Series 4: non-parametric ARCH

Sylvain

8 April 2016

Outline

- Getting the data
- Modelling the data
- Specific exercise

Getting the data

Introduction

In this series we look at the log-returns of the BMW stock between June 1986 and March 1990. Unfortunately we do not have the original time stamps nor the stock price.

$$X_t = log(P_t/P_{t-1}),$$

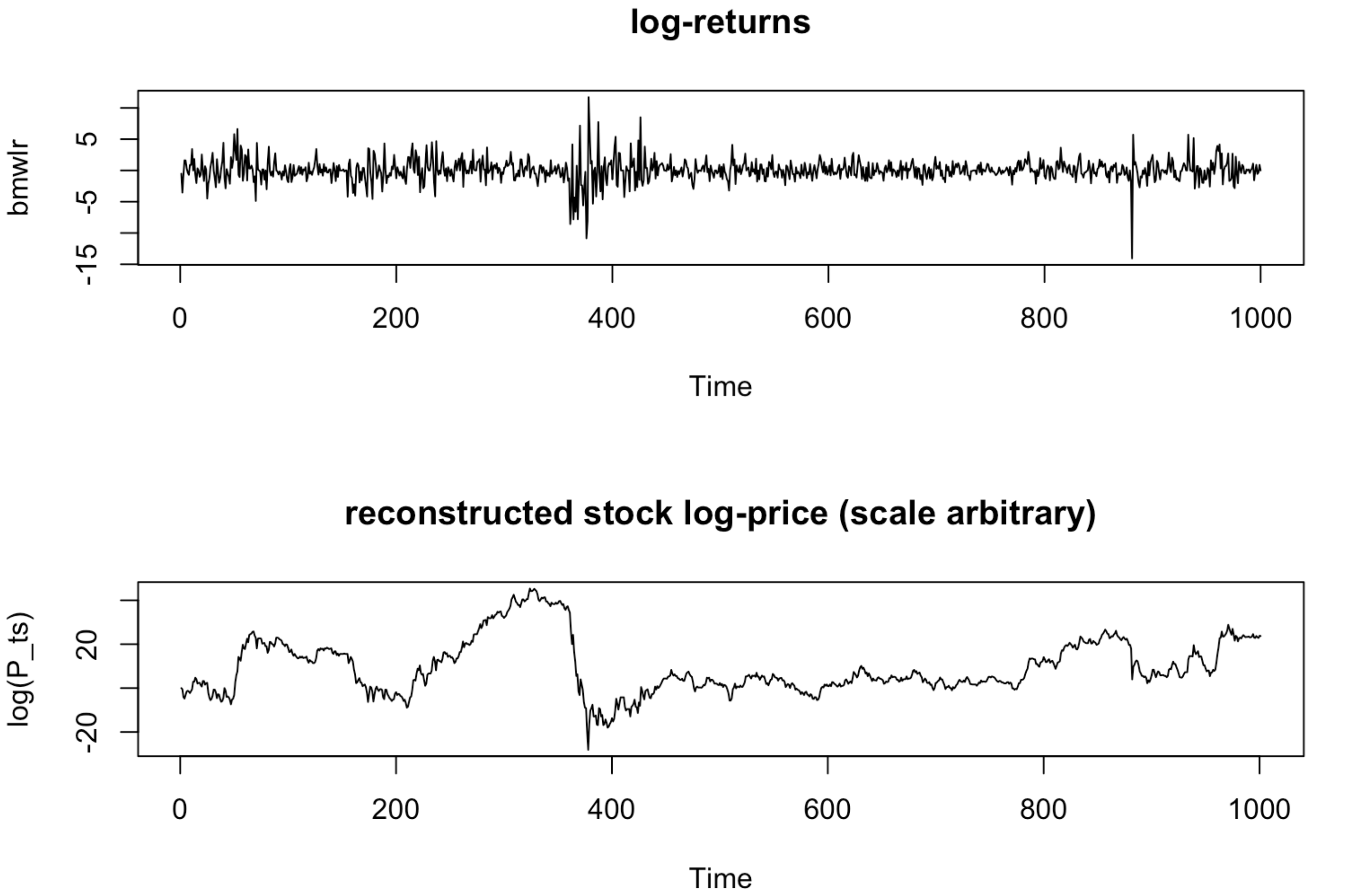
where P_t is the stock price. Given P_0, can reconstruct the original time series:

$$\begin{aligned} \frac{P_t}{P_{t-1}} &= exp(X_t) \\ P_t &= P_{t-1} exp(X_t). \end{aligned}$$

