COMPUTATIONAL FINANCE & RISK MANAGEMENT
UNIVERSITY of WASHINGTON
Department of Applied Mathematics

Backtesting Risk Models

Amath 546/Econ 589
Eric Zivot
Spring 2013
Updated: May 1, 2013

Backtesting Terminology: Rolling Forecasts

Example: 10 yr sample 1999-2009 (250 trading days per year)
T = 2500 days, W_E = 500 days, W_T = 2000 days
Rolling 1-step ahead forecasts

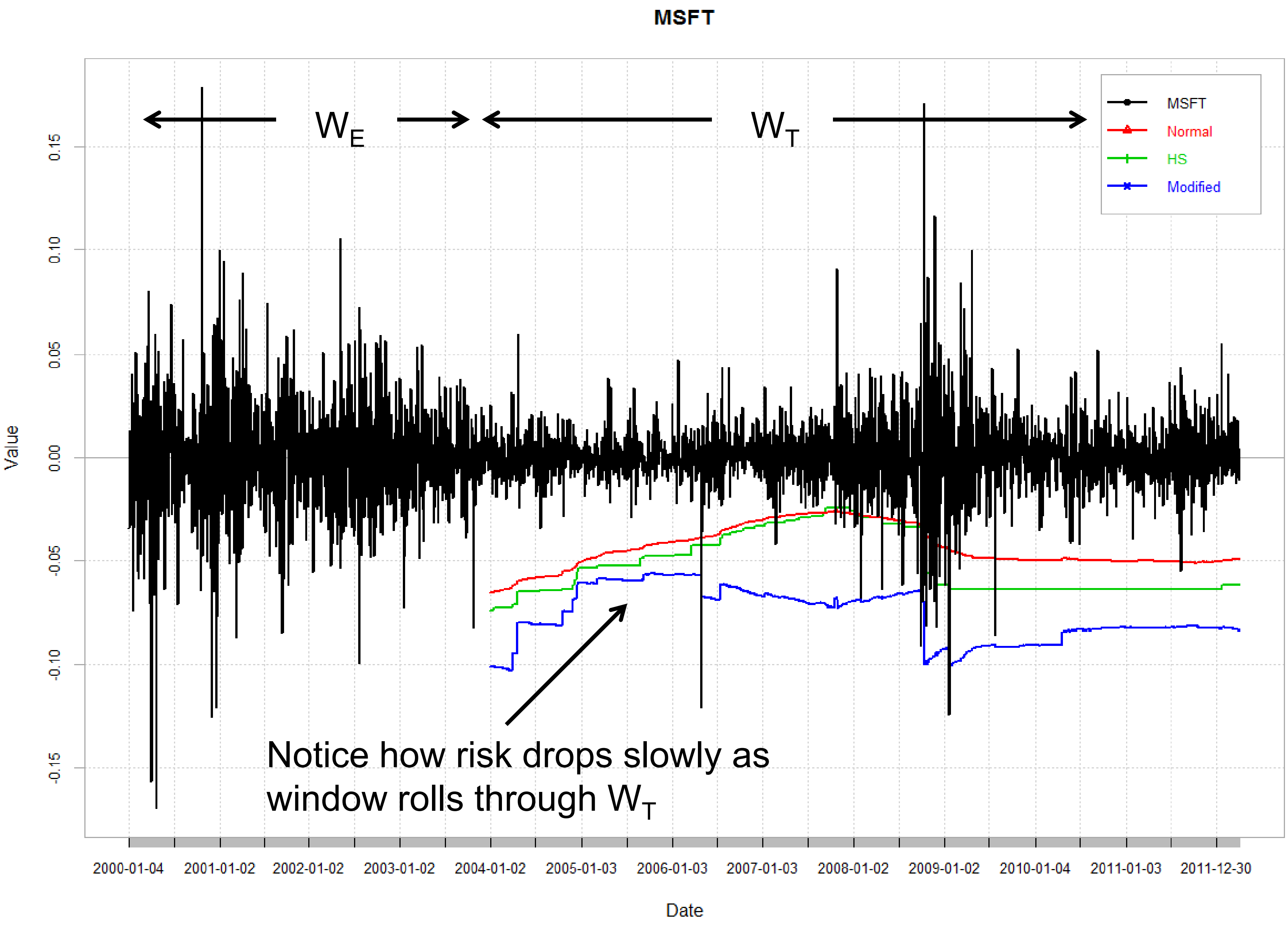
Start date	End date	VaR Forecast date
1999-01-01	2000-12-31	VaR(2001-01-01)
1999-01-02	2001-01-01	VaR(2001-01-02)
⋮	⋮	⋮
2007-12-31	2009-12-30	VaR(2009-12-31)

