Lecture I.- Fundamentals of Time Series Econometrics

Marcelo Villena, PhD
Santa María University

August , 2022

Outline

1. The nature of time series data
2. Statistical time series modeling
3. Time series decomposition
4. Dependency measures
5. Stationarity
6. Multiple linear regression in time series models
7. References

The nature of time series data

- The main objective of time series analysis is to develop mathematical models that provide plausible descriptions for the sample data.
- There are two basic methodological approaches to time series modeling:

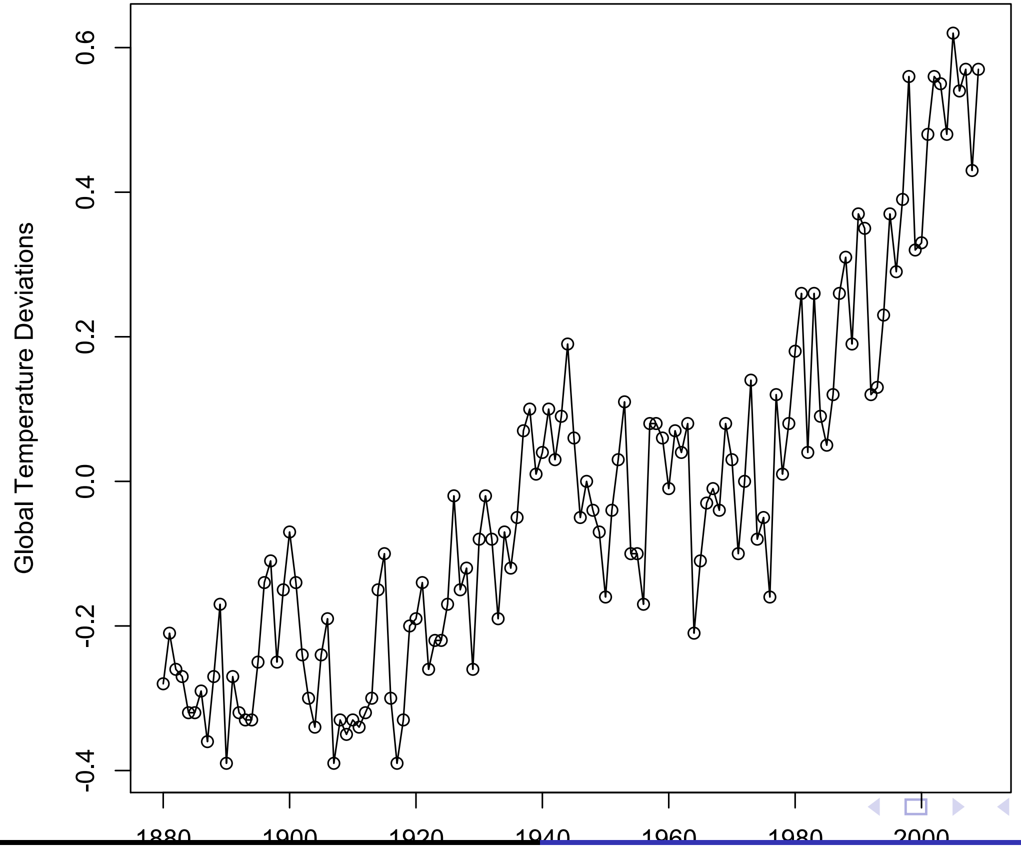
1. The time domain approach
2. The frequency domain approach

The nature of time series data

- On the other hand, **the frequency-domain approach** assumes that the main features of interest in the time series analysis are related to peri odic or systematic sinusoidal variations that are naturally found in the bulk of the data.
- These periodic variations are often caused by intervening biological, physical, or environmental phenomena. The study of periodicity extends to economics and the social sciences, where one may be interested in annual periodicities in series such as monthly unemployment or monthly birth rates.
- In spectral analysis, the participation of the various types of periodic variation in a time series is carried out by separately evaluating the variance associated with each type of periodicity.

Example 1: Climate change

- Our first example of a time series is the temperature of the earth.



Example 1: Climate change

- We observe an apparent upward trend in the series during the latter part of the 20th century, which has been used as an argument for the global warming hypothesis. Note also the rather pronounced upward trend around 1970. The question of interest to global warming proponents and opponents is whether the overall trend is natural, or whether it is human-caused.

Example 1: Climate change

R Code

```
rm(list=ls())
mydata < −read.csv (“/Users/marcelovillena/Desktop/gtemp.csv”,
header = TRUE, stringsAsFactors = FALSE) plot(mydata,
type=“o”, ylab=“Global Temperature Deviations”)
```

Example 2: Financial time series

- In finance it is always preferable to work with asset returns, rather than directly using the asset price. There are two ways to convert price into returns:

$$R_t = \frac{p_t - p_{t-1}}{p_{t-1}} * 100$$

$$R_t = ln\left(\frac{p_t}{p_{t-1}}\right) * 100$$

where, R_t denotes the return to time t, p_t denotes the price of the asset at the time t, and ln denotes the natural logarithm. In this formulation we ignore dividends, or assume that the price series have already been adjusted for them.

Example 2: Financial time series

- Log-returns have the desirable property of being interpreted as continuously compounded returns. In addition, they can be simply summed, so as to obtain returns over longer periods:

$$
\begin{aligned}
r_1 &= ln\frac{p_1}{p_0} = lnp_1 - lnp_0 \\
r_2 &= ln\frac{p2}{p_1} = lnp_2 - lnp_1 \\
r_3 &= ln\frac{p3}{p_2} = lnp_3 - lnp_2 \\
r_4 &= ln\frac{p4}{p_3} = lnp_4 - lnp_3 \\
r_5 &= ln\frac{p5}{p_4} = lnp_5 - lnp_4
\end{aligned}
$$

$$r_1 + r_2 + r_3 + r_4 + r_5 = lnp_5 - lnp_0 = ln\frac{p_5}{p_0}$$

Example 2: Financial time series

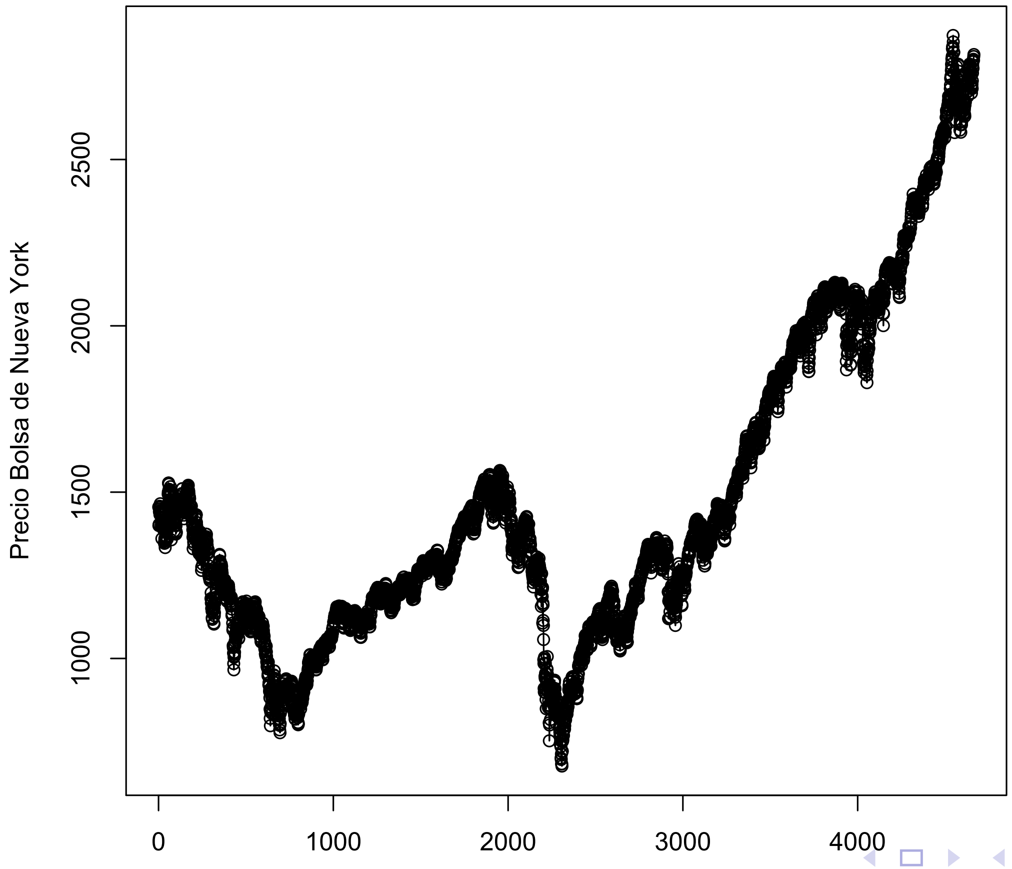
As a second example, we will calculate the returns of the New York Stock Exchange, ´ındice ”S&P 500”, extracting daily data since the year 2000, from the site: https://finance.yahoo.com/

R Code

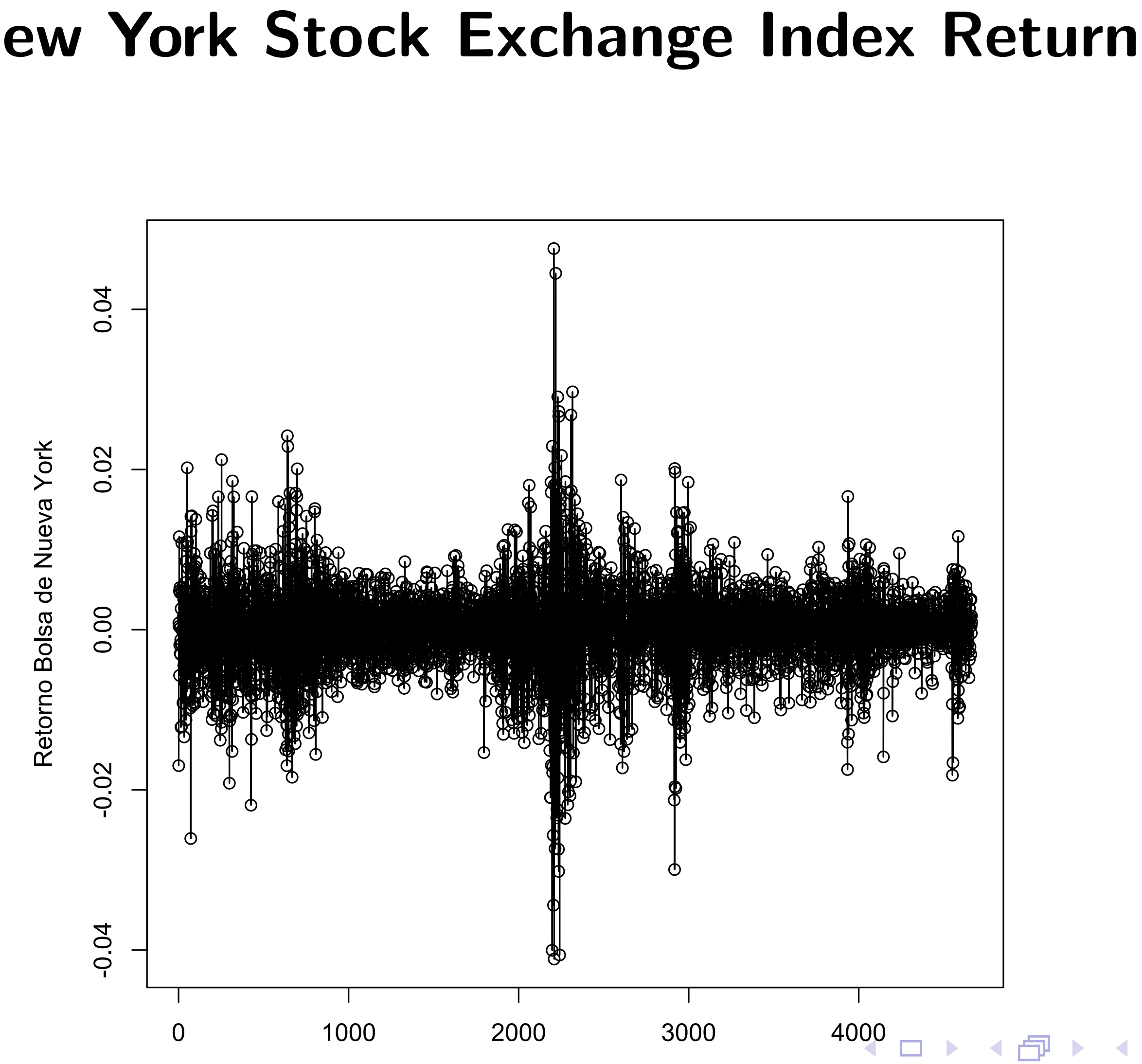
```
rm(list=ls())
mydata < − read.csv (“/Users/marcelovillena/Desktop/sp.csv”,
header = TRUE, stringsAsFactors = FALSE)
precio < − mydata$”Adj.Close”
plot.ts(precio, type=”o”, ylab=“New York Stock Exchange Price”)
lnprecio < − log10(precio)
Dlnprecio < − diff(lnprecio,1)
plot.ts(Dlnprecio, type=“o”, ylab=“New York Stock Exchange
Return”)
```

Example 2: Financial time series

New York Stock Exchange Index

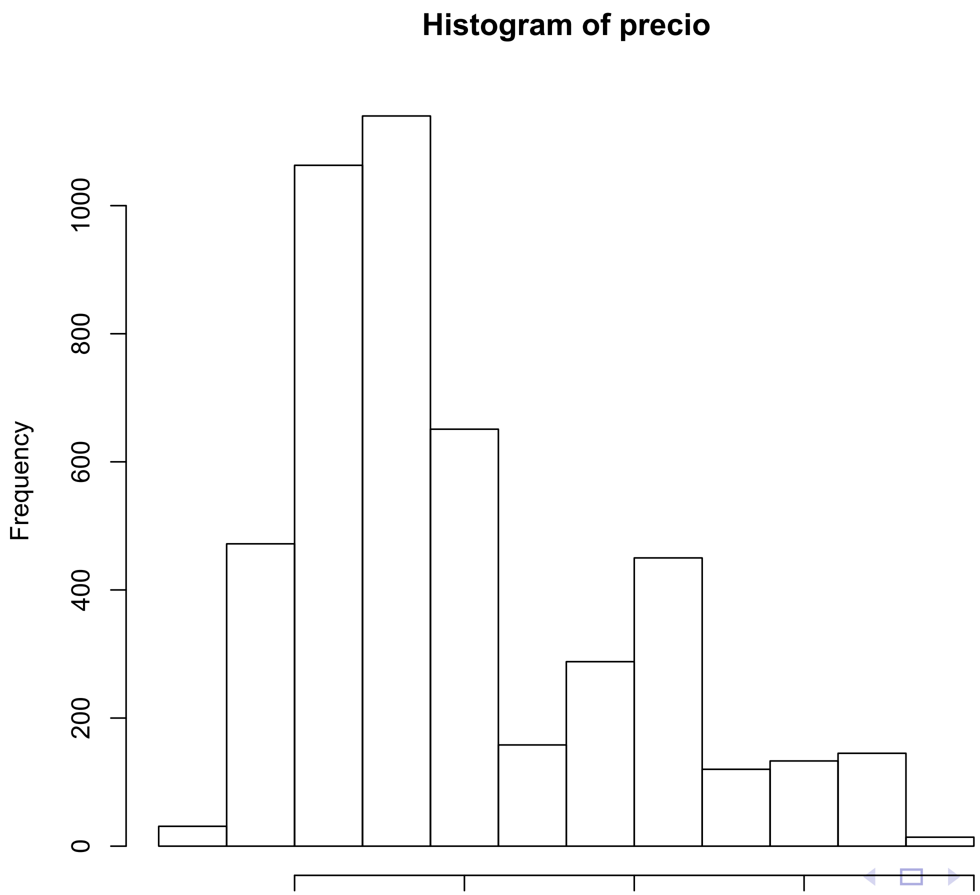


Example 2: Financial time series



Example 2: Financial time series

- Prices are apparently not normal, apparently they are log-normal.



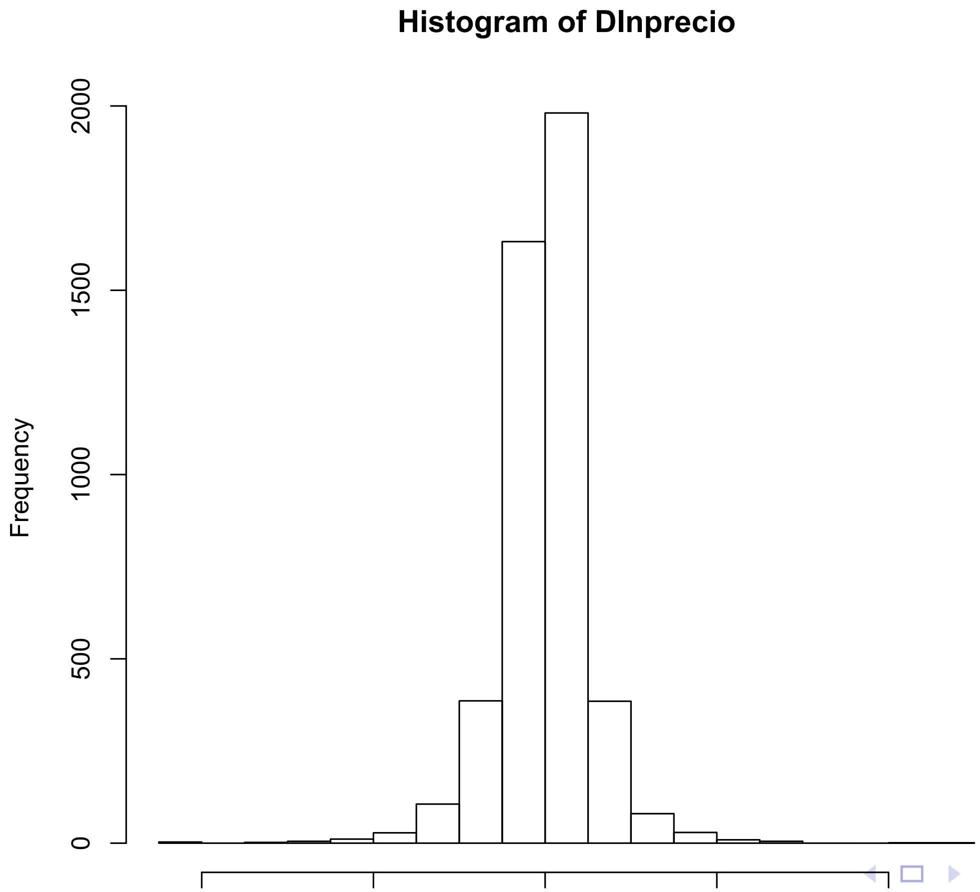
Example 2: Financial time series

R Code

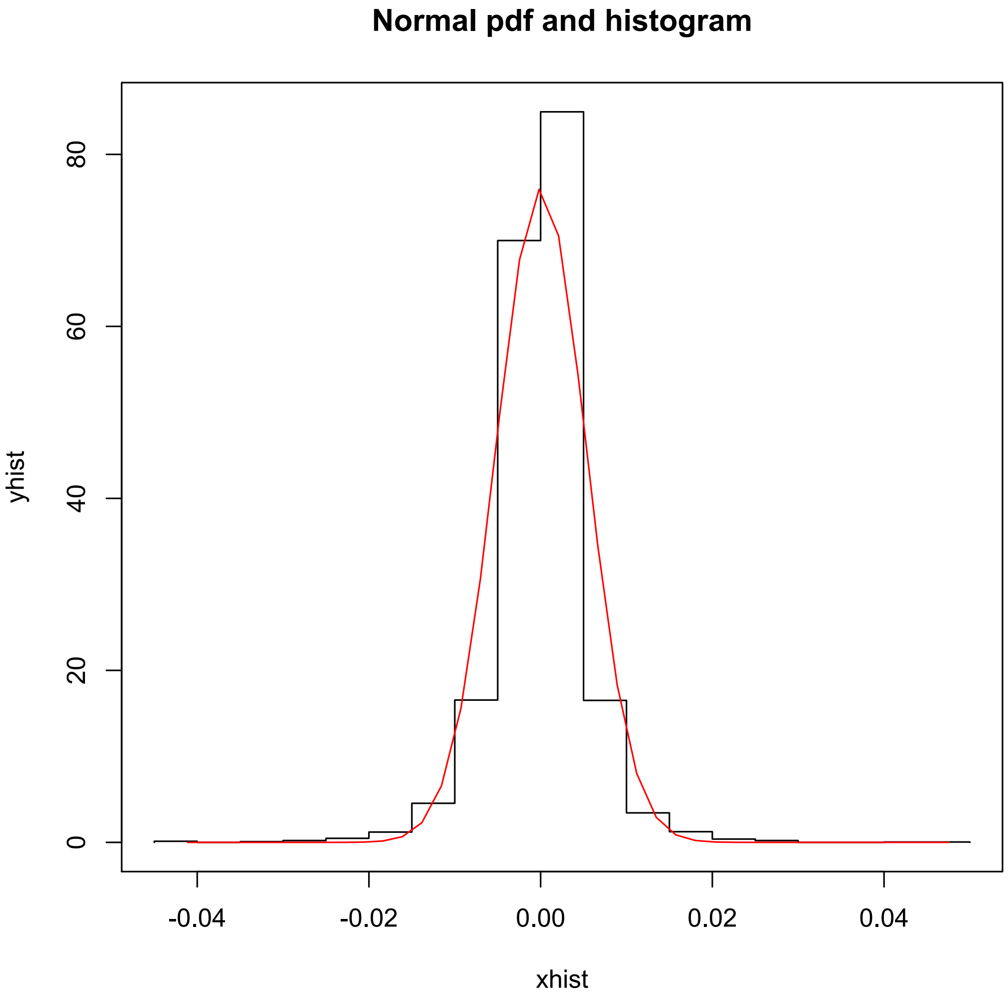
```
h < − hist(Dlnprecio,breaks=15)
xhist < − c(min(h$breaks),h$breaks)
yhist < − c(0,h$density,0)
xfit < − seq(min(Dlnprecio),max(Dlnprecio),length=40)
yfit < − dnorm(xfit,mean=mean(Dlnprecio),sd=sd(Dlnprecio))
plot(xhist,yhist,type=”s”,ylim=c(0,max(yhist,yfit)), main=“Normal
pdf and histogram”)
lines(xfit,yfit, col=“red”)
shapiro.test(Dlnprecio)
```

Example 2: Financial time series

- Price returns do appear normal, which is a desirable property for statistical analysis.



Example 2: Financial time series



Statistical time series modeling

- In order to model the data, which apparently fluctuate randomly over time, we assume that a time series is defined as a collection of random variables.
- For example, we can model a time series as a sequence of random variables, $x_1, x_2, x_3, ...$, where the random variable x_1 denotes the value taken by the series at the first time point, the variable x_2 denotes the value for the second time period, and so on.
- In general, a collection of random variables, $\{x_t\}$, indexed by t is known as a stochastic process. In this text, t will be typically discrete and vary over the integers $t = 0, \pm 1, \pm 2, , ...$, or some subset of the integers. The values observed in a stochastic process are known as the realization of the stochastic process.

White Noise

- A widely used time series is that represented by a collection of uncorrelated random variables, ϵ_t, with average 0 and finite variance σ_ϵ^2. Time series generated from uncorrelated variables are used for example to model noise in engineering applications, where it is called white noise. We will sometimes denote this process as $\epsilon_t \sim \epsilon_n(0, \sigma_\epsilon^2)$. The designation "white" originates from the analogy with white light and indicates that all possible period oscillations are present with the same strength.

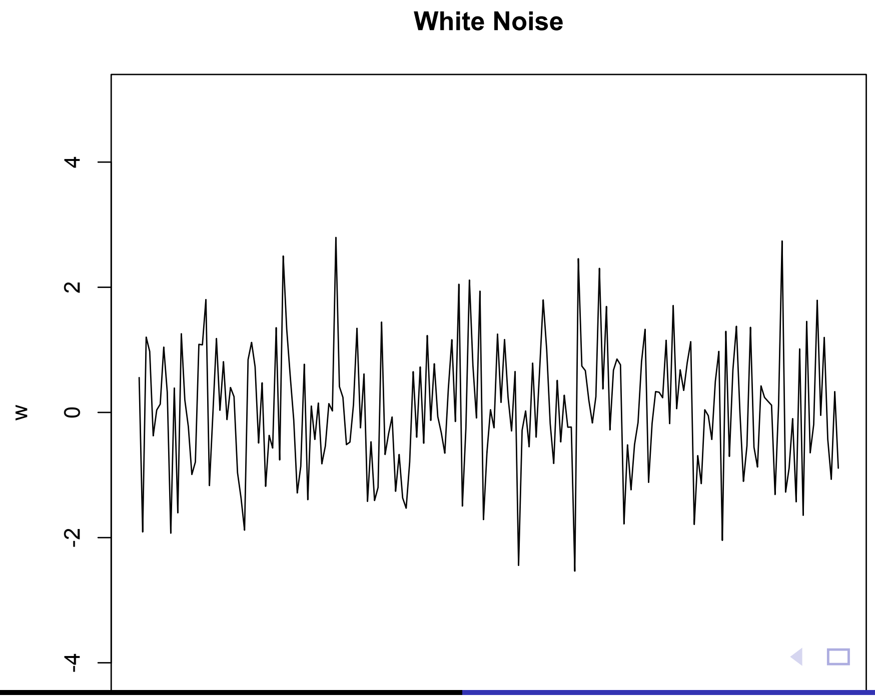
White Noise

- Occasionally, we will also require that the noise be independent and identically distributed (iid) random variables with mean 0 and variance σ_ϵ^2. Distinguish this case by saying independent white noise, or by writing $\epsilon_t \sim iid(0, \sigma_\epsilon^2)$.
- Another particularly useful white noise series is Gaussian white noise, in which the w_t are independent normal random variables, with mean 0 and variance σ_ϵ^2; or more succinctly, $\epsilon_t \sim iid\ N(0, \sigma_\epsilon^2)$.
- The figure below shows a collection of 500 of these random variables, with $\sigma_\epsilon^2 = 1$, drawn in the order in which they were drawn.

White Noise

R Code

```
set.seed(154)
w = rnorm(200,0,1)
plot.ts(w, ylim=c(-3,3), main=”White Noise”)
```



Random Walk

- A simple example for modeling a trending (non-stationary) stochastic time series is a Random Walk with drift:

$$x_t = \delta + x_{t-1} + \epsilon_t$$

- For $t = 1, 2, \ldots$, with an initial conditionl $x_0 = 0$, and where ϵ_t is white noise. The constant δ is referred to as drift, and when $\delta = 0$, is simply called a random walk. The term random walk derives from the fact that, when $\delta = 0$, the value of the time series over time t, is the value of the series over time $t - 1$ The movement will be completely random and determined by ϵ_t.
- Note that we can rewrite the above equation as a cumulative sum of the white noise variables. That is:

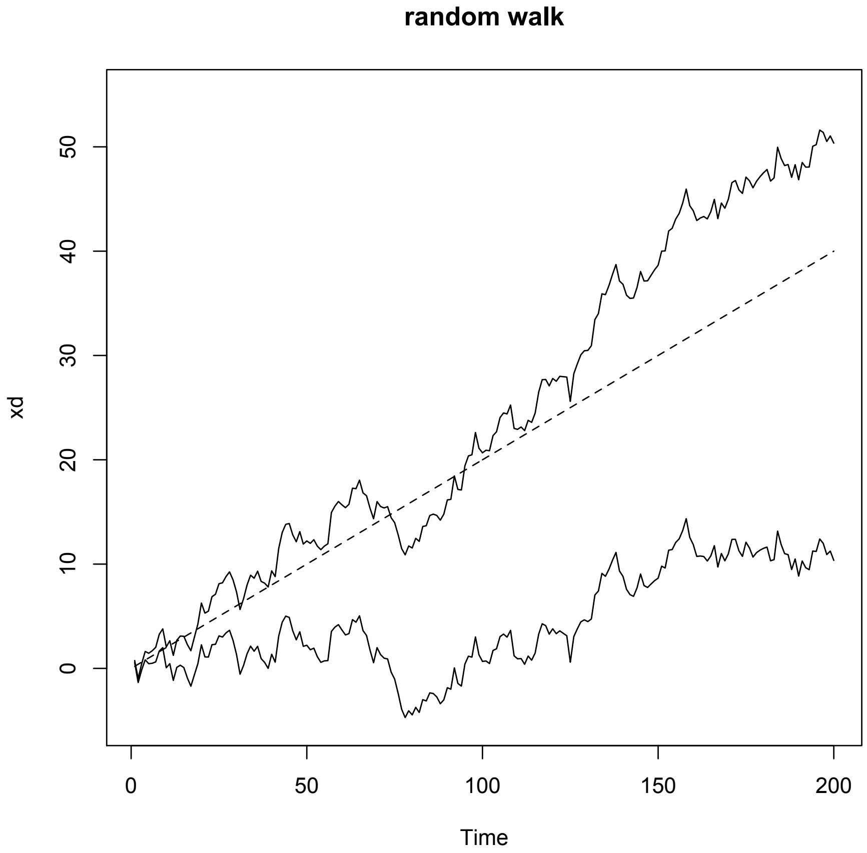
$$x_t = \delta t + \sum^{t} \epsilon_t$$

Random Walk

R Code

```
set.seed(154)
w = rnorm(200,0,1)
x = cumsum(w)
wd = w + 0.2
xd = cumsum(wd)
plot.ts(xd, ylim=c(-5,55), main=“random walk”)
lines(x)
lines(0.2*(1:200), lty=“dashed”)
```

Random Walk



Random walk, $\sigma_\epsilon = 1$, with drift $\delta = 0.2$ (upper jagged line), without drift, $\delta = 0$ (lower jagged line), and a straight line with

Moving Average

- We could replace the series of white noise ϵ_t by a moving average that smooths the series. For example, consider replacing ϵ_t by an average of its current value and its immediate neighbors in the past and future. In other words:

$$v_t = 1/3(\epsilon_{t-1} + \epsilon_t + \epsilon_{t+1})$$

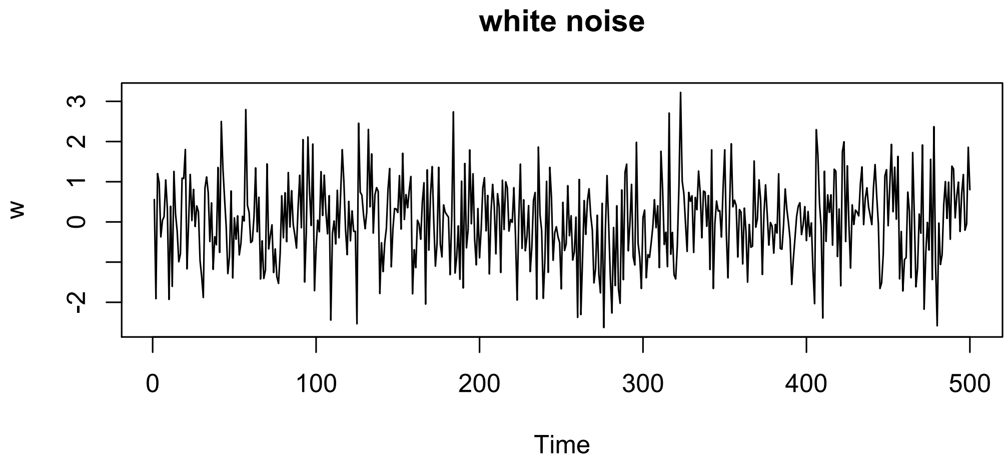
- As we will see in the following example, moving averages produce a smoother version than the original series, reflecting the fact that slower oscillations become more evident, and some of the faster oscillations are eliminated.

