02 Stationary time series (Part II)

Andrius Buteikis, andrius.buteikis@mif.vu.lt
http://web.vu.lt/mif/a.buteikis/

Forecasting

So far we thought of the information set as containing the available past history of the series, $\Omega_T = \{Y_T, Y_{T-1}, ...\}$, where we imagined the history as having begun in the infinite past. Based upon that information set, we want to find the optimal forecast of Y at some future time $T + h$.

If Y_t is a stationary process, then the forecast tends to the process mean as h increases. Therefore, the forecast is only interesting for several *small* values of h.

Our forecast method is always the same: write out the process for the future time period, $T + h$ and project it on what is known at time T when the forecast is made. We denote the forecast as $Y_{T+h|T}, \quad h \geq 1$.

Point forecasts can be calculated using the following three steps.

1. If needed, expand the equation so that Y_t is on the left hand side and all other terms are on the right;
2. Rewrite the equation by replacing T by $T + h$;
3. On the right hand side of the equation, replace future observations by their forecasts, future errors ($\epsilon_{T+j}, 0 < j \leq h$) by zero, and past errors by the corresponding residuals.

Forecasting MA(q) process

Consider, for example, an MA(1) process:

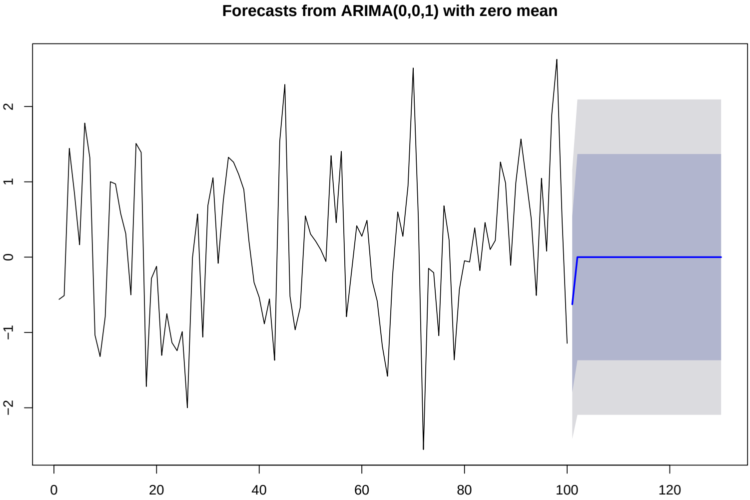
$$Y_t = \mu + \epsilon_t + \theta\epsilon_{t-1}, \quad \epsilon_t \sim WN(0, \sigma^2)$$

We have:

$$\begin{aligned} Y_{T+1} = \mu + \epsilon_{T+1} + \theta\epsilon_T &\Rightarrow Y_{T+1|T} = \mu + 0 + \theta\epsilon_T \\ Y_{T+2} = \mu + \epsilon_{T+2} + \theta\epsilon_{T+1} &\Rightarrow Y_{T+2|T} = \mu + 0 + 0 \\ &\quad \dots \\ &\quad Y_{T+h|T} = \mu \end{aligned}$$

The forecast quickly approaches the (sample) mean of the process and starting at $h = q + 1$ - coincides with it. When h increases, the accuracy of the forecast diminishes up to the moment $h = q + 1$, whereupon it becomes constant.

An example of an MA(1) process: $Y_t = \epsilon_t + 0.5\epsilon_{t-1}$:



Forecasting AR(p) process

Consider, for example, an AR(1) process:

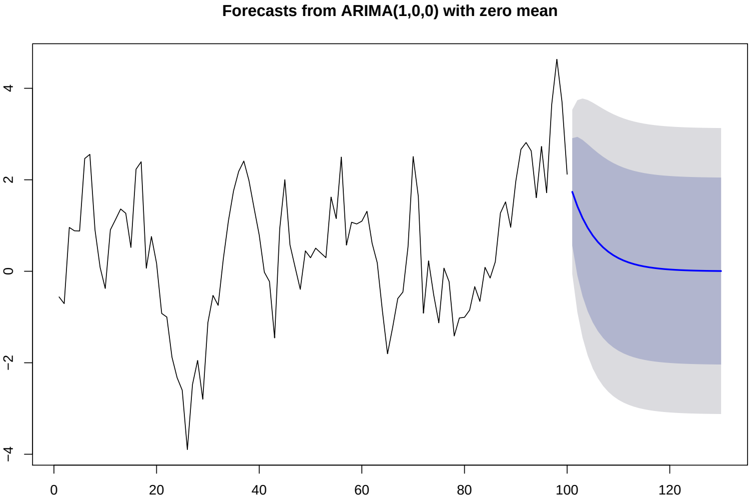
$$Y_t = \phi Y_{t-1} + \epsilon_t, \quad \epsilon_t \sim WN(0, \sigma^2)$$

We have:

$$Y_{T+1} = \phi Y_T + \epsilon_{T+1} \Rightarrow Y_{T+1|T} = \phi Y_T + 0$$
$$Y_{T+2} = \phi Y_{T+1} + \epsilon_{T+2} \Rightarrow Y_{T+2|T} = \phi Y_{T+1} + 0 = \phi^2 Y_T$$
$$\dots$$
$$Y_{T+h|T} = \phi^h Y_T$$

The forecast tends to the (sample) mean exponentially fast, but never reaches it. When h increases, the accuracy of the forecast diminishes but never reaches the limit.

An example of an AR(1) process: $Y_t = 0.85Y_{t-1} + \epsilon_t$:



Forecasting ARMA(p,q) process

Consider, for example, an ARMA(1,1) process:

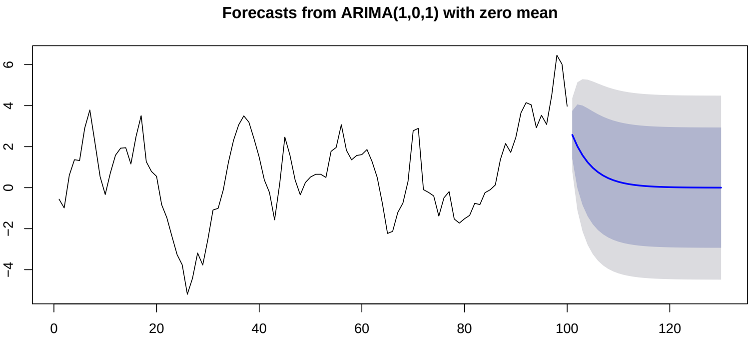
$$Y_t = \phi Y_{t-1} + \epsilon_t + \theta \epsilon_{t-1}, \quad \epsilon_t \sim WN(0, \sigma^2)$$

We have:

$$\begin{aligned} Y_{T+1} = \phi Y_T + \epsilon_{T+1} + \theta \epsilon_T &\Rightarrow Y_{T+1|T} = \phi Y_T + 0 + \theta \epsilon_T \\ Y_{T+2} = \phi Y_{T+1} + \epsilon_{T+2} + \theta \epsilon_{T+1} &\Rightarrow Y_{T+2|T} = \phi Y_{T+1} + 0 + 0 = \phi^2 Y_T + \phi \theta \epsilon_T \\ &\dots \\ &Y_{T+h|T} = \phi^h Y_T + \phi^{h-1} \theta \epsilon_t \end{aligned}$$

Similar to the AR(p) process, the ARMA(p,q) process tends to the average, but never reaches it.

