



GARCH MODELS IN R

**Markets take the stairs up,
but the elevator down**

Kris Boudt

Professor of finance and econometrics

How?

Change the argument `distribution.model` of `ugarchspec()` from `"norm"` to `"sstd"`:

```
garchspec <- ugarchspec(  
  mean.model=list(armaOrder=c(0,0)),  
  variance.model=list(model="sGARCH"),  
  distribution.model = "norm")
```



```
garchspec <- ugarchspec(  
  mean.model=list(armaOrder=c(0,0)),  
  variance.model=list(model="sGARCH"),  
  distribution.model = "sstd")
```



The normal GARCH model

Under the model assumptions

$$\begin{aligned} R_t &= \mu_t + e_t \\ e_t &\sim N(0, \sigma_t^2) \end{aligned}$$

it follows that

$$\frac{R_t - \mu_t}{\sigma_t} \sim N(0, 1)$$



Let's test

- Caveat: The normality of the standardized returns follows from an assumption
- Let's compute the standardized returns and test whether the assumption is correct.



Estimated standardized returns

- Formula

$$Z_t = \frac{R_t - \hat{\mu}_t}{\hat{\sigma}_t}$$

- Calculation in R

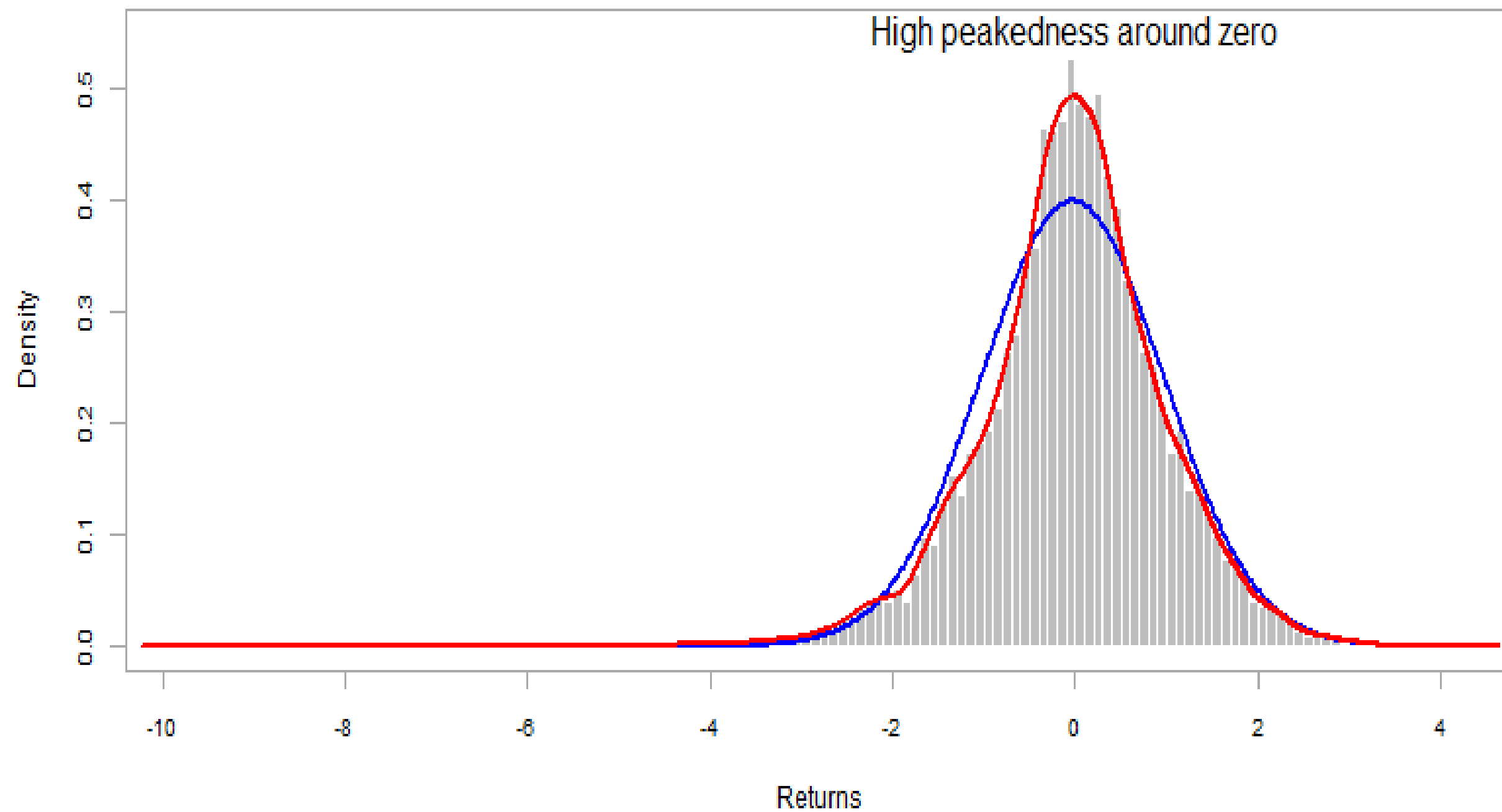
```
# obtain standardized returns  
stdret <- residuals(garchfit, standardize = TRUE)
```