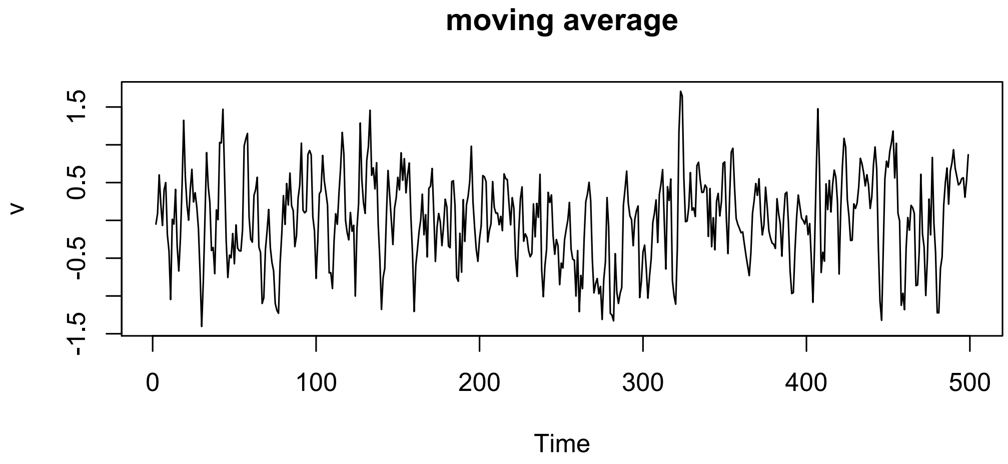
Moving Average

R Code

```
w = rnorm(500,0,1)
v = filter(w, sides=2, rep(1/3,3)) par(mfrow=c(2,1))
plot.ts(w, main=“white noise”)
plot.ts(v, main=“moving average”)
```

Moving Average





Autoregressions

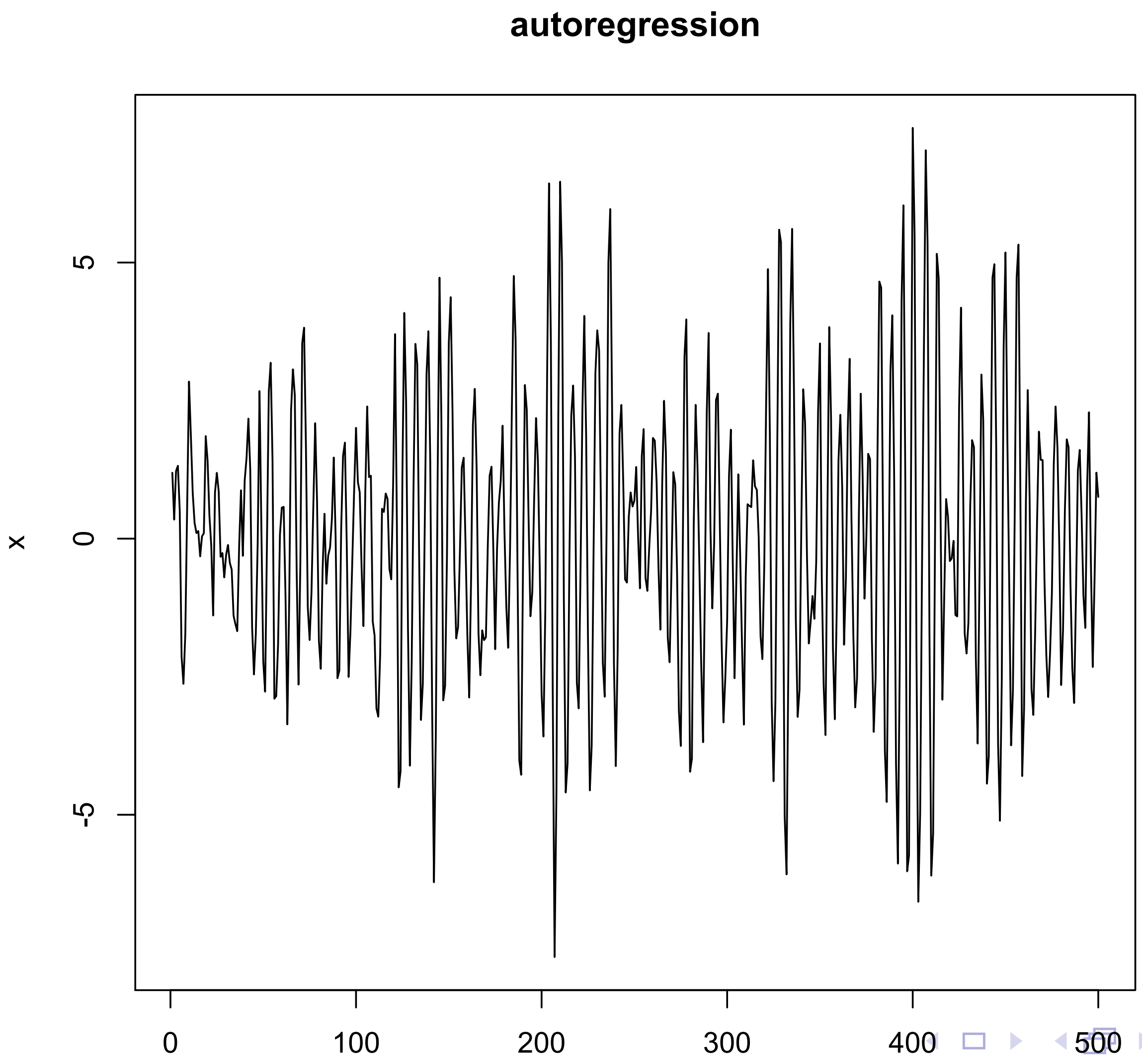
- Suppose again that we consider the white noise series w_t as input, and calculate the output using a second order equation, i.e:
$$x_t = x_{t-1} - 0.9x_{t-2} + \epsilon_t$$
- This equation represents a regression or prediction of the current value x_t of a time series as a function of the two previous values of the series, which is why we use the name autoregression.

Autoregressions

R Code

```
w = rnorm(550,0,1)
x = filter(w, filter=c(1,-.9), method= “recursive”)[-(1:50)]
plot.ts(x, main= “autoregression”)
```

Autoregressions



Time series decomposition

Time series are usually decomposed into:

1. A trend T_t.
2. A seasonal component S_t.
3. An irregular element I_t.

For example

$$T_t = 2 + 0.1t;$$

$$S_t = 6.5cos(\pi/60)$$

and

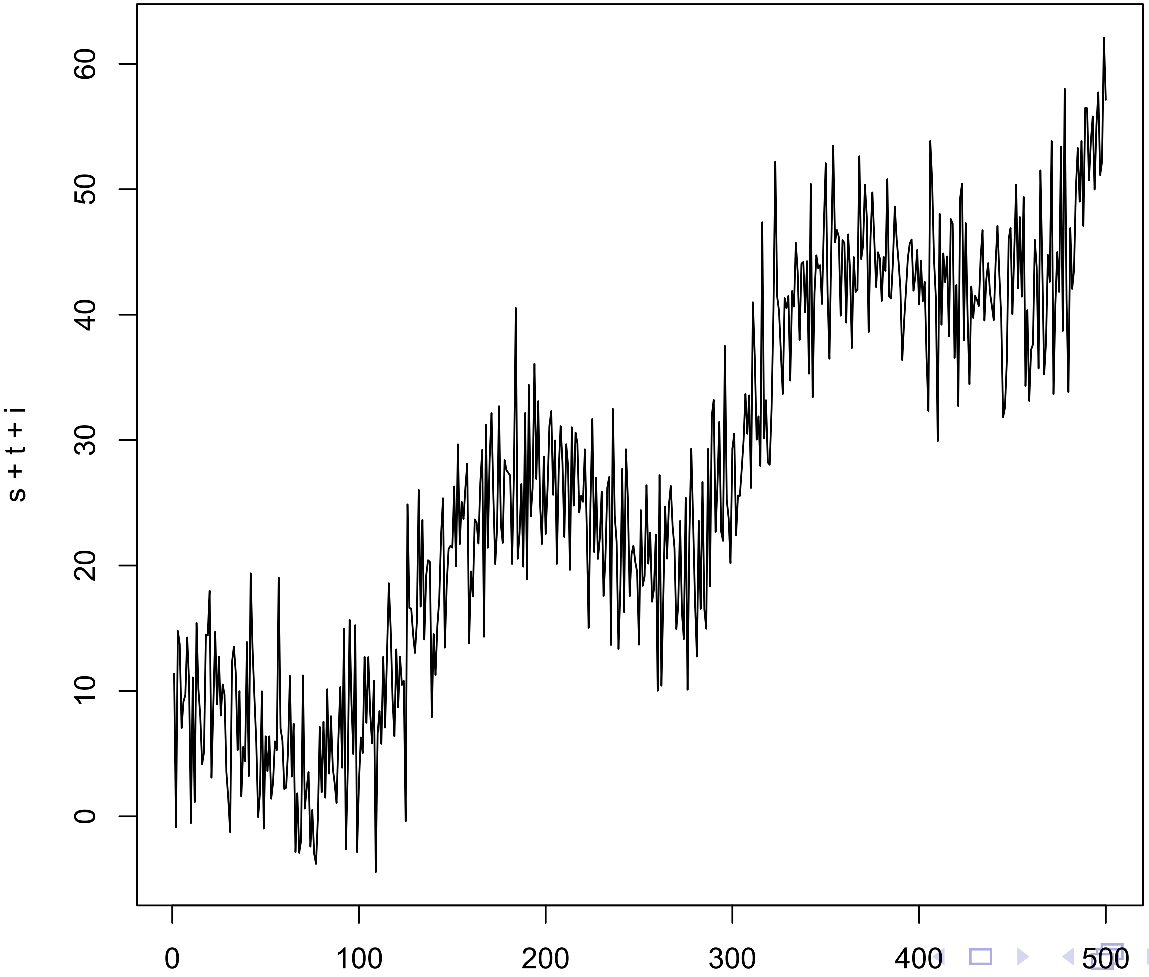
$$I_t \sim N(\mu = 0, \sigma^2 = 0.5).;$$

Time series decomposition

R Code

```
rm(list=ls())
t = 2 + 0.1 * 1 : 500
s = 6.5 * cos(pi * 1 : 500/90)
set.seed(154)
i = rnorm(500, 0, 5)
plot.ts(s + t + i)
```

Time series decomposition



Time series decomposition

- In general, time series can contain one or a combination of all the above elements, either additively or multiplicatively:

$$x_t = T_t + S_t + I_t$$

$$x_t = T_t * S_t * I_t$$

- The first specification is characterized by having each component independently, which makes it possible to decompose the series into a sum of the three factors.
- The second specification, on the other hand, arises when the trend (T_t), seasonality (S_t), and irregularity (I_t) are dependent on each other.

Time series decomposition

- In general, the trend changes the mean of the series, while the seasonal component has a pattern that repeats, for example on a monthly or quarterly basis. The irregular component, in spite of not having a well-defined pattern, can be forecast; in fact, forecasters use correlations with the irregular component to make their forecasts. In longer periods, however, the irregular component exhibits a tendency to revert to zero.
- Time series forecasting then attempts to predict each of these components individually. As we have seen, the global time series forecast groups each of these components in an additive or multiplicative way.

Trend Decomposition - Hodrick -Prescott Filter

- In economics, the Hodrick-Prescott (HP) filter allows to separate the trend and cyclic components for x_t.
- This method consists of obtaining a smoothed series S_t from the original x_t, by means of a solution to the optimization problem suggested in the following equation. Once solved, it allows to estimate both the cycle and the trend of the series.

$$min \sum_{t=1}^{n} (x_t - S_t)^2 + \lambda \sum_{t=2}^{n-1} [(S_{t+1} - S_t) - (S_t - S_{t-1})]^2$$

- The suggested values for λ depend on the periodicity of x_t, and are: 14400 (monthly), 1600 (quarterly) and 100 (yearly). On the other hand, once the cyclic component $(x_t - S_t)$ is obtained from the HP filter, it can be interpreted as the gap between its actual value x_t and potential S_t

Seasonal component decomposition
Difference Transformations

- Difference transformations are used to capture the seasonal component of the series:
  A first difference is defined as:

$$\triangle x_t = (x_t - x_{t-1})$$

- Logically, the second difference is defined as:

$$\triangle^2 x_t = (x_t - x_{t-1}) - (x_{t-1} - x_{t-2})$$

- Previously we saw that the first difference of the logarithm, could be interpreted as the percentage change of the variable, achieving the symmetry of the stock price.

Seasonal component decomposition
Dummy Variables

A dummy variable, D, is a binary variable that takes the following form:

- D=1 if the observation has specific characteristics.
- D=0 if it does not have them.

For example:

$$x_t = \beta_0 + \beta_1 z_t + \beta_2 D + \beta_3 D z_t$$

$$D = 1 => x_t = (\beta_0 + \beta_2) + (\beta_1 + \beta_3) z_t$$

$$D = 0 => x_t = \beta_0 + \beta_1 z_t$$

Dummy variables can be used to change the slope and/or intercept in a linear model, which allows capturing seasonality in the series, for example with dummy variables by quarter or season.

Dependency measures

- As we saw earlier, a time series can be viewed as a collection of n random variables at arbitrary integer time points t_1, t_2, t_n, for any positive integer there is a joint distribution function, evaluated as the probability that the values of the series are jointly less than n constants, $c_1, c_2, \cdots, c_n$, i.e.:

$$F(c_1, c_2, \cdots, c_n) = P(x_{t_1}, \leq c_1, x_{t_2} \leq c_2, \cdots x_{t_n} \leq c_n)$$

Dependency measures

- Unfortunately, the multinomial distribution function cannot usually be written easily unless the variables are jointly normal, in which case the joint density has the form:

$$f(\mathbf{x}) = (2\pi)^{-2/n} \mid \Gamma \mid^{-1/2} exp\{-1/2(\mathbf{x}-\mu)'\Gamma^{-1}\ (\mathbf{x}-\mu)\}$$

- where $\mid \cdot \mid$ indicates determinant and Γ the covariance matrix.
- Although the joint distribution function allows the data to be fully described, its manipulation is very complex, and its graphical display impossible.

Dependency measures

- The marginal distribution functions:

$$F_t(x) = P\{x_t \leq x\}$$

- or the corresponding marginal density function.

$$f_t(x) = \frac{\partial F_t(x)}{\partial x}$$

- When they exist, they provide valuable information for examining the marginal behavior of the series.
- If x_t is Gaussian with mean μ_t y varianza σ_t^2, $x_t \sim N(\mu_t, \sigma_t^2)$, marginal density is given by:

$$f_t(x) = \frac{1}{\sigma_t \sqrt{2\pi}} exp(-\frac{1}{2\sigma_t^2}(\mathbf{x} - \mu_t)^2)$$

Dependency measures

- The mean function, known in statistics as the first central moment, is defined as:

Definition

$$\mu_{xt} = E(x_t) = \int_{-\infty}^{+\infty} x f_t(x) dx$$

- E denotes **the expected value operator**.

Dependency measures

- The **autocovariance function**, known in statistics as the second central moment, is defined as:

Definition

$$\gamma_x(s,t) = cov(x_s, x_t) = E[(x_s - \mu_s)(x_t - \mu_t)]$$

- In this case, $\gamma_x(s,t) = \gamma_x(t,s)$ for all points of s and t. If $\gamma(s,t) = 0$ we can ensure that x_s and x_t are not linearly related. On the other hand, if x_s and x_t are besides bivariate normal, we can assure that they are independent.

Dependency measures

- It is clear that if $s = t$, the autocovariance is reduced to **variance**

Definition

$$\gamma_x(t, t) = E[(x_t - \mu_t)^2] = var(x_t)$$

Dependency measures

- The **Autocorrelation Function, denoted by ACF**, measures the linear predictability of the series in time t. That is, we predict x_t, using only the value x_s. Assuming that both series have finite variances, we have the following definition:

Definition

$$\rho(s,t) = \frac{\gamma(s,t)}{\sqrt{\gamma(s,s)\gamma(t,t)}}$$

- It can be easily shown that $-1 \leq \rho(s,t) \leq 1$. If we can predict xt perfectly from xs through a linear relationship, $x_t = \beta_0 + \beta_1 x_s$, then the correlation will be $+1$ when $\beta_1 > 0$, and -1 when $\beta_1 < 0$.
- Thus, we have an approximate measure of the ability to

Dependency measures

- The **cross-covariance function** between two series, x_t e y_t, is given by:

Definition

$$\gamma_{xy}(s,t) = cov(x_s, y_t) = E[(x_s - \mu_{xs})(y_t - \mu_{yt})].$$

- The **cross-correlation function (CCF)** is given by:

Definition

$$\rho_{xy}(s,t) = \frac{\gamma_{xy}(s,t)}{\sqrt{\gamma_x(s,s)\gamma_y(t,t)}}$$

Dependency measures

- We can easily extend the above formulations to the case of two-series measures, e.g., $x_{t1}, x_{t2}, \cdots, x_{tr}$, i.e., multivariate time series with r components.
- In this case, the case of cross-covariance of the extension is:

$$\gamma_{xy}(j,k) = cov(x_{sj}, y_{sk}) = E[(x_s - \mu_{xs})(y_{tj} - \mu_{y_{tk}})].$$

Stationarity

Definition

A time series is strictly stationary if the probabilistic behavior of each set of values $\{x_{t_1}, x_{t_2}, ..., x_{t_k}\}$ eis identical to that of the same set displaced in time, i.e. $\{x_{t_1+h}, x_{t_2+h}, ..., x_{t_k+h}\}$.

In other words:

$$P(x_{t_1}, \leq c_1, \cdots x_{t_k} \leq c_k) = P(x_{t_1} + h, \leq c_1 \cdots x_{t_k+h} \leq c_k)$$

for all $k = 1, 2, ...,$ all periods $t_1, t_2, ..., t_k$, all numbers $c_1, c_2, ..., c_k$, and all time shifts $h = 0, \pm 1, \pm 2, ...$

Stationarity

- If a time series is strictly stationary, then all multivariate distribution functions for subsets of variables must be equal to their counterparts in the shifted set. For example, when $k = 1$:

  $$P\{x_s \leq c\} = P\{x_t \leq c\}$$

  for any point in time s y t.
- When $k = 2$ we have:

  $$P\{x_s \leq c_1, x_t \leq c_2\} = P\{x_{s+h} \leq c_1, x_{t+h} \leq c_2\}$$

- For any point of s, t and h. If the variance function exists, then: $gamma(s, t) = gamma(s + h, t + h)$.
- In this context, is a random walk with drift strictly stationary?

Stationarity

A weakly stationary time series x_t is a finite variance process such that:

Definition

(i) the mean-value function, μ_t, is constant and does not depend on time t, and (ii) the autocovariance function, $\gamma(s, t)$, depends on s and t only through their difference $|s - t|$.

From now on, we will use the term stationary to mean weakly stationary; if a process is strictly stationary, we will use the term strictly stationary. An important case in which stationarity implies strict stationarity is if the series is Gaussian (i.e., all finite distributions of the series are Gaussian).

Multiple linear regression in time series models

Next we introduce (remember) the classical linear regression model.

- Let **X** be a matrix of $n \times k$ entries where we have n observations for k independent variables.
- Let **Y** be a vector of n observations of the dependent variable.
- It is possible to propose a linear estimation model that relates the independent variables to the dependent variable **X** and the variable **Y**:

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & X_{11} & X_{21} & \cdots & X_{k1} \\ 1 & X_{12} & X_{22} & \cdots & X_{k2} \\ \vdots & \vdots & \ddots & \vdots & \\ 1 & X_{1n} & X_{2n} & \cdots & X_{kn} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_n \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

Multiple linear regression in time series models

The model can also be written in a compact form as:

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon$$

We see that this model presents systematic (deterministic) components ($\mathbf{X}\beta$) and stochastic (ϵ). The objective is to determine the coefficients β_i that linearly relate the variables X_i and Y. For this we use the method of ordinary least squares (OLS). The least squares criterion seeks to minimize the sum of the squares of the residuals: $Min \sum e^2$.

Multiple linear regression in time series models

- Next, we review the derivation of the OLS method. First, the vector of residuals can be obtained as:

$$e = Y - X\hat{\beta}$$

- Where $\hat{\beta}$ represents the vector estimator β.
- Thus the sum of the square of the errors will be:

$$\mathbf{e}'\mathbf{e} = \begin{bmatrix} e_1 & e_2 & \dots & e_n \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} = \begin{bmatrix} e_1^2 + e_2^2 + \dots + e_n^2 \end{bmatrix}$$

Multiple linear regression in time series models

- On the other hand, it can also be written as:

$$\begin{aligned}\mathbf{e}'\mathbf{e} &= \left(\mathbf{Y}-\mathbf{X}\hat{\beta}\right)'\left(\mathbf{Y}-\mathbf{X}\hat{\beta}\right)\\ &= \mathbf{Y}'\mathbf{Y}-\hat{\beta}'\mathbf{X}'\mathbf{Y}-\mathbf{Y}'\mathbf{X}\hat{\beta}+\hat{\beta}'\mathbf{X}'\mathbf{X}\hat{\beta}\\ &= \mathbf{Y}'\mathbf{Y}-\mathbf{2}\hat{\beta}'\mathbf{X}'\mathbf{Y}+\hat{\beta}'\mathbf{X}'\mathbf{X}\hat{\beta}\end{aligned}$$

- To minimize the square of the residuals we resort to differential calculus:

$$\frac{\partial(\mathbf{e}'\mathbf{e})}{\partial\hat{\beta}} = -2\mathbf{X}'\mathbf{Y}+2\mathbf{X}'\mathbf{X}\hat{\beta}=0$$

- Then $\hat{\beta}$ will be a minimum of $\mathbf{e}'\mathbf{e}$ if the second derivative is positive or equivalently $\mathbf{X}$ is positive definite.

Multiple linear regression in time series models

- From the previous expression we obtain:

$$\mathbf{X}'\mathbf{X}\hat{\beta} = \mathbf{X}'\mathbf{Y}$$

- Finally, multiplying by $(\mathbf{X}'\mathbf{X})^{-1}$ on both sides, we obtain $\hat{\beta}$:

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$$

- The ease of OLS calculation has influenced its popularity. The estimators are obtained through simple matrix operations.

Properties of OLS estimators

- The OLS estimation is the best linear unbiased estimator (BLUE). The proof of this proposition is provided by the Gauss-Markov theorem.

1. Unbiased: $E(\hat{\beta}) = \beta$, i.e. the expected value of the estimator is the true value of the unknown parameter.
2. Best: Minimum variance.

Gauss-Markov Theorem

Assumptions

- There is a linear relationship between $\mathbf{X}$ and $\mathbf{Y}$
- No multicollinearity ($\mathbf{X}$ is linearly independent)
- $E(\epsilon|\mathbf{X}) = 0$. Equivalently $E(\mathbf{Y}) = \mathbf{X}\beta$
- $E(\epsilon\epsilon'|\mathbf{X}) = \sigma^2\mathbf{I}$. Errors are homocedastic and there is no autocorrelation.
- $\mathbf{X}$ and ϵ are unrelated. $Cov(\mathbf{X}\epsilon) = 0$

Usually, all of these assumptions are checked in the diagnostic testing process.

Theorem Proof

OLS estimators are the best linear unbiased estimators for β (BLUE)

- $\hat{\beta}$ es un estimador insesgado de β.

$$\begin{aligned}\hat{\beta} &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}\\ &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\left(\mathbf{X}\beta+\epsilon\right)\\ &= \beta+(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\epsilon\end{aligned}$$

$$\begin{aligned}E\left(\hat{\beta}\right) &= E\left(\beta+(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\epsilon\right)\\ &= E\left(\beta\right)+E\left((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\epsilon\right)\\ &= \beta+(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'E(\epsilon)\\ &= \beta\end{aligned}$$

Time Series Regression

- $\hat{\beta}$ is a linear estimator of β.

$$\begin{aligned} \hat{\beta} &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} \\ &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\left(\mathbf{X}\beta + \epsilon\right) \\ &= \beta + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\epsilon \\ &= \beta + A\epsilon \end{aligned}$$

- Prove that it is a minimum variance estimator.

Statistical Evaluation of Estimated Regressions

- **The coefficient of determination** R^2 is a measure of goodness of fit, the degree to which the independent variables jointly explain the variation in the dependent variable over its mean. R^2 increases each time the number of regressors, k, increases, relative to the sample size, n, regardless of the theoretical justification for including additional variables. In the limit, if $n = k + 1$, $R^2 = 1$ but such a regression has zero explanatory power.

Statistical Evaluation of Estimated Regressions

- **Adjusted** R^2 ttakes into account the number of regressors relative to the sample size. The adjusted R^2 is particularly useful to evaluate the relative fit of a set of regressions estimated for the same dependent variable but with a different number of independent variables. A mechanical criterion for model selection is to maximize the adjusted R^2.

Statistical Evaluation of Estimated Regressions

- **Test t**
- The t-tests are hypothesis tests on the estimated parameters to determine whether they are individually significantly different from zero. Null hypothesis: H0: $\beta_j = 0$.

$$\frac{\hat{\beta}_j - \beta_j}{SE(\hat{\beta}_j)} = \frac{\hat{\beta}_j - 0}{SE(\hat{\beta}_j)} = \frac{\hat{\beta}_j}{SE(\hat{\beta}_j)} \sim t(n-k-1)$$

$SE(\hat{\beta}_j)=$ standard error of the estimated parameter

Nonsense regression and spurious

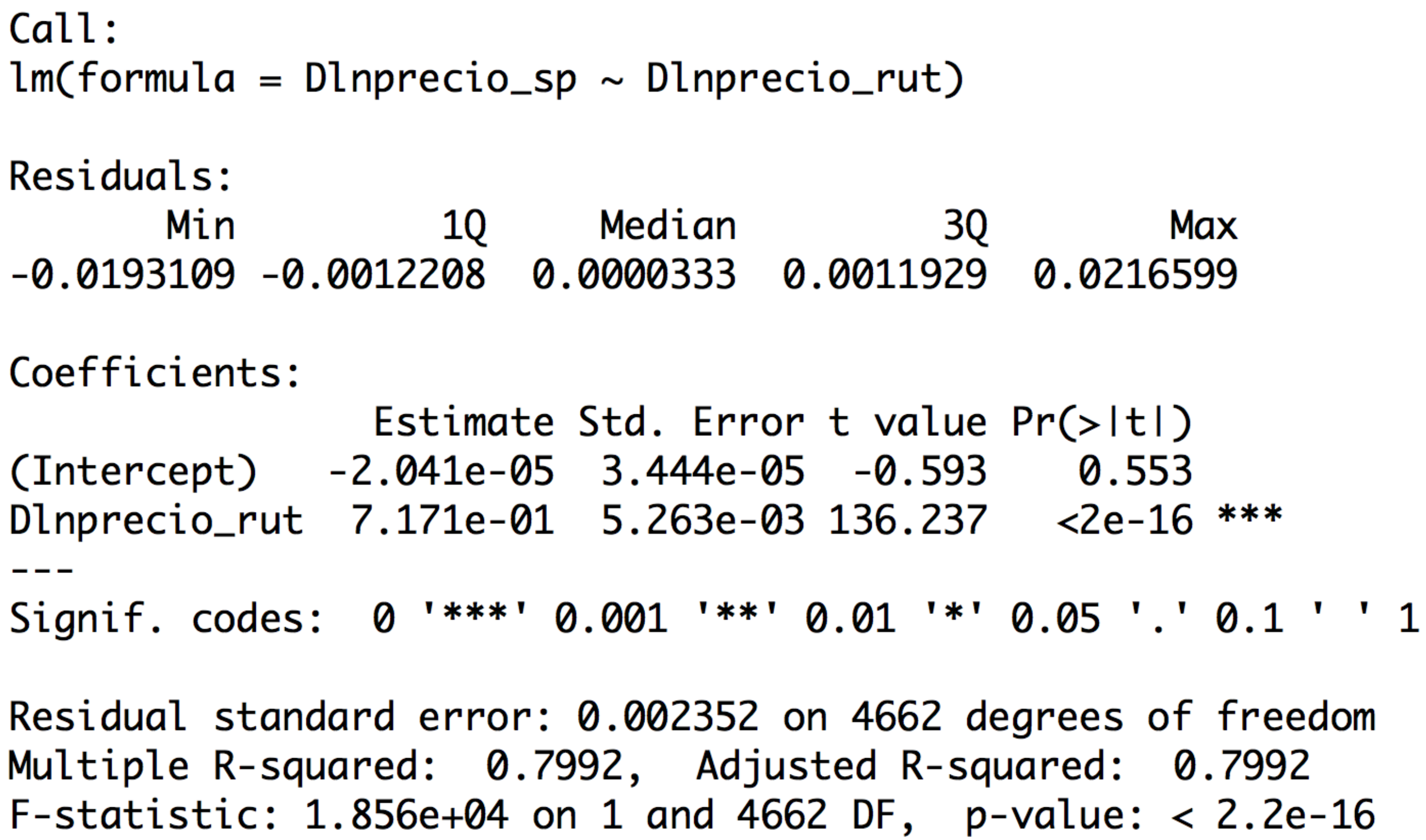
- **Nonsense regressions**, are mutually independent time series that produce good indicators in the regression, due to the high level of serial correlation in each series.

- **Spurious regressions** occur when the data depend on a third common factor, for example: a time trend. The spurious relationship gives the impression that there is a statistical link between two variables, which is invalidated when examined objectively.

Example 3: Regressions

- As a further example, we calculate the beta of a financial asset. The beta is a measure of systematic risk, which is measured with respect to the relation of the returns of the asset, with those of the diversified index of the market, in this case the S&P index already studied.
- In this case we calculate the beta of another index, the Russell 2000 (denoted by RUT) which measures the performance of small cap companies.

Example 3: Regressions

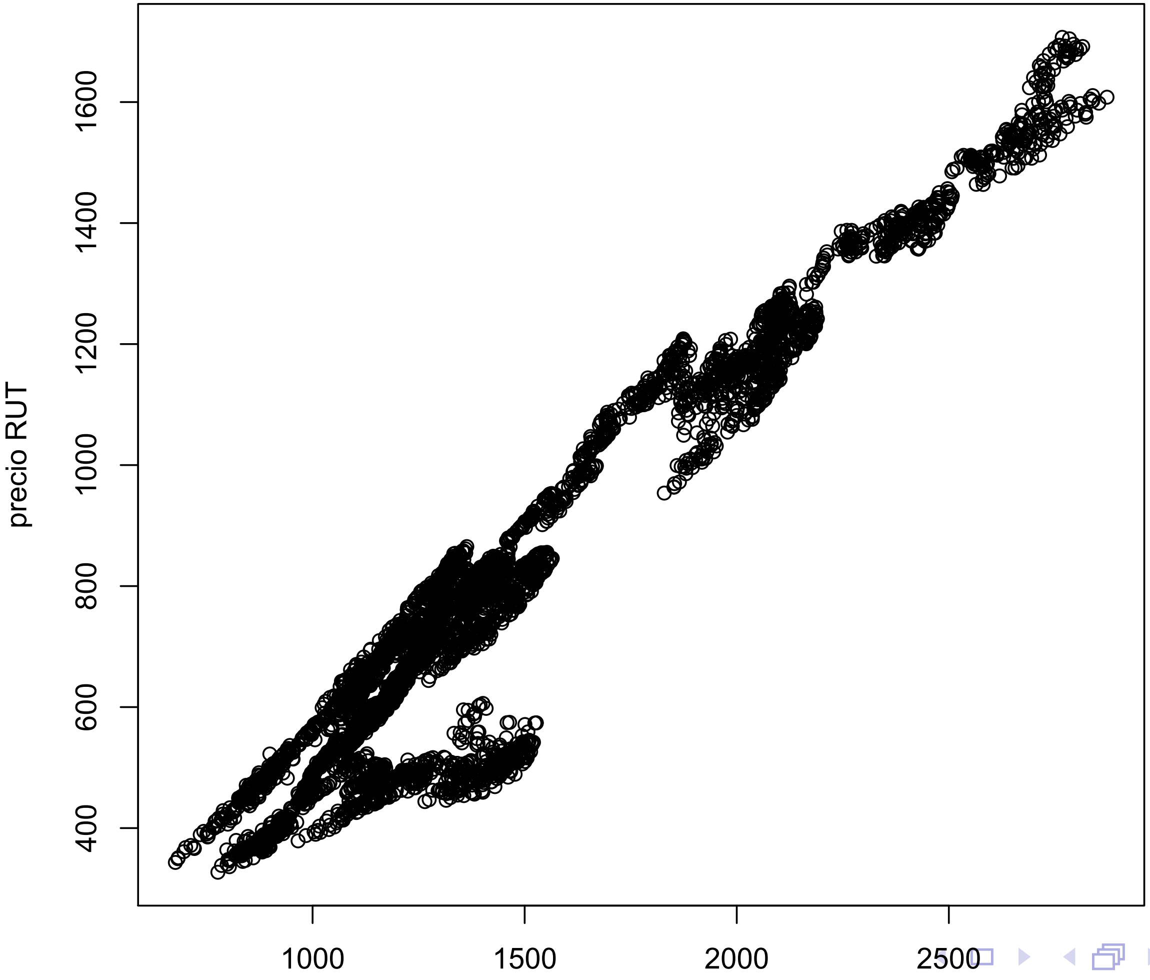


```
Call:
lm(formula = Dlnprecio_sp ~ Dlnprecio_rut)

Residuals:
       Min         1Q     Median         3Q        Max 
-0.0193109 -0.0012208  0.0000333  0.0011929  0.0216599 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)   -2.041e-05  3.444e-05  -0.593    0.553    
Dlnprecio_rut  7.171e-01  5.263e-03 136.237   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.002352 on 4662 degrees of freedom
Multiple R-squared:  0.7992,	Adjusted R-squared:  0.7992 
F-statistic: 1.856e+04 on 1 and 4662 DF,  p-value: < 2.2e-16
```

R Code

```
rm(list=ls())
mydata < − read.csv (“/Users/marcelovillena/Desktop/sp.csv”,
header = TRUE, stringsAsFactors = FALSE)
precio_sp < − mydata$“Adj.Close”: lnprecio sp < − log(precio_sp
) ;
Dlnprecio sp < − diff(lnprecio sp ,1)
mydata < − read.csv (“/Users/marcelovillena/Desktop/rut.csv”,
header = TRUE, stringsAsFactors = FALSE)
precio_rut < − mydata$“rut”; lnprecio rut < − log(precio rut ) ;
Dlnprecio rut < − diff(lnprecio rut ,1)
reg1 < − lm ( Dlnprecio sp˜ Dlnprecio rut)
summary(reg1)
```

Example 3: Regressions



Example 3: Regressions

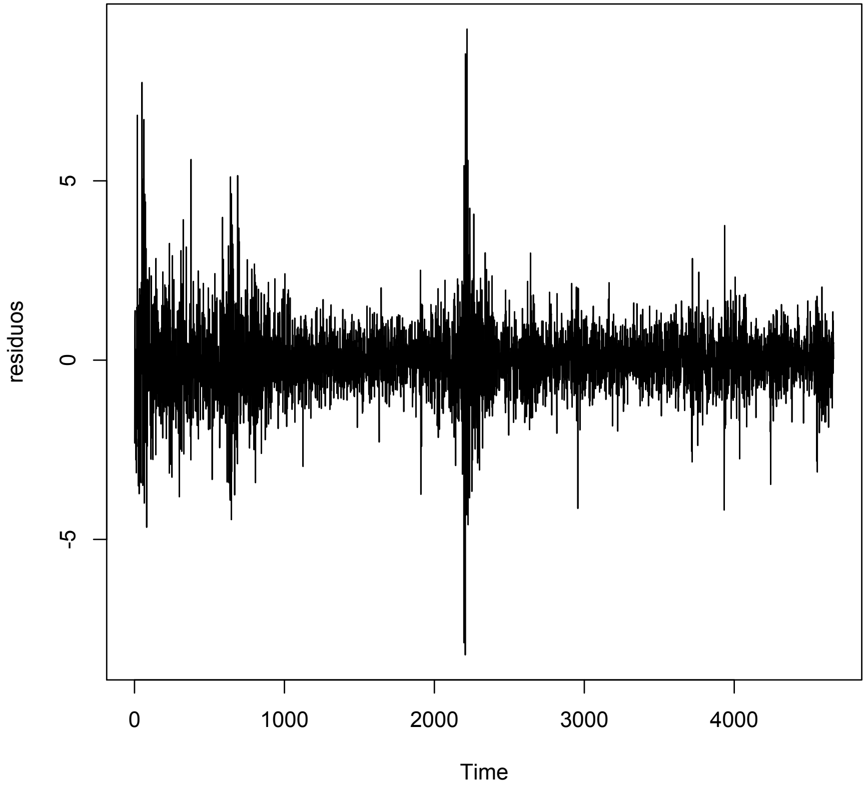
- From the following code we store the residuals and analyze the assumptions required for a good regression.

