Time Series Forecasting

Henrika Langen

Central European University
Department of Economics and Business

Winter 2024

Structure of the Lecture

1. Introduction to time series
2. Modeling time series and evaluating forecasting performance
3. Exponential smoothing models
4. **ARIMA models**
5. Regression models and advanced forecasting models
6. Machine Learning and Deep Learning methods

ARIMA Models

Topics

- Stationarity and differencing
- Autoregressive models
- Moving average models
- Non-seasonal and seasonal ARIMA models

Stationarity

A stationary time series is one whose **statistical properties are stable over time**.

- **Weak Stationarity**: The expected value $E[Y_t]$ and autocorrelation of the process underlying the time series remain constant over time.
- **Strong Stationarity**: The distribution of the underlying process remains constant over time

In the following, we focus on weak stationarity.

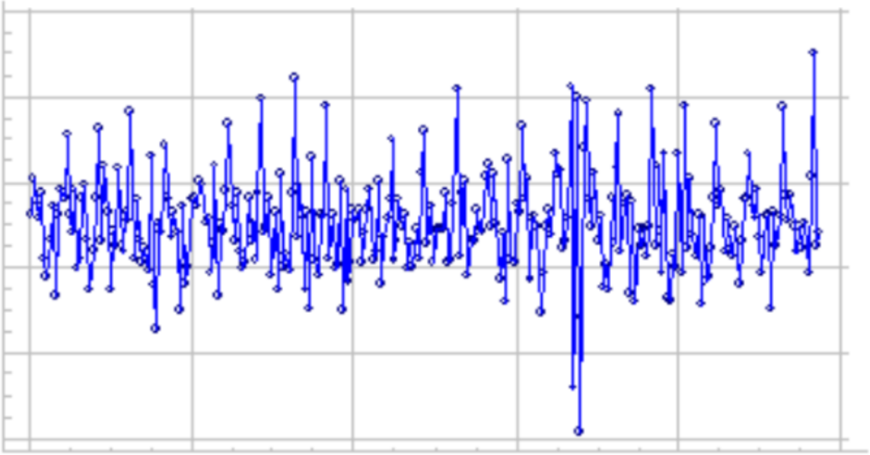
Why is stationarity important?

- The stability of their statistical properties makes stationary series comparably easy to model

$\Rightarrow$ For stationary time series, it's easier to develop reliable models and make accurate forecasts

$\Rightarrow$ It's often advisable to transform a non-stationary series to become stationary

Stationarity



Stationary time series look pretty similar at any point in time. In time plots, they show a mostly horizontal path (although some cyclic behavior may occur) with consistent variance.

Stationarity

Test for stationarity

- **Check ACF plot:** The ACF plot of stationary series drops to zero quickly, while that of non-stationary series tends to decrese slowly.
- **Formal tests:**
  - The Augmented Dickey–Fuller test tests the null hypothesis (H_0) that a series is non-stationary.

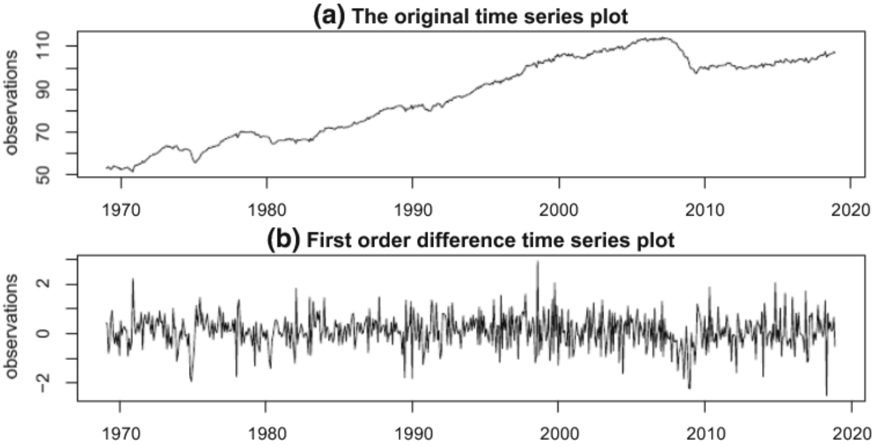
Data Transformations: Differencing

In order to transform a non-stationary time series into a stationary one, we can calculate a differenced version of our original series:

- **Simple/first-order differencing:** Compute the differences between consecutive observations to eliminate trends in the series.
  $y'_t = y_t - y_{t-1}$
- **Second-order differencing**: If the differenced data still exhibits trends, it may be necessary to difference the series another time.
  $y''_t = y_t - y_{t-1} - (y_{t-1} - y_{t-2}) = y_t - 2y_{t-1} + y_{t-2}$
- **Seasonal differencing:** Calculate the difference between an observation and the previous observation from the same season.
  $y'_t = y_t - y_{t-m}$, where m is the seasonal frequency.