An example of an ARMA(1,1) process: $Y_t = 0.85Y_{t-1} + \epsilon_t + 0.5\epsilon_{t-1}$:



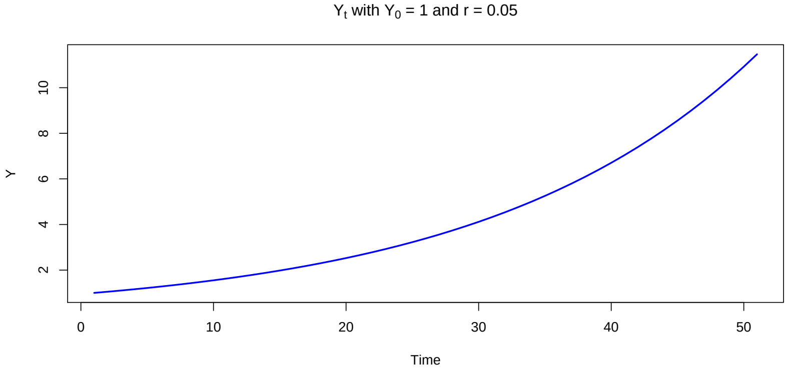
- The forecast $Y_{T+h|T}$ of an MA(q) process in $h = q$ steps reaches its average and then does not change anymore;
- The forecast $Y_{T+h|T}$ of an AR(p) or ARMA(p,q) process tends to the average, but never reaches it. The speed of convergence depends on the coefficients;

Financial Volatility

Consider Y_t growing annually at rate r:

$$Y_t = (1+r)Y_{t-1} = (1+r)^2 Y_{t-2} = ... = (1+r)^t Y_0 = e^{t \cdot log(1+r)} Y_0$$

The values of Y_t lie on an exponent:

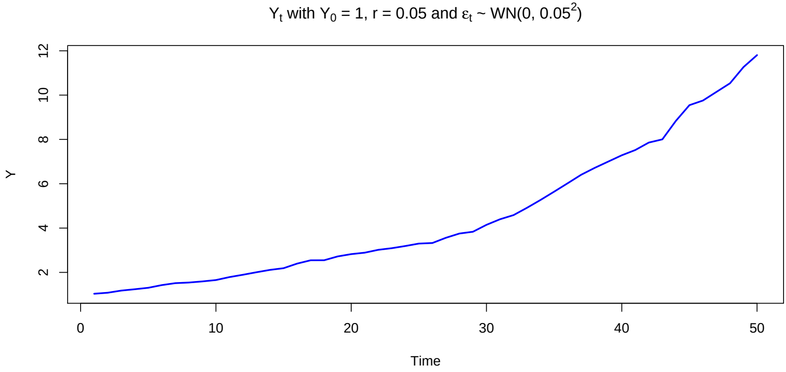


In order for the model to represent a more realistic growth, let us introduce an economic shock component, $\epsilon_t \sim WN(0, \sigma^2)$.

Thus, our model is now:

$$Y_t = (1 + r + \epsilon_t)Y_{t-1} = \Pi_{s=1}^{t}(1 + r + \epsilon_s) \cdot Y_0 = e^{\sum_{s=1}^{t} log(1+r+\epsilon_s)} \cdot Y_0$$

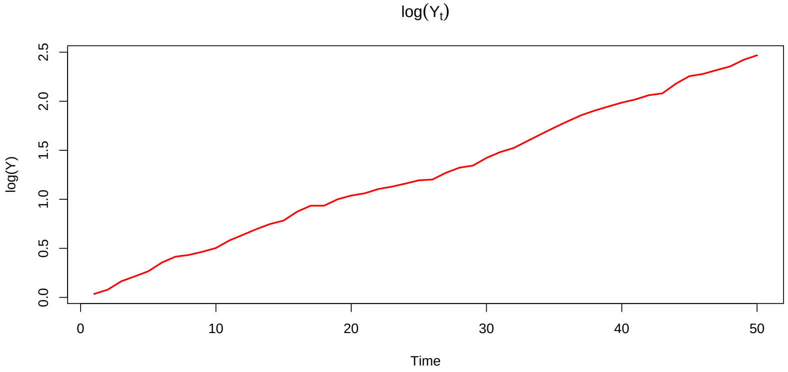
The values of Y_t are again close to the exponent:



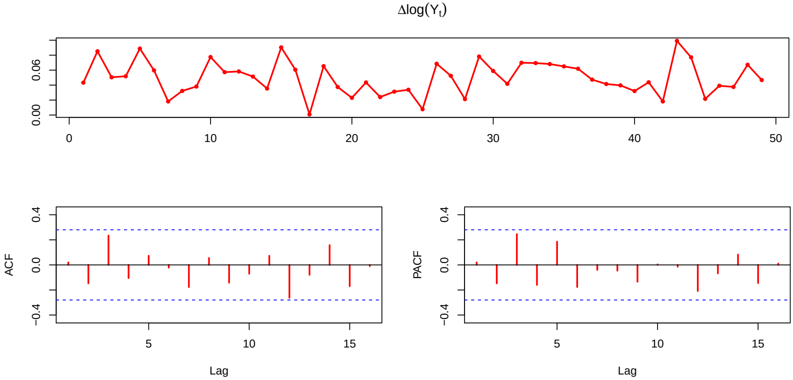
Note: $\mathbb{E}Y_t = e^{t \cdot log(1+r)}Y_0$, thus Y_t is not stationary.

We can take the differences: $\Delta Y_t = Y_t - Y_{t-1}$ but they are also not stationary. We can also take the logarithms and use the equality $log(1 + x) \approx x$ (using Taylor's expansions of a function around 0):

$$\tilde{Y}_t = log Y_t = log Y_0 + \sum_{x=1}^{t} log(1 + r + \epsilon_s) \approx log Y_0 + rt + \sum_{s=1}^{t} \epsilon_s$$



$\tilde{Y}_t$ is *still* not stationary, **however** its differences $\Delta \tilde{Y}_t = r + \epsilon_t$ **are stationary**.



The differences, in this case, also have an economic interpretation - it is the series of (logarithmic) returns, i.e. annual *growth* of Y_t.