R Code

```
residuos < − rstandard(reg1)
valores.ajustados < − fitted(reg1)
plot(valores.ajustados, residuos)
qqnorm(residuos)
qqline(residuos
```

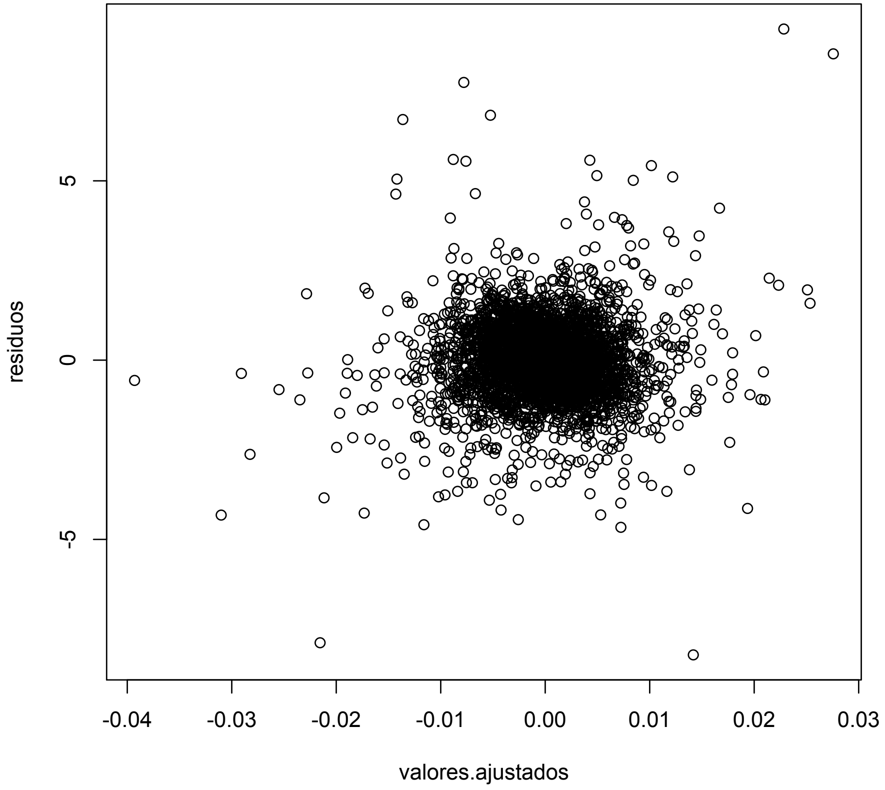
Example 3: Regressions

- Homocedasticity



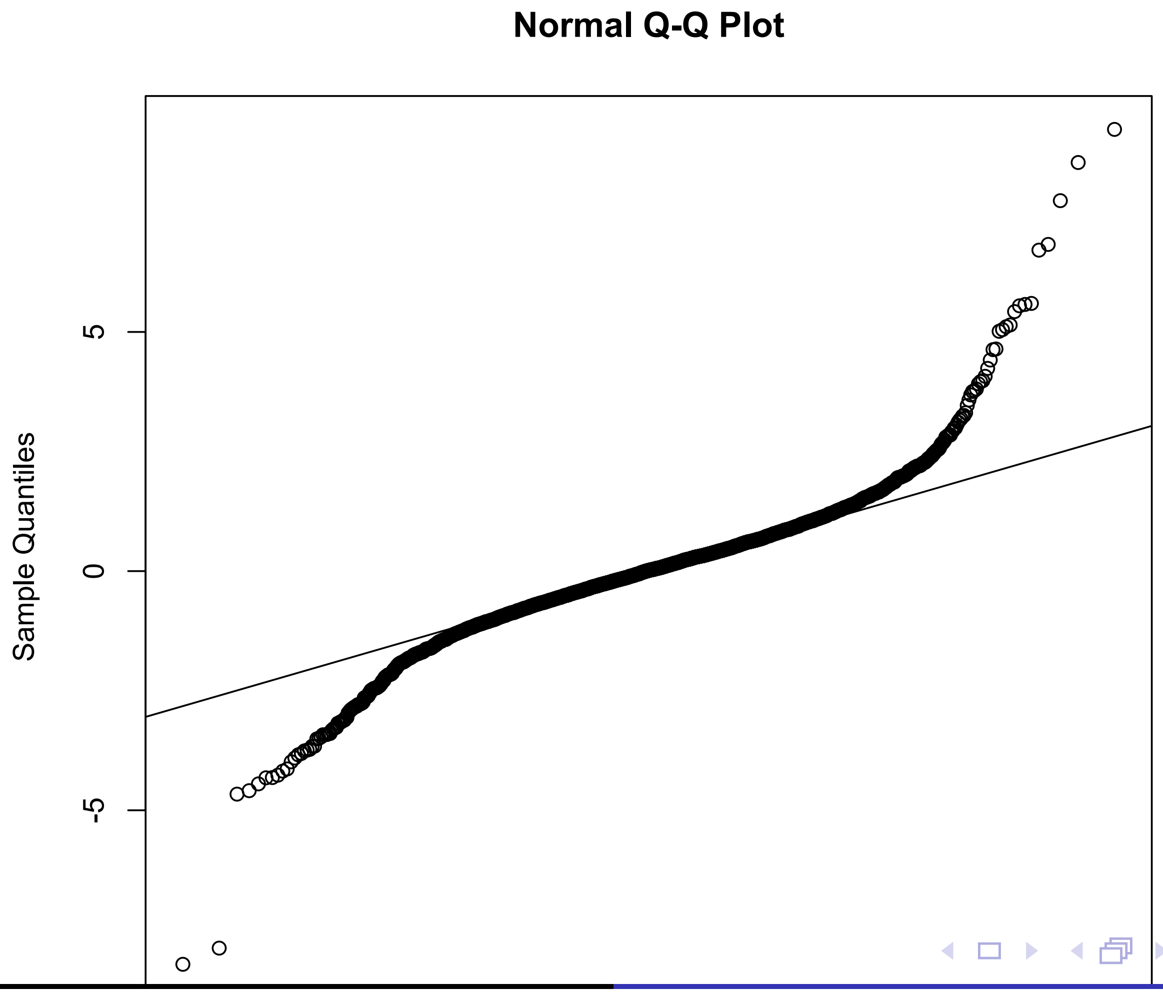
Example 3: Regressions

- Homocedasticity



Example 3: Regressions

- Normality of residuals



Homework 1

1. Our first task will be to install R and to collect different macro variables (at least 5) with different frequencies (daily, monthly, quarterly, yearly) of a country of your choice. The data will be used throughout the course.
2. Graph and comment on the data.
3. Obtain and comment on the descriptive statistics of the data.
4. Analyse the seasonality of the data.
5. Check if the variables are random walks.
6. Comment very briefly this paper: https://www.journals.uchicago.edu/doi/pdf/10.1086/654107 Campbell, J. Y., & Mankiw, N. G. (1989).

Assignments should always be presented in powerpoint, and should be accompanied by the data and code used.

References

[1] Ruey S Tsay. Analysis of financial time series. financial econometrics, a wiley-interscience publication, 2002.

[2] Robert H. Shumway and David S. Stoffer. Time series analysis and its applications with R examples, 3rd edn. Springer, 2011.

[3] James D Hamilton. Time series analysis. Princeton University Press, 1995.

[4] William H Greene. Econometric analysis. Pearson Education, 2003.

[5] Damodar N Gujarati. Basic econometrics. McGraw-Hill Education, 2009.

[6] John Y Campbell, Andrew W Lo, Archie Craig MacKinlay, et al. The econometrics of financial markets, volume 2. princeton University press Princeton, NJ, 1997.

Lecture II.- Time Series with Stationary Variables:

ARDL models

Marcelo Villena, PhD
Santa María University

September 12, 2022

Outline

1. Non-stationary integrated processes and the unit root test
   - On Detrending
   - On the decomposition of a series
   - Unit Root
2. Autoregressive and distributed lag models
   - Distributed lagged model
   - Ad hoc estimation
   - Koyck's method
   - Autoregressive Distributed Lag Models (ARDL)
3. Examples
   - Okun's Law
   - Phillips Curve
4. References

Example from the previous class... Detrending global temperature

- As we saw in the previous class, the evolution of the global temperature showed a linear trend, so we can assume that it can be written as:

$$x_t = \mu_t + y_t$$

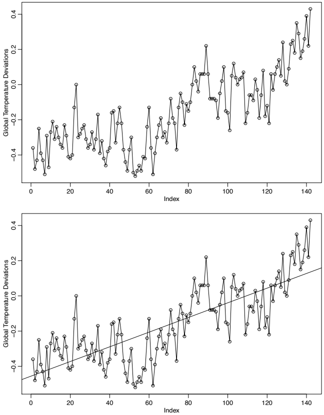
- We will see two ways of decomposing the series, “filtering” the trend.

Example from the previous class... Detrending global temperature

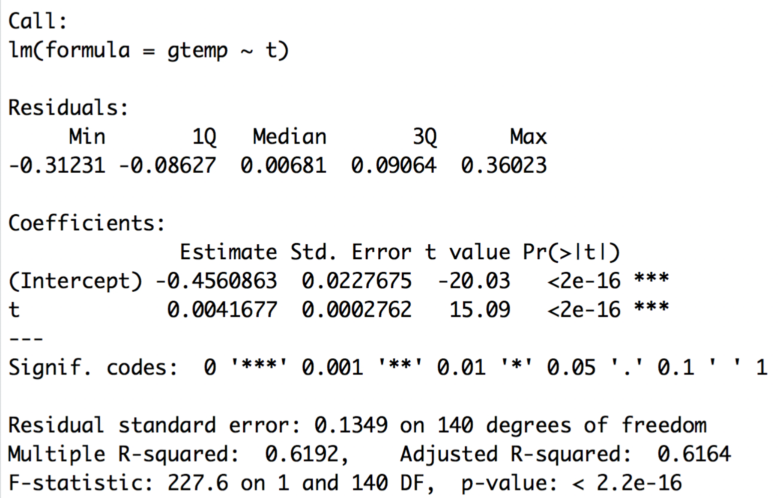
R Code

```
rm(list=ls())
mydata< −read.csv (“gtemp.csv”)
gtemp< −mydata$“gtem”
plot(gtemp, type=“o”, ylab= “Global Temperature Deviations”)
t< −1:142
summary(reg < − lm(gtemp˜ t))
plot(gtemp, type=“o”, ylab=“Global Temperature Deviations”)
abline(reg)
```

Example from the previous class... Detrending global temperature



Example from the previous class... Detrending global temperature



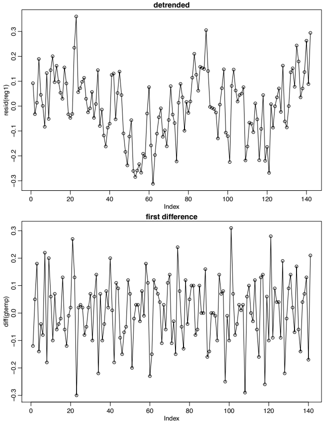
```
Call:
lm(formula = gtemp ~ t)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.31231 -0.08627  0.00681  0.09064  0.36023 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.4560863  0.0227675  -20.03   <2e-16 ***
t            0.0041677  0.0002762   15.09   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1349 on 140 degrees of freedom
Multiple R-squared:  0.6192,    Adjusted R-squared:  0.6164 
F-statistic: 227.6 on 1 and 140 DF,  p-value: < 2.2e-16
```

Example from the previous class... Detrending global temperature

R Code

```
reg1= lm(gtemp˜ time(gtemp), na.action=NULL)
par(mfrow=c(2,1))
plot(resid(reg1), type=“o”, main=“detrended”)
plot(diff(gtemp), type=“o”, main=“first difference”)
```

Example from the previous class... Detrending global temperature

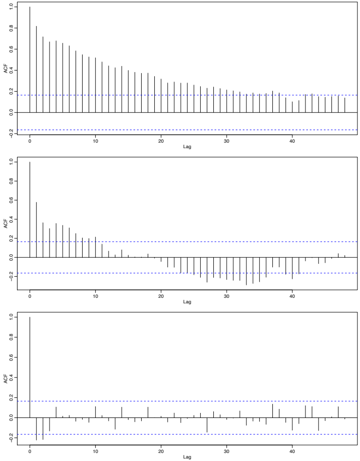


Example from the previous class... Detrending global temperature

R Code

```
par(mfrow=c(3,1))
acf(gtemp, 48, main=“gtemp”)
acf(resid(reg), 48, main=“detrended”)
acf(diff(gtemp), 48, main=“first difference”)
mean (diff (gtemp))
sd (diff (gtemp)) / sqrt (longitud (diff (gtemp)))
```

Example from the previous class... Detrending global temperature



On the decomposition of a series

- In the graphs we can appreciate that the first difference of the series produces different results than the trend removal by trend regression.
- In the case of the ACF graphs, the differenced process shows minimal autocorrelation, which may imply that the global temperature series is similar to a random walk with drift.
- It is interesting to note that this series could be seen as a random walk with drift.
- The mean of the differenced series, which is an estimate of the drift, is approximately ,0066, but with a large standard error.

On the decomposition of a series

- An advantage of differencing over the estimation of a trend, to eliminate trends, is that no parameters are estimated in the differencing operation. A disadvantage, however, is that differencing does not yield an estimate of the stationary process y_t.
- Thus, if an estimate of y_t is essential, then estimating a trend may be the most appropriate way to remove trends from the series. If the goal is to force the data to stationarity, then differencing may be more appropriate. Differencing is also a viable tool if the trend is fixed.
- In the U.S., the official decomposition and seasonal adjustment procedure is called ”seasonal adjustment. X-13-ARIMA
- **http://www.census.gov/srd/www/x13as/**

Non-stationary integrated processes and the unit root test

Recall that if a time series is stationary, its mean, variance and autocovariance (at different lags) remain the same regardless of the point in time at which they are measured, i.e. they are time invariant. On the other hand, we have seen that stationarity is a desirable characteristic, for example, in terms of the normality of the variables.

However, in practice we encounter:

1. **Non-stationary processes:** When a stochastic time series process is time-dependent.
2. **Integrated Processes:** a non-stationary process, which can be transformed to a stationary process by differentiating.

Integrated Processes

With respect to Integrated Processes, we start by defining:

- The sequence x_t is integrated of order d, $I(d)$, if it requires to be differentiated d times to become stationary.
- **All Integrated Processes are non-stationary, but not all non-stationary processes are integrated.**
- If the sequence x_t has a unit ration, then, it is an integrated process, and of that non-stationary.

Consequences of Integrated Processes (Unit Root)

- It is important to note that standard statistical tests are not appropriate when OLS is applied to integrated processes, see for example Granger 1974.
- If the sequence x_t is a unit ration process, then any shock has a permanent (non-decaying) effect. Hence, the time series is properly modeled by assuming a stochastic trend. The time series can then be defined as stationary differentiable, and the trend should be taken out by differentiating.
- In this context, **the terms non-stationarity, random walk, unit root and stochastic trend are considered synonymous.**

Is random walk nonstationarity?

$$y_t = y_{t-1} + \varepsilon_t$$

$$Var(y_t) = Var(y_{t-1} + \varepsilon_t)$$

$$Var(y_t) = Var(y_{t-1}) + \sigma_\varepsilon^2$$

$$Var(y_t) = Var(y_{t-2} + \varepsilon_{t-1}) + \sigma_\varepsilon^2$$

$$Var(y_t) = Var(y_{t-2}) + 2\sigma_\varepsilon^2$$

repeating this for t steps:

$$Var(y_t) = Var(y_0) + t\sigma_\varepsilon^2$$

If we assume that $y0$ is given:

$$Var(yt) = t\sigma_\varepsilon^2$$

The variance of the process increases with time, and therefore RW os not stationary.

Test of Unit Root

- Consider the following autoregressive process:

$$x_t = \alpha_1 x_{t-1} + \varepsilon_t \qquad (1)$$

- If $\alpha_1 = 1$, the sequence x_t is a unit root.
- The standard test to prove this hypothesis is to subtract x_{t-1} from the above equation such that:

$$\triangle xt = \gamma xt - 1 + \varepsilon t \qquad (2)$$

- where $\gamma = \alpha_1 - 1$, and $\triangle x_t = x_t - x_{t-1}$. In this context, proving the hypothesis that equation (1) has a unit ration, $\alpha_1 = 1$, is equivalent to proving the hypothesis of $\gamma = 0$ in equation (2).

Test of Unit Root

- This is basically **the Dickey-Fuller (DF) approach for unit roots**, see e.g., Dickey Fuller 1981.
- Additionally there is the **Augmented Dickey-Fuller test (ADF)**, and many other tests that are based on similar logic, which we will use during the course.

Example of Unit Root Test - Dickey-Fuller

R Code

```
install.packages(“tseries”)
library(tseries)
adf.test(gtemp)
adf.test(resid(reg1))
adf.test(diff(gtemp))
```

Example of Unit Root Test - Dickey-Fuller

```
Augmented Dickey-Fuller Test
data: gtemp
Dickey-Fuller = -2.0624, Lag order = 5, p-value = 0.5505
alternative hypothesis: stationary


Augmented Dickey-Fuller Test
data: resid(reg1)
Dickey-Fuller = -2.0624, Lag order = 5, p-value = 0.5505
alternative hypothesis: stationary


Augmented Dickey-Fuller Test
data: diff(gtemp)
Dickey-Fuller = -6.8179, Lag order = 5, p-value = 0.01
alternative hypothesis: stationary
```

Distributed lagged model

- In regression analysis with time series data, when the regression model includes not only current values but also lagged (past) values of the explanatory variables (the X's), it is called a distributed lagged model.
- If the model includes one or more lagged values of the dependent variable among its explanatory variables, it is called an autorregressive model.

Distributed lagged model

Thus,

$$Y_t = \alpha + \beta_0 X_t + \beta_1 X_{t-1} + \beta_2 X_{t-2} + u_t$$

represents a distributed lagged model, while

$$Y_t = \alpha + \beta X_t + \gamma Y_{t-1} + u_t$$

is an example of an autoregressive model. The latter are also known as dynamic models, since they indicate the trajectory over time of the dependent variable relative to its past value(s).

Distributed lagged model

- More generally, we would write
$$Y_t = \alpha + \beta_0 X_t + \beta_1 X_{t-1} + \beta_2 X_{t-2} + \cdots + \beta_k X_{t-k} + u_t$$
- which is the distributed lags model with a finite lag of k periods. The coefficient β_0 is known as the short-run or impact multiplier because it gives the change in the mean value of Y that follows a unit change in X in the same period.1
- Technically, β_0 is the partial derivative of Y with respect to X_t, β_1 with respect to X_{t-1}, β_2 with respect to X_{t-2}, and so on. Symbolically, $\frac{\partial Y_t}{\partial X_{t-k}} = \beta_k$.

Distributed lagged model

- If the change in X remains the same from the beginning, then $(\beta_0 + \beta_1)$ gives the change in (the mean value of) Y in the next period $(\beta_0 + \beta_1 + \beta_2)$ in the one that follows, and so on. These partial sums are denoted as interim, or intermediate, multipliers. Finally, after k periods we obtain:
$$\sum \beta i = \beta_0 + \beta_1 + \beta_2 + ... + \beta_k = \beta$$
- which is known as the long-run or total distributed lag multiplier, provided that the sum β exists (we will explain this later). If we define
$$\beta_i^* = \frac{\beta i}{\sum \beta i} = \frac{\beta i}{\beta}$$
- we obtain "standardized" β_i. The partial sums of the standardized β_i give the proportion of the long-run, or total, impact felt during a certain period.

Distributed lagged model

- **Autoregressive and distributed lag models are very common in economic analysis**.
- We will study them in detail in order to find out the following:

1. What is the role of lags in economics?
2. On what grounds are lags justified?
3. Is there any theoretical justification for the lagged models common in empirical econometrics?

Distributed lagged model

4. What is the relationship, if any, between autoregressive models and distributed lag models, and can they be derived from each other?
5. What are some statistical problems related to the estimation of such models?
6. Does the lagged-ahead relationship between variables imply causality? If so, how is it measured?

On the the nature of lagged phenomena

1. **Psychological reasons.** As a result of force of habit (inertia), people do not change their consumption habits immediately after a price reduction or an increase in income, perhaps because the process of change entails some immediate disadvantage.
2. **Technological reasons.** Suppose that the price of capital relative to labor is reduced, so that it is economically feasible to substitute labor for capital. Of course, the addition of capital takes time (gestation period). Moreover, if the price fall is expected to be temporary, firms may not rush to substitute labor for capital, especially if they expect that after the temporary fall the price of capital may rise beyond its previous level. Sometimes, imperfect knowledge also explains lags.

On the the nature of lagged phenomena

3. **Institutional reasons.** These reasons also contribute to lags. For example, contractual obligations may prevent companies from switching from one source of labor or raw materials to another. For example, those who placed funds in long-term savings accounts with fixed terms, such as one, three or seven years, are "locked in," even though money market conditions now allow higher returns elsewhere.

Estimation of distributed lag models

- We already established that distributed lag models play a very useful role in economics, but how do we estimate such models?
- Suppose we have the following model of distributed lags for one explanatory variable:

$$Y_t = \alpha + \beta_0 X_t + \beta_1 X_{t-1} + \beta_2 X_{t-2} + \ldots + u_t$$

- where we have not defined the lag length, i.e., how far back in the past we wish to go. Such a model is called an infinite lag model, while a model of the type shown above is called a finite distributed lag model because the lag length k is specified.

Estimation of distributed lag models

How do we estimate α and the β's of this equation? We can adopt two approaches:

1. **ad hoc estimation** and
2. **a priori constraints on the** $\beta's$, if we assume that (the $\beta's$) follow a systematic pattern.

Ad hoc estimation of distributed lag models.

- Since the explanatory variable X_t is assumed to be nonstochastic (or at least uncorrelated with the disturbance term u_t), equally nonstochastic are X_{t-1}, X_{t-2}, and so on. Therefore, in principle, the ordinary least squares (OLS) method is applicable.
- This is the approach of **Alt** and **Tinbergen** who suggest that to estimate a distributed lag models, we proceed sequentially, i.e., first regress Y_t on X_t, then regress Y_t on X_t and X_{t-1}, then regress Y_t on X_t , X_{t-1}, and X_{t-2},and so on. This sequential process stops when the regression coefficients of the lagged variables start to become statistically insignificant and/or the coefficient of at least one variable changes its sign from positive to negative, or vice versa.

Ad hoc disadventages.

Although ad hoc estimation seems straightforward and unobtrusive, it has many disadvantages, including the following:

1. There is no a priori guidance on the maximum length the lag should be.
2. As successive lags are estimated, fewer degrees of freedom remain, thus weakening statistical inference somewhat.
3. More importantly, in economic time series data, successive (lagged) values tend to be highly correlated; thus, multi-linearity comes to the fore.

Koyck's method for distributed lag models

- Koyck proposed a different method of estimating distributed lag models. Suppose we start with an infinite distributed lags model. If all β have the same sign, Koyck assumes that they reduce geometrically as follows.

$$\beta_k = \beta_0 \lambda^k \qquad k = 0, 1, ...$$

- where λ, such that $0 < \lambda < 1$ is known as the rate of decline, or decay, of the distributed lag, and where $1 - \lambda$ is known as the rate of adjustment.
- What the model postulates is that each successive β coefficient is numerically lower than each previous β (this statement is due to the fact that $\lambda < 1$), implying that, as one returns to the distant past, the effect of that lag on Y_t becomes progressively smaller, a very reasonable assumption.

Koyck's method for distributed lag models

Note these features of Koyck's scheme:

1. by assuming non-negative values for λ, Koyck eliminates the possibility that the β will change sign;
2. by assuming that $\lambda < 1$, he gives less weight to β in the distant past than today; and
3. It ensures that the sum of the β, which provides the long-run multiplier.

The proof of Koyck's model

The proof of Koyck's model is fairly straightforward. Given the following distributed lags model:

$$Y_t = \alpha + \beta_0 X_t + \beta_1 X_{t-1} + \beta_2 X_{t-2} + ... + u_t$$

Applying the transformation $\beta_k = \beta_0 \lambda^k$ we have:

$$Y_t = \alpha + \beta_0 X_t + \beta_0 \lambda^1 X_{t-1} + \beta_0 \lambda^2 X_{t-2} + ... + u_t$$

Then we lag the equation by one period:

$$Y_{t-1} = \alpha + \beta_0 X_{t-1} + \beta_0 \lambda^1 X_{t-2} + \beta_0 \lambda^2 X_{t-3} + ... + u_t$$

The proof of Koyck's model

Then we multiply by λ:

$$\lambda Y_{t-1} = \lambda\alpha + \lambda\beta_0 X_{t-1} + \lambda\beta_0\lambda^1 X_{t-2} + \lambda\beta_0\lambda^2 X_{t-3} + ... + u_t$$

By subtracting both equations we have:

$$Y_t - \lambda Y_{t-1} = \alpha(1-\lambda) + \beta_0 X_t + (u_t - \lambda u_{t-1})$$

By reordering we have:

$$Y_t = \alpha(1-\lambda) + \beta_0 X_t + \lambda Y_{t-1} + v_t$$

where $v_t = (u_t - \lambda u_{t-1})$ is a moving average of u_t and u_{t-1}.

On the problems of Koyck's model

- The presence of autocorrelation can lead to misleading results as they violate the assumptions of the Gauss Markov Theorem.
- However, in the presence of correlated errors, we can still proceed to fit a model makes up for these violations.
- Considering this, the least squares estimator is no longer unbiased, but it does have the desirable large sample property of consistency, and if the errors are normally distributed, it is best in a large sample sense.
- It is also important to note that is assumed to be uncorrelated random errors with zero mean and constant variance.
- **In such cases, the time series assumption that the error term is independent of current, past, and future values of is no longer valid.**

Least Squares Estimation

In the presence of serially correlated errors, the consequences of least squares estimation are similar to the consequences of ignoring the presence of heteroskedasticity, namely

1. The least squares estimator is still a linear unbiased estimator, but is no longer best.
2. The formulas for the standard errors usually computed for the least square estimator are no longer correct. Although the usual least squares standard errors are not the correct ones, it is possible to compute correct standard errors for the least squares estimator when the errors are serially correlated. These standard errors are known as HAC (heteroskedasticity and autocorrelation consistent) standard errors, or Newey-West standard errors, and they are analogous to heteroskedasticity consistent, or **White, standard errors**.

Adaptive expectations

- Suppose we postulate the following model:

$$Y_t = \beta_0 + \beta_1 X_t^* + u_t$$

- For example, let us assume that $Y =$ demand for money (real cash balances) $X^* =$ normal or expected long term or equilibrium interest rate, u optimal $u =$ error term.
- Since the expectations variable X^* is not directly observable, we can propose the following hypothesis on how expectations are shaped:

$$X_t^* - X_{t-1}^* = \gamma(X_{t-1} - X_{t-1}^*)$$

- where γ, such that $0 < \gamma \leq 1$, is known as the expectations coefficient. This hypothesis is known as the adaptive expectations, progressive expectations or learning-by-error hypothesis, popularized by **Cagan and Friedman.**

Adaptive expectations

- Replacing:

$$Y_t = \beta_0 + \beta_1\gamma X_t + \beta_1(1-\gamma)X^*_{t-1} + u_t$$

- Now lagging our original equation one period, multiply it by $1-\gamma$, and subtracting the equation above, we obtain

$$Y_t = \gamma\beta_0 + \gamma\beta_1 X_t + (1-\gamma)Y_{t-1} + v_t$$

- where $v_t = u_t - (1-\gamma)u_{t-1}$
- Similar to the Koyck model!

$$Y_t = \alpha(1-\lambda) + \beta_0 X_t + \lambda Y_{t-1} + v_t$$

Adaptive expectations

- Until the rational expectations (RE) hypothesis, first put forward by J. Muth and later disseminated by Robert Lucas and Thomas Sargent, the adaptive expectations (AE) hypothesis was very popular in empirical economics.
- **Proponents of the RE hypothesis argue that the RE hypothesis is inadequate because the formulation of expectations is based only on past values of a variable, whereas the RE hypothesis assumes "that individual economic agents use currently available and relevant information in the formation of their expectations and do not rely solely on past experience...".**

Autoregressive Distributed Lag Models

- An autoregressive distributed lag model (ARDL) is a model that contains both independent variables and their lagged values as well as the lagged values of the dependent variable.
- In its more general form, with p lags of Y_t and q lags of X_t, an $ARDL(p, q)$ model can be written as:

$$Y_t = \delta + \theta_1 Y_{t-1} + ... + \theta_p Y_{t-p} + \delta_1 X_{t-1} + ... + \delta_q X_{t-q} + v_t$$

Autoregressive Distributed Lag Models

- The ARDL has several advantages.
  - It captures the dynamic effects from the lagged X's and the lagged $Y's$ by including a sufficient number of lags of Y and X,
  - We can eliminate serial correlation in the errors.
  - An ARDL model can be transfomed into one with only lagged X's,.

About Model Selection

- It may happen that several models describe the time series satisfactorily, making it necessary to select the most appropriate model.
- This selection process can be simple or a bit more complex, so it is necessary to use model selection criteria.
- The most common model selection criteria are the **AIC (Akaike Information Criterion)** and the **BIC (Bayesian Information Criterion)** which is a Bayesian extension of the first one.

Information Criteria

Definition

$$AIC = log\hat{\sigma}_k^2 + \frac{n+2k}{n}$$

where $\hat{\sigma}_k^2 = \frac{SSE_k}{n}$, and k is the number of model parameters, n the sample size, and SSE_k is equal to the sum of the squared residuals under the model k ($SSE_k = \sum_{t=1}^{n}(x_t - \bar{x})^2$).

The value of k that produces the minimum AIC represents the best model. The idea is that minimizing $\hat{\sigma}_k^2$ represents a reasonable objective, except that it decreases monotonically as k increases. Therefore, we should penalize the error variance by a term proportional to the number of parameters.

Information Criteria

Definitions

$$AICc = log\hat{\sigma}_k^2 + \frac{n+k}{n-k-2}$$

$$AICc = log\hat{\sigma}_k^2 + \frac{klogn}{n}$$

- BIC is also known as the **Schwarz Information Criterion (SIC)**. Several simulation studies have verified that BIC is adequate to obtain the correct order in large samples, while AICc tends to be superior in smaller samples where the relative number of parameters is large.

An Example: Okun's Law

- We will apply the finite distributed lag model to Okun's Law.
- Okun was an economist who posited that there was a relationship between the change in unemployment from one period to the next and the rate of growth of output in the economy.
- Mathetmatically, Okun's Law can be expressed as:

$$U_t - U_{t-1} = \gamma(G_t - G_N)$$

- where U_t is the unemployment rate in period t. G_t is the growth rate of output in period t, and G_N is the "normal" growth rate, which we assume constant over time.