t	$t + W_E - 1$	$VaR(t+W_E)$
1	500	VaR(501)
2	501	VaR(502)
⋮	⋮	⋮
1999	2499	VaR(2500)

Backtesting Unconditional VaR Models

```
# set up estimation window and testing window
> n.obs = nrow(MSFT.ret)
> w.e = 1000          # estimation window
> w.t = n.obs - w.e   # test window
> alpha = 0.99

# loop over testing sample, compute VaR and record hit rates
backTestVaR <- function(x, p = 0.95) {
  normal.VaR = as.numeric(VaR(x, p=p, method="gaussian"))
  historical.VaR = as.numeric(VaR(x, p=p, method="historical"))
  modified.VaR = as.numeric(VaR(x, p=p, method="modified"))
  ans = c(normal.VaR, historical.VaR, modified.VaR)
  names(ans) = c("Normal", "HS", "Modified")
  return(ans)
}
> VaR.results = rollapply(as.zoo(MSFT.ret), width=w.e,
+                          FUN = backTestVaR, p=0.99,
+                          by.column = FALSE,
+                          align = "right")
> VaR.results = lag(VaR.results, k=-1)
> chart.TimeSeries(merge(MSFT.ret, VaR.results),
+                   legend.loc="topright")
```

Backtesting Unconditional VaR Models



Notice how risk drops slowly as window rolls through W_T

Backtesting Unconditional VaR

```
# Summarize VaR violations
> violations.mat = matrix(0, 3, 5)
> rownames(violations.mat) = c("Normal", "HS", "Modified")
> colnames(violations.mat) = c("En1", "n1", "1-alpha",
+                              "Percent“, “VR”)
> violations.mat[, "En1"] = (1-alpha)*w.t
> violations.mat[, "1-alpha"] = 1 - alpha

> for(i in colnames(VaR.results)) {
+   VaR.violations = as.zoo(MSFT.ret[index(VaR.results), ])
+                           < VaR.results[, i]
+   violations.mat[i, "n1"] = sum(VaR.violations)
+   violations.mat[i, "Percent"] = sum(VaR.violations)/w.t
+   violations.mat[i, "VR"]
+         = violations.mat[i,"n1"]/violations.mat[i, "En1"]
+ }
```

Summary of VaR Violations

```
> violations.mat
           En1 n1 1-alpha  Percent     VR
Normal   20.82 38    0.01 0.018252 1.8252
HS       20.82 32    0.01 0.015370 1.5370
Modified 20.82  4    0.01 0.001921 0.1921
```