Load the data and try to reconstruct the price

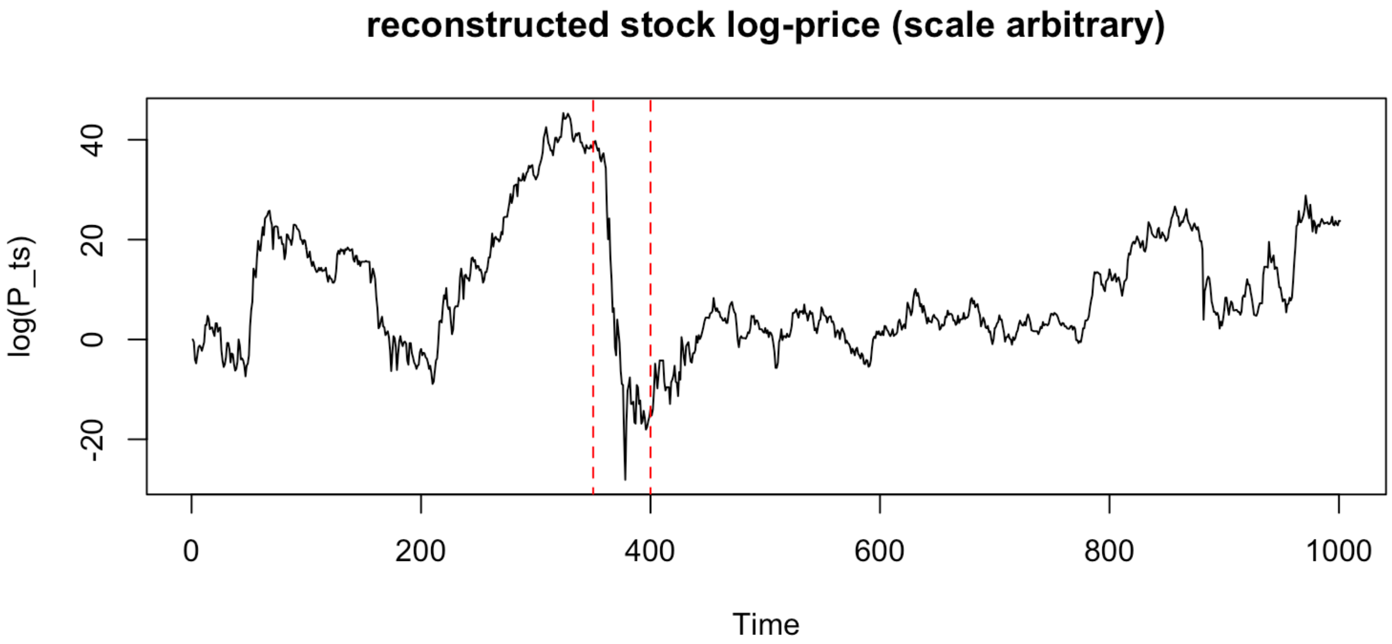
```r
bmwlr <- scan("http://stat.ethz.ch/Teaching/Datasets/bmw.dat")
## extract original price: we need a sequence P_t which is of length n+1 and
## with index starting at 0:
n <- length(bmwlr)
P_ts <- numeric(n + 1)
P_ts[1] <- 1
for (i in 2:(n + 1)) {
    P_ts[i] <- P_ts[i - 1] * exp(bmwlr[i - 1])
}
```

First look at the data



First look at the data

Any idea what happened then??



Critical thinking

Do you see any problem with the data? Is it possible to have such log-returns? the maximum observed value is 11.7, which would mean that $P_t = 122'900 * P_{t-1}$!! do you think we observe such things in reality? the data are actually , not . You can do two things:

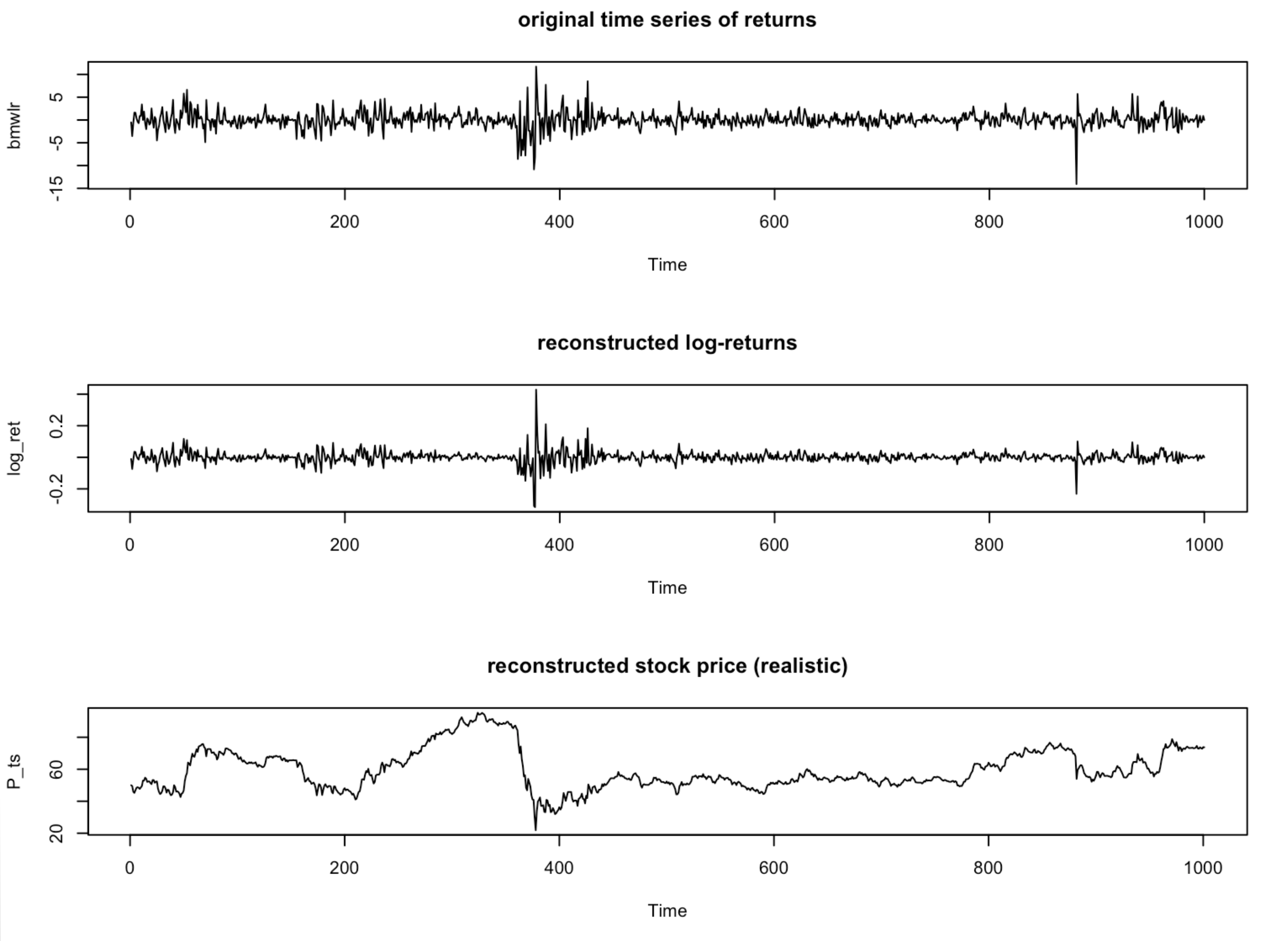
- Ignore this problem and just treat them as log-returns. It's not realistic but who cares?
- Try to reconstruct some plausible log-returns from the data (optional).

Reconstructed log-returns

if the data are returns, then given P_0 we can easily reconstruct the series as $P_t = P_t + x_t$. Let us assume that $P_0 = 50$ which seems like a reasonable value.

```
n <- length(bmwlr)
P_ts <- numeric(n + 1)
P_ts[1] <- 50
for (i in 2:(n + 1)) {
    P_ts[i] <- P_ts[i - 1] + bmwlr[i - 1]
}
log_ret <- log(P_ts[-1]/P_ts[-(n + 1)])
```

Reconstructed log-returns



Lessons

Always check the validity of your data. Ask yourself this type of questions:

- Is the range of data plausible?
- Should some of the data always be positive/negative?
- How are missing values encoded (often something like -999 or other monster)
- Any other feature that you can think of given the data at hand. Never take the data as simple x and y without a meaning!

Modelling the data

ARCH and GARCH

We want a model which explains that the variance is not constant in time. A typical choice is a ARCH model:

$$
\begin{aligned}
X_t &= \sigma_t \epsilon_t, \quad \text{where } E(\epsilon_t) = 0, \mathrm{Var}(\epsilon_t) = 1, \\
\sigma_t^2 &= \nu(X_{t-1}).
\end{aligned}
$$

ν is called the volatility-function and is often modelled parametrically, however in this series we try to do it non-parametrically. In a GARCH model the variance is also autocorrelated as:

$$\sigma_t^2 = \nu(X_{t-1}, \sigma_{t-1}^2).$$

Non-parametric ARCH

Here we want to model ν non-parametrically. We restrict ourselves to a ARCH model because a GARCH model requires an iterative procedure where the unobserved σ_t process has to be estimated. Notice that we can rewrite the model as follows:

$$Y_t = X_t^2 = \nu(X_{t-1}) + \eta_t, \quad \text{where } \eta_t = \nu(X_{t-1})(\epsilon_t^2 - 1).$$

