COMPUTATIONAL FINANCE & RISK MANAGEMENT

UNIVERSITY *of* WASHINGTON

Department of Applied Mathematics

# Introduction to Trading Systems

Guy Yollin

Applied Mathematics
University of Washington

# Outline

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.x.x/library/quantstrat/demo`

# Outline

1. The Luxor strategy
   - Introduction to the Luxor strategy
   - The Luxor strategy implemented in quantstrat

2. Optimization in quantstrat

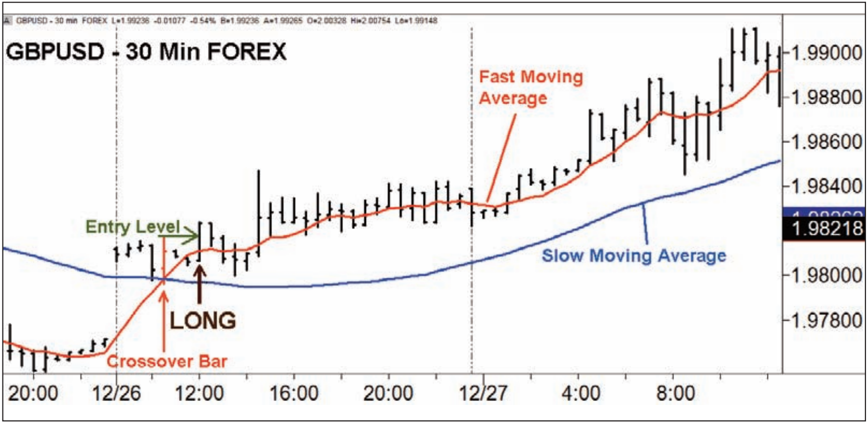
3. Stoploss orders

# Outline

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

# Luxor strategy

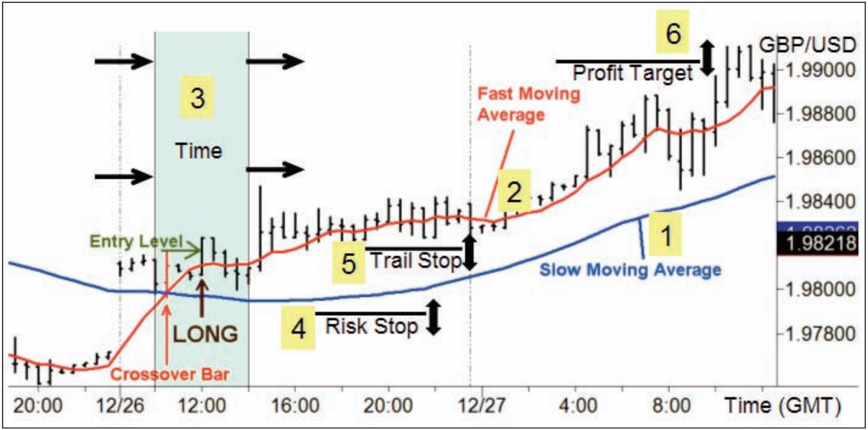
The Luxor strategy is presented in Trading Systems by Emilio Tomasini and Urban Jaekle (Harriman House, 2009)



- 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 targe |
| 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

# Outline

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

```
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

```
dim(GBPUSD)
```

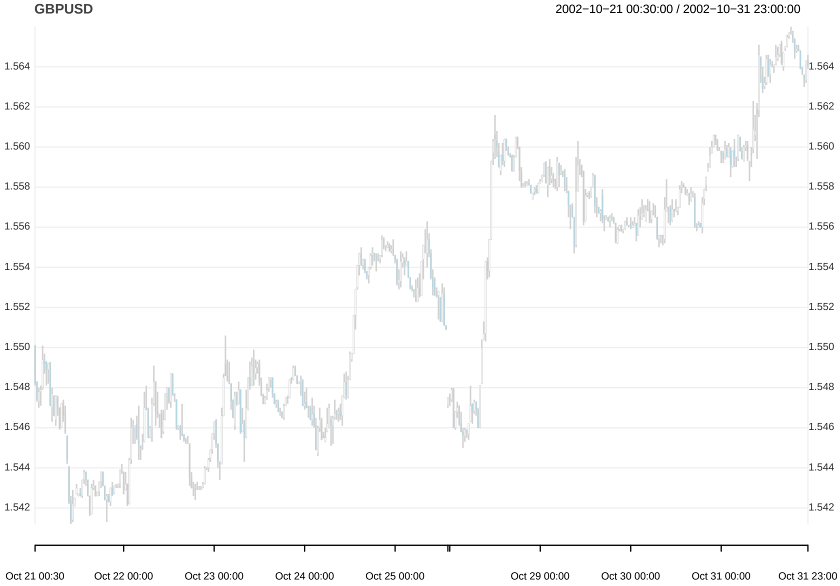
```
## [1] 411   5
```

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

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

# 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

```
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

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

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

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

```r
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

```
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
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

```
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

# Short exit rule

