



COMPUTATIONAL FINANCE & RISK MANAGEMENT

UNIVERSITY *of* WASHINGTON

Department of Applied Mathematics

Introduction to Trading Systems

Guy Yollin

Applied Mathematics
University of Washington

- 1 The Luxor strategy
 - Introduction to the Luxor strategy
 - The Luxor strategy implemented in quantstrat
- 2 Optimization in quantstrat
- 3 Stoploss orders
 - Overview of Stoploss order implementation
 - Optimizing the Luxor stoploss order

Lecture references

- TradeAnalytics project page on R-forge:
<http://r-forge.r-project.org/projects/blotter/>
 - documents and demos for:
 - blotter package
 - quantstrat package
(specifically the luxor demo scripts)[†]
- Using quantstrat by Jan Humme & Brian Peterson
<http://www.rinfinance.com/agenda/2013/workshop/Humme+Peterson.pdf>
- R-SIG-FINANCE:
<https://stat.ethz.ch/mailman/listinfo/r-sig-finance>

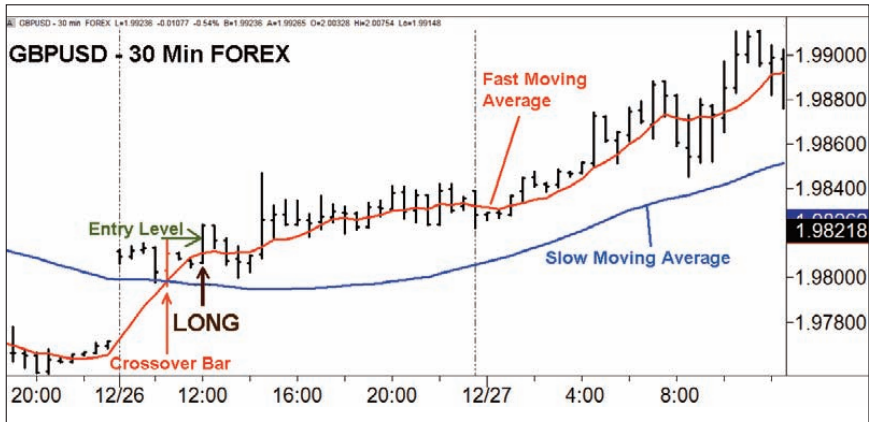
[†]demos are located in the directory: `.../R-3.0.1/library/quantstrat/demo`

- 1 The Luxor strategy
 - Introduction to the Luxor strategy
 - The Luxor strategy implemented in quantstrat
- 2 Optimization in quantstrat
- 3 Stoploss orders

- 1 The Luxor strategy
 - Introduction to the Luxor strategy
 - The Luxor strategy implemented in quantstrat
- 2 Optimization in quantstrat
- 3 Stoploss orders

Luxor strategy

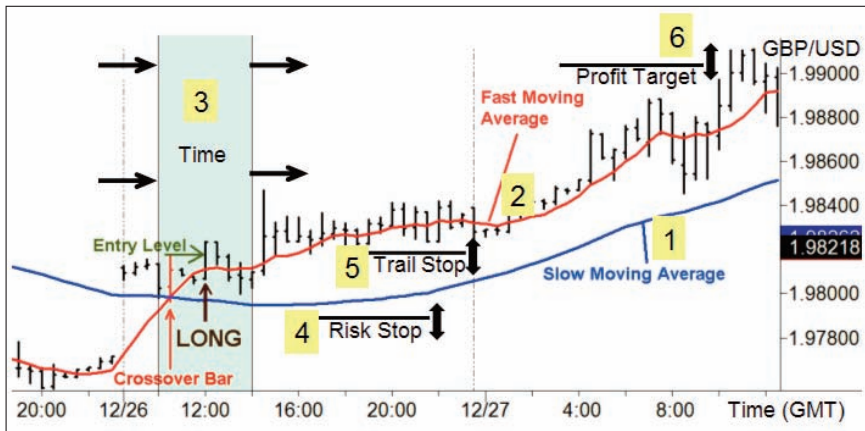
The Luxor strategy is presented in Trading Systems by Emilio Tomasini and Urban Jaekle (Harriman House, 2009)



- Luxor is basically a intra-day dual moving average crossover system

Luxor strategy

Fully implemented, the Luxor system includes 6 separate optimizable design elements; the Luxor demo implements all of these features



Luxor demo in quantstrat

The Luxor demo is the most fully-feature demonstration of quanstrat capabilities. This demo was developed by Jan Humme[†]

Script	Description
luxor.1.strategy.basic.R	simple Luxor strategy
luxor.2.add.paramsets.R	MA optimization pre-processing
luxor.3.paramset.sma.R	optimize dual moving averages
luxor.4.paramset.timespan.R	optimize timespan
luxor.5.strategy.ordersets.R	prep stoploss, trailing stop, profit target
luxor.6.paramset.stoploss.R	optimize stoploss
luxor.6.paramset.stoptrailing.R	optimize trailing stop
luxor.6.paramset.takeprofit.R	optimize profit target
luxor.7.exit+risk.R	apply optimum parameters
luxor.8.walk.forward.R	walk forward analysis

[†]Much of the material in this presentation has been taken directly from Jan's excellent demo and R/Finance 2013 Workshop slides; credit to the original author. Jan maintains a blog at <http://www.opentrades.nl/>

Luxor data included in quantstrat

Item	Setting
Instrument	GBPUSD Forex
Time frame	2002-10-21 to 2002-10-31
Time length	10-days
Time resolution	30-minute OHLC bars

Simplified Luxor System

EasyLanguage code

```
Inputs: FastLength( 3 ), SlowLength( 30 );
```

```
Variables:
```

```
Fast(0), Slow(0), GoLong(False), GoShort(False),  
BuyStop(0), SellStop(0), BuyLimit(0), SellLimit(0);
```

```
Fast = Average(Close, FastLength);
```

```
Slow = Average(Close, SlowLength);
```

```
GoLong = Fast > Slow;
```

```
GoShort = Fast < Slow;
```

```
If Fast crosses above Slow then begin
```

```
BuyStop = High + 1 point;
```

```
BuyLimit = High + 5 points;
```

```
end;
```

```
If Fast crosses below Slow then begin
```

```
SellStop = Low - 1 point;
```

```
SellLimit = Low - 5 points;
```

```
end;
```

```
If GoLong and C < BuyLimit then
```

```
Buy ("Long") next bar at BuyStop Stop;
```

```
If GoShort and C > SellLimit then
```

```
Sell Short ("Short") next bar at SellStop Stop;
```

compare to
Jaekle & Tomasini
Text 3.1

- 1 The Luxor strategy
 - Introduction to the Luxor strategy
 - The Luxor strategy implemented in quantstrat
- 2 Optimization in quantstrat
- 3 Stoploss orders

Initialize instruments and load historic data

R Code:

```
> Sys.setenv(TZ="UTC")
> library(quantstrat)

> initDate = '2002-10-21'
> .from=initDate
> .to='2002-10-31'

> currency(c('GBP', 'USD'))
> exchange_rate('GBPUSD', tick_size=0.0001)

> getSymbols.FI(Symbols='GBPUSD',
               dir=system.file('extdata',package='quantstrat'),
               from=.from, to=.to)
> GBPUSD = to.minutes30(GBPUSD)
> GBPUSD = align.time(GBPUSD, 1800)
```

- `getSymbols.FI` reads symbol data from a file

Initialize instruments and load historic data

R Code:

```
> dim(GBPUSD)
```

```
[1] 411 5
```

```
> last(GBPUSD,5)
```

	GBPUSD.Open	GBPUSD.High	GBPUSD.Low	GBPUSD.Close	GBPUSD.Volume
2002-10-31 21:00:00	1.5648	1.5648	1.5639	1.5639	0
2002-10-31 21:30:00	1.5639	1.5640	1.5636	1.5637	0
2002-10-31 22:00:00	1.5636	1.5636	1.5630	1.5633	0
2002-10-31 22:30:00	1.5633	1.5643	1.5632	1.5643	0
2002-10-31 23:00:00	1.5643	1.5646	1.5639	1.5640	0

```
> myTheme<-chart_theme()
```

```
> myTheme$col$dn.col<-'lightblue'
```

```
> myTheme$col$dn.border <- 'lightgray'
```

```
> myTheme$col$up.border <- 'lightgray'
```

```
> chart_Series(GBPUSD,theme=myTheme)
```

GBPUSD 30-minute bars



System parameters

R Code:

```
> # moving average lengths
> .fast = 10
> .slow = 30

> # optimization range
> .FastSMA = (1:20)
> .SlowSMA = (30:80)

> # trade parameters
> .threshold = 0.0005
> .orderqty = 100000
> .txnfees = -6 # round-trip fee

> # stop loss amount
> .stoploss <- 0.30/100
> .StopLoss = seq(0.05, 0.6, length.out=48)/100

> # trading window
> .timespan = 'T00:00/T23:59'

> # number of optimization samples
> .nsamples=80
```

