A Short Introduction to Time Series Analysis and Forecasting

L. Torgo

ltorgo@fc.up.pt
Faculdade de Ciências / LIAAD-INESC TEC, LA
Universidade do Porto

Oct, 2016

Time Series Data
A Definition

Definition

- A time series is a set of observations of a variable that are ordered by time.
- E.g.,
  $x_1, x_2, \cdots, x_{t-1}, x_t, x_{t+1}, \cdots, x_n$
  where x_t is the observation of variable X at time t.
- A multivariate time series is a set of observations of a set of variables over a certain period of time.

The Main Goals of Time Series Analysis

Explanation

Obtaining a Time Series Model help us to have
a Deeper Understanding of the Mechanism
that Generated the Observed Time Series Data.

Forecasting

- Given: $x_1, x_2, \cdots, x_{t-1}, x_t$ *The Past!*
- Obtain: a time series model
- Which is able to make predictions concerning:
  $x_{t+1}, \cdots, x_n$ *The Future!*

Other Goals

Time Series Data Mining

Main Time Series Data Mining Tasks

- *Indexing (Query by Content)*
  Given a query time series Q and a similarity measure $D(Q, X)$ find the most similar time series in a database **D**
- *Clustering*
  Find the natural goupings of a set of time series in a database **D** using some similarity measure $D(Q, X)$
- *Classification*
  Given an unlabelled time series Q, assign it a label C from a set of pre-defined labels (classes)

FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Time Series Data in R

- R has several data structures capable of handling time series data
- In our illustration we will use the infra-structure provided by package `xts`

```
library(xts)
data(ice.river,package='tseries')
ice.river[1:4,]

##      flow.vat flow.jok prec temp
## [1,]     16.1     30.2  8.1  0.9
## [2,]     19.2     29.0  4.4  1.6
## [3,]     14.5     28.4  7.0  0.1
## [4,]     11.0     27.8  0.0  0.6

ir <- xts(ice.river[,1],
          seq.Date(as.Date('1972-01-01'),
                   by='day',
                   len=nrow(ice.river)))
```


Time Series Data in R - indexing examples

```
ir[1:3]

##            [,1]
## 1972-01-01 16.1
## 1972-01-02 19.2
## 1972-01-03 14.5

ir['1973-05-02']

##            [,1]
## 1973-05-02 11.6

head(ir['1972-01'],2)

##            [,1]
## 1972-01-01 16.1
## 1972-01-02 19.2

head(ir['1972-01-23/1972-02-02'],2)

##            [,1]
## 1972-01-23  6.9
## 1972-01-24  6.9
```

```
head(ir['/1972-01-10'],2)

##            [,1]
## 1972-01-01 16.1
## 1972-01-02 19.2

head(ir['1974-12-21/'],2)

##            [,1]
## 1974-12-21 4.65
## 1974-12-22 5.16

head(ir['1972-01-23/1972-02'],2)

##            [,1]
## 1972-01-23  6.9
## 1972-01-24  6.9
```

Summaries of Time Series Data

- Standard descriptive statistics (mean, standard deviation, etc.) do not allways work with time series (TS) data.
- TS may contain trends, seasonality and some other systematic components, making these stats misleading.
- So, for proving summaries of TS data we will be interested in concepts like trend, seasonality and correlation between sucessive observations of the TS.

FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO

Types of Variation

Seasonal Variation

Some time series exhibit a variation that is annual in period, e.g. demand for ice cream.

Other Cyclic Variation

Some time series have periodic variations that are not related to seasons but to other factors, e.g. some economic time series.

Trends

A trend is a long-term change in the mean level of the time series.

FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO

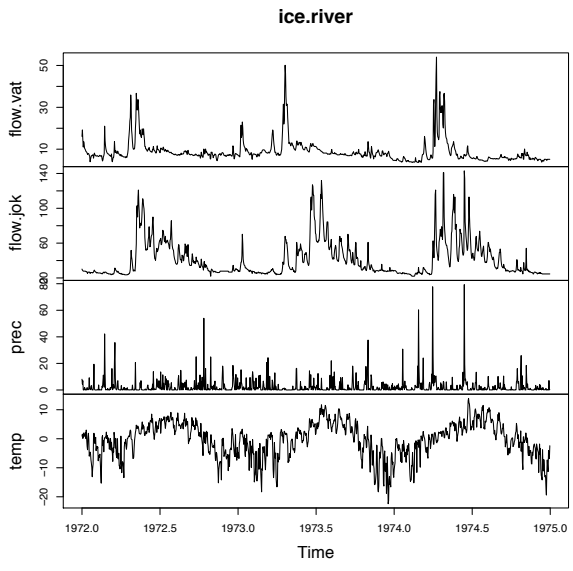
Stationarity

An Informal Definition

A time series is said to be stationary if
- there is no systematic change in mean (no trend),
- if there is no systematic change in variance and
- if strictly periodic variations have been removed.

