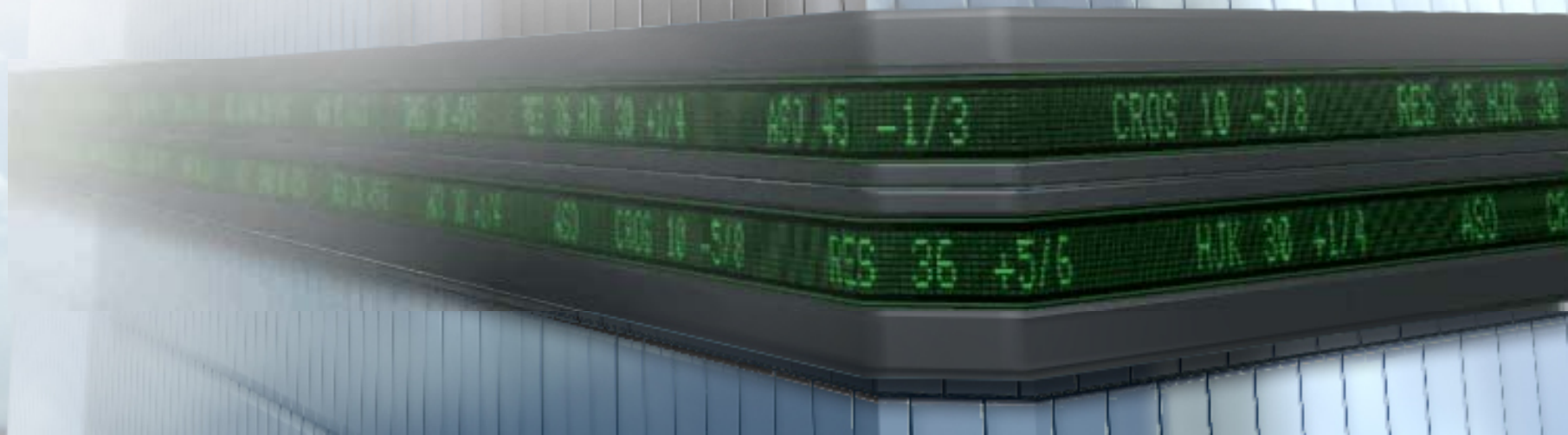
COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

Math 4075

Caitrian Boucher
Boya Liu
Manya Naidu
Andrej Ficnar
Adam Luchansky
Aleks Navratil

TRADING STRATEGY IN SOYABEAN FUTURES



Outline

- Intro and Background
- Statistical Tests & Trend-Following Characterization
- Implementation Details
- Strategy & Optimization
- Risk, Return & Other Results
  - First four years
  - In sample/out of sample
- Conclusions