An Example: Okun's Law

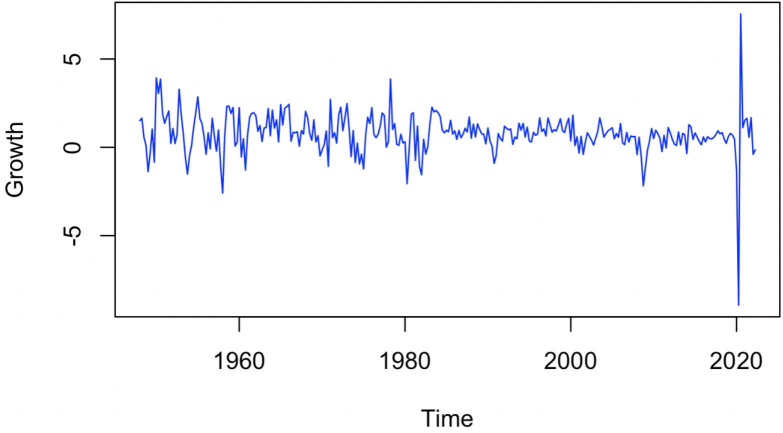
- We can rewrite the above equation in more familiar notation of the multiple regresion model by denoting the change in unemployment as: $\triangle U_t = U_{Ut} - U_{t-1}$. We then set $\gamma = \beta$ and $G_N = \alpha$. Including an error term to our equation yields:
$$\triangle U_t = \alpha + \beta_0 G_t + \mu_t$$
- Acknowledging the changes in output are likely to have a distributed-lag effect on unemployment - not all of the effect will take place instantenously. We can then further expand our equation to:
$$\triangle U_t = \alpha + \beta_0 G_t + \beta_1 G_{t-1} + \beta_2 G_{t-2} + ... + \beta_q G_{t-q} + \mu_t$$

An Example: Okun's Law

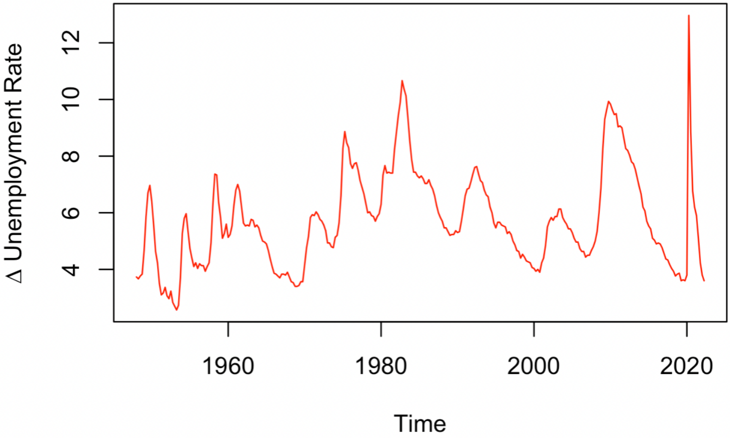
R Code

```
Okun<-read.csv(”Okun.csv”)
g <- ts(Okun$G, start=c(1948,1), frequency=4)
u <- ts(Okun$U, start=c(1948,1), frequency=4)
ts.plot(g, col='blue', ylab='Growth')
ts.plot(u, col='red', ylab='Δ Unemployment Rate')
```

An Example: Okun's Law



An Example: Okun's Law

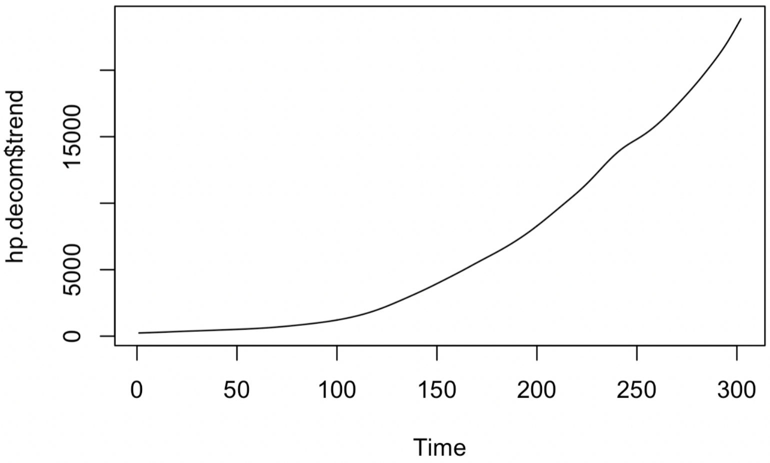


An Example: Okun’s Law

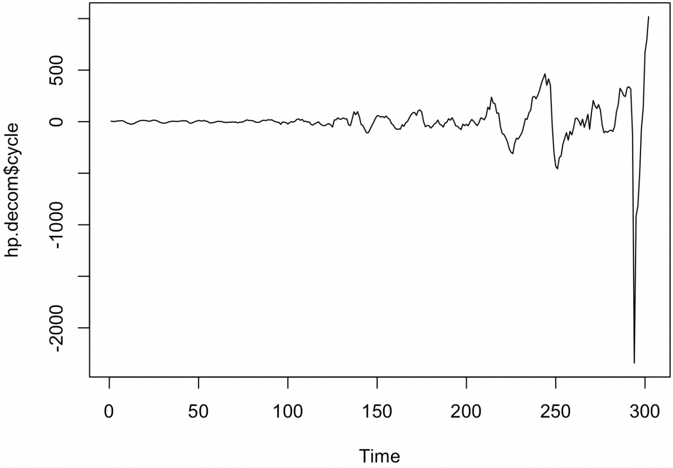
R Code

```
library(quantmod)
library(mFilter)
getSymbols(’GDP’,src=’FRED’) plot(GDP)
hp.decom <- hpfilter(GDP, freq = 1600, type = ”lambda”)
ts.plot(hp.decom$trend)
ts.plot(hp.decom$cycle)
g_Comp <- decompose(g) plot(g_Comp)
```

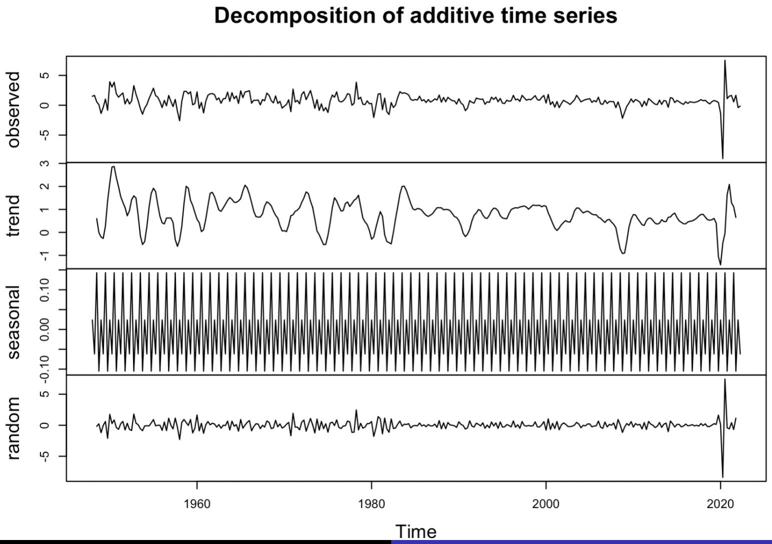
An Example: Okun's Law



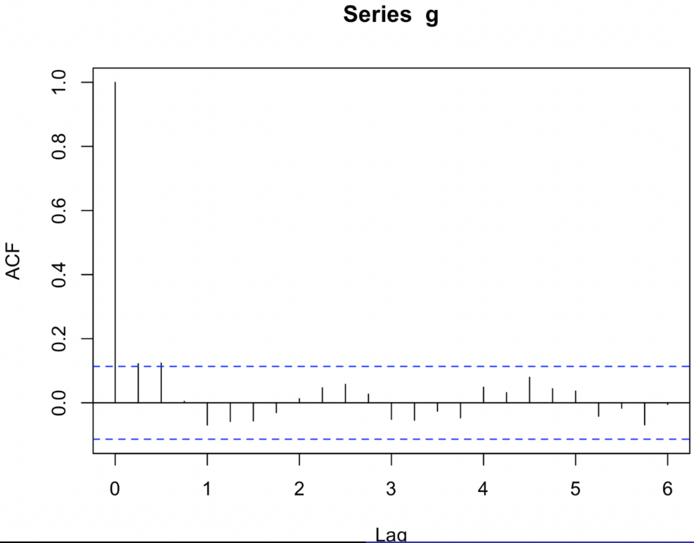
An Example: Okun's Law



An Example: Okun's Law



An Example: Okun's Law

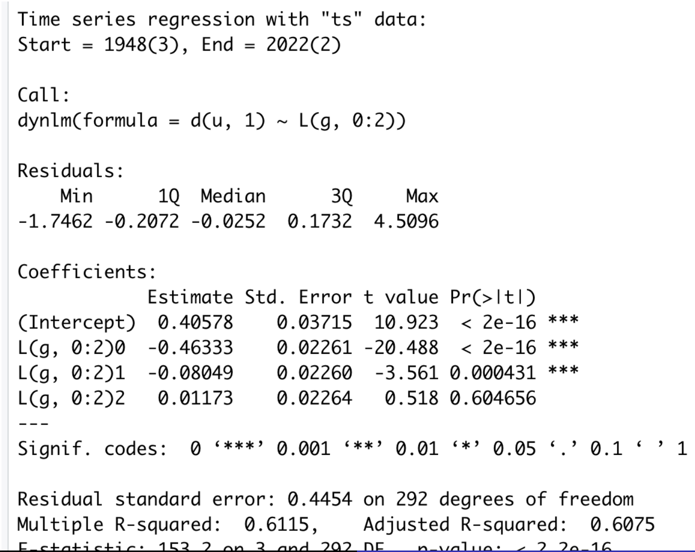


An Example: Okun’s Law

R Code

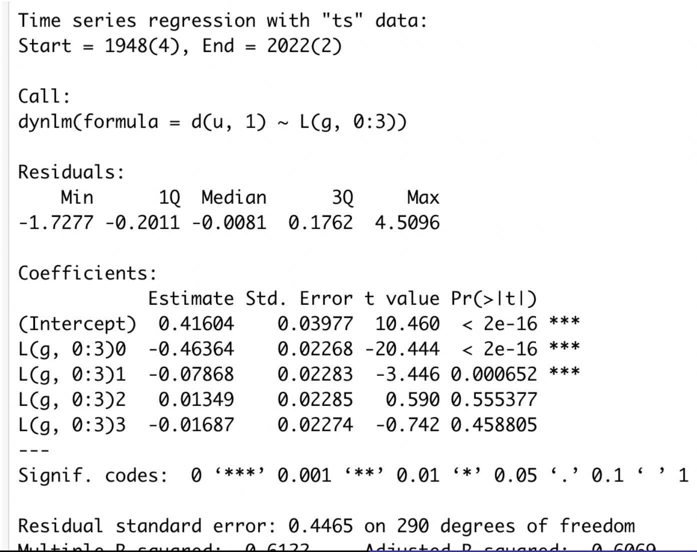
```
acf(g, type=’correlation’, plot=FALSE)$acf
acf(g, type=’correlation’)
okun.lag2 <- dynlm(d(u, 1) ˜ L(g, 0:2))
okun.lag3 <- dynlm(d(u, 1) ˜ L(g, 0:3))
summary(okun.lag2)
summary(okun.lag3)
```

An Example: Okun's Law



```
Time series regression with "ts" data:
Start = 1948(3), End = 2022(2)

Call:
dynlm(formula = d(u, 1) ~ L(g, 0:2))

Residuals:
    Min      1Q  Median      3Q     Max 
-1.7462 -0.2072 -0.0252  0.1732  4.5096 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.40578    0.03715  10.923  < 2e-16 ***
L(g, 0:2)0  -0.46333    0.02261 -20.488  < 2e-16 ***
L(g, 0:2)1  -0.08049    0.02260  -3.561 0.000431 ***
L(g, 0:2)2   0.01173    0.02264   0.518 0.604656    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4454 on 292 degrees of freedom
Multiple R-squared:  0.6115,    Adjusted R-squared:  0.6075 
F-statistic: 153.2 on 3 and 292 DF,  p-value: < 2.2e-16
```

An Example: Okun's Law



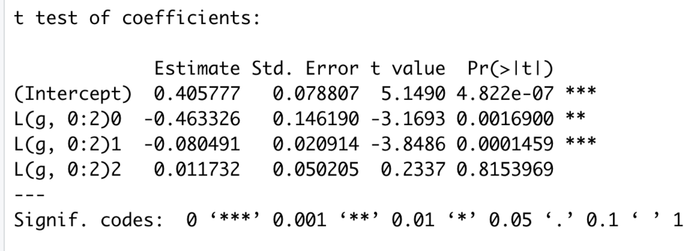
```
Time series regression with "ts" data:
Start = 1948(4), End = 2022(2)

Call:
dynlm(formula = d(u, 1) ~ L(g, 0:3))

Residuals:
    Min      1Q  Median      3Q     Max
-1.7277 -0.2011 -0.0081  0.1762  4.5096

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.41604    0.03977  10.460  < 2e-16 ***
L(g, 0:3)0  -0.46364    0.02268 -20.444  < 2e-16 ***
L(g, 0:3)1  -0.07868    0.02283  -3.446 0.000652 ***
L(g, 0:3)2   0.01349    0.02285   0.590 0.555377
L(g, 0:3)3  -0.01687    0.02274  -0.742 0.458805
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4465 on 290 degrees of freedom
Multiple R-squared:  0.6122,	Adjusted R-squared:  0.6069
```

An Example: Okun's Law

R Code

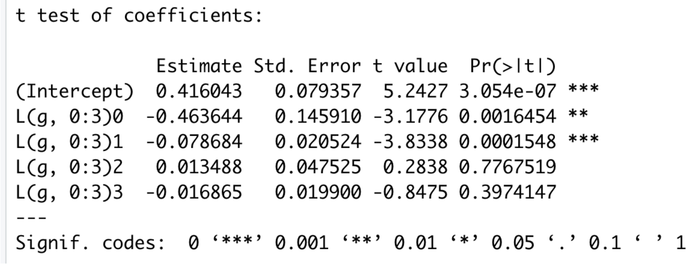
```
# HAC (heteroskedasticity and autocorrelation consistent)
standard errors
library(lmtest)
library(sandwich)
coeftest(okun.lag2, vcov=vcovHAC(okun.lag2))
coeftest(okun.lag3, vcov=vcovHAC(okun.lag3))
```

An Example: Okun's Law



```
t test of coefficients:

             Estimate Std. Error t value  Pr(>|t|)
(Intercept)  0.405777   0.078807  5.1490 4.822e-07 ***
L(g, 0:2)0  -0.463326   0.146190 -3.1693 0.0016900 **
L(g, 0:2)1  -0.080491   0.020914 -3.8486 0.0001459 ***
L(g, 0:2)2   0.011732   0.050205  0.2337 0.8153969
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

An Example: Okun's Law



```
t test of coefficients:

               Estimate Std. Error t value  Pr(>|t|)    
(Intercept)  0.416043   0.079357  5.2427 3.054e-07 ***
L(g, 0:3)0  -0.463644   0.145910 -3.1776 0.0016454 ** 
L(g, 0:3)1  -0.078684   0.020524 -3.8338 0.0001548 ***
L(g, 0:3)2   0.013488   0.047525  0.2838 0.7767519    
L(g, 0:3)3  -0.016865   0.019900 -0.8475 0.3974147    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

An Example: The Phillips Curve

- The Phillips Curve is an empirical model that describes the relationship between inflation and unemployment and was named after A.W. Phillips, the economist who discovered this relationship. Mathematically, the relationship between unemployment and inflation can be expressed as:

$$INF_t = \beta_1 - \beta_2 \triangle U_t + \mu_t$$

- where INF_t is the inflation rate during period, INF^E_{t-1} denotes inflationary expectations during period t, and $\triangle U_t = U_{Ut} - U_{t-1}$. We can rewrite the above equation using more familiar regression terms as:

$$INF_t = INF^E_{t-1} - \gamma(U_t - U_N)$$

Homework 2

- Test the econometric validity of one of these models, for at least three time periods.
- Are the series under analysis stationary?
- Do the results iof the model mprove using seasonally adjusted series?
- Compare the robustness of the Ad Hoc and Koyck models, for at least three time periods.
- How does perform the ARDL model?
- What is the role and interpretation of the results in these famous time series models?
- What are the economic conclussions that yopu can extract from your model?

References

- F.F. Alt, “Distributed Lags”, Econometrica, vol. 10, 1942, pp. 113-128.
- J. Tinbergen, “Long-Term Foreign Trade Elasticities”, Metroeconomica, vol. 1, 1949, pp. 174-185.
- L.M. Kx‘oyck, Distributed Lags and Investment Analysis, North Holland, Ámsterdam, 1954.
- Milton Friedman, A Theory of the Consumption Function, National Bureau of Economic Research, Princeton University Press, Princeton, Nueva Jersey, 1957.

Lecture III.- Univariate Time Series Models

Marcelo Villena, PhD
Santa María University

October, 2023

EX UMBRA IN SOLEM

Outline

1. ARIMA models: modeling the short-term
   - AR Models
   - MA models
   - ARMA models
2. Model selection
3. Application - Short-term Inflation
4. References

ARIMA models: modeling the short-term

- In 1970, George Box and Gwilym Jenkins two engineers with a statistical background, systematized statistical models for the analysis of univariate time series, see [1].
- In their seminal work, Box & Jenkings proposed a methodology that take into account the dependence between data.
- Thus, each observation is modeled as a function of the previous values, the time dimension therefore plays a fundamental role in the statistical analysis.

ARIMA models: modeling the short-term

- Box-Jenkins prediction models belong to the family of **linear algebraic models**, which consider a real time series as a probable realization of a certain **stochastic process**.
- These models are known by the generic name of **ARIMA (Auto-regressive Integrated Moving Average)**, which derives from its three components Autoregressive (AR), Integrated (I) Moving Averages (MA).
- Modeling a time series with this methodology involves identifying a suitable ARIMA model. that fits the series under study. Besides, it must contains the minimum necessary elements to describe the phenomenon and is useful for forecasting.

About the backshift operator

Backshift operator

We start defining the backshift operator as:

$$Bx_t = x_{t-1} \tag{1}$$

$$B^2x_t = B(Bx_t) = Bx_{t-1} = x_{t-2} \tag{2}$$

Así:

$$B^kx_t = x_{t-k} \tag{3}$$

Thus we have that the first difference can be defined in terms of lags, in other words the backshift operator:

$$\triangle x_t = x_t - x_{t-1} = (1-B)x_t \tag{4}$$

In general:

$$\triangle^d x_t = (1-B)^d x_t \tag{5}$$

Example Autoregressive Process of Order 1: AR(1)

AR(p)

An autoregressive model of order p, often shortened to $AR(p)$, has the form:

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \ldots + \phi_p x_{t-p} + \epsilon_t \tag{6}$$

where x_t is a stationary series, and $\phi_1, \phi_2, \ldots, \phi_p$ are constant. If the mean of x_t is μ, then we can replace $x_t - \mu$ in (6)

$$x_t - \mu = \phi_1(x_{t-1} - \mu) + \phi_2(x_{t-2} - \mu) + \ldots + \phi_p(x_{t-p} - \mu) + \epsilon_t \tag{7}$$

Rearranging terms

$$x_t = \alpha + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \ldots + \phi_p x_{t-p} + \epsilon_t \tag{8}$$

where $\alpha = \mu(1 - \phi_1 - \phi_2 \ldots \phi_p)$

Example Autoregressive Process of Order 1: AR(1)

AR(p)

Using the backward operators $AR(p)$ looks like:

$$(1 - \phi_1 B + \phi_2 B^2 - \ldots - \phi_p B^p) \tag{9}$$

or even more concisely

$$\phi(B)x_t = \epsilon_t \tag{10}$$

Example Autoregressive Process of Order 1: AR(1)

- In an AR(1) process the variable x_t is explained only by its past value x_{t-1}:
$$x_t = \phi x_{t-1} + \varepsilon_t \tag{11}$$
- where as we know ε_t is a white noise process with zero mean and constant variance σ^2, and ϕ is the parameter yo estimate.
- To verify that the AR(1) model is stationary we must prove the following two conditions.

Example Autoregressive Process of Order 1: AR(1)

(1) Stationary in mean

$$E(x_t) = E(\phi x_{t-1} + \varepsilon_t) = \phi E(x_{t-1}) \tag{12}$$

- In order for the process to be stationary, the mean must be constant and finite in time, $E(x_t) = E(x_{t-1})$, which implies:

$$E(x_t)\,(1-\phi) = 0$$

$$E(x_t) = \frac{0}{1-\phi} \tag{13}$$

- Therefore, for the process to be stationary the pairameter must be diffrent from 0, $\phi \neq 0$.

Example Autoregressive Process of Order 1: AR(1)

(2) Stationary in covariance To verify that the AR(1) model is stationary, the variance must be constant and finite in time:

$$\gamma = E(x_t - E(x_t))^2 = E(\phi x_{t-1} + \varepsilon_t - 0)^2 = \phi^2 var(x_{t-1}) + \sigma_2 \quad (14)$$

- Assuming that the process is stationary:

$$E(x_t)^2 = var(x_{t-1}) = var(x_t) = \gamma$$

- From here we have that:

$$\gamma = \phi^2 \gamma + \sigma_2$$

- Therefore:

$$\gamma = \frac{\sigma_2}{1 - \phi^2} \quad (15)$$

- Then for this process to be stationary, it is necessary that $|\phi| < 1$.

Example Autoregressive Process of Order 1: AR(1)

- If it is satisfied that $|\phi| < 1$, then we can represent the AR(1) model as a linear process given by:

$$x_t = \sum_{j=0}^{\infty} \phi^j \epsilon_{t-j} \tag{16}$$

- Equation (16) is called t**he causal stationary solution of the model**. The term causal refers to the fact that x_t does not depend on the future. In fact, by simple substitution,

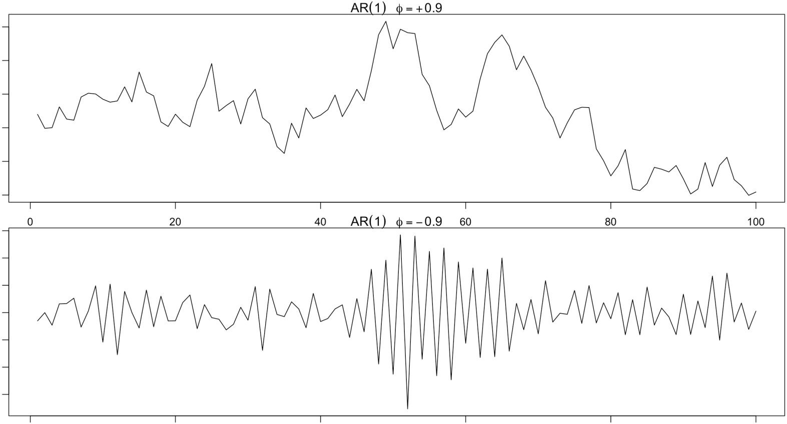
$$\underbrace{\sum_{j=0}^{\infty} \phi^j \epsilon_{t-j}}_{x_t} = \underbrace{\phi \left(\sum_{k=0}^{\infty} \phi^k \epsilon_{t-1-k} \right)}_{x_{t-1}} + \epsilon_t$$

AR(1) model simulation

R Code

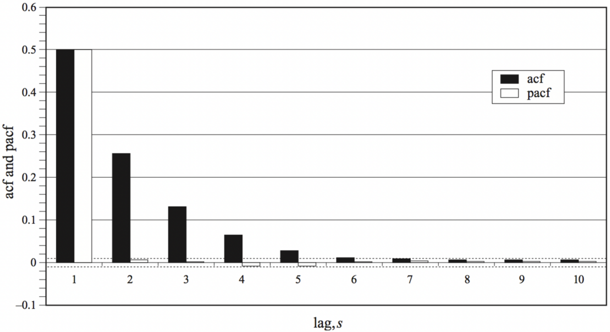
```
par(mar=c(1,1,1,1))
par(mfrow=c(2,1))
plot(arima.sim(list(order=c(1,0,0), ar=.9), n=100), ylab=“x”,
main=(expression(AR(1)   phi==+.9)))
plot(arima.sim(list(order=c(1,0,0), ar=-.9), n=100), ylab=“x”,
main=(expression(AR(1)   phi==-.9)))
```

AR(1) model simulation



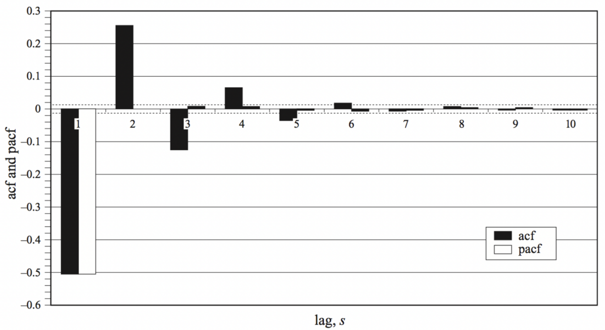
AR(1) model identification

In the case of an AR type process, the correlogram, graphical representation of the autocorrelation function, will have a damped behavior towards zero with all positive values, in case $\theta > 0$, or alternating the sign, starting with negative, if $\theta < 0$.



Sample autocorrelation and partial autocorrelation functions for a more rapidly decaying AR(1) model: $y_t = 0.5y_{t-1} + u_t$

AR(1) model identification



Sample autocorrelation and partial autocorrelation functions for a more rapidly decaying AR(1) model with negative coefficient: $y_t = -0.5y_{t-1} + u_t$

Moving Average - MA (q)

- As an alternative to the autoregressive representation in which the x_t on the left hand side of the equation is assumed to be linearly combined, the q-order moving average model, abbreviated as $MA(q)$, assumes that the white noise ϵ_t usually on the right hand side of the equation, are linearly combined to model the observed data.

ARIMA models: modeling the short-term

Definition: Moving Average - MA (q)

$$x_t = \epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2} + \ldots + \theta_q\epsilon_{t-q} \tag{17}$$

where there are q lags of the moving average ϵ_t and $\theta_1 + \theta_2 + \ldots + \theta_q$ are parameters.

Although it is not necessary, we assume that ϵ_t is a white noise series.

ARIMA models: modeling the short-term

Definition: Moving Average - MA (q)

We can also write the process $MA(q)$ in the equivalent form:

$$x_t = \theta_t(B)\epsilon_t \tag{18}$$

where θ_t is the moving average operator defined as:

$$\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \ldots + \theta_q B^q \tag{19}$$

Unlike the autoregressive process, the moving average process is stationary for any value of the parameters $\theta_1 + \theta_2 + \ldots + \theta_q$.

Interpretation of the moving average model - MA(q)

- Just as an autoregressive model is intuitively simple to understand, the formulation of a moving average model is often not intuitive. What does it mean that a random variable is explained in terms of errors made in previous periods, where do these errors come from, what is the justification for such a model? In fact, a moving average model can be obtained from an autoregressive model by making successive substitutions.

Interpretation of the moving average model - MA(q)

Assume an $AR(1)$ model, with no independent term:

$$x_t = \phi x_{t-1} + \epsilon_t \tag{20}$$

if we consider $t-1$ instead of$t$the model would be in this case:

$$x_{t-1} = \phi x_{t-2} + \epsilon_{t-1} \tag{21}$$

replacing:

$$x_t = \phi^2 x_{t-2} + \phi \epsilon_{t-1} + \epsilon_t \tag{22}$$

Interpretation of the moving average model - MA(q)

If we now substitute x_{t-2} by its autoregressive expression and so on we arrive at a model of the type:

$$x_t = \epsilon_t + \theta\epsilon_{t-1} + \theta^2\epsilon_{t-2} + \ldots + \theta^q\epsilon_{t-q} \tag{23}$$

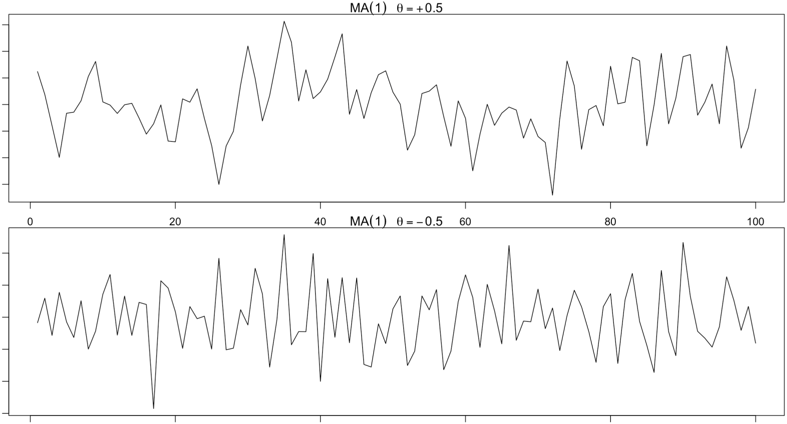
which is the expression, without an independent term, of a moving average model as the one discussed above. In fact, strictly speaking, the passage from one model to the other should be done in reverse, from a MA to an AR, using the general Wold decomposition theorem.

MA(1) model simulation

R Code

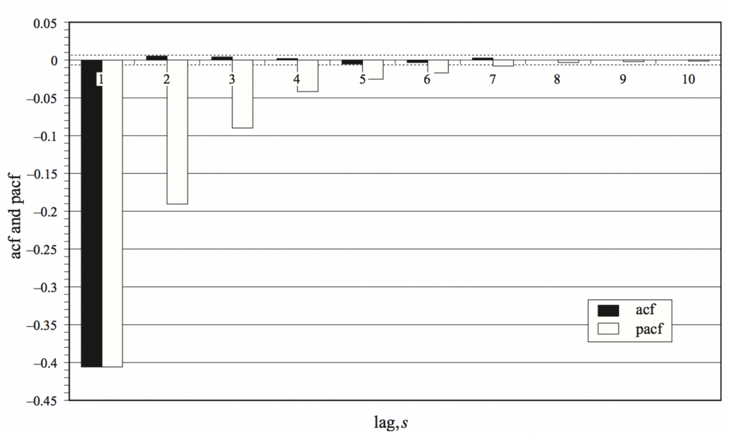
```
par(mfrow = c(2,1))
plot(arima.sim(list(order=c(0,0,1), ma=.5), n=100), ylab=“x”,
main=(expression(MA(1)   theta==+.5)))
plot(arima.sim(list(order=c(0,0,1), ma=-.5), n=100), ylab=“x”,
main=(expression(MA(1)   theta==-.5)))
```

MA(1) model simulation



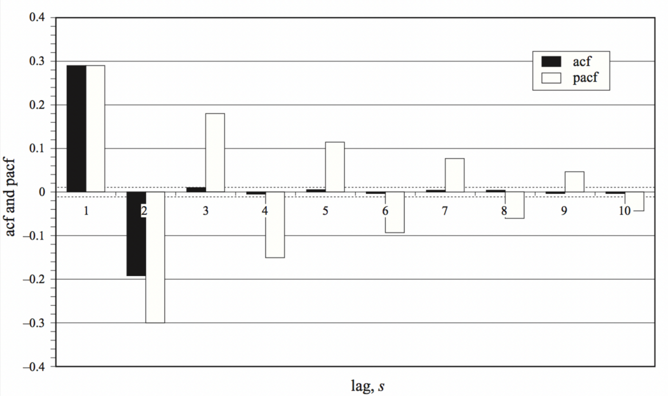
Identification of MA model

For the identification of all the components of the MA model, as we saw for the AR model, we use the autocorrelation function (AFC) and the partial autocorrelation function (PAFC), and thus proceed to the identification of the components, based on the graphs of the different models.



Sample autocorrelation and partial autocorrelation functions for an MA(1) model: $y_t = -0.5u_{t-1} + u_t$

Identification of MA model



Sample autocorrelation and partial autocorrelation functions for an MA(2) model: $y_t = 0.5u_{t-1} - 0.25u_{t-2} + u_t$

ARMA models

Definition: Autoregressive moving average - ARMA (p, q)

A time series $\{x_t, t = 0, \pm 1, \pm 2, \dots\}$ is an ARMA(p, q) process, if it is stationary and

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_p x_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \dots + \theta_q \epsilon_{t-q} \tag{24}$$

The parameters p and q are called autoregressive orders and moving averages, respectively.

If x_t has a non-zero mean μ, we establish that $\alpha = \mu(1 - \theta_1, \dots - \theta_q)$ and we can rewrite the model as:

$$x_t = \alpha + \phi_1 x_{t-1} + \dots + \phi_p x_{t-p} + w_t + \theta_1 \epsilon_{t-1} + \dots + \theta_q \epsilon_{t-q}. \tag{25}$$

Invertibility

- **A time series is invertible if the errors can be inverted in a representation of past observations.** Thus, for example, as we have already seen, the AR model is always invertible. In the case of the ARMA model, the roots of the following equations must be analyzed to ensure invertibility.

$$\phi(z) = 1 + \phi_1 z + \phi_2 z^2 + \ldots + \phi_p z^p \tag{26}$$

$$\theta(z) = 1 + \theta_1 z + \theta_2 z^2 + \ldots + \theta_q z^q \tag{27}$$

Invertibility

- In particular the ARMA model will be invertible if and only if $\theta(z) \neq 0$ for$|z| \leq 1$ In general, the eigenvalues are the solution of $det(A - \lambda I) = 0$, we see that this is the characteristic polynomial of the equations we defined above.
- Therefore, we see that the eigenvalues of A are the inverse of the roots of the characteristic polynomial, and that convergence of the backward iteration occurs when the roots of the characteristic polynomial lie inside the unit circle.

Stationarity and Invertibility

- Wold showed that all stationary stochastic covariance processes could be decomposed as the sum of deterministic and linearly indeterministic processes which were uncorrelated with all lags; that is, if y_t is the stationary covariance, then:

$$y_t = x_t + z_t \tag{28}$$

- where x_t is a stationary deterministic process in covariance and z_t is linearly indeterministic, with $Cov(x_t, z_s) = 0$ for all t and s. This result provides a theoretical basis for Box and Jenkins' proposal to model scalar covariance stationary (unseasonalized) processes such as ARMA processes.

ARMA models (p,q)

- As indicated above, when $q = 0$, the model is called the autoregressive model of order p, $AR(p)$, and when $p = 0$, the model is called the moving average model of order q, $MA(q)$.
- It is useful to write ARIMA models using the AR operator and the MA operator described above. In particular, the $ARMA(p, q)$ model can then be written concisely as:

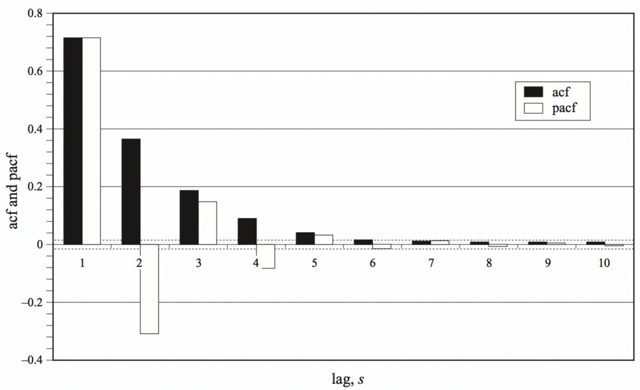
$$\phi(B)x_t = \theta(B)\epsilon_t. \tag{29}$$

- **ARIMA models (p, i, q)** The ARMA model gains its I and becomes ARIMA when it must be integrated to achieve stationarity. The index I will then be the number of times it must be differenced.

ARMA model identification

- The autocorrelation function (AFC) and the partial autocorrelation function (PAFC) are used to identify all the components of the ARMA model, and the seasonal and non-seasonal components are identified separately, based on the graphs of the different models.

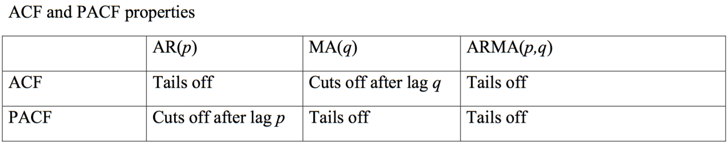
ARMA model identification



Sample autocorrelation and partial autocorrelation functions for an ARMA(1, 1) model: $y_t = 0.5y_{t-1} + 0.5u_{t-1} + u_t$

ARMA model identification

Summing up



ACF and PACF properties

	AR(p)	MA(q)	ARMA(p,q)
ACF	Tails off	Cuts off after lag q	Tails off
PACF	Cuts off after lag p	Tails off	Tails off

SARIMA model

- ARIMA models are also capable of modeling a wide range of seasonal data. The so-called SARIMA models, Seasonal ARIMA models, are obtained by including additional seasonal terms in the ARIMA models we have seen so far, as follows:

$$ARIMA(p,d,q)(P,D,Q)m \tag{30}$$

  where $m =$ number of periods per season.