Stock and bond returns (or similar financial series) can be described as having an average return of $\mathtt{r}$ but otherwise seemingly unpredictable from the past values (i.e. resembling *WN*): $Y_t = r + \epsilon_t, \quad \epsilon_t \sim WN(0, \sigma^2)$. Although the sequence may initially appear to be *WN*, there is strong evidence to suggest that it is not an *independent* process.

As such, we shall try to create a model of residuals: $e_t = \hat{\epsilon}_t$, i.e. centered returns $Y_t - \bar{Y}_t = Y_t - \hat{r}$ of real stocks that posses some interesting empirical properties:

- high volatility events tend to cluster in time (i.e. *persistency* or inertia of volatility);
- Y_t is uncorrelated with its lags, **but** Y_t^2 *is* correlated with $Y_{t-1}^2, Y_{t-2}^2, ...$;
- Y_t is heavy-tailed, i.e. the right tail of its density decreases slower than that of the Gaussian density (this means that Y_t take big values more often than Gaussian random variables).

Note: **volatility** = the conditional standard deviation of the stock return: $\sigma_t^2 = Var(r_t|\Omega_{t-1})$, where Ω_{t-1} - the information set available at time $t-1$.

An introductory example:

Let's say P_t denote the price of a financial asset at time t. Then, the log returns:

$$R_t = log(P_t) - log(P_{t-1})$$

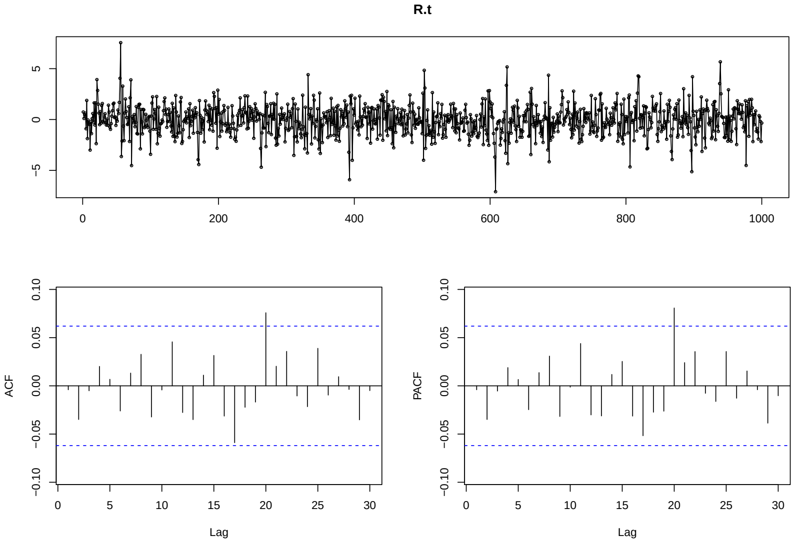
could be typically modeled as a stationary time series. An ARMA model for the series R_t would have the property that the conditional variance R_t is independent of t. However, in practice this is not the case. Lets say our R_t data is generated by the following process:

```
set.seed(346)
n = 1000
alpha = c(1, 0.5)
epsilon = rnorm(mean = 0, sd = 1, n = n)
R.t = NULL
R.t[1] = sqrt(alpha[1]) * epsilon[1]
for(j in 2:n){
  R.t[j] = sqrt(alpha[1] + alpha[2] * R.t[j-1]^2) * epsilon[j]
}
```

i.e., $R_t, t > 1$, nonlinearly depends on its past values.

If we plot the data and the ACF and PACF plots:

```
forecast::tsdisplay(R.t)
```



and perform the Ljung-Box test

```r
Box.test(R.t, lag = 10, type = "Ljung-Box")$p.value
```

```
## [1] 0.9082987
```

```r
Box.test(R.t, lag = 20, type = "Ljung-Box")$p.value
```

```
## [1] 0.3846643
```

```r
Box.test(R.t, lag = 25, type = "Ljung-Box")$p.value
```

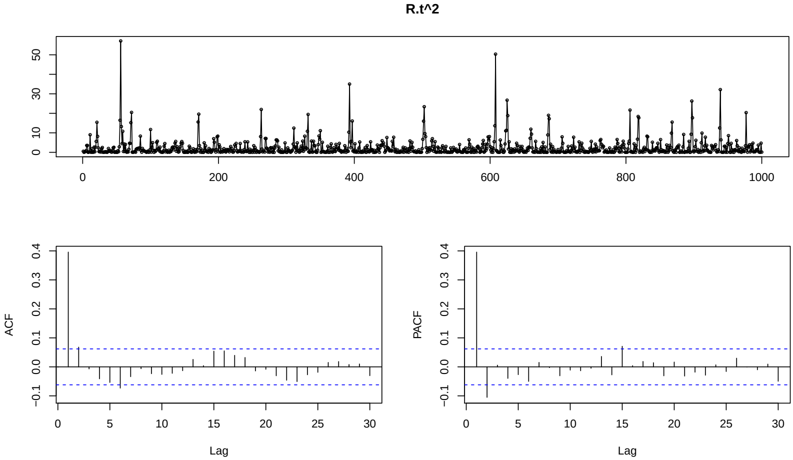
```
## [1] 0.4572007
```

We see that for all cases `p-value > 0.05`, so we do not reject the null hypothesis that the autocorrelations are zero. The series appears to be *WN*.

But we know that this is not the case from the data generation code.

If we check the ACF and PACF of the squared log-returns, R_t^2:

```r
forecast::tsdisplay(R.t^2)
```



The squared log-returns are autocorrelated in the first couple of lags.

From th `Ljung-Box` test:

```
Box.test(R.t^2, lag = 10, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  R.t^2
## X-squared = 174.37, df = 10, p-value < 2.2e-16
```

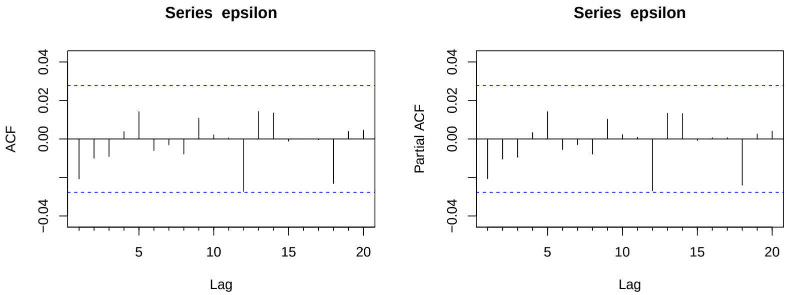
we do not reject the null hypothesis that the squared log-returns are autocorrelated.