```
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

# Apply strategy, update portfolio, and plot results

```
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)
```

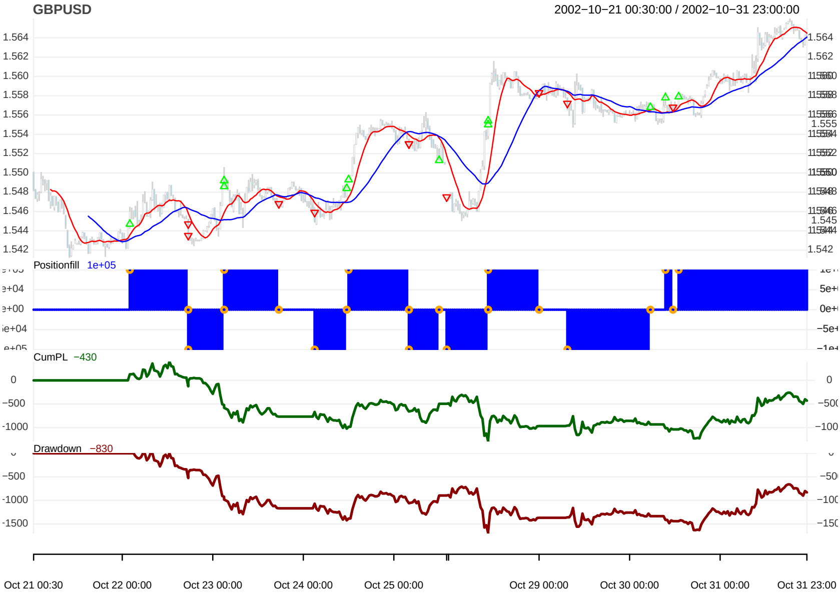
```
PerformanceAnalytics:::textplot(t(tradeStats(portfolio.st, 'GBPUSD')))
```

```
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)
```

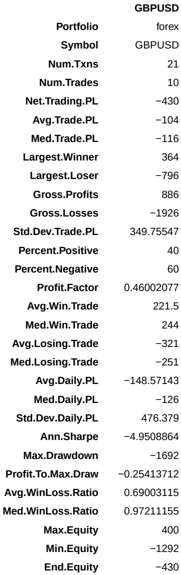
```
ob <- getOrderBook(portfolio.st)$forex$GBPUSD
ob.df <- data.frame(Date=time(ob),coredata(ob))
PerformanceAnalytics:::textplot(ob.df,show.rownames=F)
```

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

# GBPUSD 30-minute bars

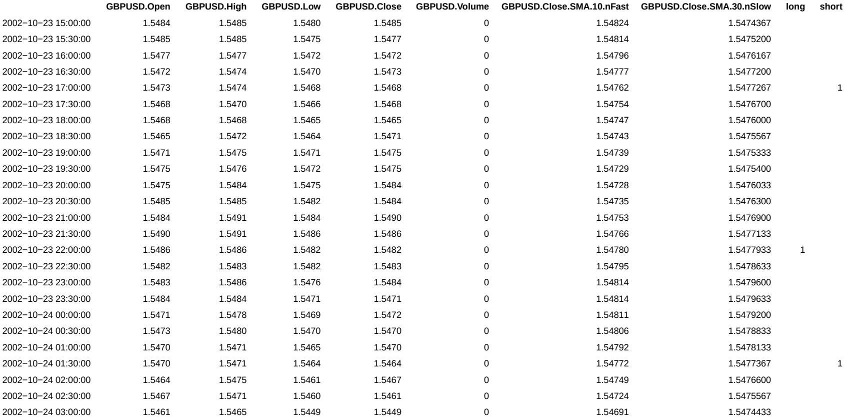


# GBPUSD 30-minute bars



| | 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 | 349.75547 |
| Percent.Positive | 40 |
| Percent.Negative | 60 |
| Profit.Factor | 0.46002077 |
| Avg.Win.Trade | 221.5 |
| Med.Win.Trade | 244 |
| Avg.Losing.Trade | −321 |
| Med.Losing.Trade | −251 |
| Avg.Daily.PL | −148.57143 |
| Med.Daily.PL | −126 |
| Std.Dev.Daily.PL | 476.379 |
| Ann.Sharpe | −4.9508864 |
| Max.Drawdown | −1692 |
| Profit.To.Max.Draw | −0.25413712 |
| Avg.WinLoss.Ratio | 0.69003115 |
| Med.WinLoss.Ratio | 0.97211155 |
| Max.Equity | 400 |
| Min.Equity | −1292 |
| End.Equity | −430 |