All of the above methods can be combined as needed to make a series stationary.

Data Transformations: Differencing



Monthly US industrial production index for consumer goods.

Source: 10.1007/s00362-021-01231-6.

Data Transformations: Variance Stabilization

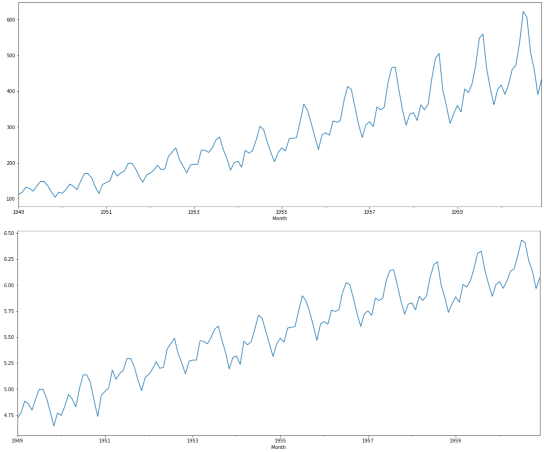
If the variation of a series increases with the level of the series, certain transformations can stabilize the variance:

- **Logarithmic transformation** (only if series does not take on values ≤ 0): $w_t = log(y_t)$, where $log()$ is the natural logarithm. Changes in log values can be interpreted as percentage changes on the original scale.
- **Box-Cox transformation** (if series also takes on values ≤ 0):

$$w_t = \begin{cases} log(y_t) & \text{if } \lambda = 0 \\ (y_t^\lambda - 1)/\lambda & \text{otherwise} \end{cases}$$

where λ is selected such that the size of the variation is similar across the whole series.
(After analyzing and forecasting the Box-Cox-transformed data, we need to back-transform the series to obtain forecasts on the original scale.)

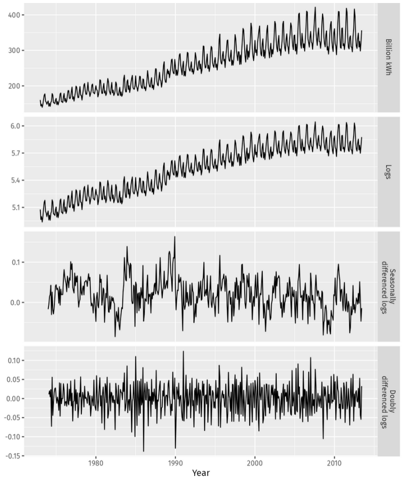
Data Transformations and Adjustments



International airline passengers per month (in thousands), level data (top) and logarithmized (bottom).

Source: Kaggle.

Data Transformations and Adjustments



US net electricity generation (billion kWh).

Source: FPP.

Backshift Notation

- Lagged values of y_t can be expressed with **the backward shift operator** B:

  E.g. $By_t = y_{t-1}$ and $B(By_t) = B^2 y_t = y_{t-2}$
- B can be convenient for representing differenced series:
  - 1st-order differencing: $y'_t = y_t - y_{t-1} = y_t - By_t = (1-B)y_t$
  - 2nd-order differencing:

    $$\begin{aligned} y''_t &= (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) = y_t - 2y_{t-1} + y_{t-2} \\ &= y_t - 2By_t + B^2 y_t = (1 - 2B + B^2) y_t = (1-B)^2 y_t \end{aligned}$$
  - dth-order differencing: $(1-B)^d y_t$
  - seasonal differencing followed by 1st-order differencing:
    $(1-B)(1-B^m)y_t = (1 - B - B^m + B^{m+1})y_t = y_t - y_{t-1} - (y_{t-m} - y_{t-m-1})$

ARIMA Models

- While exponential smoothing models focus on describing trend and seasonality, ARIMA-models aim at capturing the autocorrelations in the data.
- Exponential smoothing methods are suitable for non-stationary data, while ARIMA models should be used with stationary data only ($\Rightarrow$ make data stationary before fitting ARIMA model).

$\Rightarrow$ Exponential smoothing and ARIMA-models complement each other.

- ARIMA-models combine three components:
    - Auto-regression (AR): relation between present and past values of y_t.
    - Integration/Differencing (I): differencing in order to make the time series stationary.
    - Moving Average (MA): relation between present value of y_t and residuals from past predictions.

