Fitting_GARCH

Sarah-Katharina Umek

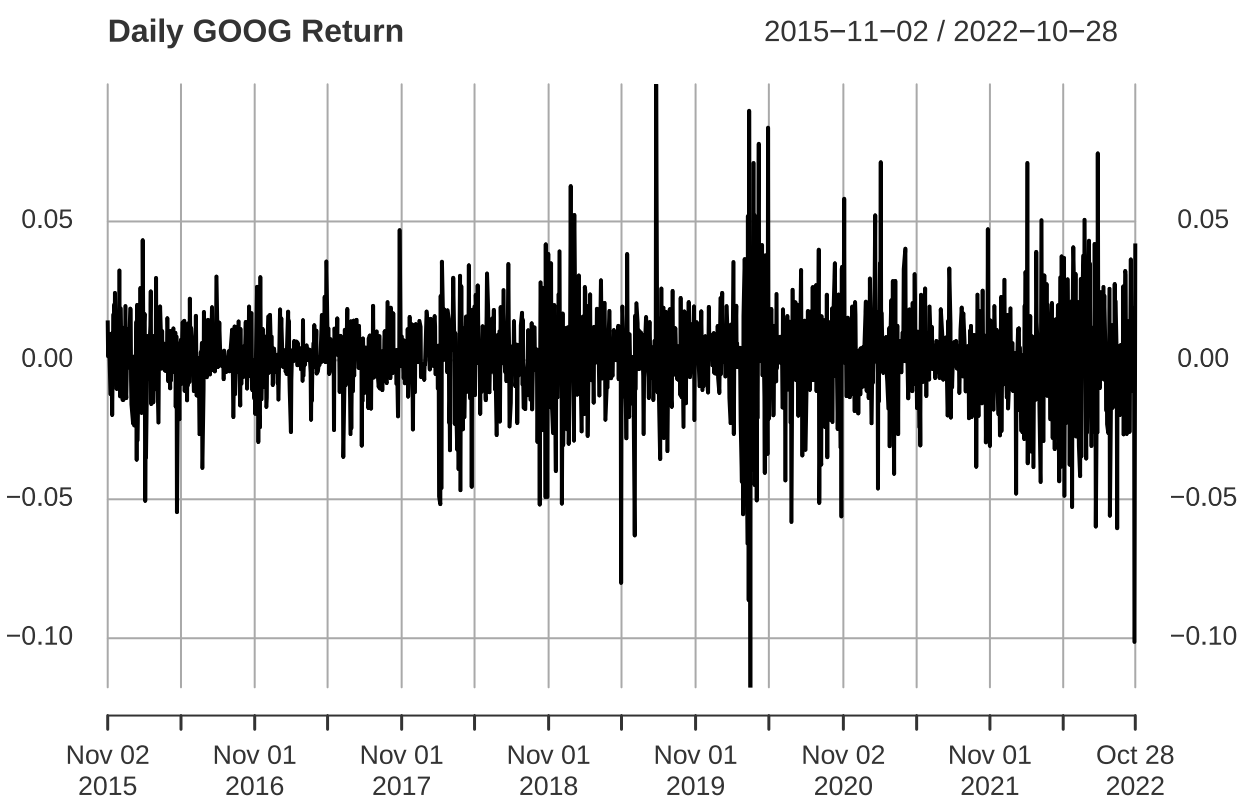
2022-10-25

Considering a time series observations} $y_t, t = 1, ..., T$ of your choice and investigate, if predictability in the conditional mean and/or the conditional variance is present:

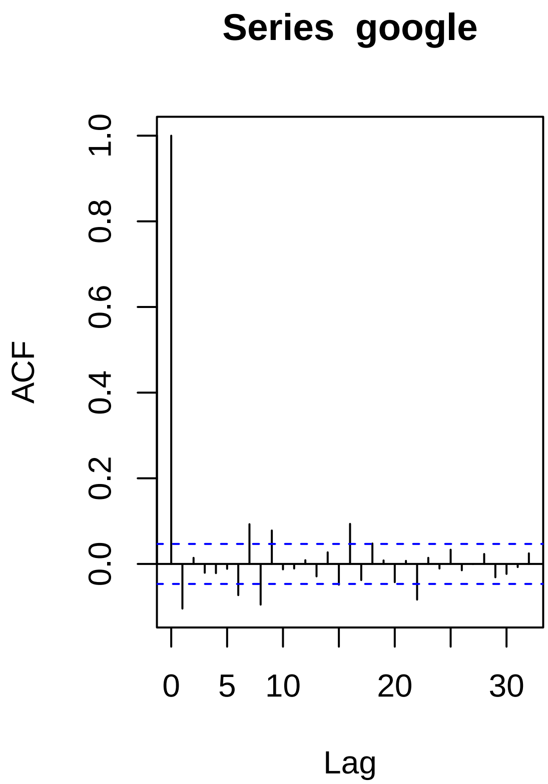
```
getSymbols("GOOG", from = "2015-10-30", to = "2022-10-30", warnings = FALSE,
    auto.assign = TRUE)
```

