Applied Economic Forecasting

8. G(ARCH) Models

Section 1

Introduction

Market Returns: The S&P 500

Recall, the price returns of a series can be computed as

$$R_t = \frac{(P_t - P_{t-1})}{P_{t-1}} \approx \log\left(\frac{P_t}{P_{t-1}}\right)$$

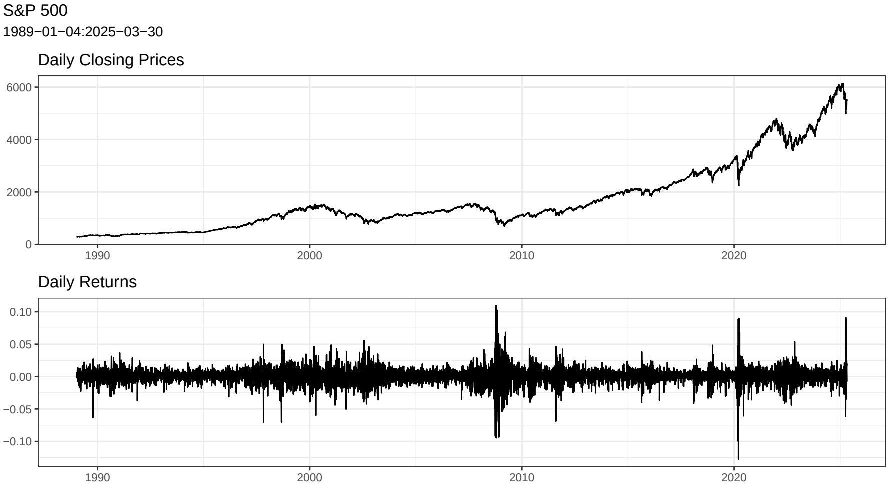
Where P_t is the current price of the stock (for example) and P_{t-1} is yesterday's price.

```
sp500 <- getSymbols(Symbols = "^GSPC", src = "yahoo",
                    from = "1989-01-04",
                    to = "2025-04-30", auto.assign = FALSE) |>
  fortify() |> as_tsibble(index = Index) |>
  mutate(ret = GSPC.Close |> log() |> difference()) # Compute returns

a <- sp500 |> autoplot(GSPC.Close) + labs(title = "Daily Closing Prices")

b <- sp500 %>% autoplot(ret) + labs(title = "Daily Returns")
```

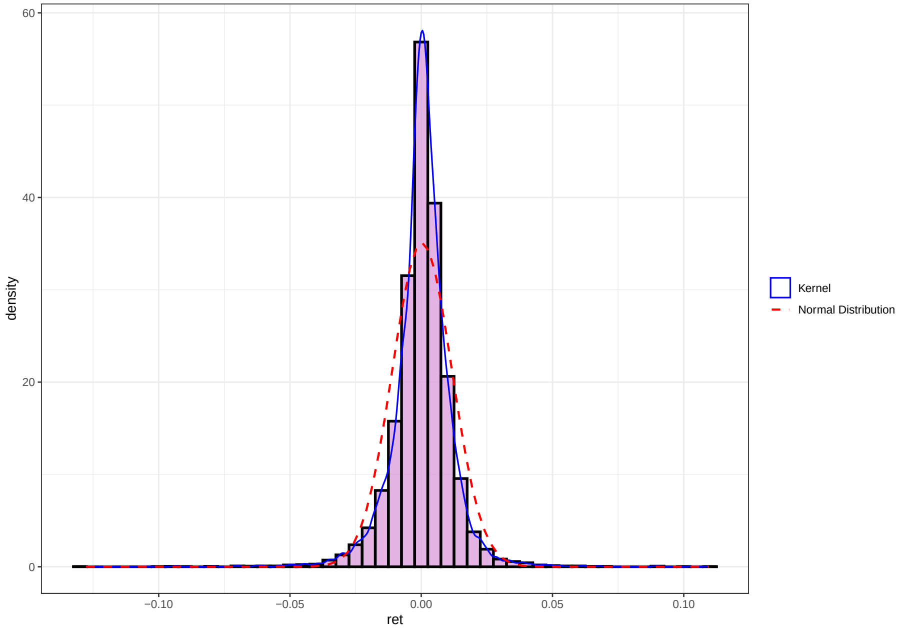
```r
a/b +
  plot_annotation(title = "S&P 500",
                  subtitle = "1989-01-04:2025-03-30") &
  labs(x = NULL, y = NULL)
```



```r
# Compute sample mean and sd
mu.r <- mean(sp500$ret, na.rm = TRUE)
sig.r <- sd(sp500$ret, na.rm = TRUE)

hist <- sp500 |>
  ggplot(aes(x = ret)) +
  geom_histogram(binwidth = 0.005, aes(y =after_stat(density)),
                 fill = "plum", alpha = 0.8, col = "black", size = 1) +
  geom_density(aes(col = "Kernel"), lwd = 0.6) +
  stat_function(fun = dnorm, args = list(mean = mu.r, sd = sig.r),
                aes(col = "Normal Distribution"), linetype = "dashed",
                linewidth = 0.8) +
  scale_color_manual(values = c("Kernel" = "blue",
                                "Normal Distribution" = "red"),
                     name = NULL)
```

```
hist
```



density
ret
60
40
20
0
−0.10
−0.05
0.00
0.05
0.10
Kernel
Normal Distribution

```r
require(psych)

rt <- sp500$ret |> data.frame()

describe(rt)[c("mean","median", "sd", "skew", "kurtosis")] |>
  knitr::kable(digits = 3,
               caption = "Descriptive Statistics of Daily Returns")
```

