Statistical Arbitrage Trading Strategy

Backtesting Using KDB+/Q

Kim Kuen Tang

19. April 2012

Abstract

This talk is about the pairs trading strategy under the framework of stochastic control approach. We formulate the problem as the maximum of an expected terminal wealth under an utility function for a given risk parameter. This value function is solved using a separation ansatz which will help us to obtain a closed-form solution of the optimal positions. This strategy is then backtested completely in KDB+/Q. Numerical results are shown for the pnl, maximum drawdown and sharpe ratio.

Outline

What is a trading strategy?

Mathematician William F. Donaghue: “Thorp, my advice is to buy low sell high. “

Three steps for a successful alpha model:

step	**skill**
idea	visionary
development and backtesting	quantitative and programming
successful real world implementation	entrepreneurial

This talk is about the idea, development and backtesting.

Orderbook and Trade

- An order book is the list of orders a trading venue (in particular stock exchanges) uses to record the interest of buyers and sellers.
- The trade data contains the last sale, last size and last volume of the trade.
- Use metrics of the orderbooks and trades to predict the outcomes of the returns from midprices or trades.

We are interested in the following equation

$$h_k(r_{k+1}) = f_k(\bar{o}_k, \bar{t}_k)$$

Here r_{k+1} denotes the returns, $(\bar{o}_k, \bar{t}_k)$ contains all the information of orderbooks and trades until k.

At least two ways to choose h_k and f_k

- Regression
  $r_{k+1} = \beta^{\mathrm{T}} \cdot f(\bar{o}_k, \bar{t}_k) + \epsilon_{k+1}$
- Support Vector Machine
  $sign\,(r_{k+1}) = sign\left(\beta^{\mathrm{T}} \cdot f(\bar{o}_k, \bar{t}_k) + \beta_0\right)$

This talk is not about these models. For completeness we just mentioned it here but we will not show any results.

Pairs Trading

- Identify a pair of stocks where the log prices is cointegrated
- Model the spread of the log prices using an Ornstein-Uhlenbeck process.
- If observations are larger (smaller) than the predicted value we take a long (short) position and unwind it when the spread reverts.

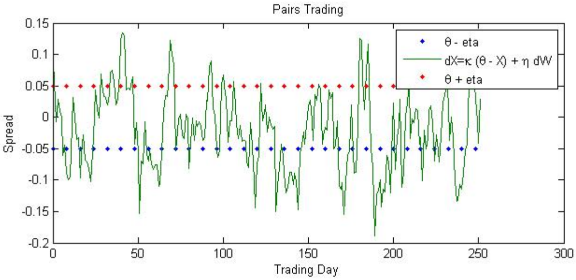


Figure: Example path for the Ornstein-Ulenbeck process with

Model Specification

With $M = M_t, B = B_t, A = A_t, X = X_t, W = W_t$ and $Z = Z_t$ we describe the asset, spread and wealth dynamics by the following formulas:

- $\mathrm{d}B/B = \mu \mathrm{d}t + \sigma \mathrm{d}Z$ (price dynamic)
- $X = \ln A - \ln B$ (spread definition)
- $\mathrm{d}X = \kappa(\theta - X)\,\mathrm{d}t + \eta \mathrm{d}W$ (spread dynamic)
- $\mathrm{d}Z\mathrm{d}W = \rho \mathrm{d}t$ (correlation)
- $\mathrm{d}M = rM\mathrm{d}t$ (risk-free asset)