Note that in these cases statistics like mean, standard deviation, variance, etc., bring relevant information!

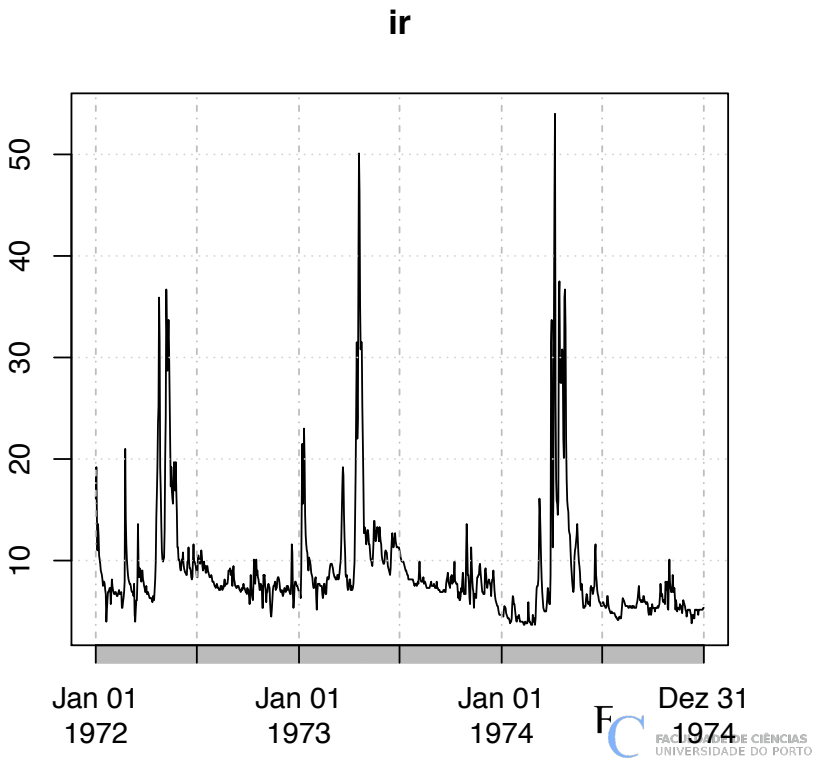
Time Plots



- Ploting the time series values against time is one of the most important tools for analysing its behaviour.
- Time plots show important features like trends, seasonality, outliers and discontinuities.

Time Plots in R

```
plot(ir)
```



Transformations - I

Plotting the data may suggest transformations :

To stabilize the variance

Symptoms: trend with the variance increasing with the mean.
Solution: logarithmic transformation.

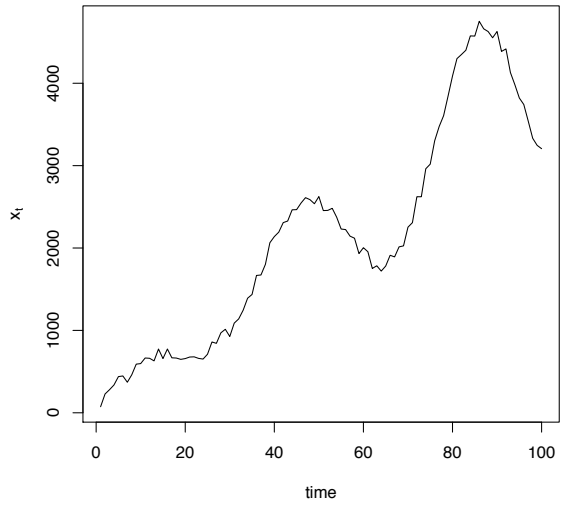
To make the seasonal effects additive

Symptoms: there is a trend and the size of the seasonal effect increases with the mean(multiplicative seasonality).
Solution: logarithmic transformation.

