Estimating Volatilities and Correlations

Thomas Fillebeen

February 21, 2014

Abstract

Using historical CRSP data to produce estimates of current and future levels of volatilies and correlations. The vignette explores exponentially weighted moving average (EWMA), and generealized autoregressive conditional heteroscedasticity (GARCH(1,1)). These models recognize that volatilities and correlations are not constant. The models attempt to track variations in volatilities or correlation and forecast them into the future.

Contents

1 EWMA Covariance and Correlation

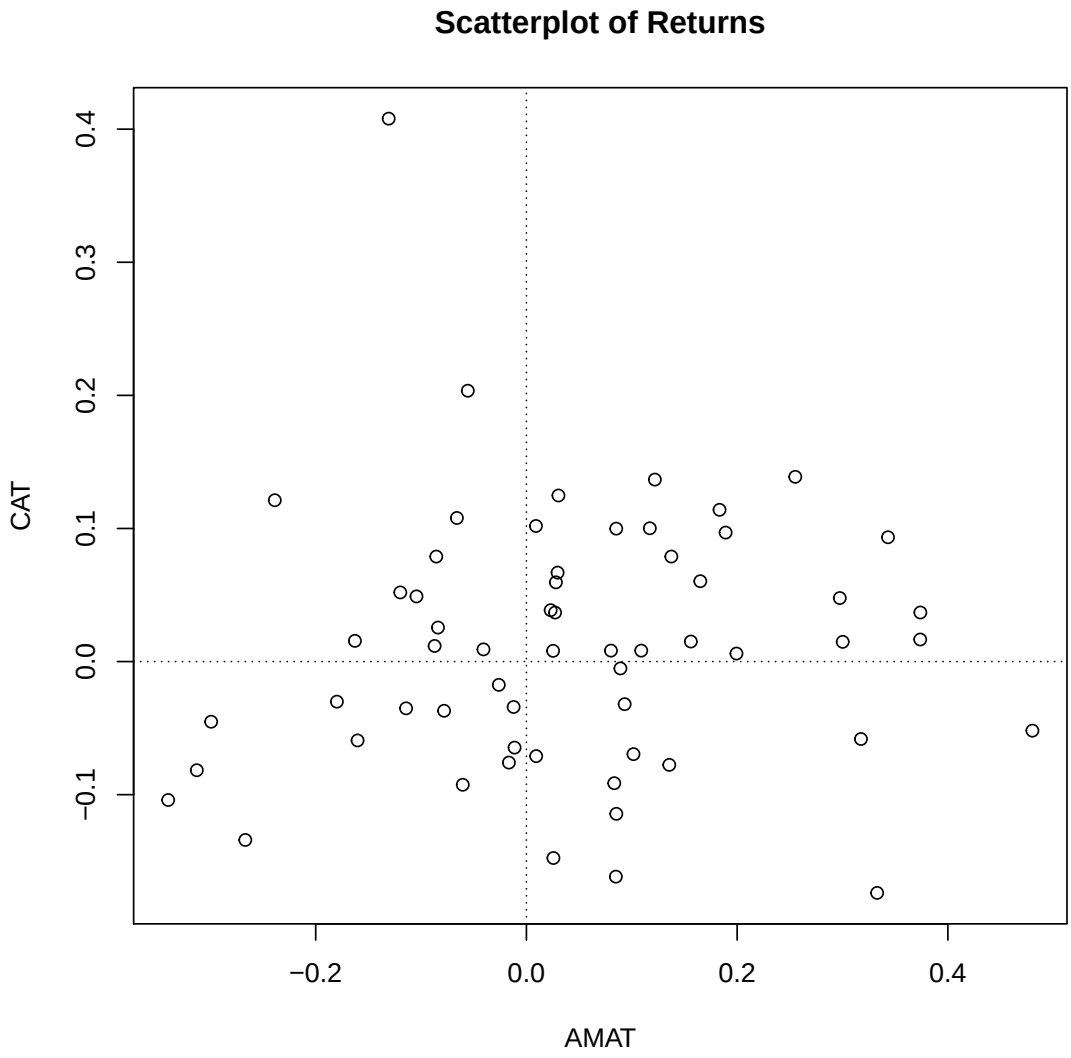
1.1 Selected Returns Time Series

```
# 'Load the GARPFRM package and CRSP dataset for analysis.
suppressMessages(library(GARPFRM))
suppressMessages(library(rugarch))
suppressMessages(library(rmgarch))
options(digits = 4)
```

```r
data(crsp.short)
R <- largecap.ts[, 1:4]
```