Define system parameters and initialize quantstrat objects

R Code:

```
> portfolio.st = 'forex'
> account.st = 'IB1'
> strategy.st = 'luxor'

> rm.strat(portfolio.st)
> rm.strat(account.st)

> initPortf(portfolio.st, symbols='GBPUSD', initDate=initDate, currency='USD')
> initAcct(account.st, portfolios=portfolio.st,initDate=initDate,currency='USD')
> initOrders(portfolio.st, initDate=initDate)
> strategy(strategy.st, store=TRUE)
```


Moving average indicators

R Code:

```
> add.indicator(strategy.st, name = "SMA",
  arguments = list(
    x = quote(C1(mktdata)[,1]),
    n = .fast
  ),
  label="nFast"
)

> add.indicator(strategy.st, name="SMA",
  arguments = list(
    x = quote(C1(mktdata)[,1]),
    n = .slow
  ),
  label="nSlow"
)
```

- `C1(mktdata)[,1]` returns the first column named Close
- In `mktdata` the indicator columns will be named `nFast` and `nSlow`
- Even though `n` is assigned in the strategy definition, it can still be overridden via the `parameters` argument of `applyStrategy`

Crossover signals

R Code:

```
> add.signal(strategy.st, name='sigCrossover',
             arguments = list(
               columns=c("nFast","nSlow"),
               relationship="gte"
             ),
             label='long'
)

> add.signal(strategy.st, name='sigCrossover',
             arguments = list(
               columns=c("nFast","nSlow"),
               relationship="lt"
             ),
             label='short'
)
```

- In `mktdata` the signal columns will be named `long` and `short`

Long entry rule

R Code:

```
> add.rule(strategy.st, name='ruleSignal',  
  arguments=list(sigcol='long' , sigval=TRUE,  
    orderside='long' ,  
    ordertype='stoplimit',  
    prefer='High',  
    threshold=.threshold,  
    orderqty=+.orderqty,  
    replace=FALSE  
  ),  
  type='enter',  
  label='EnterLONG'  
)
```

- Order type is stoplimit
- Stop-limit price is High + threshold (based on prefer)
- Not replacing any pending exit orders
- Name given to label appears in orderbook

Short entry rule

R Code:

```
> add.rule(strategy.st, name='ruleSignal',  
  arguments=list(sigcol='short', sigval=TRUE,  
    orderside='short',  
    ordertype='stoplimit',  
    prefer='Low',  
    threshold=-.threshold,  
    orderqty=-.orderqty,  
    replace=FALSE  
  ),  
  type='enter',  
  label='EnterSHORT'  
)
```

- Order type is stoplimit
- Stop-limit price is Low - threshold (based on prefer)
- Not replacing any pending exit orders
- Name given to label appears in orderbook

Long exit rule

R Code:

```
> add.rule(strategy.st, name='ruleSignal',  
  arguments=list(sigcol='long' , sigval=TRUE,  
    orderside='short',  
    ordertype='market',  
    orderqty='all',  
    TxnFees=.txnfees,  
    replace=TRUE  
  ),  
  type='exit',  
  label='Exit2LONG'  
)
```

- Transaction fees added to exit
- Replace any pending open orders

Short exit rule

R Code:

```
> add.rule(strategy.st, name='ruleSignal',  
  arguments=list(sigcol='short', sigval=TRUE,  
    orderside='long' ,  
    ordertype='market',  
    orderqty='all',  
    TxnFees=.txnfees,  
    replace=TRUE  
  ),  
  type='exit',  
  label='Exit2SHORT'  
)
```

- Transaction fees added to exit
- Replace any pending open orders

Apply strategy, update portfolio, and plot results

R Code:

```
> out <- applyStrategy(strategy.st, portfolio.st)
> updatePortf(portfolio.st, Symbols='GBPUSD',
  Dates=paste(':',as.Date(Sys.time()),sep=''))

> chart.Posn(portfolio.st, "GBPUSD",
  TA="add_SMA(n=10,col=2);add_SMA(n=30,col=4)",theme=myTheme)

> PerformanceAnalytics:::textplot(t(tradeStats(portfolio.st, 'GBPUSD'))))

> mk <- mktdata['2002-10-23 15:00::2002-10-24 03:00']
> mk.df <- data.frame(Date=time(mk),coredata(mk))
> PerformanceAnalytics:::textplot(mk.df,show.rownames=F)

> ob <- getOrderBook(portfolio.st)$forex$GBPUSD
> ob.df <- data.frame(Date=time(ob),coredata(ob))
> PerformanceAnalytics:::textplot(ob.df,show.rownames=F)

> PerformanceAnalytics:::textplot(perTradeStats(portfolio.st,"GBPUSD"),
  show.rownames=F)

> chart.ME(portfolio.st,'GBPUSD',type='MAE',scale='percent')
```

GBPUSD 30-minute bars



GBPUSD 30-minute bars

	GBPUSD
Portfolio	forex
Symbol	GBPUSD
Num.Txns	21
Num.Trades	10
Net.Trading.PL	-430
Avg.Trade.PL	-104
Med.Trade.PL	-116
Largest.Winner	364
Largest.Loser	-796
Gross.Profits	886
Gross.Losses	-1926
Std.Dev.Trade.PL	349.7555
Percent.Positive	40
Percent.Negative	60
Profit.Factor	0.4600208
Avg.Win.Trade	221.5
Med.Win.Trade	244
Avg.Losing.Trade	-321
Med.Losing.Trade	-251
Avg.Daily.PL	-148.5714
Med.Daily.PL	-126
Std.Dev.Daily.PL	476.379
Max.Drawdown	-1692
Profit.To.Max.Draw	-0.2541371
Avg.WinLoss.Ratio	0.6900312
Med.WinLoss.Ratio	0.9721116
Max.Equity	400
Min.Equity	-1292
End.Equity	-430

The mktdata object

	GBPUSD.Open	GBPUSD.High	GBPUSD.Low	GBPUSD.Close	GBPUSD.Volume	SMA.nFast	SMA.nSlow	long	short
2002-10-23 15:00:00	1.5484	1.5485	1.5480	1.5485	0	1.54824	1.547437		
2002-10-23 15:30:00	1.5485	1.5485	1.5475	1.5477	0	1.54814	1.547520		
2002-10-23 16:00:00	1.5477	1.5477	1.5472	1.5472	0	1.54796	1.547617		
2002-10-23 16:30:00	1.5472	1.5474	1.5470	1.5473	0	1.54777	1.547720		
2002-10-23 17:00:00	1.5473	1.5474	1.5468	1.5468	0	1.54762	1.547727		1
2002-10-23 17:30:00	1.5468	1.5470	1.5466	1.5468	0	1.54754	1.547670		
2002-10-23 18:00:00	1.5468	1.5468	1.5465	1.5465	0	1.54747	1.547600		
2002-10-23 18:30:00	1.5465	1.5472	1.5464	1.5471	0	1.54743	1.547557		
2002-10-23 19:00:00	1.5471	1.5475	1.5471	1.5475	0	1.54739	1.547533		
2002-10-23 19:30:00	1.5475	1.5476	1.5472	1.5475	0	1.54729	1.547540		
2002-10-23 20:00:00	1.5475	1.5484	1.5475	1.5484	0	1.54728	1.547603		
2002-10-23 20:30:00	1.5485	1.5485	1.5482	1.5484	0	1.54735	1.547630		
2002-10-23 21:00:00	1.5484	1.5491	1.5484	1.5490	0	1.54753	1.547690		
2002-10-23 21:30:00	1.5490	1.5491	1.5486	1.5486	0	1.54766	1.547713		
2002-10-23 22:00:00	1.5486	1.5486	1.5482	1.5482	0	1.54780	1.547793	1	
2002-10-23 22:30:00	1.5482	1.5483	1.5482	1.5483	0	1.54795	1.547863		
2002-10-23 23:00:00	1.5483	1.5486	1.5476	1.5484	0	1.54814	1.547960		
2002-10-23 23:30:00	1.5484	1.5484	1.5471	1.5471	0	1.54814	1.547963		
2002-10-24 00:00:00	1.5471	1.5478	1.5469	1.5472	0	1.54811	1.547920		
2002-10-24 00:30:00	1.5473	1.5480	1.5470	1.5470	0	1.54806	1.547883		
2002-10-24 01:00:00	1.5470	1.5471	1.5465	1.5470	0	1.54792	1.547813		
2002-10-24 01:30:00	1.5470	1.5471	1.5464	1.5464	0	1.54772	1.547737		1
2002-10-24 02:00:00	1.5464	1.5475	1.5461	1.5467	0	1.54749	1.547660		
2002-10-24 02:30:00	1.5467	1.5471	1.5460	1.5461	0	1.54724	1.547557		
2002-10-24 03:00:00	1.5461	1.5465	1.5449	1.5449	0	1.54691	1.547443		