- We use uppercase notation for the seasonal parts of the model and lowercase notation for the non-seasonal parts of the model. The seasonal part of the model consists of terms that are very similar to the non-seasonal components of the model, but involve seasonal period regressors.

Statistical evaluation of an ARIMA model

- **Statistical significance of the parameters**: The coefficients obtained in the estimation that are not significantly different from zero, at a significance level of 5%, are not necessary and should be eliminated.
- **Stationarity and invertibility of the estimated model**: For values of the estimated coefficients close to the non-stationarity frontier, it is convenient to carry out a unit root test.
- **Stability of the estimated model**: Even if the parameters are significant, the model can be rejected if there is a strong correlation between the model parameters. This occurs when the correlation coefficient has an absolute value greater than 0.7, then it is convenient to try alternative models.

About Model Selection

- It may happen that several models describe the time series satisfactorily, making it necessary to select the most appropriate model.
- This selection process can be simple or a bit more complex, so it is necessary to use model selection criteria.
- The most common model selection criteria are the **AIC (Akaike Information Criterion)** and the **BIC (Bayesian Information Criterion)** which is a Bayesian extension of the first one.

Information Criteria

Definition

$$AIC = log\hat{\sigma_k^2} + \frac{n+2k}{n}$$

where $\hat{\sigma_k^2} = \frac{SSE_k}{n}$, and k is the number of model parameters, n the sample size, and SSE_k is equal to the sum of the squared residuals under the model k ($SSE_k = \sum_{t=1}^{n}(x_t - \bar{x})^2$).

- The value of k that produces the minimum AIC represents the best model. The idea is that minimizing $\hat{\sigma_k^2}$ represents a reasonable objective, except that it decreases monotonically as k increases. Therefore, we should penalize the error variance by a term proportional to the number of parameters.

Information Criteria

Definitions

$$AICc = log\hat{\sigma}_k^2 + \frac{n+k}{n-k-2}$$

$$AICc = log\hat{\sigma}_k^2 + \frac{klogn}{n}$$

- BIC is also known as the **Schwarz Information Criterion (SIC)**. Several simulation studies have verified that BIC is adequate to obtain the correct order in large samples, while AICc tends to be superior in smaller samples where the relative number of parameters is large.

About Model Selection

- Ultimately, one model is better than another if its prediction is better. On the other hand, we will say that **a prediction is better than another when it makes a smaller extra-sampling error.**
- Thus, the accuracy of the methods used to forecast can be measured for example through the loss function: **Mean Square Error (MSE)**, in order to understand which model provides a better out-of-sample forecast over another. That is:

$$\textbf{MSE} = \frac{1}{T}\sum_{t=1}^{T}(x_t - \hat{x}_t)^2 \tag{31}$$

- where x_t corresponds to the actual value of the series at time t and $\hat{x}$ corresponds to the value predicted by the proposed model at the same instant.

About Model Selection

- Other model selection criteria that consider the extra-sampling error are: i) the Mean Absolute Error (MAD), and ii) Mean Absolute Percentage Error (MAPE).

$$\textbf{MAD} = \frac{1}{T}\sum_{t=1}^{T} |x_t - \hat{x}_t| \tag{32}$$

$$\textbf{MAPE} = \frac{1}{T}\sum_{t=1}^{T} \left|1 - \frac{x_t}{\hat{x}_t}\right| \tag{33}$$

Example CPI

Considering monthly CPI data from January 2013 to date in Chile, obtained from the Central Bank’s website, we will try to predict the CPI (original series).

R Code

```
rm(list=ls())
data< −read.csv (“ipc.csv”)
ipc < − ts(data[,2],start = c(2013,1), end=c(2018, 6), frequency = 12)
plot.ts(ipc, xlab=’Years’, ylab = “Indice de Precios al Comsumidor’)
```

Example CPI

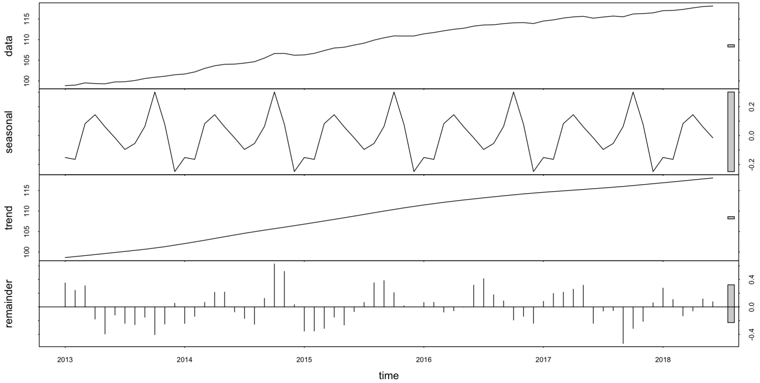
Finding the order of the model. Trend, stationarity, autocorrelation.

R Code

```
# Descomposición
fit < − stl(ipc, s.window=“period”)
plot(fit)
# Test de raíz unitaria
adf.test(ipc)
adf.test(diff(ipc))
# Función de autocorrelación (AFC) y autocorrelación parcial (PAFC)
acf(diff(ipc),lag=36,lwd=3)
pacf(diff(ipc),lag=36,lwd=3)
```

Example CPI

- **Series decomposition**

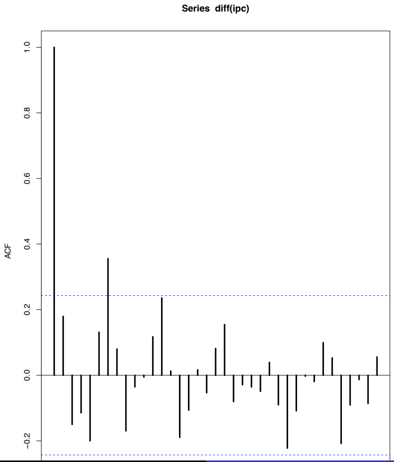


Example CPI - Unit root test

```
Augmented Dickey-Fuller Test
data: ipc
Dickey-Fuller = -0.11148, Lag order = 4, p-value =0.99
alternative hypothesis: stationary

Augmented Dickey-Fuller Test
data: diff(ipc)
Dickey-Fuller = -5.8024, Lag order = 3, p-value = 0.01
alternative hypothesis: stationary
```

Example CPI

Autocorrelation Function (AFC) and Partial Autocorrelation Function (PAFC)



Example CPI - Forecast

R Code

```
train.series =ipc [1 : 44]
test.series = ipc [45 : 62]
arima.model=arima(train.series, order=c(0,1,1))
forecast=predict(arima.model, length(test.series)
mse < −sum((forecast$pred-test.series)∧2)/length(test.series)
mad < − sum(abs(forecast$pred-test.series))/length(test.series)
mape < − sum(abs( 1 -
forecast$pred/test.series))/length(test.series)
fit < − auto.arima(ipc)
summary(fit)
plot(fit)
mape < − sum(abs(1 - test.series/f[[“mean”]]))/length(test.series)
accuracy(fit)
```

Example CPI - output ARIMA (0, 1, 1)

Call:
arima(x = train.series, order = c(0, 1, 1))
Coefficients:
ma1
0.8205
s.e. 0.0906
σ^2 estimated as 0.1029 : $loglikelihood = -12.68, aic = 29.37$

forecast ARIMA (0, 1, 1)
mse [1] 69.80031

Example CPI - Forecast - output ARIMA (0, 1, 1)

```
$pred
Time Series:
Start = 45
End = 54
Frequency = 1
[1] 113.6141 113.6253 113.6292 113.6307 113.6311 113.6313
[7] 113.6314 113.6314 113.6314 113.6314
$se
Time Series:
Start = 45
End = 54
Frequency = 1
[1] 0.3128668 0.6841962 0.9882296 1.2406559 1.4565974
[6] 1.6465783 1.8174943 1.9738906 2.1188485 2.2545301
```

Example CPI - output auto.arima

Series: ipc
ARIMA(0,1,1)(0,0,1)[12] with drift

Coefficients:

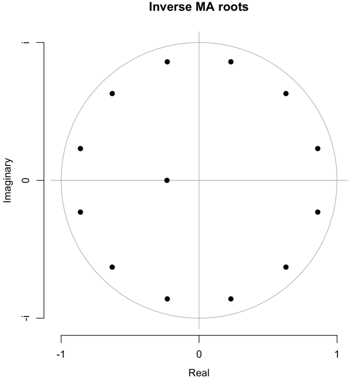
```
            ma1       sma1      drift
            0.2329    0.2483    0.2909
s.e.        0.1443    0.1396    0.0500
```

σ^2 estimated as 0.07771 : *loglikelihood* $= -8.01$
$AIC = 24.02$ $ICc = 24.69$ $BIC = 32.72$

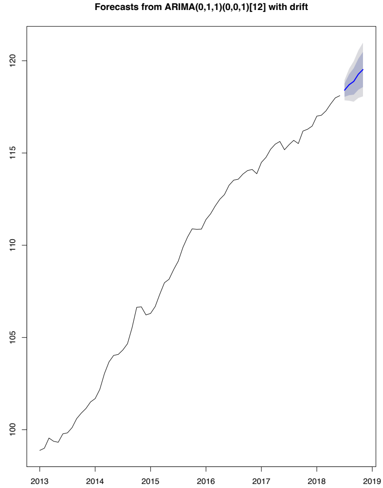
Training set error measures:

```
                  ME        RMSE        MAE         MPE
Training set 0.00467571 0.2701877 0.2012356 0.005434612
                  MAPE       MASE        ACF1
Training set 0.185618 0.05414794 -0.03368001
```

Example: CPI - nverse MA roots - auto.arima



Example: CPI - Forecast auto.arima



Box-Jenkins modelling procedure

(1) Data preparation involves transformations and differencing. Transformations of the data (such as square roots or logarithms) can help stabilize the variance in a series where the variation changes with the level. This often happens with business and economic data. Then the data are differenced until there are no obvious patterns such as trend or seasonality left in the data. "Differencing" means taking the difference between consecutive observations, or between observations a year apart. The differenced data are often easier to model than the original data.

Box-Jenkins modelling procedure

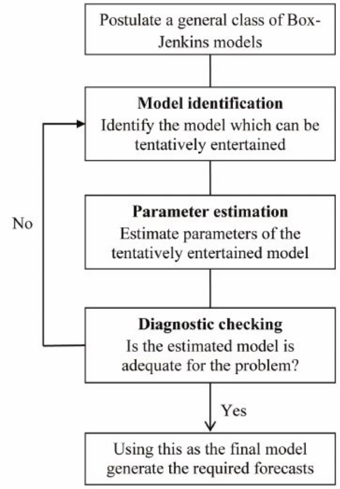
(2) Model selection in the Box-Jenkins framework uses various graphs based on the transformed and differenced data to try to identify potential ARIMA processes which might provide a good fit to the data. Later developments have led to other model selection tools such as Akaike's Information Criterion.
(3) Parameter estimation means finding the values of the model coefficients which provide the best fit to the data. There are sophisticated computational algorithms designed to do this.

Box-Jenkins modelling procedure

(4) Model checking involves testing the assumptions of the model to identify any areas where the model is inadequate. If the model is found to be inadequate, it is necessary to go back to Step 2 and try to identify a better model.

(5) Forecasting is what the whole procedure is designed to accomplish. Once the model has been selected, estimated and checked, it is usually a straight forward task to compute forecasts.

Box-Jenkins modelling procedure



Postulate a general class of Box-Jenkins models

Model identification
Identify the model which can be tentatively entertained

Parameter estimation
Estimate parameters of the tentatively entertained model

Diagnostic checking
Is the estimated model is adequate for the problem?

Using this as the final model generate the required forecasts

Homework 3

- Calibrate and evaluate (out-of-sample) the following models for the cpi (consumer price index) of a country of your choice:

$$ln(cpi_t) = \alpha_t + \delta_t t + \psi_t ln(cpi_{t-1}) + \varepsilon_t$$

1 Average of the last 5 years, average of the last 10 years.

2 $\psi = 1$, $\delta = 0$, and $\alpha = 0$ or $\alpha \neq 0$, a random walk with drift and without drift.

3 α constant, $\delta = 0$ and ψ follows an AR(1).

4 α and δ constant, ψ follows an AR(1).

5 AR(1), AR(2), AR(3).

Homework 3

6 MA(1), MA(2), MA(3).

7 ARIMA(1,1,0), ARIMA(0,1,1), ARIMA(1,1,1)..

8 α ,δ and ψ contants follow random paths with independent innovations.

9 $\delta = 0$, α and ψ contants follow random paths with independent innovations.

10 α constant, $\delta = 0$ and ψ contantsfollows a random walk.

References

- [1] BOX, G.E.P. and G.M. JENKINS (1970) Time series analysis: Forecasting and control, San Francisco: Holden-Day.
- [2] MAKRIDAKIS, S., S.C. WHEELWRIGHT, and R.J. HYNDMAN (1998) Forecasting: methods and applications, New York: John Wiley & Sons.
- [3] PANKRATZ, A. (1983) Forecasting with univariate Box–Jenkins models: concepts and cases, New York: John Wiley & Sons.

Lecture IV.- Vector Autoregressive Models

Marcelo Villena, PhD
Santa María University

October, 2023

Outline

1. Introduction
   - Notation and Concepts
   - Vector autoregressive models compared with structural equations models
2. Choosing the optimal lag length for a VAR
   - Practical Example
3. Stability of VAR processes
   - Primitive versus Standard Form of VARs
   - Conditions for stability
4. On Causality
   - Impulse response functions (IRF)
5. Homework 4
6. References

Vector autoregressive models - VAR

- So far we have considered models that impose a unidirectional relationship: the right-hand side variable is influenced by the left-hand side variables, but not vice versa.
- There are many cases in which the opposite should also be allowed, i.e., all variables should affect each other.
- Sometimes causality is assumed, although as we will see later, high correlation is not synonymous of causality.
- In this context, and since Sims' critique in the early 1980s, multivariate data analysis, in the context of **vector autoregressive models (VARs)**, has become a standard tool in econometrics.

Vector autoregressive models - VAR

- **Vector autoregressive models (VARs)** are a system of two or more time series that is modeled considering lags of the variables and the dynamic interaction that may exist between them.

Vector autoregressive models - VAR

- It consists mainly of two dimensions, the number of variables (g) and the number of lags (k). The simplest case is a bivariate VAR:

$$y_{1t} = \beta_{10} + \beta_{11} y_{1t-1} + \ldots + \beta_{1k} y_{1t-k} + \alpha_{11} y_{2t-1} + \ldots + \alpha_{1k} y_{2t-k} + \mu_1 t$$

$$y_{2t} = \beta_{20} + \beta_{21} y_{2t-1} + \ldots + \beta_{2k} y_{2t-k} + \alpha_{21} y_{1t-1} + \ldots + \alpha_{2k} y_{1t-k} + \mu_2 t$$

where u_{it} is an error term with $E(u_{it}) = 0$, $i = 1, 2$; $E(u_{1t} u_{2t}) = 0$.

- The assumption of independence from errors can be relaxed, as we will see later. The analysis could be extended for example to a VAR model (g), where we have g variables and g equations.

Notation and Concepts

- An important feature of VAR models is the simplicity of their notation. For example, consider the case above, where $k = 1$. We can write this model as:

$$y_{1t} = \beta_{10} + \beta_{11} y_{1t-1} + \alpha_{11} y_{2t-1} + \mu_1 t$$
$$y_{2t} = \beta_{20} + \beta_{21} y_{2t-1} + \alpha_{21} y_{1t-1} + \mu_2 t$$

or

$$\begin{pmatrix} y_{1t} \\ y_{2t} \end{pmatrix} = \begin{pmatrix} \beta_{10} \\ \beta_{20} \end{pmatrix} + \begin{pmatrix} \beta_{11} & \alpha_{11} \\ \alpha_{21} & \beta 21 \end{pmatrix} \begin{pmatrix} y_{1t-1} \\ y_{2t-1} \end{pmatrix} + \begin{pmatrix} \mu_{1t} \\ \mu_{2t} \end{pmatrix} \qquad (1)$$

or even more compactly as

$$\underset{gx1}{y_t} = \underset{gx1}{\beta_0} + \underset{gxg}{\beta_1}\ \underset{gx1}{y_{t-1}} + \underset{gx1}{\mu_{1t}} \qquad (2)$$

Vector autoregressive models - VAR

- This model can be extended to the case where there are k delays of each variable in each equation

$$\underset{gx1}{y_t} = \underset{gx1}{\beta_0} + \underset{gxg}{\beta_1}\,\underset{gx1}{y_{t-1}} + \underset{gxg}{\beta_2}\,\underset{gx1}{y_{t-2}} + \cdots \underset{gxg}{\beta_k}\,\underset{gx1}{y_{t-k}} + \underset{gx1}{\mu_{1t}} \tag{3}$$

- We can also extend this to the case where the model includes first difference terms and cointegration relations (VECM, vector error corrections models, that we will see later in the course).

Vector autoregressive models - VAR

- VAR models explain endogenous variables only by their own history, in addition to deterministic regressors.
- In contrast, structural VAR models (hereinafter SVAR, for Structural VAR) allow explicit modeling of contemporary interdependence between left-sided variables.
- This type of model tries to circumvent the shortcomings of VAR models.

Advantages of the VAR Model

- **It is not necessary to specify which variables are endogenous or exogenous - they are all endogenous.**
- It allows the value of a variable to depend on more than its own lags or combinations of white noise terms, so they are more general than our well-known ARIMA model.
- As long as we don't have contemporary terms on the right side of the equations, we can use OLS separately in each equation.

Disadvantages of the VAR Model

- VAR models are atheoretical (as are ARIMA models).
- How do you decide the length of the appropriate lag?
- There are so many parameters!
    - If we have g equations for the g variables, and we have k lags of each of the variables in each equation, we have to estimate $(g + kg^2)$parameters. For example, if g $=$ 3, and k $=$ 3, we will have 30 parameters!!!
- We have to ensure that all components of the VAR model are stationary?

Why VAR?

- VARs are useful in several contexts:

1. forecasting a collection of related variables where no explicit interpretation is required;
2. testing whether one variable is useful in forecasting another (the basis of Granger causality tests);
3. impulse response analysis, where the response of one variable to a sudden but temporary change in another variable is analysed;
4. forecast error variance decomposition, where the proportion of the forecast variance of each variable is attributed to the effects of the other variables.

Choosing the optimal lag length for a VAR

- There are two possible approaches: i) cross-equation constraints, and ii) information criteria

Cross-equation constraints

- In the spirit of (unconstrained) VAR modeling, each equation must have the same lag length.
- Suppose a bivariate VAR(8) was estimated using quarterly data with 8 lags for the two variables in each equation, and we want to examine the constraint that the coefficients of lags 5 to 8 are jointly zero.
- **This can be done using a likelihood ratio test.**

Choosing the optimal lag length for a VAR

- We denote the variance-covariance matrix of the residuals (given by $\hat{\mu}\hat{\mu}'/T$), as $\hat{\sum}$.
- The likelihood ratio test of this joint hypothesis is given by:

$$LR = T\left[log|\hat{\sum_{r}}| - log|\hat{\sum_{u}}|\right]$$

- where$\hat{\sum}_r$ is the variance-covariance matrix of the residuals for the restricted model (with 4 lags), $\hat{\sum}_u$ is the variance-covariance matrix of the residuals for the unrestricted VAR model (with 8 lags), and T is the sample size.
- The test statistic is distributed asymptotically as a χ^2 with degrees of freedom equal to the total number of constraints.

Choosing the optimal lag length for a VAR

- In the case of the previous VAR, we are restricting 4 lags of two variables in each of the two equations, a total of $4 * 2 * 2 = 16$ constraints.
- In the general case where we have a VAR with p equations, and we want to impose the restriction that the last lags have zero coefficients, there would be a total of p2q restrictions.
- **Disadvantages:** The performance of the LR test is complicated and requires an assumption of normality of disturbances.

Choosing the optimal lag length for a VAR

Information Criteria for VAR Lag Selection

Multivariate versions of the information criteria are required. These can be defined as:

- **Akaike information criterion**

$$MAIC = ln|\sum| + 2k'/T$$

- **Bayesian or Schwarz criterion**

$$MSBIC = ln|\sum| + k'/Tln(T)$$

$$MHQIC = ln|\sum| + 2k/Tln(ln(T))$$

where all notation holds and k is the total number of regressors in all equations, which will be equal to $g2k + g$ for the g equations, each with k delays for the variable g, plus a constant term in each equation. The values of the information criteria are constructed for $0, 1, ...$ delays (up to some pre-specified maximum of k).

Example of a VAR model in R

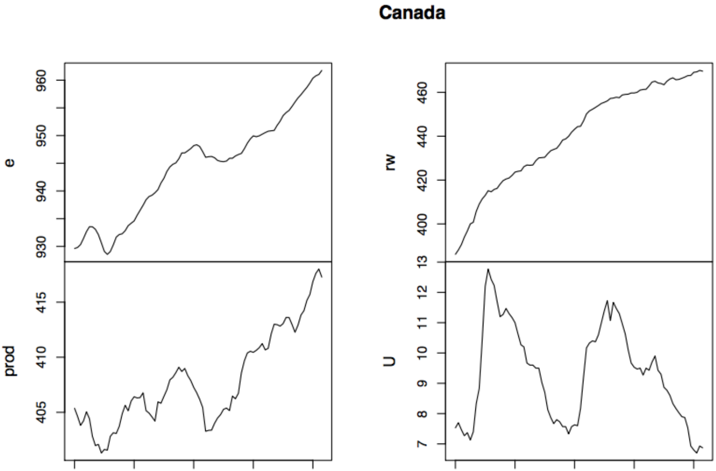
- Lütkepohl & Krätzig (2004) used the following series:
    - labor productivity (**prod**) defined as the logarithmic difference between GDP and employment,
    - the logarithm of employment (**e**),
    - the unemployment rate (**U**) and
    - real wages (**rw**), defined as the logarithm of the real wage index.
    - Data was obtained from the OECD database, and cover the first quarter of 1980 to the fourth quarter of 2004.

Example of a VAR model in R

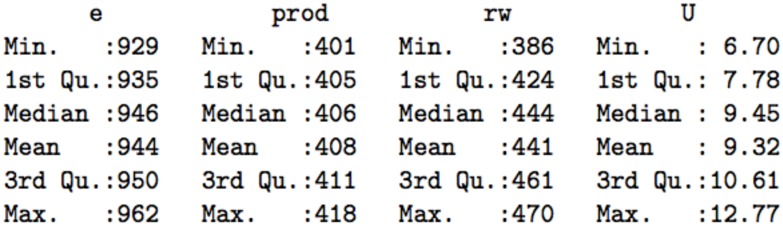
R Code

```
rm(list=ls())
install.packages(“vars”)
library(“vars”)
data(“Canada”)
summary(Canada)
plot(Canada, nc = 2, xlab = “”)
adf2 < − summary(ur.df(Canada[, “prod”], type = “drift”, lags = 1))
```

Example of a VAR model in R

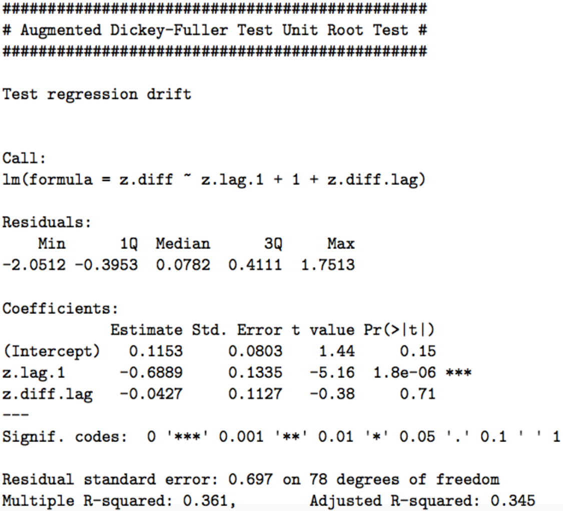


Example of a VAR model in R



```
      e               prod              rw                U
Min.   :929    Min.   :401    Min.   :386    Min.   : 6.70
1st Qu.:935    1st Qu.:405    1st Qu.:424    1st Qu.: 7.78
Median :946    Median :406    Median :444    Median : 9.45
Mean   :944    Mean   :408    Mean   :441    Mean   : 9.32
3rd Qu.:950    3rd Qu.:411    3rd Qu.:461    3rd Qu.:10.61
Max.   :962    Max.   :418    Max.   :470    Max.   :12.77
```

Example of a VAR model in R



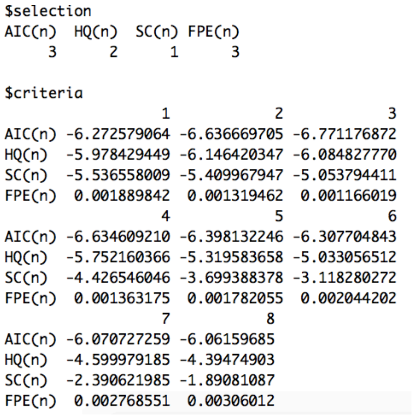
```
###############################################
# Augmented Dickey-Fuller Test Unit Root Test #
###############################################

Test regression drift


Call:
lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)

Residuals:
    Min      1Q  Median      3Q     Max
-2.0512 -0.3953  0.0782  0.4111  1.7513

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.1153     0.0803    1.44     0.15
z.lag.1      -0.6889     0.1335   -5.16  1.8e-06 ***
z.diff.lag   -0.0427     0.1127   -0.38     0.71
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.697 on 78 degrees of freedom
Multiple R-squared: 0.361,      Adjusted R-squared: 0.345
```

Example of a VAR model in R

R Code

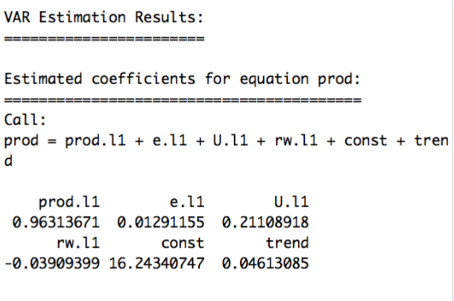
```
VARselect(Canada, lag.max = 8, type = “both”)
Canada < − Canada[, c(“prod”, “e”, “U”, “rw”)]
p1ct < − VAR(Canada, p = 1, type = “both”)
p1ct
summary(p1ct, equation = “e”)
plot(p1ct, names = “e”)
ser11 < − serial.test(p1ct, lags.pt = 16, type = “PT.asymptotic”)
ser11$serial
norm1 < − normality.test(p1ct)
norm1$jb.mul p
rd < − predict(plct, n.ahead = 10, ci = 0.95, dumvar = NULL)
print(prd)
plot(prd, “single”)
```

Example of a VAR model in R



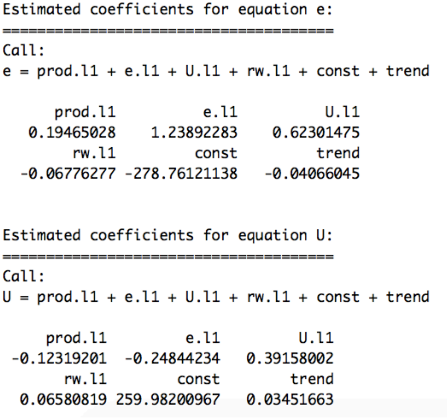
```
$selection
AIC(n)  HQ(n)  SC(n) FPE(n)
     3      2      1      3

$criteria
                  1            2            3
AIC(n) -6.272579064 -6.636669705 -6.771176872
HQ(n)  -5.978429449 -6.146420347 -6.084827770
SC(n)  -5.536558009 -5.409967947 -5.053794411
FPE(n)  0.001889842  0.001319462  0.001166019
                  4            5            6
AIC(n) -6.634609210 -6.398132246 -6.307704843
HQ(n)  -5.752160366 -5.319583658 -5.033056512
SC(n)  -4.426546046 -3.699388378 -3.118280272
FPE(n)  0.001363175  0.001782055  0.002044202
                  7           8
AIC(n) -6.070727259 -6.06159685
HQ(n)  -4.599979185 -4.39474903
SC(n)  -2.390621985 -1.89081087
FPE(n)  0.002768551  0.00306012
```

Example of a VAR model in R



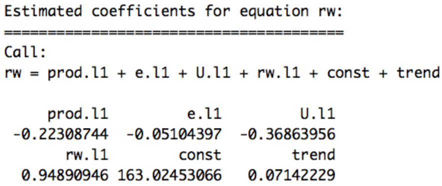
```
VAR Estimation Results:
=======================

Estimated coefficients for equation prod:
=========================================
Call:
prod = prod.l1 + e.l1 + U.l1 + rw.l1 + const + tren
d

    prod.l1        e.l1        U.l1 
 0.96313671  0.01291155  0.21108918 
      rw.l1       const       trend 
-0.03909399 16.24340747  0.04613085 
```

Example of a VAR model in R



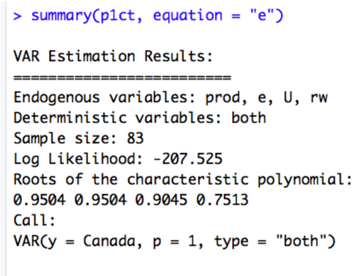
```
Estimated coefficients for equation e:
======================================
Call:
e = prod.l1 + e.l1 + U.l1 + rw.l1 + const + trend

      prod.l1          e.l1          U.l1
   0.19465028    1.23892283    0.62301475
        rw.l1         const         trend
  -0.06776277 -278.76121138   -0.04066045


Estimated coefficients for equation U:
======================================
Call:
U = prod.l1 + e.l1 + U.l1 + rw.l1 + const + trend

     prod.l1         e.l1         U.l1
 -0.12319201  -0.24844234   0.39158002
       rw.l1        const        trend
  0.06580819 259.98200967   0.03451663
```

Example of a VAR model in R



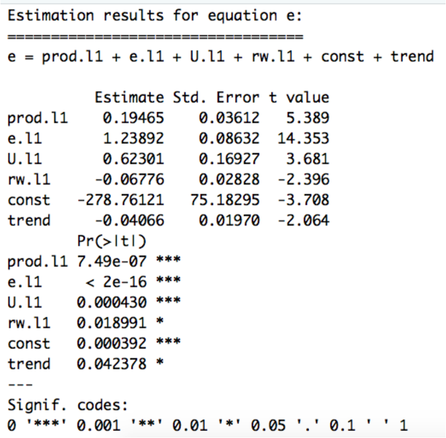
```
Estimated coefficients for equation rw:
=======================================
Call:
rw = prod.l1 + e.l1 + U.l1 + rw.l1 + const + trend

    prod.l1         e.l1         U.l1
 -0.22308744  -0.05104397  -0.36863956
       rw.l1        const        trend
  0.94890946 163.02453066   0.07142229
```

Example of a VAR model in R



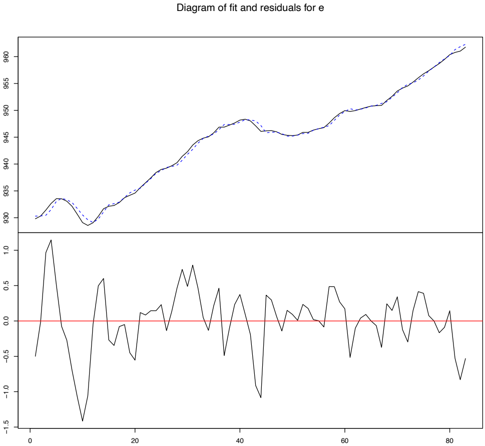
```
> summary(p1ct, equation = "e")

VAR Estimation Results:
=========================
Endogenous variables: prod, e, U, rw
Deterministic variables: both
Sample size: 83
Log Likelihood: -207.525
Roots of the characteristic polynomial:
0.9504 0.9504 0.9045 0.7513
Call:
VAR(y = Canada, p = 1, type = "both")
```

Example of a VAR model in R

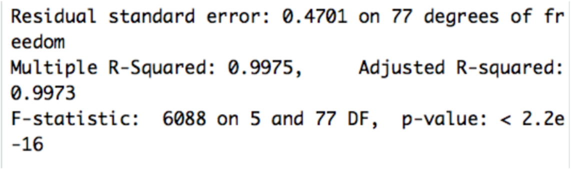


```
Estimation results for equation e:
==================================
e = prod.l1 + e.l1 + U.l1 + rw.l1 + const + trend

          Estimate Std. Error t value
prod.l1    0.19465    0.03612   5.389
e.l1       1.23892    0.08632  14.353
U.l1       0.62301    0.16927   3.681
rw.l1     -0.06776    0.02828  -2.396
const   -278.76121   75.18295  -3.708
trend     -0.04066    0.01970  -2.064
        Pr(>|t|)
prod.l1 7.49e-07 ***
e.l1     < 2e-16 ***
U.l1    0.000430 ***
rw.l1   0.018991 *
const   0.000392 ***
trend   0.042378 *
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Example of a VAR model in R

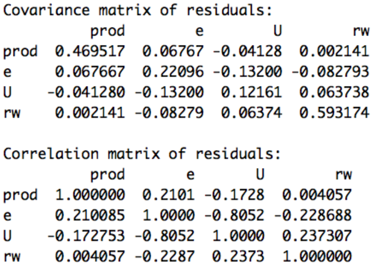


Example of a VAR model in R



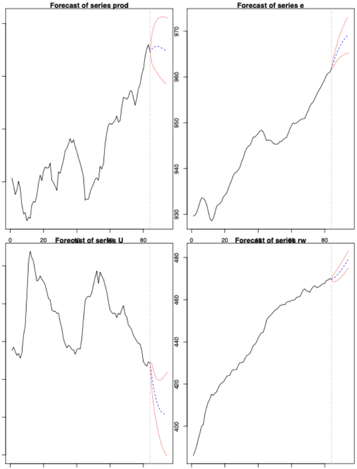
```
Residual standard error: 0.4701 on 77 degrees of fr
eedom
Multiple R-Squared: 0.9975,     Adjusted R-squared:
0.9973
F-statistic:  6088 on 5 and 77 DF,  p-value: < 2.2e
-16
```

Example of a VAR model in R