Table 1: Descriptive Statistics of Daily Returns

	mean	median	sd	skew	kurtosis
sp500.ret	0	0.001	0.011	-0.381	10.9

The S&P 500: Annualized Volatility

Key Points

- SP500 Returns are mean zero.
- The variability of the price return changes over the sample period.
  - We can use the standard deviation (σ) to measure this variability (volatility) over time.
- Over the full sample, the standard deviation of returns was approximately 1.138% per day.
- We can annualize the daily volatility by multiplying σ by the square root of the number of trading days ($\sqrt{252}$).

```
(sd.sp500 * sqrt(252) -> annual.sd.sp500)
```

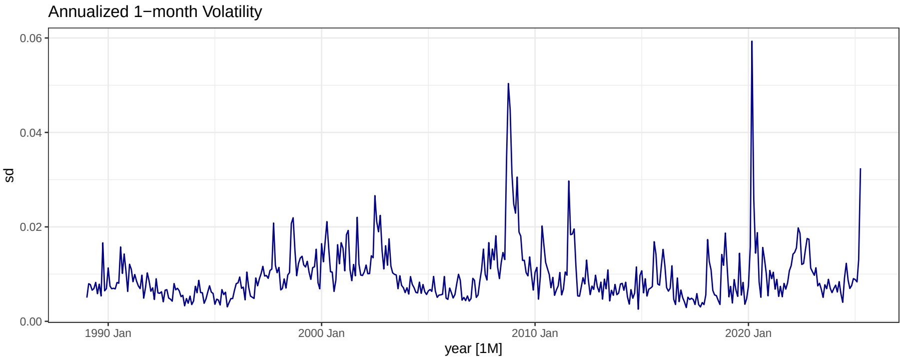
```
## [1] 0.181
```

The SP500 has an annualized volatility of 18.06%.

The S&P 500: Time Varying Volatility

Instead of the static approach above, and realizing that the volatility varies over time, we could compute the **historical** standard deviation yearly or monthly, for example, instead.

```
(a <- sp500 |> index_by(year = yearmonth(Index)) |>
  summarise(sd = sd(ret, na.rm = TRUE)) |>
  autoplot(sd, col = "blue4") + labs(title = "Annualized 1-month Volatility") )
```

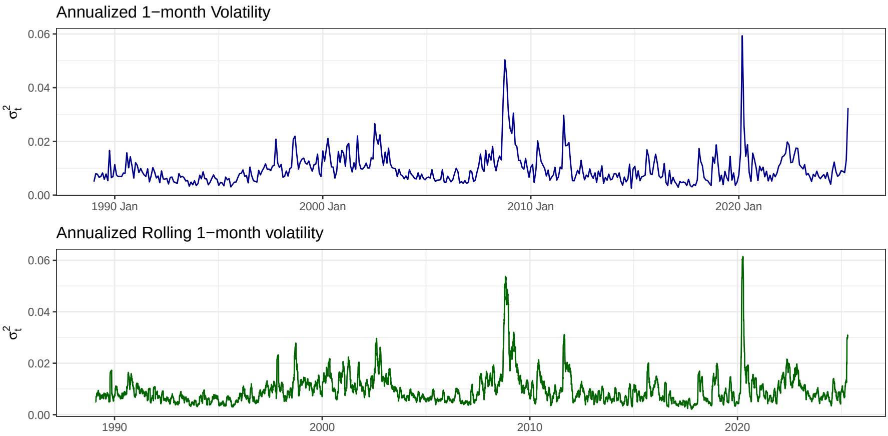


The S&P 500: Rolling Volatility

Conversely, we can compute the volatility using a rolling window approach. In the example, we will compute the 1-month rolling volatility. We will use a standard 21-days window (the average number of trading days within a month). We could just as easily compute the 2-month (42-days window) or 3-month (63-days window) rolling volatility.

```
sp500 <- sp500 |> mutate(
  sd = slider::slide_dbl(ret, ~ sd(.x, na.rm = TRUE),
                          .before = 20, .complete = TRUE))
b <-   sp500 |> autoplot(sd, colour = "darkgreen") +
  ggtitle("Annualized Rolling 1-month volatility")
```

```r
a/b & labs(y = bquote(sigma[t]**2), x = NULL)
```



These observations form the motivation for the GARCH Model framework.

