Open for Innovation
KNIME

[L4-TS] Introduction to Time Series Analysis

KNIME AG

Agenda

- Introduction: What is Time Series Analysis
- Today’s Task, Dataset & Components
- Descriptive Analytics: Load, Clean, Explore
- Descriptive Analytics: Non-stationarity, Seasonality, Trend
- Quantitative Forecasting: Classical techniques
- ARIMA Models: ARIMA(p,d,q)
- Machine Learning based Models
- Hyperparameter Optimization
- Quick Intro to LSTM Networks
- Example of Time Series Analysis on Spark
- Conclusions & Summary

Introduction
What is Time Series Analysis?

Introduction

Since social and economic conditions are **constantly changing over time**, data analysts must be able to **assess and predict the effects of these changes**, in order to suggest the most appropriate actions to take

- It's therefore required to use appropriate **forecasting techniques** to support business, operations, technology, research, etc.
- **More accurate** and **less biased** forecasts can be one of the most effective driver of performance in many fields

→ **Time Series Analysis**, using statistical methods, allows to enhance comprehension and predictions on any quantitative variable of interest (sales, resources, financial KPIs, logistics, sensors' measurements, etc.)

Applications

The fields of application of **Time series Analysis** are numerous: *Demand Planning* is one of the most common application, however, from industry to industry there are other possible uses. For instance:

Logistics & Transportation

- Forecasting of **shipped packages:** workforce planning

Retail grocery

- Forecasting of **sales during promotions:** optimizing warehouses

Insurance

- **Claims prediction:** determining insurance policies

Manufacturing

- **Predictive Maintenance:** improving operational efficiency

Energy & Utilities

- **Energy load forecasting:** better planning and trading strategies

TS data vs. Cross Sectional data

A Time series is made up by **dynamic data** collected over time! Consider the differences between:

1. Cross Sectional Data

- Multiple objects observed at a particular point of time
- *Examples*: customers' behavioral data at today's update, companies' account balances at the end of the last year, patients' medical records at the end of the current month, …

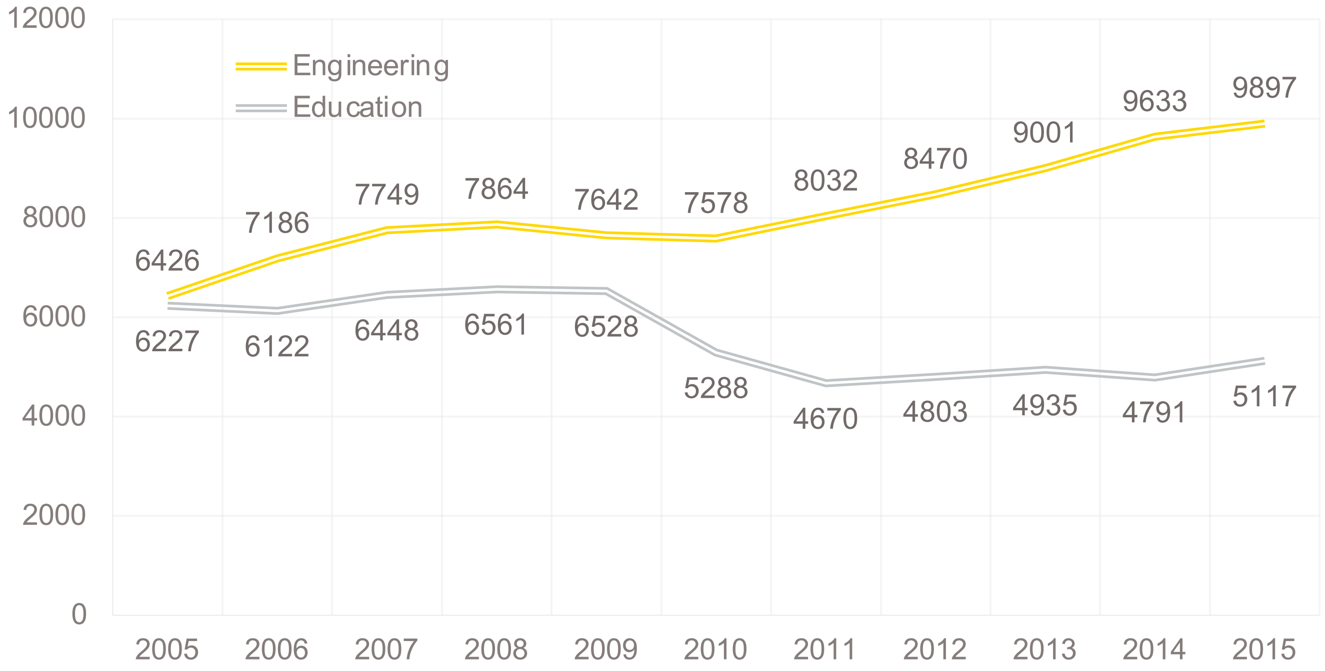
2. Time Series Data

- One single object (product, country, sensor, ..) observed over multiple equally-spaced time periods
- *Examples:* quarterly Italian GDP of the last 10 years, weekly supermarket sales of the previous year, yesterday's hourly temperature measurements, …

Examples

Time series example 1

Numbers of Doctorates Awarded in US, annual data – Engineering Vs. Education



At a glance

Annual data

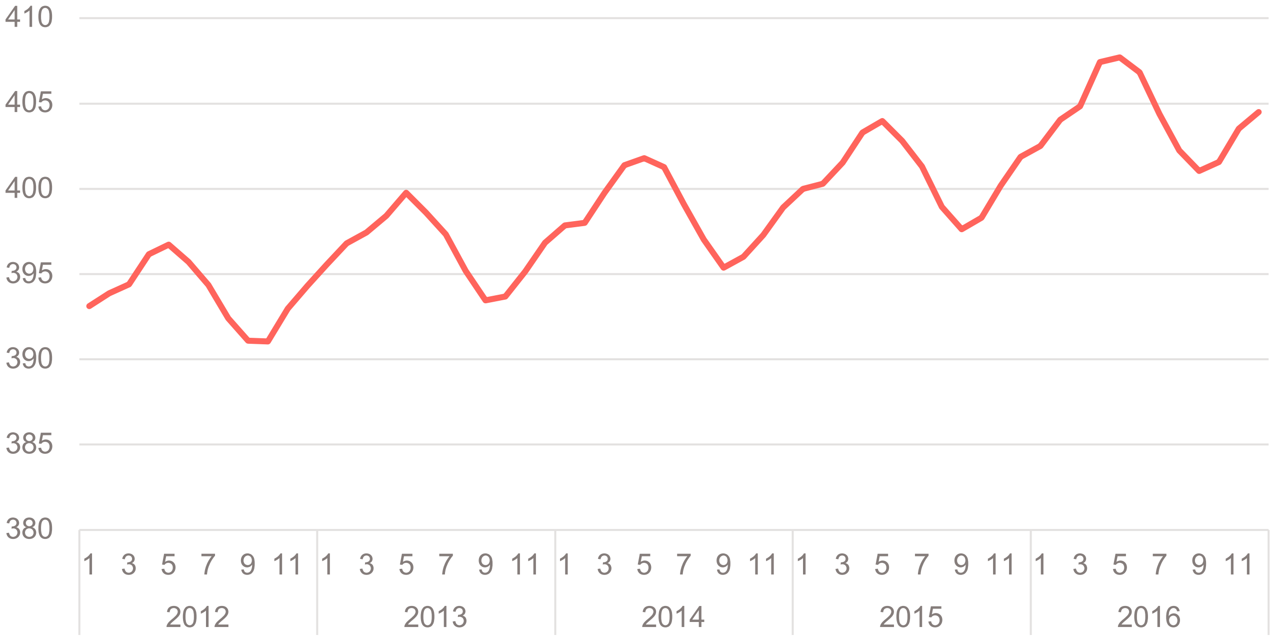
Different «directions»

No big fluctuations

Examples

Time series example 2

Monthly carbon dioxide concentration (globally averaged over marine surface sites)



At a glance

Monthly basis data

Regular pattern

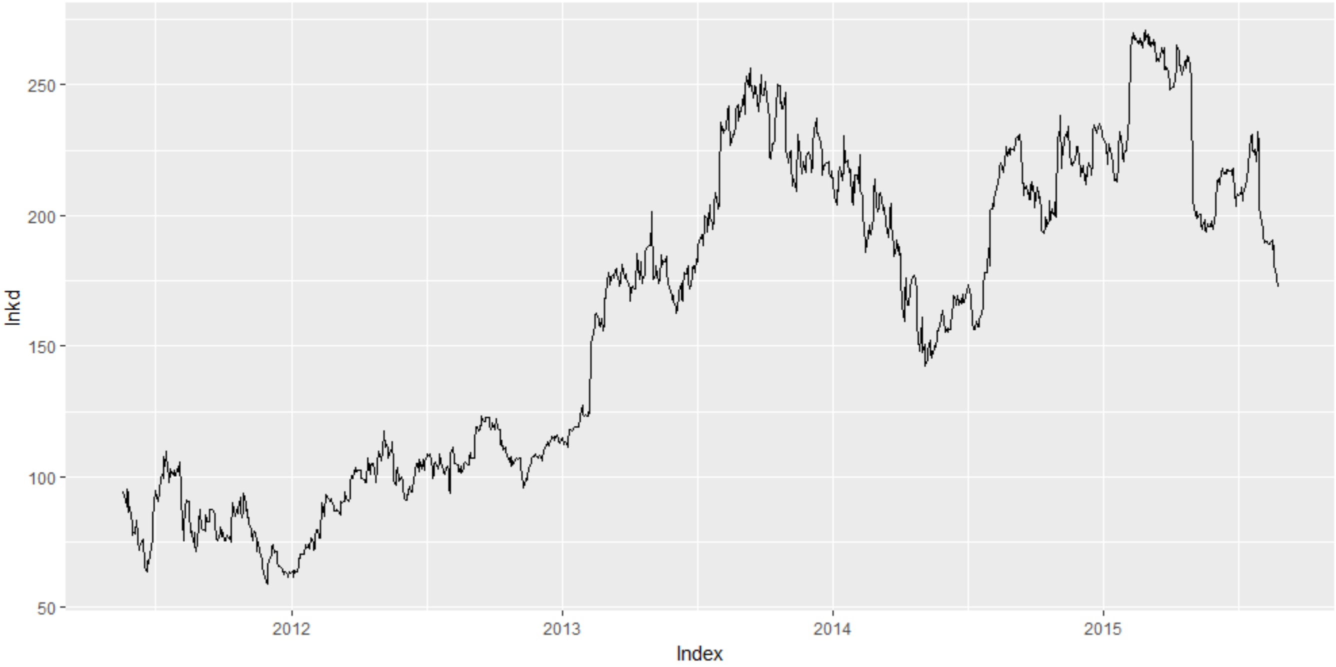
Constant fluctuations

Average value increases year by year

Examples

Time series example 3

LinkedIn daily stock market closing price



At a glance

Daily basis data

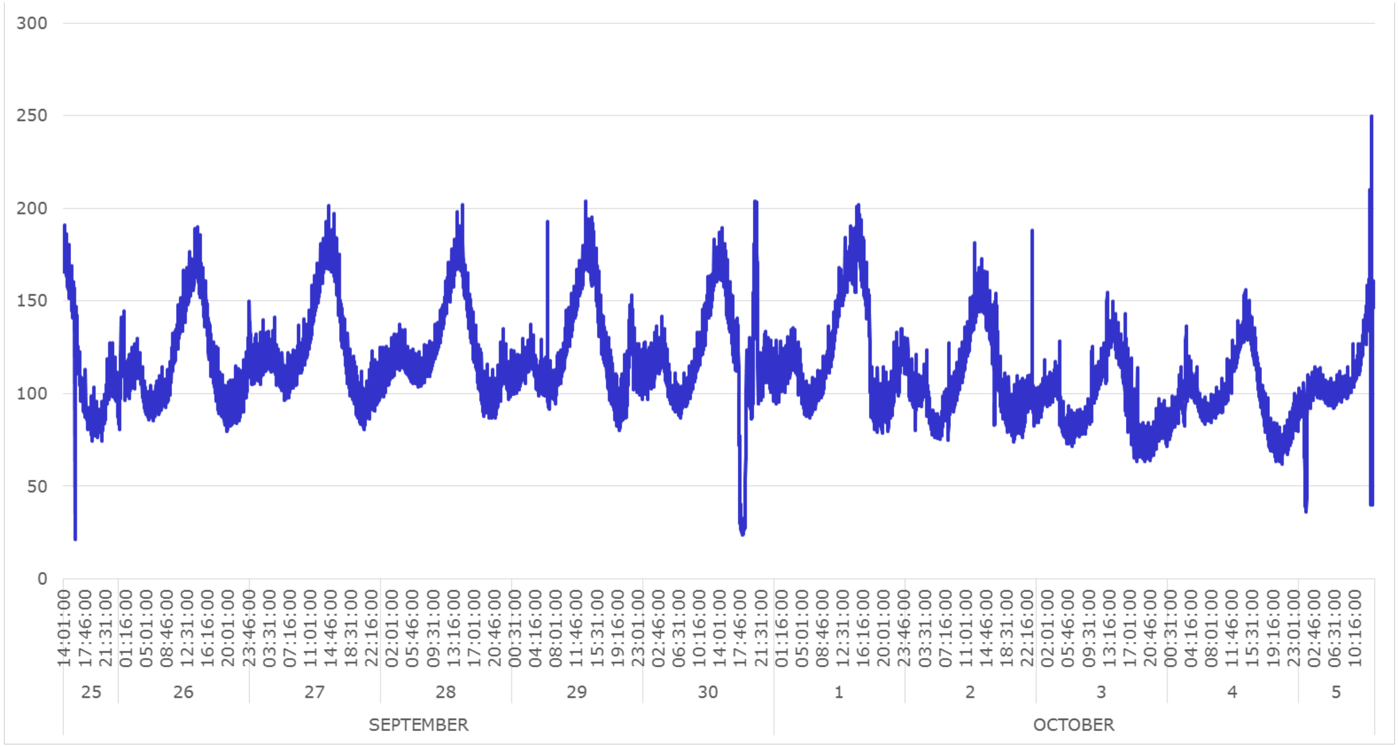
Very irregular dynamic

Many sudden changes

Examples

Time series example 4

Number of photos uploaded on the Instagram every minute (regional sub-sample)