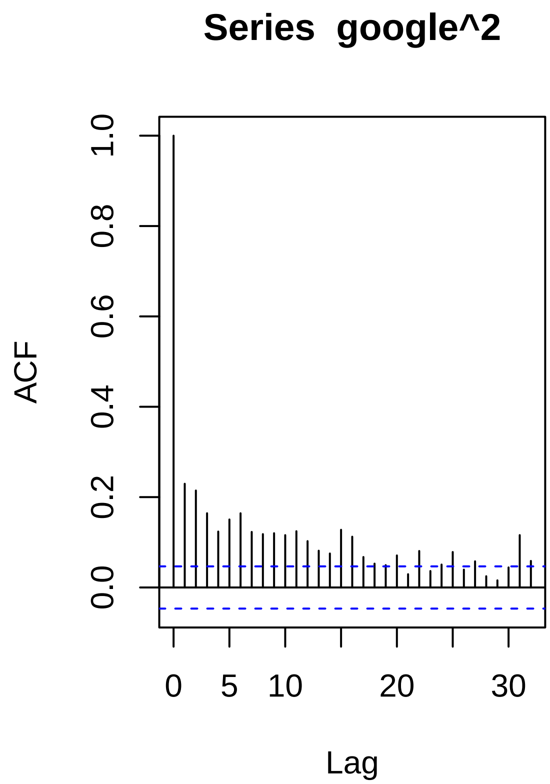
```
## [1] "GOOG"
```

```
google <- Cl(na.omit(GOOG))
google <- na.omit(diff(log(google)))
plot(google, col = "black", main = "Daily GOOG Return", xlab = "Time")
```




```
par(mfrow = c(1, 2))
acf(google)
acf(google^2)
```





Continuing by estimating an ARMA-GARCH model for this time series and choosing an appropriate model orders.

```
model_specs <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1,
    1)), mean.model = list(armaOrder = c(1, 1), include.mean = FALSE),
    distribution.model = "norm")

fit <- ugarchfit(data = google, spec = model_specs)
infocriteria(fit)
```

```
##
## Akaike       -5.445242
## Bayes        -5.429700
## Shibata      -5.445258
## Hannan-Quinn -5.439498
```

```
model_specs2 <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1,
    1)), mean.model = list(armaOrder = c(0, 0), include.mean = FALSE),
    distribution.model = "norm")

fit2 <- ugarchfit(data = google, spec = model_specs2)

infocriteria(fit2)
```

```
##
## Akaike       -5.446536
## Bayes        -5.437212
## Shibata      -5.446542
## Hannan-Quinn -5.443091
```

```r
model_specs2_t <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1,
    1)), mean.model = list(armaOrder = c(0, 0), include.mean = FALSE),
    distribution.model = "std")

fit2_t <- ugarchfit(data = google, spec = model_specs2_t)

infocriteria(fit2_t)
```

```
##
## Akaike       -5.582260
## Bayes        -5.569827
## Shibata      -5.582271
## Hannan-Quinn -5.577666
```

```r
model_specs3 <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1,
    1)), mean.model = list(armaOrder = c(0, 1), include.mean = FALSE),
    distribution.model = "norm")

fit3 <- ugarchfit(data = google, spec = model_specs3)
infocriteria(fit3)
```

```
##
## Akaike       -5.446131
## Bayes        -5.433698
## Shibata      -5.446141
## Hannan-Quinn -5.441536
```

```r
model_specs4 <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1,
    1)), mean.model = list(armaOrder = c(1, 0), include.mean = FALSE),
    distribution.model = "norm")

fit4 <- ugarchfit(data = google, spec = model_specs4)
infocriteria(fit4)
```

```
##
## Akaike       -5.446134
## Bayes        -5.433701
## Shibata      -5.446144
## Hannan-Quinn -5.441539
```

```r
model_specs5 <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1,
    1)), mean.model = list(armaOrder = c(2, 2), include.mean = FALSE),
    distribution.model = "norm")

fit5 <- ugarchfit(data = google, spec = model_specs5)
infocriteria(fit5)
```

```
##
## Akaike       -5.445923
## Bayes        -5.424166
## Shibata      -5.445955
## Hannan-Quinn -5.437883
```

```
model_specs6 <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(2,
    2)), mean.model = list(armaOrder = c(1, 1), include.mean = FALSE),
    distribution.model = "norm")

fit6 <- ugarchfit(data = google, spec = model_specs5)
infocriteria(fit6)
```

```
##
## Akaike       -5.445923
## Bayes        -5.424166
## Shibata      -5.445955
## Hannan-Quinn -5.437883
```

