Open for Innovation
KNIME

Time Series Analysis with the KNIME Analytics Platform

Maarit Widmann
Corey Weisinger

Meet the Speakers

Maarit Widmann is a Data Scientist and an educator at KNIME; the author behind the KNIME self-paced courses and a teacher in the KNIME courses. She is also a co-author of the From Modeling to Model Evaluation e-book and she publishes regularly in the KNIME blog. She holds a Master's degree in Data Science and a Bachelor's degree in Sociology.

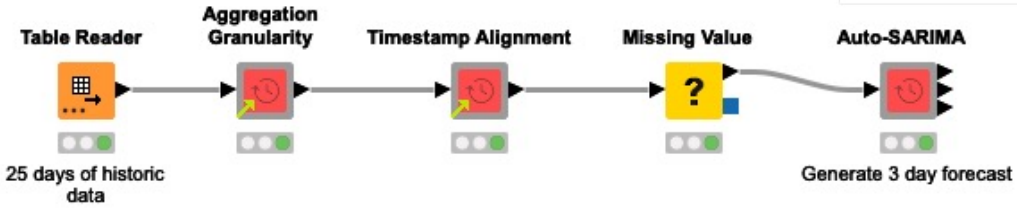
https://www.linkedin.com/in/maarit-widmann-02641a170/

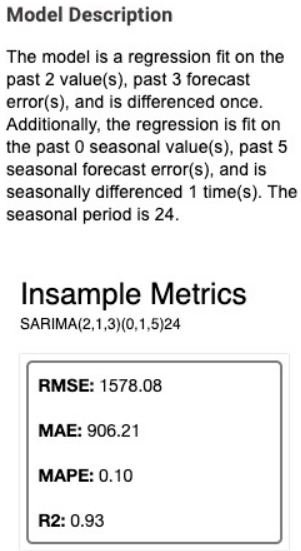
Corey Weisinger studied Mathematics at Michigan State University and works as a Data Scientist with KNIME where he focuses on Time Series Analysis, Forecasting, and Signal Analytics. He is the creator and instructor of the KNIME Time Series Analysis course, author of the e-book: Alteryx to KNIME, and creator of the KNIME Time Series Analysis components.

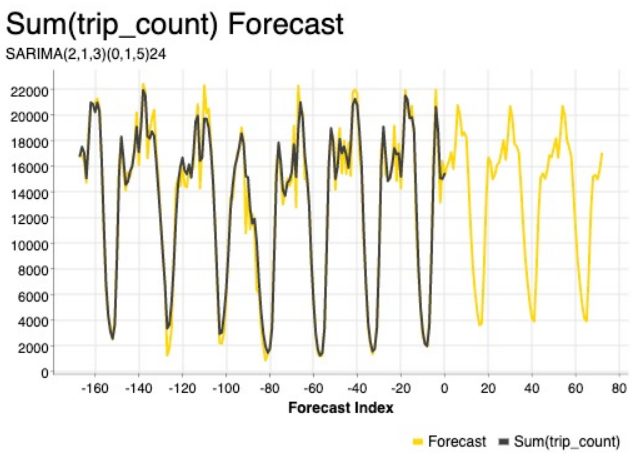
https://www.linkedin.com/in/corey-weisinger-709525121/

Screenshot of workflow

- https://kni.me/w/zA31U45GPJciXf7d







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 analysis:** 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)

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

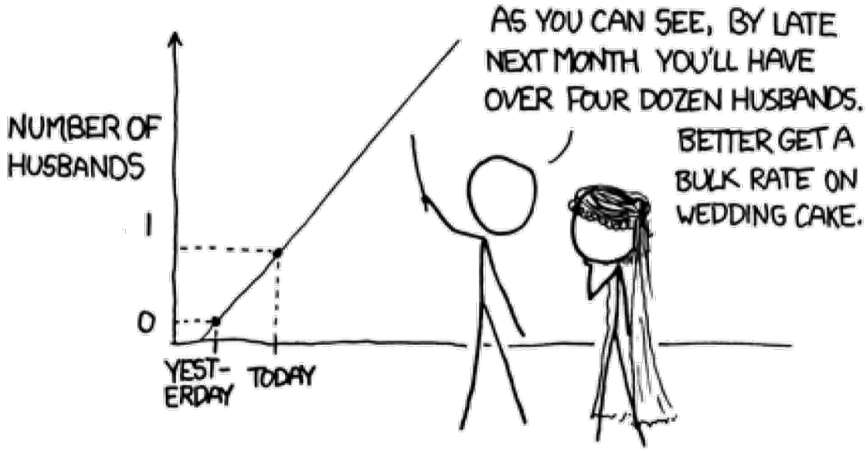
- One single object (product, country, sensor, ..) observed over multiple equally-spaced time periods
- *Examples:* quarterly Italian GDP of the last 10 years, weekly supermarket sales of the previous year, yesterday's hourly temperature measurements, …

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..