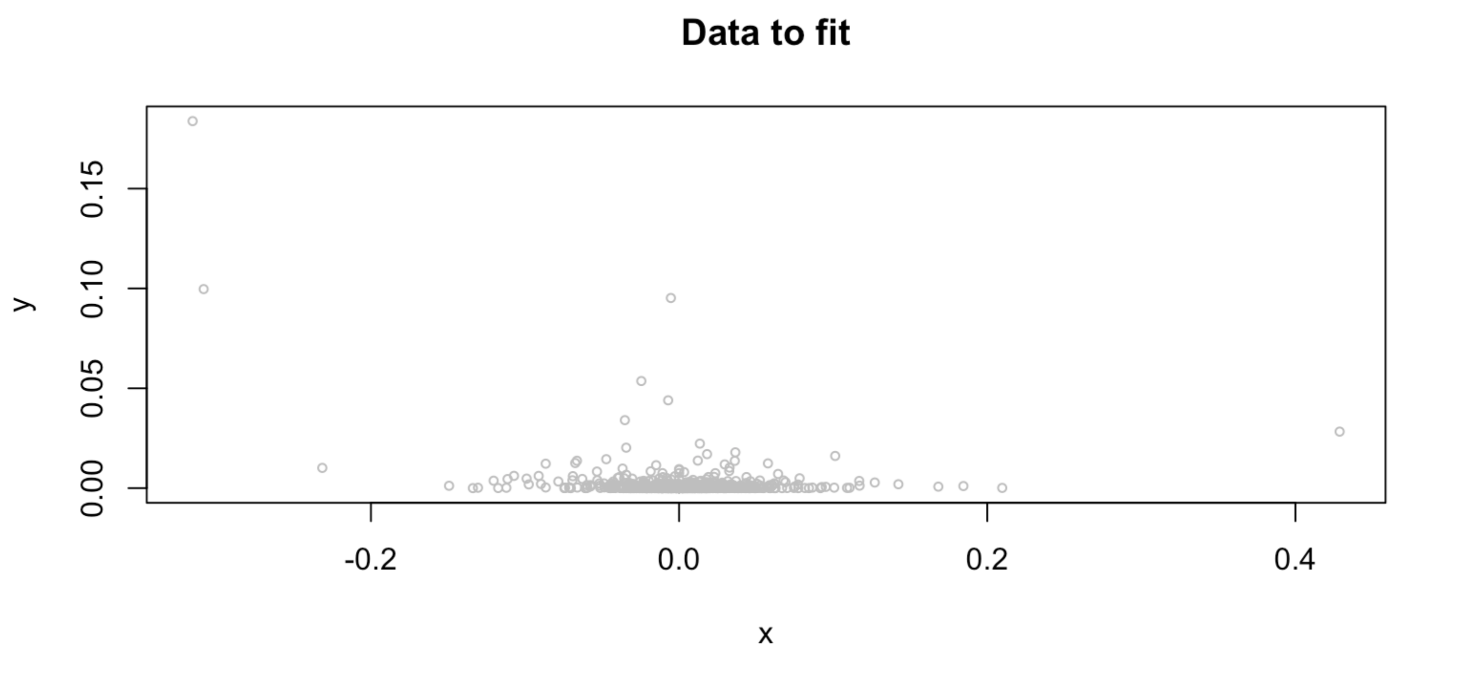
Because $E(\eta_t) = 0$ we can try to estimate ν by regressing Y_t on X_t using our favorite estimator.

Remark: You can try to show that $E(\eta_t) = 0$ in task e), but only if you feel like it.

Second look at the data to fit

Let's construct the data for the regression and see what we are trying to do:

```
x <- bmwlr[-length(bmwlr)]  # last value of xt cannot be used for x
y <- bmwlr[-1]^2  # first value of xt cannot be used for y
```