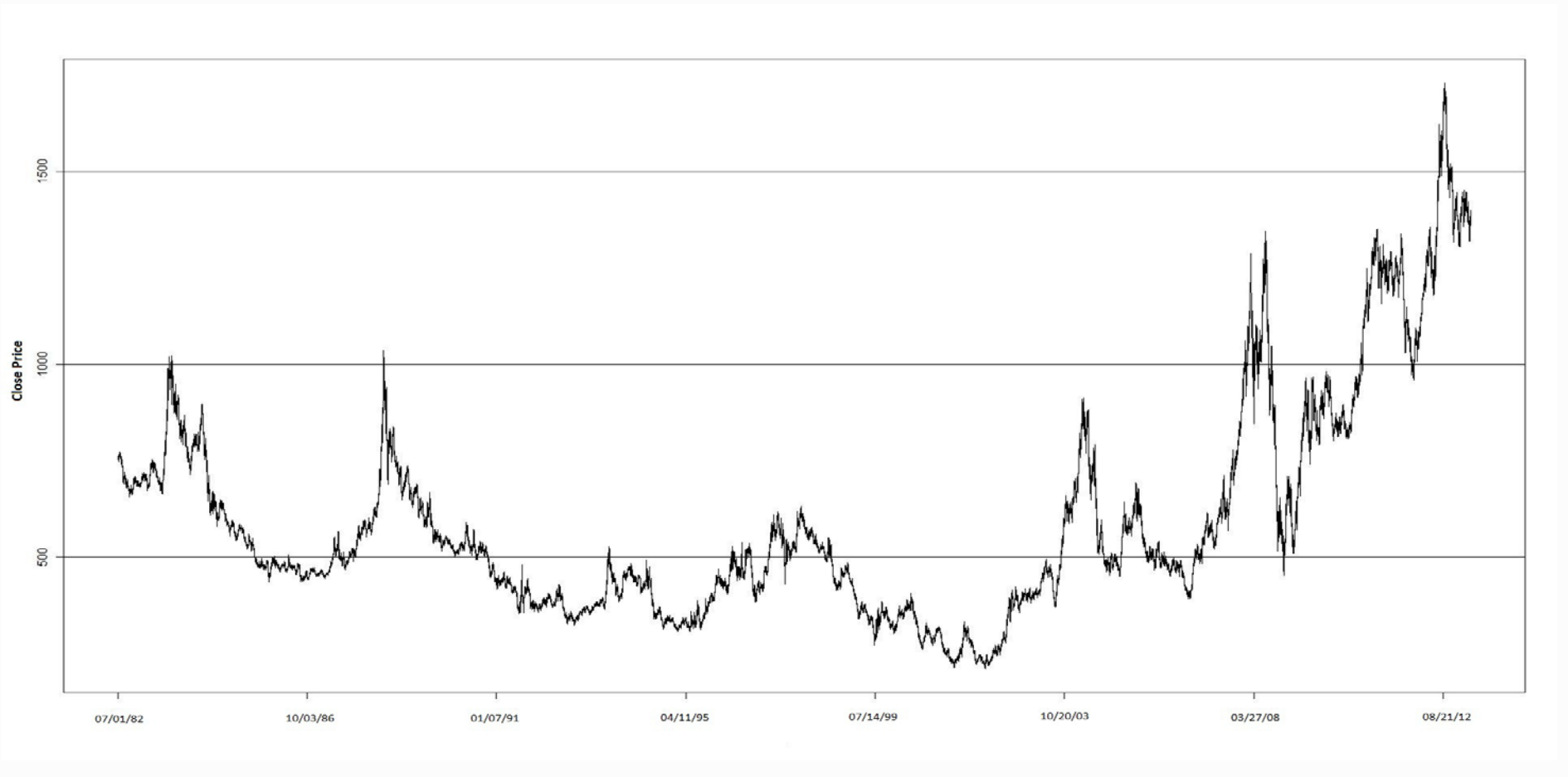
Introduction

Why is there a market for Soybeans?

- ✓ Soybeans are used globally for everything from poultry feedstock to vegetable oil.
- ✓ The US dominates the market, accounting for over 50% of global production. Brazil is the next biggest producer, with 20% of global production.
- ✓ You've probably enjoyed un-ripened soybeans from the pod (edamame)

Soybean Futures Market Specifications	
TickData Ticker	SY
Exchange	CBOT-CME
Currency	USD
Contract Value	71,850
Contract Size	5,000 bushels (~136 metric tons)
Point Value	50
Tick Size	12.50/contract
Data from	7/1/1982
Pit Session	9:30 - 13:15
Bid/Ask spread	12.5
Slippage	24

SY Historicals 1982-2013



Statistical Tests & Trend-Following Characterization

[Variance Ratio Test Results]