In comparison, for a simple $\epsilon_t \sim WN(0,1)$ process:

```
set.seed(123)
epsilon = rnorm(mean = 0, sd = 1, n = 5000)
```

The ϵ_t process is not serially correlated:

```
par(mfrow = c(1, 2))
forecast::Acf(epsilon, lag.max = 20)
forecast::Pacf(epsilon, lag.max = 20)
```

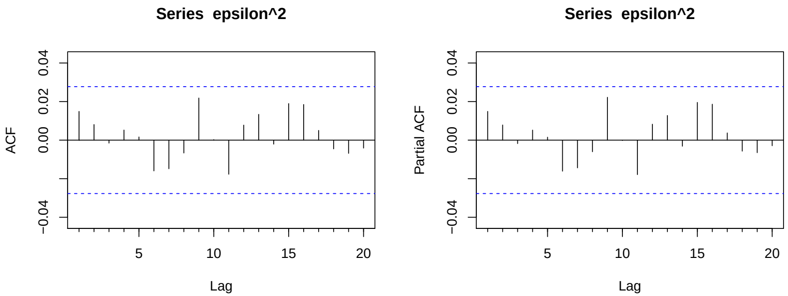



```
Box.test(epsilon, lag = 10, type = "Ljung-Box")$p.val
```

```
## [1] 0.872063
```

The ϵ_t^2 process is also not serially correlated:

```
par(mfrow = c(1, 2))
forecast::Acf(epsilon^2, lag.max = 20)
forecast::Pacf(epsilon^2, lag.max = 20)
```



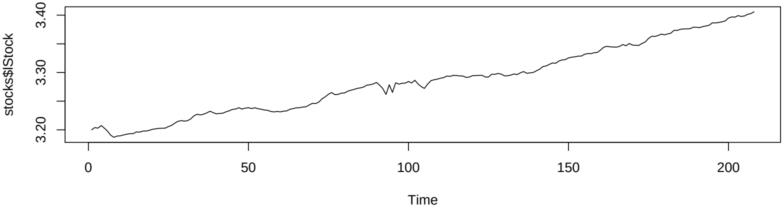

```
Box.test(epsilon^2, lag = 10, type = "Ljung-Box")$p.val
```

```
## [1] 0.7639204
```

So, R_t only *appeared* to be a *WN* process, unless we also analyse R_t^2.

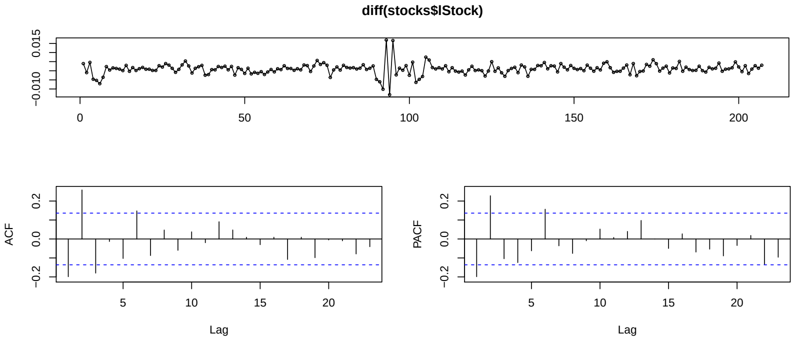
The following example stock data contains weekly data for logarithms of stock prices, $log(P_t)$:

```
suppressPackageStartupMessages({require(readxl)})
txt1 <- "http://uosis.mif.vu.lt/~rlapinskas/(data%20R&GRETL/"
txt2 <- "stock.xls"
tmp = tempfile(fileext = ".xls")
#Download the file
download.file(url = paste0(txt1, txt2),
              destfile = tmp, mode = "wb")
#Read it as an excel file
stocks <- read_excel(path = tmp)
plot.ts(stocks$lStock)
```



The differences do not pass WN checks:

```r
tsdisplay(diff(stocks$lStock))
```



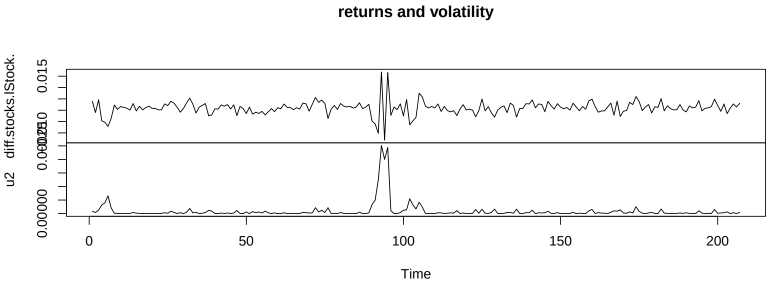

```r
Box.test(diff(stocks$lStock), lag = 10)$p.value
```

```
## [1] 3.014097e-05
```

The basic idea behind volatility study is that the series is serially uncorrelated, but it is a dependent series.

Let us calculate the volatility as $\hat{u}_t^2$ from $\Delta log(Y_t) = \alpha + u_t$

```
mdl <- lm(diff(stocks$lStock) ~ 1)
u <- residuals(mdl)
u2<- u^2
plot.ts(data.frame(diff(stocks$lStock), u2),
        main = "returns and volatility")
```



Note the small volatility in stable times and large volatility in fluctuating return periods.

We have learned that the AR process is able to model persistency, which, in our case, may be called `clustering of volatility`. Consider an AR(1) model of volatility (for this example we assume u_t^2 is WN):

$$u_t^2 = \alpha + \phi u_{t-1}^2 + w_t, \quad w_t \sim WN$$

```
library(forecast)
u2.mdl <- Arima(u2, order = c(1, 0, 0), include.mean = TRUE)
coef(u2.mdl)
```

```
##          ar1    intercept
## 7.335022e-01 9.187829e-06
```

Remember that for a *stationary* process u_t^2: $\mathbb{E}u_t^2 = \mu$. So $\mu = \alpha/(1-\phi)$. The `Arima` function returns the `intercept`, however, *if* the model has an **autoregressive part**, it is actually the **process mean**.

```
#To get the alpha coefficient of an AR process:
#alpha = mu *(1-phi)
unname(coef(u2.mdl)[2] * (1 - coef(u2.mdl)[1]))
```

```
## [1] 2.448536e-06
```

The resulting model:

$$u_t^2 = 0.00000245 + 0.7335u_{t-1}^2 + w_t$$

Might be of great interest to an investor wanting to purchase this stock.

- Suppose an investor has just observed that $u_{t-1}^2 = 0$, i.e. the stock price changes by its average amount in period $t - 1$. The investor is interested in predicting volatility in period t in order to judge the likely *risk* involved in purchasing the stock. Since the error is *unpredictable*, the investor ignores it (it could be positive or negative). So, the predicted volatility in period t is `0.00000245`.
- If the investor observed $u_{t-1}^2 = 0.0001$, then he would have predicted the volatility at period t to be `0.00000245 + 0.00007335 = 7.58e-05`, which is almost **31 times** bigger.

