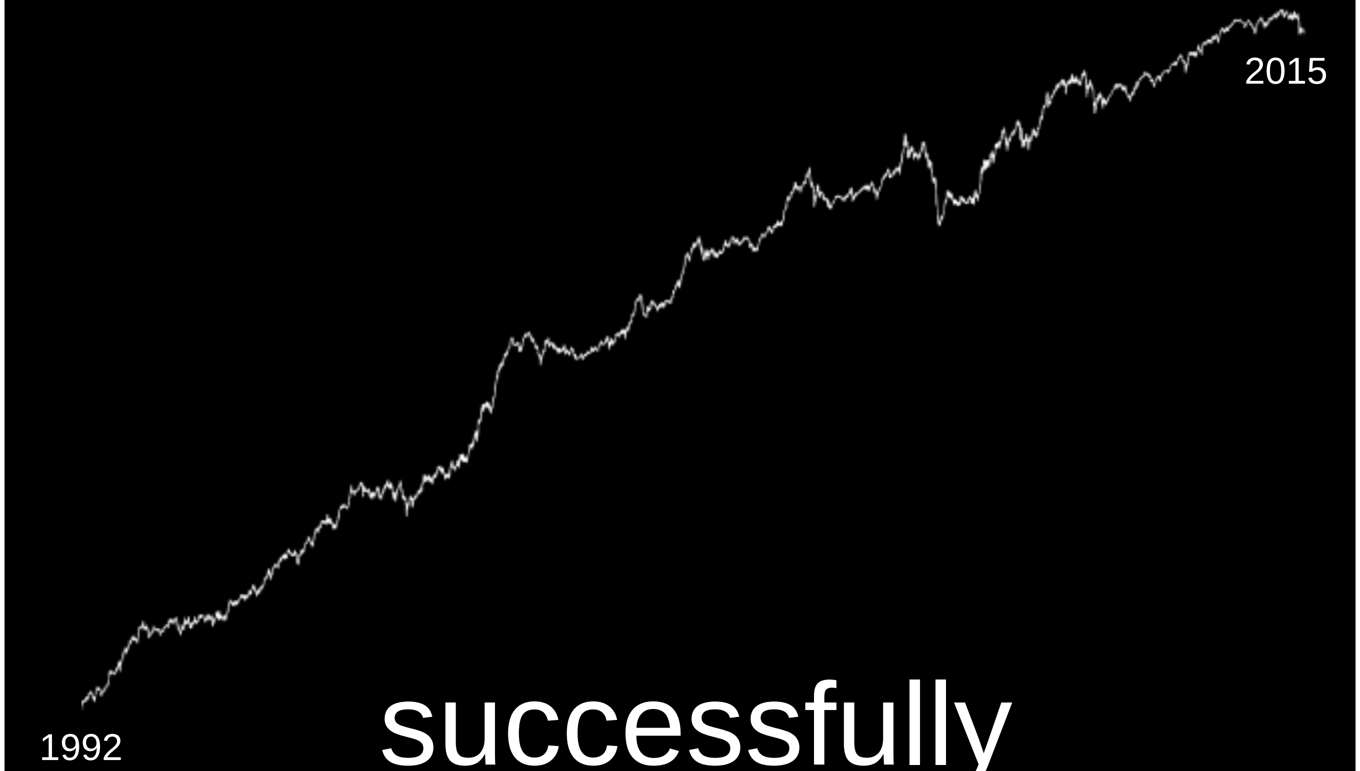
MARTIN FROEHLER | NOBIE REDMON

```
import numpy as np

def mySettings():
    settings={}
    settings['markets']     = ['CASH', 'F_AD', 'F_BO', 'F_BP', 'F_C', 'F_CC', 'F_CD', 'F_CL',
'F_CT', 'F_DX', 'F_EC', 'F_ED', 'F_ES', 'F_FC', 'F_FV', 'F_GC', 'F_HG', 'F_HO', 'F_JY',
'F_KC', 'F_LB', 'F_LC', 'F_LN', 'F_MD', 'F_MP', 'F_NG', 'F_NQ', 'F_NR', 'F_O', 'F_OJ', 'F_PA',
'F_PL', 'F_RB', 'F_RU', 'F_S', 'F_SB', 'F_SF', 'F_SI', 'F_SM', 'F_TU', 'F_TY', 'F_US', 'F_W',
'F_XX', 'F_YM']
    settings['slippage']    = 0.05
    settings['budget']      = 1000000
    settings['lookback']    = 504
    return settings

def myTradingSystem(DATE, OPEN, HIGH, LOW, CLOSE, settings):
    rsi1 = RSI(CLOSE,100)
    rsi2 = RSI(CLOSE,500)
    p = ((rsi1 - 50) + (rsi2 - 50)) / 2
    p[p < 0] = p[p <0] / 2
    p[np.isinf(p)]=0
    return p, settings

def RSI(CLOSE,period):
    closeMom = CLOSE[1:,:] - CLOSE[:-1,:]
    up   = closeMom >= 0
    down = closeMom < 0
    out = 100 - 100 / (1 + (np.mean(up[-(period+1):,:],axis=0) / np.mean(down[-
(period+1):,:],axis=0)))
    return out
```

