Series 4

1. The dataset `bmw` is a time series of log returns of the BMW stock (business-daily, between June 1986 and March 1990). The log return is defined as follows:

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

where P_t is the stock price at time t. Log returns can be modelled by

$$X_t = \sigma_t \epsilon_t, \text{ where } \mathbf{E}[\epsilon_t] = 0, \mathrm{Var}(\epsilon_t) = 1, \quad (1)$$

ϵ_t independent of $\{X_s;\ s < t\}$, $\sigma_t^2 = v(X_{t-1})$, where $v: \mathbb{R} \mapsto \mathbb{R}^+$ is the so-called "volatility function". Thus, X_t depends on $\{X_s;\ s < t\}$ only through X_{t-1} (Markov-property).

The model can be fitted by nonparametric regression of the function v in

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

is treated as error term. In task e) you will prove that $\mathbf{E}[\eta_t] = 0$.
Note: Other usual model assumptions on errors, such as independence, are not fulfilled by η_t, but with some effort (don't try!) it can be shown that v can be optimally estimated by the same estimation methods as if the η_t were independent errors.

a) Model (1) is often chosen for this kind of data because it leads to observations that are not autocorrelated[1], but dependent. Dependency can be verified by showing that under the model, $\mathrm{Cov}\left(X_t^2, X_{t-h}^2\right) \neq 0,\ h > 0$ (complicated). Plot and interpret the autocorrelation functions of X_t and X_t^2 for the BMW-dataset.
The data can be read into R by

```
> bmwlr <- scan("http://stat.ethz.ch/Teaching/Datasets/bmw.dat")
```

`bmwlr` should be a vector of 1000 observations.
R-hint: Function `acf`. For example, "autocorrelation of lag 1" (in the plot, with 1000 observations, indicated as lag 1 out of 999) means correlation between X_t and X_{t-1}. The plot shows also an acceptance region (at 5%-significance level) for testing the null hypothesis of uncorrelated observations.

b) Fit the data using the nonparametric regression methods Nadaraya-Watson, local polynomial and smoothing splines for the regression function v.
Comment on the results and compare the fits obtained using the mentioned nonparametric estimators. Look at the estimated volatility function as a function of X_t and at the estimated implied volatility as a function of time.
R-hint: Use `loess` for local polynomial, `smooth.spline` for smoothing splines and `ksmooth` for Nadaraya-Watson kernel regression.
The methods have no consistent way to choose the *degree of smoothness*, but you will learn soon how to do it with cross-validation. For the present series we give you the parameters to use. For `loess` the smoothness is defined by the parameter `span`, which indicates the fraction of data to include in support of the kernel (expressed as a number between 0 and 1). For the exercise you can use the default value of 0.75, but try to play with it and see how it influences the results.
For `smooth.spline` you can define the smoothness in terms of equivalent degrees of freedom (edf), which can be computed as the trace of the hat matrix (see lecture notes for more details). Again you can play with different values, but for the sake of comparison you can use the same edf as in `loess`, which can be recovered from the output as `fit$trace.hat`.
For `ksmooth` the smoothness is defined in term of the bandwidth. Try to play different values and see how it influences the result. To ensure that you have the same smoothness as for the other method, you need to numerically search for the bandwidth value that results in the same edf. To save you this trouble, here is the answer: use `h=3.54`.
Remark: `ksmooth` internally reorder its x input in increasing order, so you will lose the time ordering. To recover it, you have to do the following:

[1] "Autocorrelated" refers to correlation over time, i.e., correlation between X_t and $X_{t-h},\ h > 0$.

```
ox <- order(x)
fit <- ksmooth(x,y,...)
fit$x <- fit$x[order(ox)]
fit$y <- fit$y[order(ox)]
```

Check model assumptions, but don't spend too much time on this since the structure of the data is pretty unclear. Note that for computing residuals it is necessary to know the fitted values at the data points. For `ksmooth` they are provided via argument `x.points` and for `loess` and `smooth.spline` via `fitted()`.

c) Fit the data using the functions `glkerns` (kernel regression with global optimal bandwidth) and `lokerns` (kernel regression with local optimal bandwidth) of the R package `lokern`. Compare the fits. Plot the local bandwidths from `lokerns` and compare them to the global bandwidth of the function `glkerns`. How does the local bandwith relate to the density of the data?

Remark: It is not so easy to control how the function optimizes the bandwidth internally and can easily lead to misleading results. In the present case be careful to pass the argument `is.rand=FALSE` to specifiy that the design is fixed and `hetero=FALSE`. Even though the errors are heteroscedastic, this would have to be modelled explicitely and cannot be done automatically by the function.

d) **(optional)** Compute $\mathbf{E}[X_t|X_{t-1}, X_{t-2}, \ldots]$, $\mathrm{Var}(X_t|X_{t-1}, X_{t-2}, \ldots)$, $\mathrm{Cov}(X_t, X_{t-h})$, $h > 0$.

Background about conditional expectations:

For two (possibly multi-dimensional) random variables X and Y, the conditional distribution $P_{Y|X=x}$ can be uniquely defined for $P-$almost all values of X. Define $h(x) = \mathbf{E}[Y|X = x]$ as the expectation of Y under the conditional distribution $P_{Y|X=x}$. For the random variable X, $h(X) = \mathbf{E}[Y|X]$ is a random variable. Here is a very useful equation (the so-called tower property), which will be needed for parts a and b:

$$\mathbf{E}[Y] = \mathbf{E}[\mathbf{E}[Y|X]], \tag{2}$$

the outer expectation taken over the distribution of X. Conditional variances and covariances are defined analogously.

e) **(optional)** Show $\mathbf{E}[\eta_t] = 0$.

Preliminary discussion: Friday, April 08.

Deadline: Friday, April 15.