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

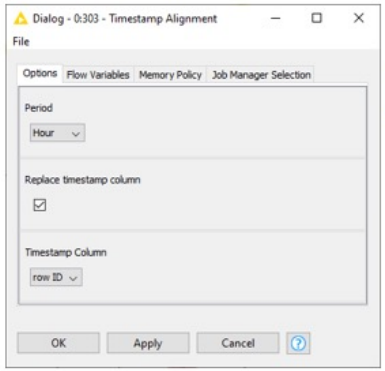
GOAL: Reduce uncertainty and improve the accuracy of our forecasts

Component: Timestamp Alignment

- Acquire continuously spaced data
- In today's example we verify a record exists for every hour
- Otherwise create a missing value

cluster_26	Row ID
3.78	2010-03-24T22:00
3.85	2010-03-24T23:00
3.83	2010-03-25T01:00
3.95	2010-03-25T02:00
3.83	2010-03-25T03:00
3.75	2010-03-25T04:00

Input: Time series to check for uniform sampling



Output: Time series with skipped skipped sampling times

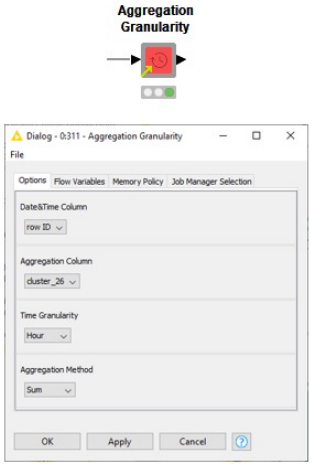
cluster_26	Row ID
3.78	2010-03-24T22:00
3.85	2010-03-24T23:00
?	2010-03-25T00:00
3.83	2010-03-25T01:00
3.95	2010-03-25T02:00
3.83	2010-03-25T03:00
3.75	2010-03-25T04:00

Component: Aggregation Granularity

- Extract granularities (year, month, hour, etc.) from a timestamp and aggregate (sum, average, mode, etc.) data at the selected granularity
- In today's example we calculate the total energy consumption by hour, day, and month

Input: Time series to aggregate

cluster_26	Row ID
3.25	2009-07-15T00:00
3.20	2009-07-15T01:00
3.04	2009-07-15T02:00
3.03	2009-07-15T03:00
3.01	2009-07-15T04:00
2.94	2009-07-15T05:00
...	...
3.18	2009-07-19T19:00
3.11	2009-07-19T20:00
3.01	2009-07-19T21:00
3.26	2009-07-19T22:00
3.20	2009-07-19T23:00



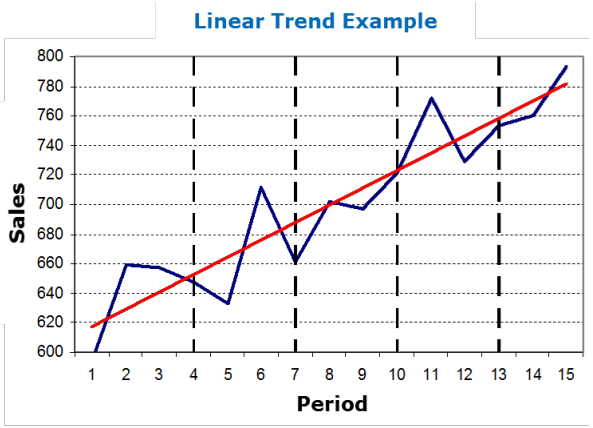
Output: Aggregated time series

Aggregated Timestamp	Sum (cluster_26)	Granularity
2009-07-15	185.82	Day
2009-07-16	182.49	Day
2009-07-17	177.87	Day
2009-07-18	116.48	Day
2009-07-19	83.95	Day

Time Series Properties: Main Elements

- TREND

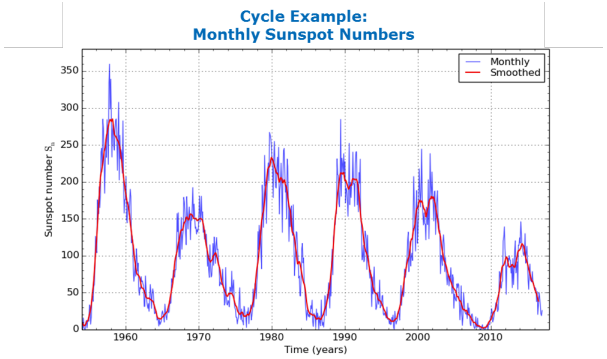
  The general direction in which the series is running during a long period
  A **TREND** exists when there is a long-term increase or decrease in the data.
  It does not have to be necessarily linear (could be exponential or others functional form).