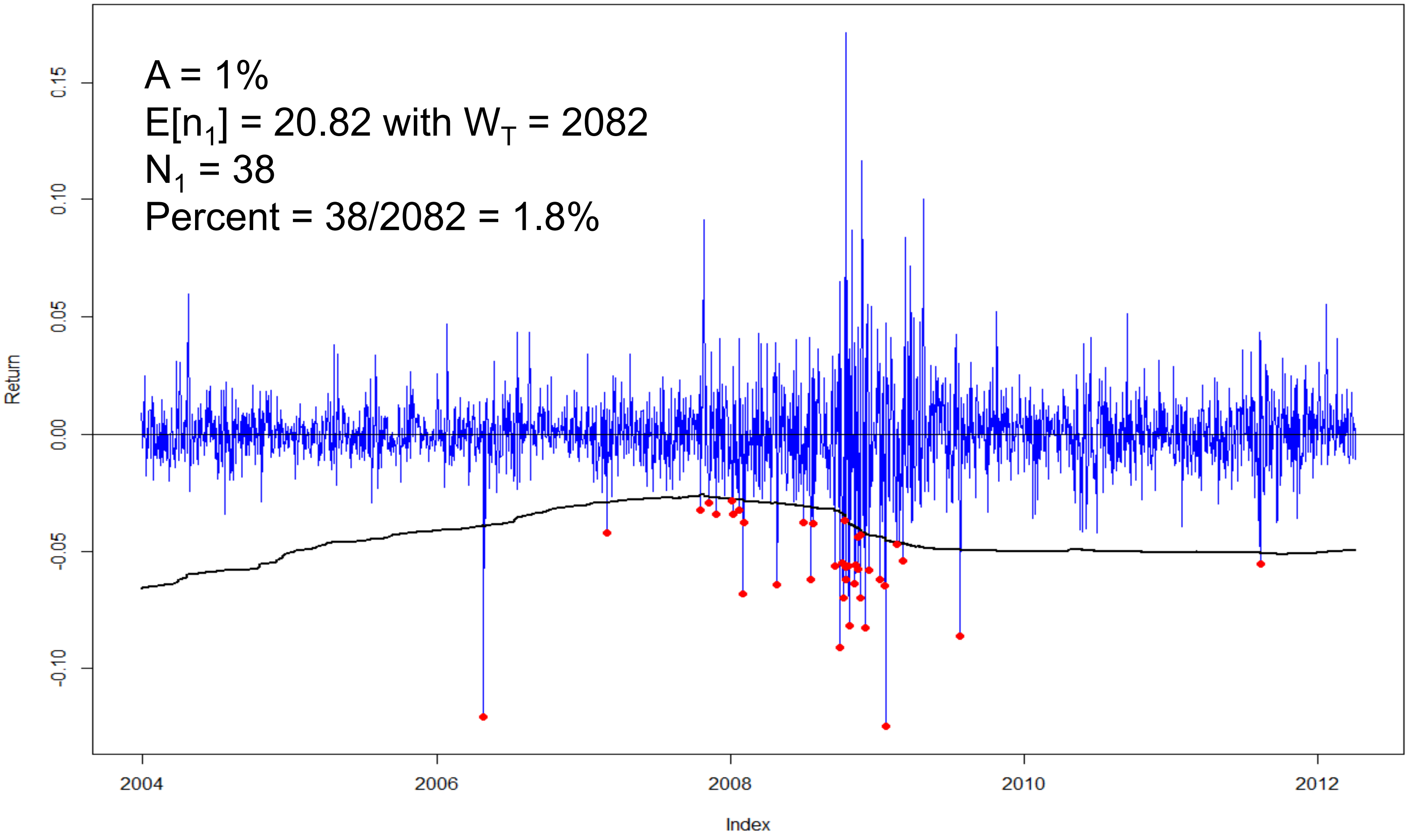
Results:

Normal VaR and HS underforecast risk

Modified (Cornish-Fisher) VaR overforecasts risk

Normal VaR Violations

Rolling VaR estimates W_E=1000



A = 1%

$E[n_1]$ = 20.82 with W_T = 2082

N_1 = 38

Percent = 38/2082 = 1.8%

© Eric Zivot 2008

Backtesting Unconditional VaR

```
# Use rugarch VaRTest() function
> VaR.test = VaRTest(1-alpha,
+                    actual=coredata(MSFT.ret[index(VaR.results),]),
+                    VaR=coredata(VaR.results[,"Normal"]))
> names(VaR.test)
[1]"expected.exceed" "actual.exceed” "uc.H0“ "uc.LRstat“ "uc.critical"
[6] "uc.LRp“ "uc.Decision“ "cc.H0” "cc.LRstat“ "cc.critical"
[11] "cc.LRp” "cc.Decision"
# LR test for correct # of exceedances (Kupiec Test)
> VaR.test[1:6]
$expected.exceed
[1] 20
$actual.exceed
[1] 38
$uc.H0
[1] "Correct Exceedances“
$uc.LRstat
[1] 11.51
$uc.critical
[1] 3.841
$uc.LRp
[1] 0.000692
$uc.Decision
[1] "Reject H0"
```

Reject H0: VaR model produces the correct number of exceedances at the 1% level

Backtesting Unconditional VaR

```
# LR tests for independence of exceedances
> VaR.test[8:12]
$cc.H0
[1] "Correct Exceedances & Independent"