# The mktdata object

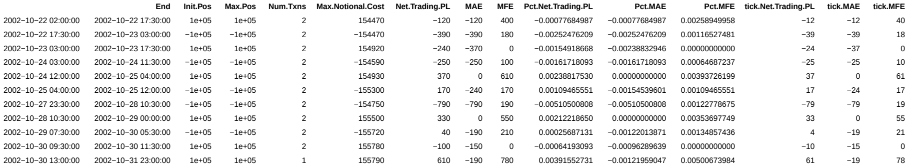


| | GBPUSD.Open | GBPUSD.High | GBPUSD.Low | GBPUSD.Close | GBPUSD.Volume | GBPUSD.Close.SMA.10.nFast | GBPUSD.Close.SMA.30.nSlow | long | short |
|---|---|---|---|---|---|---|---|---|---|
| 2002−10−23 15:00:00 | 1.5484 | 1.5485 | 1.5480 | 1.5485 | 0 | 1.54824 | 1.5474367 | | |
| 2002−10−23 15:30:00 | 1.5485 | 1.5485 | 1.5475 | 1.5477 | 0 | 1.54814 | 1.5475200 | | |
| 2002−10−23 16:00:00 | 1.5477 | 1.5477 | 1.5472 | 1.5472 | 0 | 1.54796 | 1.5476167 | | |
| 2002−10−23 16:30:00 | 1.5472 | 1.5474 | 1.5470 | 1.5473 | 0 | 1.54777 | 1.5477200 | | |
| 2002−10−23 17:00:00 | 1.5473 | 1.5474 | 1.5468 | 1.5468 | 0 | 1.54762 | 1.5477267 | | 1 |
| 2002−10−23 17:30:00 | 1.5468 | 1.5470 | 1.5466 | 1.5468 | 0 | 1.54754 | 1.5476700 | | |
| 2002−10−23 18:00:00 | 1.5468 | 1.5468 | 1.5465 | 1.5465 | 0 | 1.54747 | 1.5476000 | | |
| 2002−10−23 18:30:00 | 1.5465 | 1.5472 | 1.5464 | 1.5471 | 0 | 1.54743 | 1.5475567 | | |
| 2002−10−23 19:00:00 | 1.5471 | 1.5475 | 1.5471 | 1.5475 | 0 | 1.54739 | 1.5475333 | | |
| 2002−10−23 19:30:00 | 1.5475 | 1.5476 | 1.5472 | 1.5475 | 0 | 1.54729 | 1.5475400 | | |
| 2002−10−23 20:00:00 | 1.5475 | 1.5484 | 1.5475 | 1.5484 | 0 | 1.54728 | 1.5476033 | | |
| 2002−10−23 20:30:00 | 1.5485 | 1.5485 | 1.5482 | 1.5484 | 0 | 1.54735 | 1.5476300 | | |
| 2002−10−23 21:00:00 | 1.5484 | 1.5491 | 1.5484 | 1.5490 | 0 | 1.54753 | 1.5476900 | | |
| 2002−10−23 21:30:00 | 1.5490 | 1.5491 | 1.5486 | 1.5486 | 0 | 1.54766 | 1.5477133 | | |
| 2002−10−23 22:00:00 | 1.5486 | 1.5486 | 1.5482 | 1.5482 | 0 | 1.54780 | 1.5477933 | 1 | |
| 2002−10−23 22:30:00 | 1.5482 | 1.5483 | 1.5482 | 1.5483 | 0 | 1.54795 | 1.5478633 | | |
| 2002−10−23 23:00:00 | 1.5483 | 1.5486 | 1.5476 | 1.5484 | 0 | 1.54814 | 1.5479600 | | |
| 2002−10−23 23:30:00 | 1.5484 | 1.5484 | 1.5471 | 1.5471 | 0 | 1.54814 | 1.5479633 | | |
| 2002−10−24 00:00:00 | 1.5471 | 1.5478 | 1.5469 | 1.5472 | 0 | 1.54811 | 1.5479200 | | |
| 2002−10−24 00:30:00 | 1.5473 | 1.5480 | 1.5470 | 1.5470 | 0 | 1.54806 | 1.5478833 | | |
| 2002−10−24 01:00:00 | 1.5470 | 1.5471 | 1.5465 | 1.5470 | 0 | 1.54792 | 1.5478133 | | |
| 2002−10−24 01:30:00 | 1.5470 | 1.5471 | 1.5464 | 1.5464 | 0 | 1.54772 | 1.5477367 | | 1 |
| 2002−10−24 02:00:00 | 1.5464 | 1.5475 | 1.5461 | 1.5467 | 0 | 1.54749 | 1.5476600 | | |