This kind of information can be incorporated into financial models of investor behavior.

Weak WN and Strong WN

- A sequence of **uncorrelated** random variables (with zero mean and constant variance) is called a *weak* WN;
- A sequence of **independent** random variables (with zero mean and constant variance) is called a *strong* WN;

If ϵ_t is a strong WN then so is ϵ_t^2 or any other function of ϵ_t.

Let $\Omega_s = \mathcal{F}(\epsilon_s, \epsilon_{s-1}, ...)$ be the set containing all the information on the past of the process.

If ϵ_t is a *strong* WN, then:

- conditional mean $\mathbb{E}(\epsilon_t|\Omega_{t-1}) = 0$;
- conditional variance $Var(\epsilon_t|\Omega_{t-1}) = \mathbb{E}(\epsilon_t^2|\Omega_{t-1}) = \sigma^2$

Now we shall present a model of *weak* WN process (its variance is constant) such that its *conditional variance* or *volatility* may change over time. The simplest way to model this kind of phenomenon is to use the ARCH(1) model.

From the rules for the mean :

$$\mathbb{E}(X + \alpha) = \mu + \alpha$$

and the variance

$$Var(\beta \cdot X + \alpha) = \beta^2 \cdot \sigma^2$$

we can modify the random variables to have different mean and variance:

$$\epsilon_t \sim \mathcal{N}(0, 1) \Rightarrow (\beta \cdot \epsilon_t + \mu) \sim \mathcal{N}(\mu, \quad \beta^2 \cdot 1)$$

If we take $\beta = \sigma_t$, we can have the variance change depending on the time t. We can then specify the volatility (i.e. standard deviation) as a separate equation and estimate its parameters.

Auto Regressive Conditional Heteroscedastic (ARCH) model

The core idea of the ARCH model is to effectively describe the dependence of volatility on recent (centered) returns r_t.

The ARCH(1) model can be written as:

$$\begin{cases} r_t &= \epsilon_t \\ \epsilon_t &= \sigma_t z_t \\ \sigma_t^2 &= \mathbb{E}(\epsilon_t^2 | \Omega_{t-1}) = \omega + \alpha_1 \epsilon_{t-1}^2 \end{cases}$$

where:

- z_t are (0,1) - Gaussian or Student (or similar symmetric) i.i.d. random variables (strong WN);
- $\omega, \alpha_1 > 0$;
- $\mathbb{E}(\epsilon_t) = 0$, $Var(\epsilon_t) = \omega/(1-\alpha_1)$, $Cov(\epsilon_{t+h}, \epsilon_t) = 0, \forall t \geq 0$ and $|h| \geq 1$. Also, $Var(\epsilon_t) \geq 0 \Rightarrow 0 \leq \alpha_1 < 1$.

An ARCH process is stationary. If the returns are not centered, then the first equation is $r_t = \mu + \epsilon_t$.

$ARCH(q)$:

The ARCH process can also be generalized:

$$\begin{cases} r_t & = \mu + \epsilon_t \\ \epsilon_t & = \sigma_t z_t \\ \sigma_t^2 & = \omega + \alpha_1 \epsilon_{t-1}^2 + ... + \alpha_q \epsilon_{t-q}^2 \end{cases}$$

$AR(P) - ARCH(q)$:

It may also be possible that the returns r_t themselves are autocorrelated:

$$\begin{cases} r_t & = \mu + \phi_1 r_{t-1} + ... + \phi_p r_{t-P} + \epsilon_t \\ \epsilon_t & = \sigma_t z_t \\ \sigma_t^2 & = \omega + \alpha_1 \epsilon_{t-1}^2 + ... + \alpha_q \epsilon_{t-q}^2 \end{cases}$$

Continuing the stock example (1)

Recall that our 'naive' log stock return data volatility model was:

$$\widehat{u^2}_t = 0.00000245 + 0.7335\widehat{u^2}_{t-1}$$

Because the coefficient of u_{t-1}^2 was significant - it could indicate that u_t^2 is probably an $ARCH(1)$ process.

```r
suppressPackageStartupMessages({library(fGarch)})
mdl.arch <- garchFit(~ garch(1,0), diff(stocks$lStock),
                     trace = FALSE)
mdl.arch@fit$matcoef
```

```
##             Estimate   Std. Error  t value     Pr(>|t|)
## mu      1.048473e-03 1.132355e-04 9.259222 0.000000e+00
## omega   2.400242e-06 3.904157e-07 6.147914 7.850864e-10
## alpha1  6.598808e-01 1.571422e-01 4.199260 2.677887e-05
```

So, our model looks like:

$$\begin{cases} \Delta \widehat{log(stock_t)} & = \mu = 0.001048 \\ \\ \widehat{\sigma^2}_t & = \omega + \alpha_1 \widehat{\sigma^2}_{t-1} = 2.4 \cdot 10^{-6} + 0.660 \widehat{\sigma^2}_{t-1} \end{cases}$$

Recall from `tsdisplay(diff(stocks$lStock))` that the returns are not WN (they might be an AR(6) process). To find the proper conditional mean model for the returns, we use `auto.arima` function.

```
mdl.ar <- auto.arima(diff(stocks$lStock), max.p = 10, max.q = 0)
mdl.ar$coef #AR(7) model is recommended
```

```
##          ar1          ar2          ar3          ar4
## -0.134997783  0.249189502 -0.095223779 -0.167506460 -0.024943
##          ar6          ar7    intercept
##  0.159953621 -0.028619401  0.000983335
```

We combine it with ARCH(1) to create a AR(7)-ARCH(1) model:

```r
mdl.arch.final <- garchFit(~ arma(7,0) + garch(1,0),
                           diff(stocks$lStock),
                           trace = FALSE)
mdl.arch.final@fit$matcoef
```

```
##             Estimate   Std. Error    t value     Pr(>|t|)
## mu      1.193945e-03 1.730481e-04  6.8994954 5.218714e-12
## ar1    -1.236738e-01 7.070313e-02 -1.7491979 8.025682e-02
## ar2     8.081154e-02 4.427947e-02  1.8250341 6.799588e-02
## ar3    -3.825929e-02 4.558812e-02 -0.8392383 4.013356e-01
## ar4    -1.069443e-01 3.932896e-02 -2.7192253 6.543502e-03
## ar5     7.208729e-03 3.970051e-02  0.1815777 8.559141e-01
## ar6     1.635547e-01 3.580176e-02  4.5683442 4.915924e-06
## ar7    -1.124515e-01 3.388652e-02 -3.3184725 9.051122e-04
## omega   2.045548e-06 3.566767e-07  5.7350195 9.750115e-09
## alpha1  6.503373e-01 1.721740e-01  3.7772104 1.585947e-04
```

The Generalized ARCH (GARCH) model

Although the ARCH model is simple, it often requires many parameters to adequately describe the volatility process of an asset return. To reduce the number of coefficients, an alternative model must be sought.

