Time Series Decomposition: Data Processing for Forecasting

Corey Weisinger

September 21st, 2023

Open for Innovation KNIME

What is a Time Series?

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 <u>at a particular point of time</u>
- *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, …

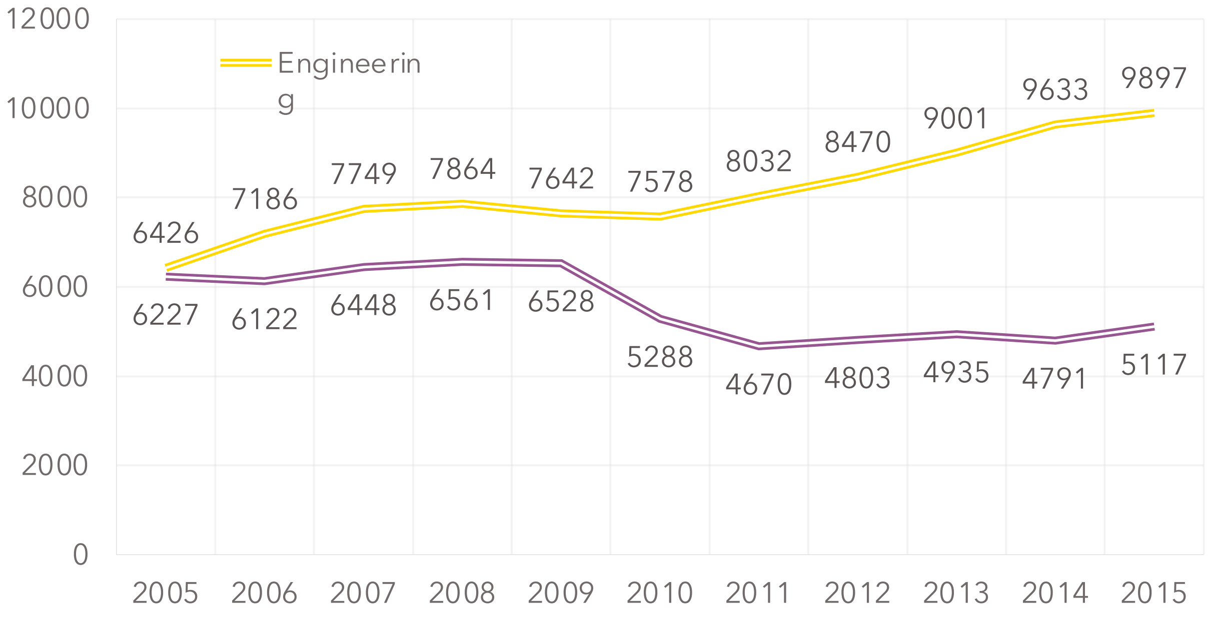
2. Time Series Data

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

Example

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

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 $Y_1, Y_2, Y_3, \ldots Y_t, \ldots, Y_T$ since data are usually collected at discrete points in time