- 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

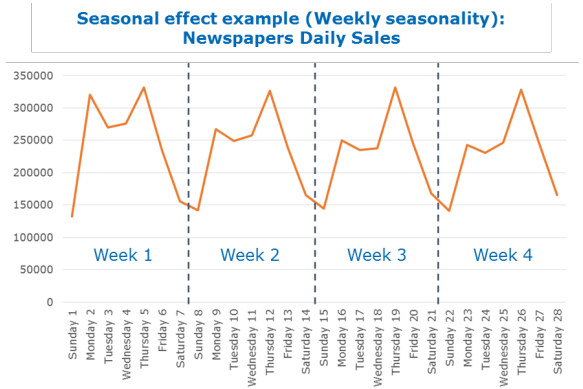


Time Series Properties: Main Elements

- **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.

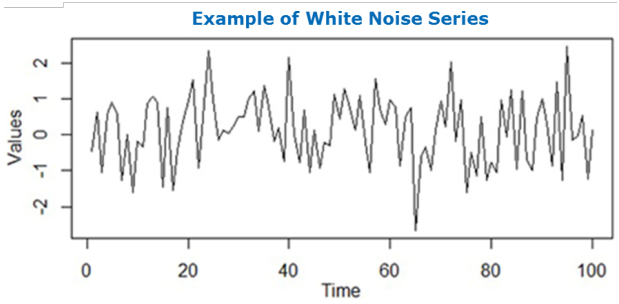


- **RESIDUAL**

  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



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)

- **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

Naïve Prediction

- Predict values by the most recent known value

$$\hat{y}_{T+h|T} = y_T,$$

where y_T is the most recent known value and *h=1,2,3*

- Best predictor for true random walk data

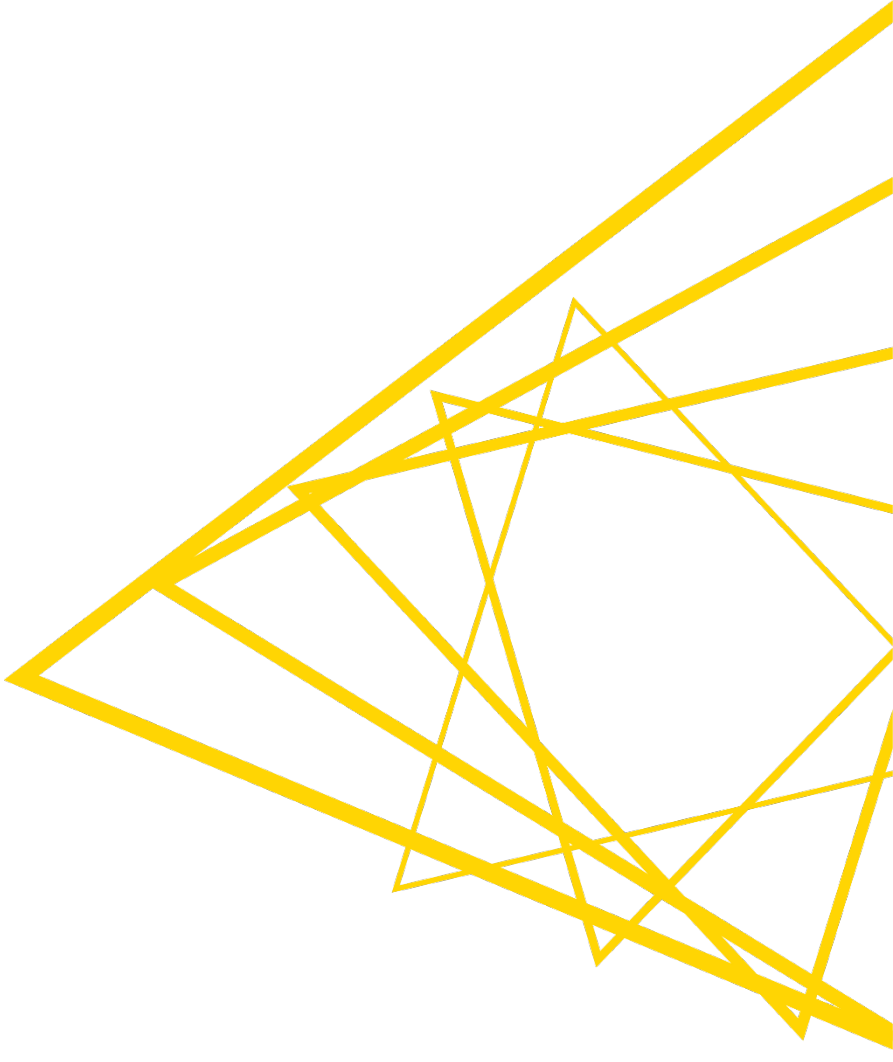
Which model?