21 Lines of Code

manage

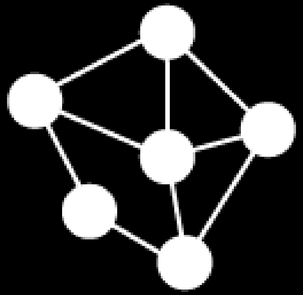
$25 Million



successfully

(5921% over 23 years)

This Is Quantitative Finance



Trading Algorithms developed by scientists

+

High-net-worth individuals and Institutional Investors

$300 Billion Industry

Quantitative Hedge Funds

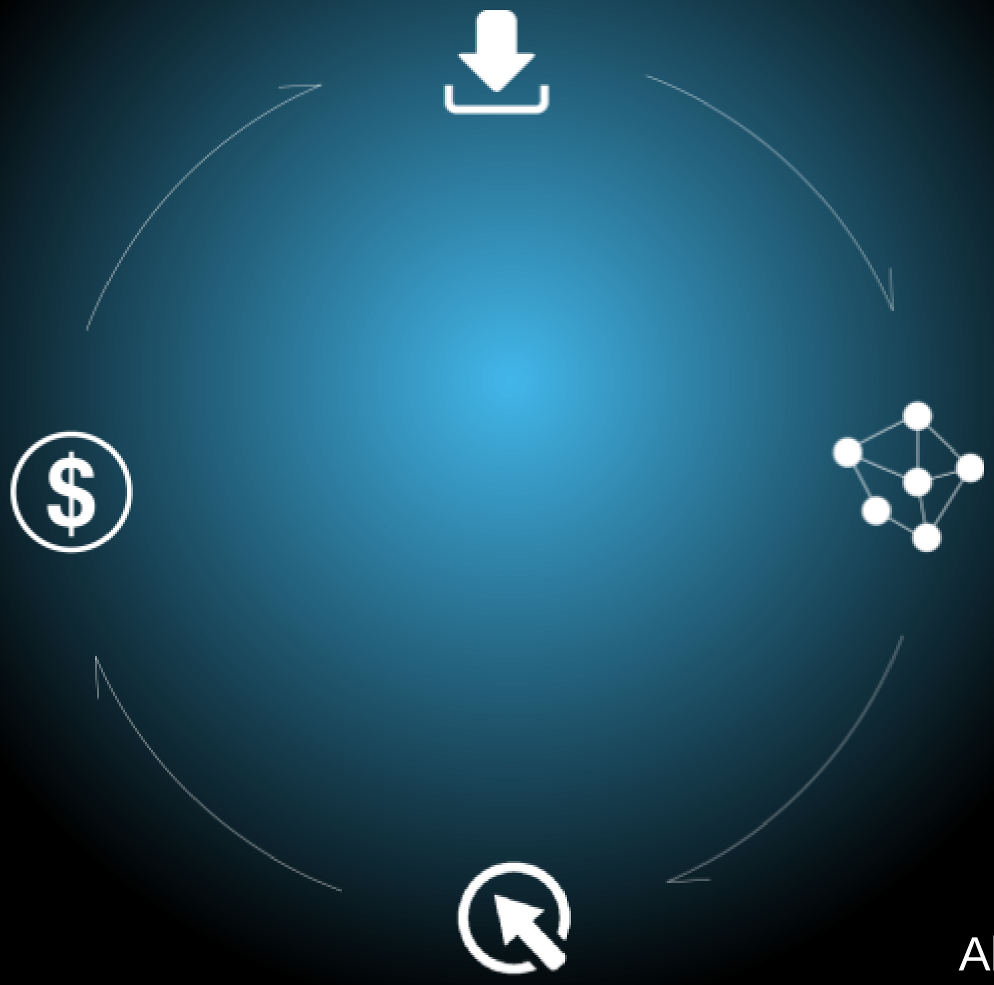
What We Do

Quants

Capital

We connect user generated quantitative trading systems with capital from institutional investors. Our users pocket 10% of the profits without investing their own money.