```
Covariance matrix of residuals:
          prod        e        U        rw
prod  0.469517  0.06767 -0.04128  0.002141
e     0.067667  0.22096 -0.13200 -0.082793
U    -0.041280 -0.13200  0.12161  0.063738
rw    0.002141 -0.08279  0.06374  0.593174

Correlation matrix of residuals:
          prod       e       U        rw
prod  1.000000  0.2101 -0.1728  0.004057
e     0.210085  1.0000 -0.8052 -0.228688
U    -0.172753 -0.8052  1.0000  0.237307
rw    0.004057 -0.2287  0.2373  1.000000
```

Example of a VAR model in R



Primitive versus Standard Form of VARs

- Does the VAR model include contemporaneous terms?
- So far, we have assumed that the VAR model is of the form:

$$y_{1t} = \beta_{10} + \beta_{11} y_{1t-1} + \alpha_{11} y_{2t-1} + \mu_1 t$$

$$y_{2t} = \beta_{20} + \beta_{21} y_{2t-1} + \alpha_{21} y_{1t-1} + \mu_2 t$$

- But what if the equations had a contemporaneous feedback term?

$$y_{1t} = \beta_{10} + \beta_{11} y_{1t-1} + \alpha_{11} y_{2t-1} + \alpha_{12} y_{2t} + \mu_1 t$$

$$y_{2t} = \beta_{20} + \beta_{21} y_{2t-1} + \alpha_{21} y_{1t-1} + \alpha_{22} y_{1t} + \mu_2 t$$

Primitive versus Standard Form of VARs

- We can write this as:

$$\begin{pmatrix} y_{1t} \\ y_{2t} \end{pmatrix} = \begin{pmatrix} \beta_{10} \\ \beta_{20} \end{pmatrix} + \begin{pmatrix} \beta_{11} & \alpha_{11} \\ \alpha_{21} & \beta_{21} \end{pmatrix} \begin{pmatrix} y_{1t-1} \\ y_{2t-1} \end{pmatrix} + \begin{pmatrix} \alpha_{12} & 0 \\ 0 & \alpha_{22} \end{pmatrix} \begin{pmatrix} y_{1t-1} \\ y_{2t-1} \end{pmatrix} + \begin{pmatrix} \mu_{1t} \\ \mu_{2t} \end{pmatrix}$$

- This VAR is in its primitive form. . . .

Primitive versus Standard Form of VARs

- We can take the contemporary LHS terms (left-hand side) and write:

$$\begin{pmatrix} 1 & -\alpha_{12} \\ -\alpha_{22} & 1 \end{pmatrix} \begin{pmatrix} y_{1t} \\ y_{2t} \end{pmatrix} = \begin{pmatrix} \beta_{10} \\ \beta_{20} \end{pmatrix} + \begin{pmatrix} \beta_{11} & \alpha_{11} \\ \alpha_{21} & \beta_{21} \end{pmatrix} \begin{pmatrix} y_{1t-1} \\ y_{2t-1} \end{pmatrix} + \begin{pmatrix} \mu_{1t} \\ \mu_{2t} \end{pmatrix}$$

or

$$By_t = \beta_0 + \beta_1 y_{t-1} + \mu_t$$

We can then multiply both sides by B^-1 and obtain:

$$y_t = B^-1\beta_0 + B^-1\beta_1 y_{t-1} + B^-1\mu_t$$

or

$$y_t = A_0 + A_1 y_{t-1} + e_t$$

This is known as a VAR in its Standard Form, and can be estimated by OLS.

Stability of VAR processes

- An important characteristic of a VAR(p) process is its stability.
- This means that it generates stationary time series with time invariant means, variances and covariances, given sufficient initial values. One can verify this by evaluating the characteristic polynomial:

$$det(I_K - A_{1z} - \dots - A_p z^p) = 0.$$

- For $|z| \leq 1$.
- If the solution of the above equation has a root for $z = 1$, then some or all of the variables in the VAR(p) process are integrated of order one, i.e., $I(1)$.

Stability of VAR processes

- In practice, the stability of an empirical VAR (p) process can be analyzed by considering the complementary form and calculating the eigenvalues of the coefficient matrix.
- A VAR (p) process can be written as a VAR (1) process:

$$\xi_t = A\xi_{t-1} + \nu_t$$

$$\xi_t = \begin{bmatrix} y_t \\ \vdots \\ y_{t-p+1} \end{bmatrix}, A = \begin{bmatrix} A_1 & A_2 & \dots & A_{p-1} & A_p \\ I & 0 & \dots & 0 & 0 \\ 0 & I & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & I & 0 \end{bmatrix}, \nu_t = \begin{pmatrix} \mu_t \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

- The dimensions of vectors ξ_t and ν_t are $(KP \times 1)$ and the dimension of the matrix A is (Kp × Kp). Again, if the eigenvalue modules of A are less than one, then the VAR (p)

Impulse response functions (IRF)

- Given a sample of endogenous variables $y_1, \ldots, y_T$, and sufficient prior sample values$y_{-p+1}, \ldots, y_0$, the coefficients of a VAR(p) process can be estimated efficiently by means of least squares applied separately to each of the equations.
- Once a VAR(p) model has been estimated, the avenue is open for further analysis.
- A researcher might/should be interested in diagnostic tests, such as tests of autocorrelation, heteroscedasticity or non-normality in the error term.
- However, he might be more interested in causal inference, forecasting the dynamic behavior of the empirical model, i.e. impulse response functions (IRF) and forecast error variance decomposition (FEVD).

Impulse response functions (IRF)

- **The impulse response functions (IRF)** and **the forecast error variance decomposition (FEVD)** are based on Wold's moving average decomposition for stable VAR (p) processes which is defined as:.

$$y_t = \Phi_0 u_t + \Phi_1 u_{t-1} + \Phi_2 u_{t-2} + \dots$$

$\Phi_0 = I_K$ and Φ_s can be calculated recursively according to:

$$\Phi_s = \sum_{j=1}^{s} \Phi_{s-j} A_j$$

for $s = 1, 2, \dots$ where $A_j = 0$ for $j > p$.

Impulse response functions (IRF)

- Finally, the predictions for horizons $h \leq 1$ of an empirical process VAR (p) can be generated recursively according to:

$$y_{T+h|T} = A_1 y_{T+h-1|T} + \ldots + A_p y_{T+h-p|T}$$

- where $y_{T+j|T} = y_{T+j}$ for $j \leq 0$.

Impulse response functions (IRF)

The forecast error covariance matrix is given as:

$$Cov \begin{bmatrix} y_{T+1} - y_{T+1|T} \\ \vdots \\ y_{T+h} - y_{T+h|T} \end{bmatrix} = \begin{bmatrix} I & 0 & \dots & 0 \\ \Phi_1 & I & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ \Phi_{h-1} & \Phi_{h-2} & \dots & I \end{bmatrix} (\sum_u \otimes I_h) \begin{bmatrix} I & 0 & \dots & 0 \\ \Phi_1 & I & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ \Phi_{h-1} & \Phi_{h-2} & \dots & I \end{bmatrix}^T$$

and matrices Φ_i are the empirical coefficient matrices of the Wold moving average representation of a stable VAR(p) process as shown above. The operator $\otimes$ is the Kronecker product.

Example of Impulse response functions (IRF)

- VAR models are often difficult to interpret.
- Solutions to this problem are the construction of impulse response functions and variance decompositions.
- Impulse response functions show the responsiveness of the dependent variables in the VAR to shocks to the error term.
- A unit shock is applied to each variable and its effects are stored.

Example of Impulse response functions (IRF)

- Consider, for example, a simple bivariate VAR(1):

$$y_{1t} = \beta_{10} + \beta_{11} y_{1t-1} + \alpha_{11} y_{2t-1} + \mu_{1t}$$

$$y_{2t} = \beta_{20} + \beta_{21} y_{2t-1} + \alpha_{21} y_{1t-1} + \mu_{2t}$$

- Initially, at $t = 1$ we assume a shock in the t ´error term μ_{11} of the first equation.

Example of Impulse response functions (IRF)

- This shock has a direct effect on y_{11}, of exactly the same amount.
- Considering that y_{21} is not yet effected, and assuming that $\mu_{21} = 0$ with $t = 1, \ldots. T$.
- In the second period $(t = 2)$, the original impact still has an effect on the lagged value of y_1.
- The effect on y_{12}, is $\beta_{11}\mu_{11}$, and the effect on y_{22}, is $\beta_{21}\mu_{11}$. In the third period, the effect on y_{13}, is not only $\beta_{11}(\beta_{11}\mu_{11})$, but also´en $\beta_{12}(\beta_{21}\mu_{11})$. Consequently, the effect on y_{23} is $\beta_{21}(\beta_{21}\mu_{11}) + \beta_{22}(\beta_{21}\mu_{11})$.

Example of Impulse response functions (IRF)

- **Example Orthogonal Impulse Responses**
- In the above impulse response model, we assume that the error terms of the different equation are uncorrelated.
- However, this assumption is rather implausible. A hypothetical shock in a single equation does not respond to a realistic fitting process. To control the correlation between the error terms, we have to use orthogonal impulse response sequences.

Example of Impulse response functions (IRF)

- **Example Orthogonal Impulse Responses**
- The idea is to modify the original moving average construction so that the residuals are uncorrelated, i.e., the residuals must be orthogonal to each other. Therefore, we can write:

$$y_t = \sum_{k=1}^{\infty} \hat{C}_k \nu_{t-k}$$

- where $\hat{C}_k = C_k G$ and G is a transformation matrix with the property GGI (Cholesky decomposition).

Example of Impulse response functions (IRF)

- The error terms of the modified system are as follows $\nu_{t-k} = G - 1\mu_{t-k}$. The variance-covariance matrix of μ_{t-k} is diagonal, according to the properties of G.
- However, the matrix G is not clearly defined by the Cholesky decomposition ($\Omega = G - 1G\prime - 1$, where Ω is the original variance-covariance matrix). In addition, we have to specify the order of the variables.
- The chosen order assumes the causal relationship between the variables. **The impulse response results may depend strongly on the order of the variables, especially when they are highly correlated.**

Variance decompositions

- Variance decompositions also allow us to examine the dynamics of VAR models.
- They provide the proportion of the movements in the dependent variables that are due to their "own" shocks, versus the shocks of other variables.
- The variance decomposition gives information about the relative importance of each variable shock in the VAR.

Example of Impulse response functions (IRF)

- However, in general, this is not true.
- The error terms would always be correlated to some degree.
- The dynamic fit of the reciprocal dependence is not immediately considerable.
- The impulse response test shows the effects of an exogenous shock on the whole process over time.
- Therefore, one can detect the dynamic relationships over time.

The impulse response function and variance decompositions:

- **The ordering of the variables.**
- Therefore, the notion of examining the effect of innovations separately has little meaning, since they have a common component.
- What is done is to ”orthogonalize” the innovations.
- Initially, look at the adjustment of the endogenous variables over time, after a hypothetical shock at t.
- This adjustment is compared with the process of orthogonalization. This adjustment is compared to the time series process without a shock, i.e., the real process.

The impulse response function and variance decompositions:

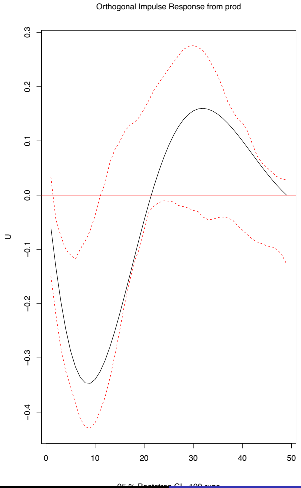
- **The ordering of the variables.**
- The impulse response sequences plot the difference between these two time paths. In the bivariate VAR, this problem can be addressed by assigning the entire effect of the common component to the first of the two variables in the VAR.
- In the general case where there are more variables, the situation is more complex, but the interpretation is the same.

Example of Impulse response functions (IRF)

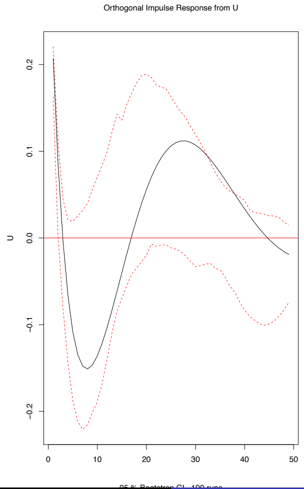
R Code

```
var.irf < − irf(p1ct, response = “U”, n.ahead = 10, boot = TRUE)
plot(var.irf )
var.irf1 < − irf(p1ct, impulse = “e”, response = “U”, n.ahead = 10,
boot = TRUE)
plot(var.irf1)
fevd.U < − fevd(p1ct, n.ahead = 48)$U
summary(fevd.U)
```

Example of Impulse response functions (IRF)



Example of Impulse response functions (IRF)

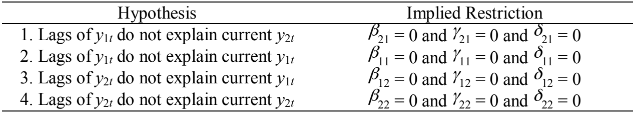


Example of Impulse response functions (IRF)

- This shock has a direct effect on y_{11}, of exactly the same amount.
- Considering that y_{21} is not yet effected, and assuming that $\mu_{21} = 0$ with $t = 1, ... T$.
- In the second period $(t = 2)$, the original impact still has an effect on the lagged value of y_1.
- The effect on y_{12}, is $\beta_{11}\mu_{11}$, and the effect on y_{22}, is $\beta_{21}\mu_{11}$. In the third period, the effect on y_{13}, is not only $\beta_{11}(\beta_{11}\mu_{11})$, but also´en $\beta_{12}(\beta_{21}\mu_{11})$. Consequently, the effect on y_{23} is $\beta_{21}(\beta_{21}\mu_{11}) + \beta_{22}(\beta_{21}\mu_{11})$.

Significance of a Block and Causality Test

- We might be interested in testing the following hypotheses, and their implicit constraints on the parameter matrices:



Hypothesis	Implied Restriction
1. Lags of y_{1t} do not explain current y_{2t}	$\beta_{21} = 0$ and $\gamma_{21} = 0$ and $\delta_{21} = 0$
2. Lags of y_{1t} do not explain current y_{1t}	$\beta_{11} = 0$ and $\gamma_{11} = 0$ and $\delta_{11} = 0$
3. Lags of y_{2t} do not explain current y_{1t}	$\beta_{12} = 0$ and $\gamma_{12} = 0$ and $\delta_{12} = 0$
4. Lags of y_{2t} do not explain current y_{2t}	$\beta_{22} = 0$ and $\gamma_{22} = 0$ and $\delta_{22} = 0$

- Each of these four hypotheses is an F-test, since each set of parameters is extracted from an equation. These tests can also be called Granger Causality Tests.

Significance of a Block and Causality Test

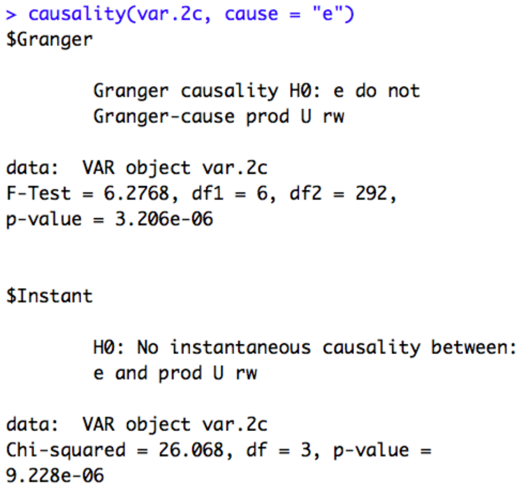
- Granger causality tests attempt to answer questions such as
  - Do changes in y_1 cause changes in y_2?
- If y_1 causes y_2, lags of y_{1} should be significant in the equation y2.
- If this is the case, then y_1 is said to "Granger-cause" y_2. If y_2 causes y_1, lags of y_2 must be significant in the y_1 equation.
- **If both sets of lags are significant, a "bidirectional causality relationship" exists.**

Example of a VAR model in R

R Code

```
var.cg < − VAR(Canada, p = 2, type = “const”)
causality(var.cg, cause = “e”)
grangertest(prod˜ e, order=4)
for (i in 1:4)
{
cat(”LAG =”, i)
print(causality(VAR(mydata, p = i, type = “const”), cause =
“e”)Granger )
}
```

Example VAR Model - Granger Causality Test



```
> causality(var.2c, cause = "e")
$Granger

        Granger causality H0: e do not
        Granger-cause prod U rw

data:  VAR object var.2c
F-Test = 6.2768, df1 = 6, df2 = 292,
p-value = 3.206e-06


$Instant

        H0: No instantaneous causality between:
        e and prod U rw

data:  VAR object var.2c
Chi-squared = 26.068, df = 3, p-value =
9.228e-06
```

Summary

- Advantages of the VAR model: a) it is relatively easy to specify and estimate, b) variables can be non-stationary, c) errors can be contemporaneously correlated.
- Disadvantages of the VAR model: many parameters.
- Thus, we have broaden our understanding of the relationship between time series, allowing for the possibility of feedbacks from idiosyncratic shocks.
- This dynamics can be captured by means of a vector autoregressive model (VAR), which is essentially a generalization of the analysis of autoregressive processes, in which, instead of considering a single variable, a vector of variables is considered.

Homework 4

Calibrate a VAR type model, and develop the following activities:
i) Find the unit root of each variable.
ii) Select the optimal lags of the model.
iii) Calibrate the model and interpret its results.
iv) Check the OLS estimation of the Standard Form.
v) Analyze the residuals and the stability of the model.
vi) Perform impulse analysis and variance decomposition.
vii) Perform out-of-sample forecasting using rolling windows.
viii) Compare the errors of your model, with the best model you can develop (remember Homework 2 and 3).

References

- Christopher A Sims. Macroeconomics and reality. Econometrica: Journal of the Econometric Society, pages 1–48, 1980.
- Helmut Lütkepohl & Markus Krätzig. Applied time series econometrics. Cambridge University Press, 2004.
- Hamilton, J. D. Time series analysis. Princeton University Press, Princeton, 1994.

Lecture V.- Cointegration Analysis

Marcelo Villena, PhD
Santa María University

November 2022

Outline

1. Introduction
2. Cointegration
   - Engle-Granger Cointegration Test
   - Error Correction Model
   - Johansen Cointegration Test
   - ARDL Models
3. Homework
4. References

Introduction

- If two or more non-stationary time series follow a common (or equilibrium) path in the long run, we can speak of cointegration. The classical test for cointegration boils down to determining whether a linear combination of the series is stationary or not. If, for example, two time series are cointegrated by a common factor (cointegrating vector), it is not possible to use a standard VAR approach. We have to account for this relationship and use an error correction model to obtain correct results.

Cointegration

- Suppose that $Y_t = I(1)$ and $X_t = I(1)$. Then Y_t and X_t are cointegrated, $CI(1,1)$, if there exists a β, such that $Y_t - \beta X_t = \varepsilon_t = I(0)$.
- This implies that there is a long-run relationship between Y_t and X_t, i.e. they do not "separate" over time. Hence the relation, $Y_t = \beta X_t + \varepsilon_t$, makes sense.
- If Y_t and X_t are not cointegrated, i.e., ε_t is also $I(1)$, then Y_t and X_t will become increasingly separated over time, and hence there will be no long-run relationship between these variables. Any regression of Y_t in X_t is "Spurious".

Cointegration

- In general, if Y_t and X_t are both $I(d)$, then Y_t and X_t are $CI(d, b)$ if $Y_t - \beta X_t = \varepsilon_t = I(d-b), b > 0$. If Y_t and X_t are cointegrated, this means that it is possible to model the long-run relationship between Y_t and X_t
- This is an alternative modeling strategy to trend elimination through differencing. The trend elimination procedure in general loses information.
- Therefore, it is possible to apply a "Seasonal Cointegration" procedure rather than trying to eliminate the seasonal effect by differencing.

Engle-Granger Cointegration Test.

- Estimate the cointegrated regression, $Y_t = \beta X_t + \varepsilon_t = I(0)$ using OLS to obtain the residuals et. Apply the Dickey-Fuller test (DF) and/or the Augmented Dickey-Fuller test (ADF) to examine whether the residuals have unit roots.
- If the hypothesis of unit roots is not rejected, this implies that ε_t is $I(1)$, which implies that Y_t and X_t are NOT cointegrated. It is important to note that the critical values for the DF and ADF tests are not valid to be used for Cointegration. Engle and Granger have calculated appropriate critical values.

Error Correction Model

- If Y_t and X_t are cointegrated, then there is a long-run relationship between these series and the short-run dynamics of this relationship can be described by the error correction model (ECM).
- Long-term relationship:

$$Y_t = \beta X_t + \varepsilon_t \tag{1}$$

Error Correction Model

- The error correction model, i.e. the short term dynamics is given by:

$$\triangle Y_t = \alpha \triangle X_t + \varphi[Y_{t-1} - \beta X_{t-1}] + \varepsilon_t \tag{2}$$

- where ε_t= white noise, i.e. $I(0)$.
- Interpretation: the current change in Y_t consists of two components:

(i) $\alpha \triangle X_t$: the short-run response to the current changes in X_t, and
(ii) $\varphi[Y_{t-1} - \beta X_{t-1}]$: the partial correction of the previous deviation of Y_t from its desired long-run level.

Error Correction Model

The two-step Engle-Granger procedure.
(i) Estimate the Cointegration regression´´ to obtain an estimate of the long-run parameter, and then,
(ii) Use the residuals to estimate the error correction model.

Cointegration in practice

- **Example IPSA - CU - S&P**
- We will analyze the long-run relationship between the Chilean stock market (IPSA), the US stock market (S&P 500), and the price of copper (cu).
- In particular, we will test the hypothesis of integration. First we present the stationarity test for the three variables.

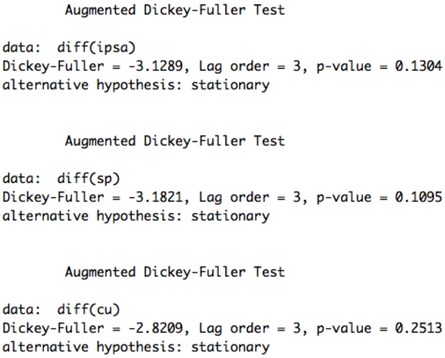
Cointegration in practice

R Code

```
library(tseries) # df, adf
library(dynlm) # time series regression
mydata< −read.csv (“cointegration.csv”)
ipsa< −ts(mydata$IPSA,frequency=12, start = c(2010,1))
cu< −ts(mydata$CU,frequency=12, start = c(2010,1))
sp< −ts(mydata$s.p,frequency=12, start = c(2010,1))
summary(mydata) adf.test(diff(ipsa)); adf.test(diff(sp));
adf.test(diff(cu))
```

Cointegration in practice

- **Example IPSA - CU - S&P**



```
        Augmented Dickey-Fuller Test

data:  diff(ipsa)
Dickey-Fuller = -3.1289, Lag order = 3, p-value = 0.1304
alternative hypothesis: stationary


        Augmented Dickey-Fuller Test

data:  diff(sp)
Dickey-Fuller = -3.1821, Lag order = 3, p-value = 0.1095
alternative hypothesis: stationary


        Augmented Dickey-Fuller Test

data:  diff(cu)
Dickey-Fuller = -2.8209, Lag order = 3, p-value = 0.2513
alternative hypothesis: stationary
```

Cointegration in practice

- **Example IPSA - CU - S&P**
- We now run the two-stage Engle-Granger cointegration model. We first look for a long-run relationship of the variables, and find that the IPSA and CU relationship is the strongest.

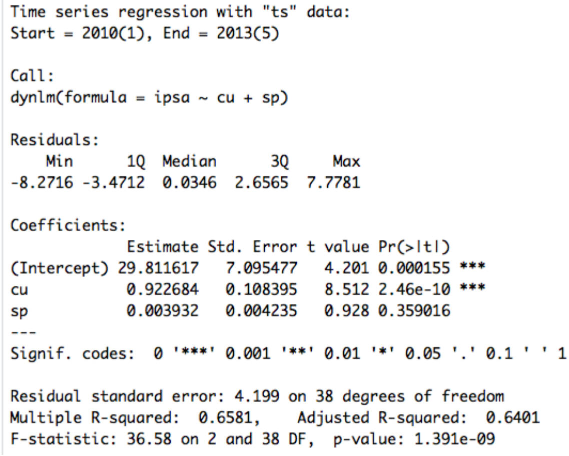
Cointegration in practice

R Code

```
ipsa.reg1 < − dynlm(ipsa˜ cu + sp)
summary(ipsa.reg1)
ipsa.reg2 < − dynlm(ipsa˜ L(cu, 1:3))
summary(ipsa.reg2)
ipsa.reg3 < − dynlm(ipsa˜ cu)
summary(ipsa.reg3)
residuos < − ipsa.reg3[[“residuals”]]
plot(residuos);
adf.test(residuos); qqnorm(residuos); qqline(residuis)
```

Cointegration in practice

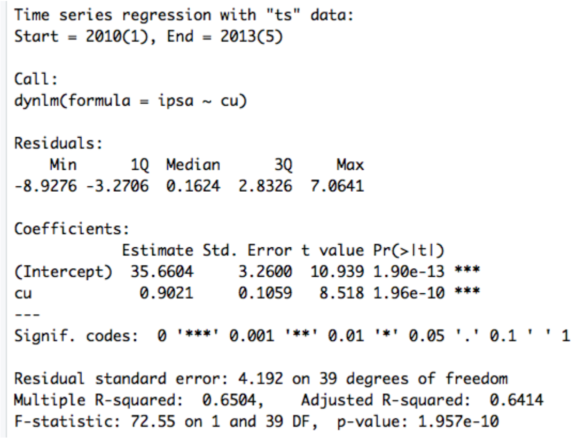
- **Example IPSA - CU - S&P**



```
Time series regression with "ts" data:
Start = 2010(1), End = 2013(5)

Call:
dynlm(formula = ipsa ~ cu + sp)

Residuals:
    Min      1Q  Median      3Q     Max 
-8.2716 -3.4712  0.0346  2.6565  7.7781 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 29.811617   7.095477   4.201 0.000155 ***
cu           0.922684   0.108395   8.512 2.46e-10 ***
sp           0.003932   0.004235   0.928 0.359016    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.199 on 38 degrees of freedom
Multiple R-squared:  0.6581,	Adjusted R-squared:  0.6401 
F-statistic: 36.58 on 2 and 38 DF,  p-value: 1.391e-09
```

Cointegration in practice

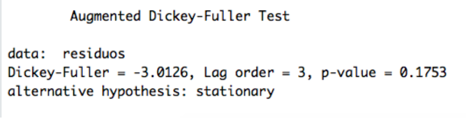
- **Example IPSA - CU - S&P**



```
Time series regression with "ts" data:
Start = 2010(1), End = 2013(5)

Call:
dynlm(formula = ipsa ~ cu)

Residuals:
    Min      1Q  Median      3Q     Max
-8.9276 -3.2706  0.1624  2.8326  7.0641

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  35.6604     3.2600  10.939 1.90e-13 ***
cu            0.9021     0.1059   8.518 1.96e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.192 on 39 degrees of freedom
Multiple R-squared:  0.6504,	Adjusted R-squared:  0.6414
F-statistic: 72.55 on 1 and 39 DF,  p-value: 1.957e-10
```

Cointegration in practice

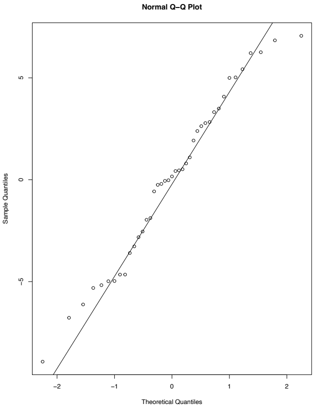
- **Example IPSA - CU - S&P**



```
        Augmented Dickey-Fuller Test

data:  residuos
Dickey-Fuller = -3.0126, Lag order = 3, p-value = 0.1753
alternative hypothesis: stationary
```

Cointegration in practice

- **Example IPSA - CU - S&P**



Cointegration in practice

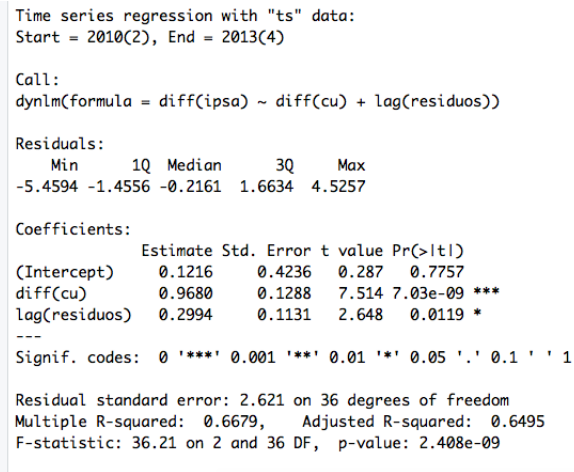
- **Example IPSA - CU - S&P**
- The ECM using a dynamic regression

R Code

```
ipsa.reg4 < − dynlm(diff(ipsa)˜ diff(cu) + lag(residuos))
summary(ipsa.reg4)
```

Cointegration in practice

- **Example IPSA - CU - S&P**



```
Time series regression with "ts" data:
Start = 2010(2), End = 2013(4)

Call:
dynlm(formula = diff(ipsa) ~ diff(cu) + lag(residuos))

Residuals:
    Min      1Q  Median      3Q     Max 
-5.4594 -1.4556 -0.2161  1.6634  4.5257 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)     0.1216     0.4236   0.287   0.7757    
diff(cu)        0.9680     0.1288   7.514 7.03e-09 ***
lag(residuos)   0.2994     0.1131   2.648   0.0119 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.621 on 36 degrees of freedom
Multiple R-squared:  0.6679,    Adjusted R-squared:  0.6495 
F-statistic: 36.21 on 2 and 36 DF,  p-value: 2.408e-09
```

Johansen Cointegration Test

- **Testing for and Estimating Cointegrating Systems Using the Johansen Technique Based on VARs**
- The Johansen test, see Johansen(1988), is a cointegration test that allows more than one cointegrating relationship, unlike the Engle-Granger method.
- There are two types of Johansen test, either trace or eigenvalue, and the inferences may be slightly different.
- This test is based on maximum likelihood estimation and two statistics: maximum eigenvalues and a trace statistic. This is related to the rank of the matrix. If the rank is zero, there is no cointegration relationship. If the rank is one, there is one, if two there are two and so on.

Cointegration in a VAR: Vector Error-Correction Models

- **In our VAR analysis, we have assumed that the variables in the model are stationary and ergodic.**
- On the other hand, we recently saw that variables that are individually non-stationary can be cointegrated. For the simple case of two variables and a cointegrating relationship, we saw that an error-correction model is the appropriate econometric specification.
- In this model, the equation is differentiated and an error correction term is included, which measures the deviation of the previous period from the long-run equilibrium.
- We now consider how cointegrated variables can be used in a VAR using a **vector error correction model (VEC model)**.

Cointegration in a VAR: Vector Error-Correction Models

- **In general, when variables are non stationary,** a VAR model in levels is not appropriate since it is a spurious regression which is a non-interpretable regression.
- **However, although variables are non stationary but when cointegrations exist, a VAR model in levels can be estimated which has a long-term interpretation.**
- In other words, the cointegration indicates one or more long-run equilibriums or stationary relationships among non-stationary variables.
- To determine whether VAR model in levels is possible or not, we need to transform VAR model in levels to a VECM model in differences (with error correction terms), to which the Johansen test for cointegration is applied.

Cointegration in a VAR: Vector Error-Correction Models

- A two-variable VEC model If two series I(1), say Y_t and X_t, are cointegrated, then there exists a unique α_0 and α_1 such that $\nu_t = y_t - \alpha_0 - \alpha_1 x_t$ is I(0). In the one-equation cointegration model, we saw that the error correction model:, had the following form.

$$\begin{gathered}\triangle y_t = \beta_0 + \beta_1 \triangle x_t + \lambda \nu_{t-1} + \varepsilon_t = \\ \beta_0 + \beta_1 \triangle x_t + \lambda(y_{t-1} - \alpha_0 - \alpha_1 x_{t-1}) + \varepsilon_t\end{gathered} \tag{3}$$

Cointegration in a VAR: Vector Error-Correction Models

- All terms in the above equation are $I(0)$, provided that the coefficients α (the "cointegration vector") are known or at least consistently estimated.
- The terminus ν_{t-1} is the magnitude by which y was above or below its long-run equilibrium value in the previous period.
- The coefficient λ (which we expect to be negative) represents **the amount of "correction" of this period,** $(t-1)$, disequilibrium occurring in period t.

Johansen Cointegration Test

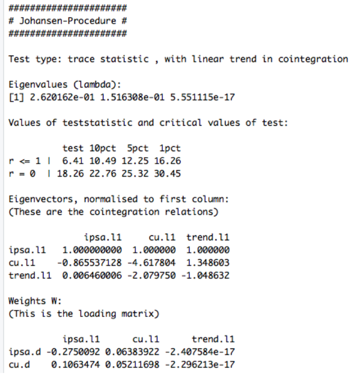
- Johansen Cointegration Test

R Code

```
install.packages(“urca”)
library(urca)
data_ipsa< −data.frame(ipsa,cu)
cointegration < − ca.jo(data_ipsa, type=“trace”, ecdet=“trend”,
spec=“transitory”)
summary(cointegration)
```

Cointegration in practice

- **Example IPSA - CU - S&P**



```
######################
# Johansen-Procedure #
######################

Test type: trace statistic , with linear trend in cointegration

Eigenvalues (lambda):
[1] 2.620162e-01 1.516308e-01 5.551115e-17

Values of teststatistic and critical values of test:

          test 10pct  5pct  1pct
r <= 1 |  6.41 10.49 12.25 16.26
r = 0  | 18.26 22.76 25.32 30.45

Eigenvectors, normalised to first column:
(These are the cointegration relations)

              ipsa.l1     cu.l1  trend.l1
ipsa.l1   1.000000000  1.000000  1.000000
cu.l1    -0.865537128 -4.617804  1.348603
trend.l1  0.006460006 -2.079750 -1.048632