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**

ARIMA Models

ARIMA(p,d,q)



Goal of this Section

1. Introduction to ARIMA
2. (S)ARIMA Models
3. (S)ARIMA Model selection

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 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
- $y_{t-1}, y_{t-2}, \ldots, y_{t-p}$= 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: 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: ARMA and ARIMA

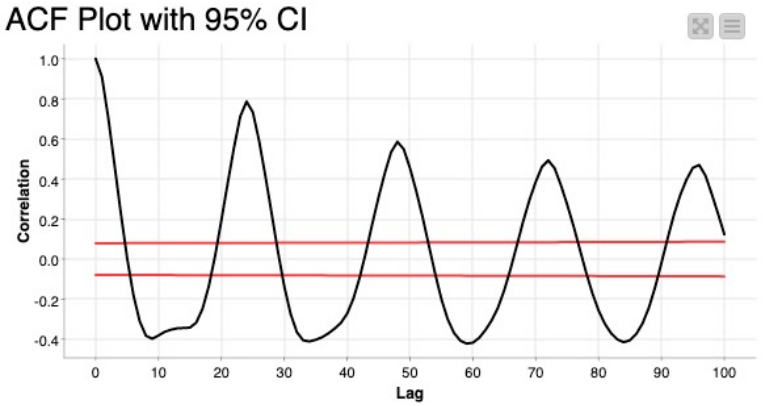
If we combine autoregression and a moving average model, we obtain an **ARMA(p,q)** model:

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

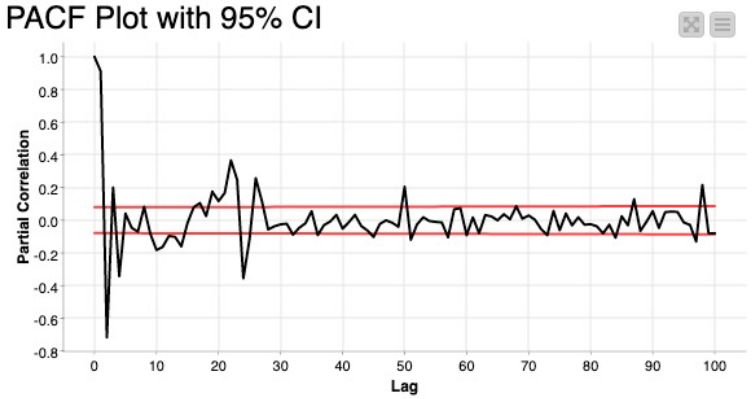
To use an ARMA model, the series must be **STATIONARY**!

- 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

ACF and PACF



- Auto Correlation Function
- Correlation of time series vs lagged copies
- Repeated spikes indicate seasonality
- Used to find q and Q



- Partial Auto Correlation Function
- Removes effect of serial correlation from ACF
- Should decay to zero
- Used to find p and P

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

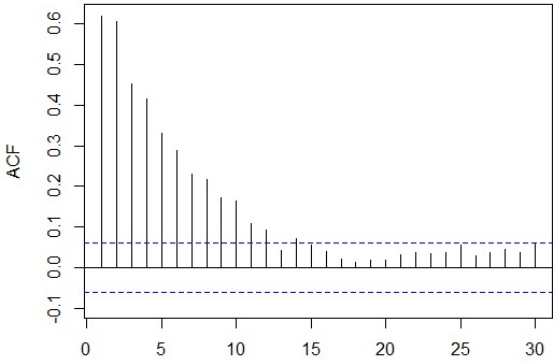
ARIMA Models: Model identification

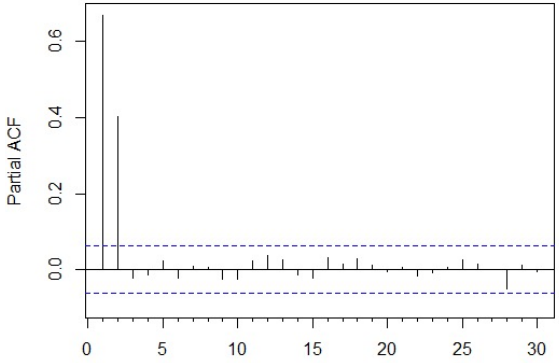
Specifically:

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

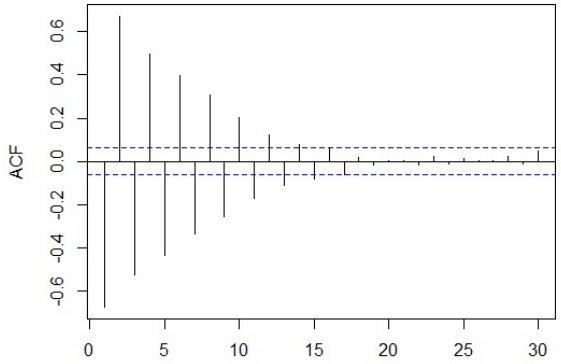
ARIMA Models: Model identification

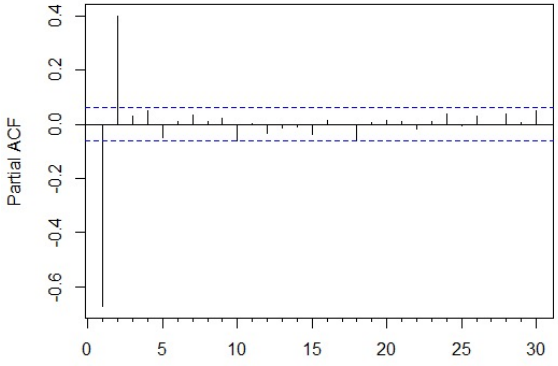
$AR(2)$: Φ1>0, Φ2>0





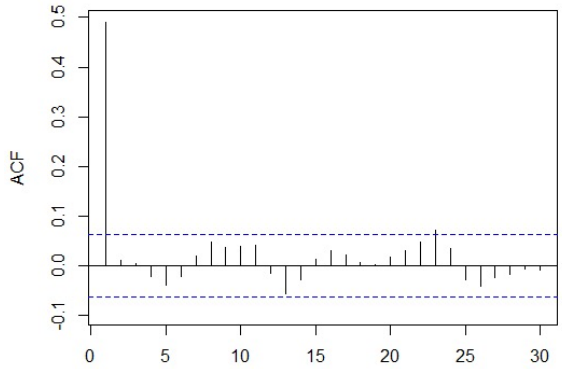
$AR(2)$: Φ1<0, Φ2>0

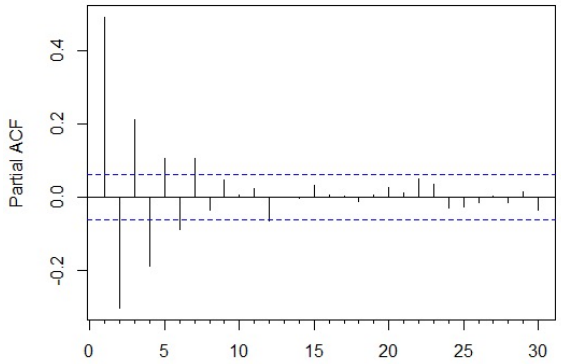




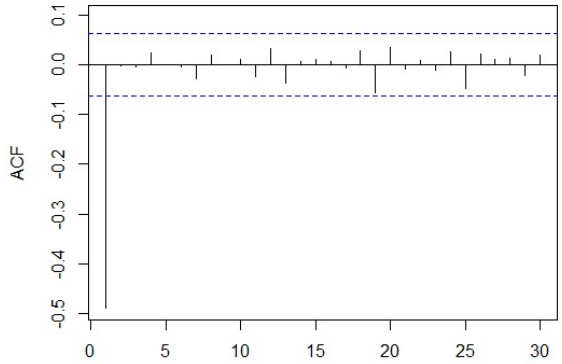
ARIMA Models: Model identification

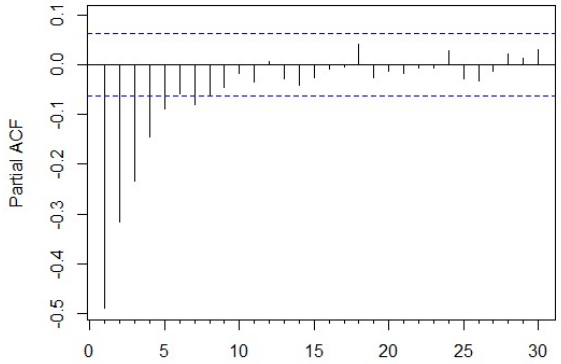
$MA(1)$: θ1>0





$MA(1)$: θ1<0





ARIMA Models: Seasonal ARIMA

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

$$\underbrace{ARIMA(p,d,q)}_{\text{Non-seasonal part of the model}}\ \underbrace{(P,D,Q)s}_{\text{Seasonal part of the model}}$$