| 2002−10−24 02:30:00 | 1.5467 | 1.5471 | 1.5460 | 1.5461 | 0 | 1.54724 | 1.5475567 | | |
| 2002−10−24 03:00:00 | 1.5461 | 1.5465 | 1.5449 | 1.5449 | 0 | 1.54691 | 1.5474433 | | |

# Order book

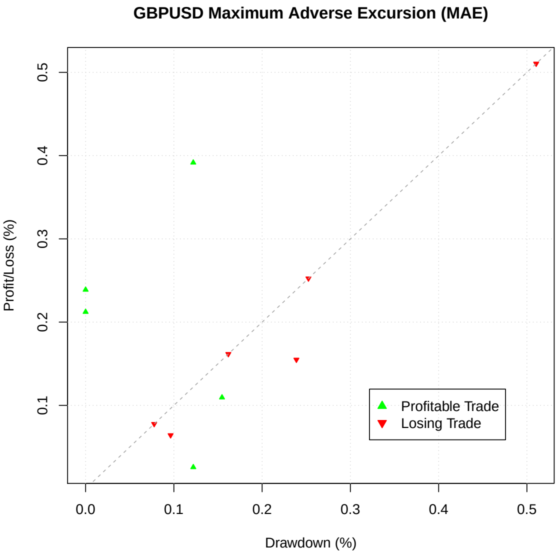
| | Order.Qty | Order.Price | Order.Type | Order.Side | Order.Threshold | Order.Status | Order.StatusTime | Prefer | Order.Set | Txn.Fees | Rule | Time.In.Force |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2002−10−21 23:30:00 | 1e+05 | 1.5447 | stoplimit | long | 5e−04 | closed | 2002−10−22 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 | −0.00077684987 | −0.00077684987 | 0.00258949958 | −12 | −12 | 40 |
| 2002−10−22 17:30:00 | 2002−10−23 03:00:00 | −1e+05 | −1e+05 | 2 | −154470 | −390 | −390 | 180 | −0.00252476209 | −0.00252476209 | 0.00116527481 | −39 | −39 | 18 |
| 2002−10−23 03:00:00 | 2002−10−23 17:30:00 | 1e+05 | 1e+05 | 2 | 154920 | −240 | −370 | 0 | −0.00154918668 | −0.00238832946 | 0.00000000000 | −24 | −37 | 0 |
| 2002−10−24 03:00:00 | 2002−10−24 11:30:00 | −1e+05 | −1e+05 | 2 | −154590 | −250 | −250 | 100 | −0.00161718093 | −0.00161718093 | 0.00064687237 | −25 | −25 | 10 |
| 2002−10−24 12:00:00 | 2002−10−25 04:00:00 | 1e+05 | 1e+05 | 2 | 154930 | 370 | 0 | 610 | 0.00238817530 | 0.00000000000 | 0.00393726199 | 37 | 0 | 61 |
| 2002−10−25 04:00:00 | 2002−10−25 12:00:00 | −1e+05 | −1e+05 | 2 | −155300 | 170 | −240 | 170 | 0.00109465551 | −0.00154539601 | 0.00109465551 | 17 | −24 | 17 |
| 2002−10−27 23:30:00 | 2002−10−28 10:30:00 | −1e+05 | −1e+05 | 2 | −154750 | −790 | −790 | 190 | −0.00510500808 | −0.00510500808 | 0.00122778675 | −79 | −79 | 19 |
| 2002−10−28 10:30:00 | 2002−10−29 00:00:00 | 1e+05 | 1e+05 | 2 | 155500 | 330 | 0 | 550 | 0.00212218650 | 0.00000000000 | 0.00353697749 | 33 | 0 | 55 |
| 2002−10−29 07:30:00 | 2002−10−30 05:30:00 | −1e+05 | −1e+05 | 2 | −155720 | 40 | −190 | 210 | 0.00025687131 | −0.00122013871 | 0.00134857436 | 4 | −19 | 21 |
| 2002−10−30 09:30:00 | 2002−10−30 11:30:00 | 1e+05 | 1e+05 | 2 | 155780 | −100 | −150 | 0 | −0.00064193093 | −0.00096289639 | 0.00000000000 | −10 | −15 | 0 |
| 2002−10−30 13:00:00 | 2002−10−31 23:00:00 | 1e+05 | 1e+05 | 1 | 155790 | 610 | −190 | 780 | 0.00391552731 | −0.00121959047 | 0.00500673984 | 61 | −19 | 78 |