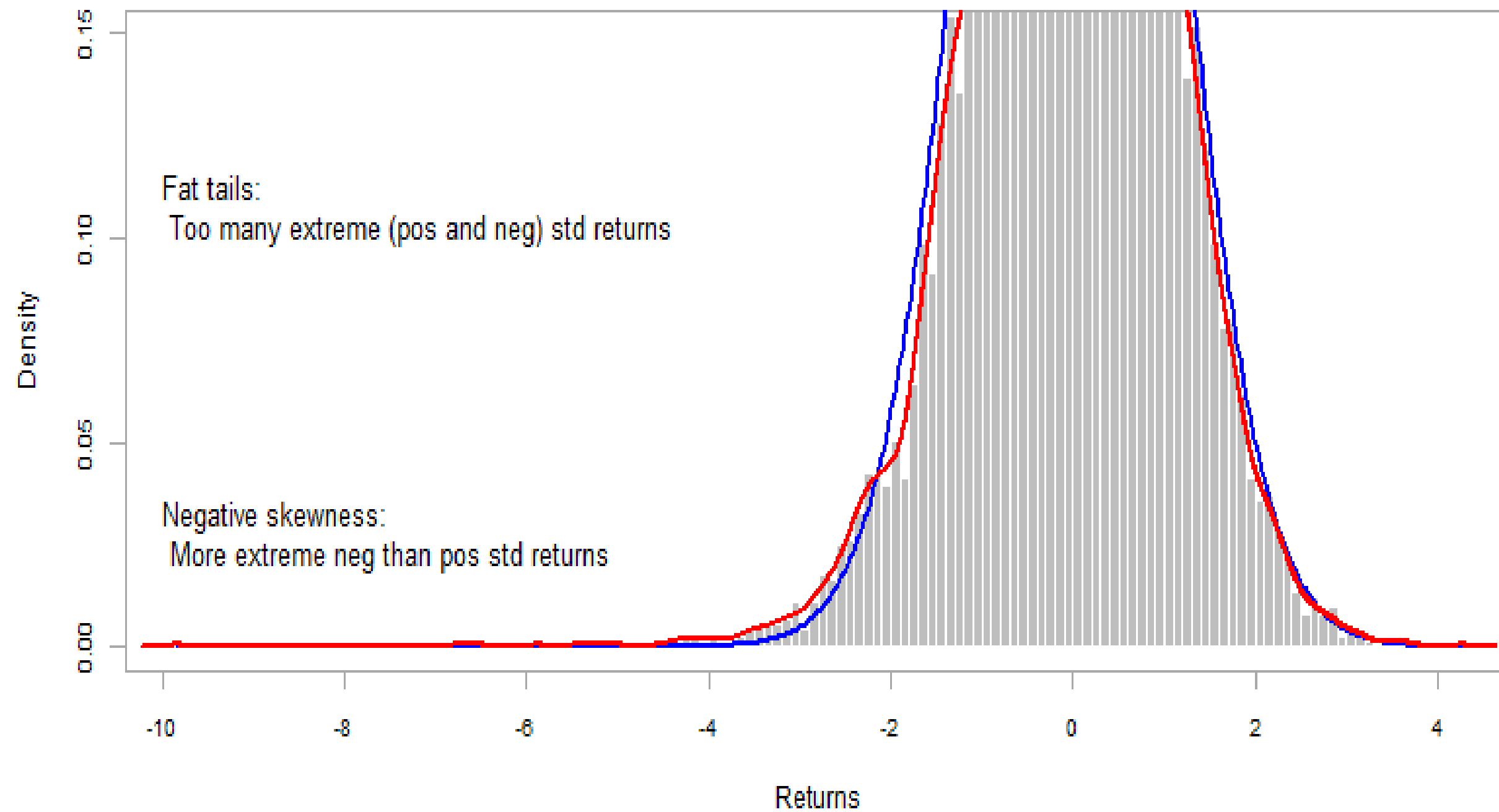


Testing the normality assumption

- Visual analysis

```
library(PerformanceAnalytics)
chart.Histogram(sp500ret, methods = c("add.normal", "add.density"),
               colorset=c("gray", "red", "blue"))
```