Order book

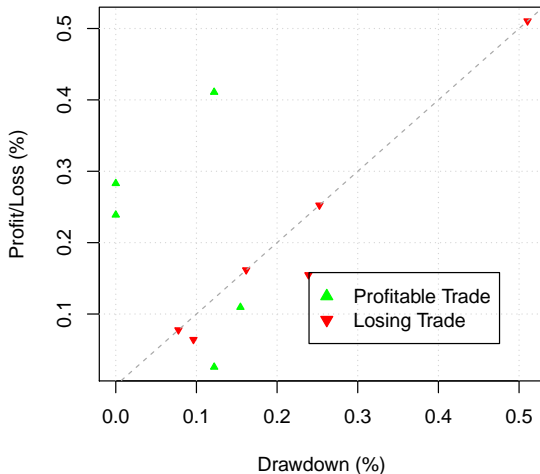
	Order.Qty	Order.Price	Order.Type	Order.Side	Order.Threshold	Order.Status	Order.StatusTime	Prefer	Order.Set	Txn.Fees	Rule
2002-10-21 23:30:00	1e+05	1.5447	stoplimit	long	5e-04	closed	2002-10-22 02:00:00	High		0	EnterLONG
2002-10-22 17:00:00	all	1.5453	market	long		closed	2002-10-22 17:30:00			-6	Exit2SHORT
2002-10-22 17:00:00	-1e+05	1.5447	stoplimit	short	-5e-04	closed	2002-10-22 17:30:00	Low		0	EnterSHORT
2002-10-23 02:30:00	all	1.5485	market	short		closed	2002-10-23 03:00:00			-6	Exit2LONG
2002-10-23 02:30:00	1e+05	1.5492	stoplimit	long	5e-04	closed	2002-10-23 03:00:00	High		0	EnterLONG
2002-10-23 17:00:00	all	1.5468	market	long		closed	2002-10-23 17:30:00			-6	Exit2SHORT
2002-10-23 17:00:00	-1e+05	1.5463	stoplimit	short	-5e-04	replaced	2002-10-23 22:00:00	Low		0	EnterSHORT
2002-10-23 22:00:00	1e+05	1.5491	stoplimit	long	5e-04	replaced	2002-10-24 01:30:00	High		0	EnterLONG
2002-10-24 01:30:00	-1e+05	1.5459	stoplimit	short	-5e-04	closed	2002-10-24 03:00:00	Low		0	EnterSHORT
2002-10-24 11:00:00	all	1.5476	market	short		closed	2002-10-24 11:30:00			-6	Exit2LONG
2002-10-24 11:00:00	1e+05	1.5493	stoplimit	long	5e-04	closed	2002-10-24 12:00:00	High		0	EnterLONG
2002-10-25 03:30:00	all	1.5535	market	long		closed	2002-10-25 04:00:00			-6	Exit2SHORT
2002-10-25 03:30:00	-1e+05	1.553	stoplimit	short	-5e-04	closed	2002-10-25 04:00:00	Low		0	EnterSHORT
2002-10-25 11:30:00	all	1.5528	market	short		closed	2002-10-25 12:00:00			-6	Exit2LONG
2002-10-25 11:30:00	1e+05	1.5533	stoplimit	long	5e-04	replaced	2002-10-25 12:00:00	High		0	EnterLONG
2002-10-25 12:00:00	-1e+05	1.5508	stoplimit	short	-5e-04	closed	2002-10-27 23:30:00	Low		0	EnterSHORT
2002-10-28 10:00:00	all	1.5538	market	short		closed	2002-10-28 10:30:00			-6	Exit2LONG
2002-10-28 10:00:00	1e+05	1.555	stoplimit	long	5e-04	closed	2002-10-28 10:30:00	High		0	EnterLONG
2002-10-28 23:30:00	all	1.5582	market	long		closed	2002-10-29 00:00:00			-6	Exit2SHORT
2002-10-28 23:30:00	-1e+05	1.5572	stoplimit	short	-5e-04	replaced	2002-10-29 06:30:00	Low		0	EnterSHORT
2002-10-29 06:30:00	1e+05	1.5594	stoplimit	long	5e-04	replaced	2002-10-29 07:00:00	High		0	EnterLONG
2002-10-29 07:00:00	-1e+05	1.5572	stoplimit	short	-5e-04	closed	2002-10-29 07:30:00	Low		0	EnterSHORT
2002-10-30 05:00:00	all	1.5563	market	short		closed	2002-10-30 05:30:00			-6	Exit2LONG
2002-10-30 05:00:00	1e+05	1.5578	stoplimit	long	5e-04	closed	2002-10-30 09:30:00	High		0	EnterLONG
2002-10-30 11:00:00	all	1.5569	market	long		closed	2002-10-30 11:30:00			-6	Exit2SHORT
2002-10-30 11:00:00	-1e+05	1.5558	stoplimit	short	-5e-04	replaced	2002-10-30 12:00:00	Low		0	EnterSHORT
2002-10-30 12:00:00	1e+05	1.5579	stoplimit	long	5e-04	closed	2002-10-30 13:00:00	High		0	EnterLONG

Per-trade stats

	End	Init.Pos	Max.Pos	Num.Txns	Max.Notional.Cost	Net.Trading.PL	MAE	MFE	Pct.Net.Trading.PL	Pct.MAE	Pct.MFE	tick.Net.Trading.PL	tick.MAE	tick.MFE
2002-10-22 02:00:00	2002-10-22 17:30:00	1e+05	1e+05	2	154470	-120	-120	400	-0.0007768499	-0.0007768499	0.0025894996	-12	-12	40
2002-10-22 17:30:00	2002-10-23 03:00:00	-1e+05	-1e+05	2	-154470	-390	-390	180	-0.0025247621	-0.0025247621	0.0011652748	-39	-39	18
2002-10-23 03:00:00	2002-10-23 17:30:00	1e+05	1e+05	2	154920	-240	-370	0	-0.0015491867	-0.0023883295	0.0000000000	-24	-37	0
2002-10-24 03:00:00	2002-10-24 11:30:00	-1e+05	-1e+05	2	-154590	-250	-250	100	-0.0016171809	-0.0016171809	0.0006468724	-25	-25	10
2002-10-24 12:00:00	2002-10-25 04:00:00	1e+05	1e+05	2	154930	370	0	610	0.0023881753	0.0000000000	0.0039372620	37	0	61
2002-10-25 04:00:00	2002-10-25 12:00:00	-1e+05	-1e+05	2	-155300	170	-240	170	0.0010946555	-0.0015453960	0.0010946555	17	-24	17
2002-10-27 23:30:00	2002-10-28 10:30:00	-1e+05	-1e+05	2	-154750	-790	-790	190	-0.0051050081	-0.0051050081	0.0012277868	-79	-79	19
2002-10-28 10:30:00	2002-10-29 00:00:00	1e+05	1e+05	3	155500	440	0	550	0.0028295820	0.0000000000	1.0000000000	44	0	Inf
2002-10-29 07:30:00	2002-10-30 05:30:00	-1e+05	-1e+05	2	-155720	40	-190	210	0.0002568713	-0.0012201387	0.0013485744	4	-19	21
2002-10-30 09:30:00	2002-10-30 11:30:00	1e+05	1e+05	2	155780	-100	-150	0	-0.0006419309	-0.0009628964	0.0000000000	-10	-15	0
2002-10-30 13:00:00	2002-10-31 23:00:00	1e+05	1e+05	1	155790	640	-190	780	0.0041080942	-0.0012195905	0.0050067398	64	-19	78

Maximum adverse excursion

GBPUSD Maximum Adverse Excursion (MAE)



Outline

- 1 The Luxor strategy
- 2 Optimization in quantstrat
- 3 Stoploss orders

Optimization in quantstrat

Optimization in quantstrat is implemented using a concept call a paramset; along with paramsets, there are distributions and constraints.

- paramset** A paramset is a collection of variables that will be optimized subject to their range of allowed values (distribution) and any constraints between them
- distribution** A distribution in a paramset is simply the range of values that a variable is allowed to take (e.g. $\text{fastMA} = 1:20$)
- constraint** A constraint is a relationship that must be true between two distributions in a paramset (e.g. $\text{fastMA} < \text{slowMA}$)

Optimization functions in quantstrat

The following functions implement parameter optimization in quantstrat:

- `add.distribution` Creates a distribution in paramset, where a distribution consists of the name of a variable in a strategy component plus a range of values for this variable.
- `add.constraint` Creates a constraint on 2 distributions in a paramset, i.e. a restriction limiting the allowed combinations from the ranges for distribution 1 and distribution 2.
- `apply.paramset` Runs `applyStrategy` once for each parameter combination as specified by the parameter distributions and constraints in the paramset. `apply.paramset` will do parallel processing on multiple cores if available.