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

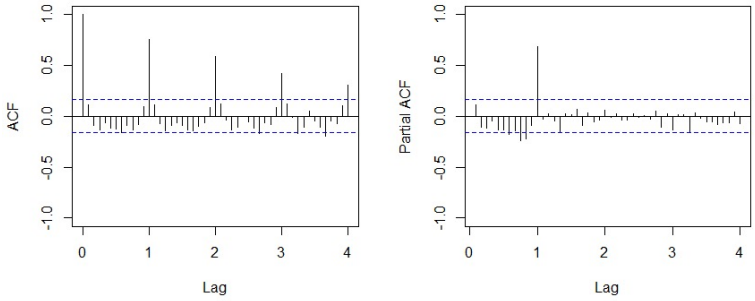
For example, an $ARIMA(0,0,0)(0,0,1)_{12}$ model will show:

- 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

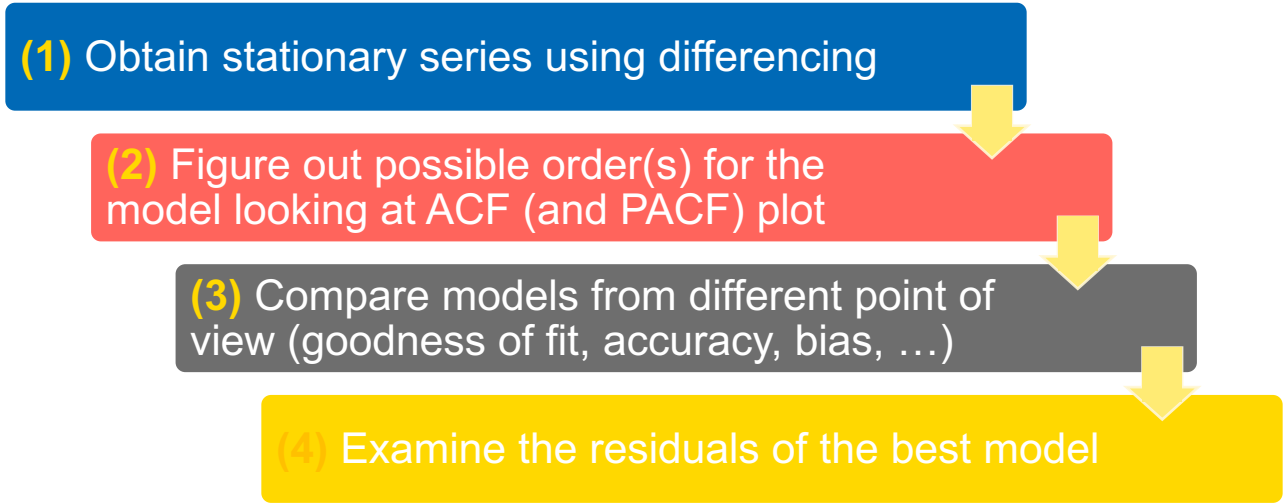
Example of $ARIMA(0,0,0)(1,0,0)_{12}$ process



ARIMA Model selection criteria

Manual procedure (outline)

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



ARIMA Model selection criteria

Manual procedure (details)

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

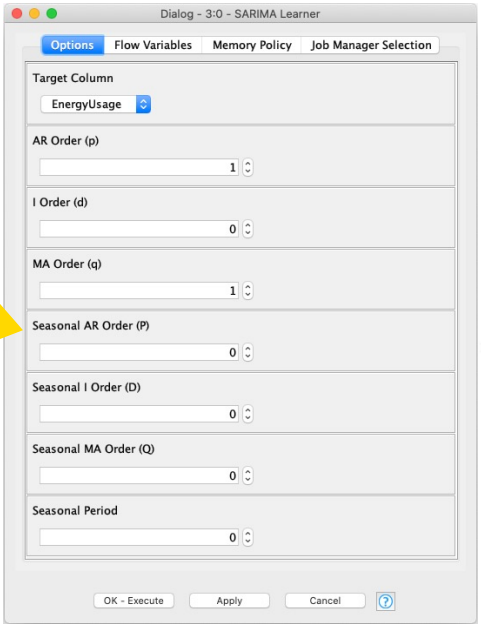
1. If the series is not stationary, **use differencing (simple and/or seasonal) in order to obtain a stationary series** → together with graphical analysis, there are specific statistical tests (e.g. ADF) useful to understand if the series is stationary
2. Examine the **ACF/PACF of the stationary series 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!**