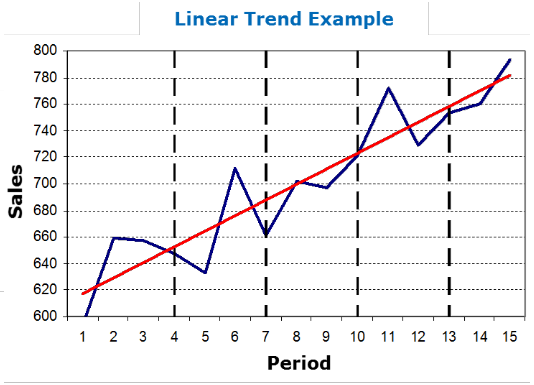
Observation at time t

- 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**

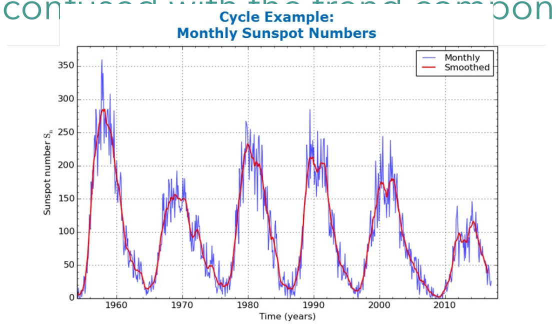
Time Series Decomposition

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

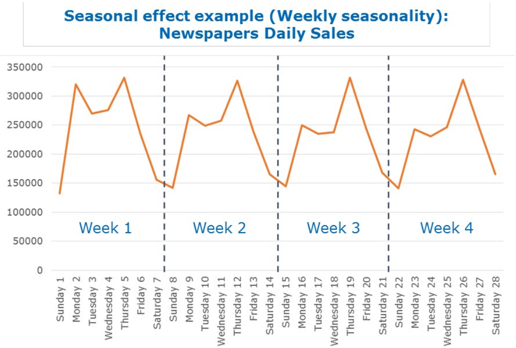


Time Series Properties: Main Elements

- **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.

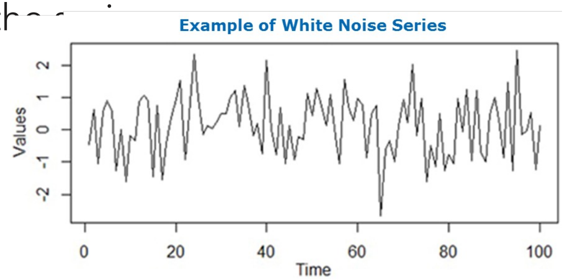


- **RESIDUAL**

  Whatever remains after the other components have been taken into account

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

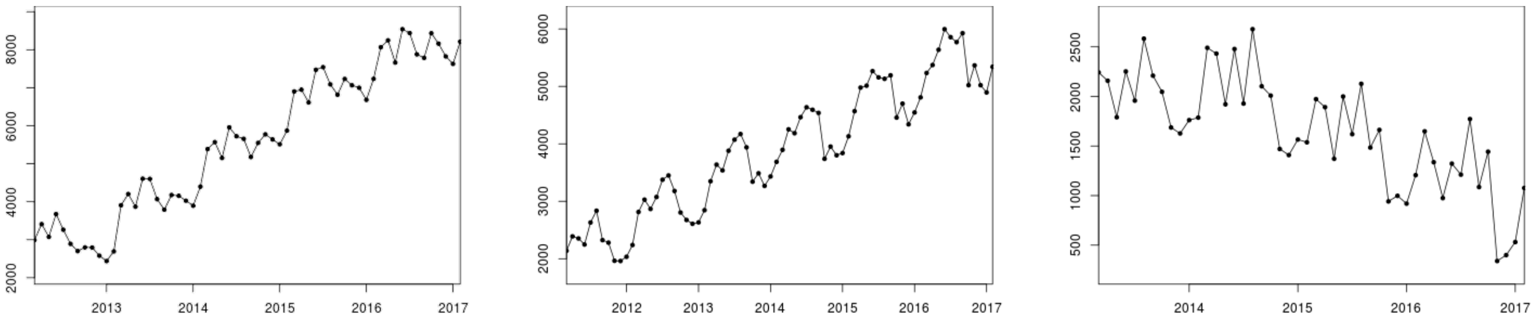
  Typically, it is assumed to be the sum of a set of random factors (e.g. a **white noise series**) not relevant for describing the dynamics of the series