Solution

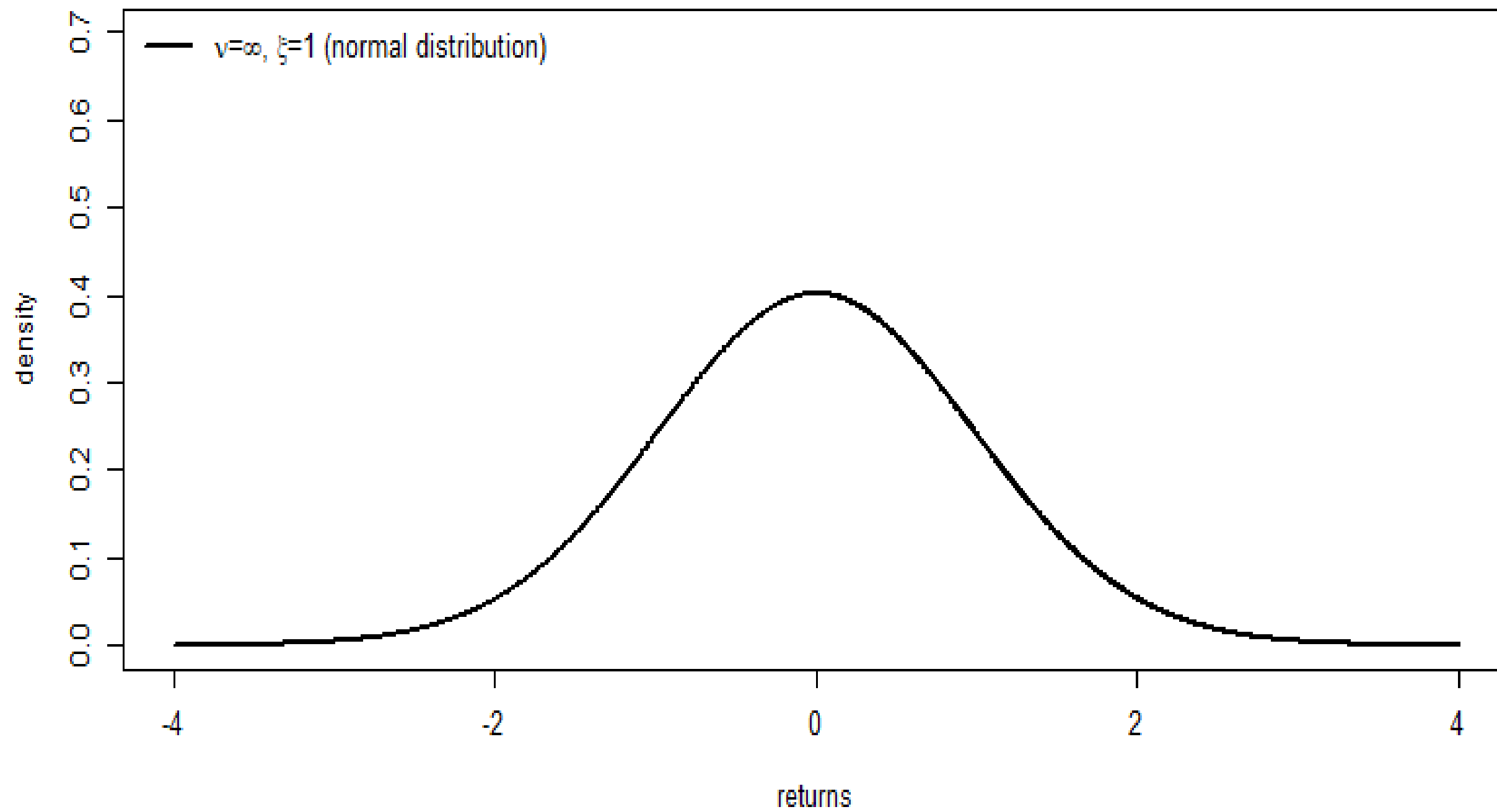
- A realistic distribution thus needs to accommodate the presence of
 - fat tails: higher probability to observe large (positive or negative) returns than under the normal distribution
 - skewness: asymmetry of the return distribution
- In `rugarch` this is possible with the **skewed student t** distribution:

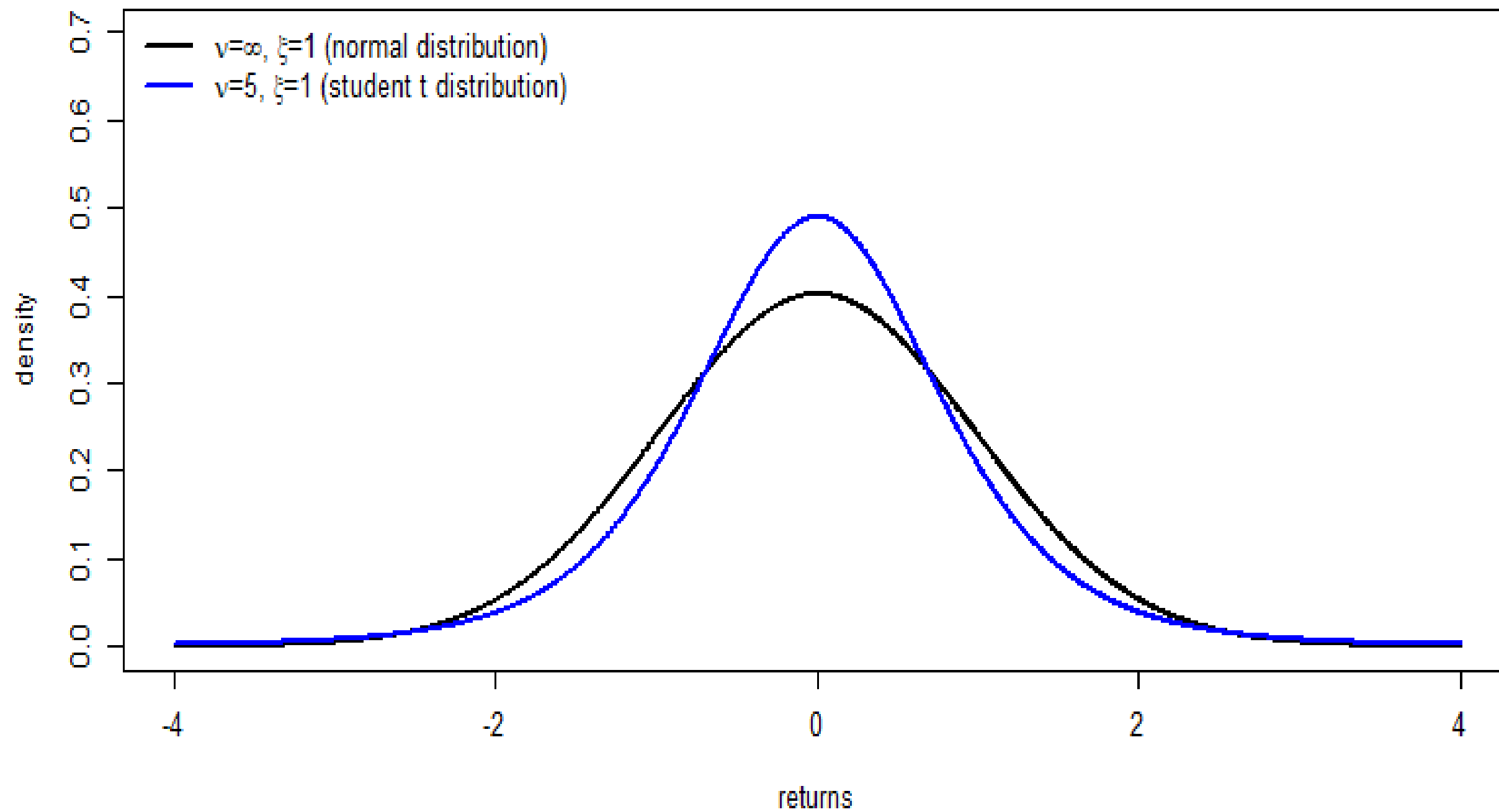
```
garchspec <- ugarchspec(distribution.model = "sstd")
```