To remove trend

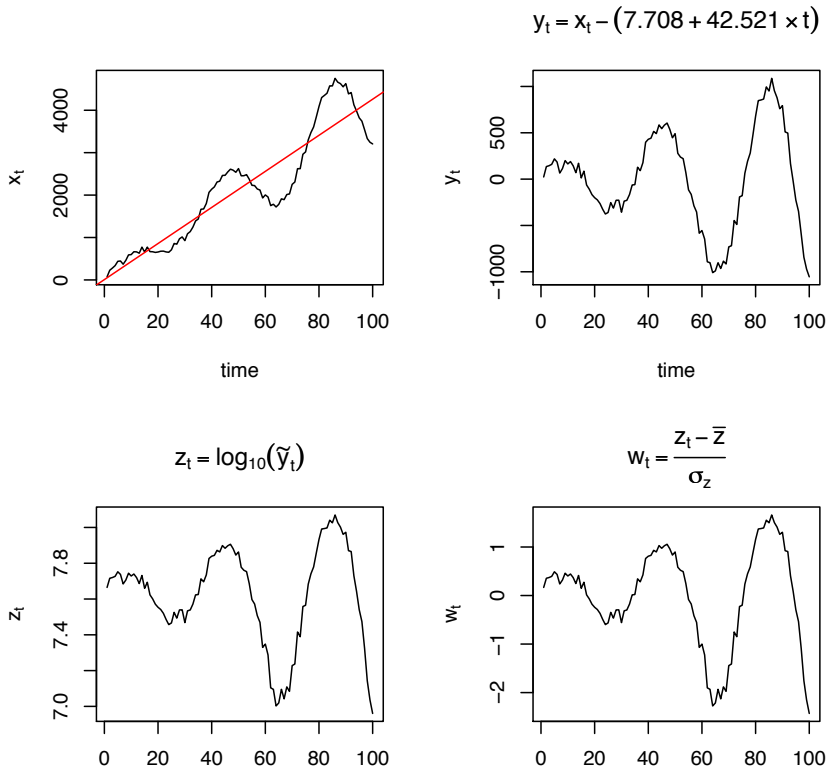
Symptoms: there is systematic change on the mean.
Solution 1: first order differentiation ($\nabla X_t = X_t - X_{t-1}$).
Solution 2: model the trend and subtractit from the original series ($Y_t = X_t - r_t$).

Transformations - a simple example (1)



An example time series with trend and a multiplicative seasonality effect.

Transformations - a simple example (2)



Some useful functions in R

```
(s <- ir[1:10])
```

```
##              [,1]
## 1972-01-01 16.10
## 1972-01-02 19.20
## 1972-01-03 14.50
## 1972-01-04 11.00
## 1972-01-05 13.60
## 1972-01-06 12.50
## 1972-01-07 10.50
## 1972-01-08 10.10
## 1972-01-09  9.68
## 1972-01-10  9.02
```

```
diff(s)
```

```
##              [,1]
## 1972-01-01    NA
## 1972-01-02  3.10
## 1972-01-03 -4.70
## 1972-01-04 -3.50
## 1972-01-05  2.60
## 1972-01-06 -1.10
## 1972-01-07 -2.00
## 1972-01-08 -0.40
## 1972-01-09 -0.42
## 1972-01-10 -0.66
```

```
diff(s,diff=2)
```

```
##              [,1]
## 1972-01-01    NA
## 1972-01-02    NA
## 1972-01-03 -7.80
## 1972-01-04  1.20
## 1972-01-05  6.10
## 1972-01-06 -3.70
## 1972-01-07 -0.90
## 1972-01-08  1.60
## 1972-01-09 -0.02
## 1972-01-10 -0.24
```

```
log10(s)
```

```
##                 [,1]
## 1972-01-01 1.2068259
## 1972-01-02 1.2833012
## 1972-01-03 1.1613680
## 1972-01-04 1.0413927
## 1972-01-05 1.1335389
## 1972-01-06 1.0969100
## 1972-01-07 1.0211893
## 1972-01-08 1.0043214
## 1972-01-09 0.9858754
## 1972-01-10 0.9552065
```