The investment behaviour is described by a process h which is progressively measureable. It models the portfolio weight for stock A and $-h$ for stock B. If V denotes the wealth dynamic of our self-financing portfolio then the dynamic is given by

$$\mathrm{d}V/V = h\mathrm{d}A/A - h\mathrm{d}B/B + r\mathrm{d}t$$

Observe that

$$A = B\exp(X) = f(B, X)$$

Using Ito's formula:

$$\mathrm{d}f = \dot{f}\,\mathrm{d}t + (\nabla' f)(\mathrm{d}B, \mathrm{d}X) + 0.5(\mathrm{d}B, \mathrm{d}X)(\nabla^2 f)(\mathrm{d}B, \mathrm{d}X)$$

with

$$\dot{f} = 0; f_1 = A/B; f_2 = A; f_{11} = 0; f_{12} = A/B; f_{22} = A$$

we conclude that

$$\begin{aligned} \mathrm{d}A/A - \mathrm{d}B/B =& \mathrm{d}X + \mathrm{d}B/B \cdot \mathrm{d}X + (\mathrm{d}X)^2 \\ =& \left(\kappa\left(\theta - X\right) + \eta^2/2 + \rho\sigma\eta\right)\mathrm{d}t + \eta\mathrm{d}W \end{aligned}$$

Problem Formulation

Now we seek to find the optimal position h where it is able to solve the expression below

$$\sup_{h \in \alpha(t,T)} E^{t,v,x}[U(V(T))] \text{ for an utility function } U \tag{1}$$

s. t.

$$\begin{aligned} U(x) &= x^{\gamma}/\gamma \text{ for a } \gamma \leq -1 \\ \mathrm{d}V/V &= h\mathrm{d}A/A - h\mathrm{d}B/B + r\mathrm{d}t \end{aligned}$$

Here $\alpha(t, T)$ contains all progressively and admissible processes. If we denote with $G = G(t, v, x)$ the expression in (1) then it follows with the Bellman-Principle that

$$G(t, v, x) = \sup_{h \in \alpha(t,s)} E^{t,v,x}[G(s, V(s), X(s))]$$

The Ito-integral gives the following expression

$$0 = \quad \sup_{h\in\alpha(t,s)} E^{t,v,x}\left[\int_t^s \mathrm{d}G\right]$$

Divide the right hand side by $s-t$ and let s go to t to get

$$0 = \sup_{h\in\alpha(t,T)} \mathrm{drift}(\mathrm{d}G) = \sup_h h^2 \cdot \{*\} + h \cdot \{*\} + * \qquad (2)$$

The first order condition implies

$$\begin{aligned} 0 = & \quad 2 \cdot \hat{h} \cdot \{*\} + \{*\} \\ = & \quad \hat{h}\eta^2 v \cdot G_{vv} + \eta^2 \cdot G_{vx} + (-\kappa(x-\theta) + 0.5 \cdot \eta^2 \rho\sigma\eta) \cdot G_v \end{aligned}$$

Assume that $G_{vv} < 0$, it yields

$$\hat{h} = -(\eta^2 \cdot G_{v,x} + (-\kappa(x-\theta) + 0.5 \cdot \eta^2 \rho\sigma\eta) \cdot G_v)/(\eta^2 v \cdot G_{v,v}) \qquad (3)$$

Now plug back the above expression into (2) we get a differential equation which depends only on the value function G.

$$\begin{aligned} 0 =& \eta^2 G_t G_{v,v} - 0.5\eta^4 G_{v,x}^2 - 0.5b^2 G_v^2 - b\eta^2 G_v G_{v,x} \\ &+ 0.5\eta^4 G_{v,v} G_{x,x} + r\eta^2 v G_v G_{v,v} - \kappa(x-\theta)\eta^2 G_x G_{v,v} \\ =& \mathfrak{L}G \end{aligned} \tag{4}$$

Here $b = -\kappa(x-\theta) + 0.5\eta^2 + \rho\sigma\eta$.
To obtain a closed form solution we will consider the following separation ansatz.