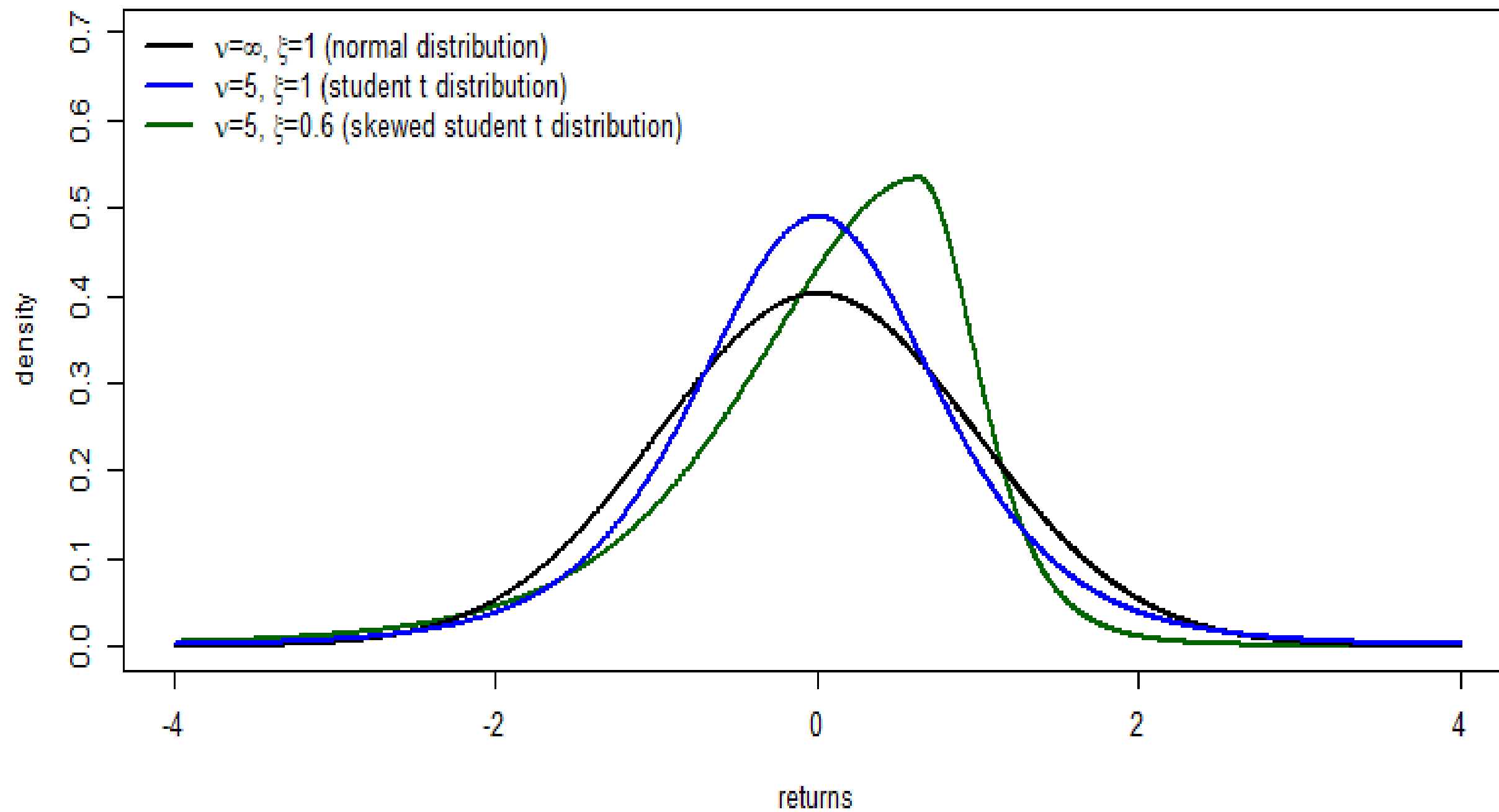


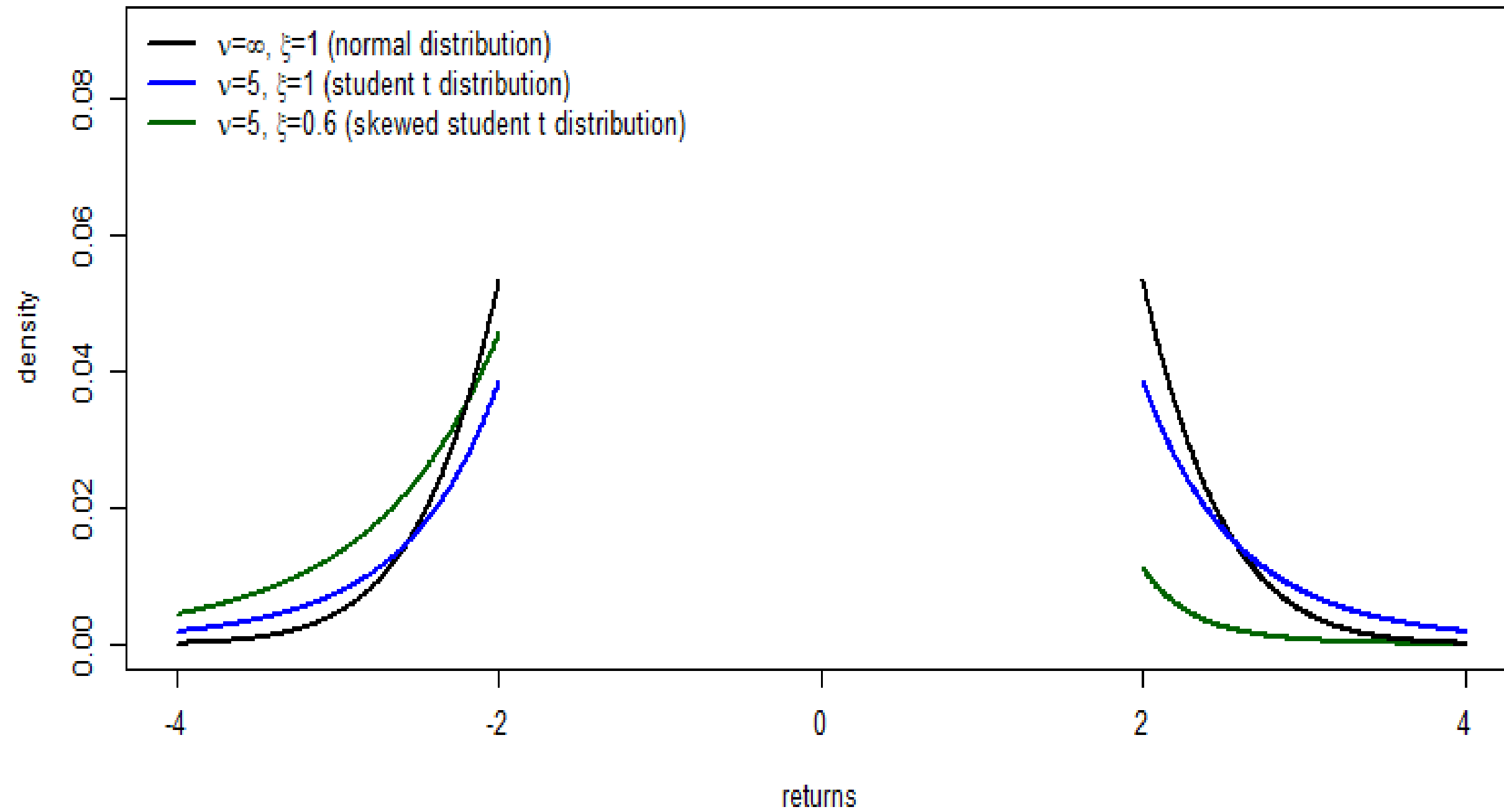
Parameters of the skewed student t distribution

- Compared to the normal distribution, the skewed student t distribution has two extra parameters:
 - Degrees of freedom parameter ν (in rugarch: `shape`): the lower is ν the fatter the tails.
 - Skewness parameter ξ (in rugarch: `skew`) : when $\xi = 1$: symmetry. When $\xi < 1$: negative skewness. For $\xi > 1$: positive skewness.
- Special cases:
 - When $\nu = \infty$ and $\xi = 1$: normal distribution.
 - When $\xi = 1$: student t distribution.









GARCH model estimation with skewed student t

- Set argument `distribution.model` to "sstd"

```
garchspec <- ugarchspec(mean.model = list(armaOrder = c(0,0)),  
                        variance.model = list(model = "sGARCH"),  
                        distribution.model = "sstd")
```

- Estimate the model

```
garchfit <- ugarchfit(data = sp500ret, spec = garchspec)
```

- We obtain

```
coef(garchfit)
```

mu	omega	alpha1	beta1	skew	shape
5.669200e-04	6.281258e-07	7.462984e-02	9.223701e-01	9.436331e-01	6.318621e+00



GARCH MODELS IN R

Let's practice!



GARCH MODELS IN R

**Size and sign of e_t matter
for volatility prediction!**

