Statistics and Mathematics

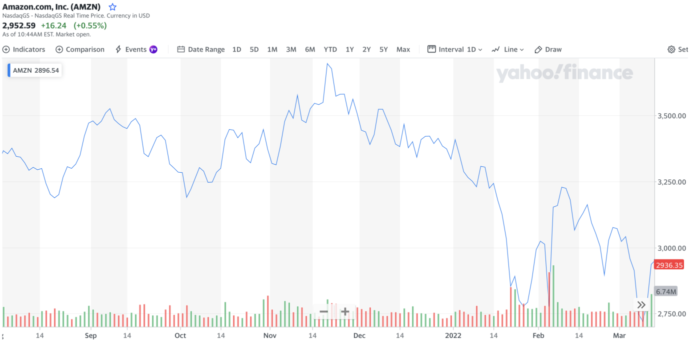
Arthur Enders

RWTH Business School

Time Series Analysis

What are time series?

The main aspect of time series data compared to other data is **time**. Time series are thus data points collected over time: time will always be on the x-axis. Time is a crucial variable that **inherits important information** for the underlying data.



Amazon.com, Inc. (AMZN)

NasdaqGS - NasdaqGS Real Time Price. Currency in USD

2,952.59 +16.24 (+0.55%)

As of 10:44AM EST. Market open.

Time Series Aspects

Data over time includes certain difficulties that are unique to time series data. Specifically designed models for time series data take care of these **aspects**:

- Trends
- Stationarity/Non-Stationarity
- Auto-correlation
- Seasonality or cycles
- Conditional variance

Applications

Time series modelling is **widely applied in different industries** ranging from finance to health care. Popular applications include:

- Economic forecasting (revenue, sales, demand ...)
- Financial modelling (stock prices, volatility, ...)
- Predicting customer behavior (user views, new users, ...)
- Anomaly detection (defect detection, ECG signals, ...)

Application Example

Example: Amazon needs to predict the demand for **socks** in December 2022 in North Rhine-Westphalia, Germany.

- The company uses **time series modelling** to **forecast** the demand and thus the required inventory.
- The forecast is crucial to meet customer needs, lower shipping time and reduce inventory waste.
- Any large forecast errors lead to substantial declines in revenue and customer satisfaction.

Time Series Models

The time series models that we will look at:

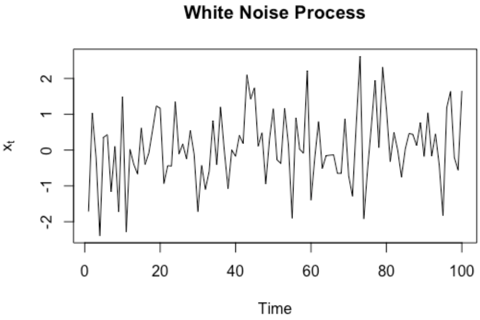
- Autoregressive Model (AR)
- Moving Average Model (MA)
- ARMA
- AR Integrated MA (ARIMA)
- Seasonal ARIMA (SARIMA)
- AR Conditional Heteroskedasticity Model (ARCH)
- Generalized ARCH (GARCH)

We can use these models to analyze the time series and make **forecasts for the future**.

Stationarity

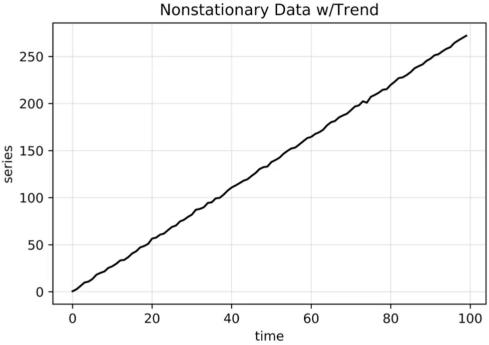
Most time series models require stationary time series data. A time series is **stationary** if it has these 3 properties:

- Constant mean
- Constant variance
- No periodic/seasonal component



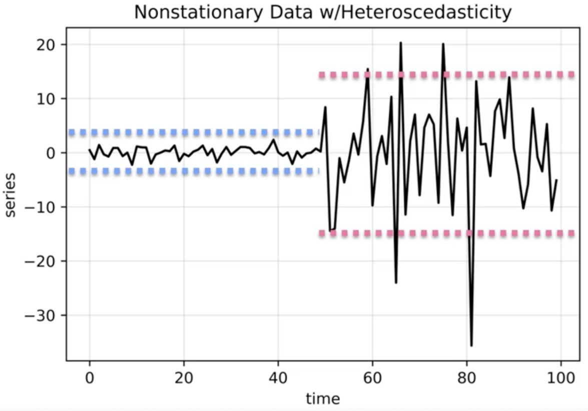
Stationarity - Counterexamples

No Constant Mean: If the time series has an underlying **trend** (in this case increasing), the mean is not constant but changing throughout time.