# Maximum adverse excursion



# 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. fastMA = 1:20)

constraint — A constraint is a relationship that must be true between two distributions in a paramset (e.g. fastMA < 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.distribution.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
args(add.distribution)


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

```r
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
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
args(add.distribution.constraint)

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

```r
add.distribution.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
rm.strat(portfolio.st)
rm.strat(account.st)
```

```r
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

```
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] 1.0000000 1.4142136 1.7320508 2.0000000 2.2360680 2.4494897 2.6457513
## [8] 2.8284271
```

- 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

```
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,
##     paramsets, ...)
## NULL
```

Main arguments:

| | |
|---|---|
| strategy.st | text name of the strategy |
| paramset.label | text name of the paramset |
| portfolio.st | text name of the portfolio |
| nsamples | if nsamples $>$ 0 then take a sample of size nsamples from the paramset |

# Optimize parameters

```r
results <- apply.paramset(strategy.st, paramset.label='SMA',
  portfolio.st=portfolio.st, account.st=account.st, nsamples=0)
```

```r
head(names(results),20)
```

```
##  [1] "forex.1"    "tradeStats" "forex.2"    "forex.3"    "forex.4"
##  [6] "forex.5"    "forex.6"    "forex.7"    "forex.8"    "forex.9"
## [11] "forex.10"   "forex.11"   "forex.12"   "forex.13"   "forex.14"
## [16] "forex.15"   "forex.16"   "forex.17"   "forex.18"   "forex.19"
```

```r
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

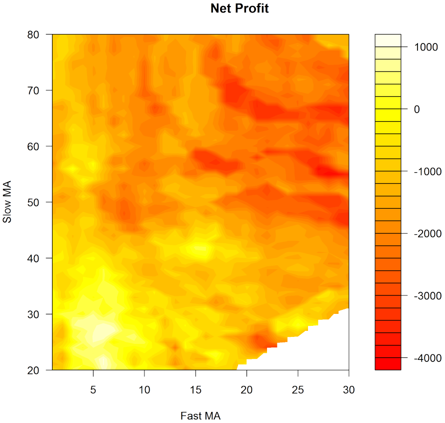
| | 1 | 20 | 40 | 61 | 83 | 106 | 130 | 155 | 181 | 208 |
|---|---|---|---|---|---|---|---|---|---|---|
| **nFAST** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **nSLOW** | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| **Portfolio** | forex.1 | forex.20 | forex.40 | forex.61 | forex.83 | forex.106 | forex.130 | forex.155 | forex.181 | forex.208 |
| **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** | −372.000000 | −1214.000000 | −808.000000 | −716.000000 | −796.000000 | −1136.000000 | −918.000000 | −688.000000 | −418.000000 | −280.000000 |
| **Avg.Trade.PL** | −11.55555556 | −39.79310345 | −26.71428571 | −25.23076923 | −28.30769231 | −41.38461538 | −37.30434783 | −27.30434783 | −18.17391304 | −14.00000000 |
| **Med.Trade.PL** | −86.0000000 | −116.0000000 | −106.0000000 | −96.0000000 | −96.0000000 | −96.0000000 | −96.0000000 | −96.0000000 | −96.0000000 | −96.0000000 |
| **Largest.Winner** | 944.000000 | 944.000000 | 944.000000 | 954.000000 | 944.000000 | 674.000000 | 674.000000 | 674.000000 | 694.000000 | 674.000000 |
| **Largest.Loser** | −336 | −506 | −506 | −506 | −506 | −506 | −506 | −506 | −506 | −406 |
| **Gross.Profits** | 2282.000000 | 2208.000000 | 2208.000000 | 2134.000000 | 2054.000000 | 1714.000000 | 1754.000000 | 1984.000000 | 2008.000000 | 2040.000000 |
| **Gross.Losses** | −2594.0000 | −3362.0000 | −2956.0000 | −2790.0000 | −2790.0000 | −2790.0000 | −2612.0000 | −2612.0000 | −2426.0000 | −2320.0000 |
| **Std.Dev.Trade.PL** | 261.08256 | 268.25379 | 266.72122 | 280.69803 | 273.90959 | 234.96776 | 249.75086 | 259.25250 | 257.85755 | 284.94875 |
| **Percent.Positive** | 29.6296296 | 24.1379310 | 25.0000000 | 23.0769231 | 23.0769231 | 23.0769231 | 26.0869565 | 26.0869565 | 30.4347826 | 25.0000000 |
| **Percent.Negative** | 70.370370 | 75.862069 | 75.000000 | 76.923077 | 76.923077 | 76.923077 | 73.913043 | 73.913043 | 69.565217 | 75.000000 |
| **Profit.Factor** | 0.8797224364 | 0.6567519334 | 0.7469553451 | 0.7648745520 | 0.7362007168 | 0.6143369176 | 0.6715160796 | 0.7595712098 | 0.8276999176 | 0.8793103448 |
| **Avg.Win.Trade** | 285.250000 | 315.428571 | 315.428571 | 355.666667 | 342.333333 | 285.666667 | 292.333333 | 330.666667 | 286.857143 | 408.000000 |
| **Med.Win.Trade** | 164.000000 | 194.000000 | 194.000000 | 199.000000 | 199.000000 | 164.000000 | 184.000000 | 264.000000 | 214.000000 | 524.000000 |
| **Avg.Losing.Trade** | −136.52632 | −152.81818 | −140.76190 | −139.50000 | −139.50000 | −139.50000 | −153.64706 | −153.64706 | −151.62500 | −154.66667 |
| **Med.Losing.Trade** | −136.000000 | −136.000000 | −136.000000 | −131.000000 | −131.000000 | −131.000000 | −146.000000 | −146.000000 | −121.000000 | −176.000000 |
| **Avg.Daily.PL** | −34.66666667 | −128.22222222 | −83.11111111 | −72.88888889 | −81.77777778 | −119.55555556 | −107.25000000 | −78.50000000 | −52.25000000 | −35.00000000 |
| **Med.Daily.PL** | −8.6000000e+01 | −8.6000000e+01 | −8.4000000e+01 | −1.6000000e+01 | −1.6000000e+01 | −1.6000000e+01 | −1.1200000e+02 | −1.1200000e+02 | −3.7000000e+01 | −9.2000000e+01 |
| **Std.Dev.Daily.PL** | 524.83331 | 677.74143 | 606.37786 | 615.15048 | 621.84439 | 536.84614 | 601.95295 | 625.81764 | 601.09561 | 540.68317 |
| **Ann.Sharpe** | −1.0485543991 | −3.0033056687 | −2.1757852167 | −1.8809629198 | −2.0876315668 | −3.5352505431 | −2.8283621856 | −1.9912332197 | −1.3798853736 | −1.0276032444 |
| **Max.Drawdown** | −1118.0000 | −1960.0000 | −1624.0000 | −1558.0000 | −1604.0000 | −1604.0000 | −1870.0000 | −1870.0000 | −1684.0000 | −1568.0000 |
| **Profit.To.Max.Draw** | −0.3327370304 | −0.6193877551 | −0.4975369458 | −0.4595635430 | −0.4962593516 | −0.7082294264 | −0.4909090909 | −0.3679144385 | −0.2482185273 | −0.1785714286 |
| **Avg.WinLoss.Ratio** | 2.0893407864 | 2.0640775049 | 2.2408660352 | 2.5495818399 | 2.4540023895 | 2.0477897252 | 1.9026288923 | 2.1521184278 | 1.8918855259 | 2.6379310345 |
| **Med.WinLoss.Ratio** | 1.2058823529 | 1.4264705882 | 1.4264705882 | 1.5190839695 | 1.5190839695 | 1.2519083969 | 1.2602739726 | 1.8082191781 | 1.7685950413 | 2.9772727273 |
| **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 | −1762.00000 | −1356.00000 | −1250.00000 | −1296.00000 | −1296.00000 | −1546.00000 | −1546.00000 | −1360.00000 | −1254.00000 |
| **End.Equity** | −372.000000 | −1214.000000 | −808.000000 | −716.000000 | −796.000000 | −1136.000000 | −918.000000 | −688.000000 | −418.000000 | −280.000000 |

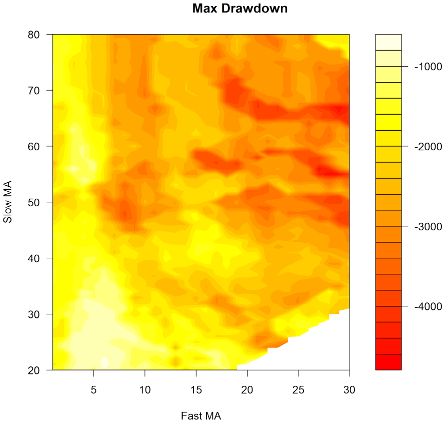
# Heatmaps of strategy performance