Kris Boudt

Professor of finance and econometrics



Negative returns induce higher leverage

- $R_t < 0$
- ↓ market value
- ↑ leverage = debt / market value
- ↑ volatility

Two equations

- Separate equations for the variance following negative and positive unexpected

return $e_t = R_t - \mu_t$:

$$\sigma_t^2 = \begin{cases} \omega + \alpha_1 e_{t-1}^2 & : e_{t-1} \leq 0 \\ \omega + \alpha_1 e_{t-1}^2 + \beta_1 e_{t-1}^2 & : e_{t-1} > 0 \end{cases}$$

In case of a positive surprise

... we take the usual GARCH(1,1) equation:

$$\sigma_t^2 = \begin{cases} \omega & : e_{t-1} \leq 0 \\ \omega + \alpha e_{t-1}^2 + \beta \sigma_{t-1}^2 & : e_{t-1} > 0 \end{cases}$$

In case of a negative surprise

- The predicted variance should be higher than after a positive surprise.
- This means a higher coefficient multiplying the squared prediction error, namely

$\alpha + \gamma$ instead of α with $\gamma \geq 0$

$$\sigma_t^2 = \begin{cases} \omega + (\alpha + \gamma)e_{t-1}^2 + \beta\sigma_{t-1}^2 & : e_{t-1} \leq 0 \\ \omega + \alpha e_{t-1}^2 + \beta\sigma_{t-1}^2 & : e_{t-1} > 0 \end{cases}$$

with $\gamma \geq 0$

= **GJR model** proposed Glosten, Jagannathan and Runkle.