Section 2

GARCH Models

Conditional vs Unconditional Variance

The **unconditional variance** is the standard measure of the variance.

$$var(x) = \mathbf{E}(x - \mathbf{E}(x))^2$$

The **conditional variance** is the **true measure of our uncertainty** about a variable given a model and the information set Ω

$$\text{cond. } var(x) = \mathbf{E}(x - \mathbf{E}(x|\Omega))^2$$

We will discuss the information matrix in detail shortly.

Motivation

- From our calculations earlier, using the standard deviation σ as our volatility measures is backward looking. We are using past data to understand what the volatility **was**.
- GARCH models are a bit more flexible and allow us to predict future volatility as well.

Useful Notations

Ω_{t-1}: The full set of information known at time t-1. For example, all the return values known (observed) at time t-1. These would include $R_{t-1}, R_{t-2}, R_{t-3}, \ldots$

$\mu_t = \mathbf{E}(R_t|\Omega_{t-1})$: This tells us that the prediction of the returns in time t is the expected value of R_t conditional on the information 1 period earlier, $t-1$.

Prediction error: $e_t = R_t - \mu_t$

Predicted variance:

$$\sigma_t^2 = var(R_t|\Omega_{t-1})$$
$$\sigma_t^2 = \mathbf{E}(R_t - \mu_t|\Omega_{t-1})$$
$$\sigma_t^2 = \mathbf{E}(e_t^2|\Omega_{t-1})$$
$$\sigma = \sqrt{\sigma_t^2}$$

Modeling the Mean

- When we are coding, we will need to decide on a formula to replace the expectation formula above.
- We could find μ_t using a rolling mean model where

$$\mu_t = \frac{1}{m} \sum_{i=1}^{m} R_{t-i}$$

- We could also achieve this using our **ARMA** models from class.

Modeling the Variance

In the case of the variance, you can take the average of the m most recently observed squared prediction errors.

That is,

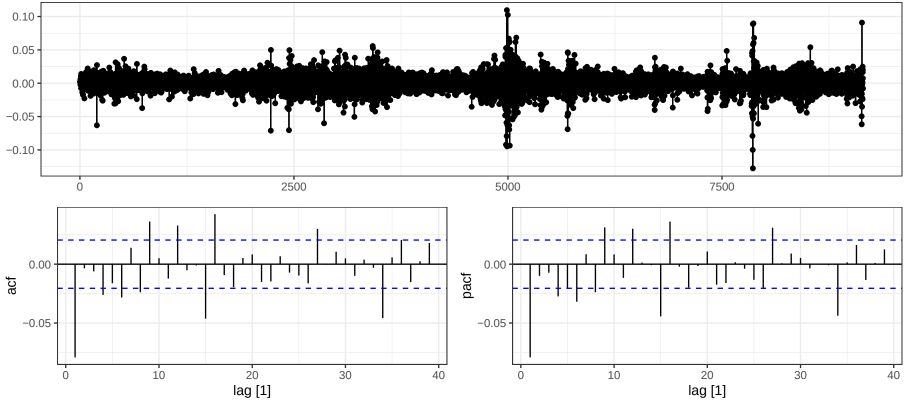
$$\sigma_t = \frac{1}{m} \sum_{i=1}^{m} e_{t-i}^2$$

Quick Notes

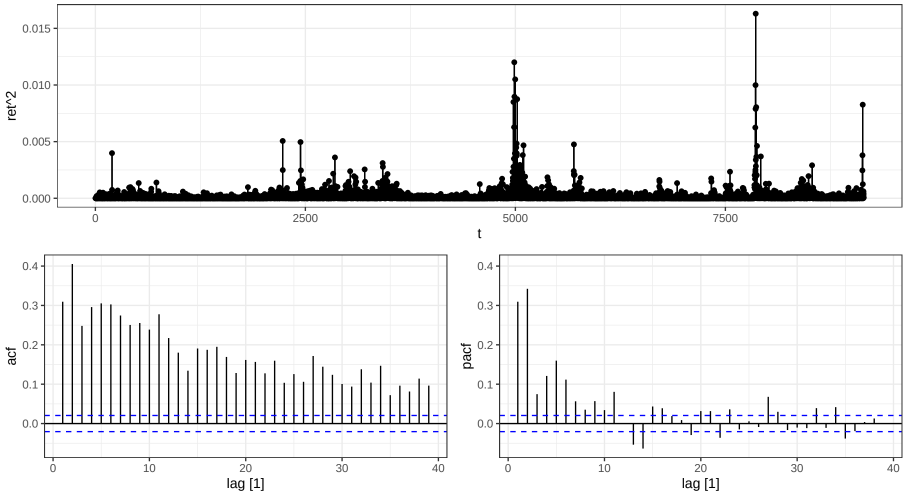
- All m observations are equally weighted in this approach regardless of when they are observed.
- We would expect, however, that the future variance is more affected by the more recent events than by those in the distant past.
  - Therefore, we can achieve a higher forecasting accuracy by giving more weight to the most recent observations (think of an exponential smoothing type of approach).
- This obvious shortcoming motivates the use of an ARCH model.