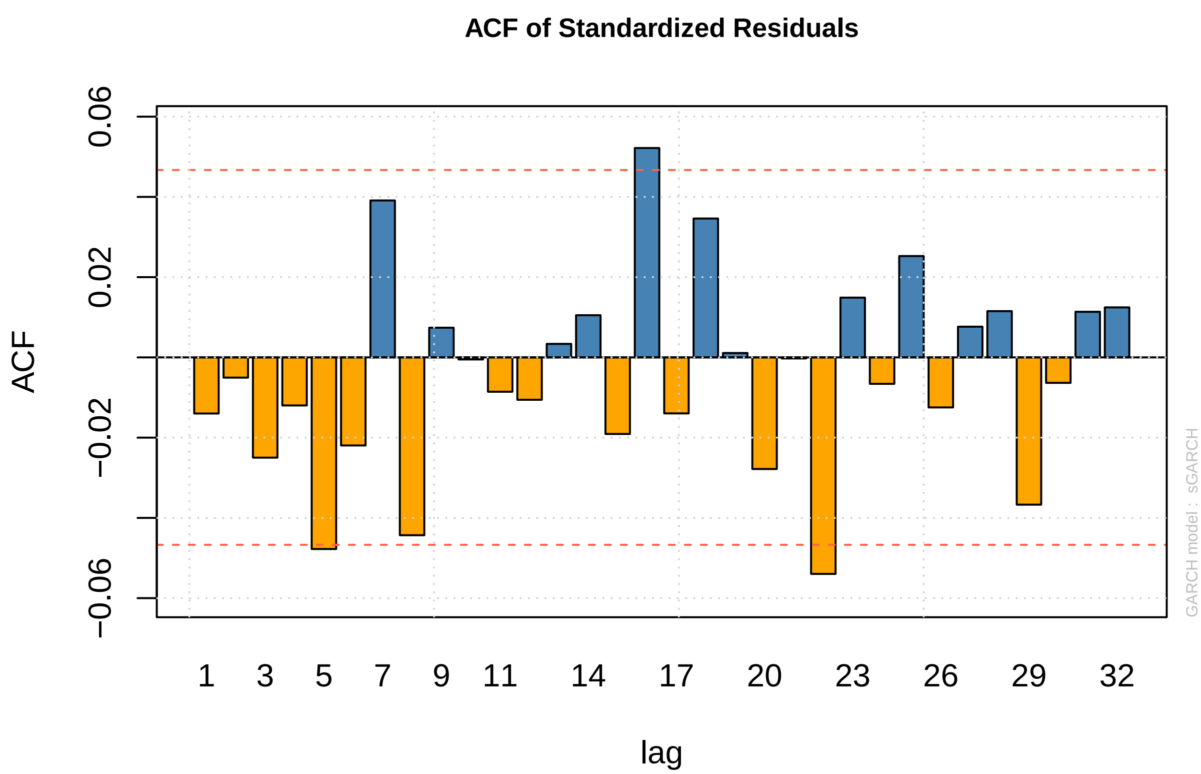
```
model_specs6_t <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(2,
    2)), mean.model = list(armaOrder = c(0, 0), include.mean = FALSE),
    distribution.model = "std")

fit6_t <- ugarchfit(data = google, spec = model_specs2_t)

infocriteria(fit6_t)
```

```
##
## Akaike       -5.582260
## Bayes        -5.569827
## Shibata      -5.582271
## Hannan-Quinn -5.577666
```

```
plot(fit2_t, which = 10)
plot(fit6_t, which = 10)
```