```
# 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



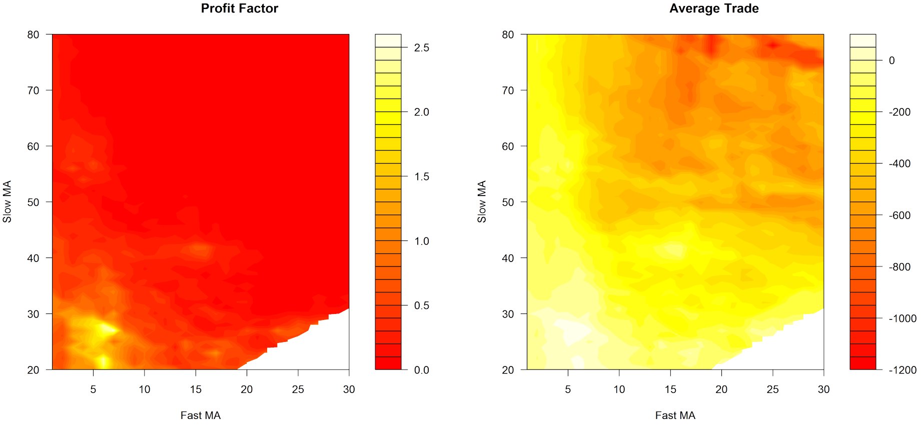


# Heatmaps of strategy performance

```r
# 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")
```

```r
# 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

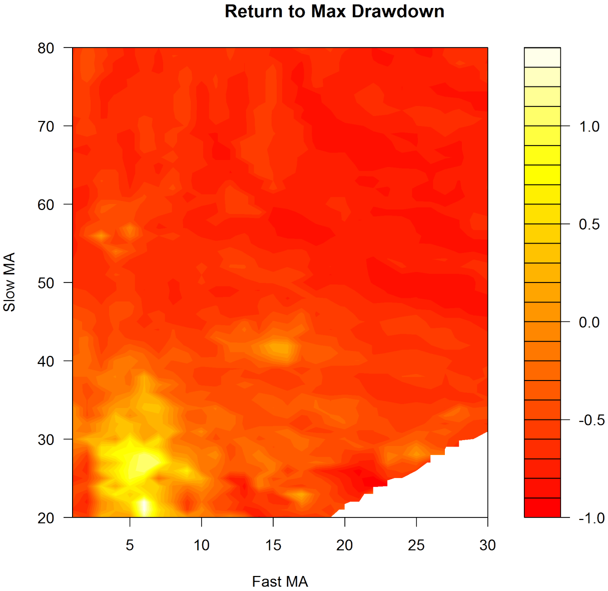


# Plot return to maximum drawdown

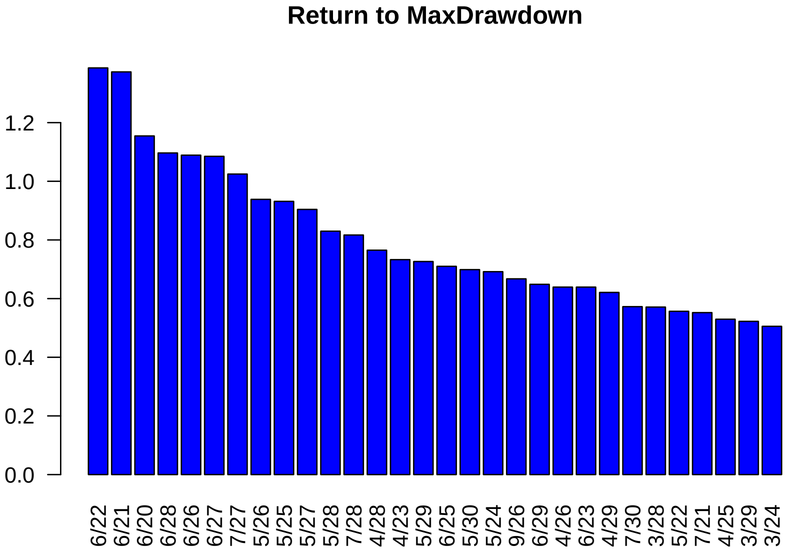
```r
# return to maxdd
z <- tapply(X=tS[,"Profit.To.Max.Draw"],
  INDEX=list(Fast=tS[,1],Slow=tS[,2]),FUN=sum)
x <- as.numeric(rownames(z))
y <- as.numeric(colnames(z))
filled.contour(x=x,y=y,z=z,color = heat.colors,xlab="Fast MA",ylab="Slow MA")
title("Return to Max Drawdown")
```

```r
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



# The tradeGraphs function

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

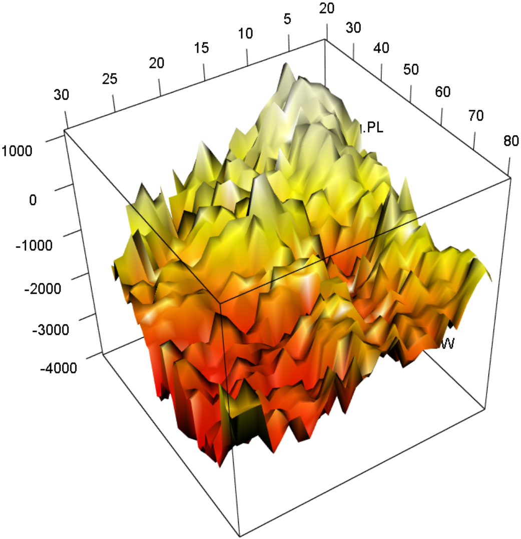
```
args(tradeGraphs)

## 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 = '')
```

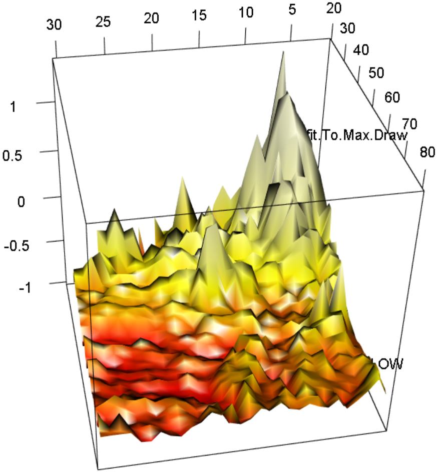
Main arguments:

| | |
|---|---|
| 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



# Outline

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

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

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

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

# Crossover signals

```
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

```
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

```
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

```
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

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

```r
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)
```

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

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

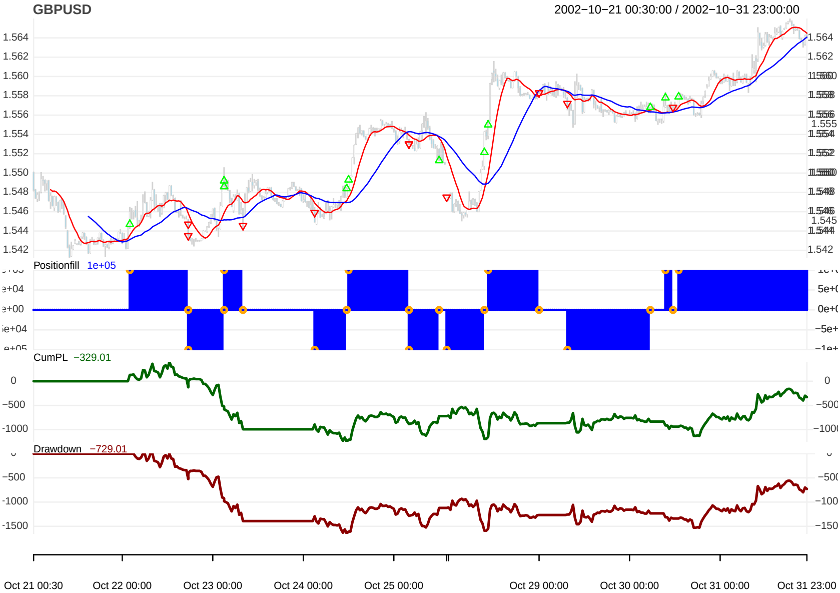
```r
PerformanceAnalytics:::textplot(t(tradeStats(portfolio.st, 'GBPUSD')))
```