$cc.LRstat
[1] 23.69

$cc.critical
[1] 5.991

$cc.LRp
[1] 7.17e-06

$cc.Decision
[1] "Reject H0"
```

Backtesting Conditional VaR: Rolling GARCH

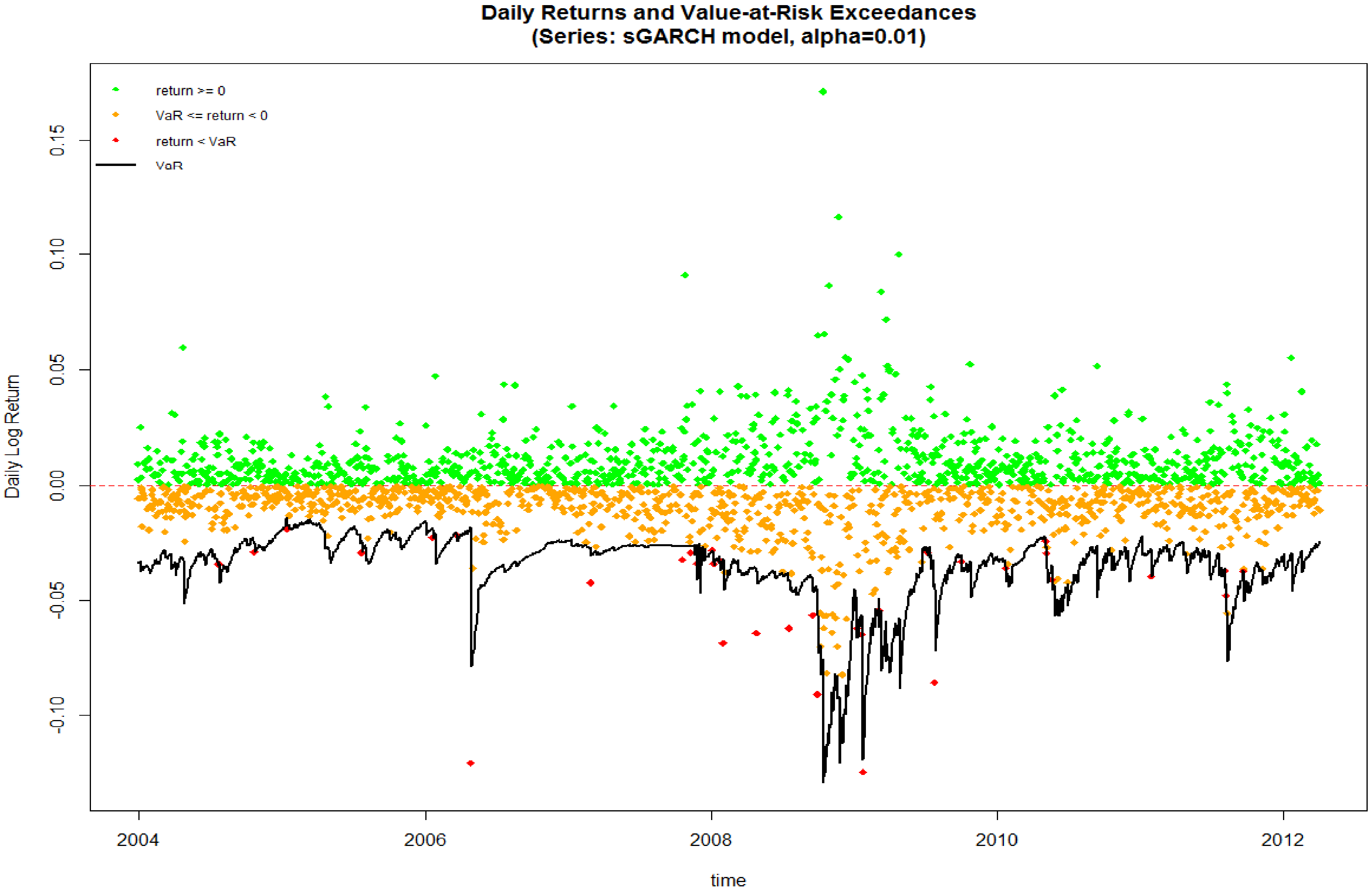
```
> MSFT.garch11.roll = ugarchroll(garch11.spec, MSFT.ret, n.ahead=1,
+                                forecast.length = w.t,
+                                refit.every=20, refit.window="moving",
+                                calculate.VaR=TRUE, VaR.alpha=0.01)
Warning messages:
1: In .makefitmodel(garchmodel = "sGARCH", f = .sgarchLLH, T = T, m = m,  :
  NaNs produced
2: In .makefitmodel(garchmodel = "sGARCH", f = .sgarchLLH, T = T, m = m,  :
  NaNs produced
3: In .makefitmodel(garchmodel = "sGARCH", f = .sgarchLLH, T = T, m = m,  :
  NaNs produced
4: In .makefitmodel(garchmodel = "sGARCH", f = .sgarchLLH, T = T, m = m,  :
  NaNs produced
5: 
solnp-->warning: Equal Lower/Upper Bounds Found. Consider

                                                   excluding fixed parameters.

6: In .makefitmodel(garchmodel = "sGARCH", f = .sgarchLLH, T = T, m = m,  :
  NaNs produced
```