How It Works For Quants



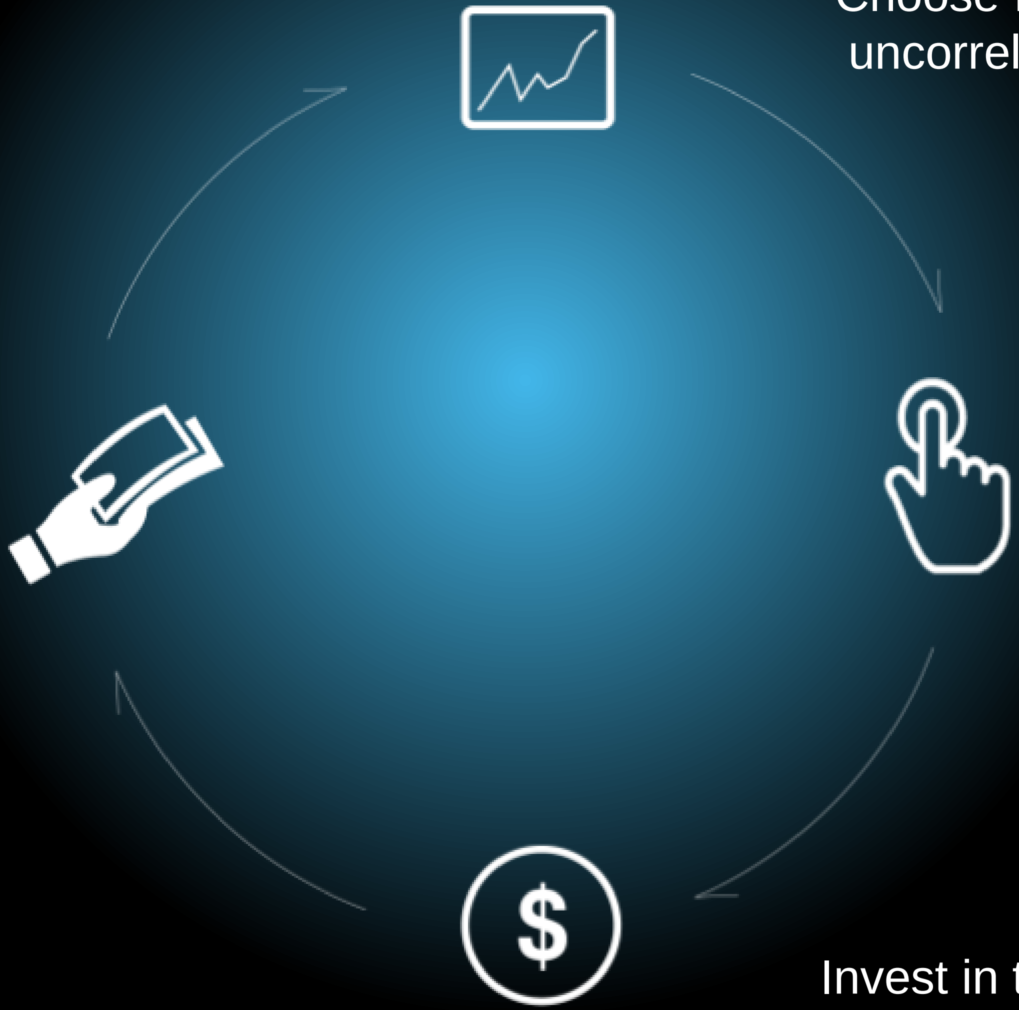
Use Quantiacs’ framework and free financial data (Python, Matlab, Octave)

Develop and test your Trading Algorithm

Submit your Trading Algorithm to market it to investors

Pocket 10% of the profits your system makes without investing your own money

How It Works For Investors



Choose from a huge variety of
uncorrelated trading systems

Pick the system that
complements your portfolio best

Invest in those systems

Pay only the industry typical
performance fee of 20%
BUT NO management fees

Agenda

7:15 Introduction

- Quantiacs Toolbox
- Concepts of quantitative trading

8:00 From 0 to system

- Evaluation – how good is a system
- Best practice and pitfalls

8:45 Q&A

The Toolbox

A framework to develop and test quantitative trading strategies

- In two languages with the same functionality: Matlab/Octave and Python
- It supports the full arsenal of both languages
- Free and open source
- Tweak it, adapt it to your needs and use it in any way you want
- Perform standardized backtests to make results comparable