At a glance

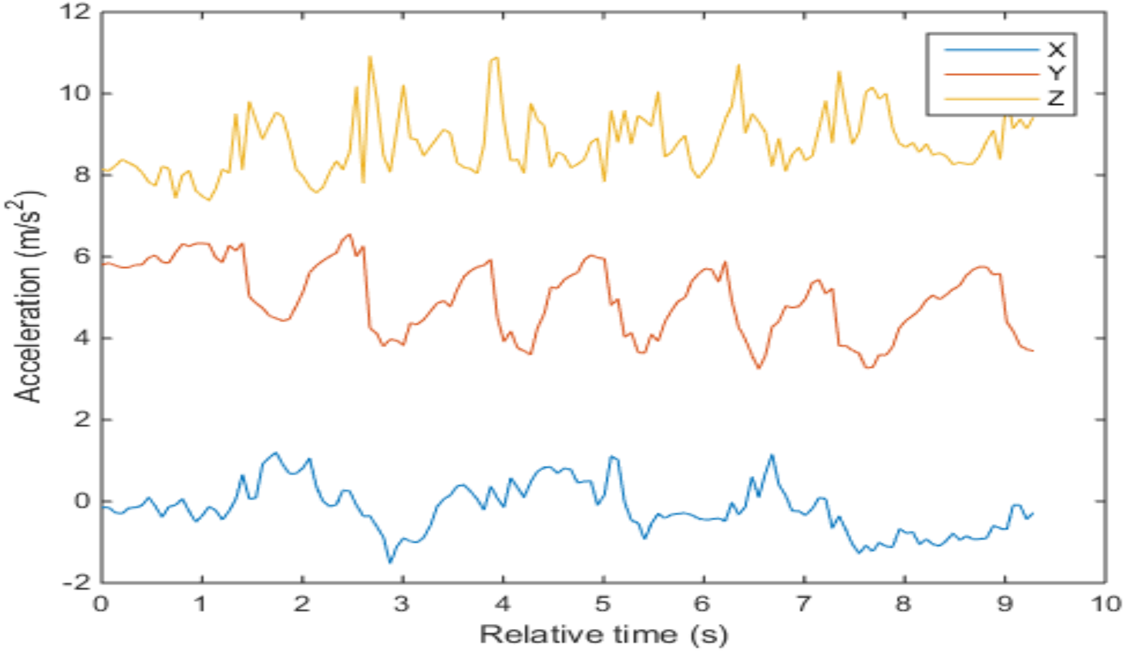
Minute basis data

Almost regular daily pattern but with some anomalies and spikes

Examples

Time series example 5

Acceleration detected by a smartphone sensors during a workout session (10 seconds)



At a glance

Milliseconds basis data

Each sensor has its own dynamics

Objectives

Main Objectives of Time Series Analysis

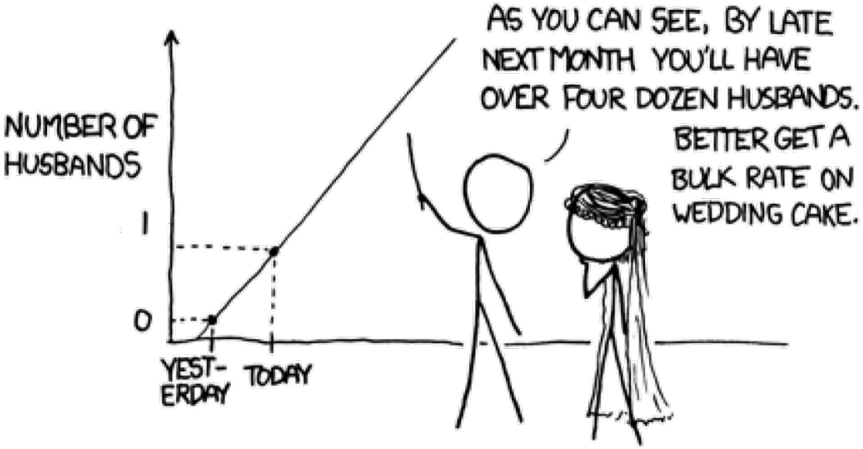
- **Summary description** (graphical and numerical) of data point vs. time
- **Interpretation** of specific series features (e.g. seasonality, trend, relationship with other series)
- **Forecasting** (e.g. predict the series values in $t+1, t+2, \ldots, t+k$)
- **Hypothesis testing and Simulation** (comparing different scenarios)

Objectives

Once someone said: ***«Forecasting is the art of saying what will happen in the future and then explaining why it didn't»***

- Frequently true... history is full of examples of «bad forecasts», just like IBM Chairman's famous quote in 1943: "*there is a world market for maybe five computers in the future.*"

The reality is that forecasting is a really tough task, and you can do really bad, just like in this cartoon..



AS YOU CAN SEE, BY LATE NEXT MONTH YOU'LL HAVE OVER FOUR DOZEN HUSBANDS. BETTER GET A BULK RATE ON WEDDING CAKE.
NUMBER OF HUSBANDS
1
0
YEST-ERDAY
TODAY

But we can do definitely better using **quantitative methods**.. and **common sense**!

GOAL: Reduce uncertainty and improve the accuracy of our forecasts

Definition

General definition: "A time series is a collection of observations made sequentially through time, whose dynamics is often characterized by short/long period fluctuations (seasonality and cycles) and/or long period direction (trend)"

Such observations may be denoted by $\boldsymbol{Y_1 , Y_2, Y_3, \dots \underset{\uparrow}{Y_t}, \dots , Y_T}$ since data are usually collected at discrete points in time

Observation at time t

- The interval between observations can be any time interval (seconds, minute, hours, days, weeks, months, quarters, years, etc.) and we assume that these time periods are **equally spaced**
- One of the most distinctive characteristics of a time series is the mutual dependence between the observations, generally called **SERIAL CORRELATION** OR **AUTOCORRELATION**

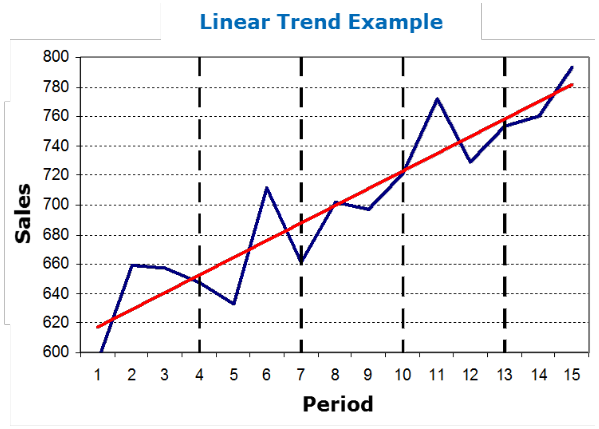
Time Series Properties: Main Elements

- TREND

  The general direction in which the series is running during a long period

  A **TREND** exists when there is a long-term increase or decrease in the data.

  It does not have to be necessarily linear (could be exponential or others functional form).

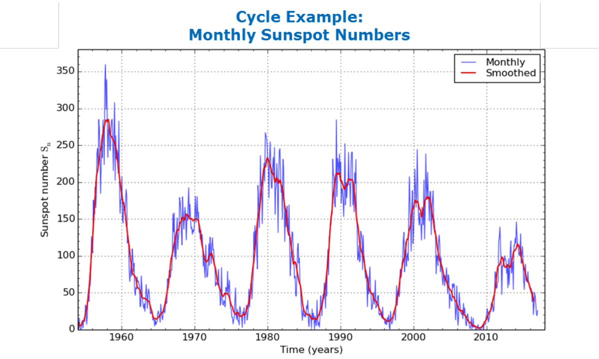


Linear Trend Example

- CYCLE

  Long-term fluctuations that occur regularly in the series A CYCLE is an oscillatory component (i.e. Upward or Downward swings) which is repeated after a certain number of years, so:

  - May vary in length and usually lasts several years (from 2 up to 20/30)
  - Difficult to detect, because it is often confused with the trend component



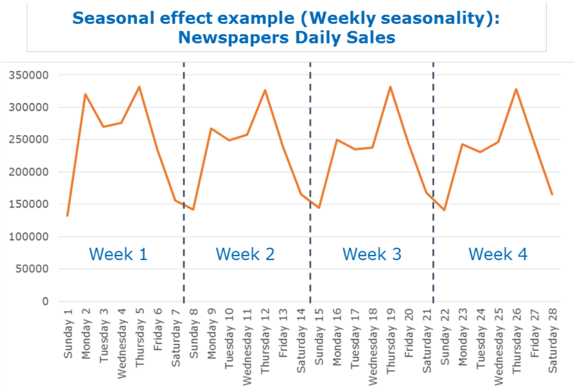
Cycle Example:
Monthly Sunspot Numbers

Time Series Properties: Main Elements

- ## SEASONAL EFFECTS

  Short-term fluctuations that occur regularly – often associated with months or quarters

  A **SEASONAL PATTERN** exists when a series is influenced by seasonal factors (e.g., the quarter of the year, the month, day of the week). Seasonality is always of a fixed and known period.

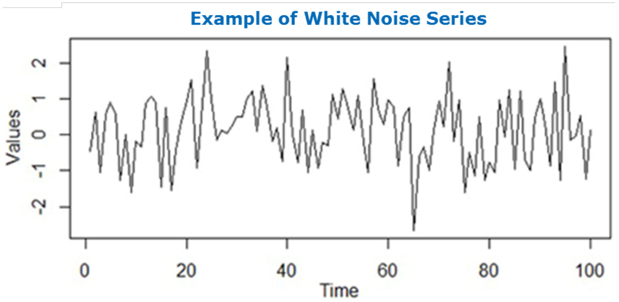


- ## RESIDUAL

  Whatever remains after the other components have been taken into account

  The residual/error component is everything that is not considered in previous components

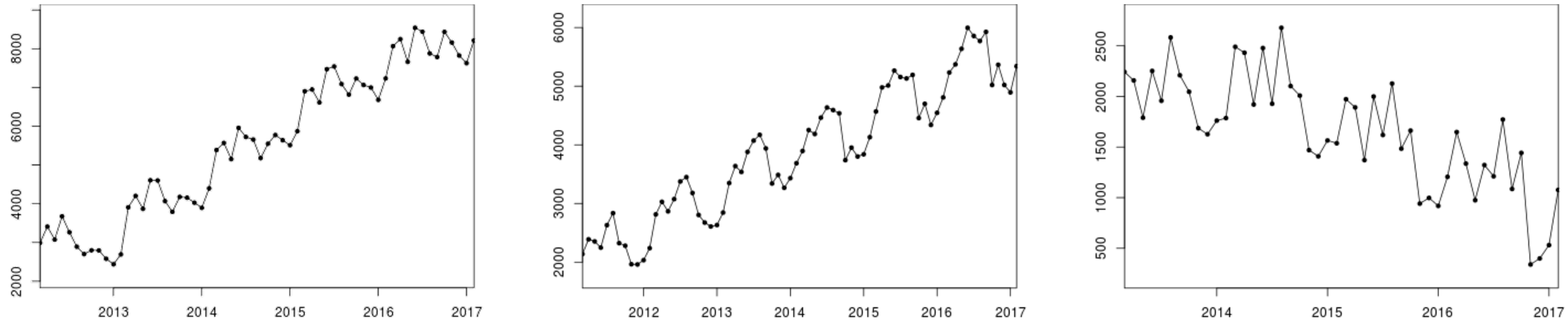
  Typically, it is assumed to be the sum of a set of random factors (e.g. a **white noise series**) not relevant for describing the dynamics of the series



Seasonal effect: additive seasonality

- When the seasonality in Additive, the dynamics of the components are **independents from each other**; for instance, an increase in the trend-cycle will not cause an increase in the magnitude of seasonal dips
- The difference of the trend and the raw data is **roughly constant in similar periods of time** (months, quarters) irrespectively of the tendency of the trend

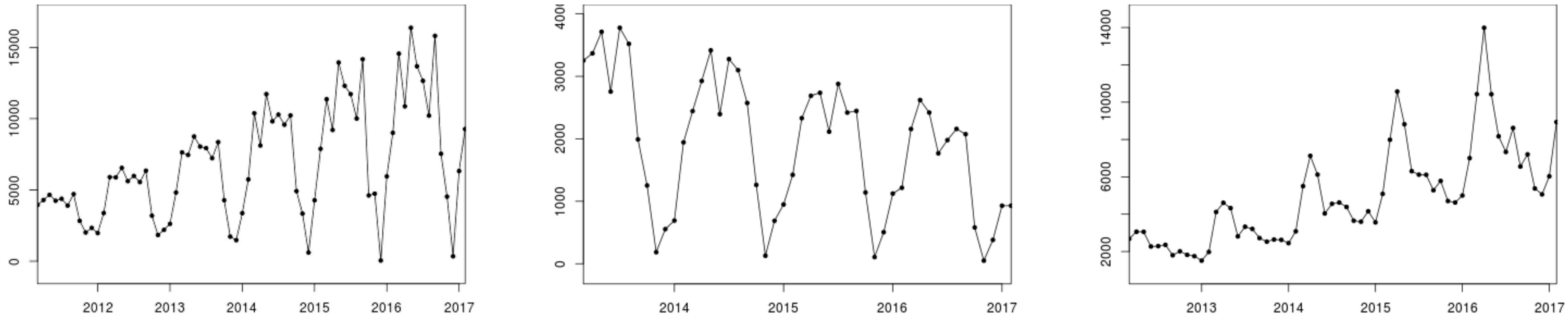
EXAMPLES OF ADDITIVE SEASONALITY



Seasonal effect: multiplicative seasonality

- In the multiplicative model the amplitude of the seasonality increase (decrease) with an increasing (decreasing) trend, therefore, on the contrary to the additive case, the **components are not independent from each other**
- When the variation in the seasonal pattern (or the variation around the trend-cycle) **appears to be proportional** to the level of the time series, then a multiplicative model is more appropriate.