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

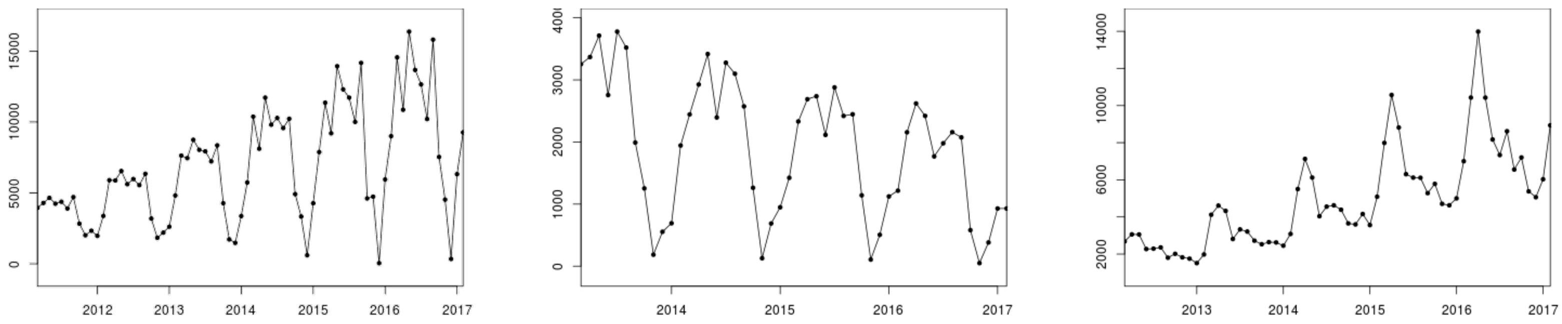
EXAMPLES OF ADDITIVE SEASONALITY



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

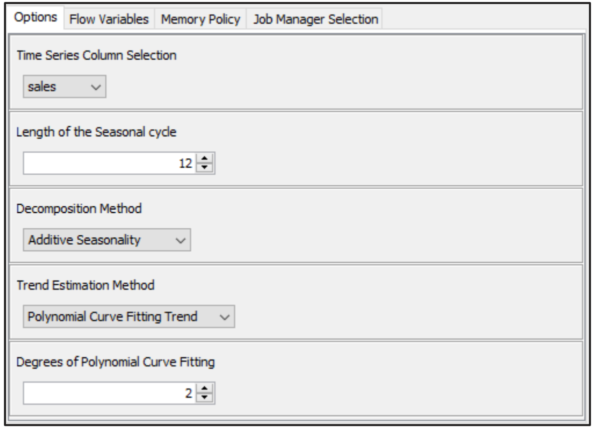


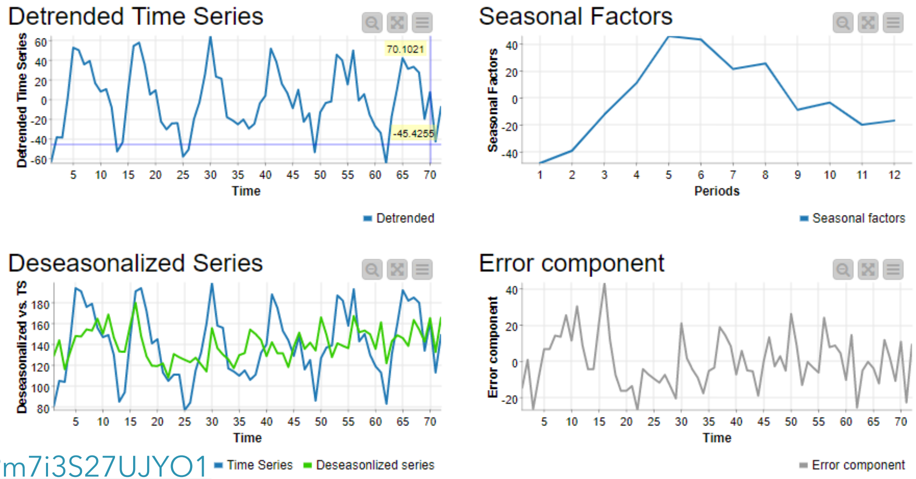
Component: Time Series Classical Decomposition

- Extract trend via non-linear curve fitting or centered moving average
- Extract additive or multiplicative seasonality

Input:
Signal to decompose

Output:
- Detrended Time Series
- Deseasonalized Time Series
- Seasonal Factors
- Fitted Time Series
- Error component





Component on the KNIME Hub: https://kni.me/c/-UFPm7i3S27UJYO1

Forecasting

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: 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
- $y_{t-1}, y_{t-2}, \ldots, y_{t-p}$ = independent variables (i.e. lagged values of y_t as predictors)
- $\phi_1, \phi_2, \ldots, \phi_p$ = 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:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