Out-of-sample forecast
period

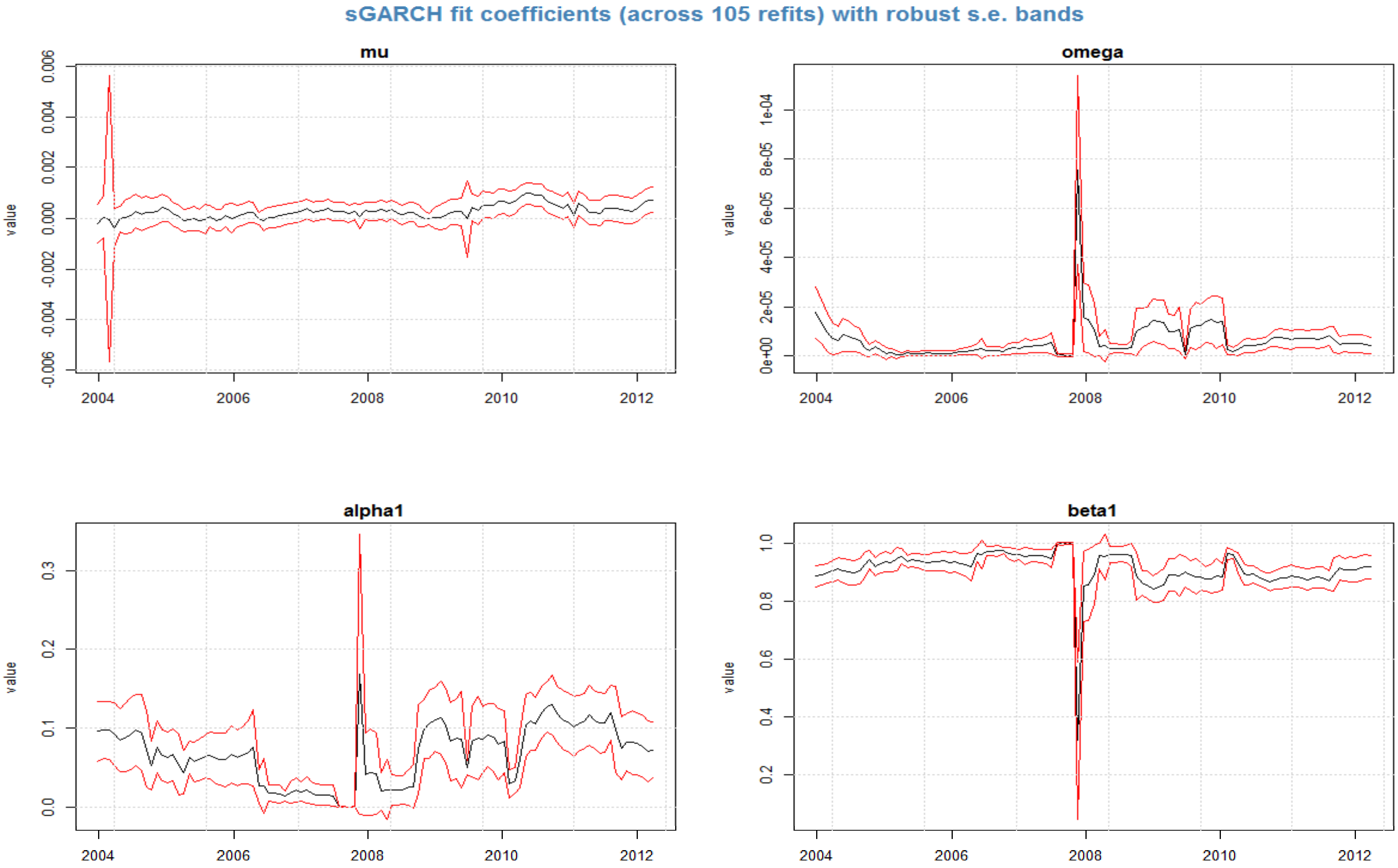
GARCH VaR Forecasts

```
> plot(MSFT.garch11.roll)

Make a plot selection (or 0 to exit):

1:   Density Forecast
2:   Sigma Forecast
3:   Series Forecast
4:   VaR Forecast
5:   Fit Coefficients (with s.e. bands)

> plot(MSFT.garch11.roll, which=4)
```

Normal GARCH VaR Forecasts



Rolling GARCH Coefficients



© Eric Zivot 2008

Evaluating GARCH VaR Forecasts

```
> report(MSFT.garch11.roll, type="VaR")
VaR Backtest Report
===========================================
Model:                            sGARCH-norm
Backtest Length: 2082
Data:

==========================================
alpha:                            1%
Expected Exceed:                  20.8
Actual VaR Exceed:                 35
Actual %:                         1.7%

Unconditional Coverage (Kupiec)
Null-Hypothesis: Correct Exceedances
LR.uc Statistic:           8.098
LR.uc Critical:           3.841
LR.uc p-value:            0.004
Reject Null:              YES

Conditional Coverage (Christoffersen)
Null-Hypothesis: Correct Exceedances and

                                           Independence of Failures

LR.cc Statistic:           9.296
LR.cc Critical:           5.991
LR.cc p-value:            0.01
Reject Null:              YES
```

GARCH(1,1) VaR estimates are not much better than unconditional estimates based on statistical tests