Define parameter range

R Code:

```
> args(add.distribution)

function (strategy, paramset.label, component.type, component.label,
        variable, weight = NULL, label, store = TRUE)
NULL

> add.distribution(strategy.st,
  paramset.label = 'SMA',
  component.type = 'indicator',
  component.label = 'nFast',
  variable = list(n = .FastSMA),
  label = 'nFAST'
)
```

- The input parameter nFast may range from 1 to 30
- Optimized parameter name is nFAST

Define parameter range

R Code:

```
> add.distribution(strategy.st,  
  paramset.label = 'SMA',  
  component.type = 'indicator',  
  component.label = 'nSlow',  
  variable = list(n = .SlowSMA),  
  label = 'nSLOW'  
)
```

- The input parameter nSlow may range from 20 to 80
- Optimized parameter name is nSLOW

Define parameter constraint

R Code:

```
> args(add.constraint)

function (strategy, paramset.label, distribution.label.1, distribution.label.2,
        operator, label, store = TRUE)
NULL

> add.constraint(strategy.st,
  paramset.label = 'SMA',
  distribution.label.1 = 'nFAST',
  distribution.label.2 = 'nSLOW',
  operator = '<',
  label = 'SMA'
)
```

- nFAST must be less than nSLOW

Re-Initialize portfolio and account

R Code:

```
> rm.strat(portfolio.st)
> rm.strat(account.st)

> initPortf(portfolio.st, symbols='GBPUSD', initDate=initDate, currency='USD')
> initAcct(account.st, portfolios=portfolio.st,
  initDate=initDate, currency='USD')
> initOrders(portfolio.st, initDate=initDate)
```

Parallel computing with foreach

- The foreach package facilitates easily-accessible parallel processing in R
- The foreach function is a for-like looping construct where each iteration of the for loop can be run in parallel if a multicore processor (now very common) is available
- Each loop iteration returns a result and these results can be combined in a variety of ways depending on their data type
- foreach requires that you register a *parallel backend*
 - On Windows platforms, doParallel is the recommend parallel backend
 - On non-Windows platforms, doMC is the recommend parallel backend

Setup parallel backend and test foreach

R Code:

```
> library(parallel)
> detectCores()

[1] 8

> if( Sys.info()['sysname'] == "Windows" )
{
  library(doParallel)
  registerDoParallel(cores=detectCores())
} else {
  library(doMC)
  registerDoMC(cores=detectCores())
}

> foreach(i=1:8, .combine=c) %dopar% sqrt(i)

[1] 1.000000 1.414214 1.732051 2.000000 2.236068 2.449490 2.645751 2.828427
```

- All sqrt operations are run in parallel via separate processes on a multi-core processor

The `apply.paramset` function

The function `apply.paramset` function will run `applyStrategy()` on `portfolio.st`, once for each parameter combination as specified by the parameter distributions and constraints in the paramset

R Code: The `apply.paramset` function

```
> args(apply.paramset)

function (strategy.st, paramset.label, portfolio.st, account.st,
         mktdata = NULL, nsamples = 0, user.func = NULL, user.args = NULL,
         calc = "slave", audit = NULL, packages = NULL, verbose = FALSE)
NULL
```

Main arguments:

<code>strategy.st</code>	text name of the strategy
<code>paramset.label</code>	text name of the paramset
<code>portfolio.st</code>	text name of the portfolio
<code>nsamples</code>	if <code>nsamples > 0</code> then take a sample of size <code>nsamples</code> from the paramset

Optimize parameters

R Code:

```
> results <- apply.paramset(strategy.st, paramset.label='SMA',  
  portfolio.st=portfolio.st, account.st=account.st, nsamples=.nsamples)  
  
> head(names(results),20)  
  
[1] "forex.1"      "tradeStats" "forex.2"      "forex.3"      "forex.4"      "forex.5"  
[7] "forex.6"      "forex.7"      "forex.8"      "forex.9"      "forex.10"     "forex.11"  
[13] "forex.12"     "forex.13"     "forex.14"     "forex.15"     "forex.16"     "forex.17"  
[19] "forex.18"     "forex.19"  
  
> tS <- results$tradeStats  
> idx <- order(tS[,1],tS[,2])  
> tS <- tS[idx,]  
> PerformanceAnalytics:::textplot(t(tS)[,1:10])
```

As of 2013-08-12, `apply.paramset` does not appear to run properly in parallel on Windows. To run on a Windows platform, load the `doParallel` package but do not call the `registerDoParallel` function; `apply.paramset` will then be able to run in sequential rather than parallel mode.

paramset trade stats

	1	31	61	91	121	151	181	211	241	271
nFAST	1	1	1	1	1	1	1	1	1	1
nSLOW	20	21	22	23	24	25	26	27	28	29
Portfolio	forex.1	forex.31	forex.61	forex.91	forex.121	forex.151	forex.181	forex.211	forex.241	forex.271
Symbol	GBPUSD	GBPUSD	GBPUSD	GBPUSD	GBPUSD	GBPUSD	GBPUSD	GBPUSD	GBPUSD	GBPUSD
Num.Txns	55	59	57	53	53	53	47	47	46	40
Num.Trades	27	29	28	26	26	26	23	23	23	20
Net.Trading.PL	-372.00000	-1214.00000	-808.00000	-716.00000	-796.00000	-1136.00000	-918.00000	-688.00000	-418.00000	-280.00000
Avg.Trade.PL	-11.5555556	-39.7931034	-26.7142857	-25.2307692	-28.3076923	-41.3846154	-37.3043478	-27.3043478	-18.1739130	-14.0000000
Med.Trade.PL	-86.0000000	-116.000000	-106.000000	-96.000000	-96.000000	-96.000000	-96.000000	-96.000000	-96.000000	-96.000000
Largest.Winner	944.00000	944.00000	944.00000	954.00000	944.00000	674.00000	674.00000	674.00000	694.00000	674.00000
Largest.Loser	-336	-506	-506	-506	-506	-506	-506	-506	-506	-406
Gross.Profits	2282.00000	2208.00000	2208.00000	2134.00000	2054.00000	1714.00000	1754.00000	1984.00000	2008.00000	2040.00000
Gross.Losses	-2594.000	-3362.000	-2956.000	-2790.000	-2790.000	-2790.000	-2612.000	-2612.000	-2426.000	-2320.000
Std.Dev.Trade.PL	261.0826	268.2538	266.7212	280.6980	273.9096	234.9678	249.7509	259.2525	257.8575	284.9487
Percent.Positive	29.629630	24.137931	25.000000	23.076923	23.076923	23.076923	26.086957	26.086957	30.434783	25.000000
Percent.Negative	70.37037	75.86207	75.000000	76.92308	76.92308	76.92308	73.91304	73.91304	69.56522	75.000000
Profit.Factor	0.879722436	0.656751933	0.748955345	0.764874552	0.736200717	0.614336918	0.671516080	0.759571210	0.827699918	0.879310345
Avg.Win.Trade	285.250000	315.42857	315.42857	355.66667	342.33333	285.66667	292.33333	330.66667	286.85714	408.000000
Med.Win.Trade	164.00000	194.00000	194.00000	199.00000	199.00000	164.00000	184.00000	264.00000	214.00000	524.00000
Avg.Losing.Trade	-136.5263	-152.8182	-140.7619	-139.5000	-139.5000	-153.6471	-153.6471	-151.6250	-154.6667	-154.6667
Med.Losing.Trade	-136.00000	-136.00000	-136.00000	-131.00000	-131.00000	-131.00000	-146.00000	-146.00000	-121.00000	-176.00000
Avg.Daily.PL	-34.6666667	-128.2222222	-83.1111111	-72.8888889	-81.7777778	-119.5555556	-107.2500000	-78.5000000	-52.2500000	-35.0000000
Med.Daily.PL	-8.600000e+01	-8.600000e+01	-8.400000e+01	-1.600000e+01	-1.600000e+01	-1.600000e+01	-1.120000e+02	-1.120000e+02	-3.700000e+01	-9.200000e+01
Std.Dev.Daily.PL	524.8333	677.7414	606.3779	615.1505	621.8444	536.8461	601.9530	625.8176	601.0956	540.6832
Max.Drawdown	-1118.000	-1960.000	-1624.000	-1558.000	-1604.000	-1604.000	-1870.000	-1870.000	-1684.000	-1568.000
Profit.To.Max.Draw	-0.332737030	-0.619387755	-0.497536946	-0.459563543	-0.496259352	-0.708229426	-0.490909091	-0.367914439	-0.248218527	-0.178571429
Avg.WinLoss.Ratio	2.089340786	2.064077505	2.240866035	2.549581840	2.454002389	2.047789725	1.902628892	2.152118428	1.891885526	2.637931034
Med.WinLoss.Ratio	1.205882353	1.426470588	1.426470588	1.519083969	1.519083969	1.251908397	1.260273973	1.808219178	1.768595041	2.977272727
Max.Equity	714.00000	198.00000	278.00000	440.00000	360.00000	308.00000	324.00000	324.00000	324.00000	314.00000
Min.Equity	-920.00000	-1762.0000	-1356.0000	-1250.0000	-1296.0000	-1296.0000	-1546.0000	-1546.0000	-1360.0000	-1254.0000
End.Equity	-372.00000	-1214.00000	-808.00000	-716.00000	-796.00000	-1136.00000	-918.00000	-688.00000	-418.00000	-280.00000