Weights W:
(This is the loading matrix)

          ipsa.l1      cu.l1      trend.l1
ipsa.d -0.2750092 0.06383922 -2.407584e-17
cu.d    0.1063474 0.05211698 -2.296213e-17
```

Cointegration in a VAR: Vector Error-Correction Models

- The expected sign of λ_x depends on the sign of α_1.
- We expect $\delta \triangle x_t / \delta x_{t-1} = -\lambda_x \alpha_1 < 0$ for the same reason that we expect $\delta \triangle y_t / \delta y_{t-1} = \lambda_y < 0$: *if* x_{t-1} is above its long-run ratio to y, then we expect $\triangle x_t$ to be negative, other things constant.

Cointegration in a VAR: Vector Error-Correction Models

- A simple and concrete example may help to clarify the role of error correction terms in a VEC model.
- Suppose that the long-run cointegration relationship is $y_t = x_t$, so that $\alpha_0 = 0$ and $\alpha_1 = 1$
- The error correction term between parentesis in each equation of the VAR system is now $y_{t-1} - x_{t-1}$, the difference between y and x in the previous period.
- Suppose that due to previous shocks, $y_{t-1} = x_{t-1} + 1$ or that y is above its long-run equilibrium relation with x by one unit (or, equivalently, x is below its long-run equilibrium relation with y by one unit).

Cointegration in a VAR: Vector Error-Correction Models

- To move toward long-run equilibrium at period t, we expect (if there are no other changes) $\triangle y_t < 0$ and $\triangle x_t > 0$.
- $\triangle y_t$ changes in response to this equilibrium by $\lambda_y(y_{t-1} - x_{t-1}) = \lambda_y$, for a stable adjustment to occur, $\lambda_y < 0$; y is too high, so it must decrease in response to the disequilibrium.
- The corresponding change in$\lambda_x(y_{t-1} - x_{t-1}) = \lambda_x$.
- Since x is "too low," the stable fit requires that the response in x be positive, so we need $\lambda_x > 0$.
- Note that if the long-run relationship between y and x were inverse $(\alpha_1 < 0)$, then x would need´ıa decrease to move toward equilibrium and we would need´ın $\lambda_x < 0$.
- The expected sign on λ_x depends on the sign of α_1.

Cointegration in preactice

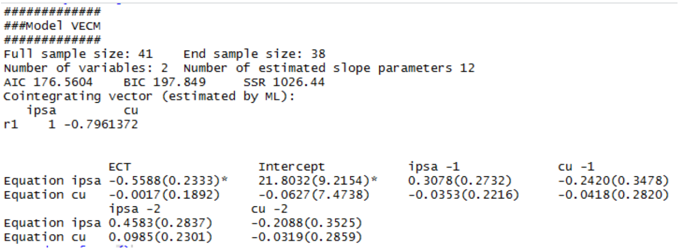
- **Example IPSA - CU - S&P**
- The ECM using dynamic regression

R Code

```
library(tsDyn)
data_ipsa< −data.frame(ipsa,cu)
#Fit a VECM with Engle-Granger 2OLS estimator:
vecm.eg< −VECM(data_ipsa, lag=2)
#Fit a VECM with Johansen MLE estimator:
vecm.jo< −VECM(data_ipsa, lag=2, estim="ML")
```

Cointegration in practice

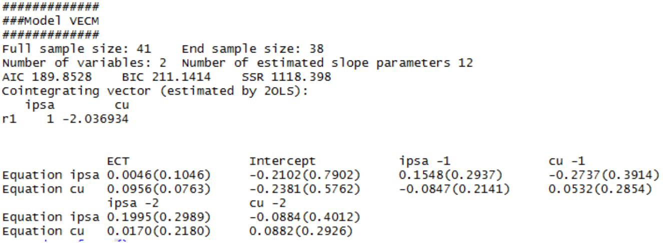
- **Example IPSA - CU - S&P**



```
#############
###Model VECM
#############
Full sample size: 41    End sample size: 38
Number of variables: 2  Number of estimated slope parameters 12
AIC 176.5604    BIC 197.849     SSR 1026.44
Cointegrating vector (estimated by ML):
   ipsa         cu
r1    1 -0.7961372


              ECT                Intercept          ipsa -1            cu -1
Equation ipsa -0.5588(0.2333)*   21.8032(9.2154)*   0.3078(0.2732)     -0.2420(0.3478)
Equation cu   -0.0017(0.1892)    -0.0627(7.4738)    -0.0353(0.2216)    -0.0418(0.2820)
              ipsa -2           cu -2
Equation ipsa 0.4583(0.2837)    -0.2088(0.3525)
Equation cu   0.0985(0.2301)    -0.0319(0.2859)
```

Cointegration in practice

- **Example IPSA - CU - S&P**



```
#############
###Model VECM
#############
Full sample size: 41     End sample size: 38
Number of variables: 2   Number of estimated slope parameters 12
AIC 189.8528     BIC 211.1414     SSR 1118.398
Cointegrating vector (estimated by 2OLS):
   ipsa        cu
r1    1 -2.036934


              ECT                Intercept           ipsa -1             cu -1
Equation ipsa 0.0046(0.1046)     -0.2102(0.7902)     0.1548(0.2937)      -0.2737(0.3914)
Equation cu   0.0956(0.0763)     -0.2381(0.5762)     -0.0847(0.2141)     0.0532(0.2854)
              ipsa -2            cu -2
Equation ipsa 0.1995(0.2989)     -0.0884(0.4012)
Equation cu   0.0170(0.2180)     0.0882(0.2926)
```

Cointegration with ARDL Models

- The ARDL cointegration technique, see Pesran et al (2001), does not require prior tests for unit roots unlike other methods.
- Consequently, the ARDL cointegration technique is preferable when dealing with variables that are integrated with different order, $I(0)$, $I(1)$ or a combination of the two.
- The long-run relationship of the underlying variables is detected through the F-statistic (Wald test). In this approach, the long-run relationship of the series is established when the Fstatistic exceeds the critical value band.
- The great advantage of this approach lies in its identification of cointegrating vectors where there are multiple cointegrating vectors.

Cointegration in preactice

- **Example IPSA - CU - S&P**

R Code

```
ipsa.reg4 < − auto.ardl(ipsa˜ cu)
ipsa.reg4 < − ardl(ipsa˜ cu)
```

Cointegration in practice

- **Example IPSA - CU - S&P**

```
$best_model

Time series regression with "ts" data:
Start = 6, End = 41

Call:
dynlm::dynlm(formula = full_formula, data = data, start = start,
    end = end)

Coefficients:
(Intercept)  L(ipsa, 1)  L(ipsa, 2)  L(ipsa, 3)          cu    L(cu, 1)
    29.2367      0.7918     -0.0708     -0.3911      1.0294     -0.9044
   L(cu, 2)    L(cu, 3)    L(cu, 4)    L(cu, 5)
     0.1513      0.3008      0.1352     -0.2699


$best_order
[1] 3 5
```

Homework V

Using the data from your previous task.

1. Test your hypotheses in light of Johansson's cointegration model.
2. Demonstrate the consistency of your previous results using the Engle-Granger Cointegration Test.
3. Calibrate and comment on the results of an ARDL model.
4. Discuss the advantages and disadvantages of the VAR model and the different cointegration models.

References

- Robert F Engle and Clive WJ Granger. Co-integration and error correction: representation, estimation, and testing. Econometrica: journal of the Econometric Society, pages 251–276, 1987.
- Søren Johansen. Statistical analysis of cointegration vectors. Journal of economic dynamics and control, 12(2-3):231–254, 1988.
- M Hashem Pesaran, Yongcheol Shin, and Richard J Smith. Bounds testing approaches to the analysis of level relationships. Journal of applied econometrics, 16(3):289–326, 2001.

Lecture VI.- Non-linear Models: Volatility Forecasting

Marcelo Villena, PhD
Santa María University

September 12, 2022

EX UMBRA IN SOLEM

Outline

1. Non-linear models
2. ARCH Models
3. GARCH models
4. Homework
5. References

An excursion into the non-linear world

- Rationale: Structural linear (and time series) models cannot explain a number of important features common to many financial data.

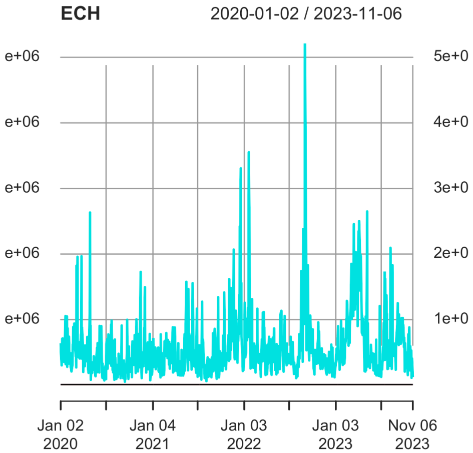
1. Leptokurtosis
2. Volatility clustering or volatility pooling
3. Leverage effects

Example of a financial series
Chilean Stock Exchange

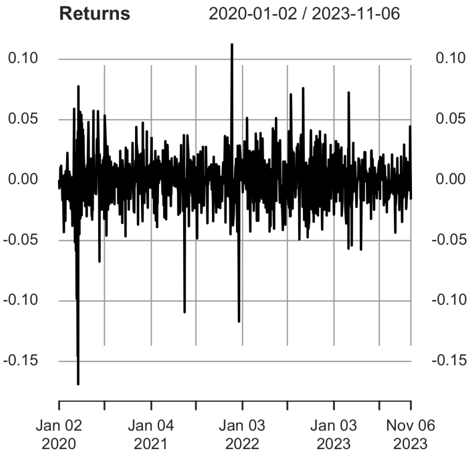
R code

```
rm(list=ls())
getSymbols(”ECH”, from=”2020-01-01”)
Returns = diff(log(Ad(ECH)))
Returns[as.character(head(index(Ad(ECH)),1))] = 0
adf.test(Ad(ECH))
adf.test(Returns)
plot(ECH)
plot(Returns)
```

Daily prices ECH - IGPA



Daily returns ECH - IGPA



An excursion into the non-linear world

Our "traditional" structural model could be something like:

$$Y_t = \beta_1 + \beta_2 X_{2t} + \beta_2 X_{2t} + \ldots + \beta_k X_{kt} + \mu_t \tag{1}$$

where μ_t = white noise, i.e. $\mu_t \backsim N(0, \sigma^2)$.

[?] define a non-linear data generation process as one that can be written as:

$$Y_t = f(\mu_t, \mu_{t-1}, \mu_{t-2}, \ldots) \tag{2}$$

where μ_t is a random error term (iid, Independent and identically distributed random variable) and f is a nonlinear function.

An excursion into the non-linear world

A more specific definition is:

$$Y_t = g(\mu_{t-1}, \mu_{t-2}, \dots) + \mu_t \sigma^2(\mu_{t-1}, \mu_{t-2}, \dots) \tag{3}$$

where g is a function of past error terms only, and σ^2 is a variance term.

Models with nonlinear $g(\cdot)$ are "nonlinear in mean", while those nonlinear in $\sigma^2(\cdot)$ are "nonlinear in variance".

Types of nonlinear models

- The linear paradigm is very useful. Many seemingly nonlinear relationships can be linearized, through an appropriate transformation. On the other hand, many relationships in finance are likely to be intrinsically nonlinear.

  There are many types of nonlinear models, e.g..
  - ARCH / GARCH
  - Switching models
  - Bilinear models
  - Neural networks Models

Tests for nonlinearity

- The ”traditional” time series analysis tools (ACF, spectral analysis, etc.) may not find evidence that we can use a linear model, but the data may still be non-independent.
- Portmanteau tests for nonlinear dependence have been developed. The simplest is the Ramsey RESET, which takes the form:

$$\hat{\mu_t} = \beta_0 + \beta_1 \hat{y}^2 + \beta_2 \hat{y}^3 + \dots \beta_{p-1} \hat{y}^p + \nu_t \tag{4}$$

- One particular nonlinear model that has proven to be very useful in finance is the ARCH model, developed by [**?**].

Revisited Heteroscedasticity

- As we saw above, an example of a structural model is:

$$Y_t = \beta_1 + \beta_2 X_{2t} + \beta_2 X_{2t} + \ldots + \beta_k X_{kt} + \mu_t \tag{5}$$

- where μ_t= white noise, i.e. $\mu_t \backsim N(0, \sigma^2)$.
- The assumption that the variance of the errors is constant is known as homoscedasticity, i.e., i.e., the variance of the errors is constant. as homocedasticity, i.e. $Var(\mu_t) = \sigma^2$
- What happens if the variance of the errors is not constant?
- **Heteroscedasticity**
- This implies that the standard error estimates could be wrong.
- In practice, the variance of errors is NOT constant over time, e.g. for financial data.

Autoregressive conditional heteroskedasticity models: ARCH models

- Let us use a model that does not assume that the variance is constant. Recall the definition of variance of μ_t.

$$\sigma_t^2 = Var(\mu_t|\mu_{t-1}, \mu_{t-2}, \dots) = E((\mu_t - E(\mu_t))^2|\mu_{t-1}, \mu_{t-2}, \dots) \tag{6}$$

- We normally assume that $E(\mu_t) = 0$, hence:

$$\sigma_t^2 = Var(\mu_t|\mu_{t-1}, \mu_{t-2}, \dots) = E(\mu_t^2|\mu_{t-1}, \mu_{t-2}, \dots) \tag{7}$$

- On what will the present value of the variance of the errors depend?
  - On the square of the previous terms of error.

Autoregressive conditional heteroskedasticity models: ARCH models

- This leads us to the model known as ARCH, ”autoregressive conditionally heteroscedastic model”:

$$\sigma_t^2 = \alpha_0 + \alpha_1 \mu_{t-1}^2 \tag{8}$$

- In particular, the above model represents an ARCH(1).

Autoregressive conditional heteroskedasticity models: ARCH models

- The full model is:

$$Y_t = \beta_1 + \beta_2 X_{2t} + \beta_2 X_{2t} + \ldots + \beta_k X_{kt} + \mu_t, \mu_t\tilde{}\,N(0, \sigma^2) \quad (9)$$

where $\sigma_t^2 = \alpha_0 + \alpha_1 \mu_{t-1}^2$

- We can easily extend this to the general case where the error variance depends on q squared lags of error squared:

$$\sigma_t^2 = \alpha_0 + \alpha_1 \mu_{t-1}^2 + \alpha_2 \mu_{t-2}^2 + \ldots + +\alpha_q \mu_{t-q}^2 \quad (10)$$

- This is an ARCH(q) model.

Autoregressive conditional heteroskedasticity models: ARCH models

- Instead of calling the variance, σ_t^2 in the literature it is usually called h_t, so the model is in short:

$$Y_t = \beta_1 + \beta_2 X_{2t} + \beta_2 X_{2t} + \ldots + \beta_k X_{kt} + \mu_t \qquad (11)$$

- with $\mu_t \backsim N(0, \sigma^2)$, and where :

$$h_t = \alpha_0 + \alpha_1 \mu_{t-1}^2 + \alpha_2 \mu_{t-2}^2 + \ldots + + \alpha_q \mu_{t-q}^2 \qquad (12)$$

Another way to represent ARCH Models

- For example, consider an ARCH (1). Instead of the representation above, we can write

$$Y_t = \beta_1 + \beta_2 X_{2t} + \beta_2 X_{2t} + \ldots + \beta_k X_{kt} + \mu_t \tag{13}$$

- with $\mu_t = \nu_t \sigma_t$, and where

$$\sigma_t = \sqrt{\alpha_0 + \alpha_1 \mu_{t-1}^2} \tag{14}$$

- The two forms represent different ways of expressing exactly the same model. The first form is easier to understand, while the second better represents the simulation of an ARCH model.

The "ARCH effect" test

[1] First, a linear regression is run, e.g.:

$$Y_t = \beta_1 + \beta_2 X_{2t} + \beta_2 X_{2t} + \ldots + \beta_k X_{kt} + \mu_t \qquad (15)$$

and the residuals are stored,
[2] Next́ the residuals are squared, and a regression is run on the q eigen lags for the q-order ARCH test, i.e., run the regression:

$$\hat{\mu}_t^2 = \gamma_0 + \gamma_1 \mu_{\widehat{t-1}}{}^2 + \gamma_2 \mu_{\widehat{t-2}}{}^2 + \ldots + \gamma_q \mu_{\widehat{t-q}}{}^2 + \nu_t \qquad (16)$$

[3] The test statistic is defined as TR2 (the number of observations multiplied by the multiple correlation coefficient) from the last regression, and is distributed as a $\chi^2(q)$.

The "ARCH effect" test

[4] The null and alternative hypotheses are:

$$H0 : \gamma_1 = 0 y \gamma_2 = 0 y \gamma_3 = 0 y \ldots \gamma_q = 0.$$
$$H1 : \gamma_1 \neq 0 y \gamma_2 \neq 0 y \gamma_3 \neq 0 y \ldots \gamma_q \neq 0.$$

- If the value of the statistical test is greater than the critical value of the distribution$\chi^2(q)$, the null hypothesis is rejected.
- Note that the ARCH test is also applied directly to the profitability, rather than to the residuals in step 1 above.

Main problems of ARCH models

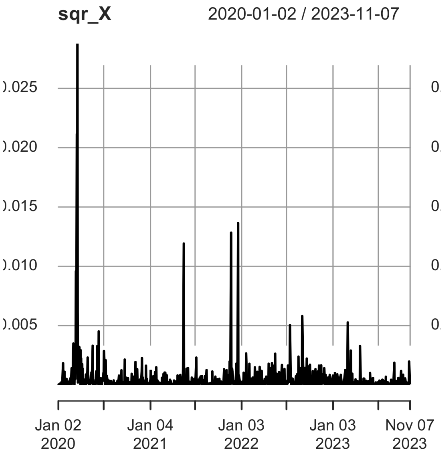
- How do we decide the best q?
  - The required value of q could be very large.
- Non-negativity constraints may be violated.
- When estimating an ARCH model, we require $\alpha_i > 0 \; \forall i = 1, 2, ..., q$ (since the variance cannot be negative).
- **A natural extension of an ARCH(q) model, which avoids some of these problems, is the GARCH model that we will see below.**

Example of an ARCH model

R code

```
# ARCH
avg_returns<-mean(Returns)
X <- Returns - avg_returns
sqr_X <- X*X plot(sqr_X)
arch1 <- lm(sqr_X˜lag(sqr_X,1)+lag(sqr_X,2))
summary(arch1)
arch2 <- lm(sqr_X˜lag(sqr_X,1))
summary(arch2)
```

Example of an ARCH model



Example of an ARCH model

```
Call:
lm(formula = sqr_X ~ lag(sqr_X, 1) + lag(sqr_X, 2))

Residuals:
       Min         1Q     Median         3Q        Max
-0.0108943 -0.0003441 -0.0002348  0.0000465  0.0207733

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.685e-04  4.872e-05   5.511 4.57e-08 ***
lag(sqr_X, 1) 8.618e-02  3.005e-02   2.868  0.00422 **
lag(sqr_X, 2) 3.586e-01  3.005e-02  11.934  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.001394 on 965 degrees of freedom
  (2 observations deleted due to missingness)
Multiple R-squared:  0.1443,    Adjusted R-squared:  0.1426
F-statistic: 81.39 on 2 and 965 DF,  p-value: < 2.2e-16
```

Example of an ARCH model

```
Call:
lm(formula = sqr_X ~ lag(sqr_X, 1))

Residuals:
       Min         1Q     Median         3Q        Max 
-0.0019881 -0.0004157 -0.0003161 -0.0000148  0.0281263 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)   4.181e-04  5.035e-05   8.303 3.39e-16 ***
lag(sqr_X, 1) 1.345e-01  3.187e-02   4.220 2.67e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.001492 on 967 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.01808,	Adjusted R-squared:  0.01707 
F-statistic: 17.81 on 1 and 967 DF,  p-value: 2.674e-05
```

Generalised ARCH - GARCH Models

- Introduced by [?] lets the conditional variance be dependent on its own lags. Thus, the variance equatioń is now:

$$\sigma_t^2 = \alpha_0 + \alpha_1 \mu_{t-1}^2 + \beta \sigma_{t-1}^2 \tag{17}$$

- This is a GARCH (1,1), which is equivalent to an ARMA (1,1) of the variance equation.
- We could also write:

$$\sigma_{t-1}^2 = \alpha_0 + \alpha_1 \mu_{t-2}^2 + \beta \sigma_{t-2}^2 \tag{18}$$

$$\sigma_{t-2}^2 = \alpha_0 + \alpha_1 \mu_{t-3}^2 + \beta \sigma_{t-3}^2 \tag{19}$$

Generalised ARCH - GARCH Models

Replacing (18) in (21):

$$\sigma_t^2 = \alpha_0 + \alpha_1 \mu_{t-1}^2 + \beta(\alpha_0 + \alpha_1 \mu_{t-2}^2 + \beta \sigma_{t-2}^2) \tag{20}$$

$$\sigma_t^2 = \alpha_0(1+\beta) + \alpha_1 \mu_{t-1}^2 (1 + \beta L) + \beta \sigma_{t-1}^2 \tag{21}$$

If we keep replacing terms, the GARCH(1,1) model can be written as an ARCH model of infinite order. Thus, we can extend GARCH(1,1) to a GARCH(p, q):

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{q} \alpha_i \mu_{t-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2 \tag{22}$$

Generalised ARCH - GARCH Models

- In general, a GARCH(1,1) model is sufficient to capture the clustered volatility of the data.

$$\sigma_t^2 = \alpha_0 + \alpha_1 \mu_{t-1}^2 + \beta_1 \sigma_{t-1}^2 \tag{23}$$

- Why is a GARCH model better than an ARCH model?
  - More parsimonious - avoids overfitting
  - Less likely to violate non-negativity constraints

The unconditional variance under the GARCH specification.

- By calculating the unconditional variance, we can estimate the standard deviation we are looking for. In this way, we will derive the unconditional variance from the ARCH and GARCH models. In addition, a note on the daily scale variation is presented.
- **ARCH Unconditional Variance**
- We assume a process that could be represented by an econometric model, for example:

$$y_t = \mu + \epsilon_t \tag{24}$$

- with $\epsilon_t \sim \left(0, \sigma_t^2\right)$. We assume that the conditional variance follows an ARCH (1) type model, i.e.:

$$\sigma_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2 \tag{25}$$

The unconditional variance under the GARCH specification.

Using the unconditional expectation operator, we have:

$$\mathbb{E}\left(\sigma_t^2\right) = \sigma_t^2$$

$$\mathbb{E}\left(\alpha_0\right) = \alpha_0$$

$$\mathbb{E}\left(\epsilon_{t-1}^2\right) = \sigma_t^2$$

We have then:

$$\sigma_t^2\left(1-\alpha_1\right) = \alpha_0 \tag{26}$$

$$\Rightarrow \sigma_t^2 = \frac{\alpha_0}{1-\alpha_1} \tag{27}$$

The unconditional variance under the GARCH specification.

- If we generalize to an ARCH model (q), we obtain:

$$\begin{aligned} \sigma_t^2 &= \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \alpha_2 \epsilon_{t-2}^2 + \cdots + \alpha_q \epsilon_{t-q}^2 \\ &= \alpha_0 + \sum_{k=1}^{q} \alpha_k \epsilon_{t-k}^2 \end{aligned} \tag{28}$$

where:

$$\mathbb{E}\left(\epsilon_{t-1}^2\right) = \mathbb{E}\left(\epsilon_{t-2}^2\right) = \cdots = \mathbb{E}\left(\epsilon_{t-q}^2\right) = \sigma_t^2$$

The unconditional variance under the GARCH specification.

then:

$$\begin{aligned}\sigma_t^2 &= \frac{\alpha_0}{1-\alpha_1-\alpha_2-\cdots-\alpha_q} \\ &= \frac{\alpha_0}{1-\sum_{k=1}^{q}\alpha_k} \qquad (29)\end{aligned}$$

The unconditional variance under the GARCH specification.

- **GARCH Unconditional Variance**
- Assume the same process given previously, but this time the variance also depends on its own *p* lags:

$$\begin{aligned} \sigma_t^2 &= \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \alpha_2 \epsilon_{t-2}^2 + \cdots + \alpha_q \epsilon_{t-q}^2 + \beta_1 \sigma_{t-1}^2 + \beta_2 \sigma_{t-2}^2 + \cdots \\ &= \alpha_0 + \sum_{k=1}^{q} \alpha_k \epsilon_{t-k}^2 + \sum_{l=1}^{p} \alpha_l \sigma_{t-l}^2 \end{aligned}$$

- The above equation gives a GARCH (p, q) model. Using

$$\mathbb{E}\left(\sigma_{t-1}^2\right) = \mathbb{E}\left(\sigma_{t-2}^2\right) = \cdots = \mathbb{E}\left(\sigma_{t-p}^2\right) = \sigma_t^2$$

The unconditional variance under the GARCH specification.

- **GARCH Unconditional Variance**
- In this way we have:

$$
\begin{aligned}
\sigma_t^2 &= \frac{\alpha_0}{1-\alpha_1-\alpha_2-\cdots-\alpha_q-\beta_1-\beta_2-\cdots-\beta_p} \\
&= \frac{\alpha_0}{1-\sum_{k=1}^{q}\alpha_k-\sum_{l=1}^{p}\beta_l}
\end{aligned}
\tag{31}
$$

The unconditional variance under the GARCH specification.

- **Scaling Volatility**
- The calculation of volatility and scaling at different time horizons is possible only in cases where changes in the log of the asset price v_t are independently and identically distributed (iid).

$$v_t = v_{t-1} + \varepsilon_t \qquad \varepsilon_t \sim (0, \sigma^2) \tag{32}$$

- Then 1 day of return is:

$$v_t - v_{t-1} = \varepsilon_t$$

- with standard deviation σ.

The unconditional variance under the GARCH specification.

- Similarly, the h-day return is:

$$v_t - v_{t-h} = \sum_{i=0}^{h-1} \varepsilon_{t-i} \tag{33}$$

with variance $h\sigma^2$ and standard deviation $\sqrt{h\sigma^2}$.

- However, the returns on high-frequency financial assets are clearly not iid ... but it is still a good approximation.

ARCH / GARCH model estimation

- Since the model is no longer of the linear form we are used to, we cannot use OLS.
- We use another technique known as maximum likelihood.
- The method works by finding the most likely values of the parameters, given the actual data.
- More specifically, we construct a likelihood function and maximize it.

ARCH / GARCH model estimation

- The steps to be followed in the estimation of an ARCH or GARCH model are as follows:

[1] Specify the appropriate equations for the mean and variance, for example, an AR (1) - GARCH (1,1):

$$y_t = \alpha + \phi y_{t-1} + \mu_t, \qquad \mu_t \sim (0, \sigma^2) \tag{34}$$

$$\sigma_t^2 = \alpha_0 + \alpha_1 \mu_{t-1}^2 + \beta_1 \sigma_{t-1}^2 \tag{35}$$

[2] Specify the likelihood function to maximize:

$$L = -(T/2)log(2\pi) - (1/2)\sum_{t=1}^{T} log(\sigma_t^2) - (1/2)\sum_{t=1}^{T}(y_t - \alpha - \phi y_{t-1})/\sigma_t^2 \tag{36}$$

[3] The computer maximizes the function, and calculates the parameters and their standard errors, standard errors

Extensions to the basic GARCH model.

- The main problems of GARCH (p, q) models are:
  - Non-negativity constraints can be violated.
  - GARCH models cannot account for leverage effects.
- In this context, since the GARCH model was developed, a large number of extensions and variants have been proposed. Three of the most important examples are the GARCH-M, EGARCH, and GJR models.
- In fact, possible solutions to the two problems posed above can be addressed by the exponential GARCH model (EGARCH) or the GJR model, which propose asymmetric GARCH models.

GARCH - in Mean

- Based on the classical risk-hedging problem, if we expect a risk to be compensated by a higher return, why not let the return of a given security be partially determined by its risk?
- [?] suggested the ARCH-M specification:

$$Y_t = \mu + \delta\sigma_{t-1} + \mu_t, \qquad \mu_t \sim (0, \sigma^2) \tag{37}$$

$$\sigma_t^2 = \alpha_0 + \alpha_1\mu_{t-1}^2 + \beta\mu_{t-1}^2 \tag{38}$$

- δ can be interpreted as a kind of risk premium. It is possible to combine all or some of these models together to obtain more complex, hybrid models - for example, an ARMA-EGARCH (1,1)-M type model.

EGARCH Model

- Suggested by [**?**].The variance equation is given by:

$$\log(\sigma_t^2) = \omega + \beta log(\sigma_{t-1}^2) + \gamma \frac{\mu_{t-1}}{\sqrt{\sigma_{t-1}^2}} + \alpha \left[\frac{|\mu_{t-1}|}{\sqrt{\sigma_{t-1}^2}} - \sqrt{\frac{2}{\pi}} \right] \quad (39)$$

- **Advantages of the model.**
- Since we model $log(\sigma_t^2)$, even if the par´ameters are negative, σ_t^2 will be positive. We can take into account the leverage effect: if the relationship between volatility and return is negative, γ, it will be negative.

GJR Model

- Due to [?]:

$$\sigma_t^2 = \alpha_0 + \alpha_1 \mu_{t-1}^2 + \beta \sigma_{t-1}^2 + \gamma \mu_{t-1}^2 I_{t-1} \tag{40}$$

- Where:

$$I_{t-1} = 1 \text{ si } \mu_{t-1} < 0$$

$$I_{t-1} = 0 \text{ si } \mu_{t-1} \geq 0$$

- For a leverage effect: $\gamma > 0$.
- We require $\alpha_1 + \gamma \geq 0$ and $\alpha_1 \geq 0$ for nonnegativity.

Multivariate GARCH Models

- Multivariate GARCH models are used to estimate and forecast covariances and correlations. The basic formulation is similar to that of the GARCH model, but where variances, as well as covariances, are allowed to vary over time.
- There are 3 main classes of multivariate GARCH formulations, which are widely used: VECH, diagonal VECH and BEKK.
- Multivariate GARCH (MGARCH) models generalize univariate GARCH models and allow us to incorporate relationships between the volatility processes of several series. We want to know, for example, how changes in the volatility of one stock affect the volatility of another stock. These relationships can be parameterized in several ways.

GARCH vs average return volatility

R code

```
fit.garch <- garch(Returns, trace=FALSE)
print(fit.garch)
coeftest(fit.garch)
sigmaGarch<-fit.garch[[”coef”]][[”a0”]]/(1-fit.garch[[”coef”]][[”a1”]]-
fit.garch[[”coef”]][[”b1”]])
sigmaAvg<-var(Returns)
```

GARCH vs average return volatility

```
Call:
garch(x = Returns, trace = FALSE)

Coefficient(s):
       a0          a1          b1
4.076e-05  1.289e-01  7.866e-01

> coeftest(fit.garch)

z test of coefficients:

     Estimate Std. Error z value  Pr(>|z|)
a0 4.0755e-05 1.0208e-05  3.9924 6.542e-05 ***
a1 1.2895e-01 1.8933e-02  6.8105 9.727e-12 ***
b1 7.8655e-01 3.4052e-02 23.0988 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

sigmaAvg = 0.0004834654
sigmaGarch = 0.0004823009

GARCH vs average return volatility

R code

```
# Rolling windows
windowLength = 40
foreLength = length(Returns) - windowLength
sigmaAvgV <- vector(mode=”character”, length=foreLength)
sigmaGarchV <- vector(mode=”character”, length=foreLength)
```

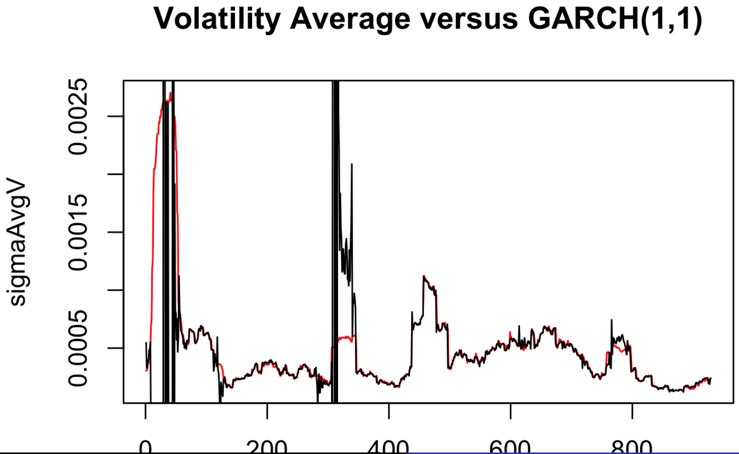
GARCH vs average return volatility

R code

```
for (d in 1:foreLength) {
ReturnsOffset = Returns[(d):(windowLength+d)]
fit.garch <- garch(ReturnsOffset, trace=FALSE)
sigmaGarch<-fit.garch[[”coef”]][[”a0”]]/(1-fit.garch[[”coef”]][[”a1”]]-
fit.garch[[”coef”]][[”b1”]])
sigmaAvg<-var(ReturnsOffset) print(sigmaGarch);
print(sigmaAvg)
sigmaGarchV[d]<-sigmaGarch
sigmaAvgV[d]<-sigmaAvg }
```

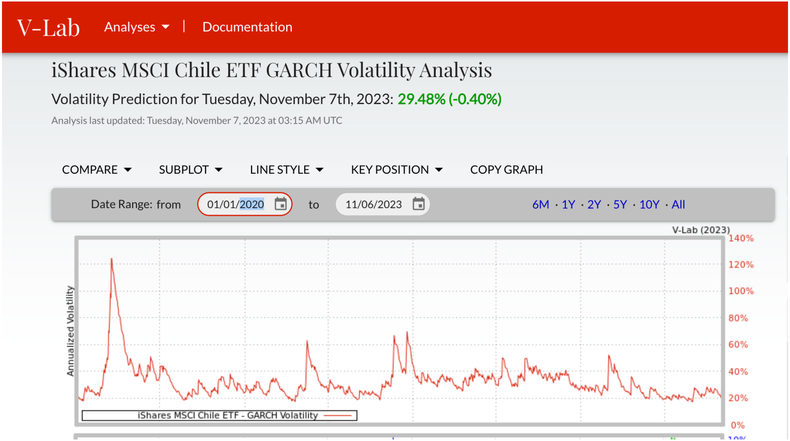
GARCH vs average return volatility

https://vlab.stern.nyu.edu/analysis/VOL.ECH:US-R.GARCH



GARCH vs average return volatility

https://vlab.stern.nyu.edu/volatility/VOL.ECH:US-R.GARCH



iShares MSCI Chile ETF GARCH Volatility Analysis

Volatility Prediction for Tuesday, November 7th, 2023: 29.48% (-0.40%)

Analysis last updated: Tuesday, November 7, 2023 at 03:15 AM UTC

Date Range: from 01/01/2020 to 11/06/2023

Homework VI

Choose a variable of your choice (e.g., exchange rate, interest rate, stock index), and using monthly data, analyze its volatility.