Autoregressive Conditional Heteroskedasticity (ARCH)

```
sp500 |> mutate(t = row_number()) |>
  update_tsibble(index = t) |>
  gg_tsdisplay(ret, plot_type = "partial") & labs(y= NULL, x = NULL)
```



```
sp500 |> mutate(t = row_number()) |>
  update_tsibble(index = t) |>
  gg_tsdisplay(ret**2, plot_type = "partial")
```



Autoregressive Conditional Heteroskedasticity (ARCH)

ARCH(p) Model Specification

$$\sigma_t^2 = \omega + \sum_{i=1}^{p} \alpha_i e_{t-i}^2$$

In an ARCH equation, the predicted variance is the sum of a constant and a weighted sum of `p` lagged observed squared prediction errors.

- If there is an ARCH effect, it can be tested by the statistical significance of the estimated coefficients.
- If they are significantly different from zero, we can conclude that there is an ARCH effect.

Autoregressive Conditional Heteroskedasticity (ARCH)

```r
# Ljung-Box Test for ARCH effects

sp500 |> mutate(t = row_number()) |>
  update_tsibble(index = t) |>
  features(ret**2, ljung_box, lag = 20)
```

```
## # A tibble: 1 x 2
##   lb_stat lb_pvalue
##     <dbl>     <dbl>
## 1  11074.         0
```

Generalized ARCH (GARCH)

In practice, we often use a GARCH(1,1) model for our empirical analysis of market and returns volatility.

$$\sigma_t^2 = \omega + \alpha e_{t-1}^2 + \beta \sigma_{t-1}^2$$

Parameter Restrictions

- $\omega, \alpha, \beta > 0$: this ensures that the variances are positive at all times.
- $\alpha + \beta < 1$: this ensures that we have stability in the system.
  - A shock to the system (through α) will die out over time.
  - The predicted variance, σ_t^2 will always return to its long run mean.
  - This implies that our variance is **mean reverting**.

$$\text{var}_{LR} = \frac{\omega}{1 - \alpha - \beta}$$

Assessing the LR Variance

Suppose for a daily return series we have the following model:

$$\sigma_t^2 = 0.000002 + 0.13e_{t-1}^2 + 0.86\sigma_{t-1}^2$$

The LR variance is

$$\frac{\omega}{1-\alpha-\beta} = \frac{0.000002}{1-0.13-0.86} = 0.0002$$

hence the LR volatility per day is $\sqrt{0.0002} \times 100 = 1.4\%$.

```r
# Parameters for alpha and beta
alpha <- 0.13
beta <- 0.86

# set LR variance equal to the sample variance
var.lr <- var(sp500$ret, na.rm = TRUE)

# Compute omega
omega <- var.lr*(1-alpha-beta)

# pred. errors
e <- sp500$ret - mean(sp500$ret,na.rm=TRUE)
e2 <- (e**2)[-1] ## Drop the first observation

# Predicting the conditional variance
N <- nrow(sp500)-1
ht <- rep(NA,N)

for(i in 1:N){
if(i==1){
  ht[i] <- var.lr # set first observation to the sample variance
} else
  ht[i] <- omega + alpha*e2[(i-1)] + beta*ht[(i-1)]
}
```

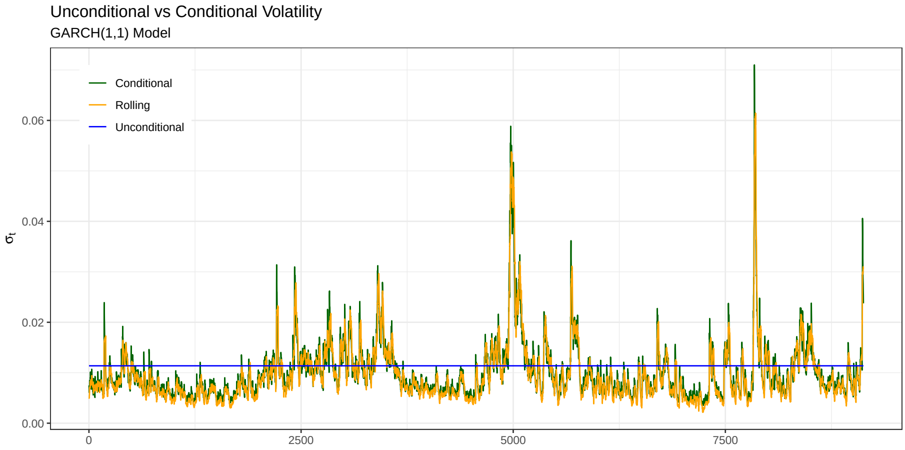
```r
## Plot Uncond. and Cond. (implied) Volatility

vols <- data.frame(vol.unc = sqrt(var.lr),
           vol.cond = sqrt(ht)[-c(1:19)],
           historical = sqrt(sp500$ret**2)[-c(1:20)],
           sd = (sp500 |> pull(sd))[-c(1:20)])

vols_long <- vols |>
  mutate(t = 1:n()) |>
  pivot_longer(cols = c(vol.unc, vol.cond, sd), names_to = "Type",
               values_to = "Vol") |>
  mutate(Type = recode(Type,
                       vol.unc = "Unconditional",
                       vol.cond = "Conditional",
                       sd = "Rolling"))

v <- ggplot(vols_long, aes(x = t, y = Vol, color = Type)) +
  geom_line() +
  scale_color_manual(
    values = c("Unconditional" = "blue",
               "Conditional" = "darkgreen",
               "Rolling" = "orange"))
```

```r
v + labs(title = "Unconditional vs Conditional Volatility",
         subtitle = "GARCH(1,1) Model", col = NULL,
         y = bquote(sigma[t]), x = NULL) +
  theme(legend.position = c(0.1, 0.85))
```