Heatmaps of strategy performance

R Code: Plot code

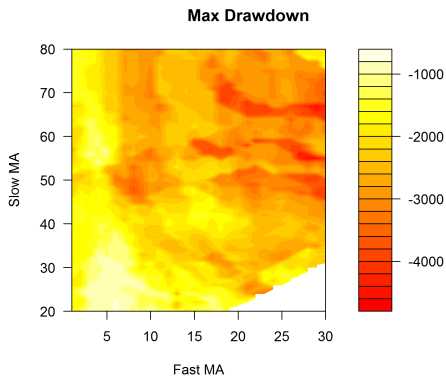
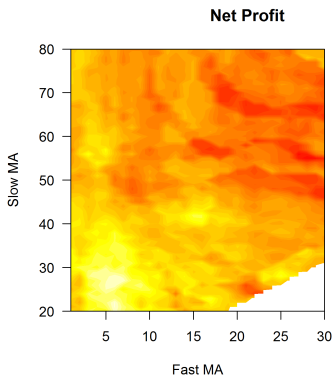
```
> # net profit
> z <- tapply(X=tS[, "End.Equity"], INDEX=list(Fast=tS[,1], Slow=tS[,2]), FUN=sum)
> z[1:5, 1:10]
```

	Slow									
Fast	20	21	22	23	24	25	26	27	28	29
1	-372	-1214	-808	-716	-796	-1136	-918	-688	-418	-280
2	-832	-892	-1298	-1128	-446	-676	-1200	-1170	-970	-398
3	272	42	2	272	458	-82	320	210	586	536
4	86	202	222	598	178	464	560	446	762	662
5	58	364	550	396	606	816	822	792	722	632

```
> x <- as.numeric(rownames(z))
> y <- as.numeric(colnames(z))
> filled.contour(x=x,y=y,z=z,color = heat.colors,xlab="Fast MA",ylab="Slow MA")
> title("Net Profit")

> # maxdd
> z <- tapply(X=tS[, "Max.Drawdown"], INDEX=list(Fast=tS[,1], Slow=tS[,2]), FUN=sum)
> x <- as.numeric(rownames(z))
> y <- as.numeric(colnames(z))
> filled.contour(x=x,y=y,z=z,color = heat.colors,xlab="Fast MA",ylab="Slow MA")
> title("Max Drawdown")
```

Heatmaps of strategy performance



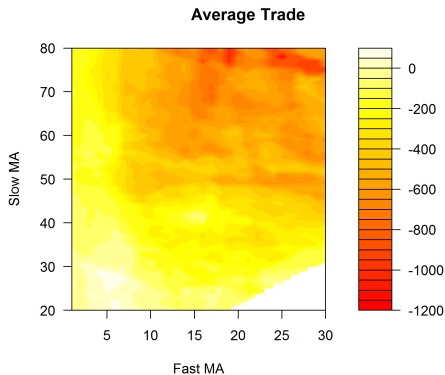
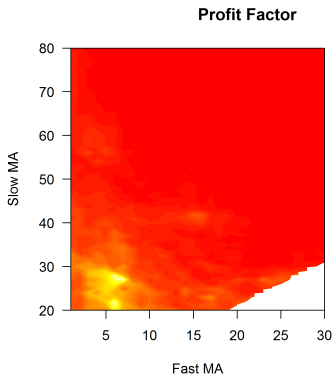
Heatmaps of strategy performance

R Code: Plot code

```
> # profit factor
> z <- tapply(X=tS[, "Profit.Factor"], INDEX=list(Fast=tS[,1], Slow=tS[,2]), FUN=sum)
> x <- as.numeric(rownames(z))
> y <- as.numeric(colnames(z))
> filled.contour(x=x, y=y, z=z, color = heat.colors, xlab="Fast MA", ylab="Slow MA")
> title("Profit Factor")

> # avg trade P&L
> z <- tapply(X=tS[, "Avg.Trade.PL"], INDEX=list(Fast=tS[,1], Slow=tS[,2]), FUN=sum)
> x <- as.numeric(rownames(z))
> y <- as.numeric(colnames(z))
> filled.contour(x=x, y=y, z=z, color = heat.colors, xlab="Fast MA", ylab="Slow MA")
> title("Average Trade")
```

Heatmaps of strategy performance



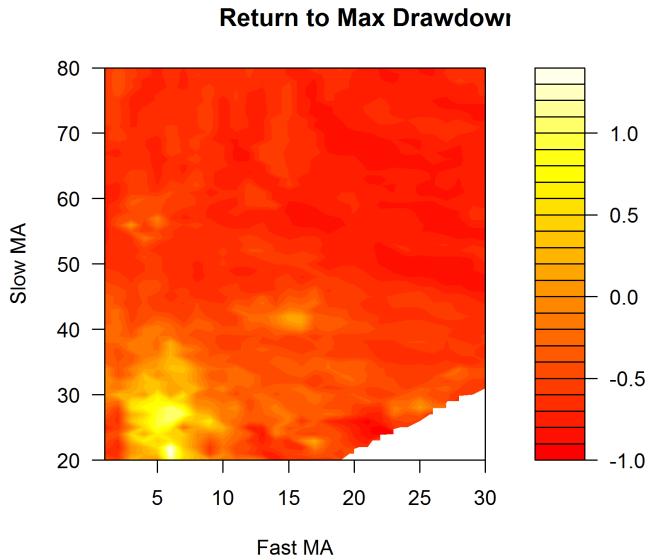
Plot return to maximum drawdown

R Code: Plot code

```
> # return to maxdd
> z <- tapply(X=tS[, "Profit.To.Max.Draw"],
  INDEX=list(Fast=tS[,1], Slow=tS[,2]), FUN=sum)
> x <- as.numeric(rownames(z))
> y <- as.numeric(colnames(z))
> filled.contour(x=x, y=y, z=z, color = heat.colors, xlab="Fast MA", ylab="Slow MA")
> title("Return to Max Drawdown")

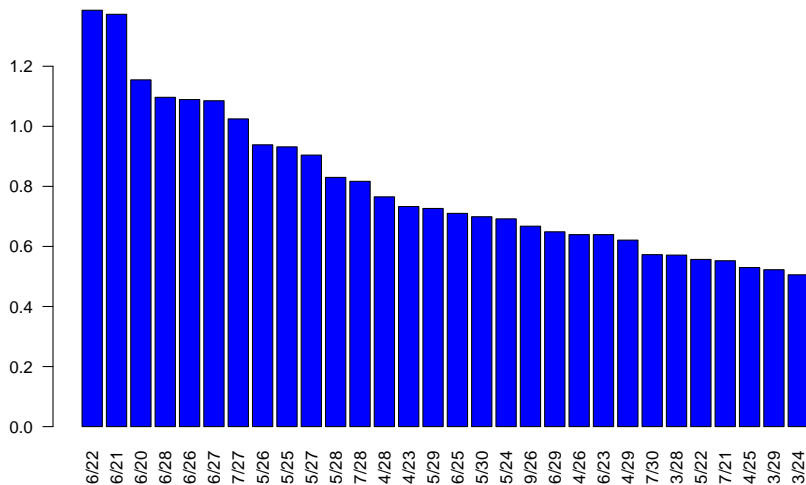
> rmdd <- tS$Profit.To.Max.Draw
> idx <- order(rmdd, decreasing=T)[1:30]
> labs <- paste(tS$nFAST[idx], tS$nSLOW[idx], sep="/")
> barplot(rmdd[idx], names.arg=labs, col=4, las=2, main="Return to MaxDrawdown")
```

