Part 1:
Financial returns and normality

In mathematical finance, it is often assumed that the dynamics of the price of stocks is driven by a geometric Brownian motion. The celebrated Black-Scholes formula, which models the price of a stock using various assumptions, critically relies on a setting under which the financial returns are independent and the logarithm of the returns are Normally distributed. In this exercise, we would like to check whether these assumptions are satisfied.

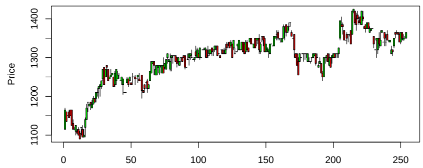

Figure 1: Tick chart of the stock price of Romande Energie in 2021

We use the `QRM` package in `R` and, as a proxy for stock prices, we use aggregated versions of them through four stock indices: SP500 (`sp500` in the `QRM` library), CAC40 (`cac40` in the `QRM` library), Nasdaq (`nasdaq` in the `QRM` library) and NIKKEI (`nikkei` in `QRM` library).

(a) Read in the data. Then, assess the stationarity of the (raw) stock indices.

(b) Create a function to transform the daily stock indices into their daily negative log returns counterparts. Plot the latter series and assess their stationarity. To compare the series, also plot the negative log returns on a common scale to all indices.

(c) Draw histograms of the negative log returns and compare them to the Normal distribution. What do you observe?

(d) Check the normality assumption of the negative log returns using QQ-plots. What is your conclusion?

(e) Formally test the normality assumption of the negative log returns using an Anderson-Darling testing procedure. Do you reject the Normal hypothesis?

(f) Use the `fitdistr()` function from the `MASS` package in order to obtain the (maximum-likelihood estimated) parameters of distributions you could imagine for the negative log returns. Try to fit at least two different distributions on the data and, using an information criteria (such as the AIC), decide which distributional framework fits best for each of the series.

(g) If this has not been done in (f), fit a t-distribution to the negative log returns using `fitdistr()`. Using a QQ-plot for each of the series, decide whether the fit is better than with a Normal distribution, based on your answer in (d).

Part 2:
Financial time series, volatility and the random walk hypothesis

Another crucial hypothesis in asset pricing is the so-called homoscedasticity. In the Black-Scholes formula, this comes from the assumption that the geometric Brownian motion modelling the price of a stock has a constant volatility. We would also like to check this assumption.

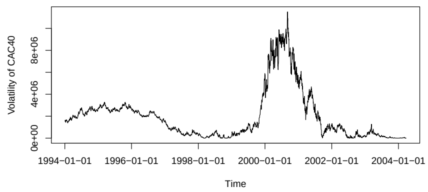

Figure 2: Volatility of the CAC40 index from 1994 to 2004

We use the same data as in Part 1, i.e. the four stock indices.

(a) Plot the ACF of all the series in Part 1 (i.e. the raw series as well as the negative log returns). What do you observe?

(b) Use a Ljung-Box procedure to formally test for (temporal) serial dependence in the series. What is your conclusion?

(c) Propose ARIMA models for each of the negative log returns series, based on visualisation tools (e.g. ACF and PACF). Select an ARIMA model using `auto.arima()` (`forecast` package) to each of the negative log returns series. Comment on the difference. Assess the residuals of the resulting models.

(d) Assess the residuals of the resulting models from (c), both their raw values and their absolute values, through visual tools (such as the ACF) and formal tests (e.g. Ljung-Box). What do you conclude about the independence assumption?

(e) Plot the volatility of the raw series of indices. What is your conclusion on the homoscedasticity assumption?

(f) Fit GARCH models to the negative log returns of each series with both standardised and skewed t-distributions, with order $(1,1)$, using the `garchFit()` function from the `fGarch` library. Assess the quality of the fit by evaluating the residuals.

(g) Residual serial correlation can be present when fitting a GARCH directly on the negative log returns. Hence, in order to circumvent this problem, it is possible to use the following two-step approach:

- fit an ARMA(p,q) on the negative log returns;
- fit a GARCH(1,1) on the residuals of the ARMA(p,q) fit.

Proceed with the above recipe. Assess the quality of the above fit.

(h) Use the `garchAuto.R` script in order to fit a GARCH on the residuals of the ARMA(p,q) from (g). Assess the quality of the fit.