

LondonR

18th September 2012

SPONSORED BY



MANGOSOLUTIONS

R TRAINING, CONSULTING & APPLICATION DEVELOPMENT





From Back-test to Trade using R Parallelization in OneTick

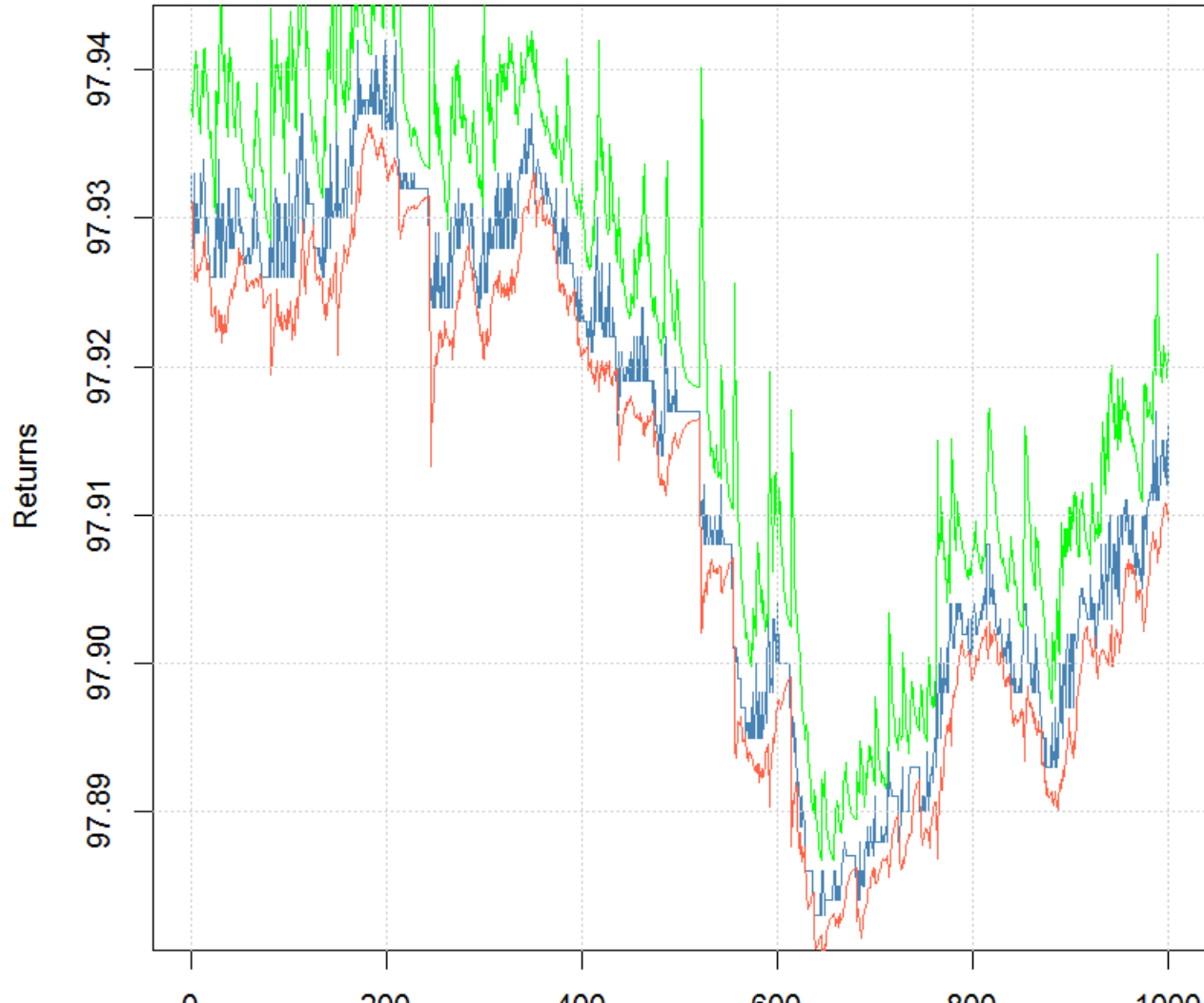
jjames@mango-solutions.com

andrew.diamond@onetick.com

Wt

- E
- T
- L
- A
- H
- S
- C
- Q

Series with with 2.5% VaR Limits



GARCH model: eGARCH

ONE MARKET DATA

||| ||| | |||
MANGOSOLUTIONS
data analysis that delivers

OneTick

Deal with idiosyncrasies of tick data

```
> setClass("TK", representation="POSIXct")
> TK <- function(x) new("TK", as.POSIXct(x, format="%Y%m%d %H:%M:%OS"))
> setOldClass("TK", S4Class="TK", where=.GlobalEnv)
>
> as.TK <- function(from) TK(as.POSIXct(from, format="%Y%m%d %H:%M:%OS", tz="GMT"))
> setAs("character", "TK", as.TK)
> setAs("TK", "POSIXct", function(from) as.POSIXct(as.numeric(from), origin="1970-01-01", tz="GMT"))
[1] "coerce<-"
> as.TK <- function(from) TK(as.POSIXct(from, format="%Y%m%d %H:%M:%OS", tz="GMT"))
> setAs("character", "TK", as.TK)
> setAs("TK", "POSIXct", function(from) as.POSIXct(as.numeric(from), origin="1970-01-01", tz="GMT"))
[1] "coerce<-"
> print.TK <- function(x, ...) print(format(as(x, "POSIXct"), format="%Y%m%d %H:%M:%OS3", ...))
> setGeneric("print.TK", print.TK)
[1] "print.TK"
> setMethod("show", signature(object="TK"), function(object) print(format(object, format="%Y%m%d %H:%M:%OS3")))
[1] "show"
```

perform (arbitrarily) garch

```
##### require(rugarch)
spec = ugarchspec()