Autoregressive component of order p

Moving Average component of order q

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: selection criteria

- 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

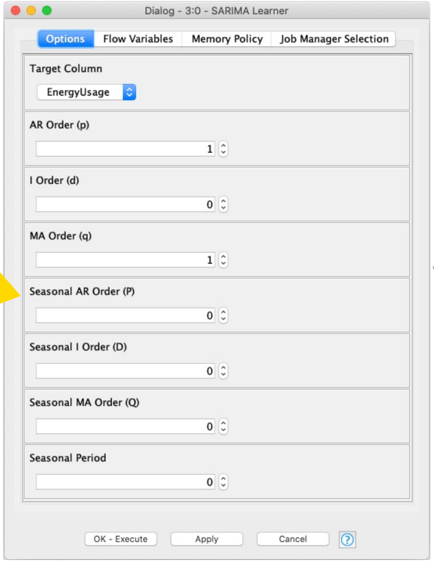
(3) Compare models from different point of view (goodness of fit, accuracy, bias, …)

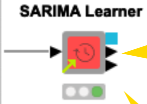
(4) Examine the residuals of the best model

Component: SARIMA Learner

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

Input: Time series, specified orders





Output: (S)ARIMA model

Output: Model performance statistics

Output: Model residuals

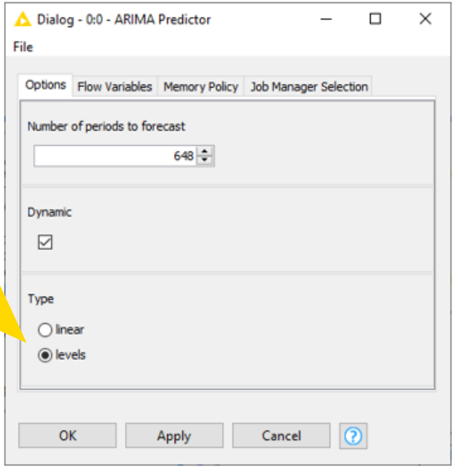
Row ID	value
RMSE	0.85
MAE	0.48
MAPE	5.47
R2	0.81
Log Likelihood	-12699.09
AIC	25406
BIC	25435
AR.L1.D.Irregular Component	0.90
AR.L1.D.Irregular Component Std Error	0.005
MA.L1.D.Irregular Component	0.008
MA.L1.D.Irregular Component Std Error	0.01

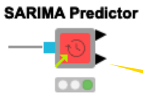
Component: SARIMA Predictor

- 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

Input: ARIMA Model

Predict differenced (linear) or original (level) time series if I > 0





Output: Forecasted values and their standard errors

Output: In-sample predictions

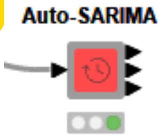
out-of-sample forecast	standard error
0.415	0.849
0.373	1.145
0.334	1.337
0.300	1.473
0.269	1.574
0.242	1.651
...	...

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

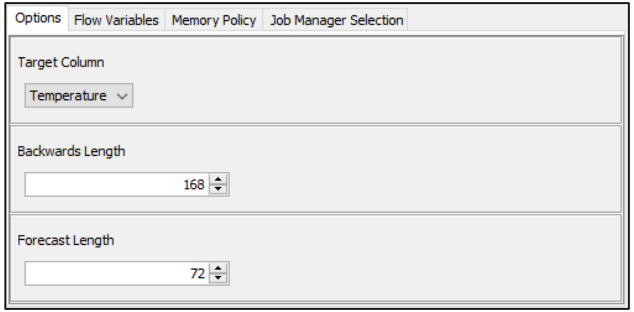
Component: Auto-SARIMA

- Creates combinations of (S)ARIMA Models using a heuristic approach
- Visualizes the in- and out-sample forecasts in its interactive view

Input:
Time series



Output:
- Out-sample forecasts
- In-sample forecasts
- Error metrics



Auto-SARIMA Summary

Model Description

The model is a regression fit on the past 2 value(s), past 3 forecast error(s), and is differenced once. Additionally, the regression is fit on the past 0 seasonal value(s), past 5 seasonal forecast error(s), and is seasonally differenced 1 time(s). The seasonal period is 24.