Section 3

GARCH Models in R

Estimating a GARCH Model

Take the GARCH (1,1) model

$$R_t = \mu + \varepsilon_t$$
$$\varepsilon_t \sim \mathbf{N}(0, h_t)$$
$$h_t = \omega + \alpha\varepsilon_{t-1}^2 + \beta h_{t-1}$$

- Notice that I replaced σ^2 from earlier with h_t since that is how many books define the variance of the residuals.
- There are 4 parameters to be estimated in this model: μ, ω,α, and β.
- A popular approach is to use the method of Maximum likelihood (MLE) to calculate the values of these parameters.

Estimating a GARCH Model (via MLE)

Probability density function (pdf) for a normal distribution is:

$$f(x_i;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[-\frac{(x_i-\mu)^2}{2\sigma^2}\right]$$

The likelihood function is therefore:

$$\mathcal{L}(\mu,\sigma) = (2\pi\sigma^2)^{-n/2}\exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2\right]$$

Taking the logs:

$$\log\mathcal{L} = -\frac{n}{2}log(2\pi) - \frac{n}{2}\boxed{log(\sigma^2)} - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\boxed{(x_i-\mu)^2}$$

Specific to our model, we have:

$$\log\mathcal{L} = -\frac{T}{2}log(2\pi) - \frac{T}{2}\log h_t - \frac{1}{2h_t}\sum_{i=1}^{T}\varepsilon^2$$

$\varepsilon_t = y_t - \mu - \sum_{i=1}^{p}\phi_i y_{t-i}$ and $h_t = \omega + \sum_{i=1}^{p}\alpha_i\varepsilon_{t-i}^2 + \sum_{j=1}^{q}\theta_j h_{t-j}$

rugarch in Action

We will use the package **rugarch** for our model estimations. The Workflow is as follows:

1. Use `ugarchspec()` function to specify the desired GARCH model

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

2. Pass `ugarchspec()` arguments to the `ugarchfit()` command.

```
fit.garch <- ugarchfit(spec = specs,
                       data = sp500[-c(1:20),]$ret)
```

3. Use the `ugarchforecast()` make predictions about the future volatility of our returns series.

```
fore.fit <- ugarchforecast(fit.garch, n.ahead = 30)
```

Useful Commands

We can extract various elements of our fitted model.

`coef()`: extracts the model coefficients

```
coef(fit.garch) %>% t() %>% format(digits = 3) %>% knitr::kable()
```

mu	omega	alpha1	beta1
5.98e-04	2.03e-06	1.03e-01	8.80e-01

`uncvariance`: delivers the **unconditional** Variance

```
unvar <- uncvariance(fit.garch)
cat("The Long run volatility of the S&P500 is",
    sqrt(unvar)*100,"%")
```

```
## The Long run volatility of the S&P500 is 1.11 %
```

Useful Commands

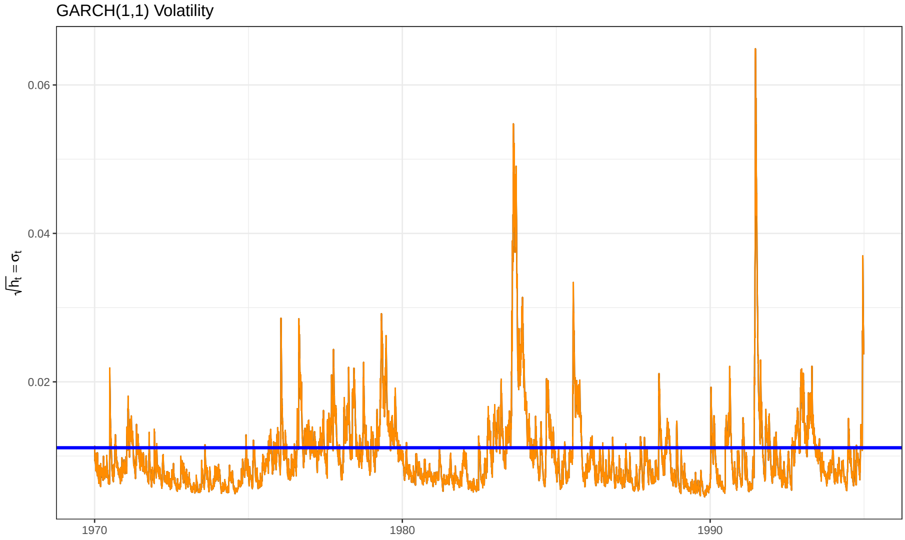
`fitted`: extracts the predicted mean

```
fitted(fit.garch)
```