$$\rho_1 = corr(\Delta p_1, \Delta p_2)$$

$$\Delta p_1 = p(t+\tau) - p(t)$$

$$\Delta p_2 = p(t+2\tau) - p(t+\tau)$$

$$VR(q) = \frac{Var(q)}{q \cdot Var(1)}$$

$$VR(q) = 1 + 2 \cdot \sum_{k=1}^{q-1} (1 - \frac{k}{q}) \cdot \rho_k$$

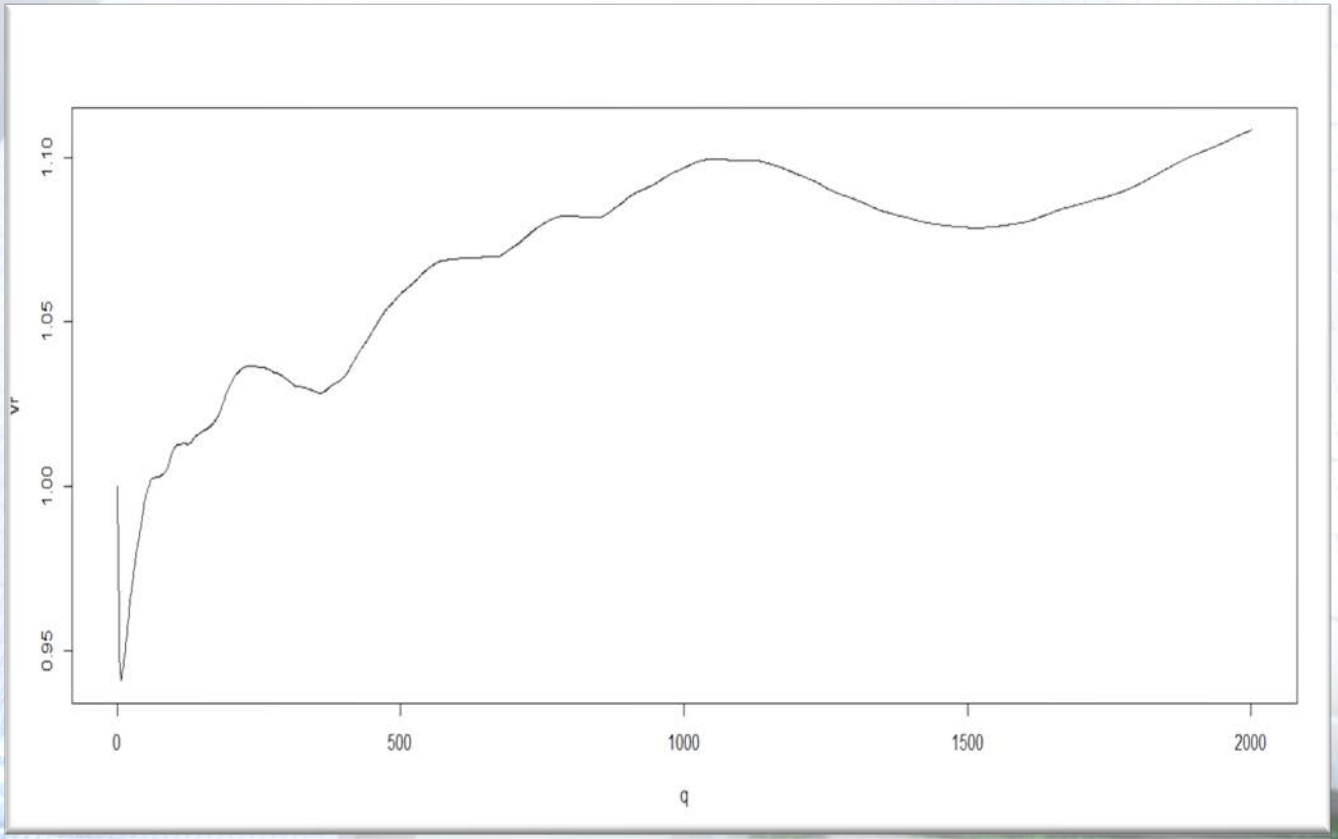
If VR>1, trend following

If VR=1, random walk

If VR<1, mean reversion

Statistical Tests & Trend-Following Characterization

[Variance Ratio Test Results]



The results of the Variance Ratio Test shows mean reversion in short term and trend following thereafter (VR>1)

Statistical Tests & Trend-Following Characterization

$$p(t) - p(t-\tau) \Rightarrow \text{Push}$$

$$p(t+\tau) - p(t) \Rightarrow \text{Response}$$

$$P(y \mid x) = \frac{P(x,y)}{P(x)} \Rightarrow \text{Conditional mean response of a 'push'}$$

$$\langle y \rangle_x = \int dy \cdot y \cdot P(y \mid x)$$

For trend-following property, we look to have on average a positive response following a positive push, and a negative response following a negative push, thus, giving us a positive-sloping data curve.

Statistical Tests & Trend-Following Characterization

[Push Response Test Results]