Return to maximum drawdown



Best return-to-max-drawdown combinations

Return to MaxDrawdown



The tradeGraphs function

The tradeGraphs function creates a 3-D rendering of the parameter surface

R Code:

```
> args(tradeGraphs)

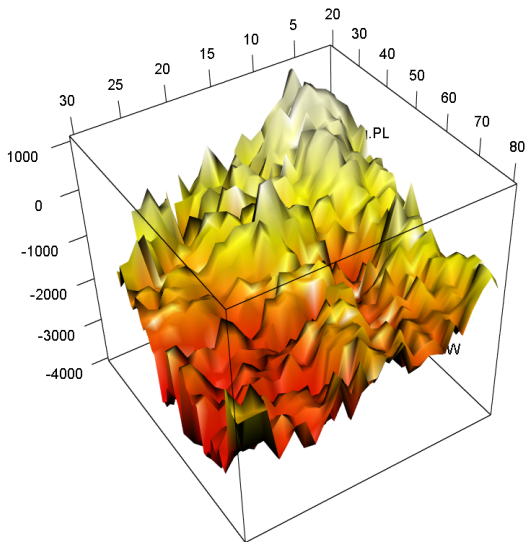
function (stats, free.params, params.filter = NULL, statistics,
          title = NULL)
NULL

> tradeGraphs (stats = tS, free.params = c("nFAST", "nSLOW"),
               statistics = c("Profit.To.Max.Draw", "Net.Trading.PL", "Max.Drawdown",
                              "Avg.Trade.PL", "Num.Trades", "Profit.Factor"), title = '')
```

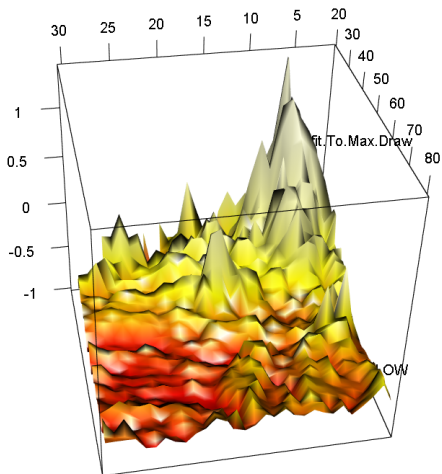
Main arguments:

stats	data frame generated by tradeStats()
free.params	vector (length 2) containing column names for x and y
statistics	vector containing the column names for z

Net profit



Return to maximum drawdown



- 1 The Luxor strategy
- 2 Optimization in quantstrat
- 3 Stoploss orders
 - Overview of Stoploss order implementation
 - Optimizing the Luxor stoploss order

Outline

- 1 The Luxor strategy
- 2 Optimization in quantstrat
- 3 Stoploss orders
 - Overview of Stoploss order implementation
 - Optimizing the Luxor stoploss order

Ordersets and order chains

To implement stop-loss or trailing-stop orders, quantstrat utilizes the concept of ordersets and order chains:

- orderset** An orderset is a collection of OCO orders
- OCO order** One-Cancels-Other (OCO) orders are grouped orders such that when one is filled, all others in the orderset are cancelled
- order chain** An order chain defines an order (child) which will be created when another order (parent) is filled

Moving average indicators

R Code:

```
> rm.strat(strategy.st)
> strategy(strategy.st, store=TRUE)

> add.indicator(strategy.st, name = "SMA",
  arguments = list(
    x = quote(C1(mktdata)[,1]),
    n = .fast
  ),
  label="nFast"
)

> add.indicator(strategy.st, name="SMA",
  arguments = list(
    x = quote(C1(mktdata)[,1]),
    n = .slow
  ),
  label="nSlow"
)
```

Crossover signals

R Code:

```
> add.signal(strategy.st, name='sigCrossover',  
  arguments = list(  
    columns=c("nFast", "nSlow"),  
    relationship="gte"  
  ),  
  label='long'  
)  
  
> add.signal(strategy.st, name='sigCrossover',  
  arguments = list(  
    columns=c("nFast", "nSlow"),  
    relationship="lt"  
  ),  
  label='short'  
)
```


Long entry rule

R Code:

```
> add.rule(strategy.st, name = 'ruleSignal',
  arguments=list(sigcol='long' , sigval=TRUE,
    replace=FALSE,
    orderside='long' ,
    ordertype='stoplimit',
    prefer='High',
    threshold=.threshold,
    TxnFees=0,
    orderqty=+.orderqty,
    osFUN=osMaxPos,
    orderset='ocolong'
  ),
  type='enter',
  timespan = .timespan,
  label='EnterLONG'
)
```

- Assigned to orderset ocolong
- osMaxPos order sizing
- timespan specified

Short entry rule

R Code:

```
> add.rule(strategy.st, name = 'ruleSignal',  
  arguments=list(sigcol='short', sigval=TRUE,  
    replace=FALSE,  
    orderside='short',  
    ordertype='stoplimit',  
    prefer='Low',  
    threshold=.threshold,  
    TxnFees=0,  
    orderqty=-.orderqty,  
    osFUN=osMaxPos,  
    orderset='ocoshort'  
  ),  
  type='enter',  
  timespan = .timespan,  
  label='EnterSHORT'  
)
```

- Assigned to orderset ocoshort
- osMaxPos order sizing
- timespan specified

Long exit rule

R Code:

```
> add.rule(strategy.st, name = 'ruleSignal',  
  arguments=list(sigcol='short', sigval=TRUE,  
    replace=TRUE,  
    orderside='long' ,  
    ordertype='market',  
    TxnFees=.txnfees,  
    orderqty='all',  
    orderset='ocolong'  
  ),  
  type='exit',  
  timespan = .timespan,  
  label='Exit2SHORT'  
)
```

- Assigned to orderset ocolong
- timespan specified

Short exit rule

R Code:

```
> add.rule(strategy.st, name = 'ruleSignal',  
  arguments=list(sigcol='long' , sigval=TRUE,  
    replace=TRUE,  
    orderside='short',  
    ordertype='market',  
    TxnFees=.txnfees,  
    orderqty='all',  
    orderset='ocoshort'  
  ),  
  type='exit',  
  timespan = .timespan,  
  label='Exit2LONG'  
)
```

- Assigned to orderset ocoshort
- timespan specified

Long position stop loss

R Code:

```
> add.rule(strategy.st, name = 'ruleSignal',
  arguments=list(sigcol='long' , sigval=TRUE,
    replace=FALSE,
    orderside='long',
    ordertype='stoplimit',
    tmult=TRUE,
    threshold=quote(.stoploss),
    TxnFees=.txnfees,
    orderqty='all',
    orderset='ocolong'
  ),
  type='chain', parent='EnterLONG',
  label='StopLossLONG',
  enabled=FALSE
)
```

- Assigned to orderset ocolong
- order change: EnterLONG -> StopLossLONG
 - When EnterLONG order is filled, a StopLossLONG order is created

Short position stop loss

R Code:

```
> add.rule(strategy.st, name = 'ruleSignal',
  arguments=list(sigcol='short' , sigval=TRUE,
    replace=FALSE,
    orderside='short',
    ordertype='stoplimit',
    tmult=TRUE,
    threshold=quote(.stoploss),
    TxnFees=.txnfees,
    orderqty='all',
    orderset='ocoshort'
  ),
  type='chain', parent='EnterSHORT',
  label='StopLossSHORT',
  enabled=FALSE
)
```

- Assigned to orderset ocoshort
- order change: EnterSHORT -> StopLossSHORT
 - When EnterSHORT order is filled, a StopLossSHORT order is created

Re-Initialize portfolio and account

R Code:

```
> rm.strat(portfolio.st)
> rm.strat(account.st)

> initPortf(portfolio.st, symbols='GBPUSD', initDate=initDate, currency='USD')
> addPosLimit(
      portfolio=portfolio.st,
      symbol='GBPUSD',
      timestamp=initDate,
      maxpos=.orderqty)
> initAcct(account.st, portfolios=portfolio.st, initDate=initDate, currency='USD')
> initOrders(portfolio.st, initDate=initDate)

> enable.rule('luxor', 'chain', 'StopLoss')
```

- Rules can be turned on and off; turn on the StopLoss rule

Apply strategy, update portfolio, and plot results

R Code:

```
> out <- applyStrategy(strategy.st, portfolio.st)
> updatePortf(portfolio.st, Symbols='GBPUSD',
  Dates=paste(':', as.Date(Sys.time()), sep=''))

> chart.Posn(portfolio.st, "GBPUSD", TA="add_SMA(n=10,col=2);add_SMA(n=30,col=4)",
  theme=myTheme)

> PerformanceAnalytics:::textplot(t(tradeStats(portfolio.st, 'GBPUSD'))))

> ob <- getOrderBook(portfolio.st)$forex$GBPUSD
> ob.df <- data.frame(Date=time(ob),coredata(ob))
> PerformanceAnalytics:::textplot(ob.df, show.rownames=F)

> PerformanceAnalytics:::textplot(perTradeStats(portfolio.st, "GBPUSD"),
  show.rownames=F)

> chart.ME(portfolio.st, 'GBPUSD', type='MAE', scale='percent')
```