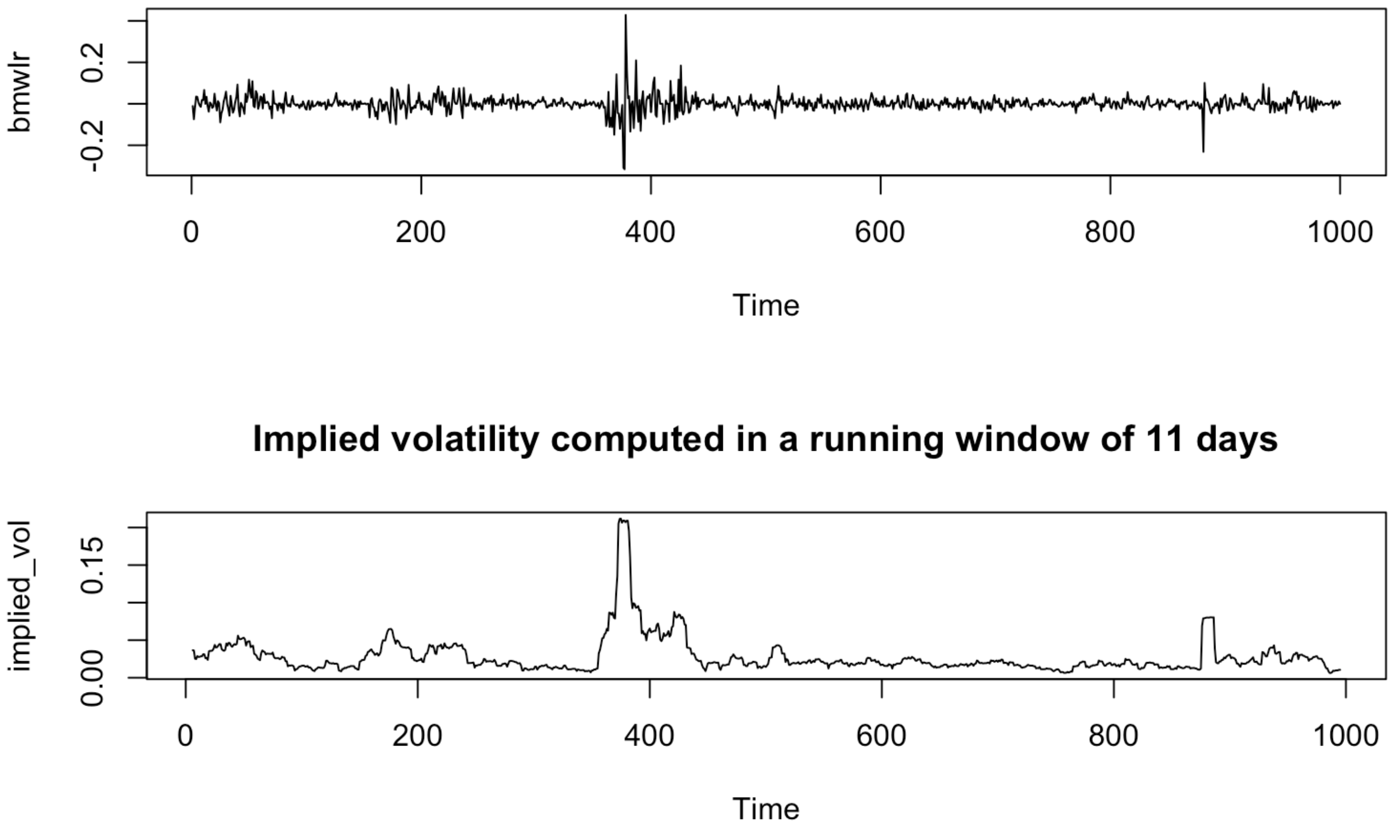
Implied volatility

The volatility is different at every time step so we can't estimate it directly. But we can try to estimate the implied volatility by using a running window.

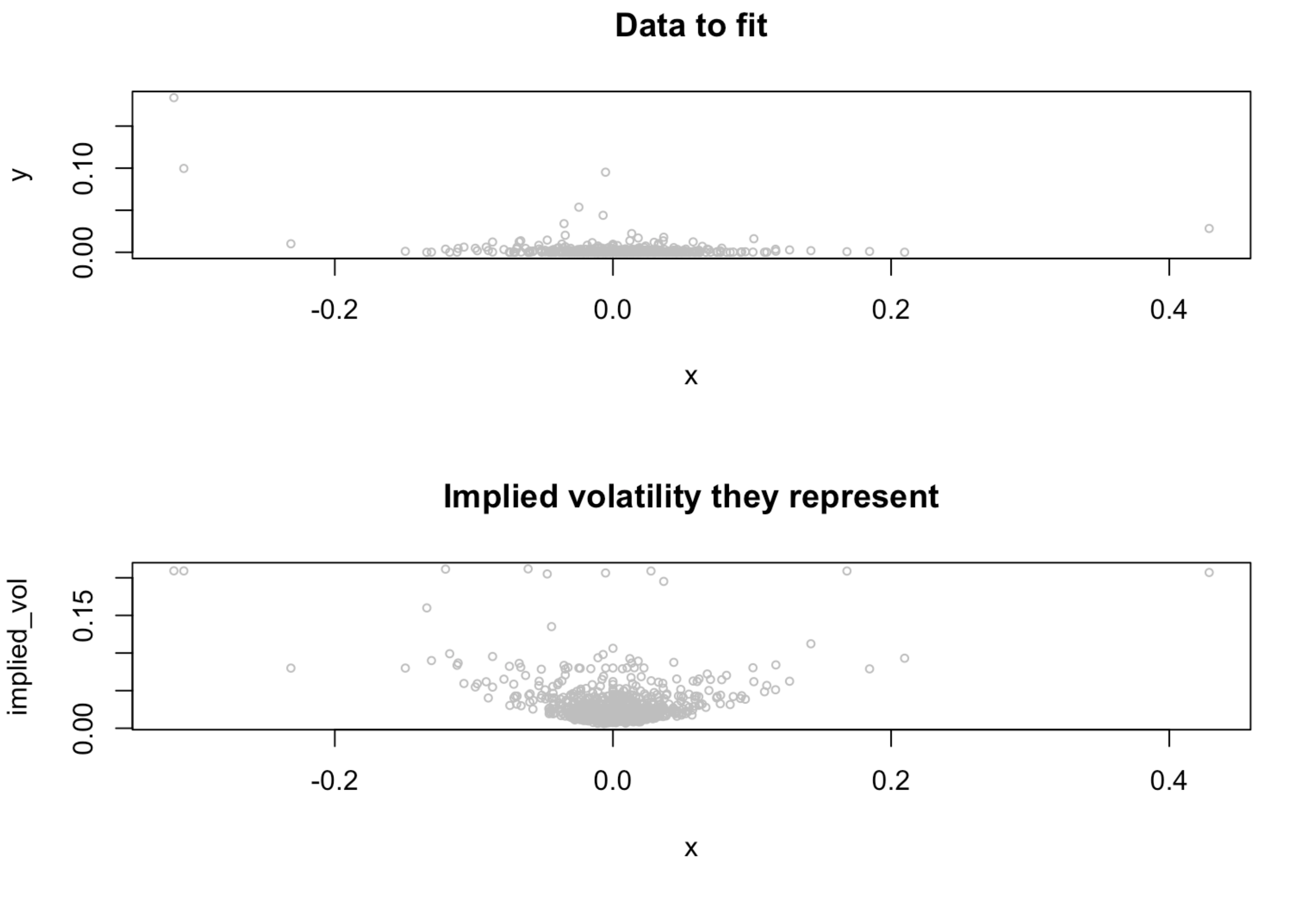
```
l <- 5  #window size = 2*l+1
implied_vol <- numeric(n - 2 * l)
vol_indices <- (l + 1):(n - l)
for (i in vol_indices) {
    local_indices <- (i - l):(i + l)
    implied_vol[i - l] <- sd(bmwlr[local_indices])
}
```