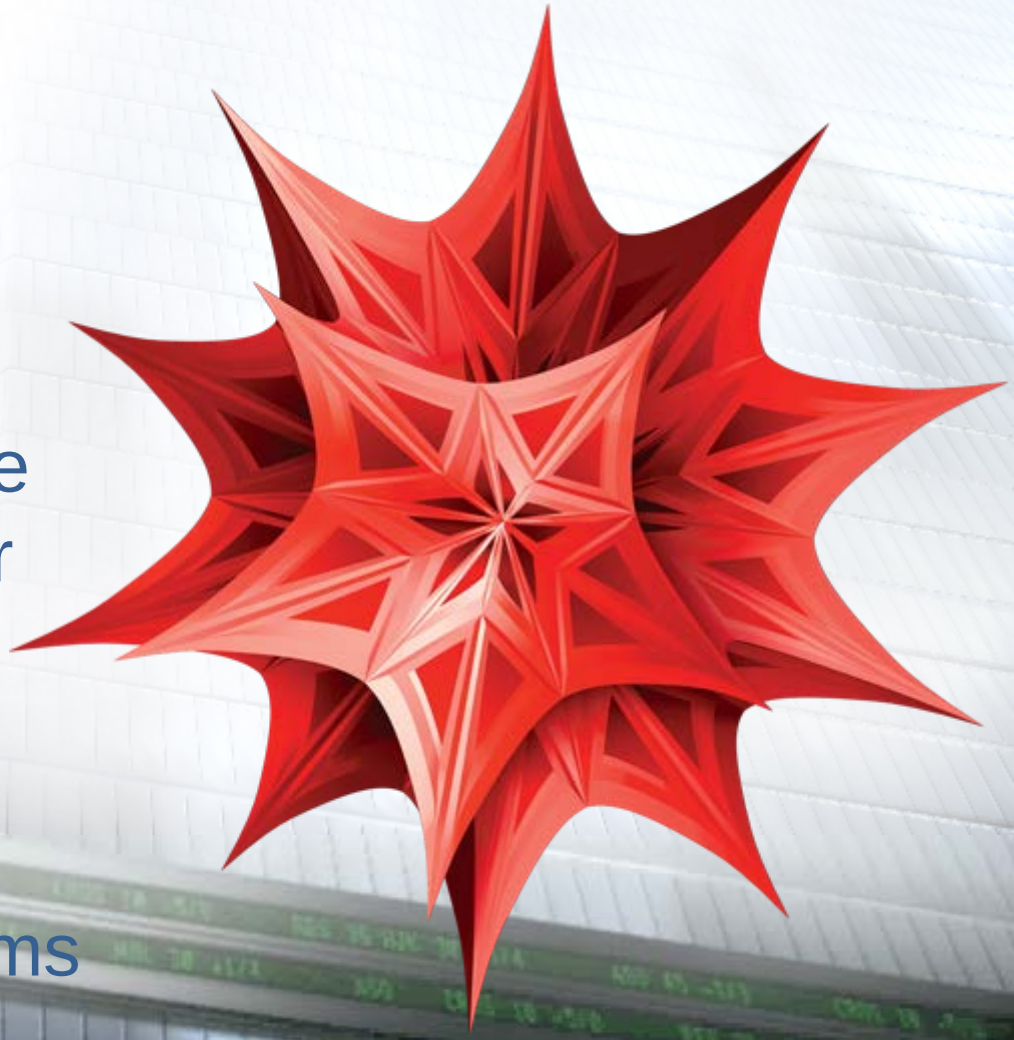
Assumptions

- ✓ There are 45 5-minute OHLC bars in one trading day, and 252 trading days per year
- ✓ Transaction cost remains fixed over the entire time period (7/1/1982 – 4/18/2013) (SLPG=$24)
- ✓ Trading size for each trade is 1 contract
- ✓ An open position must be closed before another trade is initiated
- ✓ No adjustments are made to the risk-free rate or the inflation rate over our time period

Implementation Details

Language Selection: Mathematica

- Pro's:
  - Good parallelization support
  - Some functions can be downcompiled to C for speed
  - Attractive output
  - Support for many programming paradigms



Implementation Details

Language Selection: Mathematica

- Con's:
  - Runs on the Java Virtual Machine---no access to the native instruction set on x86 chips= slow
  - Low efficiency for procedural programming
    - Language designed for pure functional programming (Map[], Scan[], NestList[], etc)

Implementation Details

Modularity and Parallelization

- Code runs on several machines simultaneously for speed
- Independent results stored in .dat files and passed between clustered machines

Very easy to distribute processing in Mathematica

```
MainLoop=Compile[
Function[
foo;
bar;
],
Parallelization->True,
CompilationTarget->”C”
]
```

Magic.

Implementation Details

Timing and Efficiency

- First effort ran in polynomial time $O(n^k)$
- Main function now runs in linear time time $O(n)$
  - After much blood, sweat, and tears
- Still didn't finish all the computations we wanted

Brute-force discrete optimization: in sample

- Divide the parameter space into a grid
- Sweep the grid, optimize for net profit/max drawdown ratio

Optimization Details

- Channel length varied from 10 to 490 in increments of 10
- Channel length also varied from 500 to 10,000 in increments of 100

- Stop percentage varied from 0.001 to 0.1 in increments of .001.