Using AIC and BIC to compare the models, we end up choosing an ARMA(0,0) GARCH(1,1) (no significant improvement for higher order ARMA models), however there is an improvement sung student-t distributions. Checking the standardized residuals for the selected model, we see that the ACF looks okay so we proceed with the Thus our model looks as follows:

$$
r_t = \mu_t + X_t, \quad \text{with } X_t = \sigma_t \varepsilon_t
$$
$$
\sigma_t^2 = \alpha_0 + \alpha_1 X_{t-1}^2 + \gamma_1 \sigma_{t-1}^2
$$

Next we discuss predictability in the light of the estimated parameters for the selected model.

```
fit2_t
```

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : sGARCH(1,1)
## Mean Model   : ARFIMA(0,0,0)
## Distribution : std
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error  t value Pr(>|t|)
## omega   0.000004    0.000004   1.0621 0.288169
## alpha1  0.094106    0.023190   4.0581 0.000049
## beta1   0.904893    0.022690  39.8807 0.000000
## shape   4.062921    0.433332   9.3760 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error  t value Pr(>|t|)
## omega   0.000004    0.000011  0.34598  0.72936
## alpha1  0.094106    0.058206  1.61678  0.10593
## beta1   0.904893    0.063002 14.36286  0.00000
## shape   4.062921    0.535657  7.58493  0.00000
##
## LogLikelihood : 4919.18
##
## Information Criteria
## ------------------------------------
##
## Akaike       -5.5823
## Bayes        -5.5698
## Shibata      -5.5823
## Hannan-Quinn -5.5777
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.3450  0.5569
```

```
## Lag[2*(p+q)+(p+q)-1][2]    0.3674  0.7593
## Lag[4*(p+q)+(p+q)-1][5]    1.9506  0.6302
## d.o.f=0
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                    0.09009  0.7641
## Lag[2*(p+q)+(p+q)-1][5]   0.48225  0.9604
## Lag[4*(p+q)+(p+q)-1][9]   1.21506  0.9755
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]   0.01345 0.500 2.000  0.9077
## ARCH Lag[5]   0.03221 1.440 1.667  0.9974
## ARCH Lag[7]   0.52528 2.315 1.543  0.9760
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  5.2604
## Individual Statistics:
## omega  2.5110
## alpha1 0.6511
## beta1  0.6115
## shape  0.7207
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          1.07 1.24 1.6
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value   prob sig
## Sign Bias           0.7281 0.4666
## Negative Sign Bias  0.4708 0.6378
## Positive Sign Bias  0.1384 0.8900
## Joint Effect        0.5943 0.8977
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     43.51     0.001104
## 2    30     41.16     0.066634
## 3    40     56.22     0.036520
## 4    50     65.49     0.057690
##
##
## Elapsed time : 0.167475
```

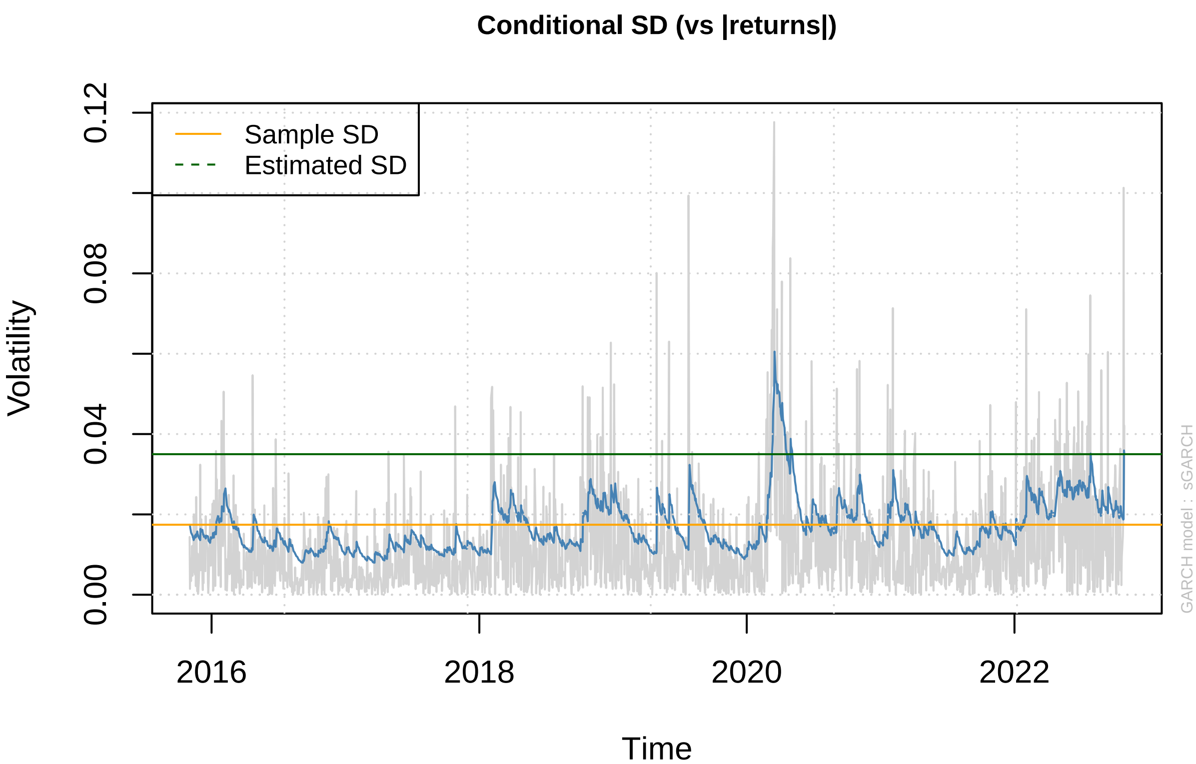
```r
sd_sample <- sd(google)
sd_sample
```