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

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

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

# GBPUSD 30-minute bars



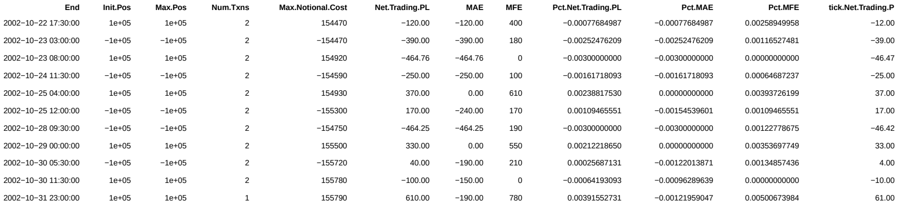
# GBPUSD 30-minute bars

| | 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** | 310.83473 |
| **Percent.Positive** | 40 |
| **Percent.Negative** | 60 |
| **Profit.Factor** | 0.48547679 |
| **Avg.Win.Trade** | 221.5 |
| **Med.Win.Trade** | 244 |
| **Avg.Losing.Trade** | −304.16833 |
| **Med.Losing.Trade** | −326 |
| **Avg.Daily.PL** | −134.14429 |
| **Med.Daily.PL** | −126 |
| **Std.Dev.Daily.PL** | 468.4729 |
| **Ann.Sharpe** | −4.5455661 |
| **Max.Drawdown** | −1648.76 |
| **Profit.To.Max.Draw** | −0.19954996 |
| **Avg.WinLoss.Ratio** | 0.72821519 |
| **Med.WinLoss.Ratio** | 0.74846626 |
| **Max.Equity** | 400 |
| **Min.Equity** | −1248.76 |
| **End.Equity** | −329.01 |

# Order book

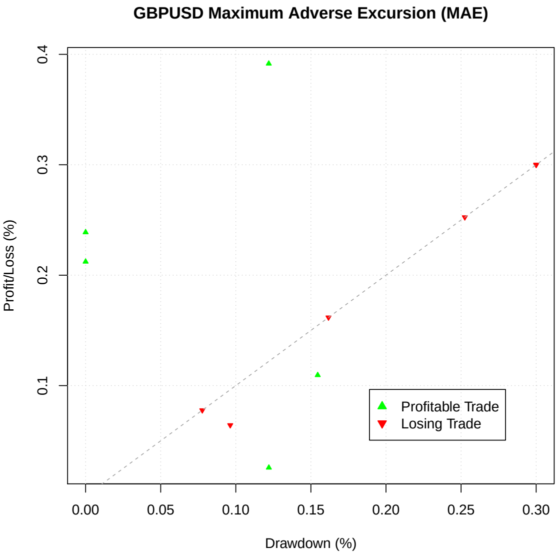
| | Order.Qty | Order.Price | Order.Type | Order.Side | Order.Threshold | Order.Status | Order.StatusTime | Prefer | Order.Set | Txn.Fees | Rule | Time.In.Forc |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2002−10−21 23:30:00 | 1e+05 | 1.5447 | stoplimit | long | 5e−04 | closed | 2002−10−22 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.P |
|---:|---:|---:|---:|---:|---:|---:|---:|---:|---:|---:|---:|
| 2002−10−22 17:30:00 | 1e+05 | 1e+05 | 2 | 154470 | −120.00 | −120.00 | 400 | −0.00077684987 | −0.00077684987 | 0.00258949958 | −12.00 |
| 2002−10−23 03:00:00 | −1e+05 | −1e+05 | 2 | −154470 | −390.00 | −390.00 | 180 | −0.00252476209 | −0.00252476209 | 0.00116527481 | −39.00 |
| 2002−10−23 08:00:00 | 1e+05 | 1e+05 | 2 | 154920 | −464.76 | −464.76 | 0 | −0.00300000000 | −0.00300000000 | 0.00000000000 | −46.47 |
| 2002−10−24 11:30:00 | −1e+05 | −1e+05 | 2 | −154590 | −250.00 | −250.00 | 100 | −0.00161718093 | −0.00161718093 | 0.00064687237 | −25.00 |
| 2002−10−25 04:00:00 | 1e+05 | 1e+05 | 2 | 154930 | 370.00 | 0.00 | 610 | 0.00238817530 | 0.00000000000 | 0.00393726199 | 37.00 |
| 2002−10−25 12:00:00 | −1e+05 | −1e+05 | 2 | −155300 | 170.00 | −240.00 | 170 | 0.00109465551 | −0.00154539601 | 0.00109465551 | 17.00 |
| 2002−10−28 09:30:00 | −1e+05 | −1e+05 | 2 | −154750 | −464.25 | −464.25 | 190 | −0.00300000000 | −0.00300000000 | 0.00122778675 | −46.42 |
| 2002−10−29 00:00:00 | 1e+05 | 1e+05 | 2 | 155500 | 330.00 | 0.00 | 550 | 0.00212218650 | 0.00000000000 | 0.00353697749 | 33.00 |
| 2002−10−30 05:30:00 | −1e+05 | −1e+05 | 2 | −155720 | 40.00 | −190.00 | 210 | 0.00025687131 | −0.00122013871 | 0.00134857436 | 4.00 |
| 2002−10−30 11:30:00 | 1e+05 | 1e+05 | 2 | 155780 | −100.00 | −150.00 | 0 | −0.00064193093 | −0.00096289639 | 0.00000000000 | −10.00 |
| 2002−10−31 23:00:00 | 1e+05 | 1e+05 | 1 | 155790 | 610.00 | −190.00 | 780 | 0.00391552731 | −0.00121959047 | 0.00500673984 | 61.00 |