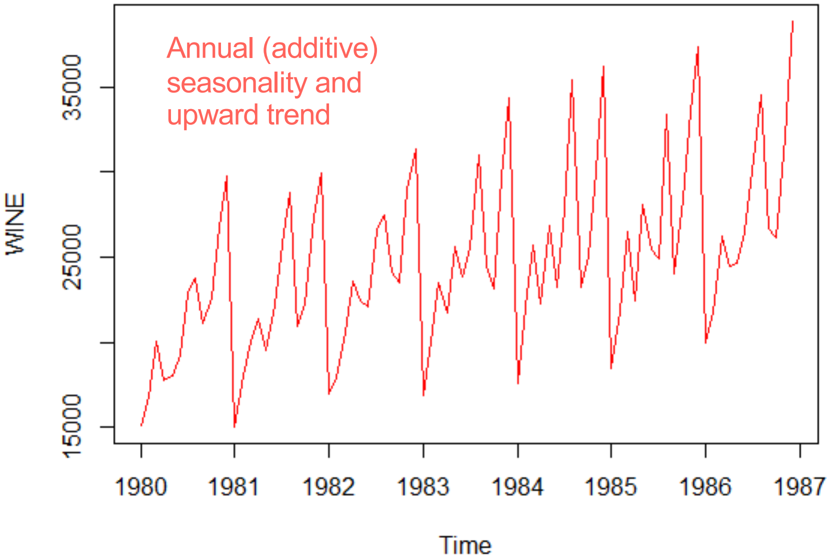
EXAMPLES OF MULTIPLICATIVE SEASONALITY



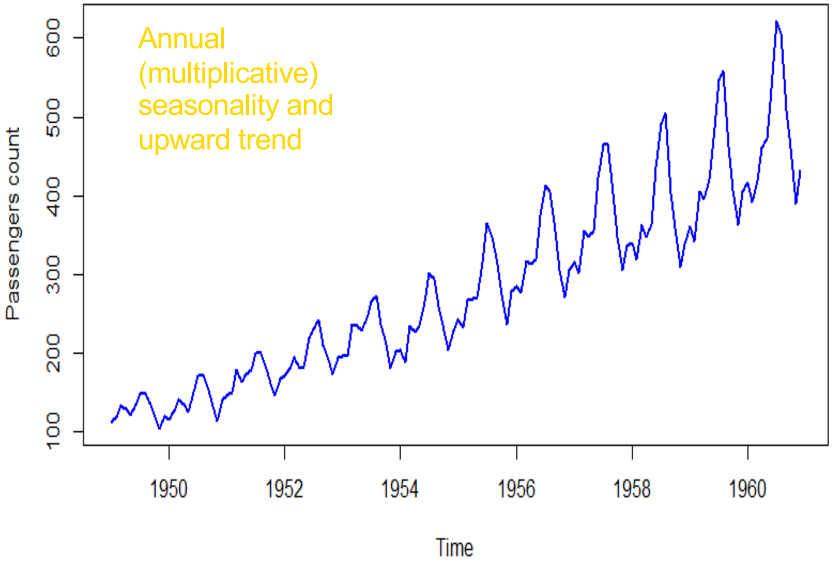
Graphical Analysis: Time Plot

- The first chart in time series analysis is the **TIME PLOT** → the observations are plotted against the time of observation, normally with consecutive observations joined by straight lines

Example of TS Plot of Australian monthly wine sales



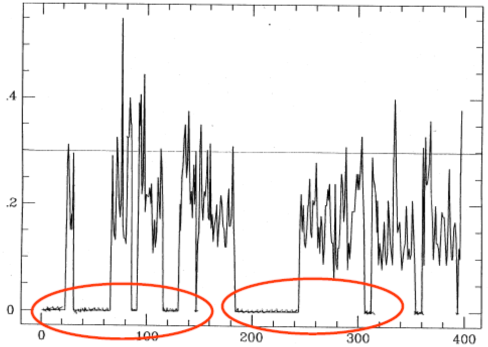
Example of TS Plot of Air Passengers (monthly) series



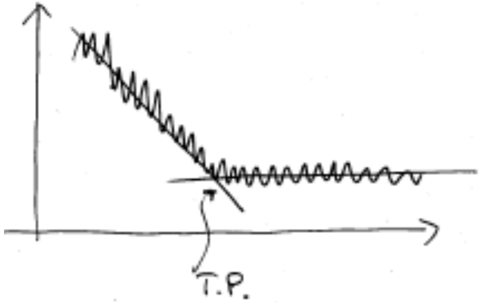
Graphical Analysis: Time Plot

- Insights you can get just from a simple **Time plot**
  - Is there a trend? Could it be linear or not?
  - Is there a seasonality effect?
  - Are there any long term cycles?
  - Are there any sharp changes in behaviour? Can such changes be explained?
  - Are there any missing values or “gap” in the series?
  - Are there any outliers, i.e. observations that differ greatly from the general pattern?
  - Is there any turning point/changing trend?

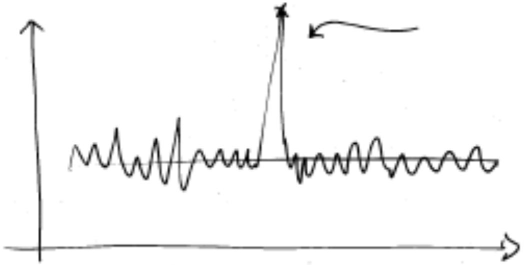
Series with gaps



Series with a turning point



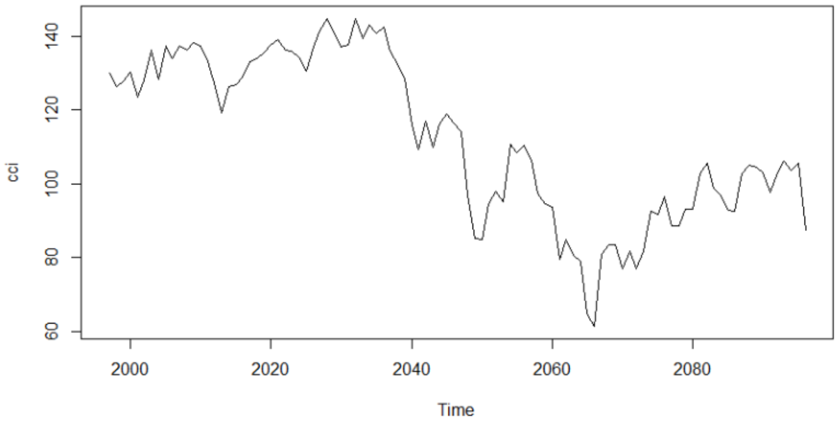
Series with an outlier



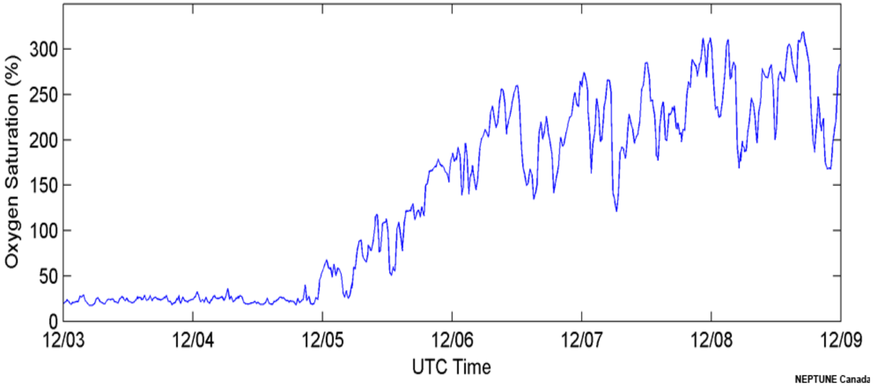
Graphical Analysis: Time Plot

- The **TIME PLOT** is very useful in cases where the series shows a very constant/simple dynamic (strong trend and strong seasonality), but in other cases could be difficult to draw clear conclusions

Consumer Cost Index



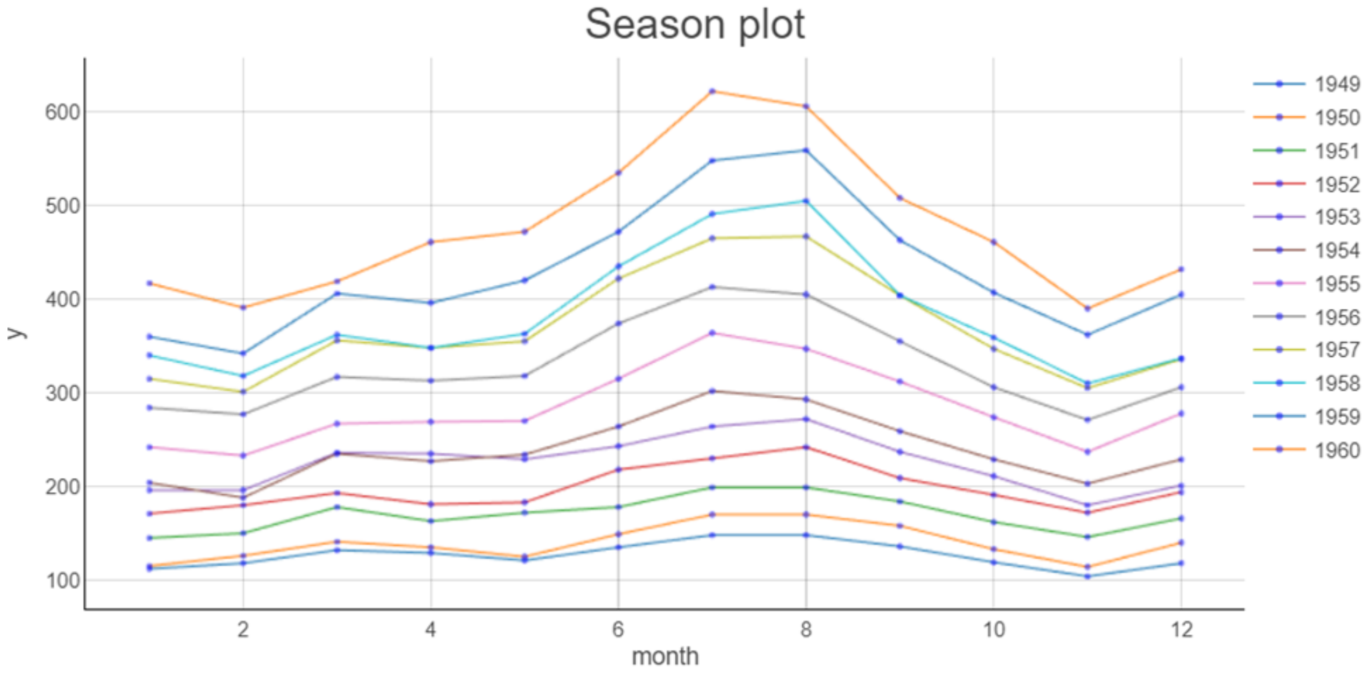
Oxygen saturation



  - Other graphical analyses and summary statistics could improve/extend the insights given by the simple time plot!

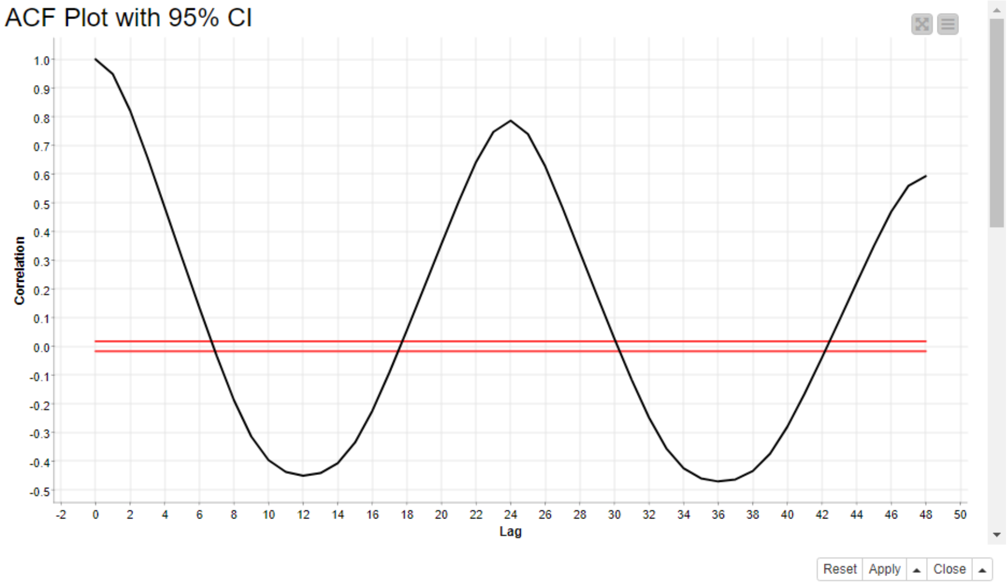
Graphical Analysis: Seasonal Plot

- Produce the **Seasonal plot** of the Time series in order to analyze more in detail the seasonal component (and possible changes in seasonality over time)



Numerical analysis: Auto Correlation Function (and ACF plot)

In order to go deeper inside the autocorrelation structure of the time series, you can create the Auto Correlation Function plot (**ACF plot**), also called *correlogram*: in this chart you can read the linear correlation index between the values in t and all the possible lags (t-1, t-2, …, t-k); the chart below shows all the correlations up to lag number 48

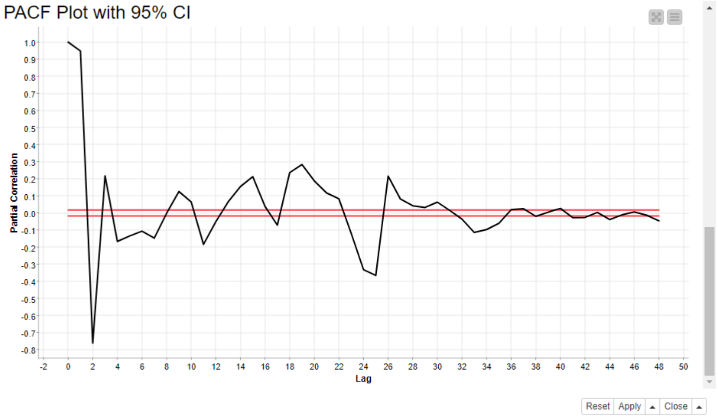


Numerical analysis: Auto Correlation Function (and ACF plot)

Together with the ACF, sometimes it is useful to analyze also the Partial Autocorrelation Function

The ACF plot shows the autocorrelations which measure the linear relationship between y_t and $y_{t-\mathrm{k}}$ for different values of *k* but consider that:

- if y_t and y_{t-1} are correlated, then y_{t-1} and y_{t-2} must also be correlated
- But then y_t and y_{t-2} might be correlated, simply because they are both connected to y_{t-1}
- → The **Partial Autocorrelation Function (PACF)** consider the linear relationship between y_t and $y_{t-\mathrm{k}}$ after *removing* the effects of other time lags $1, 2, 3, \dots, k-1$



Agenda

1. Introduction: What is Time Series Analysis
2. Today’s Task, Dataset & Components
3. Descriptive Analytics: Load, Clean, Explore
4. Descriptive Analytics: Non-stationarity, Seasonality, Trend
5. Quantitative Forecasting: Classical techniques
6. ARIMA Models: ARIMA(p,d,q)
7. Machine Learning based Models
8. Hyperparameter Optimization
9. Quick Intro to LSTM Networks
10. Example of Time Series Analysis on Spark
11. Conclusions & Summary

Descriptive Analytics

Stationarity, Seasonality, Trend

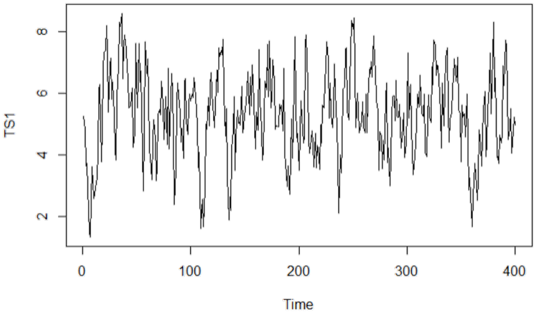
Stationarity

A time series can be defined as “**stationary**” when *its properties does not depend on the time at which the series is observed*, so that:

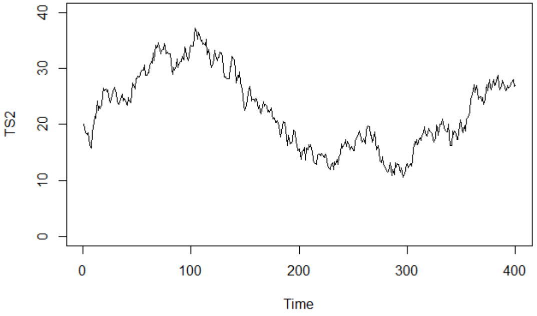
- the values oscillate frequently around the mean, independently from time
- the variance of the fluctuations remains constant across time
- the autocorrelation structure is constant over time and no periodic fluctuations exist

So, a time series that shows trend or seasonality is not stationary

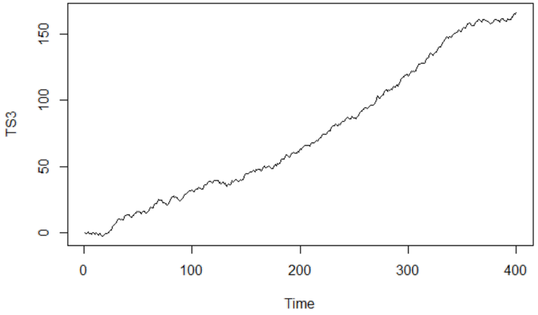
Stationary Time Series example



Non-Stationary Time Series example 1



Non-Stationary Time Series example 2



Stationarity

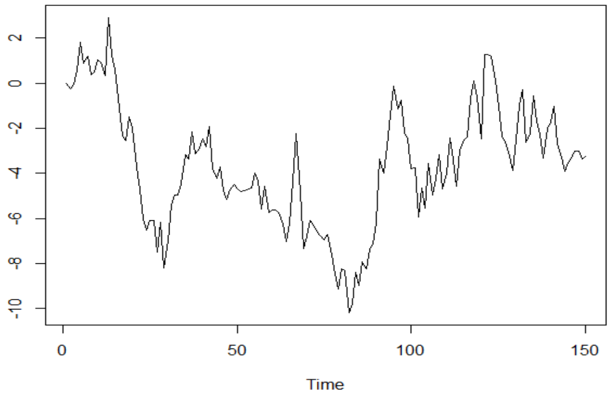
Typical examples of non-stationary series are all series that exhibit a deterministic trend (i.e. $y_t = \alpha + \beta \cdot t + \varepsilon_t$) or the so-called **“Random Walk”**

Random Walk (without drift) → $y_t = y_{t-1} + \varepsilon_t$ (where ε_t is white noise)

A random walk model is very widely used for non-stationary data, particularly financial and economic data.

- Random walks typically have:
  - long periods of apparent trends up or down
  - sudden and unpredictable changes in direction
  - variance and autocorrelation that depends on time!

Random Walk Example

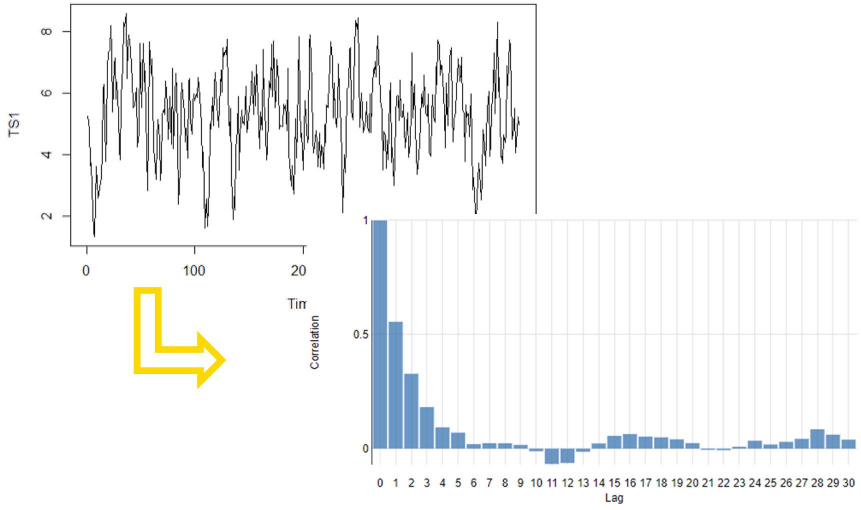


Stationarity

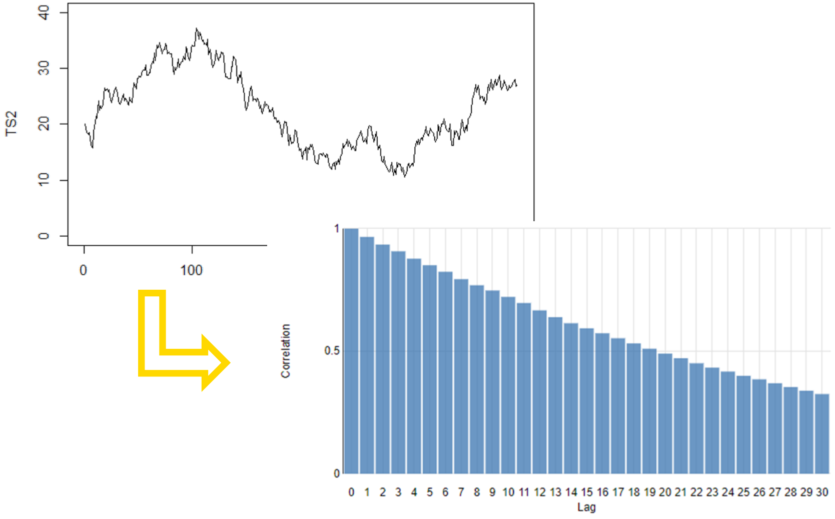
Besides looking at the time plot of the data, the ACF plot is also useful for identifying non-stationary TS:

→ for a stationary time series, the ACF will **drop to zero (i.e. within confidence bounds) relatively quickly**, while the ACF of non-stationary data **decreases slowly**

Stationary Time Series example



Non-Stationary Time Series example 1 (random walk!)



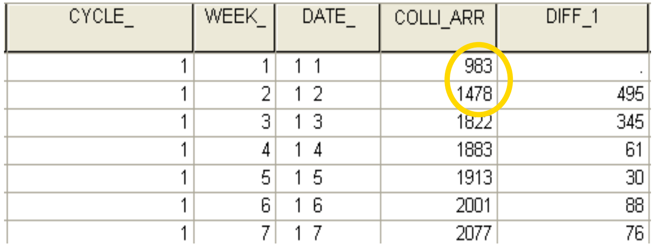
Differencing

- One way to make a time series stationary is to compute the differences between consecutive observations → This is known as DIFFERENCING
  - Differencing can help **stabilize the mean** of a time series by removing changes in the level of a time series, and so eliminating trend (and also seasonality, using a specific differencing order)
  - The **Order of Integration** for a Time Series, denoted ***I*(*d*)**, reports the minimum number of differences (d) required to obtain a stationary series (*note:* ***I*(*0*)** → it means the series is stationary!)
  - Transformations such as logarithms can help to stabilize the variance of a time series

Differenced Time Series (first order)

y_t

$y'_t = y_t - y_{t-1}$

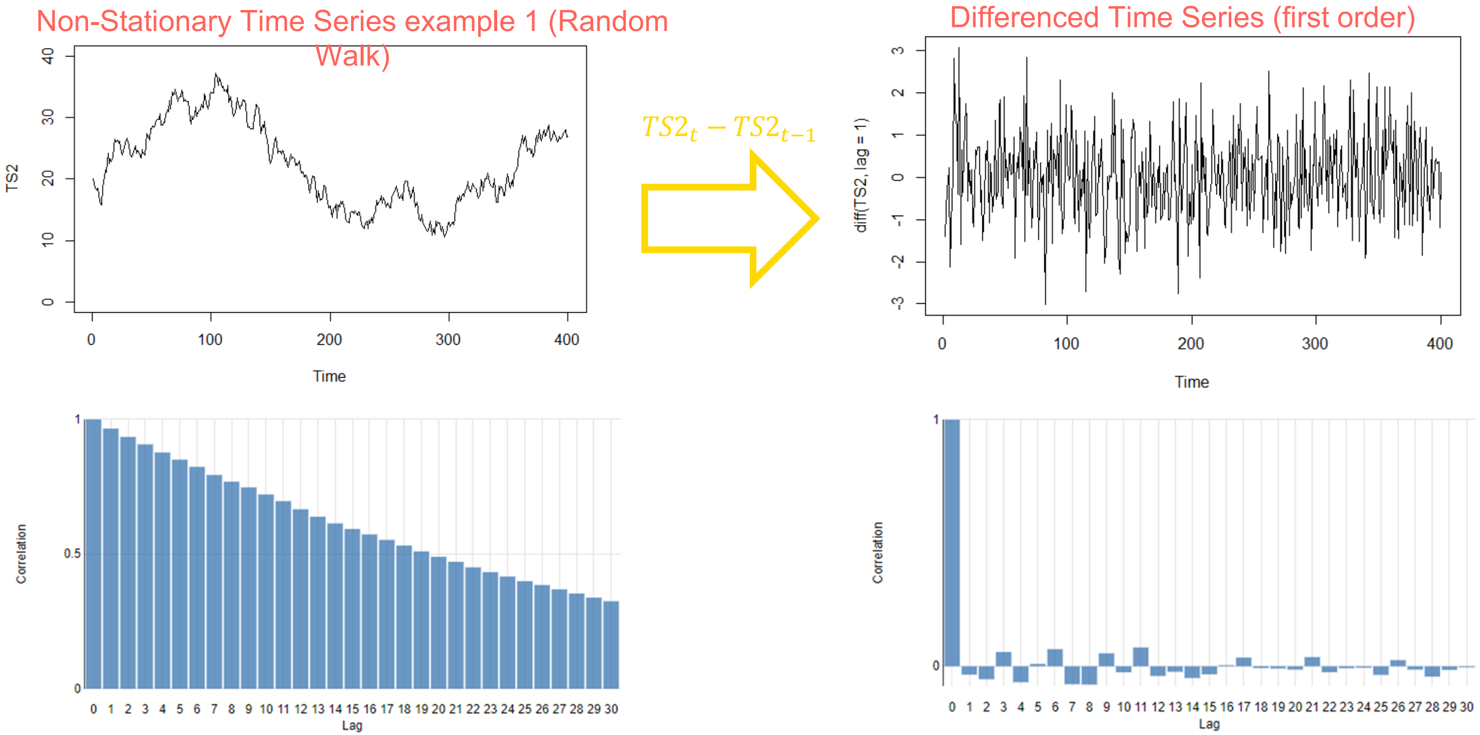


CYCLE_	WEEK_	DATE_	COLLI_ARR	DIFF_1
1	1	1 1	983	.
1	2	1 2	1478	495
1	3	1 3	1822	345
1	4	1 4	1883	61
1	5	1 5	1913	30
1	6	1 6	2001	88
1	7	1 7	2077	76

1478 – 983 = **495**

Differencing

Example: use differencing to make stationary a non-stationary series



Non-Stationary Time Series example 1 (Random Walk)

$TS2_t - TS2_{t-1}$

Differenced Time Series (first order)

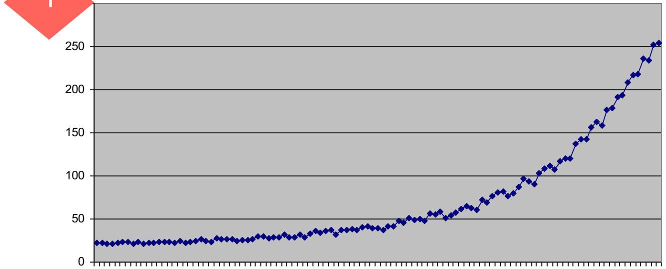
No significative autocorrelation exists → applying first differences to a random walk generates a white noise

Differencing

Occasionally the differenced data will not appear stationary and it may be necessary to difference the data a second time to obtain a stationary series ($y''_t = y'_t - y'_{t-1} = [y_t - y_{t-1}] - [y_{t-1} - y_{t-2}]$)*

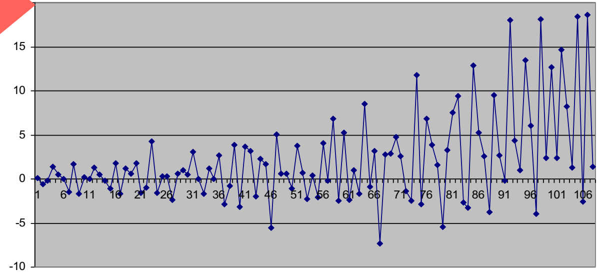
1

Original Time Series: non-stationary (mean and variance)