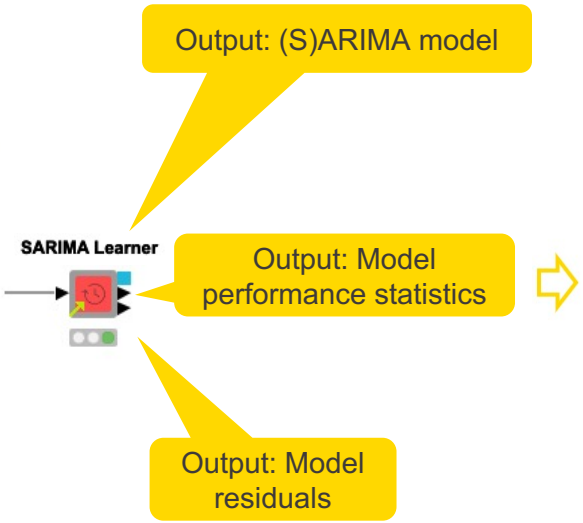
* Always consider slight variations of models selected in point 2: e.g. **vary one or both p and q from current model by 1**

Component: SARIMA Learner

- Learns (S)ARIMA model of specified orders on selected target column.

Input: Time series, specified orders

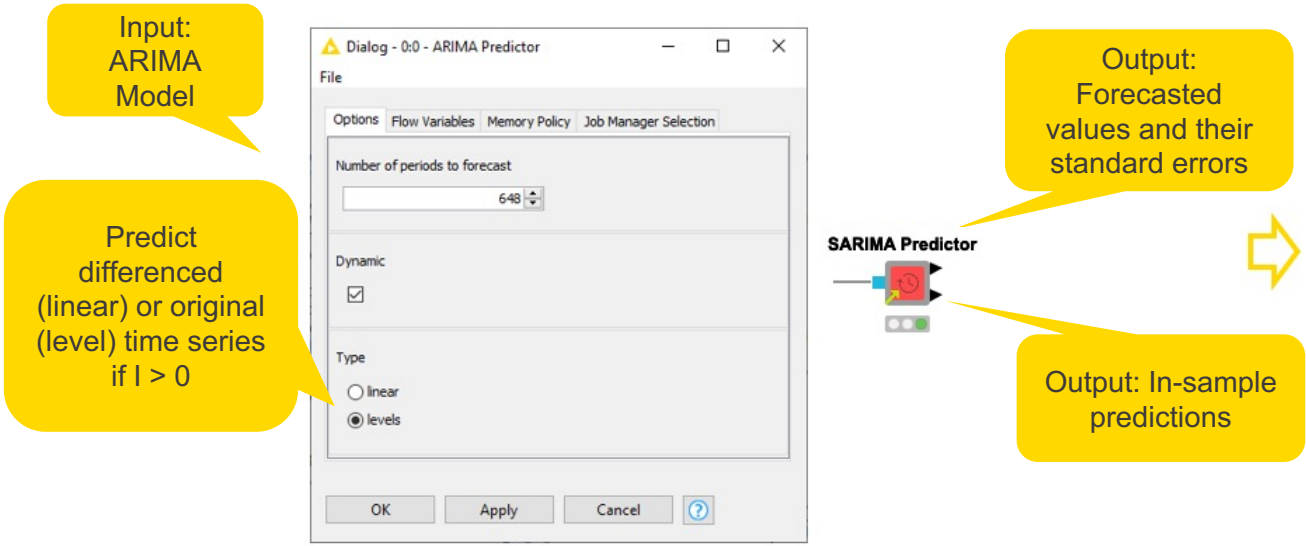




Row ID	value
RMSE	0.85
MAE	0.48
MAPE	5.47
R2	0.81
Log Likelihood	-12699.09
AIC	25406
BIC	25435
AR.L1.D.Irregular Component	0.90
AR.L1.D.Irregular Component Std Error	0.005
MA.L1.D.Irregular Component	0.008
MA.L1.D.Irregular Component Std Error	0.01

Component: SARIMA Predictor

- Generates number of forecasts set in configuration and in-sample predictions based on range used in training
- Checking the dynamic box will use predicted values for in-sample prediction

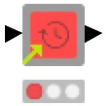


out-of-sample forecast	standard error
0.415	0.849
0.373	1.145
0.334	1.337
0.300	1.473
0.269	1.574
0.242	1.651
...	...

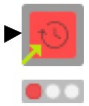
in-sample forecast
0.017
0.015
0.014
0.013
0.011
...