```
## [1] 0.01743161
```

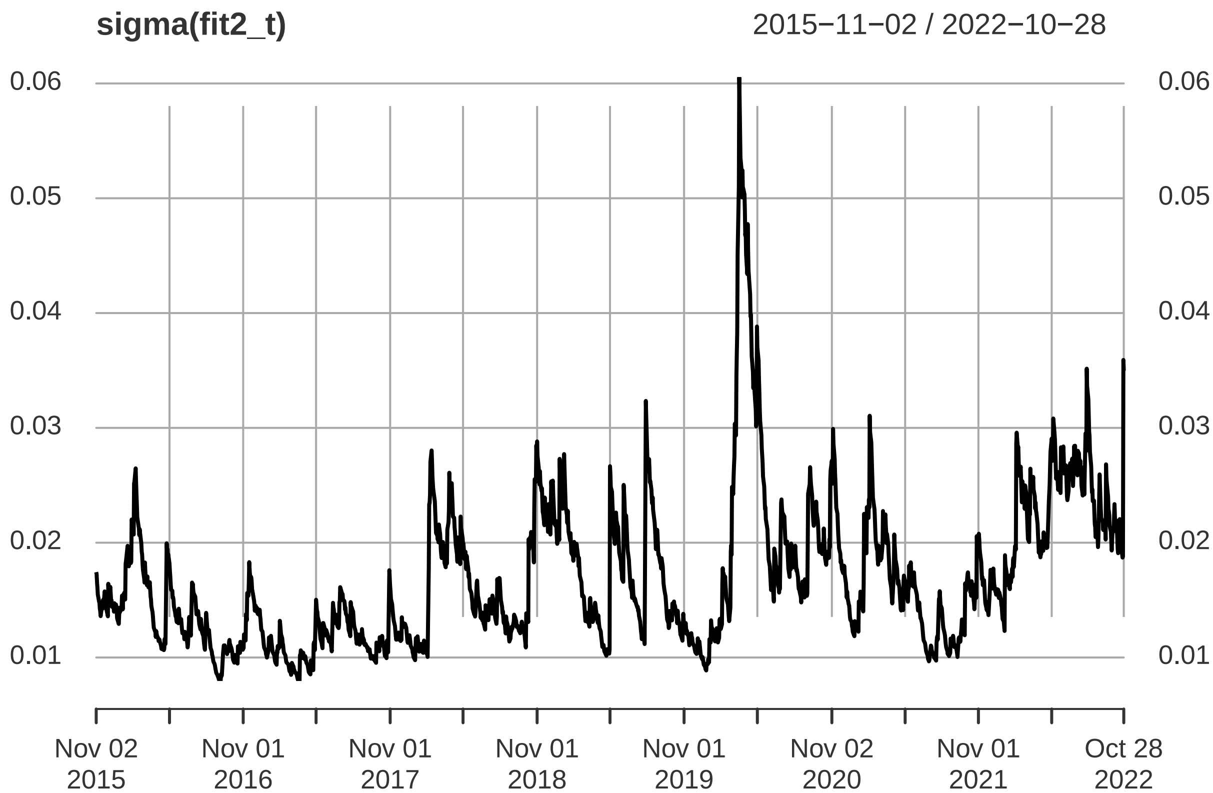
```r
sd_estimates <- as.data.frame(tail(sigma(fit2_t), n = 1))[, 1]
sd_estimates
```