How?

Change the argument `variance.model` of `ugarchspec()` from `model="sGARCH"` to `model="gjrgARCH"`:

```
garchspec <- ugarchspec(  
  mean.model=list(armaOrder=c(0,0)),  
  variance.model=list(model="sGARCH"),  
  distribution.model = "sstd")
```



```
garchspec <- ugarchspec(  
  mean.model=list(armaOrder=c(0,0)),  
  variance.model=list(model="gjrgARCH"),  
  distribution.model = "sstd")
```

Illustration on MSFT returns

- Estimate the model

```
garchfit <- ugarchfit(data = msftret, spec = garchspec)
```

- Inspect the GARCH coefficients

```
> coef(garchfit)[2:5]
```

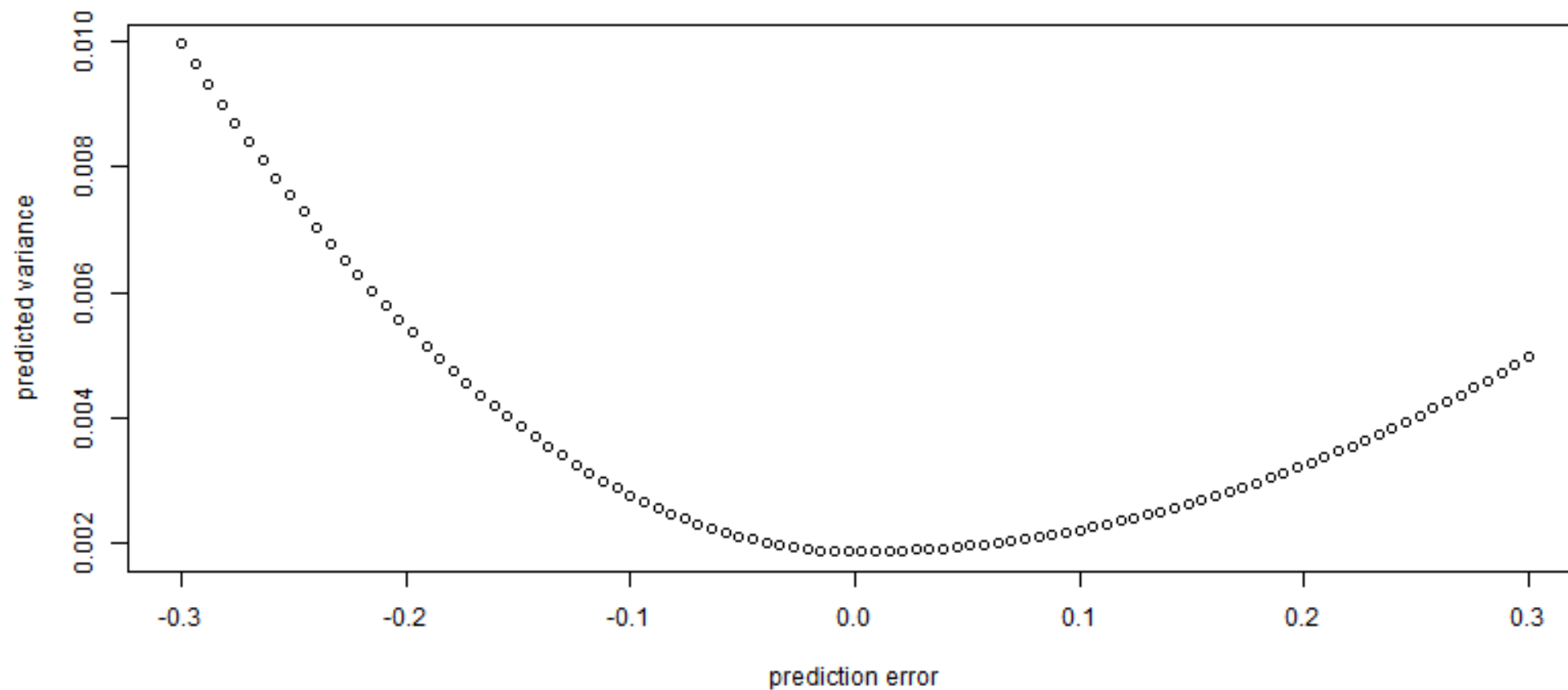
omega	alpha1	beta1	gamma1
2.007875e-06	3.423336e-02	9.363302e-01	5.531854e-02

$$\hat{\sigma}_t^2 = \begin{cases} 2.0 * 10^{-6} + 0.09e_{t-1}^2 + 0.94\sigma_{t-1}^2 & : e_{t-1} \leq 0 \\ 2.0 * 10^{-6} + 0.03e_{t-1}^2 + 0.94\sigma_{t-1}^2 & : e_{t-1} > 0 \end{cases}$$



Visualize volatility response using newsimpact()

```
out <- newsimpact(garchfit)
plot(out$zx, out$zy, xlab = "prediction error", ylab = "predicted variance")
```





GARCH MODELS IN R

**Let's estimate a GJR
GARCH model.**



GARCH MODELS IN R

No pain, no gain

Kris Boudt

Professor of finance and econometrics

GARCH-in-mean model

- Quantify the risk-reward trade-off.
- Risk: σ_t^2 . Reward: μ_t .
- GARCH-in-mean model:

$$\mu_t = \mu + \lambda \sigma_t^2$$

- $\lambda > 0$ is the risk/reward parameter indicating the increased in expected return per unit of variance risk.