If an ARMA type model is assumed for the error variance, then a $GARCH(p,q)$ model should be considered:

$$\begin{cases} r_t = \mu + \epsilon_t \\ \epsilon_t = \sigma_t z_t \\ \sigma_t^2 = \omega + \sum_{j=1}^{q} \alpha_j \epsilon_{t-j}^2 + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2 \end{cases}$$

A GARCH model can be regarded as an application of the ARMA idea to the series ϵ_t^2.

Both *ARCH* and *GARCH* are (weak) *WN* processes with a special structure of their conditional variance.

Such processes are described by an almost endless family of *ARCH* models: $ARCH$, $GARCH$, $TGARCH$, $GJR-GARCH$, $EGARCH$, $GARCH-M$, $AVGARCH$, $APARCH$, $NGARCH$, $NAGARCH$, $IGARCH$ etc.

Volatility Model Building

Building a volatility model consists of the following steps:

1. Specify a **mean equation** of r_t by testing for serial dependence in the data and, if necessary, build an econometric model (e.g. ARMA model) to remove any linear dependence.
2. Use the residuals of the mean equation, $\widehat{e}_t = r_t - \widehat{r}_t$ to test for **ARCH effects**.
3. If ARCH effects are found to be significant, one can use the PACF of $\widehat{e}_t^2$ to determine the ARCH order (may not be effective when the sample size is small). Specifying the order of a GARCH model is not easy. Only lower order GARCH models are used in most applications, say, $GARCH(1,1)$, $GARCH(2,1)$, and $GARCH(1,2)$ models.
4. Specify a volatility model if ARCH effects are statistically significant and perform a **joint** estimation of the mean and volatility equations.
5. Check the fitted model carefully and refine it if necessary.

Testing for ARCH Effects

Let $\epsilon_t = r_t - \hat{r}_t$ be the residuals of the **mean equation**. Then ϵ_t^2 are used to check for conditional heteroscedasticity (i.e. the **ARCH effects**). Two tests are available:

1. Apply the usual `Ljung-Box` statistic $Q(k)$ to ϵ_t^2. The null hypothesis is that the first k lags of ACF of ϵ_t^2 are zero:
   $H_0 : \rho(1) = 0, \rho(2) = 0, ..., \rho(k) = 0$
2. The second test for the conditional heteroscedasticity is the Lagrange Multiplier (**LM**) test, which is equivalent to the usual $F-$*statistic* for testing $H_0 : \alpha_1 = ... = \alpha_k = 0$ in the linear regression:

$$\epsilon_t^2 = \alpha_0 + \sum_{j=1}^{k} \epsilon_{t-j}^2 + e_t, \quad t = k+1, ..., T$$

Continuing the stock example (2)

Going through each of the steps:

```
tsdisplay(diff(stocks$lStock))
```

The log-returns are autocorrelated. So we need to specify an *ARMA* model for the **mean equation** via `auto.arima`:

```
mdl.auto <- auto.arima(diff(stocks$lStock))
rbind(names(mdl.auto$coef)[1:3], names(mdl.auto$coef)[4:6])
```

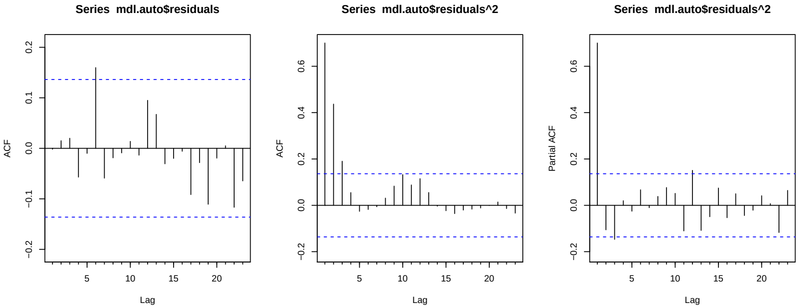
```
##      [,1]  [,2]  [,3]
## [1,] "ar1" "ar2" "ar3"
## [2,] "ar4" "ma1" "intercept"
```

The output is and ARMA(3,2) model:

$$r_t = \mu + \phi_1 r_{t-1} + \phi_2 r_{t-2} + \phi_3 r_{t-3} + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2}$$

Now, we examine the residuals of this model:

```
par(mfrow = c(1,3))
forecast::Acf(mdl.auto$residuals)
forecast::Acf(mdl.auto$residuals^2)
forecast::Pacf(mdl.auto$residuals^2)
```



We see that the ACF of the residuals are not autocorrelated, however the squared residuals **are autocorrelated**. So, we need to create a volatility model. Because the first lag of the PACF plot of the squared residuals is significantly different from zero, we need to specify an $ARCH(1)$ model for the residuals.

The final model is an $ARMA(3, 2) - ARCH(1)$:

```
mdl.arch.final <- garchFit(~ arma(3, 2) + garch(1, 0),
                           diff(stocks$lStock),
                           trace = FALSE)
mdl.arch.final@fit$matcoef
```

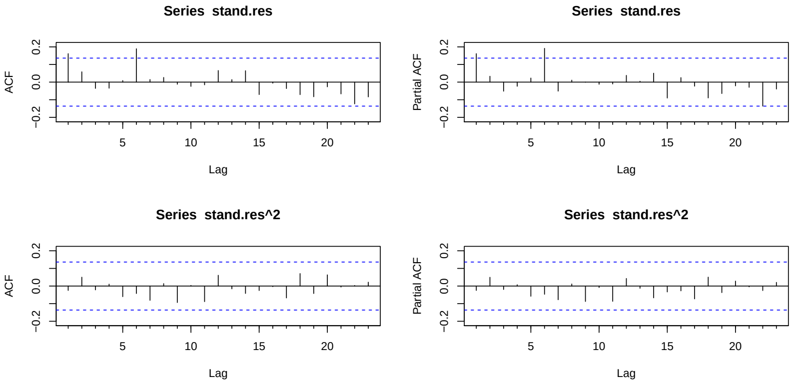
```
##              Estimate   Std. Error    t value     Pr(>|t|)
## mu       1.980586e-03 3.367634e-04  5.8812393 4.072058e-09
## ar1     -2.743818e-01 1.943599e-01 -1.4117200 1.580324e-01
## ar2     -6.001322e-01 1.365386e-01 -4.3953299 1.106047e-05
## ar3     -1.065850e-01 8.060903e-02 -1.3222460 1.860863e-01
## ma1      1.258717e-01 1.831323e-01  0.6873265 4.918770e-01
## ma2      7.018161e-01 1.486765e-01  4.7204244 2.353530e-06
## omega    2.488709e-06 4.030309e-07  6.1749835 6.617036e-10
## alpha1   6.216022e-01 1.525975e-01  4.0734767 4.631649e-05
```

```
mdl.arch.final@fit$ics
```

```
##       AIC       BIC       SIC      HQIC
## -9.359004 -9.230203 -9.361846 -9.306918
```