```
## [1] 0.0349866
```

```r
plot(fit2_t, which = 3)
abline(h = sd_sample, col = "orange")
abline(h = sd_estimates, col = "darkgreen")
legend("topleft", legend = c("Sample SD", "Estimated SD"), col = c("orange",
    "darkgreen"), lty = 1:2, cex = 0.8)
```

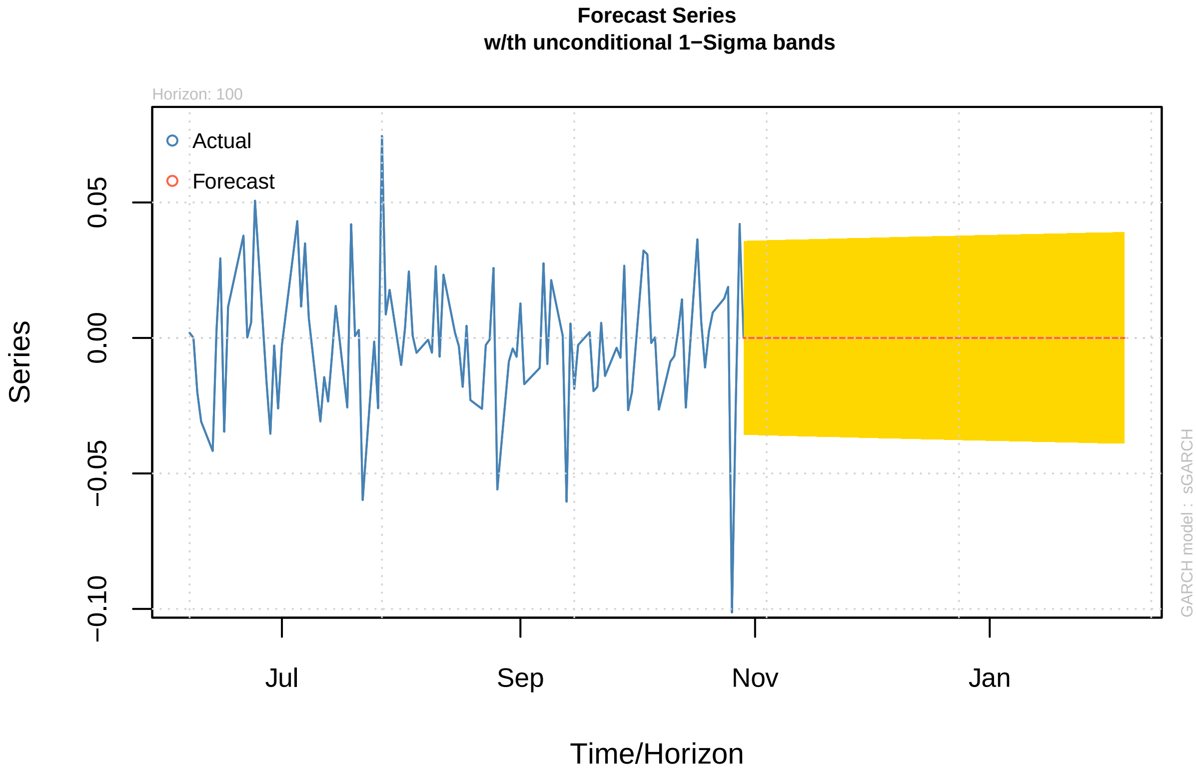



```r
plot(sigma(fit2_t))
```




```
forecast_model <- ugarchforecast(fit2_t, data = NULL, n.ahead = 100, n.roll = 0,
    out.sample = 100)

plot(forecast_model, which = 1)
```



Forecast Series
w/th unconditional 1−Sigma bands
Horizon: 100
Actual
Forecast
Series
0.05
0.00
−0.05
−0.10
Jul
Sep
Nov
Jan
Time/Horizon
GARCH model : sGARCH