- Produce contour plots for easy data visualization

The main code

- Main loop

In[33]:=

```
RoAList = {};
Do[
 ChNum = Position[ChLengthList, ChLength][[1, 1]];
 (*!!!!!!!!!!!!!!*)
 HHList = Join[Table[HHExt4[[i, ChNum]], {i, 1 + 6 Month, Length[HHExt4]}], Table[HHExt5[[i, ChNum]], {i, 1, Length[HHExt5]}],
   Table[HHExt6[[i, ChNum]], {i, 1, Length[HHExt6]}], Table[HHExt7[[i, ChNum]], {i, 1, Length[HHExt7]}],
   Table[HHExt8[[i, ChNum]], {i, 1, Length[HHExt8]}]];
 LLList = Join[Table[LLExt4[[i, ChNum]], {i, 1 + 6 Month, Length[LLExt4]}], Table[LLExt5[[i, ChNum]], {i, 1, Length[LLExt5]}],
   Table[LLExt6[[i, ChNum]], {i, 1, Length[LLExt6]}], Table[LLExt7[[i, ChNum]], {i, 1, Length[LLExt7]}],
   Table[LLExt8[[i, ChNum]], {i, 1, Length[LLExt8]}]];
 Do[
  MPTemp = 0; MP = 0; Equity = 0; TargetPrice = 0; EntryPrice = 0; PrevPeak = 0; PrevTrough = 0; DD = 0; EquityMax = 0; MaxDD = 0;
  Do[
   If[Equity > EquityMax, EquityMax = Equity;];
   DD = Equity - EquityMax;
   If[DD < MaxDD, MaxDD = DD;];
   If[MP == 0, {
     If[HighList[[i]] > HHList[[i]] && OpenList[[i]] < HHList[[i]], {MPTemp = +1; EntryPrice = HHList[[i]];}];
     If[HighList[[i]] > HHList[[i]] && OpenList[[i]] ≥ HHList[[i]], {MPTemp = +1; EntryPrice = OpenList[[i]];}];
     If[LowList[[i]] < LLList[[i]] && OpenList[[i]] > LLList[[i]], {MPTemp = -1; EntryPrice = LLList[[i]];}];
     If[LowList[[i]] < LLList[[i]] && OpenList[[i]] ≤ LLList[[i]], {MPTemp = -1; EntryPrice = OpenList[[i]];}];
    }];
   If[MP == +1, {
     PrevPeak = EntryPrice;
     If[CloseList[[i - 1]] > PrevPeak, PrevPeak = CloseList[[i - 1]];];
     TargetPriceRaw = PrevPeak (1 - StpPct);
     TargetPrice = (Quotient[TargetPriceRaw, 0.25] + Quotient[Mod[TargetPriceRaw, 0.25], 0.125]) * 0.25;
     If[TargetPrice ≥ LowList[[i]] && TargetPrice ≤ OpenList[[i]], {Equity = Equity + PL[TargetPrice, EntryPrice]; MPTemp = 0;}];
     If[TargetPrice ≥ LowList[[i]] && TargetPrice > OpenList[[i]], {Equity = Equity + PL[OpenList[[i]], EntryPrice]; MPTemp = 0;}];
    }];
   If[MP == -1, {
     PrevTrough = EntryPrice;
     If[CloseList[[i - 1]] < PrevTrough, PrevTrough = CloseList[[i - 1]];];
     TargetPriceRaw = PrevTrough (1 + StpPct);
     TargetPrice = (Quotient[TargetPriceRaw, 0.25] + Quotient[Mod[TargetPriceRaw, 0.25], 0.125]) * 0.25;
     If[TargetPrice ≤ HighList[[i]] && TargetPrice ≥ OpenList[[i]], {Equity = Equity + PL[EntryPrice, TargetPrice]; MPTemp = 0;}];
     If[TargetPrice ≤ HighList[[i]] && TargetPrice < OpenList[[i]], {Equity = Equity + PL[EntryPrice, OpenList[[i]]]; MPTemp = 0;}];
    }];
   MP = MPTemp;
   , {i, 1, 48 Month, 1}];
  RoA = Equity / Abs[MaxDD];
  RoAList = Append[RoAList, {ChLength, StpPct, RoA}];
  , {StpPct, 0.001, 0.08, 0.001}]
 , {ChLength, 500, 10 000, 100}]
```