1. Tests for nonlinearity
2. Test the ARCH effect
3. Compare the GARCH with the Average return volatility
4. Compare different types of GARCH models
5. Comment on what has happened to the volatility of your variable in the last few years.

References

[1] John Y Campbell, Andrew W Lo, Archie Craig MacKinlay, et al. The econometrics of financial markets, volume 2. princeton University press Princeton, NJ, 1997.

[2] Robert F Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. Econometrica: Journal of the Econometric Society, pages 987–1007, 1982.

[3] Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. Journal of econometrics, 31(3):307–327, 1986.

References

[4] Robert F Engle, David M Lilien, and Russell P Robins. Estimating time varying risk premia in the term structure: The arch-m model. Econometrica: journal of the Econometric Society, pages 391–407, 1987.

[5] Daniel B Nelson. Conditional heteroskedasticity in asset returns: A new approach. Econometrica: Journal of the Econometric Society, pages 347–370, 1991.

[6] Lawrence R Glosten, Ravi Jagannathan, and David E Runkle. On the relation between the expected value and the volatility of the nominal excess return on stocks. The journal of finance, 48(5):1779–1801, 1993.

Lecture VII.- 8. Forecasting in the context of Machine learning

Marcelo Villena, PhD
Santa María University

November, 2023

Outline

1. Introductio to Machine learning
2. Introduction to Neural Networks
3. Feed forward neural networks
4. Support Vector Machines (SVM)
5. Homework VII
6. References

On Machine Learning (ML)

- Machine learning (ML) is a field of study in artificial intelligence concerned with the development and study of statistical algorithms that can effectively generalize and thus perform tasks without explicit instructions. Recently, generative artificial neural networks have been able to surpass many previous approaches in performance.
- ML is known in its application across business problems under the name predictive analytics. Although not all machine learning is statistically based, computational statistics is an important source of the field's methods.

On Machine Learning (ML)

- The mathematical foundations of ML are provided by mathematical optimization (mathematical programming) methods. Data mining is a related (parallel) field of study, focusing on exploratory data analysis through unsupervised learning.
- **Unsupervised learning** in artificial intelligence is a type of machine learning that learns from data without human supervision. Unlike supervised learning, unsupervised machine learning models are given unlabeled data and allowed to discover patterns and insights without any explicit guidance or instruction.

Introduction to Neural Networks

- A popular topic in modern data analysis is the neural network, which can be classified as a semiparametric method.The literature on neural networks is enormous, and their application extends to many scientific fields with varying degrees of success. [1] provide information on neural networks from a statistical point of view.
- First, we will focus on feed-forward neural networks in which inputs are connected to one or more neurons, or nodes, in the input layer, and these nodes are connected to other layers until they reach the output layer.

Introduction to Neural Networks

- Neural networks are a machine learning framework, which attempts to mimic the learning pattern of natural biological neural networks.
- Biological neural networks have interconnected neurons with dendrites that receive inputs, and then, based on these inputs, produce an output signal via an axon to another neuron.
- We attempt to mimic this process through the use of **Artificial Neural Networks (ANN)**, which from now on we will call neural networks. The process of creating a neural network begins with the most basic form, a single perceptron.

Introduction to Neural Networks

- Subsequently, we will introduce the **Support Vector Machine (SVM)**. In machine learning, SVM are supervised learning models using algorithms that allow to analyze the data used for classification and regression analysis.
- In particular, given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new data to one of the two categories, converting it into a non-probabilistic binary linear classifier.

Introduction to Neural Networks

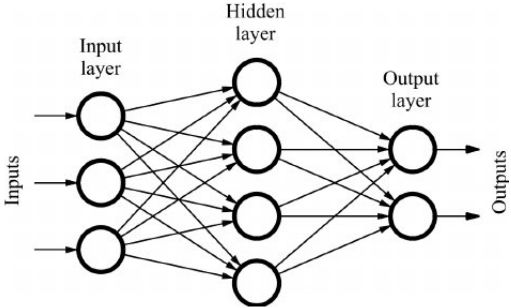
- An SVM model is a representation of the data as points in space, mapped in such a way that the examples of the categories are divided by as wide a free space as possible.
- Then, new data are mapped into that same space and are predicted to belong to a category according to the side of the space where they fall.

Feed forward neural networks

- We will develop an example of a simple feed-forward network for univariate time series analysis with a hidden layer.
- The input layer has two nodes, and the hidden layer has three. The input nodes connect forward to each and every node in the hidden layer, and these hidden nodes connect to the single node in the output layer. We call the network a feed-forward network.
- More complicated neural networks, including those with feedback connections, have proliferated, but feed-forward networks are the most relevant to our study.

Feed forward neural networks

- A feed-forward neural network with a hidden layer for univariate time series analysis.



Feed forward neural networks

- The perceptron receives inputs, multiplies them by some weight and then passes them to an activation function to produce an output. This is how, a neural network processes information from one layer to the next via an ”activation function”.

Feed forward neural networks

Consider a feed-forward network with a hidden layer. The jth node in the hidden layer is defined as:

$$h_j = f_j\left(\alpha_0 j + \sum_{i \to j} w_i j x_i\right) \tag{1}$$

where x_i s the value of the i-th input node, $f_j(.)$ is an activation function that is generally taken to be the logistic function:

$$f_j(z) = \frac{exp(z)}{1 + exp(z)'}$$

$\alpha_{0j}$is called bias, $i \to j$ means summing all input nodes feeding j, and $w_{ij}$are the weights.

Feed forward neural networks

For illustrative purposes, the j-th node of the hidden layer of the 2-3-1 forward network in the figure is:

$$h_j = \frac{exp(\alpha_{0j} + w_{1j}x_1 + w_{2j}x_2)}{1 + \alpha_0 j + w_{1j}x_1 + w_{2j}x_2)}, j = 1, 2, 3. \tag{2}$$

For the output layer, the node is defined as:

$$o = f_o\left(\alpha_{0o} + \sum_{j \to o} w_{jo} h_j\right) \tag{3}$$

where the activation function $f_o(.)$ is linear or a unit step function (Heaviside function). If $f_o(.)$is linear, then:

Feed forward neural networks

$$o = \alpha_{0o} + \sum_{j \to o} w_{jo} h_j$$

where k is the number of nodes in the hidden layer. By a unit step function, we mean $f_o(z) = 1$ if $z > 0$ and $f_o(z) = 0$ otherwise. A neuron with a unit step function is called a threshold neuron, with "1" indicating that the neuron sends its message. For example, the output of the 2-3-1 network in the figure is:

$$o = \alpha_{0o} + w_{1o} h_1 + w_{2o} h_2 + w_{3o} h_3$$

Feed forward neural networks

If the activation function is linear, we have:

$$o = \begin{cases} 1, & \text{if } \alpha_{0o} + w_{1o}h_1 + w_{2o}h_2 + w_{3o}h_3 > 0. \\ 0, & \text{if } \alpha_{0o} + w_{1o}h_1 + w_{2o}h_2 + w_{3o}h_3 \leq 0. \end{cases}$$

If $f_o(.)$ is a unit step function. By combining the layers, the output of a feed-forward neural network can be written as:

$$o = f_0 \left[\alpha_{0o} + \sum_{j \to o} w_{jo} f_j \left(\alpha_{0j} + \sum_{i \to j} w_{ij} x_i \right) \right] \quad (4)$$

Feed forward neural networks

If one also allows direct connections from the input layer to the output layer, then the network becomes:

$$o = f_0 \left[\alpha_{0o} + \sum_{i \to o} \alpha_{io} x_i + \sum_{j \to o} w_{jo} f_j \left(\alpha_{0j} + \sum_{i \to j} w_{ij} x_i \right) \right] \qquad (5)$$

the first sum is added to the input nodes. When the activation function of the output layer is linear, the direct connections from the input nodes to the output node represent a linear function between the inputs and the output.

Feed forward neural networks

Consequently, in this particular case, the model presented in eq. 5, is a generalization of linear models. For the 2-3-1 network in our figure, if the output activation function is linear, then equation 4 becomes:

$$o = \alpha_{0o} + \sum_{j=1}^{3} w_{jo} h_j$$

where h_j is obtained from the equation 2.

Feed forward neural networks

The network has 13 parameters. If we use the equation 5, the network becomes:

$$o = \alpha_{0o} + \sum_{i=1}^{2} \alpha_{io} x_i + \sum_{j=1}^{3} w_{jo} h_j$$

where h_j is obtained from the equation 2. The number of network parameters increases to 15.

Feed forward neural networks

- We refer to the function in equation 4 or 5 as a semiparametric function because its functional form is known, but the number of nodes and their offsets and weights are unknown.
- The direct connections from the input layer to the output layer in equation 5 means that the network can omit the hidden layer. We refer to such a network as a hop layer feed-forward network.

Feed forward neural networks

- Feed-forward networks are known as multilayer percentrons in the neural network literature. They can approximate any continuous function uniformly in compact sets by increasing the number of nodes in the hidden layer, see [2], [3].
- This property of neural networks is the universal approximation property of multilayer percetrons.
- **In summary, feed-forward neural networks with a hidden layer can be viewed as a way to parameterize a general continuous nonlinear function.**

Training and Forecasting

- The application of neural networks involves two steps.
  - The first step is to train the network (i.e., to construct a network, including the determination of the number of nodes and the estimation of their biases and weights).
  - The second step is inference, especially extra-sample forecasting. By comparing the comparative result of each forecast, the network that outperforms the others is selected and defined as the best network for making inferences.

Training and Forecasting

- In a time series application, suppose that $\{(r_t, x_t)|t = 1, ..., T\}$ represent the data available for the training of the network, where x_t denotes the vector of inputs, and r_t is the series of interest (e.g., log-returns of an asset).
- For a given network, suppose that o_t is the output of the network, with an input of x_t; see the model presented in equation 5.
- Training a neural network is equivalent to choosing its biases and weights so as to minimize some fit criteria, e.g., the least square of its error.

$$S^2 = \sum_{t=1}^{T} (r_t - o_t)^2$$

Training and Forecasting

- This is a nonlinear estimation problem that can be solved by several iterative methods. To guarantee the smoothness of the fitted function, some additional constraints can be added to the minimization problem above.
- In the neural network literature, the Back Propagation (BP) learning algorithm is the most popular method for training a network.
- The BP method, introduced by [4], works backward starting with the output layer, and uses a gradient rule to modify the biases and weights iteratively. Once a feed-forward neural network is constructed, it can be used to compute out-of-sample forecasts.

Training and Forecasting

- In short, once we have the output, we can compare it to a known series and adjust the weights as best we can (the weights usually start with random initialization values).
- We keep repeating this process until we have reached a maximum number of iterations allowed, or an acceptable error rate. To create a neural network, we simply start adding layers of perceptrons, creating a multilayer perceptron model of a neural network.
- We will have an input layer that directly takes function inputs and an output layer that creates the resulting outputs. The intermediate layers are known as hidden layers because they do not directly see the feature inputs or outputs.

Data pre-processing

- It is important to normalize the data before training a neural network.
- The neural network may have difficulty converging before the maximum number of iterations allowed if the data is not normalized.
- There are many different methods for data normalization. Generally, it is best to scale the data from 0 to 1, or from -1 to 1.

Example 1

- To illustrate the potential applications of neural networks in finance, we will model the monthly returns, including dividends, of the firm IBM from January 1926 to December 1999.
- We divide the data into two subsamples. The first subsample, consisting of returns from January 1926 to December 1997 for 864 observations, is used for modeling.

Example 1

- Using the model presented in equation 5, with three inputs and two nodes in the hidden layer, we obtain a 3-2-1 network for the series.
- The three entries are r_{t-1}, r_{t-2} y r_{t-3}, and the biases and weights are presented below:

$$r_t = 3.22 - 1.81 f_1(r_{t-1}) - 2.28 f_2(r_{t-1}) - 0.09 r_{t-1} - 0.05 r_{t-2} - 0.12 r_{t-3}$$

- where, $r_{t-1} = (r_{t-1}, r_{t-2}, r_{t-3})$

Example 1

and the two logistic functions are:

$$f_1(r_{t-1}) = \frac{exp(-8.34 - 18.97r_{t-1} + 2.17r_{t-2} - 19.17r_{t-3})}{1 + exp(-8.34 - 18.97r_{t-1} + 2.17r_{t-2} - 19.17r_{t-3})}$$

$$f_2(r_{t-1}) = \frac{exp(39.25 - 22.17r_{t-1} - 17.34r_{t-2} - 5.98r_{t-3})}{1 + exp(39.25 - 22.17r_{t-1} - 17.34r_{t-2} - 5.98r_{t-3})}$$

Example 1

- The standard error of the residuals for the above model is 6.56. For comparison, he also constructed an AR model for the data, resulting in the following model:

$$r_t = 1.101 + 0.077 r_{t-1} + a_t$$

  with $\sigma_a = 6.61$

- The residual standard error is slightly larger than that of the feed-forward model.

Example 1

- The monthly IBM stock returns in 1998 and 1999 form the second subsample and are used to evaluate the out-of-sample prediction performance of the neural networks.
- As a benchmark for comparison, we use the sample mean of rt in the first subsample as the 1-step forecast for all monthly returns in the second subsample. This is equivalent to assuming that the monthly IBM stock price follows a random walk with drift.
- The mean square forecast error (MSE) of the benchmark model is 91.85. For the AR (1) model, the MSE of the 1-step ahead forecasts is 91.70. Therefore, the AR (1) model slightly outperforms the benchmark. For the network, the MSE is 91.74.

Example 2 - Forecasting the SP 500

In this example we will forecast the SP 500 using univariate models. In particular, we will use a linear model, such as ARIMA, and a non-linear one, a feed-forward neural network.

R code

```
library(neuralnet)
library(nnet)
library(forecast)
getSymbols(“SPY”, from = “2000-01-01”, to = “2017-12-01”, src =
“yahoo”, adjust = TRUE, periodicity = “monthly”)
Returns = diff(log(Ad(SPY)))
Returns[as.character(head(index(Ad(SPY)),1))] = 0
```

Example 2 - Forecasting the SP 500

- **nnetar (Neural Network Time Series Forecast)** is a feed-forward neural network that considers lagged y-values as inputs, and a single hidden layer.
- The inputs are for lags from 1 to p. Several networks are fitted, each with random initial weights. The results are then averaged when calculating the predictions.

Example 2 - Forecasting the SP 500

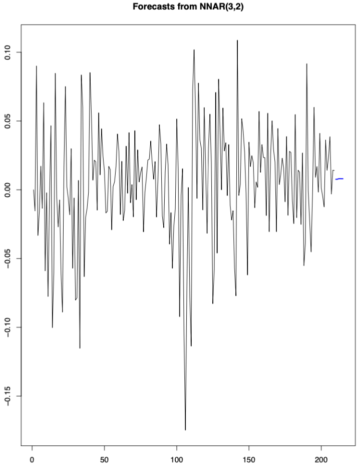
- The network is designed for single-step forecasting. The multi-step forecasts are computed recursively. For non-seasonal data, the fitted model is denoted as a NNAR (p, k) model, where k is the number of hidden nodes.
- **This is analogous to an AR (p) model but with nonlinear functions. For seasonal data, the fitted model is called a NNAR (p, P, k) [m] model, which is analogous to an ARIMA (p, 0,0) (P, 0,0) [m] model but with nonlinear functions.**

Example 2 - Forecasting the SP 500

R code

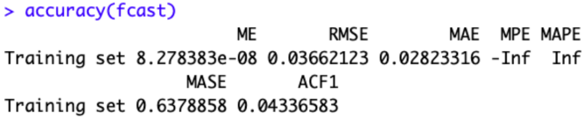
```
train=Returns[1:209]
test=Returns[210:215]
nn < − nnetar(train)
fcast < − forecast(nn, h=length(test))
autoplot(fcast)
plot(fcast)
fcast
test
accuracy(fcast)
arimaModel < − auto.arima(train)
accuracy(arimaModel)
```

Example 2 - Forecasting the SP 500

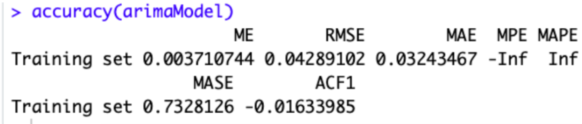


Example 2 - Forecasting the SP 500

Clearly for the period 2000-2017 the autoregressive neural network wins, considering the nonlinearities present.



```
> accuracy(fcast)
                       ME       RMSE        MAE  MPE MAPE
Training set 8.278383e-08 0.03662123 0.02823316 -Inf  Inf
                  MASE       ACF1
Training set 0.6378858 0.04336583
```



```
> accuracy(arimaModel)
                      ME       RMSE        MAE  MPE MAPE
Training set 0.003710744 0.04289102 0.03243467 -Inf  Inf
                 MASE        ACF1
Training set 0.7328126 -0.01633985
```

Support Vector Machines (SVM)

- They were created by Boser, Guyon and Vapnik in 1992 [5].
- The original formulation is motivated by the resolution of classification problems, where the basic idea consists of mapping the data from the original space to a higher dimensional space through a nonlinear transformation chosen a priori, and then constructing the optimal separation hyperplane in the new space.
- In this way, by solving a linear problem in the new space, we have a nonlinear model in the original space.
- Based on the same philosophy, the method was later extended to regression and clustering problems. Since its creation, SVM has attracted great theoretical attention, being applied with great success to practical time series prediction problems of different nature.

Support Vector Machines (SVM)

- Support Vector Regression (SVR) works on similar principles as Support Vector Machine (SVM) classification.
- **One can say that SVR is the adapted form of SVM when the dependent variable is numerical rather than categorical.**
- A major benefit of using SVR is that it is a non-parametric technique. Unlike OLS, whose results depend on Gauss-Markov assumptions, the output model from SVR does not depend on distributions of the underlying dependent and independent variables. Instead the SVR technique depends on **kernel functions**.
- Another advantage of SVR is that it permits for construction of a non-linear model without changing the explanatory variables, helping in better interpretation of the resultant

Support Vector Machines (SVM)

- The basic idea behind SVR is not to care about the prediction as long as the error (ϵi) is less than certain value. **This is known as the principle of maximal margin.**
- This idea of maximal margin allows viewing SVR as a convex optimization problem.
- The regression can also be penalized using a cost parameter, which becomes handy to avoid over-fit. SVR is a useful technique provides the user with high flexibility in terms of distribution of underlying variables, relationship between independent and dependent variables and the control on the penalty term.

Support Vector Machines (SVM)

- Among the main features of SVM are:
  1. The possibility of solving a convex problem, without entrapment in local optimums,
  2. The representation of the solution based on a fraction of the total available points (these points are called Support Vectors),
  3. The ability to generalize to new data, because the SVM algorithm is based on the principle of minimization of structural risk proposed in Vapnik's Statistical Learning Theory, and
  4. The ability to model nonlinear phenomena by means of the aforementioned transformation of the data from the original space to a higher dimensional space, a space in which a linear model is obtained that is equivalent to a linear model in the original space.

Support Vector Machines (SVM)

- A kernel function is usually used to refer to the **kernel trick**, a method of using a linear classifier to solve a nonlinear problem. It entails transforming linearly inseparable data to linearly separable ones.
- Machine learning methods are widely applied to classification and regression problems. T**he best-known kernel method for regression is support vector regression (SVR)**, which is based on the principles of statistical learning theory (Cortes and Vapnik 1995).
- **Kernel methods convert linear algorithms for use on nonlinear data by projecting the input data into a high dimensional feature space in which a linear solution is found.**

Support Vector Machines (SVM)

- **Model definitions - Kernel functions**
- A kernel is defined as a function. K, such that $\forall x, y \varepsilon K$

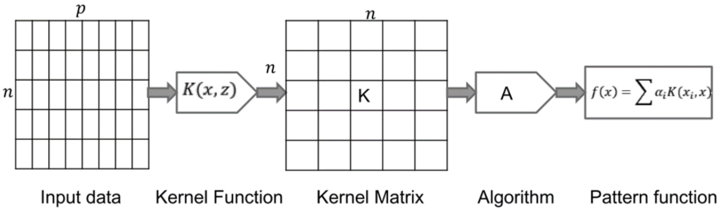
$$K(x,z) =< \Phi(x)\Phi(z) >$$

- where X is the space of the input data (finite, generally R^n); and Φ is a mapping function of the input data from X to a higher dimensional space F, where $< \bullet, \bullet >$ is the inner product of F. It can be proved that $K(x,z)$ is a kernel function if and only if the matrix $M = (K(x_i, x_j))_{i,j=1}^n$ is positive semidefinite. Some of the most common kernels are:
- Linear: $K(x, x\prime) =< x, x\prime >$
- Polinomial: $K(x, x\prime) = (< x, x\prime > +1)^d$
- RBF: $K(x, x\prime) = exp(-||x - x\prime||^2/\sigma^2)$

Support Vector Machines (SVM)

- **Model Definitions - SVM Structure**
- The SVM model can be viewed as layers of nodes, where:
    - The first layer consists of n nodes, which correspond to the input vector.
    - The second layer consists of N nodes, which is the nonlinear transformation based on support vectors.
    - The third layer contains only 1 node, which is the prediction
    - Each layer is fully connected to the next one.
    - The nodes arriving at the output node are weighted by constants, which are to be determined by the model, and then summed.

Support Vector Machines (SVM)



Support Vector Machines (SVM)

- **Model Definitions - SVM Structure**
- During the learning process, the first layer selects the bases $K(x_i; X), i = 1, ..., N$; within the set of possible bases, while the second layer constructs a linear function in the new space, which is equivalent to finding a non-linear model in the input space.
- The N selected bases are those induced by the points called Support Vectors.

Support Vector Machines (SVM)

- **Model definitions - Loss functions.** The model sought is of the form $y = f(x) + e$, where $f(x)$ is a nonlinear function and e the error. Then, one wishes to minimize the value of $y_i - f(x_i) = e$; for each i, and for this a p-loss function is used. The most common ones are:
  - Cuadrática:
  $$L(f(x), y) = (f(x) - y)^2$$
  - $\epsilon - sensible$:
  $$L(f(x), y, \epsilon) = \begin{cases} 0 & \text{si } |f(x) - y| < \epsilon \\ |f(x) - y| < \epsilon & \text{si } no \end{cases}$$
  - Huber:
  $$L(f(x), y, \epsilon) = \begin{cases} \frac{1}{2}(f(x) - y)^2 & \text{si } |f(x) - y| < \mu \\ \mu|f(x) - y| - \frac{\mu^2}{2} & \text{si } no \end{cases}$$

Support Vector Machines (SVM)

SVM Regression Algorithm

The optimization problem that finds the model weights, using p εsensitive loss function is:

$$\begin{aligned} min \quad & \frac{1}{2}\|w\|^2 \\ s.a. \quad & y_i - < w, \Phi(x_i) > -b \leq \epsilon \\ & -y_i + < w, \Phi(x_i) > +b \geq \epsilon \end{aligned}$$

Support Vector Machines (SVM)

SVM Regression Algorithm

Since there may be no solution to the above problem, it is usually reformulated as:

$$min \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\zeta_i + \zeta_i^*)$$

$$s.a. \quad y_i - < w, \Phi(x_i) > -b \leq \epsilon + \zeta_i$$

$$-y_i + < w, \Phi(x_i) > +b \geq \epsilon + \zeta_i$$

$$\zeta_i, \zeta_i^* \geq 0, i = 1, 2, \ldots l$$

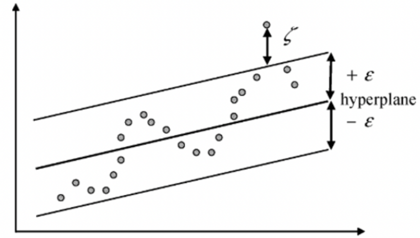
Support Vector Machines (SVM)

- The problem is that there may be no solution, so it is reformulated as: where C is a parameter pair to be fixed, representing the trade-off between model complexity and accuracy, and the parameter pair ε represents the range of tolerance to errors in the model. This problem has a solution, and it is also convex, so the optimization methods converge well to the solution, and the dual approach is much simpler than the primal problem. Once the weights w are found, then our models are:

$$y = \sum_{i=1}^{N} w_i K(X_i, x)$$

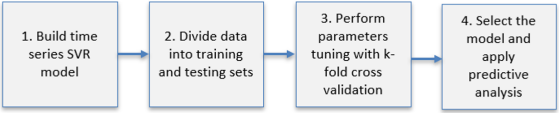
Support Vector Machines (SVM)

Thus, for example, SVR tries to find a function f(x) where the predicted values are at most $\in$ from the observed values yi, fitting inside a tube of width 2$\in$.



The Modeling and Solving Approach

- The first part of the experiment is to train the dataset. The data must be divided into training (80%) and testing (20%) data.
- To train the data, the package called **caret** (short for classification and regression training) has been used. The library provides a set of functions that attempt to streamline the process for creating predictive models..



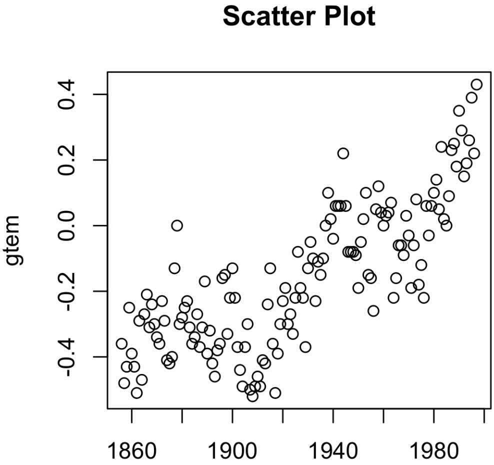
1. Build time series SVR model
2. Divide data into training and testing sets
3. Perform parameters tuning with k-fold cross validation
4. Select the model and apply predictive analysis

Example 3 - SVR

R code

```
#Read Data
data=read.csv(”gtemp.csv”, header=T)
head(data)
Y<-data$gtem X<-data$time
#Scatter
Plot plot(data, main =”Scatter Plot”)
```

Example 3 - SVR

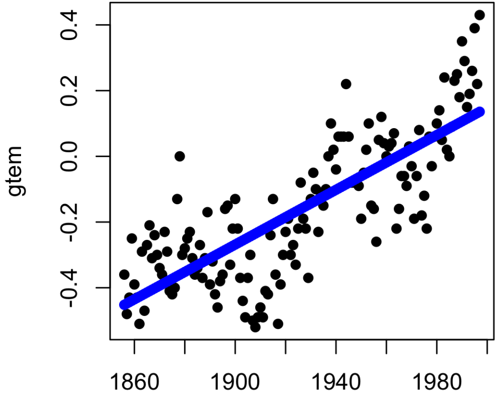


Example 3 - SVR

R code

```
#Linear Model
model=lm(Y˜X,data) abline(model)
#Scatter Plot
plot(data, pch=16)
#Predict Y using Linear Model
predY <- predict(model, data)
#Overlay Predictions on Scatter Plot
points(X, predY, col = ”blue”, pch=16)
## Calulate Root Mean Square Error (RMSE)
RMSE<-sqrt(mean((Y - predY)^2))
RMSE
```

Example 3 - SVR

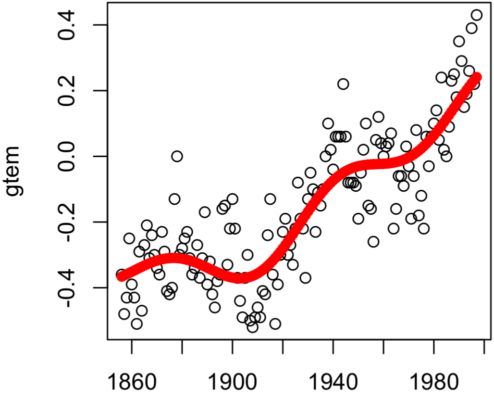


Example 3 - SVR

R code

```
library(e1071)
#Scatter Plot
plot(data)
#Regression with SVM
modelsvm=svm(Y˜X,data)
#Predict using SVM regression
predYsvm <- predict(modelsvm, data)
##Overlay SVM Predictions on Scatter Plot
points(X, predYsvm, col = ”red”, pch=16)
```

Example 3 - SVR



Example 3 - SVR

R code

```
## Calculate parameters of the SVR Model
#Find value of W
W=t(modelsvm$coefs) %*% modelsvm$SV
#Find value of b
b=modelsvm$rho
## RMSE for SVR Model
#Calculate RMSE
RMSEsvm<-sqrt(mean((Y- predYsvm)^2)) RMSEsvm
```

Perform parameters tuning with k-fold cross validation

- To determine the values of the tuning parameters, one approach is to use a resampling to estimate how well the model performs on the training set.
- There are different types of resampling methods being k-fold cross-validation one of the most common types. The process must be repeated many times and the performance estimates from each holdout set are averaged into a final overall estimate of model efficacy such that given the training set, the algorithm produces a prediction function f(x)=ϕ(xi).
- For each parameter combination the model fitness is estimated via resampling and the relationship between the tuning parameters and the model performance is evaluated.

Perform parameters tuning with k-fold cross validation

- The selected model is the one with the best average performance across the k folds. The procedure prevents **overfitting** to a subset of the training data.
- The caret package is used to perform the cross validation with a radial basis function kernel applied for parameter tuning.
- There are two tuning parameters: the radial basis function scale parameter bandwidth, and the cost value associated with support vectors.

On overfitting

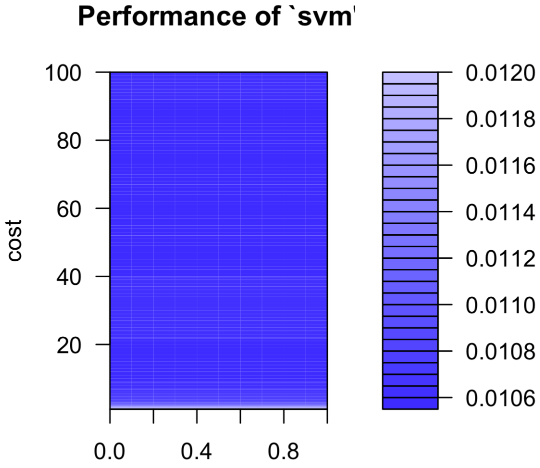
- In mathematical modeling, overfitting is **"the production of an analysis that corresponds too closely or exactly to a particular set of data, and may therefore fail to fit to additional data or predict future observations reliably".**
- An overfitted model is a mathematical model that contains more parameters than can be justified by the data. In a mathematical sense, these parameters represent the degree of a polynomial.
- The essence of overfitting is to have unknowingly extracted some of the residual variation (i.e., the noise) as if that variation represented the underlying model structure.

Example 3 - SVR

R code

```
## Optimising SVR Model and Selecting Best Model
#Tune the above SVM model
OptModelsvm=tune(svm,
Y˜X,data=data,ranges=list(elsilon=seq(0,1,0.1), cost=1:100))
print(OptModelsvm)
plot(OptModelsvm)
#Find out the best model
BstModel=OptModelsvm$best.model
#Predict Y using best model
PredYBst=predict(BstModel,data)
#Calculate RMSE of the best model
RMSEBst<-sqrt(mean((Y- PredYBst)ˆ2))
RMSEBst
```

Example 3 - SVR

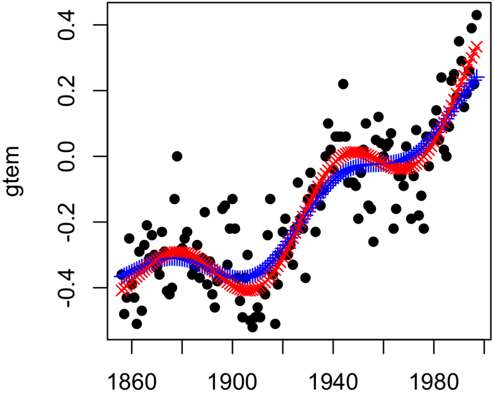


Example 3 - SVR

R code

```
## Plotting SVR Model and Tuned Model in same plot
plot(data, pch=16)
points(X, predYsvm, col = ”blue”, pch=3)
points(X, PredYBst, col = ”red”, pch=4)
points(X, predYsvm, col = ”blue”, pch=3, type=”l”)
points(X, PredYBst, col = ”red”, pch=4, type=”l”)
```

Example 3 - SVR



Homework VII

- Test the predictive power (out-of-sample) of machine learning models
    - Compare ANNs with univariate models seen previously for a previous homework, e.g. forecasting commodity price.
    - Compare SVRs with ARDL and VAR models seen previously for a previous homework, e.g. CAPM, Phillips Curve, Solow, etc.

References

[1] Bing Cheng and D Michael Titterington. Neural networks: A review from a statistical perspective. Statistical science, pages 2–30, 1994.

[2] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. Neural networks, 2(5):359–366, 1989.

[3] Tianping Chen and Hong Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. IEEE Transactions on Neural Networks, 6(4):911–917, 1995.

References

[4] AE Bryson Jr and YC Ho. Applied optimal control, blaisdell.(ristampa, 1975, hemisphere publishing, washington, dc). 1969.

[5] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In Proceedings of the fifth annual workshop on Computational learning theory, pages 144–152. ACM, 1992.

[6] Burges, C. J. 1998. "A Tutorial on Support Vector Machines for Pattern Recognition." Data Mining and Knowledge Discovery 2(2): 121–67.

[7] Cortes, C., and V. Vapnik. 1995. "Support-Vector Networks." Machine Learning 20(3): 273–97.