Remark: with a time window like this we assume that the volatility is in time, but with our ARCH model we do not capture that (no dependence of σ_t on σ_{t-1})

Implied volatility (2)



Implied volatility (3)



The exercise

Task a)

Remember: zero correlation does not imply independence!

For example with our data, $Cov(X_t, X_{t-1}) = 0$, but $Cov(X_t^2, X_{t-1}^2) \neq 0$!

Task: Check this by using the function `acf()` which compute and plot the autocorrelation of a time series at all possible lags.

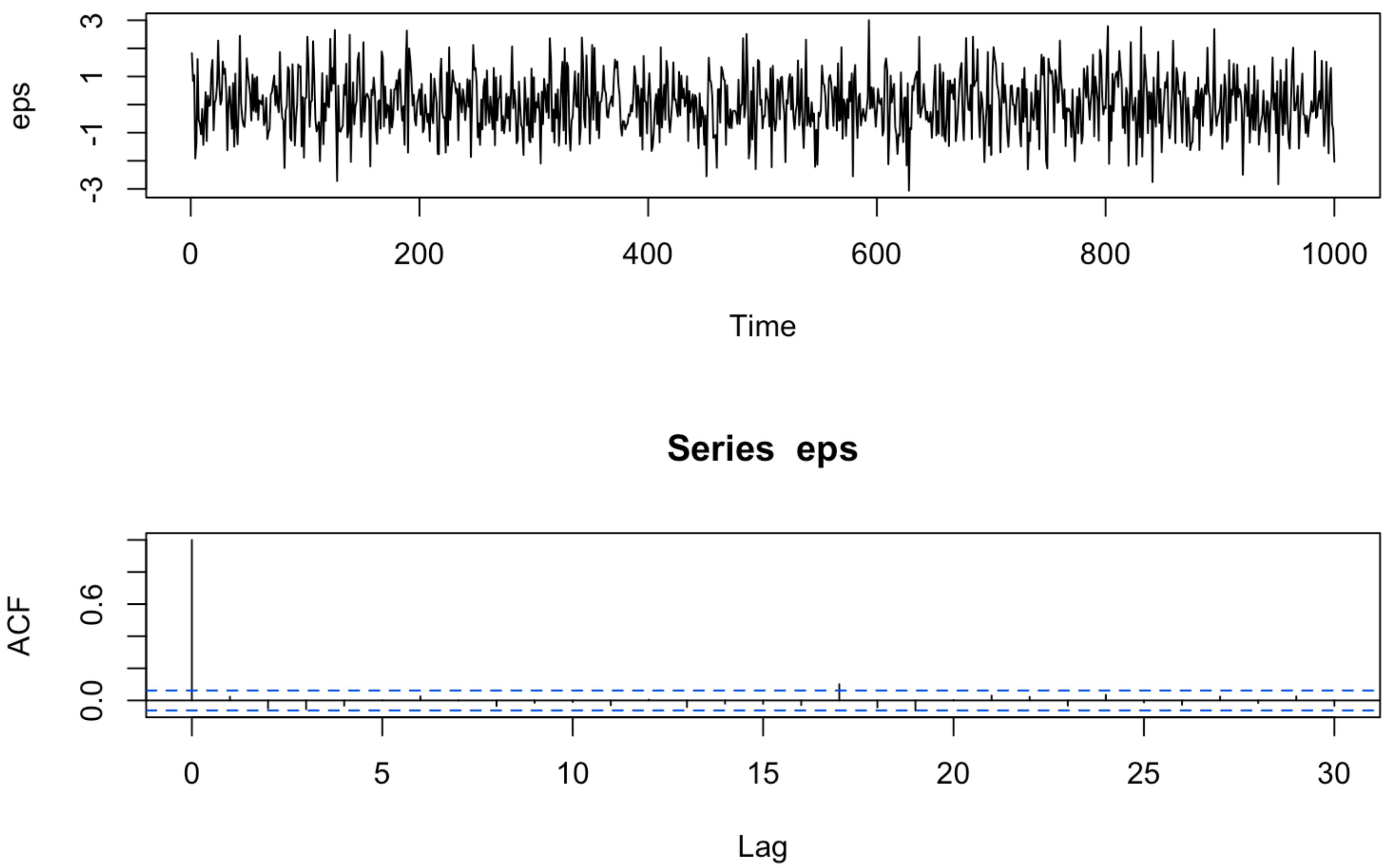
ACF

The ACF is the autocorrelation function of a random variable. It computes the autocorrelation between times separated by a given lag. At lag zero the autocorrelation is always 1 (correlation of X with itself). Let's say we generate the following data:

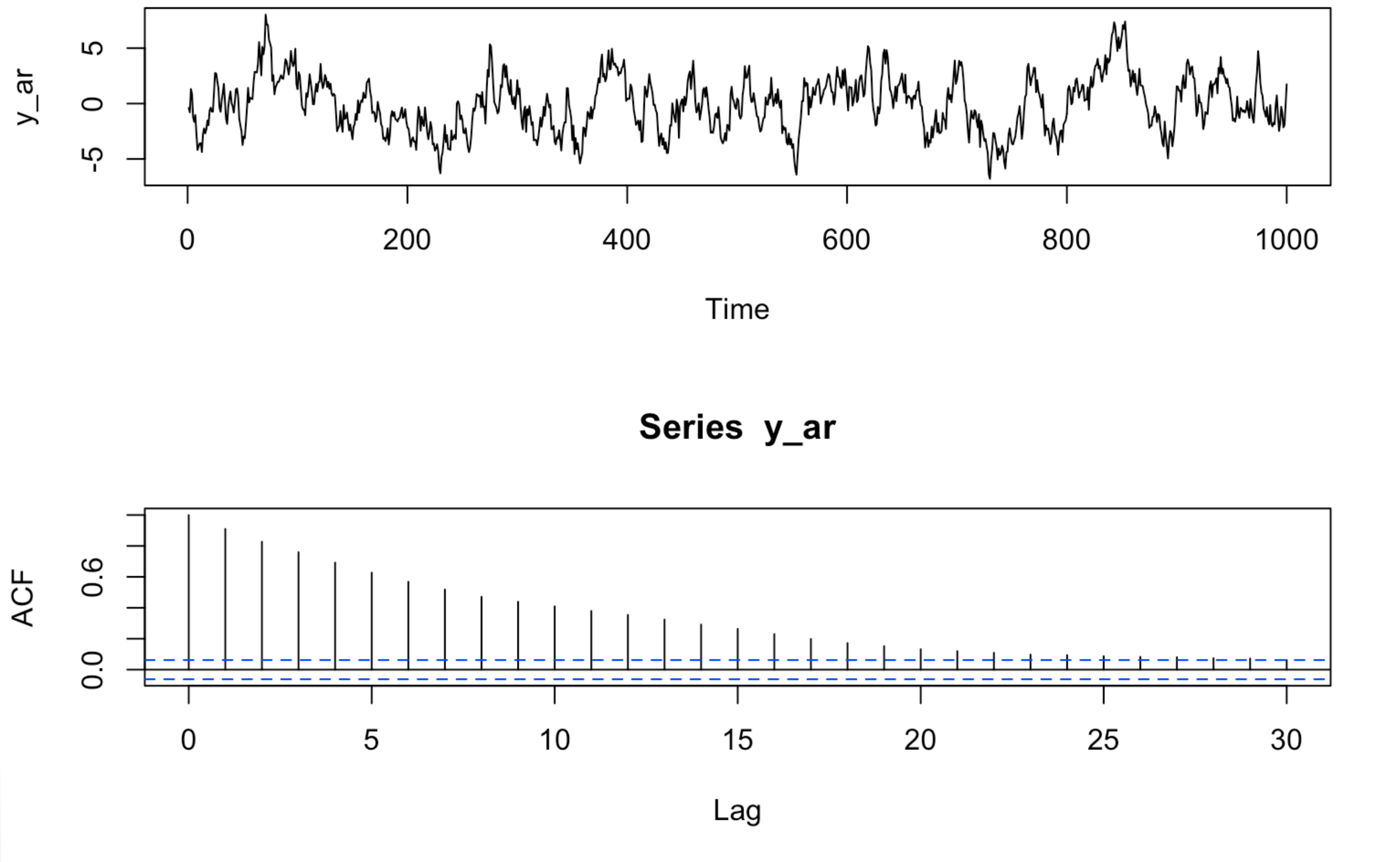
```
## white noise = no autocorrelation
eps <- rnorm(1000)

## AR(1) process: y_t = alpha * y_t-1 + eps_t
y_ar <- arima.sim(list(ar = 0.9), 1000)
```

ACF: white noise



ACF: autocorrelated noise



Task b)

Estimate the volatility function ν by fitting the data (X_t, Y_t). For the sake of the exercise we try different smoothers that you learned in class: local polynomial (with `loess`), smoothing splines(with `smooth.spline`) and Nadaraya-Watson (with `ksmooth`).

Remark: In principle you should choose the smoothness parameter with cross-validation or other objective criterion. For the moment just try to explore and play with it.

R-tips: Each function has its own peculiarities, which is something you have to get used to. Read the documentation (type `?foo`) and try to understand. For generic methods related to a kind of object you can find the help by doing `?method.object`, for example `?predict.loess`.