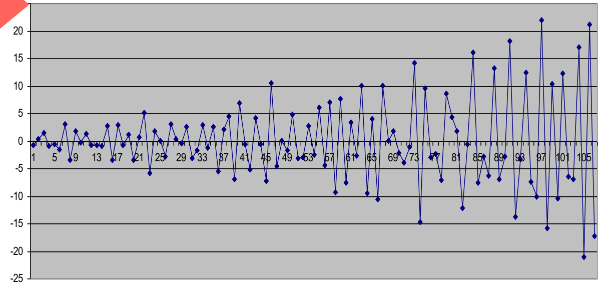
2

First Order Differencing: non-stationary (mean and variance)



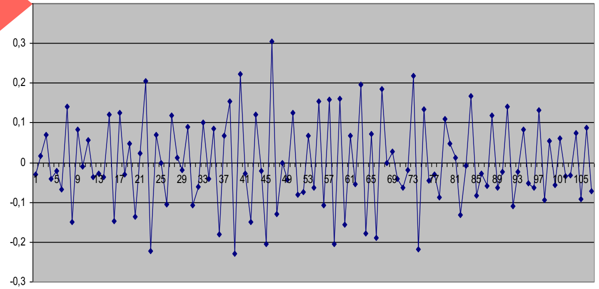
3

Second Order Diff.: stationary in mean, but not in variance



4

Double Differencing applied to Log(Series): stationary series



* it's almost never necessary to go beyond second-order differences

Differencing

A **seasonal difference** is the difference between an observation and the corresponding observation from the previous (seasonal) cycle

$$y'_t = y_t - y_{t-F}$$

Where F is the (seasonal) cycle frequency

→ *The seasonal differencing removes strong and stable seasonality pattern* (and transform into a white noise the so called "seasonal random walk", i.e. $\mathrm{y_t = y_{t-F} + \varepsilon_t}$)

Consider that:

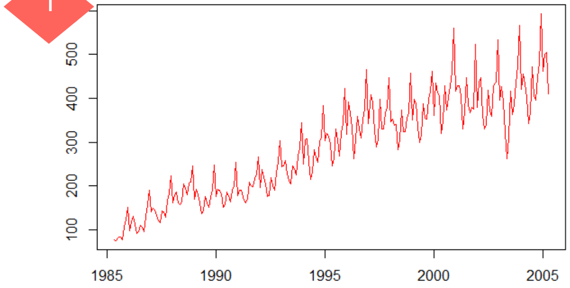
- Sometimes it's needed to apply both "simple" first differencing and seasonal differencing in order to obtain a stationary series
- It makes no difference which is done first—the result will be the same
- However, if the data have a strong seasonal pattern, it's recommended that seasonal differencing be done first because sometimes the resulting series will be stationary and there will be no need for a further non-seasonal differencing

Differencing

Consider the following example where a set of differencing has been applied to “Monthly Australian overseas visitors” TS

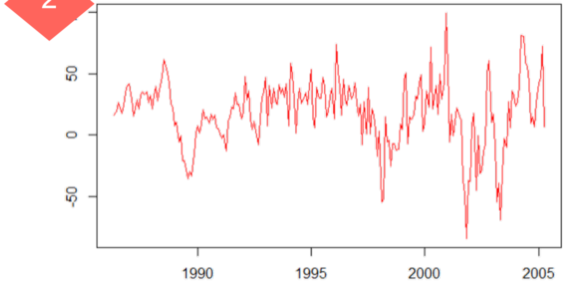
1

Original Time Series (y_t)



2

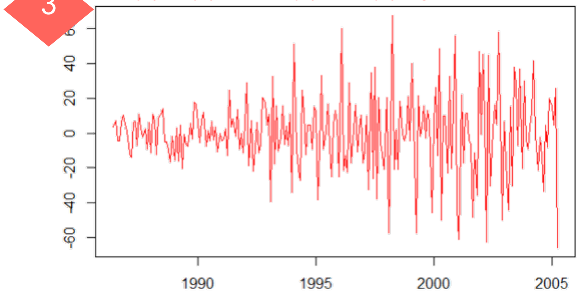
Seasonal Differencing $(y_t - y_{t-12})$



Applying first differencing to seasonal differenced series

$([y_t \ - y_{t-12}] - [y_{t-1} \ - y_{t-13}])$

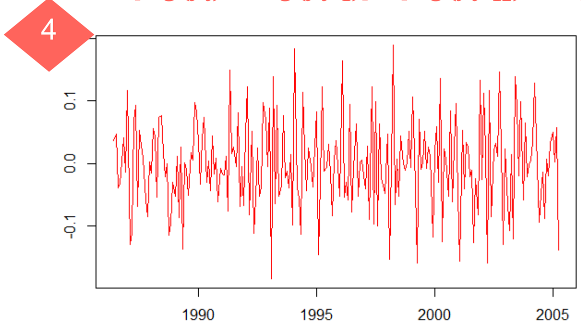
3



Use log trasformation in order to stabilize the variance

$([\boldsymbol{log}(y_t) \ - \boldsymbol{log}(y_{t-1})] - [\boldsymbol{log}(y_{t-12}) \ - \boldsymbol{log}(y_{t-13})])$

4



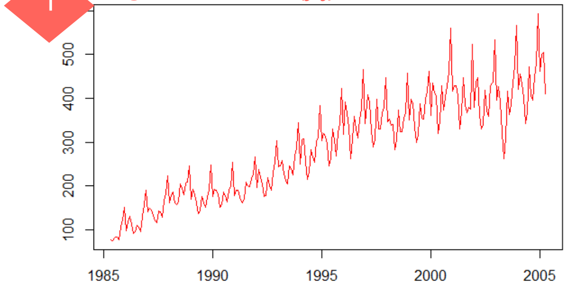
The series now appears to be stationary

Differencing

Same example of the previous slide, but changing the differencing process order → the final result is…

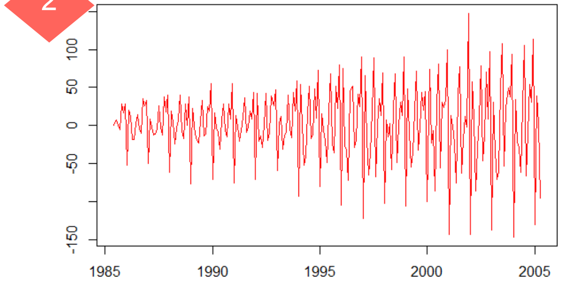
1

Original Time Series (y_t)



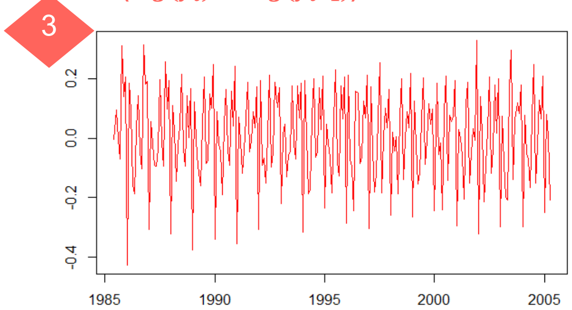
2

First Order Differencing ($y_t - y_{t-1}$)



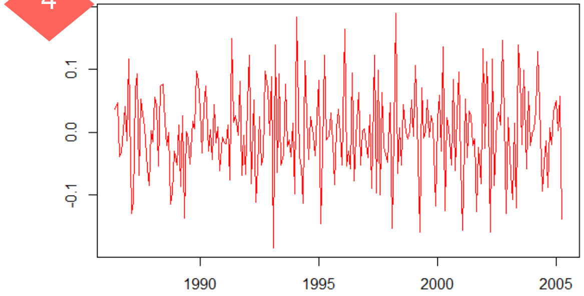
First Order Diff. after log transformation ($log(y_t)\ - log(y_{t-1})$)

3



Applying seasonal differencing to first order differenced of log series ($[log(y_t)\ - log(y_{t-1})] - [log(y_{t-12})\ - log(y_{t-13})]$)

4



The series is now stationary

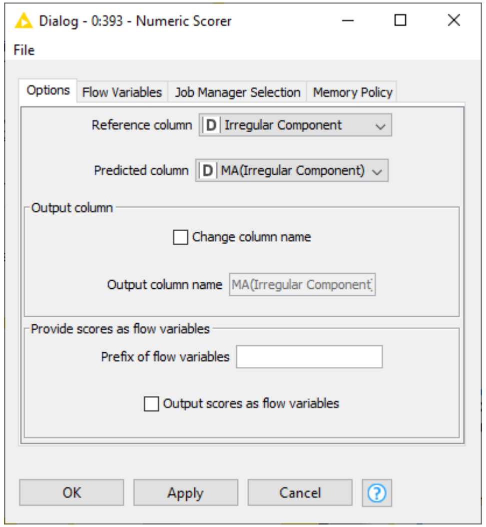
Numeric Errors: Formulas

Error Metric	Formula	Notes
R-squared	$1-\frac{\sum_{i=1}^{n}(f(x_i)-y_i)^2}{\sum_{i=1}^{n}(y_i-\overline{y})^2}$	Universal range: the closer to 1 the better
Mean absolute error (MAE)	$\frac{1}{n}\sum_{i=1}^{n}\lvert f(x_i)-y_i\rvert$	Equal weights to all distances Same unit as the target column
Mean squared error (MSE)	$\frac{1}{n}\sum_{i=1}^{n}(f(x_i)-y_i)^2$	Common loss function
Root mean squared error (RMSE)	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(f(x_i)-y_i)^2}$	Weights big differences more Same unit as the target column
Mean signed difference	$\frac{1}{n}\sum_{i=1}^{n}(f(x_i)-y_i)$	Only informative about the direction of the error
Mean absolute percentage error (MAPE)	$\frac{1}{n}\sum_{i=1}^{n}\frac{\lvert f(x_i)-y_i\rvert}{\lvert y_i\rvert}$	Requires non-zero target column values

Numeric Scorer Node

Evaluate numeric predictions

- Compare actual target column values to predicted values to evaluate goodness of fit.
- Report R^2, RMSE, MAPE, etc.



Dialog - 0:393 - Numeric Scorer

Reference column: D Irregular Component

Predicted column: D MA(Irregular Component)

Output column

☐ Change column name

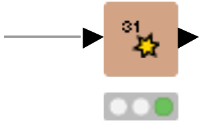
Output column name: MA(Irregular Component)

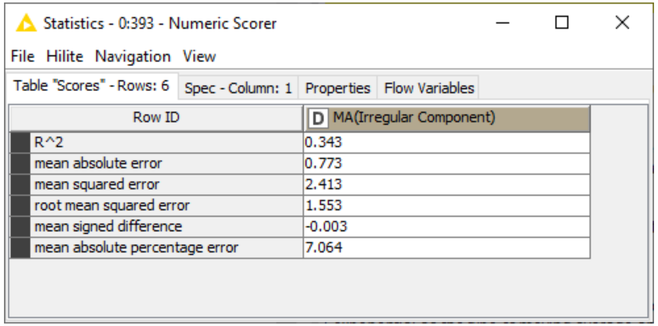
Provide scores as flow variables

Prefix of flow variables:

☐ Output scores as flow variables

Numeric Scorer





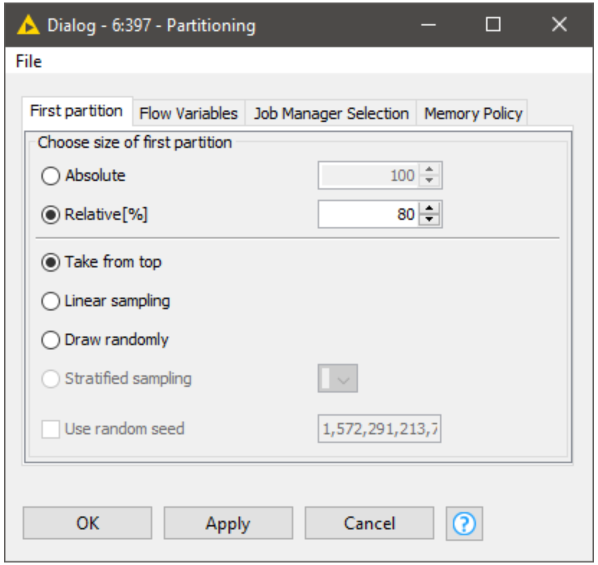
Statistics - 0:393 - Numeric Scorer

Row ID	D MA(Irregular Component)
R^2	0.343
mean absolute error	0.773
mean squared error	2.413
root mean squared error	1.553
mean signed difference	-0.003
mean absolute percentage error	7.064

Partitioning for Time Series

- When Partitioning data for training a Time Series model it is important your training data comes before your test data chronologically.
- This will mirror how the model is used in deployment, always forecasting the future.
- To do this make sure your data is properly sorted and partition with the “Take from top” option. In the KNIME node.