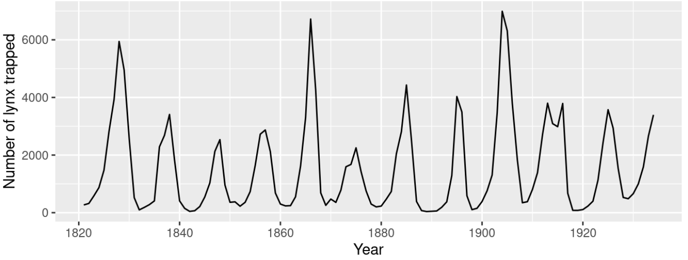
Stationarity - Counterexamples

No Constant Variance: If the time series has a differing variance across observations, the series is called **heteroscedastic**, the variance is thus not constant throughout time.



Stationarity - Counterexamples

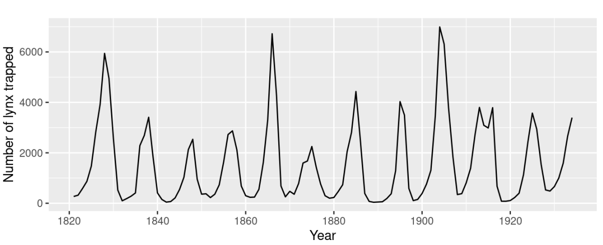
Seasonal/Periodic Component: Many time series experience **seasonality/cyclicity**. The variable is thus highly influenced by the current season/period/cycle.



Autocorrelation

Autocorrelation is a key concept and prevalent in most time series data. It means today's value is significantly dependent on (correlated with) some past value(s).

The time interval between the correlated values is called **lag**.
Example: If it rained yesterday, it is more likely to also rain today (this would imply a lag=1).

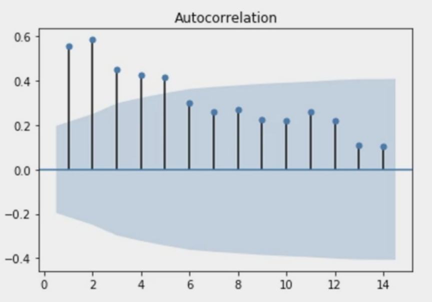


Autocorrelation Function

To check for autocorrelation, we can use a plot for the autocorrelation function: **ACF**.

It shows the correlation between the variable and its past values.

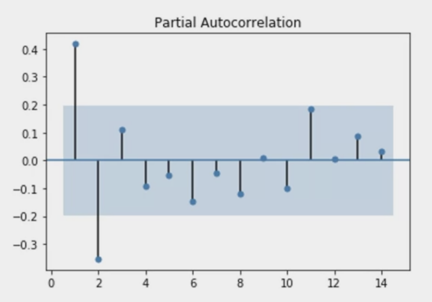
The problem with the ACF is that it does not control for other lags. For example, the correlation between y_t and y_{t-3} is probably also influenced by y_{t-1} and y_{t-2}.



Partial Autocorrelation Function

The partial autocorrelation function **PACF** also summarizes dependence on past values but it only measures partial results (controlling for other lags).

We can use the PACF plot to determine which lags to incorporate into our models. For example on this plot, we can see that only the two most recent values are statistically significant (lags 1 and 2).



Autoregressive Model

The autoregressive **(AR)** model specifies that the variable linearly depends on its own **past values**.

An AR(**p**) model is assumed to depend on the **last p values**. For **p = 2**:

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \omega_t,$$

where ϕ_1 and ϕ_2 are the regression coefficients and ω_t is the **forecast error**, similar to an ordinary linear regression with y_{t-1} and y_{t-2} as the regressors.

To determine **p**, we can use the significant lags of the PACF.

Moving Average Model

The moving average **(MA)** model specifies that the variable linearly depends on **past forecast errors**.

An MA(**q**) model is assumed to depend on the **last q values** of the forecast errors. For **q = 2**:

$$y_t = c + \theta_1 \omega_{t-1} + \theta_2 \omega_{t-2} + \omega_t,$$

where θ_1 and θ_2 are the regression coefficients and ω_{t-1} and ω_{t-2} are the previous forecast errors.