`sigma`: extracts the predicted volatilities

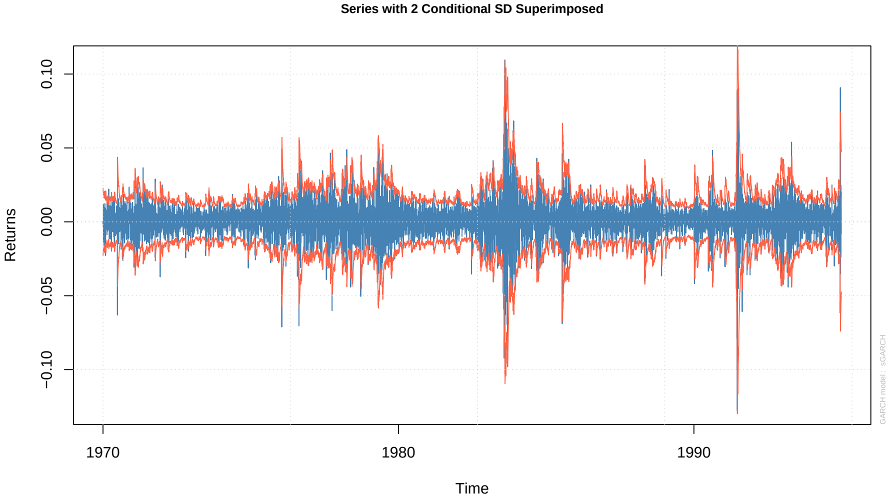
```
sig <- sigma(fit.garch) %>% autoplot() +
  geom_line(col = "darkorange") +
  geom_hline(yintercept = sqrt(unvar),
             col = 'blue', lwd = 1.2) +
  labs(title = "GARCH(1,1) Volatility",
       y = bquote(sqrt(h[t]) == sigma[t]), x = NULL)
```

```
sig
```



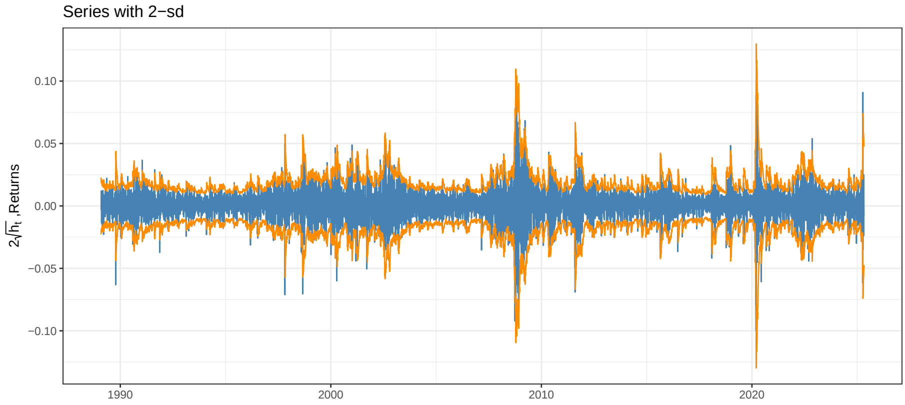
GARCH(1,1) Volatility
0.06
0.04
0.02
$\sqrt{h_t} = \sigma_t$
1970
1980
1990

```r
plot(fit.garch, which = 1)
```



```r
p.dat <- data.frame(Index = sp500$Index[-c(1:20)], ret = sp500$ret[-c(1:20)],
           ht.sqrt = fit.garch |> sigma() |> coredata())

p.dat |>
  ggplot(aes(x = Index, y = ret)) + geom_line(col = "steelblue") +
  geom_line(aes(y = 2*ht.sqrt), col = "darkorange") +
  geom_line(aes(y = -2*ht.sqrt), col = "darkorange") +
  labs(title = "Series with 2-sd", y = bquote(2*sqrt(h[t])~",Returns"), x = NULL)
```



Forecasting with GARCH Models

Example with a GARCH(1,1) model:

$$h_t = \omega + \alpha u_{t-1}^2 + \beta_1 h_{t-1}$$

with unconditional variance $\sigma^2 = \dfrac{\omega}{1-\alpha-\beta}$

write:

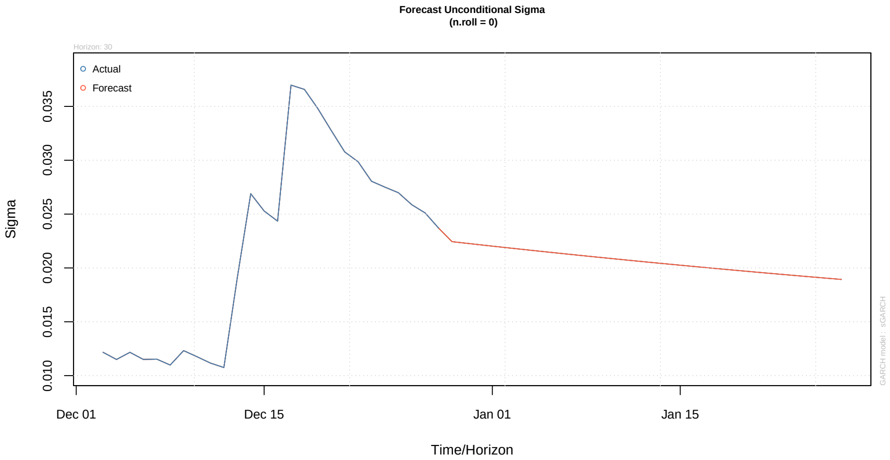
$$\begin{aligned} h_t &= \sigma^2 + \alpha\big(u_{t-1}^2 - \sigma^2\big) + \beta_1\big(h_{t-1} - \sigma^2\big) \\ h_{t+s} &= \sigma^2 + \alpha\big(u_{t+s-1}^2 - \sigma^2\big) + \beta_1\big(h_{t+s-1} - \sigma^2\big) \end{aligned}$$

Then the predicted h_{t+s} is:

$$h_{t+s} = \omega + (\alpha + \beta)\big(h_{t+s-1} - \sigma^2\big)$$

Forecasting with GARCH Models

```
fore.fit |> plot(which = 3)
```



Section 4

An Application

A Portfolio Allocation Problem

Assume that you are an investor who is interested in investing in a simple two asset portfolio.

- a risky asset (SP500) and
- a risk free asset such as the U.S. Tbills.

Based on your risk tolerance, you would like to target a 6% annualized volatility in your portfolio.

Portfolio Allocation Problem

- Step 1: Compute the Annualized volatility, `vol.annual`, as implied by the GARCH(1,1) model.
- Step 2: Compute the weights using the formula

$$w_{sp500} = \frac{\text{Target Volatility}}{\text{vol.annual}}$$

  Assume further that the sum of weights are capped at 1. We do not allow for short selling/ leveraging.

Step 2b: Plot the weights over time to show that the GARCH-based allocation is time-varying.

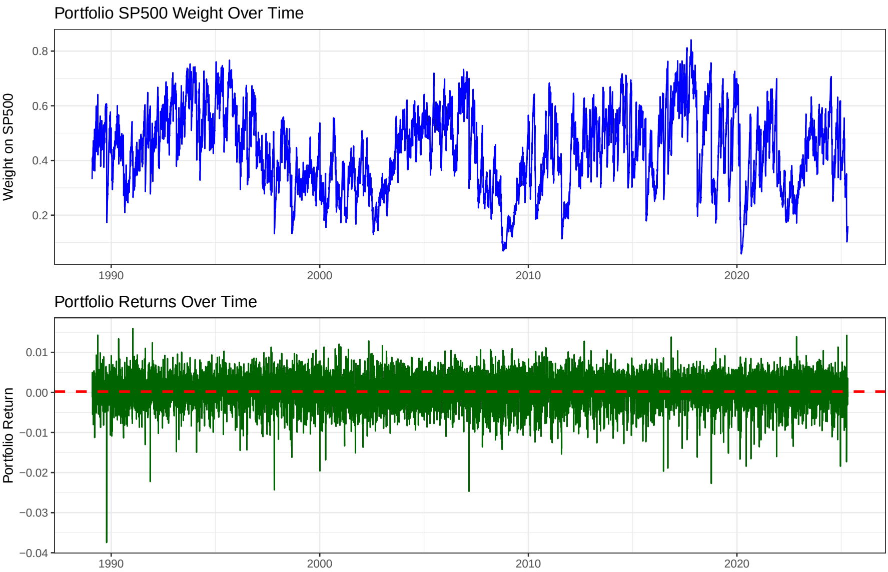
- Step 3: Simulate the portfolio returns using the weights from Step 2.
- Step 4: Plot the portfolio returns over time.

```r
# 1. Create a tibble with predicted volatility
port_data <- tibble(
  Date = sp500$Index[-c(1:20)],
  ret = sp500$ret[-c(1:20)],
  sigma = sigma(fit.garch)  # daily predicted vol.
) |>
  mutate(
sigma_annual = sigma * sqrt(252), # Annualized vol.
    target_vol = 0.06, # Target vol. 6%
    w_sp500 = pmin(1, target_vol / sigma_annual),  # Capped at 1
    rf_return = 0.05 / 252, # Daily risk-free rate
    port_ret = w_sp500 * ret + (1 - w_sp500) * rf_return  # Portfolio returns
  )
```

```r
# 2. Plot portfolio weights over time
aa <-  port_data |>
  ggplot(aes(x = Date, y = w_sp500)) +
  geom_line(color = "blue") +
  labs(title = "Portfolio SP500 Weight Over Time",
       y = "Weight on SP500", x = NULL)

# 3. Plot portfolio returns over time
bb <- port_data |>
  ggplot(aes(x = Date, y = port_ret)) +
  geom_line(color = "darkgreen") +
  geom_hline(yintercept = unique(port_data$rf_return),
             linetype = "dashed", color = "red",
             linewidth = 1) +
  labs(title = "Portfolio Returns Over Time",
       y = "Portfolio Return", x = NULL)
```

```
aa /bb
```