Finally, we check the standardized residuals $\hat{w}_t = \hat{\epsilon}_t / \hat{\sigma}_t$ to check if $\hat{w}_t$ and $\hat{w}_t^2$ are *WN*:

```
par(mfrow = c(2,2))
stand.res = mdl.arch.final@residuals / mdl.arch.final@sigma.t
forecast::Acf(stand.res); forecast::Pacf(stand.res)
forecast::Acf(stand.res^2); forecast::Pacf(stand.res^2)
```



Unfortunately, the residuals $\hat{w}_t$ still seem to be autocorrelated. In this case, more complex models should be considered, like the ones mentioned in the GARCH model slide ... **But this may not be necessary!**

These tests are performed and provided in the model output:

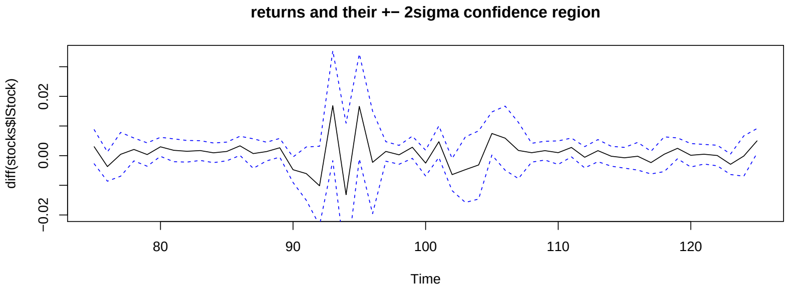
```
capture.output(summary(mdl.arch.final))[46:56]
```

```
##  [1] "Standardised Residuals Tests:"
##  [2] "                                Statistic p-Value  "
##  [3] " Jarque-Bera Test   R    Chi^2  2.981865  0.2251626"
##  [4] " Shapiro-Wilk Test  R    W      0.9941911 0.6029121"
##  [5] " Ljung-Box Test     R    Q(10)  14.81308  0.1390265"
##  [6] " Ljung-Box Test     R    Q(15)  17.92572  0.2665907"
##  [7] " Ljung-Box Test     R    Q(20)  21.14201  0.3888168"
##  [8] " Ljung-Box Test     R^2  Q(10)  5.334754  0.8677243"
##  [9] " Ljung-Box Test     R^2  Q(15)  8.492303  0.9025344"
## [10] " Ljung-Box Test     R^2  Q(20)  12.02647  0.9151619"
## [11] " LM Arch Test       R    TR^2   8.228338  0.7670416"
```

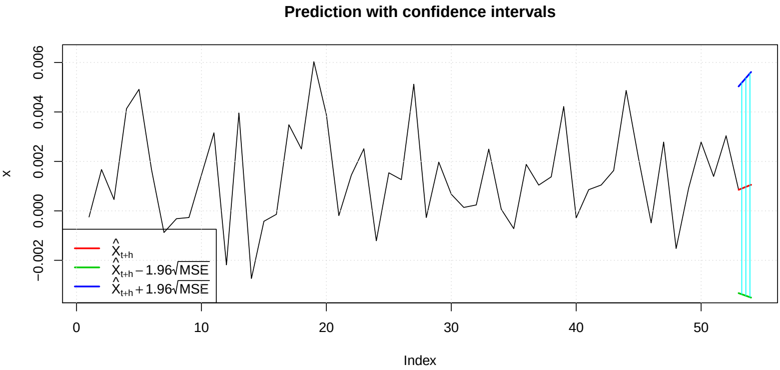
We see that `Jarque-Bera Test` and `Shapiro-Wilk Test` `p-values > 0.05`, so we do NOT reject the null hypothesis of normality of the standardized residuals `R`. The `Ljung-Box Test` for the standardized residuals `R` and `R^2` `p-values > 0.05`, so the residuals form a *WN*. Finally, the `LM Arch Test` `p-value > 0.05` shows that there are no more ARCH effects in the residuals. **So, our estimated model is correctly specified in the sense that the residual autocorrelation from the ACF/PACF plots is relatively weak!**

To explore the predictions of volatility, we calculate and plot 51 observations from the middle of the data along with the one-step-ahead predictions of the corresponding volatility $\widehat{\sigma^2}_t$:

```
d_lstock <- ts(diff(stocks$lStock))
sigma = mdl.arch.final@sigma.t
plot(window(d_lstock, start = 75, end = 125),
     ylim = c(-0.02, 0.035), ylab = "diff(stocks$lStock)",
     main = "returns and their +- 2sigma confidence region")
lines(window(d_lstock - 2*sigma, start = 75, end = 125),
      lty = 2, col = 4)
lines(window(d_lstock + 2*sigma, start = 75, end = 125),
      lty = 2, col = 4)
```



```r
predict(mdl.arch.final, n.ahead = 2, mse ="cond", plot = T)
```




```
##   meanForecast   meanError standardDeviation lowerInterval up
## 1 0.0008520921 0.002132817       0.002132817  -0.003328152
## 2 0.0010536363 0.002327369       0.002305715  -0.003507924
```

Data Sources

A useful R package for downloading financial data directly from open sources, like Yahoo Finance, Google Finance, etc., is the quantmod package. Click here for some examples.

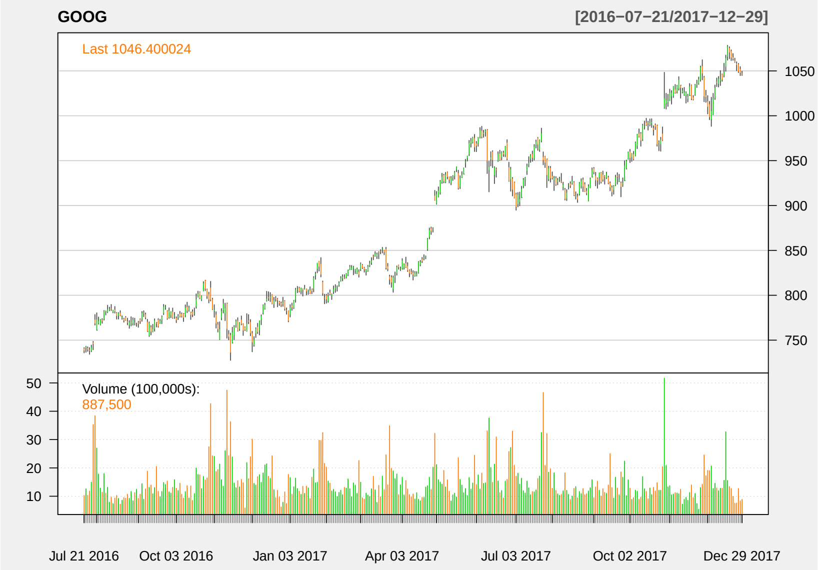
```
suppressPackageStartupMessages({library(quantmod)})
suppressMessages({
  getSymbols("GOOG", from = "2007-01-03", to = "2018-01-01")
})
tail(GOOG, 3)
```