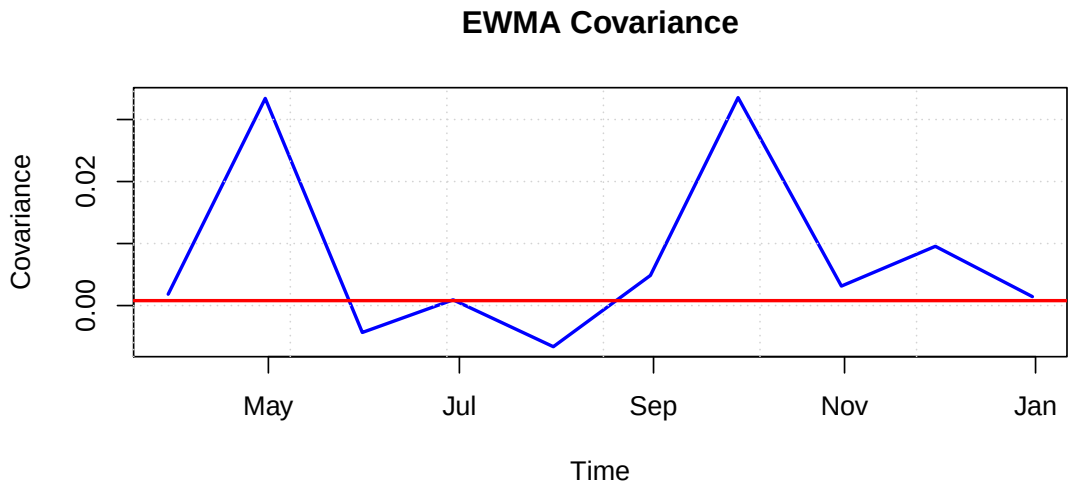
1.2 Estimating Correlation and Volatility

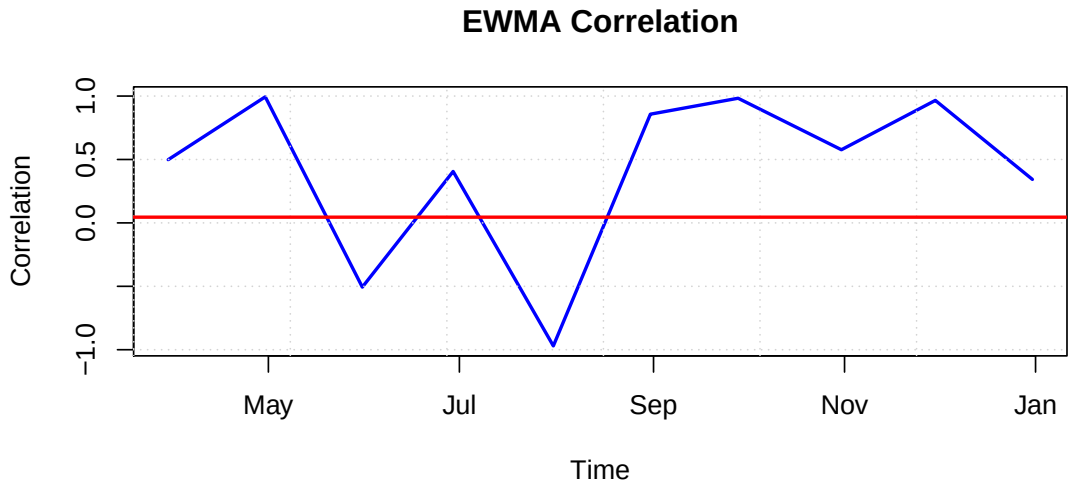
```r
# Remember: log-returns for GARCH analysis
asset1 = R[, 1]
asset2 = R[, 3]

# Create combined data series
cAssets = cbind(asset1, asset2)

# Scatterplot of returns
plot(coredata(asset1), coredata(asset2), xlab = colnames(asset1), ylab = colnames(asset2),
    main = "Scatterplot of Returns")
abline(h = 0, v = 0, lty = 3)
```