## Assume that the points are distributed
fit = ugarchfit(data = diff(EURJPY[select,3]), spec =
spec)
plot(fit, which="all")

# return just to the rate data
fit = ugarchfit(data = EURJPY[select,3], spec = spec)
plot(fit, which="all")

forc = ugarchforecast(fit, n.ahead=20)
plot(forc, which="all")
```

Look for rolling values

```
ctrl = list(rho = 1, delta = 1e-9, outer.iter = 100, tol = 1e-7)
spec = ugarchspec(variance.model = list(model = "eGARCH"),
                  distribution.model = "jsu")
bktest <- ugarchroll(spec, data = EURJPY[select,3], n.ahead = 1,
#           parallel = TRUE, parallel.control = list(pkg =
c("snowfall"), cores = 4),
          forecast.length = 100, refit.every = 25, refit.window =
"recursive",
          solver = "solnp", fit.control = list(), solver.control =
ctrl,
          calculate.VaR = TRUE, VaR.alpha = c(0.01, 0.025, 0.05))
plot(bktest, which="all")
```

Look for rolling values

```
## BID=BID_PRICE,ASK=ASK_PRICE
len<- length(BID)
mu=NA
alpha1=NA
bktest <- NA
if(len>110){
tryCatch({bktest <- ugarchroll(spec, data = BID, n.ahead = 1,
  forecast.length = 100, refit.every = 25, refit.window = "recursive",
  parallel = TRUE,
  parallel.control = list(pkg = "snowfall", cores = 4),
  solver = "solnp", fit.control = list(), solver.control = ctrl,
  calculate.VaR = TRUE, VaR.alpha = c(0.01, 0.025, 0.05))
  mu=bktest@roll["coefs"][[1]][1,"mu"]
  alpha1=bktest@roll["coefs"][[1]][1,"alpha1"]},
error=function(e)cat("Error\n"))
}
## OneTick outputs
## MU=mu,ALPHA1=alpha1, LEN=len
```