$$G(t, v, x) = f(t, x)v^\gamma$$

Here f denotes

$$f(t, x) = g(t)\exp(x\beta(t) + x^2\alpha(t))$$

Applying the Operator $\mathfrak{L}$ on the expression $g(t)\exp(x\beta(t) + x^2\alpha(t))v^\gamma$ yields

$$0 = x^2\{*\} + x\{*\} + \{*\}$$

Setting the three coefficients to zero yields the following three differential equations

$$\begin{aligned} \dot{\alpha} &= \{*\}\alpha^2 + \{*\}\alpha + \{*\} \\ 0 &= \mathfrak{B}(\alpha, \beta) \\ 0 &= \mathfrak{G}(\alpha, \beta, g) \end{aligned}$$

The first equation is a Riccati equation and $\mathfrak{B}$ and $\mathfrak{G}$ are first order linear ordinary differential equations.

With $c = \sqrt{1-\gamma}$ and $d = \exp(2\kappa(T-t)/c)$ we conclude that

$$\alpha(t) = \kappa(1-c)/(2\gamma^2) \cdot \{1 + (2c)/(1-c-(1+c)d\}$$

$$\begin{aligned}\beta(t) = \quad & 1/(2\eta^2(1-c-(1+c)d))\gamma(1-d) \cdot \\ & \{c(\eta^2 + 2\rho\sigma\eta)(1-d) - (\eta^2 + 2\rho\sigma\eta + 2\kappa\theta)\}\end{aligned}$$

$$g(t) = exp\left(-\int_t^T u(s)ds/(c^2\eta^2)\right)$$

where u satisfies a differential equation which is not of interest here.

We need to show that the expression $G = g \cdot \exp(x^2\alpha + x\beta)v^\gamma$ really satisfies the equation (3) and $G_{vv} < 0$. But this will not be shown here. Using (4) we get a closed form solution for the optimal position

$$\hat{h}(t,x) = (\beta(t) + 2x\alpha(t) - \kappa(x-\theta)/\eta^2 + \rho\sigma/\eta + 0.5)/(1-\gamma) \tag{5}$$

References can be found here.

S. Mudchanatongsuk, J. Primbs, and W. Wong. Optimal pairs trading: A stochastic control approach. Proceedings of the American Control Conference, pages 1035-1039, 2008.

In the next part we will show the backtesting implementation using KDB+/Q.

Introduction

We aim to backtest this strategy with 10000 paths. Each path will have $5 * 252$ points. Instead doing simulate-and-forget we will store all the paths in the memory for further path-dependent risk numbers calculation. For this approach we need a system that is able to store a lot of data in the memory and that is also very performant. We will use KDB+/Q.

What is KDB+/Q ?

- KDB+ is an in-memory, column-based database and Q is the query language developed by Arthur Whitney and commercialized by kx.
- Q is a terse variant of a programming language, short apl.
- The evaluation is from right to left.

We will use KDB+/Q to

- sample normal distributed random numbers.
- generate multivariate stochastic differential equations.
- calculate risk numbers like maximum drawdown, sharpe ratio and the terminal pnl.

Normal distributed random variables using Box Mueller Method

To sample normal distributed random numbers we will use the method from Box-Mueller.
Let u_1, u_2 be independentrandom variables uniform on $(0, 1)$ then the random variables

$$\sqrt{-2 \ln u_1} \cdot (\cos, \sin)(2\pi u_2)$$

will be normal distributed with mean zero and unit variance.
Code in Q.

```
tmp:`int$num*step-1
N:sqrt[-2*log tmp?1f]*/:(cos;sin)@\:2*pi*tmp?1f
`Z`W set 'num cut/:L mmu N
```

```
tmp:`int$num*step-1
```

How does it work?

`num*step-1`	multiply num with step-1
`` `int$a ``	convert a into an int
`tmp:a`	assign the value a to tmp

Putting everything together.

$$\underbrace{\mathrm{tmp} : \underbrace{`\mathrm{int}\$\mathrm{num} * step - 1}_{\text{convert right side to int}}}_{\text{assign right side to tmp}}$$

```
N: sqrt[-2*log tmp?1f]*/:(cos;sin)@\:2*pi*tmp?1f
```

How does it work?