GBPUSD 30-minute bars



GBPUSD 30-minute bars

	GBPUSD
Portfolio	forex
Symbol	GBPUSD
Num.Txns	21
Num.Trades	10
Net.Trading.PL	-329.01
Avg.Trade.PL	-93.901
Med.Trade.PL	-116
Largest.Winner	364
Largest.Loser	-470.76
Gross.Profits	886
Gross.Losses	-1825.01
Std.Dev.Trade.PL	310.8347
Percent.Positive	40
Percent.Negative	60
Profit.Factor	0.4854768
Avg.Win.Trade	221.5
Med.Win.Trade	244
Avg.Losing.Trade	-304.1683
Med.Losing.Trade	-326
Avg.Daily.PL	-134.1443
Med.Daily.PL	-126
Std.Dev.Daily.PL	468.4729
Max.Drawdown	-1648.76
Profit.To.Max.Draw	-0.19955
Avg.WinLoss.Ratio	0.7282152
Med.WinLoss.Ratio	0.7484663
Max.Equity	400
Min.Equity	-1248.76
End.Equity	-329.01

Order book

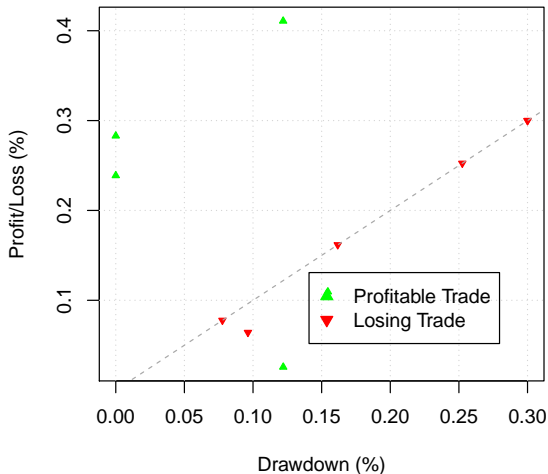
	Order.Qty	Order.Price	Order.Type	Order.Side	Order.Threshold	Order.Status	Order.StatusTime	Prefer	Order.Set	Txn.Fees	Rule
2002-10-21 23:30:00	1e+05	1.5447	stoplimit	long	5e-04	closed	2002-10-22 02:00:00	High	ocolong	0	EnterLONG
2002-10-22 02:00:00	all	1.5400659	stoplimit	long	-0.0046341	replaced	2002-10-22 17:00:00		ocolong	-6	StopLossLONG
2002-10-22 17:00:00	all	1.5453	market	long		closed	2002-10-22 17:30:00		ocolong	-6	Exit2SHORT
2002-10-22 17:00:00	-1e+05	1.5447	stoplimit	short	-5e-04	closed	2002-10-22 17:30:00	Low	ocoshort	0	EnterSHORT
2002-10-22 17:30:00	all	1.5493341	stoplimit	short	0.0046341	replaced	2002-10-23 02:30:00		ocoshort	-6	StopLossSHORT
2002-10-23 02:30:00	all	1.5485	market	short		closed	2002-10-23 03:00:00		ocoshort	-6	Exit2LONG
2002-10-23 02:30:00	1e+05	1.5492	stoplimit	long	5e-04	closed	2002-10-23 03:00:00	High	ocolong	0	EnterLONG
2002-10-23 03:00:00	all	1.5445524	stoplimit	long	-0.0046476	closed	2002-10-23 08:00:00		ocolong	-6	StopLossLONG
2002-10-23 17:00:00	-1e+05	1.5463	stoplimit	short	-5e-04	replaced	2002-10-23 22:00:00	Low	ocoshort	0	EnterSHORT
2002-10-23 22:00:00	1e+05	1.5491	stoplimit	long	5e-04	replaced	2002-10-24 01:30:00	High	ocolong	0	EnterLONG
2002-10-24 01:30:00	-1e+05	1.5459	stoplimit	short	-5e-04	closed	2002-10-24 03:00:00	Low	ocoshort	0	EnterSHORT
2002-10-24 03:00:00	all	1.5505377	stoplimit	short	0.0046377	replaced	2002-10-24 11:00:00		ocoshort	-6	StopLossSHORT
2002-10-24 11:00:00	all	1.5476	market	short		closed	2002-10-24 11:30:00		ocoshort	-6	Exit2LONG
2002-10-24 11:00:00	1e+05	1.5493	stoplimit	long	5e-04	closed	2002-10-24 12:00:00	High	ocolong	0	EnterLONG
2002-10-24 12:00:00	all	1.5446521	stoplimit	long	-0.0046479	replaced	2002-10-25 03:30:00		ocolong	-6	StopLossLONG
2002-10-25 03:30:00	all	1.5535	market	long		closed	2002-10-25 04:00:00		ocolong	-6	Exit2SHORT
2002-10-25 03:30:00	-1e+05	1.553	stoplimit	short	-5e-04	closed	2002-10-25 04:00:00	Low	ocoshort	0	EnterSHORT
2002-10-25 04:00:00	all	1.557659	stoplimit	short	0.004659	replaced	2002-10-25 11:30:00		ocoshort	-6	StopLossSHORT
2002-10-25 11:30:00	all	1.5528	market	short		closed	2002-10-25 12:00:00		ocoshort	-6	Exit2LONG
2002-10-25 11:30:00	1e+05	1.5533	stoplimit	long	5e-04	replaced	2002-10-25 12:00:00	High	ocolong	0	EnterLONG
2002-10-25 12:00:00	-1e+05	1.5508	stoplimit	short	-5e-04	closed	2002-10-27 23:30:00	Low	ocoshort	0	EnterSHORT
2002-10-27 23:30:00	all	1.5521425	stoplimit	short	0.0046425	closed	2002-10-28 09:30:00		ocoshort	-6	StopLossSHORT
2002-10-28 10:00:00	1e+05	1.555	stoplimit	long	5e-04	closed	2002-10-28 10:30:00	High	ocolong	0	EnterLONG
2002-10-28 10:30:00	all	1.550335	stoplimit	long	-0.004665	replaced	2002-10-28 23:30:00		ocolong	-6	StopLossLONG
2002-10-28 23:30:00	all	1.5582	market	long		closed	2002-10-29 00:00:00		ocolong	-6	Exit2SHORT
2002-10-28 23:30:00	-1e+05	1.5572	stoplimit	short	-5e-04	replaced	2002-10-29 06:30:00	Low	ocoshort	0	EnterSHORT
2002-10-29 06:30:00	1e+05	1.5594	stoplimit	long	5e-04	replaced	2002-10-29 07:00:00	High	ocolong	0	EnterLONG
2002-10-29 07:00:00	-1e+05	1.5572	stoplimit	short	-5e-04	closed	2002-10-29 07:30:00	Low	ocoshort	0	EnterSHORT
2002-10-29 07:30:00	all	1.5618716	stoplimit	short	0.0046716	replaced	2002-10-30 05:00:00		ocoshort	-6	StopLossSHORT
2002-10-30 05:00:00	all	1.5563	market	short		closed	2002-10-30 05:30:00		ocoshort	-6	Exit2LONG
2002-10-30 05:00:00	1e+05	1.5578	stoplimit	long	5e-04	closed	2002-10-30 09:30:00	High	ocolong	0	EnterLONG
2002-10-30 09:30:00	all	1.5531266	stoplimit	long	-0.0046734	replaced	2002-10-30 11:00:00		ocolong	-6	StopLossLONG
2002-10-30 11:00:00	all	1.5569	market	long		closed	2002-10-30 11:30:00		ocolong	-6	Exit2SHORT
2002-10-30 11:00:00	-1e+05	1.5558	stoplimit	short	-5e-04	replaced	2002-10-30 12:00:00	Low	ocoshort	0	EnterSHORT
2002-10-30 12:00:00	1e+05	1.5579	stoplimit	long	5e-04	closed	2002-10-30 13:00:00	High	ocolong	0	EnterLONG
2002-10-30 13:00:00	all	1.5532263	stoplimit	long	-0.0046737	open			ocolong	-6	StopLossLONG