Autoregressive Models (AR(p))

- **Underlying assumption:** The value of the outcome y_t is linearly dependent on its previous values.
- **Approach:** The outcome variable is modeled using a linear combination of its past values. Therefore, the term "autoregressive": a regression of the outcome on itself.
- An autoregressive model of order p (AR(p)-model) is defined as:

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

  where c is a constant and ε_t is white noise.
- **Characteristics:** The parameters $\phi_1, \dots, \phi_p$ are selected so that the model captures the shape of the series, the variance of ε_t to capture the scale.

Autoregressive Models (AR(p))

Special cases of AR(p)-models

- AR(1): If $\phi_1 = 0$, y_t is **white noise**.
- AR(1): If $\phi_1 = 1$ and $c = 0$, y_t is a **random walk**.
- AR(1): If $\phi_1 = 1$ and $c \neq 0$, y_t is a **random walk with a drift**.

Usually AR(p)-models are only used for stationary data. For an AR(p)-model to be stationary, certain conditions must hold:

- AR(1): If $-1 < \phi_1 < 1$, y_t is **stationary**.
- AR(2): If $-1 < \phi_1 < 1$ and $\phi_1 + \phi_2 < 1$, y_t is **stationary**.
- AR(p) with $p > 2$: The restrictions for stationarity are relatively complicated.

Moving Average Models

Moving average models (MA(q))

- **Approach:** The outcome variable is modeled as a weighted average of past forecasting errors.
- A moving average model of order q (MA(q)-model) is defined as:

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

  where ε_t is white noise.
- **Careful!** The MA(q) model (= weighted average of past values of ε_t for prediction purposes) should not be confused with the moving average approach used for smoothing time series and for dealing with missing values (= weighted average of y_t in a moving window for smoothing purposes).
- **Characteristics:** The parameters $\theta_1, \ldots, \theta_p$ are selected so that the model captures the shape of the series, the variance of ε_t to capture the scale.

Inverting AR(p) and MA(q) processes

- Any stationary AR(p) process can be written as an MA(∞) process. E.g.:

$$
\begin{aligned}
y_t &= \phi_1 y_{t-1} + \varepsilon_t \\
&= \phi_1(\phi_1 y_{t-2} + \varepsilon_{t-1}) + \varepsilon_t \\
&= \phi_1^2 y_{t-2} + \phi_1 \varepsilon_{t-1} + \varepsilon_t \\
&= \phi_1^2(\phi_1 y_{t-3} + \varepsilon_{t-2}) + \phi_1 \varepsilon_{t-1} + \varepsilon_t \\
&= \phi_1^3 y_{t-3} + \phi_1^2 \varepsilon_{t-2} + \phi_1 \varepsilon_{t-1} + \varepsilon_t \\
&\dots
\end{aligned}
$$

  where with $-1 < \phi_1 < 1$ the terms diminish the farther they are in the past.
- Conversely, every invertible MA(q) process can be written as an AR(∞) process.

ARMA(p,q) processes

- The ARMA(p,q) process combines AR(p) and MA(q) to capture both autoregressive relationships and relationships in error terms:

$$y_t = \phi_1 y_{t-1} + \ldots + \phi_p y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q}$$

- E.g.: An ARMA(2,1) process is given as:

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \varepsilon_t + \theta_1 \varepsilon_{t-1}$$

Modeling a time series with a non-seasonal ARIMA(p, d, q) model

- The AutoRegressive Integrated Moving Average model (ARIMA(p, d, q)) is an approach to model non-seasonal time series as an ARMA(p, q)-process
- To adequately model a time series as an ARMA(p, q) process, differencing may be necessary to achieve stationarity.
  - $\Rightarrow$ The "Integrated" component I(d) refers to d-degree differencing of the time series to attain stationarity before applying the AR(p) and MA(q) components.
- Once differenced to achieve stationarity, the series can be modeled as an ARMA(p, q) process.