`tmp?1f`	sample tmp-times uniform distributed random numbers between zero and one
`(cos;sin) @\: l`	apply the list l to each function on the left
`l1 */: l2`	multiply list l1 to each element on the right

Putting everything together.

$$\underbrace{\underbrace{\text{sqrt[-2} * \text{log tmp?1.0]}}_{\text{list of n floats}} */: \underbrace{(\cos;\sin)@\backslash: \underbrace{2 * \text{pi} * \underbrace{\text{tmp?1.0}}_{\text{n random floats between 0 and 1}}}_{\text{list of n floats}}}_{\text{apply each element on left to right}}}_{\text{element-wise multiply each element on the right to the left}}$$

```
`Z`W set 'num cut/:L mmu N
```

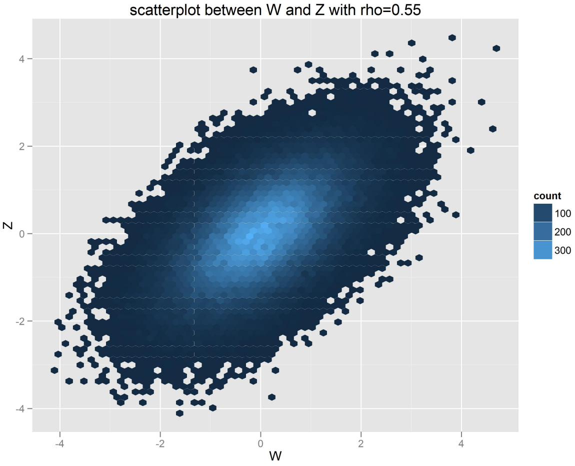
How does it work?

`L mmu N`	matrix multiplication to create correlated random numbers
`num cut/:l`	splits each element on the right into num-sized parts
`` `Z`W set' (l1;l2) ``	assign item-wise each element on the right to each element on the left

Putting eveything together.

$$\underbrace{\text{`Z`W set'} \quad \underbrace{\text{num cut/:} \quad \underbrace{\text{L mmu N}}_{\text{matrix multiplication}}}_{\text{splits each element on the right into num-sized parts}}}_{\text{assign item-wise each element}}$$

Visualization of the normal random numbers



Price and Spread Process

The s.d.e. for the asset is

$$\begin{aligned} \mathrm{d}(\log S_t) &= \mathrm{d}S_t/S_t - 0.5(\mathrm{d}S_t/S_t)^2 \\ &= (\mu - 0.5\sigma^2)\mathrm{d}t + \sigma \mathrm{d}W_t \end{aligned}$$

We solve the differential equation and take equidistant time intervalls Δ for the simulation.

$$\begin{aligned} S_{n+1} &= S_n \exp((\mu - 0.5\sigma^2)\Delta + \sigma\sqrt{\Delta}\epsilon_{n+1}) \\ &= f(S_n, \epsilon_{n+1}) \end{aligned}$$

Here ϵ_n are normal distributed random variables.

Code in Q:

```
sdelta:sqrt delta:matur%steps
f:{x*exp(delta*mu-0.5*vola*vola)+vola*sdelta*y}
A:flip f scan enlist[spot],Z
```

How does it work?

`f:{}`	f is a dyadic function with x and y as its arguments
`enlist[spot]`	put the spot into a list
`l1 , l2`	join the two lists to form a general list
`flip m`	transpose the matrix m
`f scan l`	create a new list by applying the function f to successive items of the list argument l

Putting everything together.

$$\underbrace{\text{flip}\qquad \underbrace{\text{f scan}\quad \underbrace{\underbrace{\mathit{enlist}[\mathit{spot}]}_{\text{put the spot into a list}}\quad ,\text{Z}}_{\text{join the lists together}}}_{\text{apply the function successive to the element of the list}}}_{\text{transpose the matrix to get a vector of path}}$$