How?

Change the argument `mean.model` in `ugarchspec()` from `list(armaOrder = c(0,0))` to `list(armaOrder = c(0,0), archm = TRUE, archpow = 2)`:

```
garchspec <- ugarchspec(  
  mean.model = list(armaOrder = c(0,0)),  
  variance.model = list(model = "gjrGARCH"),  
  distribution.model = "sstd")
```



```
garchspec <- ugarchspec(  
  mean.model = list(armaOrder = c(0,0), archm = TRUE, archpow = 2),  
  variance.model = list(model = "gjrGARCH"),  
  distribution.model = "sstd")
```

Application to daily S&P 500 returns

- Estimation

```
garchfit <- ugarchfit( data = sp500ret , spec = garchspec)
```

- Inspection of estimated coefficients for the mean

```
round(coef(garchfit)[1:2], 4)
      mu  archm
0.0002 1.9950
```

- Predicted mean returns

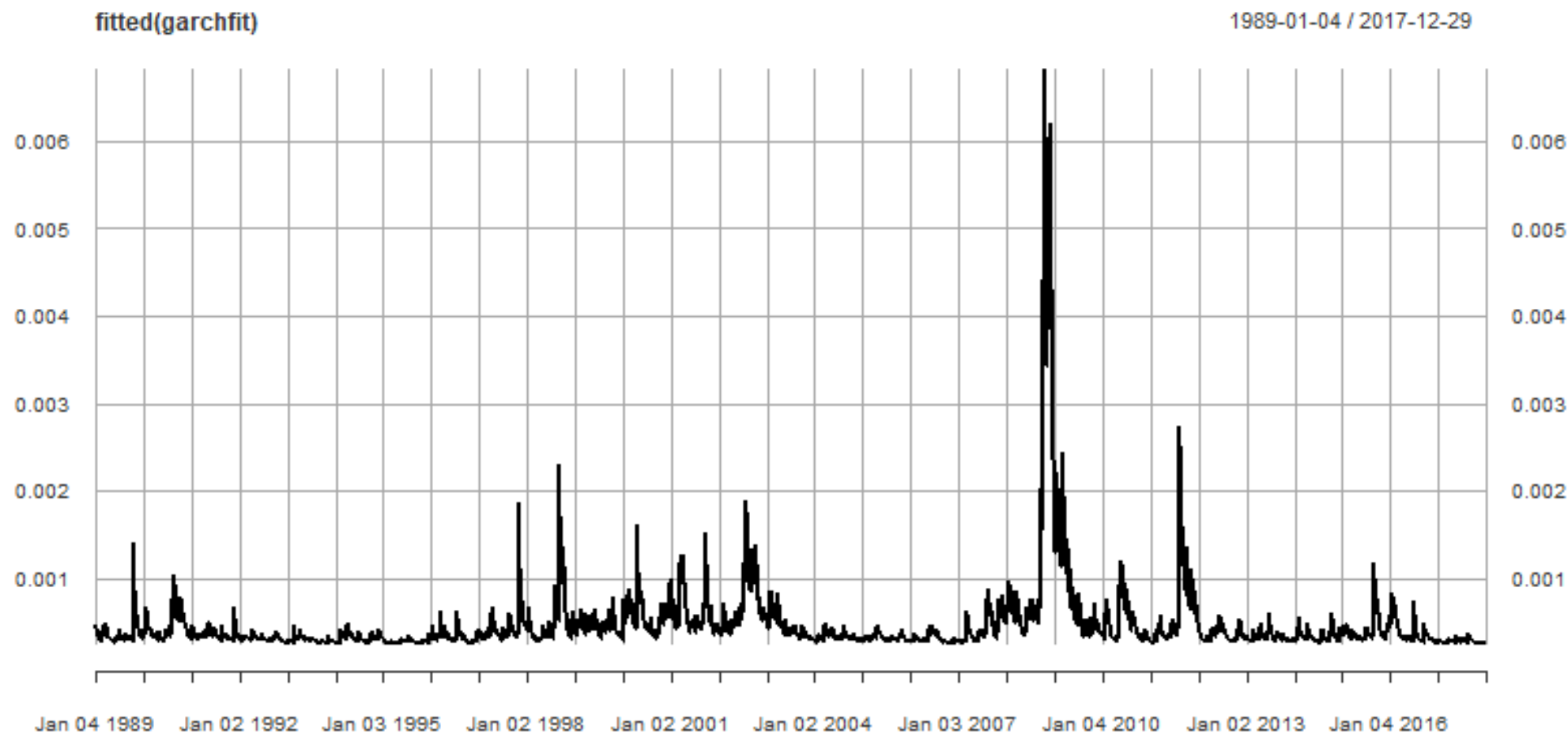
$$\hat{\mu}_t = 0.0002 + 1.9950\hat{\sigma}_t^2$$



Time series plot of predicted returns

- Plot them in R

```
plot(fitted(garchfit))
```



Today's return predicts tomorrow's return

- The GARCH-in-mean uses the financial theory of a risk-reward trade-off to build a conditional mean model
- Let's now use statistical theory to make a mean model that exploits the correlation between today's return and tomorrow's return.
- The most popular model is the AR(1) model:
 - AR(1) stands for autoregressive model of order 1
 - It predicts the next return using the deviation of the return from its long term mean value μ :

$$\mu_t = \mu + \rho (R_{t-1} - \mu)$$

A positive autoregressive coefficient

$$\mu_t = \mu + \rho (R_{t-1} - \mu)$$

- $\rho > 0$:
 - A higher (resp. lower) than average return is followed by a higher (resp. lower) than average return.
 - Possible explanation: markets underreact to news and hence there is momentum in returns.
- $|\rho| < 1$: Mean reversion: The deviations of R_t from μ are transitory.