We have built it for you, please let us know what you‘re missing

Trading System

A trading system is a Matlab/Octave or Python function with a specific template

function [p, settings] = tradingsystem(DATE, OPEN, HIGH, LOW, CLOSE, settings)

The arguments can be selected

DATE … vector of dates in the format YYYYMMDD
OPEN, HIGH, LOW, CLOSE … matrices with a column per market and a row per day.
settings … struct with the settings of the simulation

The return values need to be

p … allcoation of the available capital to the markets
settings … struct with the settings of the simulation

Settings

Use settings to define

- What markets do you want to trade?
- How much data do you need for your trading system?
- Do you want to save some of the data for an out of sample test?
- What is your transaction cost assumption (a.ka. slippage & comission)?

Settings – Matlab/Octave code

Code

```
settings.markets = {'CASH', 'F_ES', 'F_SI', 'F_YM'};
settings.slippage = 0.05;
settings.budget = 1000000;
settings.samplebegin = 19900101;
settings.sampleend = 20161231;
settings.lookback = 504;
```

Settings – Python code

Code

```
def mySettings():
    settings[markets] = ['CASH', 'F_ES', 'F_SI', 'F_YM’ ]
    settings[‘slippage’] = 0.05
    settings[‘budget’] = 1000000
    settings[‘samplebegin’] = ‘19900101’
    settings[‘sampleend’] = ‘20161231’
    settings[‘lookback’] = 504
```

Backtest mechanics

Your TS is called for each (trading) day of the specified backtesting period with the most recent market data as input, and it computes a percent allocation p for the next trading day as output.

The arguments are data matrices of size [nMarkets x settings.lookback] with the most recent market data availalbe at time t. The oldest market data is in row 1, the most recent in the last row of the data matrix.

You can use the full arsenal of Matlab/Octave and Python to compute the positions for the next period.

$p > 0$ … a long position

$p < 0$ … a short position

$p = 0$ … no position

Concepts of quantitative trading

Some quantitative trading concepts and styles:

- **Technical analysis**
- Fundamental analysis
- Sentiment analysis
- News
- Market mechanics

Technical Analysis

Methodology for forecasting the direction of prices through the study of past market data, primarily price and volume

TA uses market indicators of many sorts, most of which are mathematical transformations of price

Core beliefs

- A fundamental principle of TA is that a market's price reflects all relevant information
- Technical analysts believe that investors collectively repeat the behavior of the investors that preceded

RSI – Relative Strength index

Formula

$$closeMom(t) = CLOSE(t) - CLOSE(t-1)$$

$$up(t) = \begin{cases} 1 & \dots \quad if\ closeMom(t) \geq 0 \\ 0 & \dots \quad otherwise \end{cases}$$

$$down(t) = \begin{cases} 1 & \dots \quad if\ closeMom(t) < 0 \\ 0 & \dots \quad otherwise \end{cases}$$