Partitioning



Choose size of first partition

○ Absolute 100

● Relative[%] 80

● Take from top

○ Linear sampling

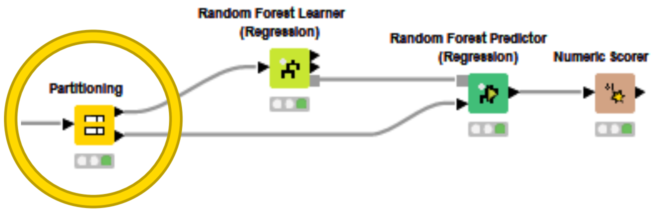
○ Draw randomly

○ Stratified sampling

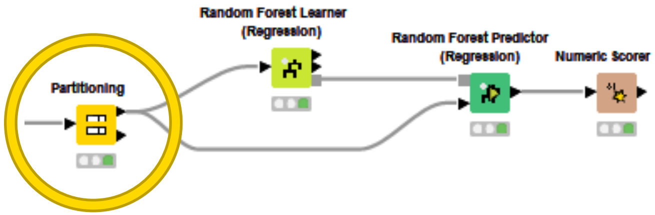
□ Use random seed 1,572,291,213,7

In-Sample vs. Out-sample

Out-Sample Static


Random Forest Learner (Regression)
Random Forest Predictor (Regression)
Numeric Scorer
Partitioning

In-Sample Static

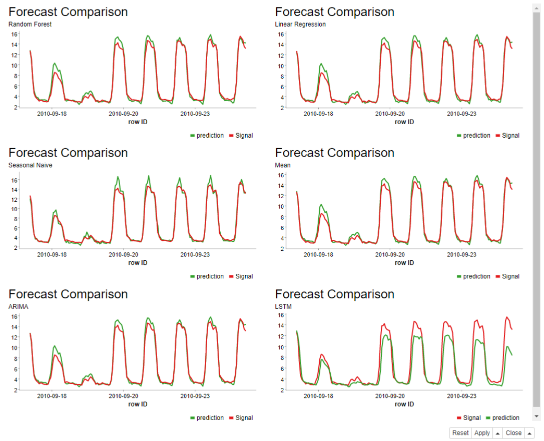

Random Forest Learner (Regression)
Random Forest Predictor (Regression)
Numeric Scorer
Partitioning

- Data used to train is the sample data
- Forecasts on the sample data are called **In-Sample** Forecasts
- Forecasts on other data are called **Out-Sample** Forecasts
- Either Forecast is called **Dynamic** if it uses prior Forecasts as its inputs, if real values are used it is called **Static**

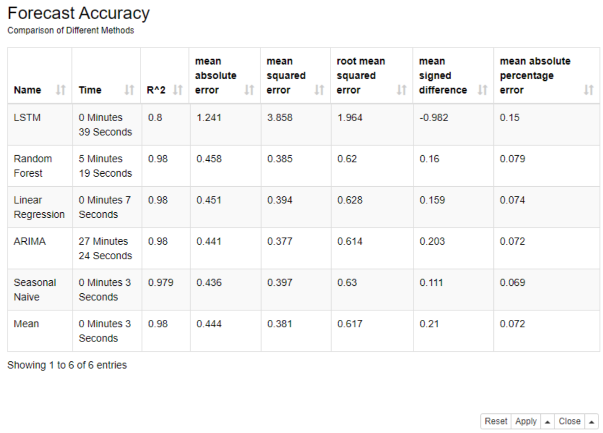
Model Evaluation

- Assess the expected forecast accuracy of your model by comparing actual and predicted time series
  - Training data vs. in-sample predictions
  - Test data vs. out-of-sample predictions

Visual comparison in a line plot:



Numeric comparison by error metrics:



Forecast Accuracy

Comparison of Different Methods

Name	Time	R^2	mean absolute error	mean squared error	root mean squared error	mean signed difference	mean absolute percentage error
LSTM	0 Minutes 39 Seconds	0.8	1.241	3.858	1.964	-0.982	0.15
Random Forest	5 Minutes 19 Seconds	0.98	0.458	0.385	0.62	0.16	0.079
Linear Regression	0 Minutes 7 Seconds	0.98	0.451	0.394	0.628	0.159	0.074
ARIMA	27 Minutes 24 Seconds	0.98	0.441	0.377	0.614	0.203	0.072
Seasonal Naive	0 Minutes 3 Seconds	0.979	0.436	0.397	0.63	0.111	0.069
Mean	0 Minutes 3 Seconds	0.98	0.444	0.381	0.617	0.21	0.072

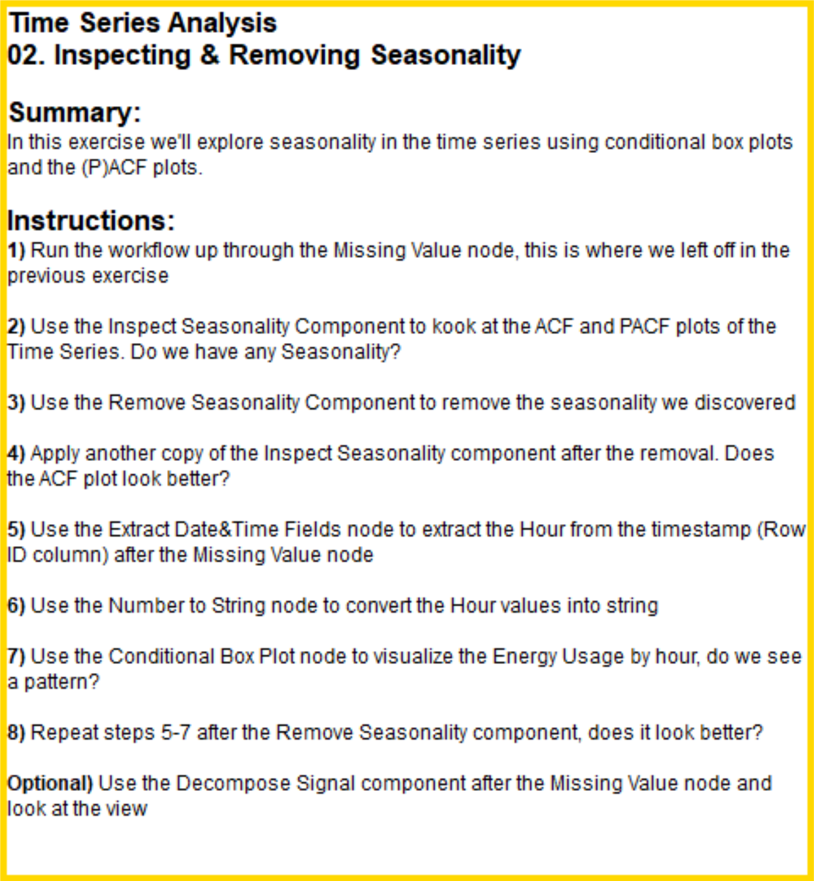
Showing 1 to 6 of 6 entries

Agenda

1. Introduction: What is Time Series Analysis
2. Today's Task, Dataset & Components
3. Descriptive Analytics: Load, Clean, Explore
4. Descriptive Analytics: Non-stationarity, Seasonality, Trend
5. Quantitative Forecasting: Classical techniques
6. ARIMA Models: ARIMA(p,d,q)
7. Machine Learning based Models
8. Hyperparameter Optimization
9. Quick Intro to LSTM Networks
10. Example of Time Series Analysis on Spark
11. Conclusions & Summary

Exercise 2: Inspecting and Removing Seasonality

- Use ACF plots to inspect seasonality from energy consumption data
- Remove seasonality and check again the ACF plot
- Compare hourly energy consumption values before and after removing seasonality
- Optional: split energy consumption data into a trend, seasonality, and residual



Time Series Analysis
02. Inspecting & Removing Seasonality

Summary:
In this exercise we'll explore seasonality in the time series using conditional box plots and the (P)ACF plots.

Instructions:
1) Run the workflow up through the Missing Value node, this is where we left off in the previous exercise

2) Use the Inspect Seasonality Component to kook at the ACF and PACF plots of the Time Series. Do we have any Seasonality?

3) Use the Remove Seasonality Component to remove the seasonality we discovered

4) Apply another copy of the Inspect Seasonality component after the removal. Does the ACF plot look better?

5) Use the Extract Date&Time Fields node to extract the Hour from the timestamp (Row ID column) after the Missing Value node

6) Use the Number to String node to convert the Hour values into string

7) Use the Conditional Box Plot node to visualize the Energy Usage by hour, do we see a pattern?

8) Repeat steps 5-7 after the Remove Seasonality component, does it look better?

Optional) Use the Decompose Signal component after the Missing Value node and look at the view

Quantitative Forecasting

Classical Techniques

Qualitative vs. Quantitative

The approaches to forecasting are essentially two: *qualitative approach and quantitative approach*

- **Qualitative forecasting** methods are adopted when historical data are not available (e.g. estimate the revenues of a new company that clearly doesn’t have any data available). They are highly subjective methods.

- **Quantitative forecasting** techniques are based on *historical quantitative data*;
  →the analyst, starting from those data, tries to understand the underlying structure of the phenomenon of interest and then to use the same historical data for forecasting purposes

Our focus

Quantitative forecasting

The basis for quantitative analysis of time series is the assumption that there are factors that influenced the dynamics of the series in the past and these factors continue to **bring similar effects in also in the future**

Main methods used in Quantitative Forecasting:

1. **Classical Time Series Analysis:** analysis and forecasts are based on identification of structural components, like trend and seasonality, and on the study of the serial correlation → *univariate time series analysis*
2. **Explanatory models:** analysis and forecasts are based both on past observations of the series itself and also on the relation with other possible predictors → *multivariate time series analysis*
3. **Machine learning models:** Different Artificial Neural Networks algorithms used to forecast time series (both in univariate or multivariate fashion)

Classical Time Series Analysis

The main tools used in the Classical Time Series Analysis are:

- **Classical Decomposition:** considers the time series as the overlap of several elementary components (i.e. trend, cycle, seasonality, error)
- **Exponential Smoothing:** method based on the weighting of past observations, taking into account the overlap of some key time series components (trend and seasonality)
- **ARIMA** (*AutoRegressive Integrated Moving Average*): class of statistical models that aim to treat the correlation between values of the series at different points in time using a regression-like approach and controlling for seasonality

Which model?

The choice of **the most appropriate method of forecasting** is influenced by a number of factors, that are:

- **Forecast horizon**, in relation to TSA objectives
- Type/amount of **available data**
- Expected **forecastability**
- Required **readability** of the results
- **Number of series** to forecast
- **Deployment** frequency of the models
- Development **complexity**
- Development **costs**

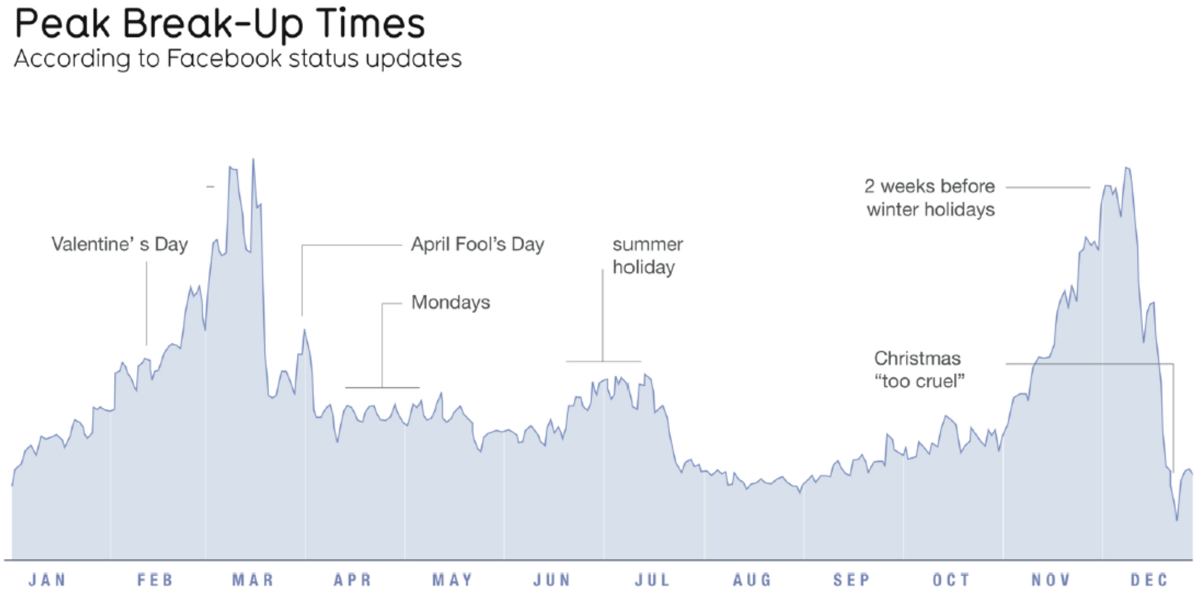
Interpretation issues

IMPORTANT: Remember that quantitative data ARE NOT JUST NUMBERS..

.. they have **a story to tell**, especially if your data are time series!

So.. always try to understand what's going on from a logical/business point of view: try to give an interpretation to the observed dynamics!

Example 1:
can you draw something useful looking at this series?



ARIMA Models

ARIMA(p,d,q)

Goal of this Section

1. Introduction to ARIMA
2. ARIMA Models
3. ARIMA Model selection
4. ARIMAX

Exponential Smoothing vs. ARIMA

While exponential smoothing models are based on a description of level, trend and seasonality in the data, ARIMA models aim to describe the **autocorrelations in the data**

REMINDER: Just as correlation measures the amount of a linear relationship between two variables, AUTOCORRELATION measures the linear relationship between *lagged values* of a time series

- There are several autocorrelation coefficients, depending on the lag length
- r_1 measures the relationship between y_t and y_{t-1}, r_2 measures the relationship between y_t and y_{t-2}, and so on

Before starting with ARIMA models is useful to give a look to a preliminary concept: what is a **linear regression model**?