A negative autoregressive coefficient

$$\mu_t = \mu + \rho (R_{t-1} - \mu)$$

- $\rho < 0$:
 - A higher (resp. lower) than average return is followed by a lower (resp. higher) than average return.
 - Possible explanation: markets overreact to news and hence there is reversal in returns.

Application to daily S&P 500 returns

- Specification and estimation of AR(1)-GJR GARCH with sst distribution

```
garchspec <- ugarchspec(  
  mean.model = list(armaOrder = c(1, 0)),  
  variance.model = list(model = "gjrGARCH"),  
  distribution.model = "sstd")  
garchfit <- ugarchfit(data = sp500ret, spec = garchspec)
```

- Estimates of the AR(1) model

```
round(coef(garchfit)[1:2], 4)
```

```
      mu      ar1  
0.0003 -0.0292
```



MA(1) and ARMA(1,1) model

- The Moving Average model of order 1 uses the deviation of the return from its conditional mean:

$$\mu_t = \mu + \theta (R_{t-1} - \mu_{t-1})$$

- ARMA(1,1) combines AR(1) and MA(1):

$$\mu_t = \mu + \rho (R_{t-1} - \mu) + \theta (R_{t-1} - \mu_{t-1})$$



How?

- MA(1)

```
garchspec <- ugarchspec(  
  mean.model = list(armaOrder = c(0, 1)),  
  variance.model = list(model = "gjrGARCH"),  
  distribution.model = "sstd")
```

- ARMA(1,1)

```
garchspec <- ugarchspec(  
  mean.model = list(armaOrder = c(1, 1)),  
  variance.model = list(model = "gjrGARCH"),  
  distribution.model = "sstd")
```



GARCH MODELS IN R

**Your turn to change the
mean.model argument**



GARCH MODELS IN R

Complexity has a price

Kris Boudt

Professor of finance and econometrics



Avoid unneeded complexity

- If you know
 - The mean dynamics are negligible
 - There is no leverage effect in the variance
 - The distribution is symmetric and fat-tailed

Then a constant mean, standard GARCH(1,1) with student t distribution is an appropriate specification to use:

```
garchspec <- ugarchspec(mean.model = list(armaOrder = c(0, 0)),  
                        variance.model = list(model = "sGARCH"),  
                        distribution.model = "std")
```



Restrict the parameter estimates

- If you know that the parameters
 - are equal to a certain value
 - or, are inside an interval
- Then you should impose this in the specification using the methods
 - `setfixed()`
 - `setbounds()`

Application to exchange rates

- Specification and estimation

```
garchspec <- ugarchspec(mean.model = list(armaOrder = c(0,0)),  
                        variance.model = list(model = "sGARCH"),  
                        distribution.model = "std")  
  
garchfit <- ugarchfit(data = EURUSDret, spec = garchspec)
```

- Estimation results

```
coef(garchfit)
```

mu	omega	alpha1	beta1	shape
-3.562136e-05	8.005123e-08	3.097322e-02	9.674496e-01	8.821902e+00



Example of setfixed()

- If you know `alpha1 = 0.05` and `shape = 6`: impose those values in the estimation.
- How? Use of `setfixed()` method on a `ugarchspec` object

```
setfixed(garchspec) <- list(alpha1 = 0.05, shape = 6)
```

- Result

```
garchfit <- ugarchfit(data = EURUSDret, spec = garchspec)
```

```
coef(garchfit)
```

mu	omega	alpha1	beta1	shape
-4.142922e-05	2.061772e-07	5.000000e-02	9.489622e-01	6.000000e+00



Bounds on parameters

- The GARCH parameters can be restricted to an interval.
- Sometimes the interval of plausible values is large:
 - To ensure the variance is positive, we require e.g. that all variance parameters $(\omega, \alpha, \beta, \gamma)$ are positive.
- Sometimes the interval of plausible values is smaller:
 - Likely values of α are in between 0.05 and 0.2
 - Likely values of β are in between 0.7 and 0.95
- Such bound constraints on the parameters can be imposed using the `setbounds()` method.



Example of setbounds()

```
setbounds(garchspec) <- list(alpha1 = c(0.05, 0.2), beta1 = c(0.8, 0.95))
```



Use your intuition to avoid unneeded complexity.

Use the information you have:

- to build simple (and smart) models
- to fix parameter values or set bounds
- to make the GARCH dynamics realistic:
 - mean reversion of the volatility around the sample standard deviation

```
sd(EURUSDret)
```

```
0.006194049
```



Volatility clusters and mean reversion of volatility



Variance targeting

- Mathematically, this means that the unconditional variance implied by the GARCH models equals the sample variance $\hat{\sigma}^2$.
- How? By setting the argument `variance.targeting = TRUE` in `variance.model` of `ugarchspec()`:

```
garchspec <- ugarchspec(mean.model = list(armaOrder = c(0,0)),  
                        variance.model = list(model = "sGARCH",  
                                             variance.targeting = TRUE),  
                        distribution.model = "std")
```

```
garchfit <- ugarchfit(data = EURUSDret, spec = garchspec)
```

```
all.equal(uncvariance(garchfit), sd(EURUSDret)^2, tol = 1e-4)
```

```
TRUE
```



GARCH MODELS IN R

**Let's impose restrictions on
the GARCH model**