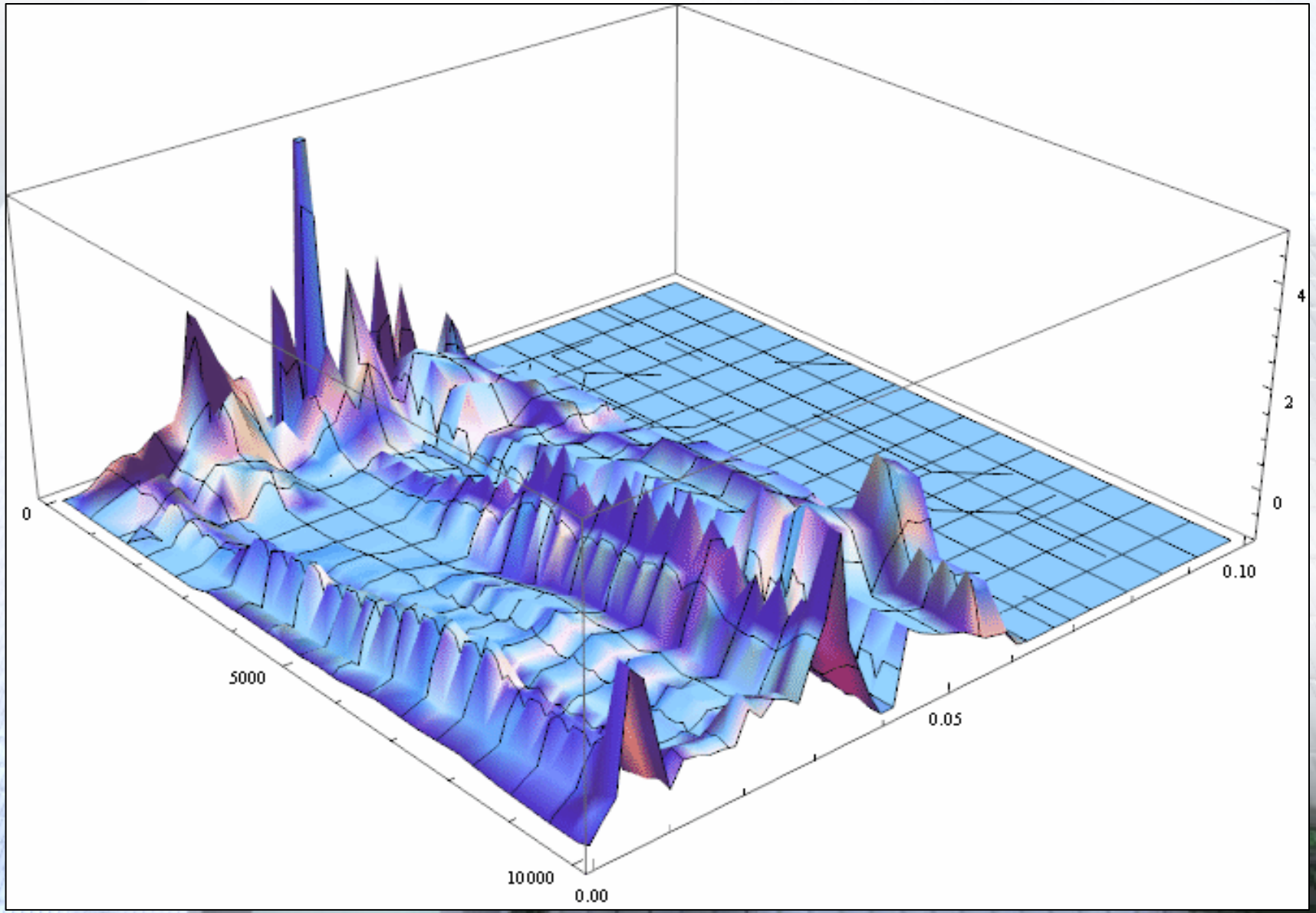
Import the data and select the interval of interest