Autocorrelation

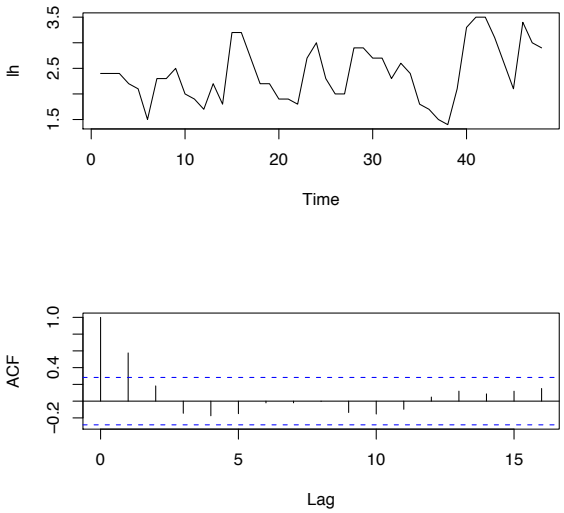
Sample Autocorrelation Coefficients

They measure the correlation between observations different distances apart.

$$r_k = \frac{\sum_{t=1}^{N-k}(x_t - \bar{x})(x_{t+k} - \bar{x})}{\sum_{t=1}^{N}(x_t - \bar{x})^2}$$

FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO

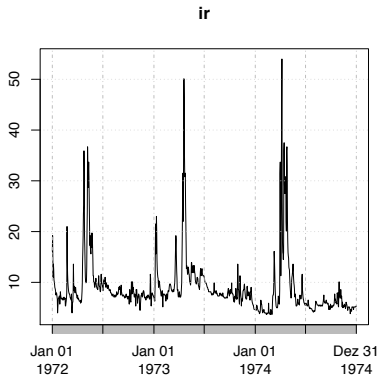
Correlogram

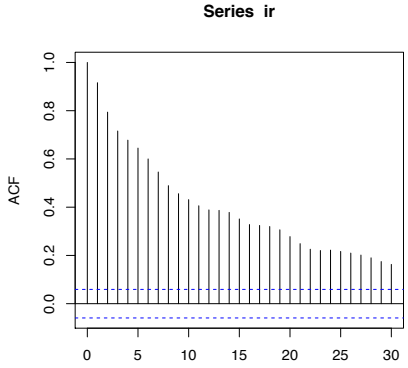


Plot the sample autocorrelation coefficients against the lags, $k = 0, 1, \cdots, M$.

Time Plots in R

```
plot(ir)
acf(ir)
```





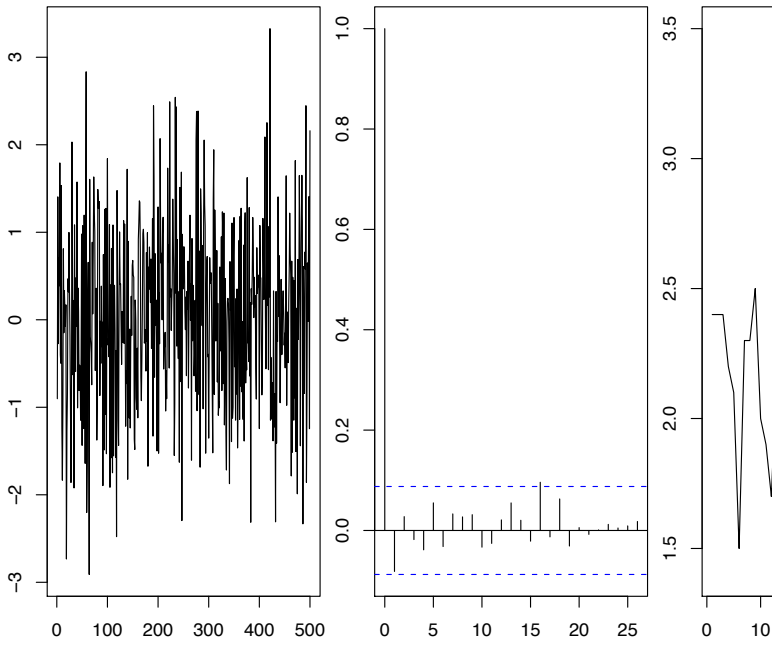
Interpreting the Correlogram

Random Series

Most r_k's near 0. Still, it is possible that 1 on 20 is significant...

Short-Term Correlation

Fairly large value of r_1 with sucessive values rapidly tending to non-significant.



Interpreting the Correlogram (cont.)