1.3 EWMA Correlation and Volatility

```
# Calculate EWMA cov and cor, applying default lambda - 0.94
cAssetsEWMACov <- EWMA(cAssets, lambda = 0.94, initialWindow = 50, cor = FALSE)
cAssetsEWMACor <- EWMA(cAssets, lambda = 0.94, initialWindow = 50, cor = TRUE)

# Plots
par(mfrow = c(2, 1))
plot(cAssetsEWMACov, asset1 = 1, asset2 = 2)
plot(cAssetsEWMACor, asset1 = 1, asset2 = 2)
```



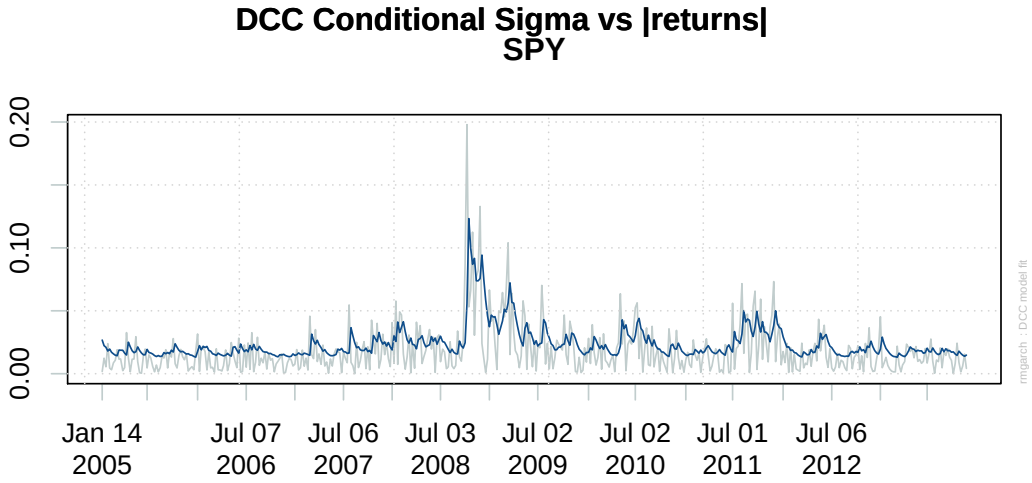


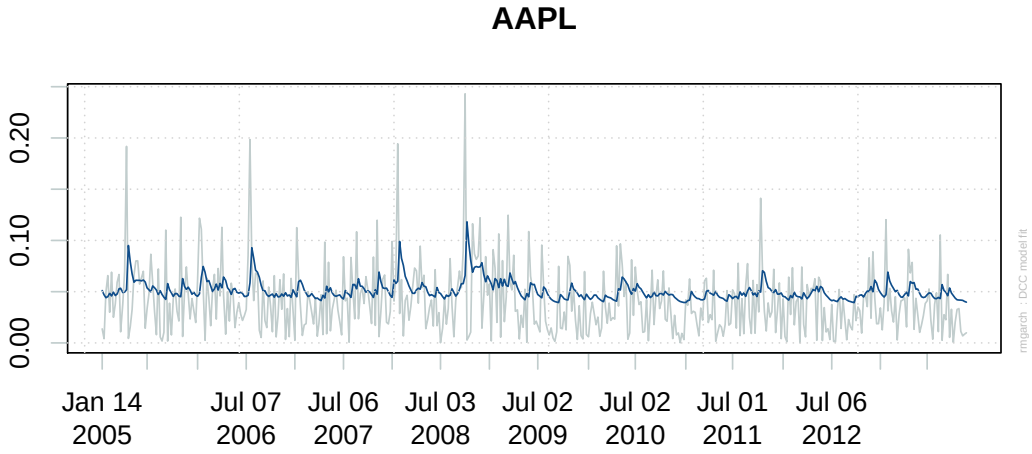

```
par(mfrow = c(1, 1))


# Compute EWMA cov and cor for longer half-life of
halfLife = log(0.5)/log(0.94) + 5
lambda = exp(log(0.5)/halfLife)
covEwma <- EWMA(cAssets, lambda)
```