The calculation of the spread is similar.

```
ekh:exp neg kappa*sdelta
skh:sqrt eta*eta%2f*kappa*1f-ekh*ekh
g:{(x*ekh)+(theta*1f-ekh)+y*skh}
spr:neg A-B:A*exp lspr:flip g scan enlist[theta],W
```

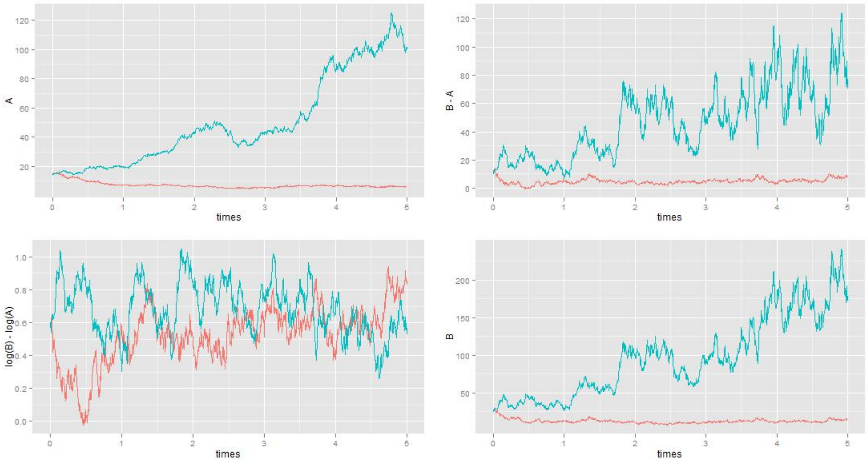


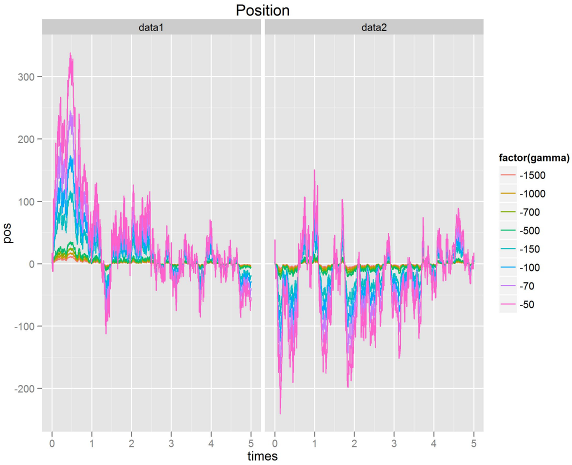
Figure: two example paths

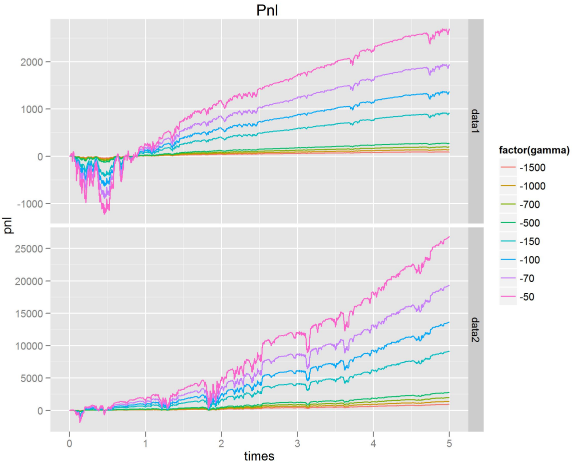
The function h calculates the optimal position. Code in Q:

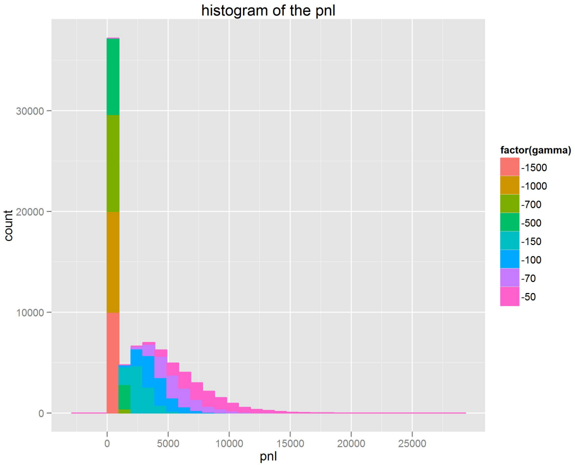
```
dpos:flip deltas flip pos:h[times;lspr]
scfs:flip sums flip cfs:neg spr*dpos
pnl:(tpnl:spr*pos)+scfs
ret:pnl%abs scfs
sh:sqrt[252]*(%) . stats:(avg;dev)@\:flip ret
mdd:{max neg x-maxs x}
mdds:mdd each pnl
```

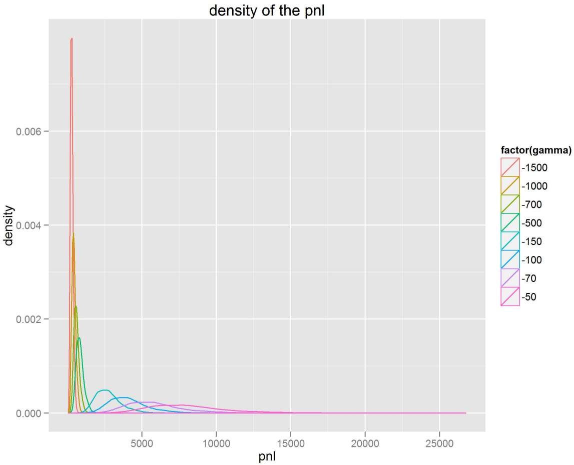
How does it work?

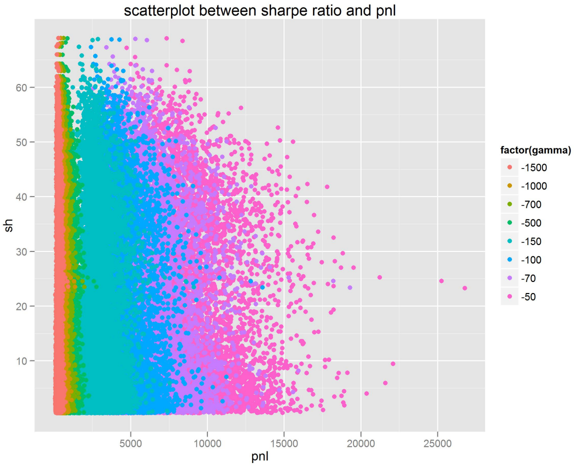
`pos:h[times;lspr]`	optimal position
`dpos:flip deltas flip pos`	calculates the daily transaction
`cfs:neg spr*dpos`	calculates the daily cashflows
`scfs:flip sums flip cfs`	calculates the total cashflows
`pnl:(tpnl:spr*pos)+scfs`	calculation of the pnl
`ret:pnl%abs scfs`	rate of return
`stats:(avg;dev)@\:flip ret`	mean and standard deviation
`sh:sqrt[252]*( % ) . stats`	annualized sharpe ratio
`{max neg x-maxs x} each pnl`	maximum drawdown