Alternating Series

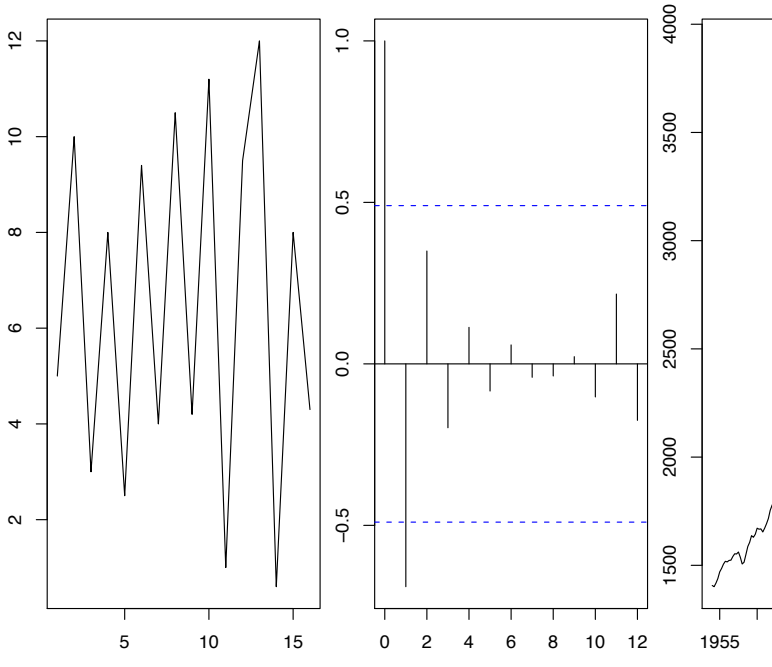
Similar pattern on the values of r_k.

Non-Stationary Series

For series with a trend the values of r_k will not go down till very large values of the lag.

Seasonal Series

The correlagram tends to exhibit the same periodicity as the original series.



Handling Real World Data

A Check List of Common Sense Things to Do (taken from Chatfield, 2004)

- Do you understand the context? Have the right variables been measured?
- Have all the time series been plotted?
- Are there missing values? If so, what should be done about them?
- Are there any outliers? If so, what should be done about them?
- Are there any discontinuities? If so, what do they mean?
- Does it make sense to transform the variables?
- Is trend present? If so, what should be done about it?
- Is seasonality present? If so, what should be done about it?

Chatfield, C. (2004): The Analysis of Time Series - an introduction. CRC.

Time Series Forecasting

- Given:
  $x_1, x_2, \cdots, x_{t-1}, x_t$ *The Past!*
- Obtain:
  a time series model
- Which is able to make predictions concerning:
  $x_{t+1}, \cdots, x_n$ *The Future!*

Goals of an Evaluation Method

- The golden rule:

  The data used for evaluating (or comparing) any models cannot be seen during model development.

- The goal of any evaluation procedure:
  - Obtain a reliable estimate of some evaluation measure.
    High probability of achieving the same score on other samples of the same population.
- Evaluation Measures
  - Predictive accuracy.
  - Model size.
  - Computational complexity.

FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO

Obtaining Reliable Estimates

- The usual techniques for model evaluation revolve around resampling.
  - Simulating the reality.
    - Obtain an evaluation estimate for unseen data.
- Examples of Resampling-based Methods
  - Holdout.
  - Cross-validation.
  - Bootstrap.

Time Series Data Are Special!

Any form of resampling changes the natural order of the data!

FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO

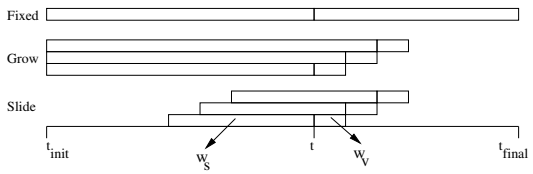
Correct Evalution of Time Series Models

- General Guidelines
  - Do not “forget” the time tags of the observations.
  - Do not evaluate a model on past data.

- A possible method
  - Divide the existing data in two time windows
    - Past data (observations till a time t).
    - “Future” data (observations after t).
  - Use one of these three learn-test alternatives
    - Fixed learning window.
    - Growing window.
    - Sliding window.

FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO

Learn-Test Strategies



Fixed Window

A single model is obtained with the available “training” data, and applied to all test period.