2 GARCH(1,1) Conditional Correlation and Variance

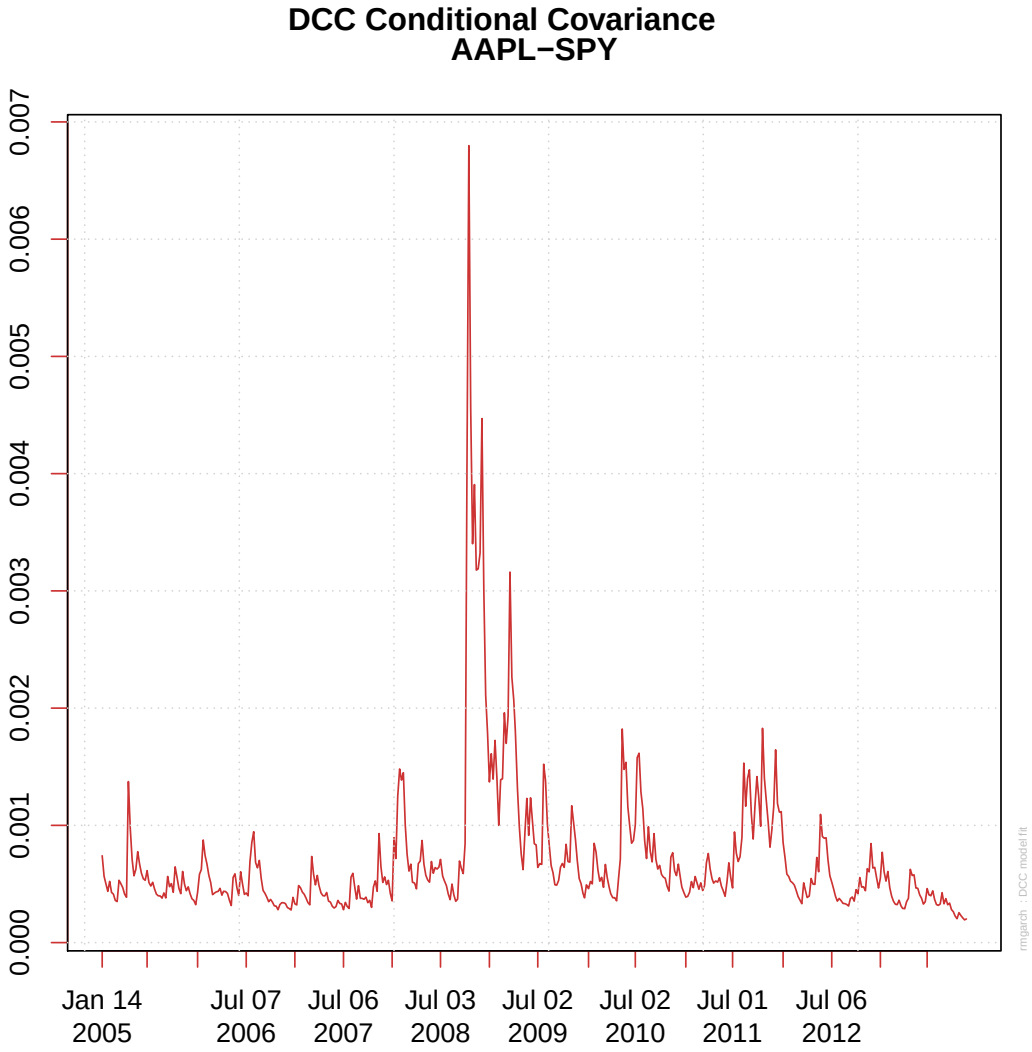
2.1 GARCH(1,1) Estimation and Plotting

```
# Garch11 testing: need 100 plus data points to forecast.  Load lengthier
# dataset.
data(returns)
cAssetsReturns = cbind(returns[, "SPY"], returns[, "AAPL"])
# Dynamic Conditional Cor/Cov
garch11 <- garch11(cAssetsReturns)

# garch11 is a dcc.fit object. There are multiple extraction functions to be
# exploited.  Illustrate garch11, dcc.fit object.

# Conditional Sigma (vs Realized Absolute Returns)
plot(garch11, which = 2)
```