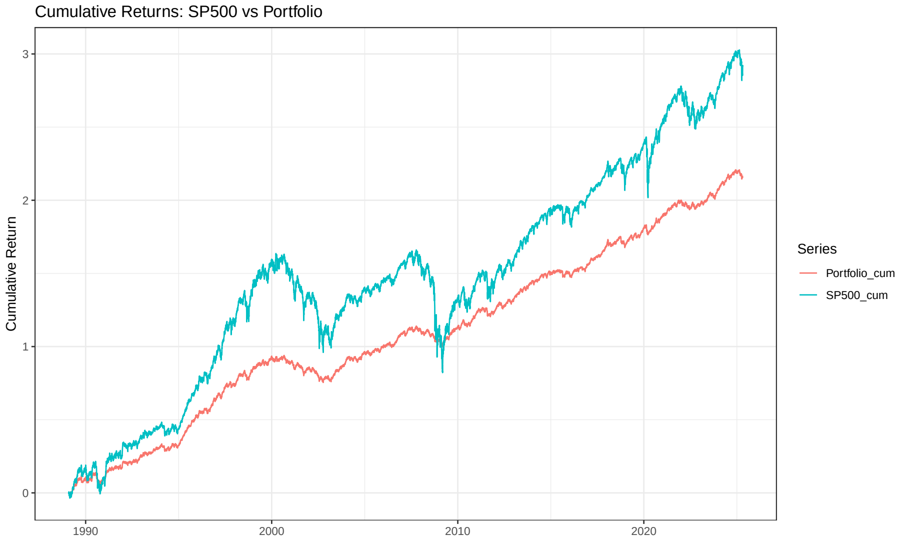
Portfolio SP500 Weight Over Time
Weight on SP500
0.8
0.6
0.4
0.2
1990
2000
2010
2020
Portfolio Returns Over Time
Portfolio Return
0.01
0.00
−0.01
−0.02
−0.03
−0.04
1990
2000
2010
2020

Cummulative Returns Comparison

```r
# Plot cumulative returns
port_data_long <- port_data |>
  mutate(SP500_cum = cumsum(ret),
         Portfolio_cum = cumsum(port_ret)) |>
  pivot_longer(cols = c(SP500_cum, Portfolio_cum),
               names_to = "Series", values_to = "Cumulative_Return")

PR <- ggplot(port_data_long, aes(x = Date, y = Cumulative_Return, color = Series))
  geom_line() +
  labs(title = "Cumulative Returns: SP500 vs Portfolio",
       y = "Cumulative Return", x = NULL)
```

PR



Cumulative Returns: SP500 vs Portfolio
Cumulative Return
3
2
1
0
1990
2000
2010
2020
Series
Portfolio_cum
SP500_cum

Performance Measures

```r
# Annualized mean return
AM <- mean(port_data$port_ret) * 252

# Annualized volatility
AV <- sd(port_data$port_ret) * sqrt(252)

# Annualized Sharpe ratio (assuming risk-free rate = 0)
sharpe <- AM/AV

cat("Annualized Mean Return: ", round(AM * 100, 2), "%\n",
    "Annualized Volatility: ", round(AV * 100, 2), "%\n",
    "Annualized Sharpe Ratio: ", round(sharpe, 2), "\n")
```

```
## Annualized Mean Return:  5.97 %
##  Annualized Volatility:  5.99 %
##  Annualized Sharpe Ratio:  1
```

Section 5

Additional Notes

Digression: Modeling conditional means and variances

Given the model

$$Y = \alpha + \beta\varepsilon$$

$\varepsilon \sim \mathbf{N}(0,1)$ then

$$\begin{aligned}\mathbf{E}(Y) &= \mathbf{E}\alpha + \beta\mathbf{E}(\varepsilon) \\ &= \alpha\end{aligned}$$

and

$$\begin{aligned}var(Y) &= var(\alpha) + \beta^2 var(\varepsilon) \\ &= \beta^2 \cdot 1\end{aligned}$$

To model the conditional mean of Y_t given

$$X_t = \{X_{jt}\}^{j=1,2,\ldots,k}$$

Model Building

Building a volatility model consists of four steps:

1. Specify a mean equation by testing for serial dependence in the data and, if necessary, building an econometric model (e.g., an ARMA model) for the return series to remove any linear dependence.
2. Use the residuals of the mean equation to test for ARCH effects.
3. Specify a volatility model if ARCH effects are statistically significant, and perform a joint estimation of the mean and volatility equations.
4. Check the fitted model carefully and refine it if necessary.