Growing Window

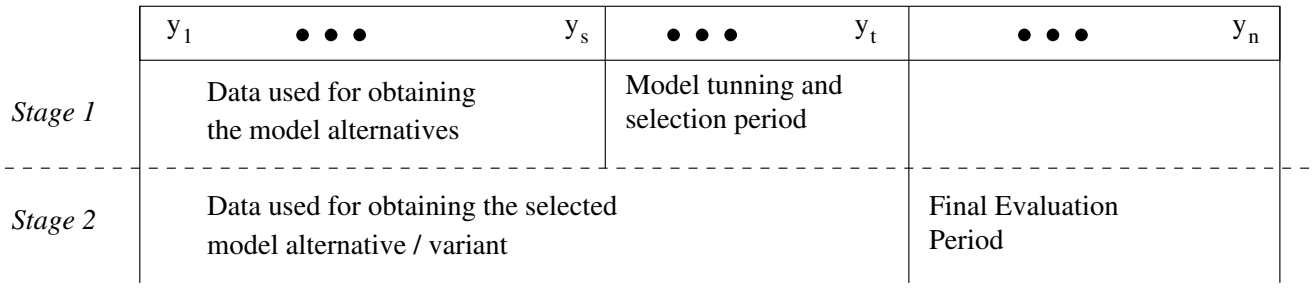
Every w_v test cases a new model is obtained using all data available till then.

Sliding Window

Every w_v test cases a new model is obtained using the previous w_s observations of the time series.

Dealing with model selection

- Most modelling techniques involve some form of parameters that usually need to be tunned.
- The following describes an evaluation methodology considering this issue:



Some Metrics for Evaluating Predictive Performance

Absolute Measures

- Mean Squared Error (MSE)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{x}_i - x_i)^2$$

- Mean Absolute Deviation (MAD)

$$MAD = \frac{1}{n}\sum_{i=1}^{n}|\hat{x}_i - x_i|$$

Relative Measures

- Theil Coefficient

$$U = \frac{\sqrt{\sum_{i=1}^{n}(\hat{x}_i - x_i)^2}}{\sqrt{\sum_{i=1}^{n}(x_i - x_{i-1})^2}}$$

- Mean Absolute Percentage Error (MAPE)

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{(\hat{x}_i - x_i)}{x_i}\right|$$

The Goal of an Experimental Comparison

- Given a set of observations of a time series X.
- Given a set of alternative modelling approaches M.
- Obtain estimates of the predictive performance of each m_i for this time series.

More specifically,
given a forecasting period size, w_{test},
and a predictive performance statistic, Err,
we want to obtain a reliable estimate of the value of Err
for each m_i.

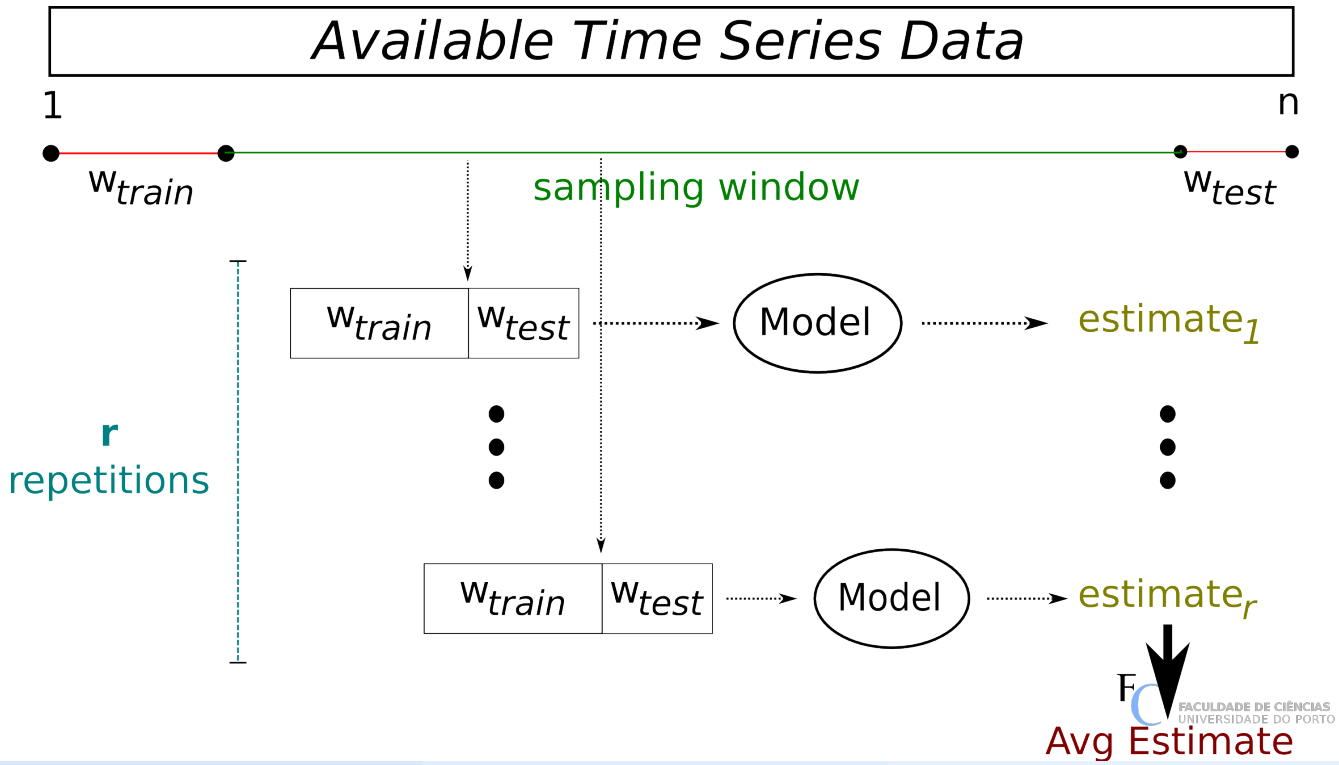
Using Monte Carlo Simulations for Obtaining Reliable Estimates of *Err*