ARIMA Models: General framework

An ARIMA model is a numerical expression indicating how the observations of a target **variable are statistically correlated with past observations of the same variable**

- ARIMA models are, in theory, the most general class of models for forecasting a time series which can be "**stationarized**" by transformations such as differencing and lagging
- The easiest way to think of ARIMA models is as fine-tuned versions of random-walk models: the fine-tuning consists of adding lags of the differenced series and/or lags of the forecast errors to the prediction equation, as needed to remove any remains of autocorrelation from the forecast errors

In an ARIMA model, in its most complete formulation, are considered:

- An **Autoregressive (AR)** component, seasonal and not
- A **Moving Average (MA)** component, seasonal and not
- The order of **Integration (I)** of the series

That's why we call it ARIMA (Autoregressive Integrated Moving Average)

ARIMA Models: General framework

The most common notation used for ARIMA models is:

$$ARIMA(p, d, q)\ (P, D, Q)s$$

where:

- **p** is the number of autoregressive terms
- **d** is the number of non-seasonal differences
- **q** is the number of lagged forecast errors in the equation
- **P** is the number of seasonal autoregressive terms
- **D** is the number of seasonal differences
- **Q** is the number of seasonal lagged forecast errors in the equation
- **s** is the seasonal period (cycle frequency using R terminology)

→ **In the next slides we will explain each single component of ARIMA models!**

ARIMA Models: Autoregressive part (AR)

In a **multiple regression model**, we predict the target variable Y using a linear combination of independent variables (predictors)→ In an **autoregression model**, we forecast the variable of interest using a linear combination of past values of the variable itself

The term autoregression indicates that it is a regression of the variable against itself

- An **Autoregressive model of order $\boldsymbol{p}$**, denoted $AR(p)$ model, can be written as

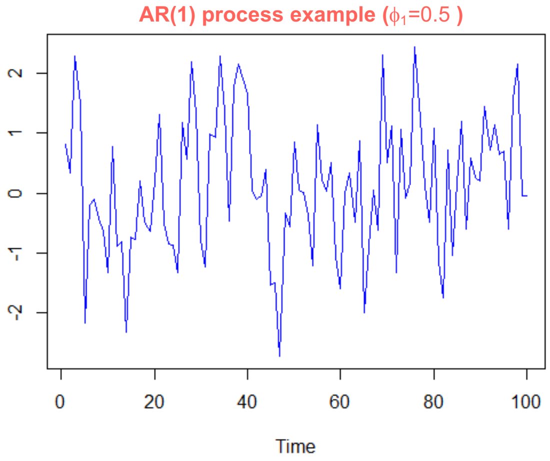
$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t$$

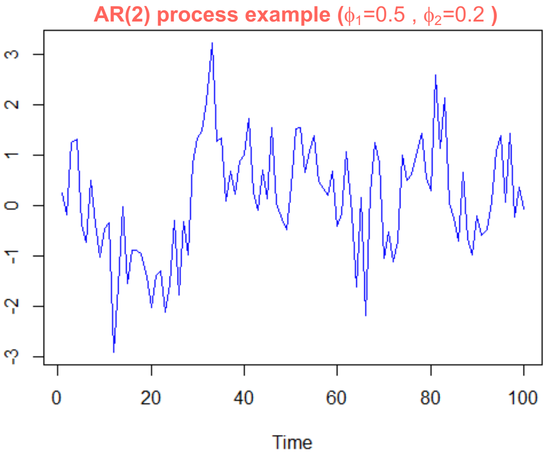
Where:

- y_t = dependent variable
- $y_{t-1}, y_{t-2}, \dots, y_{t-p}$= independent variables (i.e. lagged values of y_t as predictors)
- ϕ_1, ϕ_2, …, ϕ_p = regression coefficients
- ε_t= error term (must be white noise)

ARIMA Models: Autoregressive part (AR)

Autoregressive simulated process examples:





Consider that, in case of **AR(1)** model:

- When $\phi_1 = 0$, y_t is a white noise
- When $\phi_1 = 1$ and $c = 0$, y_t is a random walk
- In order to have a stationary series the following condition must be true: $-1 < \phi_1 < 1$

ARIMA Models: Moving Average part (MA)

Rather than use past values of the forecast variable in a regression, a Moving Average model uses **past forecast errors** in a regression-like model

In general, a moving average process of order q, MA (q), is defined as:

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q}$$

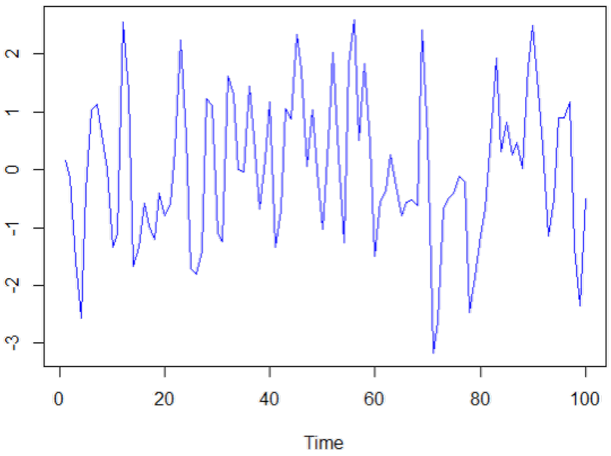
The lagged values of ε_t are not actually observed, so it is not a standard regression

Moving average models should not be confused with **moving average smoothing** (the process used in classical decomposition in order to obtain the trend component)→ A **moving average model** is used for forecasting future values while moving average smoothing is used for estimating the trend-cycle of past values

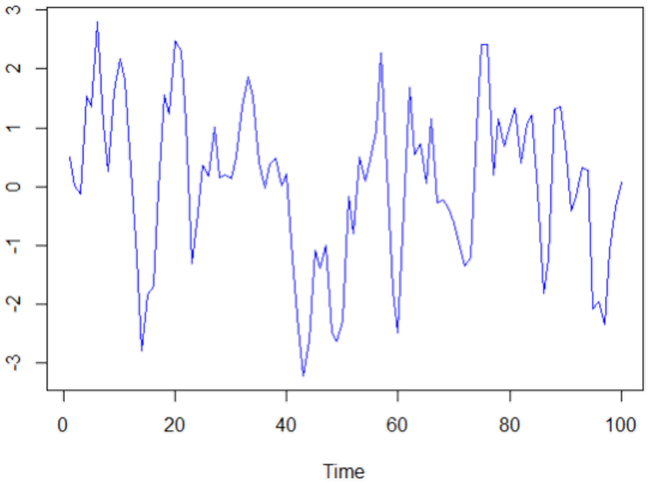
ARIMA Models: Moving Average part (MA)

Moving Average simulated process examples:

MA(1) process example (θ_1=0.7)



MA(2) process example (θ_1=0.8 , θ_2=0.5)



- Looking just the time plot it’s hard to distinguish between an AR process and a MA process!

ARIMA Models: ARMA and ARIMA

If we combine autoregression and a moving average model,
we obtain an **ARMA(p,q)** model:

$$y_t = c \underbrace{+\phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p}}_{\text{Autoregressive component of order p}} \underbrace{+ \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q}}_{\text{Moving Average component of order q}} + \varepsilon_t$$

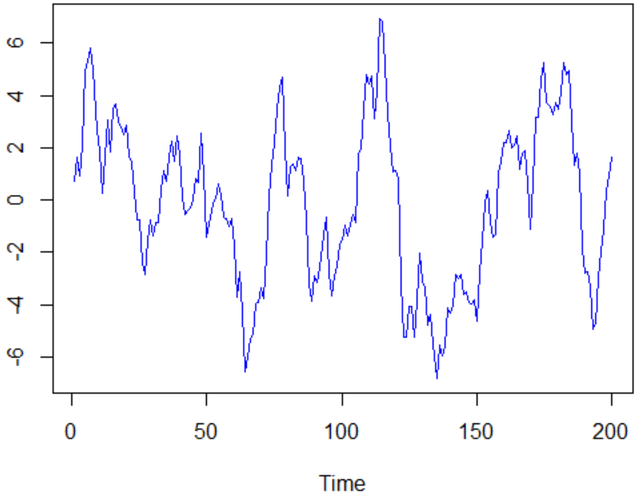
To use an ARMA model, the series must be **STATIONARY**!

- If the series is NOT stationary, before estimating and ARMA model, we need to apply one or more differences in order to make the series stationary: this is the integration process, called ***I(d)***, where d= number of differences needed to get stationarity
- If we model *the integrated* series using an ARMA model, we get an **ARIMA (p,d,q)** model where p=order of the autoregressive part; d=order of integration; q= order of the moving average part

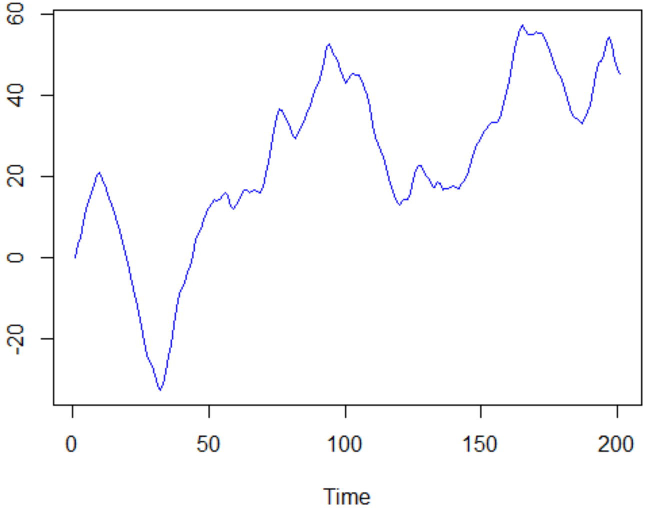
ARIMA Models: ARMA and ARIMA

ARIMA simulated process examples

ARMA(2,1) process example, equal to ARIMA(2,0,1) (ϕ_1=0.5, ϕ_2=0.4, θ_1=0.8)



ARIMA(2,1,1) process example (ϕ_1=0.5, ϕ_2=0.4, θ_1=0.8)



ARIMA Models: Model identification

General rules for model indentification based on ACF and PACF plots:

The data may follow an $\boldsymbol{ARIMA(p, d, 0)}$ model if the ACF and PACF plots of the differenced data show the following patterns:

- the ACF is exponentially decaying or sinusoidal
- there is a significant spike at lags p in PACF, but none beyond lag p

The data may follow an $\boldsymbol{ARIMA(0, d, q)}$ model if the ACF and PACF plots of the differenced data show the following patterns:

- the PACF is exponentially decaying or sinusoidal
- there is a significant spike at lags q in ACF, but none beyond lag q

→ For a general $\boldsymbol{ARIMA(p, d, q)}$ model (with both **p** and **q > 1**) both ACF and PACF plots show exponential or sinusoidal decay and it's more difficult to understand the structure of the model

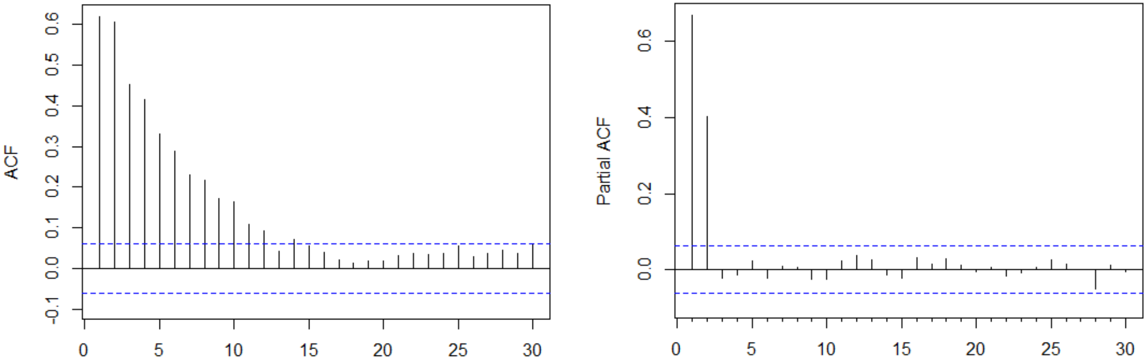
ARIMA Models: Model identification

Specifically:

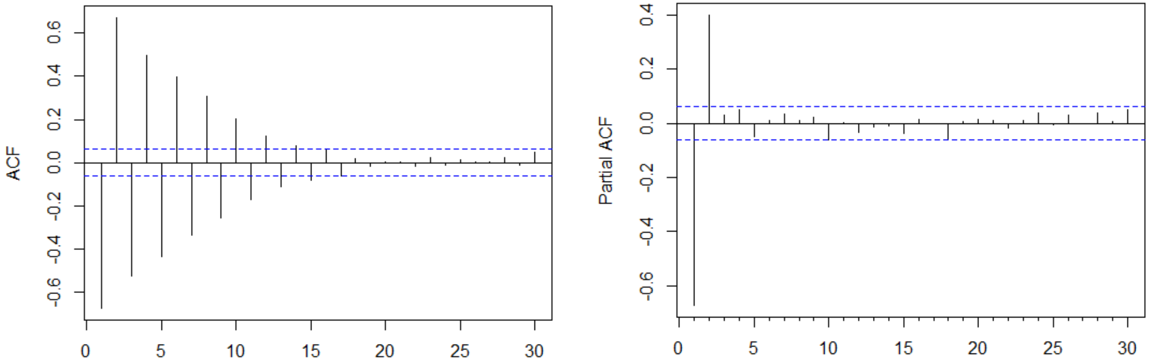
TIME SERIES	ACF	PACF
AR(1)	Exponential decay: From positive side or alternating (depending on the sign of the AR coefficient)	Peak at lag 1, then decays to zero: positive peak if the AR coefficient is positive, negative otherwise
AR(p)	Exponential decay or alternate sinusoidal decay	Peaks at lags 1 up to p
MA(1)	Peak at lag 1, then decays to zero: positive peak if the MA coefficient is positive, negative otherwise	Exponential decay: From positive side or alternating (depending on the sign of the MA coefficient)
MA(q)	Peaks at lags 1 up to q	Exponential decay or alternate sinusoidal decay