- Set the maximal and minimal channel length and steps in which we jump

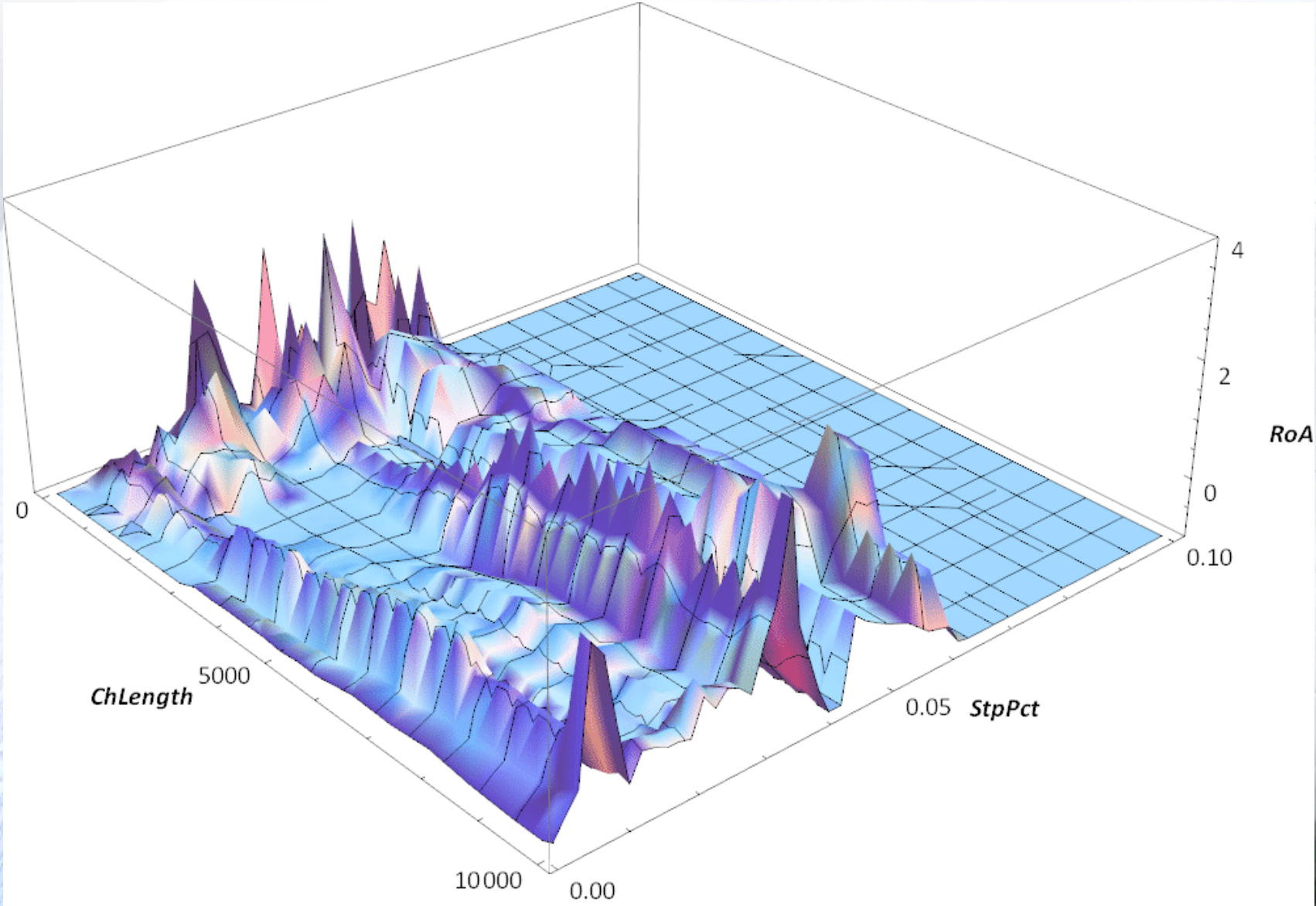
Interactive Demonstration

- aleksnavratil.com/demo.html

Optimization Histogram: 4-year in sample 1984-1988

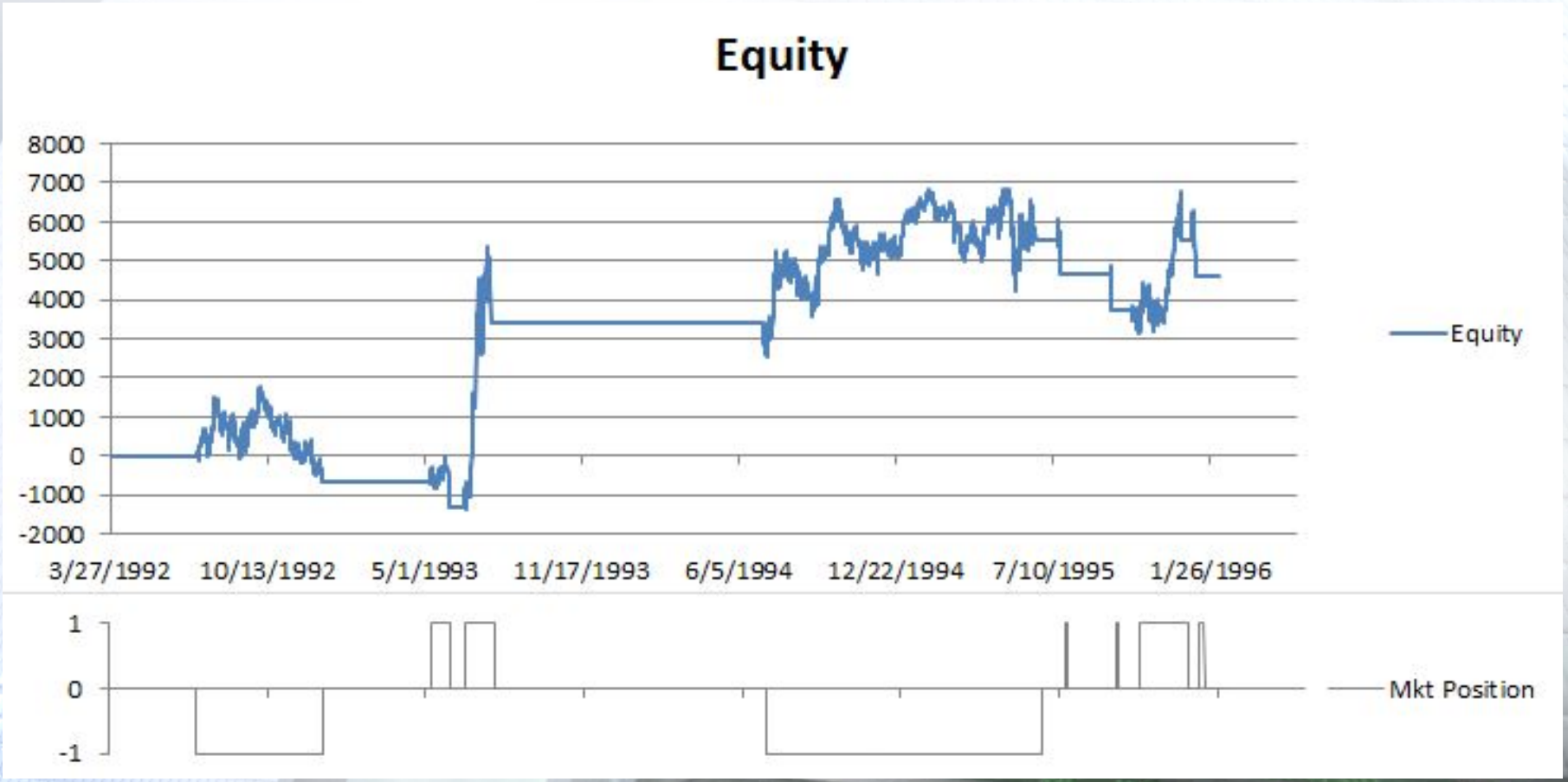


Optimization Histogram: 4-year in sample 1991-1995



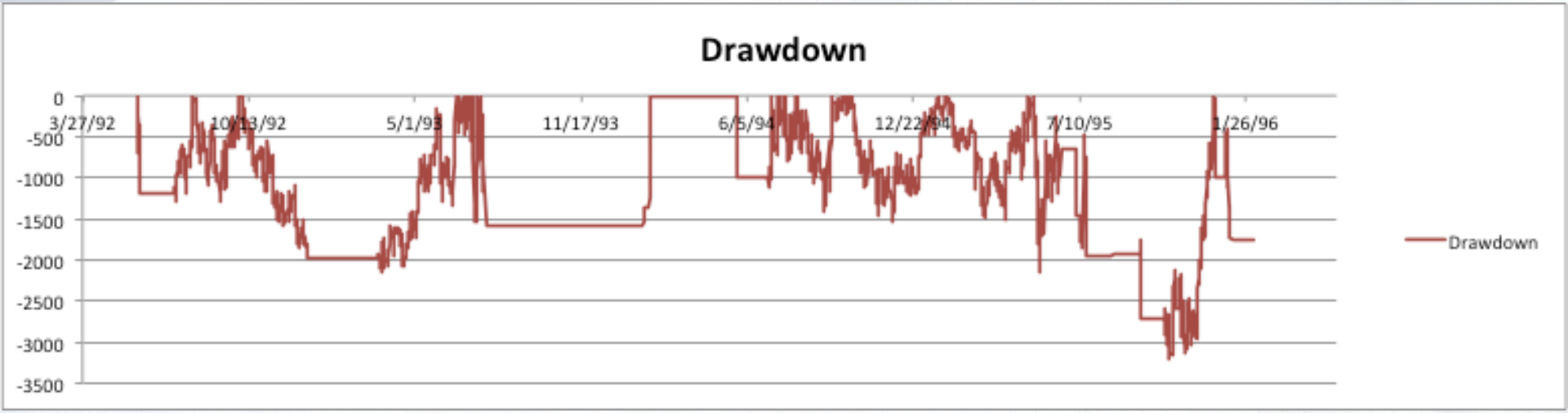
Risk, Returns & Other Results

- Equity Curve (ChLength =9700)



Risk, Returns & Other Results

- Drawdown(ChLength =9700)



In sample/out of sample table:

t/tau	3 months	6 months	9 months	12 months
4 years	1.2	0.55	0.84	-0.84
3 years	0.84	0.6	-1	-0.23
2 years	0.86	0.86	0.9	-0.9
1 year	-0.7	1.06	0.64	-0.41

Risk, Returns & Other Results

Performance Metrics	
Net Profit	4633.86
Max Drawdown	3204.65
Average Drawdown	1131.14
NP/MDD	1.44598006
Long Trade	8.60%
Short Trade	32.80%
Annual Return	82%
Std of Return	11%
Sharpe Ratio	7.45

Conclusions

- ✓ Optimal parameters are generally near:
  - ✓ Stop Percentage = 3.5%
  - ✓ Channel Length = 9700
  - ✓ This produces a RoA near 3.5
  - ✓ This market changes slowly in time
    - ✓ But it’s still necessary to update your strategy regularly

Future work

- Improve speed of code
- Possibly re-implement in a fast procedural language or a LISP for speed improvement
  - C?
  - ANSI Common Lisp?
- Run more RoA calculations
- More backtesting as new market data emerges