Insample Metrics

SARIMA(2,1,3)(0,1,5)24

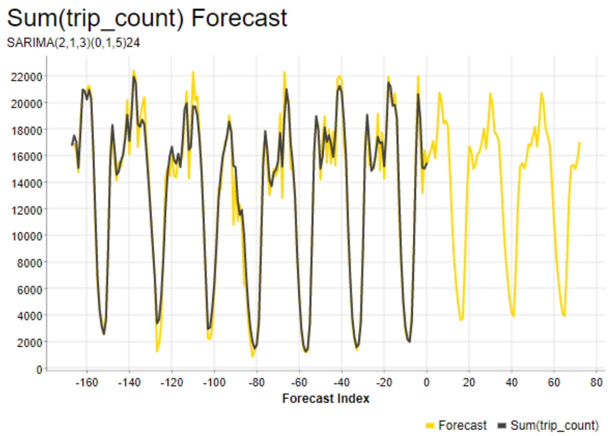
RMSE: 1578.08

MAE: 906.21

MAPE: 0.10

R2: 0.93

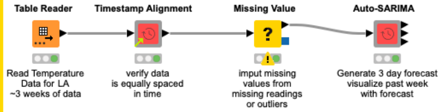
Showing 1 to 1 of 1 entries



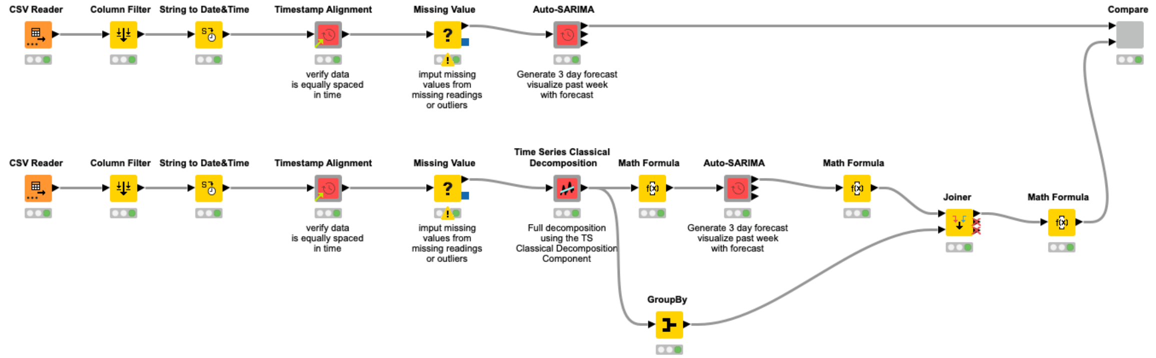
Deployment

Lets go to KNIME

Forecasting one seasonality directly with SARIMA

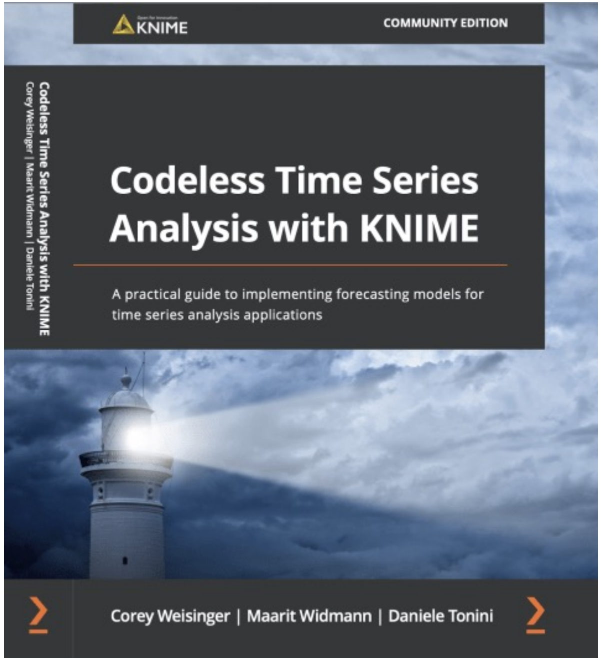


Forecasting two seasonality directly with SARIMA and with Decomposition



Codeless Time Series Analysis with KNIME

- Understand time series projects end-to-end
- Practice with use cases based on statistical and machine learning techniques



KNIME Hub

Corey.Weisinger@knime.com

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

Open for Innovation KNIME