- A possible approach would be to use our proposed method of Model Selection.
- This would give us one estimate of Err.
- More reliability is achievable if more repetitions of the process are carried out.

Monte Carlo Estimates for Time Series Forecasting

Given: a time series, a training window size, w_{train}, a testing window size, w_{test}, and a number of repetitions, r,
- randomly generate r points in the interval $]w_{train}..(n - w_{test})[$,
- for each point proceed according to our Model Selection strategy.

Using Monte Carlo Simulations for Obtaining Reliable Estimates of *Err*



Assumptions of "Classical" Linear Approaches

- *Linearity*
  The model of the time series behaviour is linear on its inputs.
- *Stationarity*
  The underlying equations governing the behaviour of the system do not change with time.

Most "classical" approaches assume stationary time series, thus one usually needs to transform non-stationary time series into stationary ones before using these tools.

Integrated ARMA (or ARIMA) Models

Definition

An integrated ARMA (or ARIMA) model of order p, d, q is a series given by

$\hat{Y_{t+1}} = \mu_Y + \sum_{i=0}^{p} \alpha_i Y_{t-i} + \sum_{m=0}^{d} \beta_i (Y_{t-m} - Y_{t-m-1}) + \sum_{k=0}^{q} \theta_k e(t-k)$

where $e(t-k) = \hat{Y_{t-k}} - Y_{t-k}$

There are many other variants, e.g. including seasonal components.

FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO

ARIMA models in R

The package **forecast**

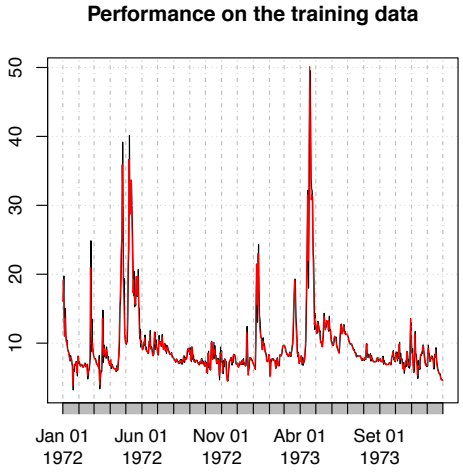
```
library(forecast)
train <- ir['/1973-12-31']
test <- ir['1974-01-01/']
model <- Arima(train,order=c(3,1,1))
model
```

```
## Series: train
## ARIMA(3,1,1)
##
## Coefficients:
##          ar1      ar2     ar3      ma1
##       0.6306  -0.3610  0.1538  -0.3196
## s.e.  0.2010   0.0706  0.0568   0.2003
##
## sigma^2 estimated as 3.059:  log likelihood=-1442.06
## AIC=2894.12   AICc=2894.2   BIC=2917.09
```

FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO

ARIMA models in R

```
pred.train <- fitted(model)
## plotting this
plot(as.xts(as.numeric(pred.train),index(train)),
     main="Performance on the training data")
lines(train,col="red")
```

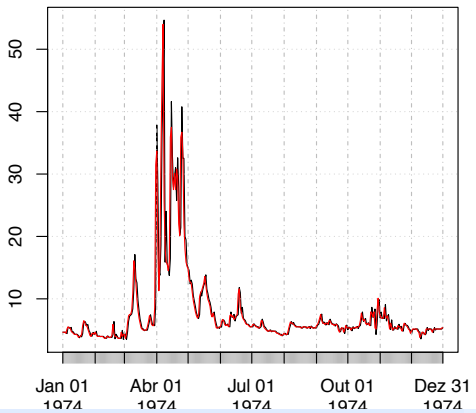


ARIMA models in R

```
preds.test <- fitted(Arima(ir,model=model))[(length(train)+1):length(ir)]
mad <- function(p,t) mean(abs(as.numeric(p)-as.numeric(t)))
cat("The average mean absolute error was ",mad(preds.test,test))


## The average mean absolute error was  0.9338048


## plotting this
plot(as.xts(as.numeric(preds.test),index(test)),
     main="Performance on the test data")
lines(test,col="red")
```



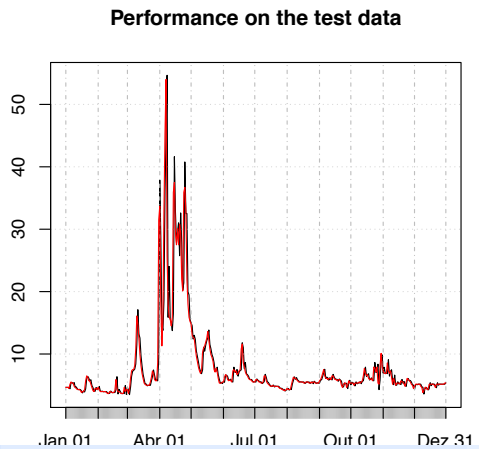
How to set the order of the ARIMA models ?

Function `auto.arima`

```
m <- auto.arima(train)
preds <- fitted(Arima(ir,model=m))[(length(train)+1):length(ir)]
cat("The average mean absolute error was ",mad(preds,test))


## The average mean absolute error was  1.118131


plot(as.xts(as.numeric(preds.test),index(test)),
     main="Performance on the test data")
lines(test,col="red")
```



Delay-Coordinate Embedding

Theorem (Takens, 1981)

Informally, it states we can uncover the dynamics of any time series given the information on e past values of the series. For that to be possible we need to know the correct embed size (how far back in time to look)

An Example of Delay-Coordinate Embedding

Example

Given the time series, $y_1, y_2, y_3, \cdots, y_{100}$, an embed dimension of 5, the resulting embed vectors are,

$$\begin{aligned} r_5 &= < y_5, y_4, y_3, y_2, y_1 > \\ r_6 &= < y_6, y_5, y_4, y_3, y_2 > \\ r_7 &= < y_7, y_6, y_5, y_4, y_3 > \\ r_8 &= < y_8, y_7, y_6, y_5, y_4 > \\ &\cdots \end{aligned}$$

Consequences of Delay-Coordinate Embedding

If the system dynamics can be captured by a certain embed, then we may try to model the relationship between the state of the system and the future values of the series.

That is, we can try to obtain a model of the form,
$Y_{t+h} = f(r_t)$

This modelling task can be handled by any multiple regression tool we have studied before!

An Example with SVMs

```
## A simple function to create an embeded data set
create.data <- function(ts,embed) {
    t <- index(ts)[-(1:(embed-1))]
    e <- embed(ts,embed)
    colnames(e) <- paste('V',embed:1,sep='')
    xts(e,t)
}
## Preparing the data
ds <- create.data(ir,5)
train <- ds['/1973-12-31']
test <- ds['1974-01-01/']
## Now obtaining an SVM
library(e1071)
m <- svm(V5 ~ .,train,cost=10)
p.s <- predict(m,test)
mad(p.s,test$V5)

## [1] 1.26
```


SVMs - 2

```
## plotting the predictions of SVM
plot(as.xts(as.numeric(p.s),index(test)),
     main="Performance on the SVM on the test data")
lines(test$V5,col="red")
```