# Maximum adverse excursion



# 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

```
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

```
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

```
add.distribution.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

```
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
results <- apply.paramset(strategy.st, paramset.label='StopLoss',
  portfolio.st=portfolio.st, account.st=account.st, nsamples=0, verbose=TRUE)
```

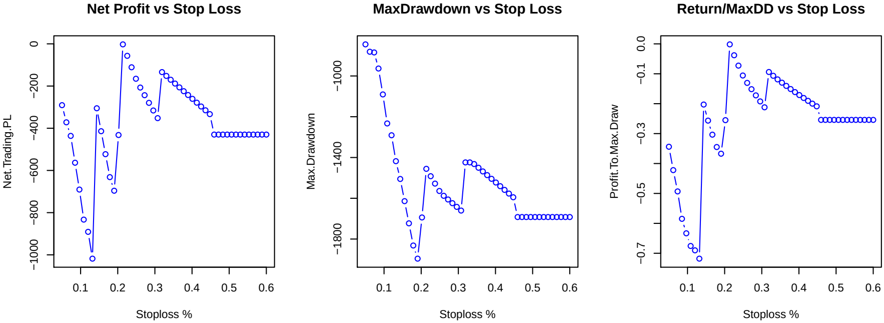
```r
tS <- results$tradeStats
idx <- order(tS[,1])
tS <- tS[idx,]
PerformanceAnalytics:::textplot(t(tS)[,1:5])
```

```r
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 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| **StopLossLONG** | 0.00050000000 | 0.00061702128 | 0.00073404255 | 0.00085106383 | 0.00096808511 |
| **StopLossSHORT** | 0.00050000000 | 0.00061702128 | 0.00073404255 | 0.00085106383 | 0.00096808511 |
| **Portfolio** | forex.1 | forex.2 | forex.3 | forex.4 | forex.5 |
| **Symbol** | GBPUSD | GBPUSD | GBPUSD | GBPUSD | GBPUSD |
| **Num.Txns** | 22 | 22 | 22 | 22 | 22 |
| **Num.Trades** | 11 | 11 | 11 | 11 | 11 |
| **Net.Trading.PL** | −290.5900000 | −371.8947518 | −436.2552837 | −563.4070922 | −690.5589007 |
| **Avg.Trade.PL** | −26.417273 | −33.808614 | −39.659571 | −51.218827 | −62.778082 |
| **Med.Trade.PL** | −86.00000 | −102.11957 | −119.99681 | −137.84681 | −155.97574 |
| **Largest.Winner** | 364 | 364 | 364 | 364 | 364 |
| **Largest.Loser** | −158.66667 | −158.66667 | −158.66667 | −158.66667 | −158.66667 |
| **Gross.Profits** | 688 | 688 | 688 | 688 | 688 |
| **Gross.Losses** | −978.5900 | −1059.8948 | −1124.2553 | −1251.4071 | −1378.5589 |
| **Std.Dev.Trade.PL** | 185.37580 | 188.13257 | 190.24294 | 195.74982 | 201.52316 |
| **Percent.Positive** | 18.181818 | 18.181818 | 18.181818 | 18.181818 | 18.181818 |
| **Percent.Negative** | 81.818182 | 81.818182 | 81.818182 | 81.818182 | 81.818182 |
| **Profit.Factor** | 0.70305235 | 0.64912106 | 0.61196066 | 0.54978113 | 0.49907189 |
| **Avg.Win.Trade** | 344.00000 | 344.00000 | 344.00000 | 344.00000 | 344.00000 |
| **Med.Win.Trade** | 344 | 344 | 344 | 344 | 344 |
| **Avg.Losing.Trade** | −108.73222 | −117.76608 | −124.91725 | −139.04523 | −153.17321 |
| **Med.Losing.Trade** | −91.333333 | −102.125745 | −120.305106 | −138.170213 | −156.343617 |
| **Avg.Daily.PL** | −41.512857 | −53.127822 | −62.322183 | −80.486727 | −98.651272 |
| **Med.Daily.PL** | −152.33333 | −152.33333 | −126.00000 | −137.56596 | −155.65628 |
| **Std.Dev.Daily.PL** | 196.94629 | 201.76042 | 201.77419 | 205.20100 | 209.62375 |
| **Ann.Sharpe** | −3.3460705 | −4.1800964 | −4.9031742 | −6.2265153 | −7.4707201 |
| **Max.Drawdown** | −845.02833 | −881.23355 | −884.58106 | −963.40709 | −1090.55890 |
| **Profit.To.Max.Draw** | −0.3438819606 | −0.4220161085 | −0.4931772808 | −0.5848068763 | −0.6332155927 |
| **Avg.WinLoss.Ratio** | 3.16373558 | 2.92104475 | 2.75382295 | 2.47401507 | 2.24582352 |
| **Med.WinLoss.Ratio** | 3.76642336 | 3.36839649 | 2.85939650 | 2.48968278 | 2.20028170 |
| **Max.Equity** | 400 | 400 | 400 | 400 | 400 |
| **Min.Equity** | −445.02833 | −481.23355 | −484.58106 | −563.40709 | −690.55890 |
| **End.Equity** | −290.5900000 | −371.8947518 | −436.2552837 | −563.4070922 | −690.5589007 |

# Stoploss analysis



COMPUTATIONAL FINANCE & RISK MANAGEMENT

UNIVERSITY *of* WASHINGTON

Department of Applied Mathematics

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