To determine **q**, we can use the significant lags of the ACF.

Let's practice!

Notebooks:

- *AR Model*

Data:

- `ice_cream.csv`

ARMA Model

The **ARMA** model combines the aspects of the AR and the MA models.

Combining an **AR(2)** with **p=2** and an **MA(2)** with **q=2** leads to an **ARMA(2,2)** model:

$$y_t = \mu + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \theta_1 \omega_{t-1} + \theta_2 \omega_{t-2} + \omega_t,$$

which thus depends on past observations values as well as past values of forecast errors.

The ARMA model assumes that the time series is **stationary**. We also need to check if the time series has a **seasonal** component, then we need to include a seasonal AR term.

ADF Test

We can test if a time series is stationary with the **Augmented Dickey-Fuller test (ADF test)**.

- *H0*: A unit root is present (the time series is non-stationary).
- *H1:* The time series is stationary.

Thus we we want to reject the null-hypothesis (p-value < 0.05).

Differencing

If we find the time series to be non-stationary, we first have to make it stationary in order to apply the time series models that rely on this assumption.
We can often transform non-stationary series into stationary series by **differencing**. Differencing is done by subtracting the previous value from each observation.

Example: Stock prices are non-stationary (but **integrated** instead). By differencing we go from stock prices to stock returns. Stock returns are usually stationary.

The differencing can be done manually, to then apply i.e. the ARMA model. Or we can directly use the **ARIMA** model that does this automatically.

ARIMA

The auto regressive integrated moving average model **(ARIMA)** is an ARMA model designed to deal with **integrated** data. The model is denoted by ARIMA(**p**,**d**,**q**) with:

- the order of the AR model **p**.
- the number of times to difference the data **d**.
- the order of the MA model **q**.

d is the number of times to perform a lag-1 difference on the integrated data Y_t.

Example:

$$\textbf{d=1} : y_t = Y_t - Y_{t-1}$$
$$\textbf{d=2} : y_t = (Y_t - Y_{t-1}) - (Y_{t-1} - Y_{t-2})$$

SARIMA

If the time series has a seasonal component, we can use the Seasonal ARIMA **(SARIMA)**. It is used to remove seasonal components.

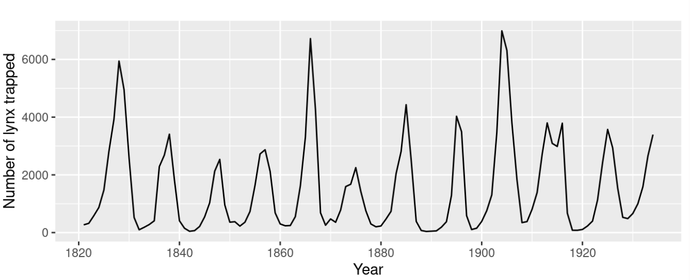
The model is denoted by **SARIMA(p,d,q)(P,D,Q)**$_M$. P, D and Q are the equivalent p, d and q of the ARIMA model but are applied seasonally with the frequency M.

Example: If there is a yearly seasonal trend and the data is given monthly, **m** should be set to **12**. Seasonal trends can then be removed with **D = 1**, and if there is interdependence of the seasonal data we can for example also set **P=1** and/or **Q=1**.

Identify Seasonal Components

We can identify if seasonal components are present by:

- ACF and PACF plots
- Seasonal subplots
- Intuition



Let’s practice!

Notebooks:

- *SARIMA*

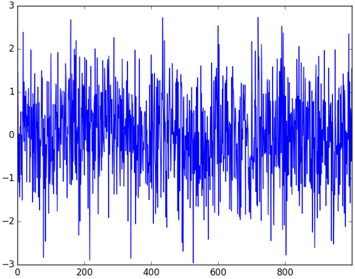
Data:

- `champagne.csv`

Evaluate the Fit

To evaluate the fit of the applied model, we should:

- examine the residuals: residuals should resemble **white noise** with a mean of 0.
- examine the forecast errors: MSE, MAE, ... (be wary of overfitting)
- examine the **likelihood** of the function
- examine **information criteria**: AIC and BIC (penalize overfitting by taking into account the amount of parameters)

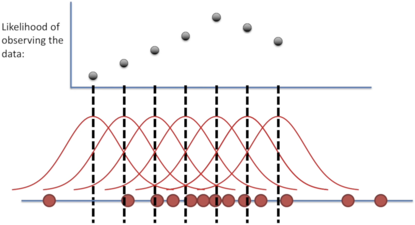


(Log) Likelihood

Let's say we have:

- some observed data points (sample): $x_1, x_2, x_3, \ldots$
- a set of probability distributions (functions) which could have generated the data. A distribution is denoted by θ.