ARIMA Models: Model identification

$AR(2)$: Φ1>0, Φ2>0

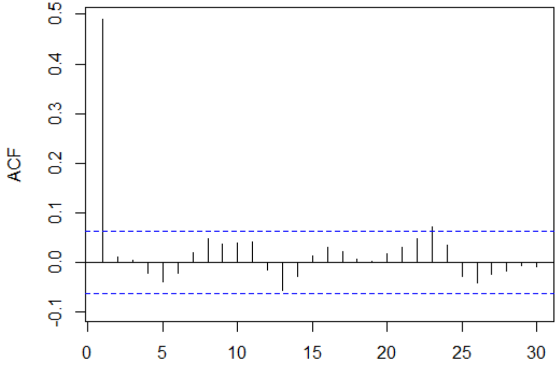


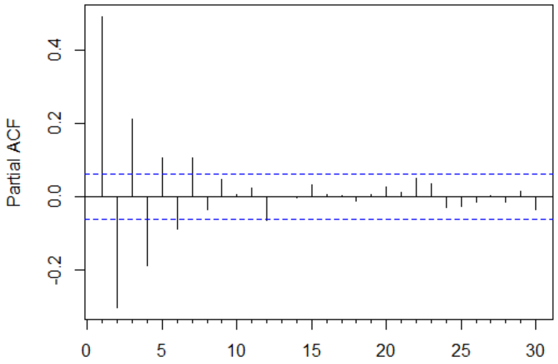
$AR(2)$: Φ1<0, Φ2>0



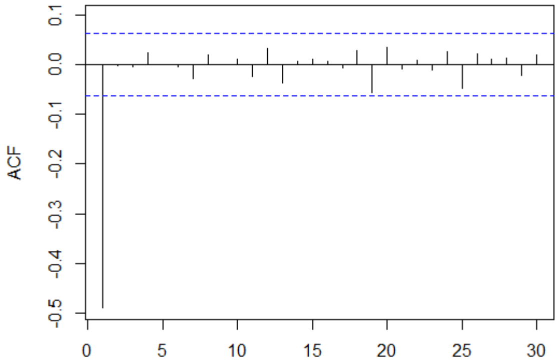
ARIMA Models: Model identification

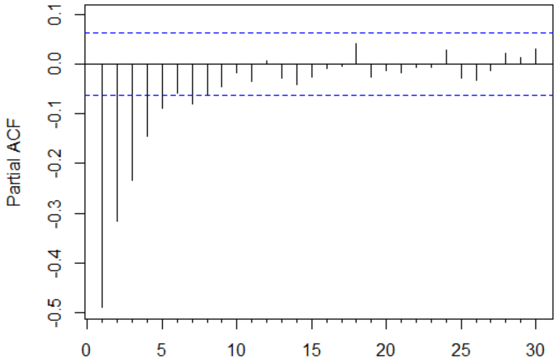
$MA(1)$: θ1>0





$MA(1)$: θ1<0

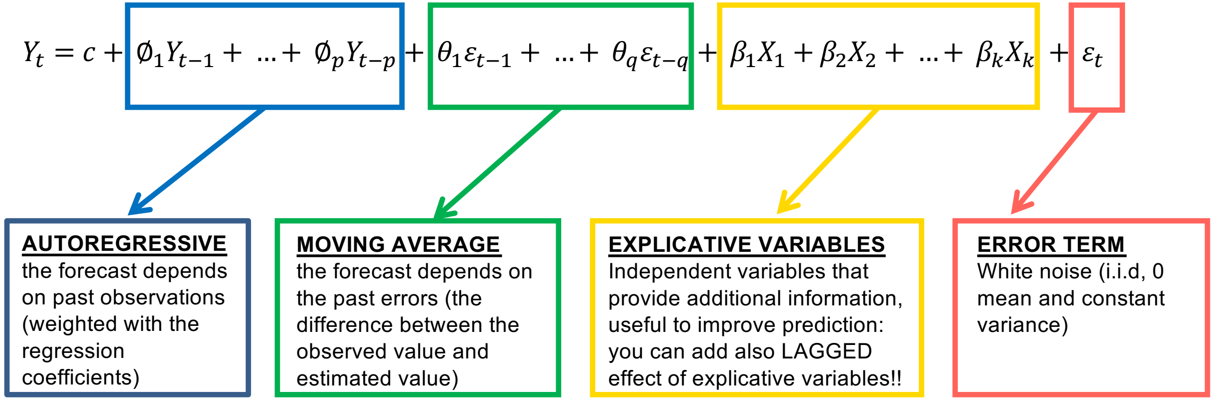




ARIMAX Models: Adding explicative variables

A **special case** of ARIMA models allows you to generate forecasts that depend on both the historical data of the target time series (Y) and on other exogenous variables (X_k)→ we call them **ARIMAX models**

- This is not possible with other classical time series analysis techniques (e.g. ETS), where the prediction depends only on past observations of the series itself
- The advantage of ARIMAX models, therefore consists in the possibility to **include additional explanatory variables** in addition to the target dependent variable lags



$$Y_t = c + \boxed{\emptyset_1 Y_{t-1} + \dots + \emptyset_p Y_{t-p}} + \boxed{\theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q}} + \boxed{\beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k} + \boxed{\varepsilon_t}$$

AUTOREGRESSIVE
the forecast depends on past observations (weighted with the regression coefficients)

MOVING AVERAGE
the forecast depends on the past errors (the difference between the observed value and estimated value)

EXPLICATIVE VARIABLES
Independent variables that provide additional information, useful to improve prediction: you can add also LAGGED effect of explicative variables!!

ERROR TERM
White noise (i.i.d, 0 mean and constant variance)

ARIMA Models: Seasonal ARIMA

A seasonal ARIMA model is formed by including **additional seasonal terms in the ARIMA models** we have seen so far

$$ARIMA\underbrace{(p,d,q)}_{\substack{\uparrow \\ \left(\begin{array}{l}\text{Non-seasonal part}\\ \text{of the model}\end{array}\right)}}\ \underbrace{(P,D,Q)}_{\substack{\uparrow \\ \left(\begin{array}{l}\text{Seasonal part}\\ \text{of the model}\end{array}\right)}}s$$

where s = number of periods per season (i.e. the frequency of seasonal cycle)

We use uppercase notation for the seasonal parts of the model, and lowercase notation for the non-seasonal parts of the model

→ As usual, d / D are the number of differences/seasonal differences necessary to make the series stationary

ARIMA Models: Seasonal ARIMA identification

The seasonal part of an AR or MA model will be seen in the seasonal lags of the PACF and ACF

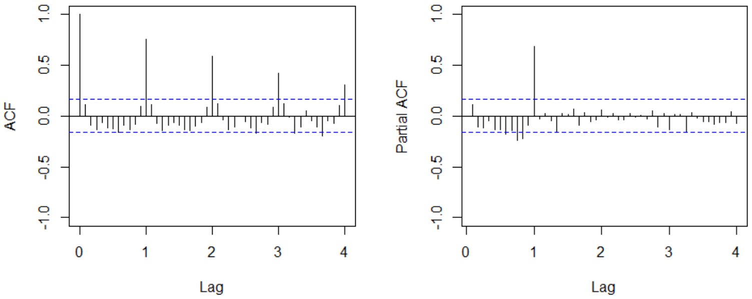
For example, an $ARIMA(0,0,0)(0,0,1)_{12}$ model will show:

- A spike at lag 12 in the ACF but no other significant spikes
- The PACF will show exponential decay in the seasonal lags; that is, at lags 12, 24, 36, …

Similarly, an $ARIMA(0,0,0)(1,0,0)_{12}$ model will show:

- Exponential decay in the seasonal lags of the ACF
- A single significant spike at lag 12 in the PACF

Example of $ARIMA(0,0,0)(1,0,0)_{12}$ process



ARIMA Models: estimation and AIC

Parameters estimation

In order to estimate an ARIMA model, normally it's used the **Maximum Likelihood Estimation (MLE)**

This technique **finds the values of the parameters which maximize the probability of obtaining the data that we have observed** → For *given values* of $(\boldsymbol{p}, \boldsymbol{d}, \boldsymbol{q})$ $(\boldsymbol{P}, \boldsymbol{D}, \boldsymbol{Q})$ (i.e. model order) the algorithm will try to **maximize the log likelihood** when finding parameter estimates

ARIMA model order

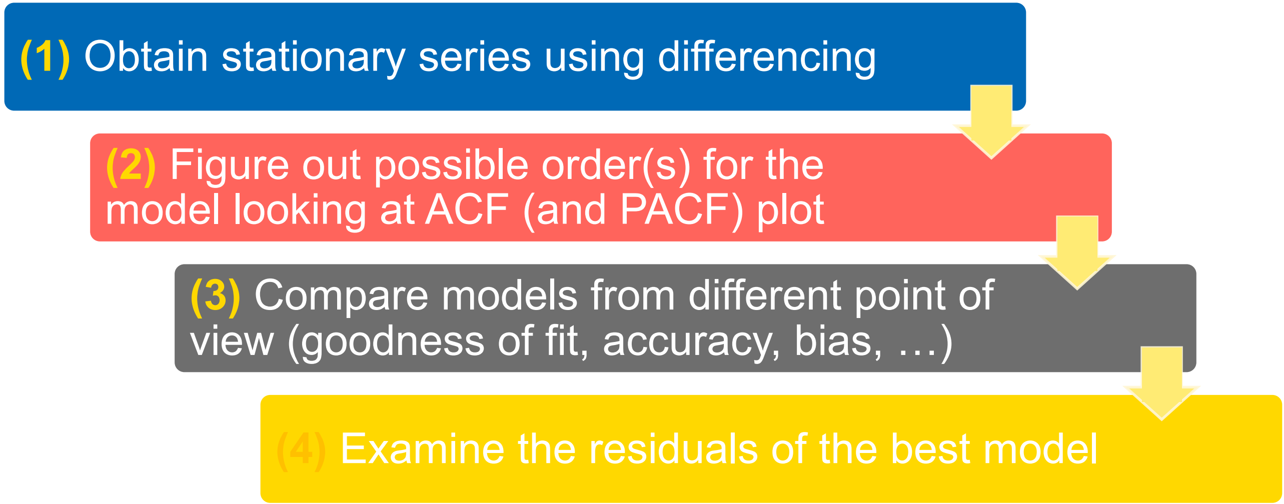
A commonly used criteria to compare different ARIMA models (i.e. with different values for $(\boldsymbol{p},\boldsymbol{q})$ $(\boldsymbol{P},\boldsymbol{Q})$ but fixed $\boldsymbol{d}$, $\boldsymbol{D}$) and to determine the optimal ARIMA order, is the **Akaike Information Criterion (AIC)**

$$\mathbf{AIC} = -2\log\left(Likelihood\right) + 2(p)$$

- where *p* is the number of estimated parameters in the model
- AIC is a goodness of fit measure
- **The best ARIMA model is that with the lower AIC** → most of automatic model selection method (e.g *auto.arima* in R) uses the AIC for determining the optimal ARIMA model order

ARIMA Model selection criteria: Manual procedure (outline)

- After preliminary analysis (and time series transformations, if needed), follow these steps:



(1) Obtain stationary series using differencing

(2) Figure out possible order(s) for the model looking at ACF (and PACF) plot

(3) Compare models from different point of view (goodness of fit, accuracy, bias, …)

(4) Examine the residuals of the best model

ARIMA Model selection criteria: Manual procedure (details)

After preliminary analysis (and time series transformations, if needed), follow these steps:

1. If the series is not stationary, **use differencing (simple and/or seasonal) in order to obtain a stationary series** → together with graphical analysis, there are specific statistical tests (e.g. ADF) useful to understand if the series is stationary
2. Examine the **ACF/PACF of the stationary series and try to obtain an idea about residual structure of correlation** → Is an AR(p) / MA(q) model appropriate or you need more complex model? Do you need to model the seasonality using seasonal autoregressive lags? **It is frequent that you need to consider more candidate models to test**
3. Try your chosen model(s)*, and **use different metrics to compare the performance**:
   - Compare goodness of fit using AIC
   - Compare accuracy using measures like MAPE (in-sample and out-of-sample!)
   - Model complexity (simple is better!)
4. Finally, **check the residuals** from your chosen model by plotting the ACF of the residuals and doing some test on the residuals (e.g. Ljung-Box test of autocorrelation) → **they must be white noise when the model is ok!**

* Always consider slight variations of models selected in point 2: e.g. **vary one or both p and q from current model by 1**

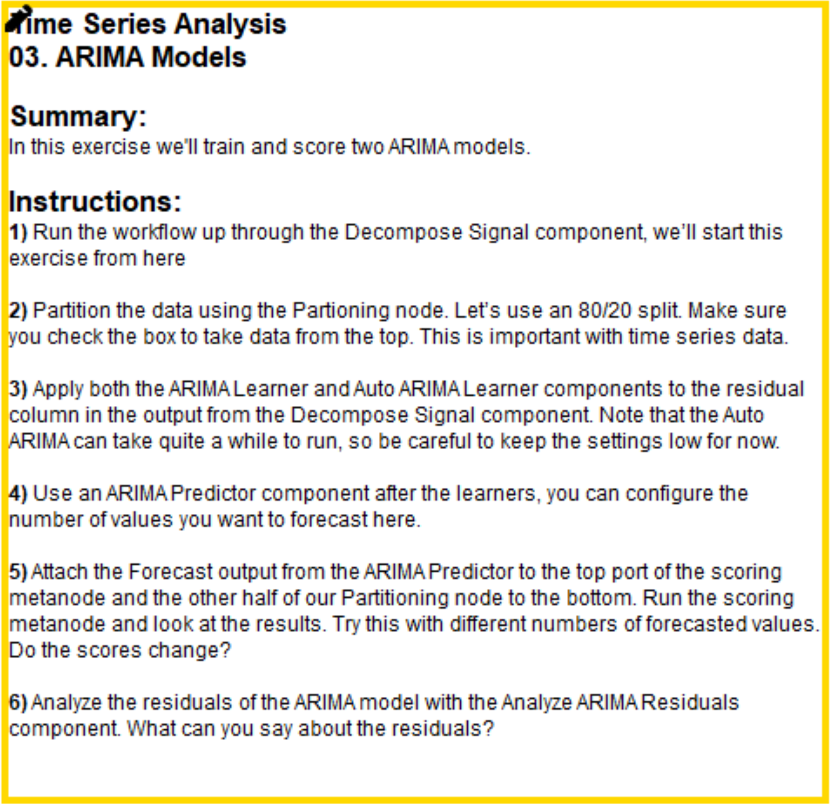
ARIMA Performance Comparison

- (2,1,1) vs (1,0,0) vs (0,1,0)

ARIMA(p,d,q)	R^2	AIC	MAPE	RMSE
ARIMA(2,1,1)	0.798	25,899	6.073	0.870
ARIMA(1,0,0)	0.808	25,405	5.466	0.871
ARIMA(0,1,0)	0.798	25,924	6.048	0.871

Exercise 3: ARIMA Models

- Train a model with both the ARIMA Learner and Auto ARIMA Learner.
- Generate a Forecast for each model using the ARIMA Predictor.
- Score your forecasts.
- Analyze ARIMA residuals.



Time Series Analysis
03. ARIMA Models

Summary:
In this exercise we'll train and score two ARIMA models.

Instructions:
1) Run the workflow up through the Decompose Signal component, we'll start this exercise from here

2) Partition the data using the Partioning node. Let's use an 80/20 split. Make sure you check the box to take data from the top. This is important with time series data.

3) Apply both the ARIMA Learner and Auto ARIMA Learner components to the residual column in the output from the Decompose Signal component. Note that the Auto ARIMA can take quite a while to run, so be careful to keep the settings low for now.

4) Use an ARIMA Predictor component after the learners, you can configure the number of values you want to forecast here.

5) Attach the Forecast output from the ARIMA Predictor to the top port of the scoring metanode and the other half of our Partitioning node to the bottom. Run the scoring metanode and look at the results. Try this with different numbers of forecasted values. Do the scores change?

6) Analyze the residuals of the ARIMA model with the Analyze ARIMA Residuals component. What can you say about the residuals?

Open for Innovation
KNIME

KNIME Time Series Analysis Course - Session 4

KNIME AG