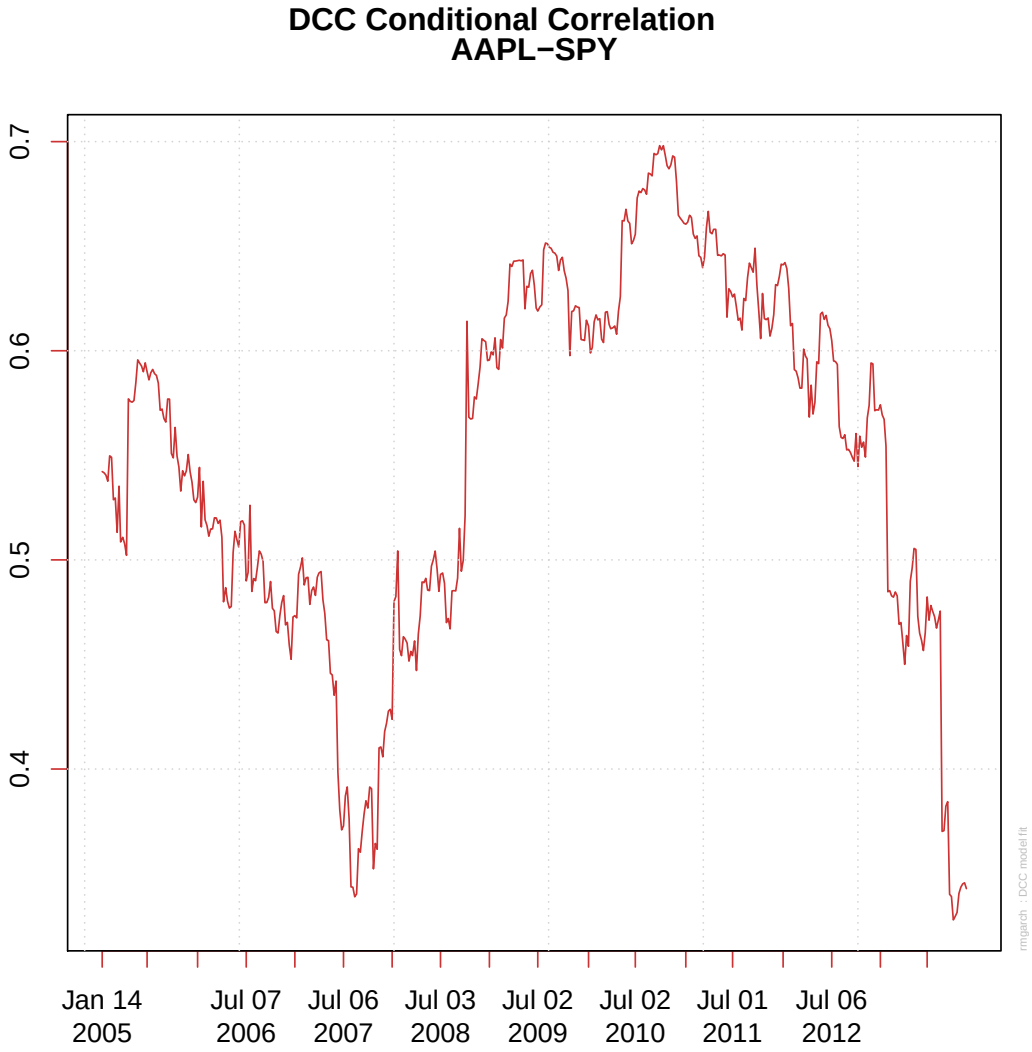





```
# Conditional covar of each series
plot(garch11, which = 3)
```




```
# Conditional cor
plot(garch11, which = 4)
```



2.2 Forecasting GARCH(1,1) Plotting

```
# Forecasting conditional vol and cor, default wd = 100
fcstGarch11 = fcstGarch11(garch11, 100)

# Many method functions - see help on DCCforecast class rshape, rskew,
# fitted, sigma, plot, rcor, rcov, show

# Show forecasts
fcstGarch11
```

```
##
## *----------------------------------*
## *       DCC GARCH Forecast         *
## *----------------------------------*
##
## Distribution         :  mvnorm
## Model                :  DCC(1,1)
## Horizon              :  100
## Roll Steps           :  0
## ------------------------------------
##
## 0-roll forecast:
##
## First 2 Correlation Forecasts
## , , 1
##
##        [,1]   [,2]
## [1,] 1.0000 0.3425
## [2,] 0.3425 1.0000
##
## , , 2
##
##        [,1]   [,2]
## [1,] 1.0000 0.3433
## [2,] 0.3433 1.0000
##
## . .
## . .
##
## Last 2 Correlation Forecasts
## , , 1
##
##        [,1]   [,2]
## [1,] 1.0000 0.4041
```

```
## [2,] 0.4041 1.0000
##
## , , 2
##
##         [,1]   [,2]
## [1,] 1.0000 0.4046
## [2,] 0.4046 1.0000

# Plot Conditional Covar Forecast
plot(fcstGarch11, which = 3)
```