The likelihood is a function $L(\theta | x_1, x_2, x_3, \ldots)$ that gives us the probability (likelihood) of observing the sample $(x_1, x_2, x_3, \ldots)$ when the data is extracted from the probability distribution θ.



Information Criteria

In order to choose the model orders (p,d,q) and evaluate the fit while taking into account the problem of **overfitting**, we can use the information criteria.

Each model has a log likelihood (l), a number of parameters (k) and a number of samples used for fitting (n). Then we calculate:

- Akaike Information Criterion: **AIC =** $2k - 2l$
- Bayesian Information Criterion: **BIC =** $\ln(n)k - 2l$

AIC and BIC are not exclusive to time series models but are also used to evaluate other models, particularly other machine learning models.

Let’s practice!

Notebooks:

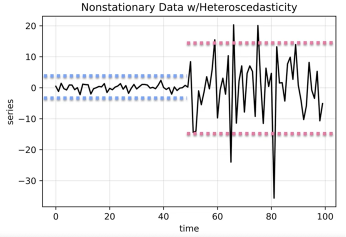
- *Model Selection*

Data:

- `catfish.csv`

Heteroscedasticity

The SARIMA model with its transformations can handle time series data with trends (non-constant mean) and seasonality. But the models we have seen so far can not yet handle **heteroscedasticity** (non-constant/changing variance).



If we evaluate our model and the residuals do not resemble white noise but instead we can see different levels of variance, our model does not fit well. We need to use other models that take this into account.

ARCH

The Autoregressive Conditional Heteroscedasticity (**ARCH**) model can deal with **heteroscedasticity** by taking into account past variances of the error terms.

The ARCH(p) model estimates the current error term by taking into account the past p (squared) error terms. An ARCH(2) model is denoted by:

$$\epsilon_t = w_t \sqrt{\alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \alpha_2 \epsilon_{t-2}^2}$$

GARCH

The Generalized Autoregressive Conditional Heteroscedasticity (**GARCH**) model extends the ARCH model by adding a moving average parameter (similar to ARMA compared to AR). GARCH is a better fit for modeling time series data when the data exhibits heteroscedasticity but also **volatility clustering**.

The GARCH(p, q) model estimates the current error term by taking into account the past p (squared) error terms and the past q variances. A GARCH(1,1) model is denoted by:

$$\sigma_t^2 = w + \alpha_0 + \alpha\epsilon_{t-1}^2 + \beta\sigma_{t-1}^2$$

GARCH Appliance

GARCH models can only be applied to forecast the variance with conditional heteroscedasticity. Thus, the GARCH model **cannot predict negative values**.

We can try to transform the series accordingly to mimic volatility. For example: transform prices to returns and then square the returns to have only positive values.

We can also apply it to, for example, model the (squared) residuals after we applied another appropriate model (i.e. OLS, AR, ARMA, ...). If we see that residuals are not yet white noise but exhibit clustered volatility, the GARCH model can be a good fit to forecast the variance of the residuals and further improve our model.

GARCH Applications

The GARCH model has **many real world applications** that mostly require a forecast of volatility.

Examples:

- Price volatility forecasting (stocks, commodities, forex, ...)
- Variance forecasting (inflation, interest rates, ...)
- Option pricing
- High frequency trading
- Risk modelling

Let’s practice!

Notebooks:

- *GARCH SP500*

Apply what we’ve learned

Let’s use a time series model to predict stock prices

If we find a fitting model that can predict if a stock goes either **up** or **down** the next day with a consistent **accuracy above 50%**, we can make guaranteed profit.

Final Project

Notebooks:

- *Trading Algo*

Reality

When we have a consistently working model, we could theoretically scale our profit to infinity...

So it is most likely **not possible in reality in the long-term**, but many traders still believe that and try their best to achieve it.

This could also be the exact reason why it does not work. If you find a working model/pattern, chances are other trades will eventually also find it and **once the market is aware, the algorithm no longer works**.

Thank you!

Arthur Enders – enders@econ.rwth-aachen.de

Computational Economics
RWTH Aachen University
Templergraben 64
52062 Aachen

www.compecon.rwth-aachen.de