Per-trade stats

	End	Init.Pos	Max.Pos	Num.Txns	Max.Notional.Cost	Net.Trading.PL	MAE	MFE	Pct.Net.Trading.PL	Pct.MAE	Pct.MFE	tick.Net.Trading.PL	tick.MAE	tick.MFE
2002-10-22 02:00:00	2002-10-22 17:30:00	1e+05	1e+05	2	154470	-120.00	-120.00	400	-0.0007768499	-0.0007768499	0.0025894996	-12.000	-12.000	40
2002-10-22 17:30:00	2002-10-23 03:00:00	-1e+05	-1e+05	2	-154470	-390.00	-390.00	180	-0.0025247621	-0.0025247621	0.0011652748	-39.000	-39.000	18
2002-10-23 03:00:00	2002-10-23 08:00:00	1e+05	1e+05	2	154920	-464.76	-464.76	0	-0.0030000000	-0.0030000000	0.0000000000	-46.476	-46.476	0
2002-10-24 03:00:00	2002-10-24 11:30:00	-1e+05	-1e+05	2	-154590	-250.00	-250.00	100	-0.0016171809	-0.0016171809	0.0006468724	-25.000	-25.000	10
2002-10-24 12:00:00	2002-10-25 04:00:00	1e+05	1e+05	2	154930	370.00	0.00	610	0.0023881753	0.0000000000	0.0039372620	37.000	0.000	61
2002-10-25 04:00:00	2002-10-25 12:00:00	-1e+05	-1e+05	2	-155300	170.00	-240.00	170	0.0010946555	-0.0015453960	0.0010946555	17.000	-24.000	17
2002-10-27 23:30:00	2002-10-28 09:30:00	-1e+05	-1e+05	2	-154750	-464.25	-464.25	190	-0.0030000000	-0.0030000000	0.0012277868	-46.425	-46.425	19
2002-10-28 10:30:00	2002-10-29 00:00:00	1e+05	1e+05	3	155500	440.00	0.00	550	0.0028295820	0.0000000000	1.0000000000	44.000	0.000	Inf
2002-10-29 07:30:00	2002-10-30 05:30:00	-1e+05	-1e+05	2	-155720	40.00	-190.00	210	0.0002568713	-0.0012201387	0.0013485744	4.000	-19.000	21
2002-10-30 09:30:00	2002-10-30 11:30:00	1e+05	1e+05	2	155780	-100.00	-150.00	0	-0.0006419309	-0.0009628964	0.0000000000	-10.000	-15.000	0
2002-10-30 13:00:00	2002-10-31 23:00:00	1e+05	1e+05	1	155790	640.00	-190.00	780	0.0041080942	-0.0012195905	0.0050067398	64.000	-19.000	78

Maximum adverse excursion

GBPUSD Maximum Adverse Excursion (MAE)



Outline

- 1 The Luxor strategy
- 2 Optimization in quantstrat
- 3 Stoploss orders
 - Overview of Stoploss order implementation
 - Optimizing the Luxor stoploss order

Define parameter range

R Code:

```
> add.distribution(strategy.st,  
  paramset.label = 'StopLoss',  
  component.type = 'chain',  
  component.label = 'StopLossLONG',  
  variable = list(threshold = .StopLoss),  
  label = 'StopLossLONG'  
)
```

- The input parameter threshold may range from 0.0005 to 0.6
- Optimized parameter name is StopLossLONG

Define parameter range

R Code:

```
> add.distribution(strategy.st,  
  paramset.label = 'StopLoss',  
  component.type = 'chain',  
  component.label = 'StopLossSHORT',  
  variable = list(threshold = .StopLoss),  
  label = 'StopLossSHORT'  
)
```

- The input parameter threshold may range from 0.0005 to 0.0240
- Optimized parameter name is StopLossSHORT

Define parameter constraint

R Code:

```
> add.constraint(strategy.st,  
  paramset.label = 'StopLoss',  
  distribution.label.1 = 'StopLossLONG',  
  distribution.label.2 = 'StopLossSHORT',  
  operator = '==',  
  label = 'StopLoss'  
)
```

- StopLossLONG must equal StopLossSHORT

Re-Initialize portfolio and account

R Code:

```
> rm.strat(portfolio.st)
> rm.strat(account.st)

> initPortf(portfolio.st, symbols='GBPUSD', initDate=initDate, currency='USD')

> addPosLimit(
    portfolio=portfolio.st,
    symbol='GBPUSD',
    timestamp=initDate,
    maxpos=.orderqty)

> initAcct(account.st, portfolios=portfolio.st,initDate=initDate,currency='USD')
> initOrders(portfolio.st, initDate=initDate)

> enable.rule('luxor', 'chain', 'StopLoss')
```

- Rules can be turned on and off; turn on the StopLoss rule

Optimize parameters

R Code:

```
> results <- apply.paramset(strategy.st, paramset.label='StopLoss',
  portfolio.st=portfolio.st, account.st=account.st, nsamples=80, verbose=TRUE)

> tS <- results$tradeStats
> idx <- order(tS[,1])
> tS <- tS[idx,]
> PerformanceAnalytics:::textplot(t(tS)[,1:5])

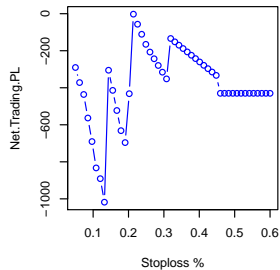
> par(mfrow=c(1,3))
> plot(100*tS$StopLossLONG, tS$Net.Trading.PL, type='b', xlab='Stoploss %',
  ylab='Net.Trading.PL', main='Net Profit vs Stop Loss',col=4)
> plot(100*tS$StopLossLONG, tS$Max.Drawdown, type='b', xlab='Stoploss %',
  ylab='Max.Drawdown', main='MaxDrawdown vs Stop Loss',col=4)
> plot(100*tS$StopLossLONG, tS$Profit.To.Max.Draw, type='b', xlab='Stoploss %',
  ylab='Profit.To.Max.Draw', main='Return/MaxDD vs Stop Loss',col=4)
> par(mfrow=c(1,1))
```

paramset trade stats

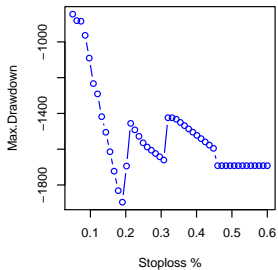
	1	50	99	148	197
StopLossLONG	0.0005000000	0.0006170213	0.0007340426	0.0008510638	0.0009680851
StopLossSHORT	0.0005000000	0.0006170213	0.0007340426	0.0008510638	0.0009680851
Portfolio	forex.1	forex.50	forex.99	forex.148	forex.197
Symbol	GBPUSD	GBPUSD	GBPUSD	GBPUSD	GBPUSD
Num.Txns	22	22	22	22	22
Num.Trades	11	11	11	11	11
Net.Trading.PL	-290.590000	-371.894752	-436.255284	-563.407092	-690.558901
Avg.Trade.PL	-26.41727	-33.80861	-39.65957	-51.21883	-62.77808
Med.Trade.PL	-86.0000	-102.1196	-119.9968	-137.8468	-155.9757
Largest.Winner	364	364	364	364	364
Largest.Loser	-158.6667	-158.6667	-158.6667	-158.6667	-158.6667
Gross.Profits	688	688	688	688	688
Gross.Losses	-978.590	-1059.895	-1124.255	-1251.407	-1378.559
Std.Dev.Trade.PL	185.3758	188.1326	190.2429	195.7498	201.5232
Percent.Positive	18.18182	18.18182	18.18182	18.18182	18.18182
Percent.Negative	81.81818	81.81818	81.81818	81.81818	81.81818
Profit.Factor	0.7030524	0.6491211	0.6119607	0.5497811	0.4990719
Avg.Win.Trade	344.0000	344.0000	344.0000	344.0000	344.0000
Med.Win.Trade	344	344	344	344	344
Avg.Losing.Trade	-108.7322	-117.7661	-124.9173	-139.0452	-153.1732
Med.Losing.Trade	-91.33333	-102.12574	-120.30511	-138.17021	-156.34362
Avg.Daily.PL	-41.51286	-53.12782	-62.32218	-80.48673	-98.65127
Med.Daily.PL	-152.3333	-152.3333	-126.0000	-137.5660	-155.6563
Std.Dev.Daily.PL	196.9463	201.7604	201.7742	205.2010	209.6238
Max.Drawdown	-845.0283	-881.2335	-884.5811	-963.4071	-1090.5589
Profit.To.Max.Draw	-0.34388196	-0.42201611	-0.49317728	-0.58480688	-0.63321559
Avg.WinLoss.Ratio	3.1637356	2.9210447	2.7538229	2.4740151	2.2458235
Med.WinLoss.Ratio	3.7664234	3.3683965	2.8593965	2.4896828	2.2002817
Max.Equity	400	400	400	400	400
Min.Equity	-445.0283	-481.2335	-484.5811	-563.4071	-690.5589
End.Equity	-290.590000	-371.894752	-436.255284	-563.407092	-690.558901

Stoploss analysis

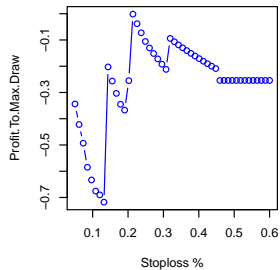
Net Profit vs Stop Loss

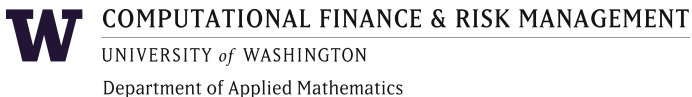


MaxDrawdown vs Stop Loss



Return/MaxDD vs Stop Loss





`http://depts.washington.edu/compfin`