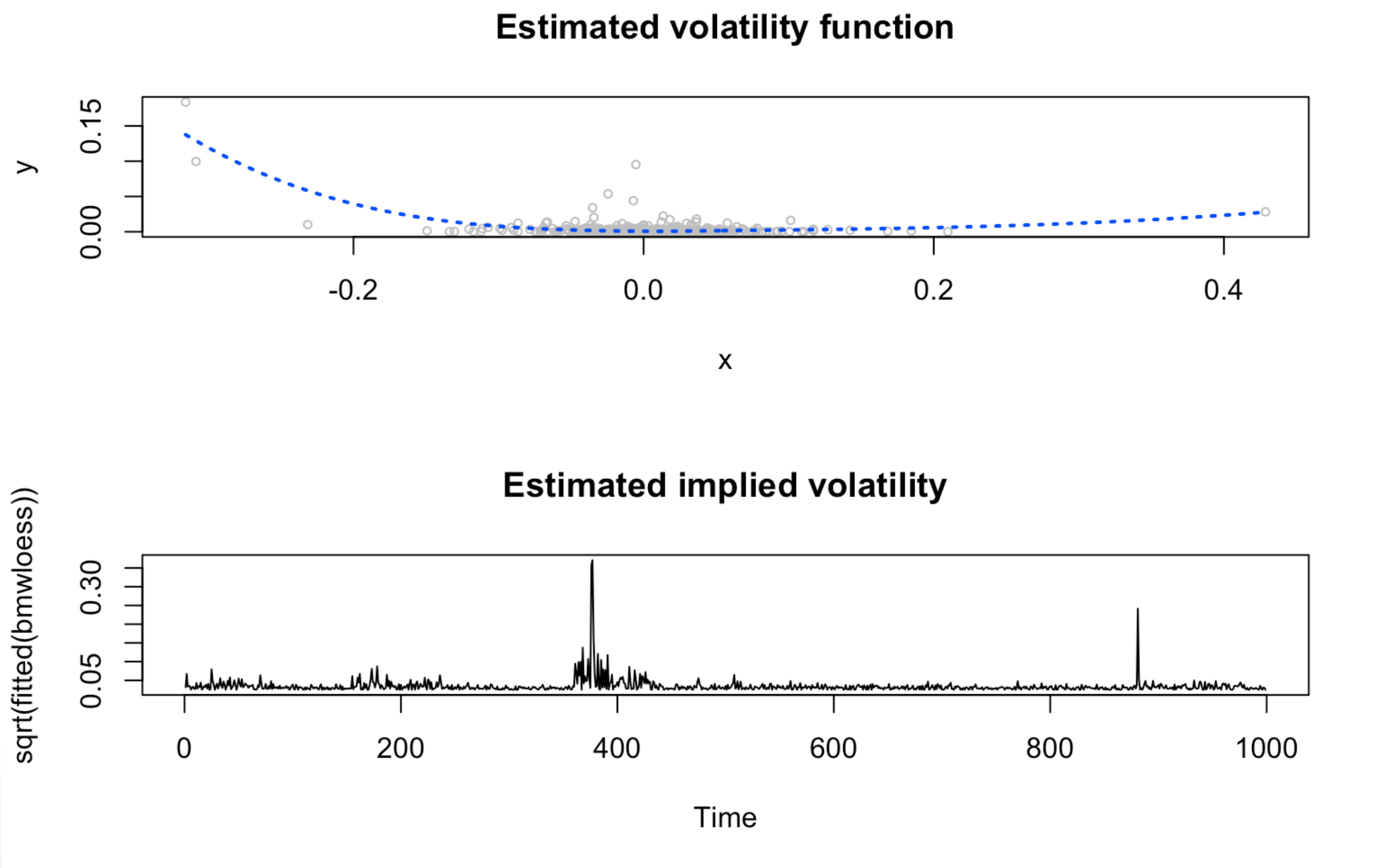
Interpretation

Try be critical: what we are trying to do in the present exercise is a bit adventurous… Ask yourself if the function that you estimate makes sense. Plot the resulting estimated stochastic volatility as a function of X_{t-1} and as a function of time.

What about the residuals?

Interpretation (2)

You should obtain something like that:



Interpretation (3)

Some questions you could ask yourself:

- Is the amount of smoothness reasonable?
- Is the estimated function meaningful (think of the application)?
- Can I interpret some of its most striking features (general shape, symmetry, etc)?
- How could I improve the model?

Task c)

Here the goal is to use yet another smoothing function on the same data. The most interesting is to try `lokerns` which compute a locally adaptive bandwidth.

The idea is that you can use a small bandwidth in area of high density and fall back on a larger bandwidth when the data are sparse.

In practice things are not that easy. Non-parametric regression is always a tricky business: be critical with the results you obtain with any kind of automated method.

Take home message

- Be critical of the data you are given.
- Be critical of the model you choose for the data.
- Be critical of the estimates you get out of your procedure.
- Have fun!