List of other Time Series Components

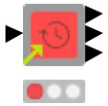
Aggregation Granularity



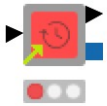
Analyze ARIMA Residuals



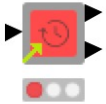
Auto-SARIMA



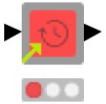
Decompose Signal



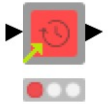
Discrete Wavelet Transform (DWT)



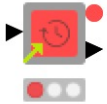
Fast Fourier Transform (FFT)



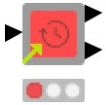
Forecast Horizon



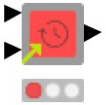
Inspect Seasonality



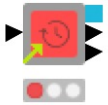
Remove Seasonality



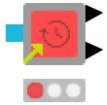
Return Seasonality



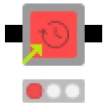
SARIMA Learner



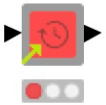
SARIMA Predictor



Spark Lag Column



Timestamp Alignment

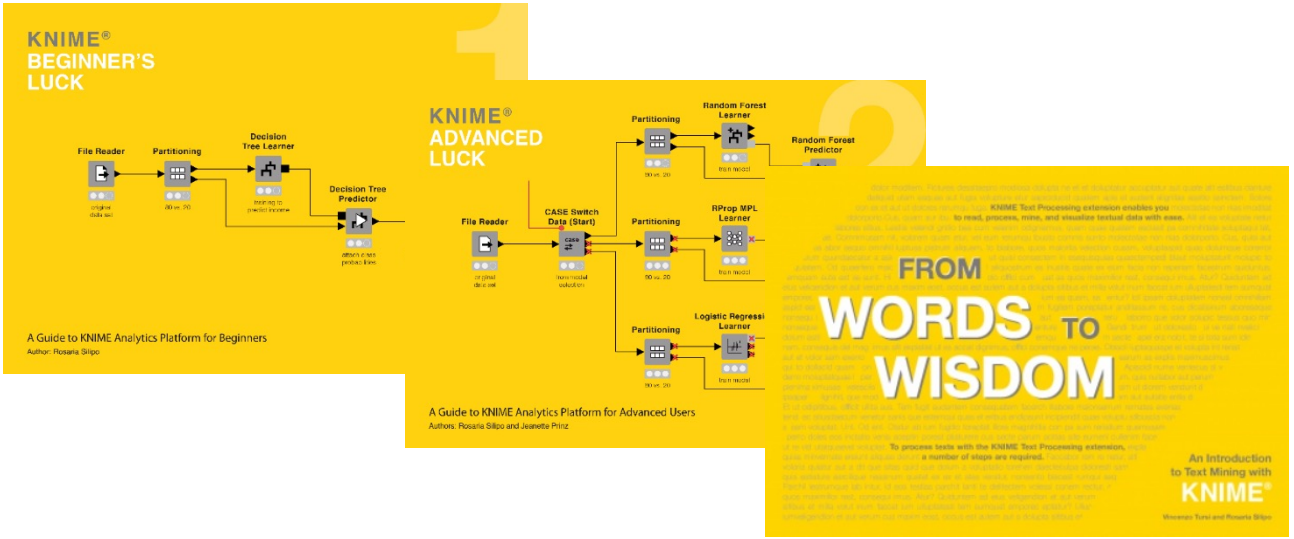


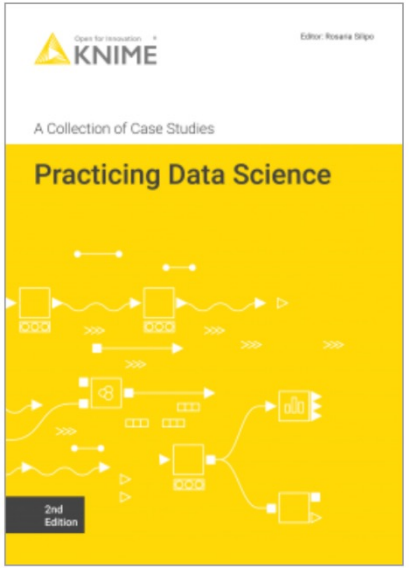
KNIME Books

Free books downloadable from **KNIME Press**

https://www.knime.com/knimepress

with code: DSDOJO-0522





Open for Innovation KNIME

References

- Hyndman, Rob J., and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.
- Gilliland, Michael, Len Tashman, and Udo Sglavo. *Business forecasting: Practical problems and solutions*. John Wiley & Sons, 2016.
- Franses, Philip Hans, and Philip Hans BF Franses. *Time series models for business and economic forecasting*. Cambridge university press, 1998.
- Chatfield, Chris, and Haipeng Xing. *The analysis of time series: an introduction with R*. CRC press, 2019.

Open for Innovation
KNIME

Thank You!

Corey.weisinger@knime.com
Maarit.widmann@knime.com