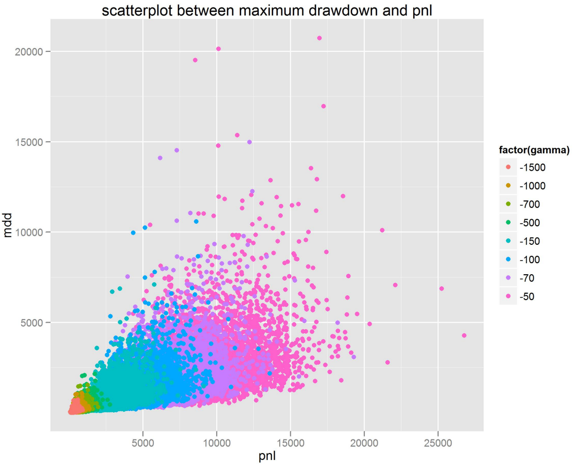

Position
data1
data2
300
200
100
0
-100
-200
pos
0
1
2
3
4
5
times
factor(gamma)
-1500
-1000
-700
-500
-150
-100
-70
-50

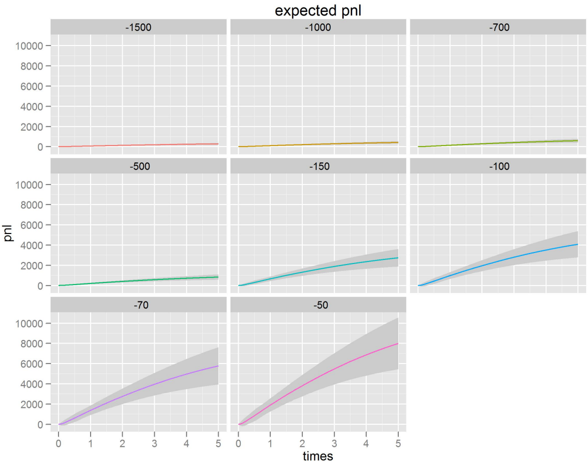

Pnl
pnl
2000
1000
0
-1000
25000
20000
15000
10000
5000
0
0
1
2
3
4
5
times
data1
data2
factor(gamma)
-1500
-1000
-700
-500
-150
-100
-70
-50

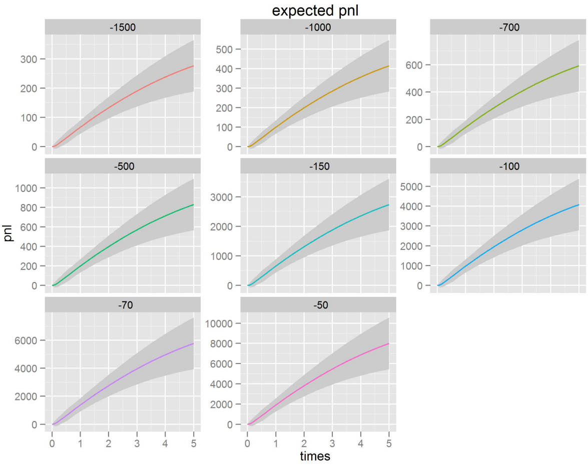

histogram of the pnl
count
30000
20000
10000
0
0
5000
10000
15000
20000
25000
pnl
factor(gamma)
-1500
-1000
-700
-500
-150
-100
-70
-50


density of the pnl
density
0.000
0.002
0.004
0.006
pnl
5000
10000
15000
20000
25000
factor(gamma)
-1500
-1000
-700
-500
-150
-100
-70
-50


scatterplot between sharpe ratio and pnl
sh
60
50
40
30
20
10
pnl
5000
10000
15000
20000
25000
factor(gamma)
-1500
-1000
-700
-500
-150
-100
-70
-50


scatterplot between maximum drawdown and pnl
mdd
20000
15000
10000
5000
pnl
5000
10000
15000
20000
25000
factor(gamma)
-1500
-1000
-700
-500
-150
-100
-70
-50


expected pnl
-1500
-1000
-700
-500
-150
-100
-70
-50
pnl
times
0
2000
4000
6000
8000
10000
0 1 2 3 4 5


expected pnl
-1500
-1000
-700
-500
-150
-100
-70
-50
pnl
times
0
1
2
3
4
5

Questions?

mailto:kuentang@vodafone.de