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

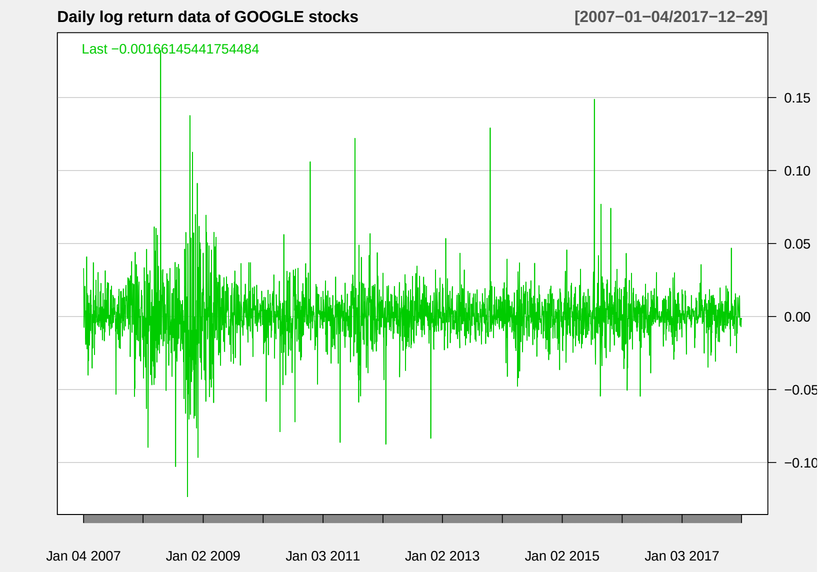
```
##            GOOG.Open GOOG.High GOOG.Low GOOG.Close
## 2017-12-27   1057.39   1058.37  1048.05    1049.37
## 2017-12-28   1051.60   1054.75  1044.77    1048.14
## 2017-12-29   1046.72   1049.70  1044.90    1046.40
##            GOOG.Volume GOOG.Adjusted
## 2017-12-27     1271900       1049.37
## 2017-12-28      837100       1048.14
## 2017-12-29      887500       1046.40
```

Time plots of daily closing price and trading volume of Google from the last 365 *trading* days:

```
chartSeries(tail(GOOG, 365), theme = "white", name = "GOOG")
```



```r
GOOG.rtn = diff(log(GOOG[, "GOOG.Adjusted"]))
chartSeries(GOOG.rtn, theme = "white",
            name = "Daily log return data of GOOGLE stocks")
```

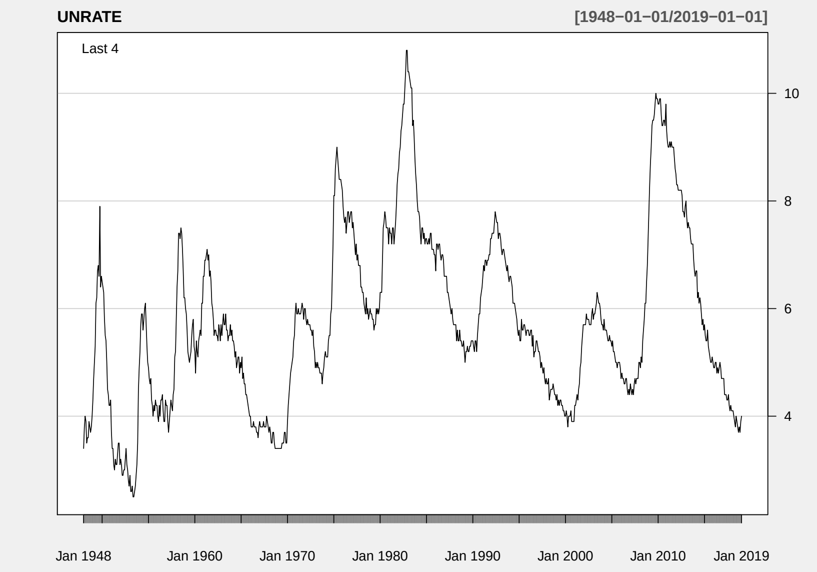


Example of getting non-financial data. Unemployment rates from FRED:

```
getSymbols("UNRATE", src = "FRED")
```

```
## [1] "UNRATE"
```

```
chartSeries(UNRATE, theme = "white", up.col = 'black')
```



Summary of Volatility Modelling (1)

Quite often, the process we want to investigate for the ARCH effects is stationary but not WN.

- Let ϵ_t be a weak $WN(0, \sigma^2)$ and consider the model $Y_t = r + \epsilon_t$, or $Y_t = \beta_0 + \beta_1 X_t + \epsilon_t$ or $Y_t = \alpha + \phi Y_{t-1} + \epsilon_t$ or similar.
- Test whether the WN shocks ϵ_t make an ARCH process: plot the graph of e_t^2 ($= \hat{\epsilon}_t^2$) - if ϵ_t is an ARCH process, this graph must show a clustering property.
- Further test whether the shocks ϵ_t form an ARCH process: test them for normality (the hypothesis must be rejected) (e.g. using `Shapiro-Wilk test of normality`).
- Further test whether the shocks ϵ_t form an ARCH process: draw the correlogram of e_t - the correlogram **must indicate WN**, but that of e_t^2 **must not** (it should be similar to the correlogram of an AR(p) process).

Summary of Volatility Modelling (2)

- To formally test whether the shocks ϵ_t form ARCH(q), test the null hypothesis $H_0 : \alpha_1 = ... = \alpha_q = 0$ (i.e. no ARCH in $\sigma_t^2 = \omega + \sum_{j=1}^{q} \alpha_j \epsilon_{t-j}^2$):
  1. Choose the **proper** $AR(q)$ model of the auxiliary regression $e_t^2 = \alpha + \alpha_1 e_{t-1}^2 + ... + \alpha_q e_{t-1}^2 + w_t$ (proper means minimum AIC and WN residuals w_t);
  2. To test H_0, use the $F - test$ (or the LM test).
- Instead of using ARCH(q) with a high order q, an often more parsimonious description of ϵ_t is usually given by **GARCH(1,1)** (or a similar lower order GARCH process);
- In order to show that the selected ARCH(q) or GARCH(1,1) model is ‘good’, test whether the residuals $\hat{w}_t = \hat{\epsilon}_t / \hat{\sigma}_t$ and $\hat{w}_t^2$ make WN (as they are expected to).