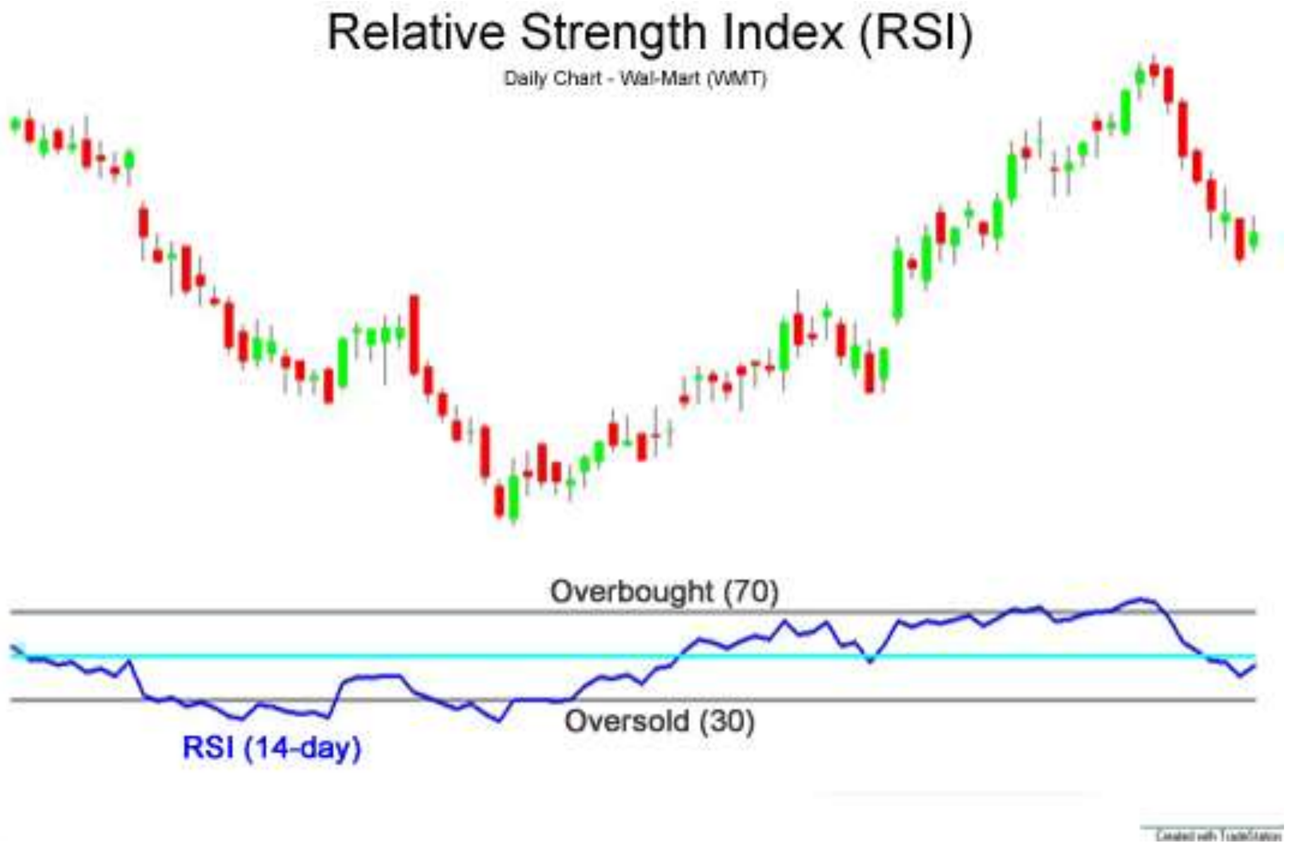
$$meanUp(t, period) = \frac{1}{period} \sum_{t}^{i=t-period+1} up(i)$$

$$meanDown(t, period) = \frac{1}{period} \sum_{t}^{i=t-period+1} down(i)$$

$$RSI(t, period) = 100 - \frac{100}{1 + \frac{meanUp(t, period)}{meanDown(t, period)}}$$

$t \dots index\ of\ the\ trading\ day \qquad period \dots number\ of\ days\ to\ compute\ the\ RSI$

RSI plot


Relative Strength Index (RSI)
Daily Chart - Wal-Mart (WMT)
Overbought (70)
Oversold (30)
RSI (14-day)
Created with TradeStation

How good is a trading system?

There is no universal number that tells you everything about a trading system

There are a lot of things to consider like

- Performance
- Volatility
- Alpha
- Drawdowns
- Correlations

Sharpe Ratio

The Sharpe Ratio is a popular performance to volatility ratio. The Formula:

$$returns_i = \frac{e_i - e_{i-1}}{e_{i-1}}$$

$$\mathrm{i} \; = \; \{2,3, \ldots , \mathrm{t}\}, \qquad t = \text{ number of tradingdays}$$

$$e \; is \; the \; portfolio \; equity \; curve \; of \; the \; Tradingsystem$$

$$volaYearly = \sqrt{252} \; * \; std(returns);$$

$$index_i \; = \; \prod_{i=2}^{t} (1 + returns_i)$$

$$returnDaily = e^{\frac{\ln(index_t)}{t}}$$

$$returnYearly = returnDaily^{252} - 1$$

$$SharpeRatio = \frac{\text{returnYearly}}{\text{volaYearly}}$$

Good practice and pitfalls

Overfitting is the natural enemy of quantitative trading

- It’s easy to fit the known past with enough parameters. Limit the number of your parameters.
- Stability. How does your model react when you change some of the Parameters by 10%
- Save some of the data for an out of sample test

Q&A

Put your skills into practice join our Q4 competition!

The best three futures trading systems submitted to our platform before December 31, 2015 get guaranteed investments of

$ 1,000,000

$ 750,000

$ 500,000

and you get to pocket 10% of the profits.

https://quantiacs.com/q4

QUANTIACS