Modeling a time series with a non-seasonal ARIMA(p, d, q) model

Example: An ARIMA(2,1,1) process is given as:

$$y_t - y_{t-1} = \phi_1(y_{t-1} - y_{t-2}) + \phi_2(y_{t-2} - y_{t-3}) + \varepsilon_t + \theta_1 \varepsilon_{t-1}$$

Or, in backshift notation:

$$(1 - \phi_1 B - \phi_2 B^2) \cdot (1 - B) \cdot y_t = (1 + \theta_1 B)\varepsilon_t$$

How to determine the parameters p, d, q for a non-seasonal ARIMA(p, d, q) model

- **Determining d:** d can be determined by checking the time plot, the ACF plot and/or applying the Augmented Dickey–Fuller tests in the original and differenced series.
- **Determining p & q:**
  - If the underlying process is of type ARIMA($p, d, 0$) or ARIMA($0, d, q$), p and q can be determined based on **ACF and PACF plots**:
    - If the ACF is quickly decaying, there is a significant peak at lag p in the PACF and none beyond lag p, an ARIMA($p, d, 0$) might be a good fit to the data.
    - If the PACF is quickly decaying, there is a significant peak at lag q in the ACF and none beyond lag q, an ARIMA($0, d, q$) might be a good fit to the data.

How to determine the parameters p, d, q for a non-seasonal ARIMA(p, d, q) model

- **Determining p & q:**
  - If the process generating a time series is likely to be more complicated than ARIMA$(p, d, 0)$ or ARIMA$(0, d, q)$, implement the **Hyndman-Khandakar algorithm for automatic ARIMA modeling**:
    1. d is determined using the Augmented Dickey–Fuller test on the original, 1-level-differenced and 2-level-differenced data.
    2. A pre-defined set of initial models is fit to the d-level-differenced data.
    3. The model with the lowest AIC$_c$ is chosen.
    4. Variations of the best model (± 1 for p and/or q, c added or removed) are fitted and the one with the lowest AIC$_c$ is chosen.
    5. Step 4 is repeated until no model with a lower AIC$_c$ is found.

How to determine the parameters p, d, q for a non-seasonal ARIMA(p, d, q) model

When a model is found that seems to fit the data well (either manually or using the Hyndman-Khandakar algorithm), it's important to examine the residuals.

- Check the ACF of the residuals.

$\Rightarrow$ The residuals should resemble white noise. If there is autocorrelation left in the residuals, try to find a better-fitting model.

- Examine the residuals by plotting a histogram.

$\Rightarrow$ The residuals should be approximately normally distributed as this facilitates the estimation of prediction intervals.

Modelling a time series with a seasonal ARIMA model

- To adapt ARIMA models to fit seasonal data, additional seasonal terms are added:

$$\text{ARIMA} \qquad \underbrace{(p,d,q)}_{\text{non-seasonal parameters}} \qquad \underbrace{(P,D,Q)_m}_{\text{seasonal parameters}}$$

  where m is the seasonal frequency.
- D-degree-seasonal differencing helps to remove seasonality and make the data stationary; P and Q help account for relationships between the same season in different seasonal cycles.
- The seasonal ARIMA-models are best written with backshift notation as they else quickly become complicated.

Modelling a time series with a seasonal ARIMA model

Example: For monthly data, a potentially suitable seasonal ARIMA-models is ARIMA$(0,1,1)(0,0,1)_{12}$:

$$\underbrace{(1-B)}_{\text{1-degree differencing}} \cdot y_t = \underbrace{(1+\theta_1 B)}_{\text{non-seasonal MA(1)}} \cdot \underbrace{(1+\Theta_1 B^{12})}_{\text{seasonal MA(1)}} \varepsilon_t$$

$$\Leftrightarrow y_t - y_{t-1} = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \Theta_1 \varepsilon_{t-12} + \theta_1 \Theta_1 \varepsilon_{t-13}$$

or ARIMA$(1,1,1)(1,1,1)_{12}$:

$$(1-\phi_1 B)(1-\Phi_1 B^{12})(1-B)(1-B^{12})y_t = (1+\theta_1 B)(1+\Theta_1 B^{12})\varepsilon_t$$

Forecasting with an ARIMA Model

Point estimates:

- Bring y_t to the left-hand side of the ARIMA equation.
- Replace t with $T+h$.
- Replace future observations with their forecasts, future errors with zero and past errors with the corresponding residuals

Prediction intervals:

- The calculation of prediction intervals for ARIMA-models is based on the assumption that the residuals are uncorrelated and normally distributed.
- Check if this holds (via histogram and residual ACF plot), before calculating prediction intervals.
- The prediction interval for 1-step-ahead forecasts can be calculated as $\hat{y}_{T+1|T} \pm 1.96\hat{\sigma}_e$ with $\hat{\sigma}_e$ the standard deviation of the residuals.
- For most ARIMA models, the calculation of multi-step forecast prediction intervals is relatively complex but they are provided in Python.

Forecasting with an ARIMA Model

Example: Forecasting based on an estimated ARIMA(1,1,1)-model:

$$
\begin{aligned}
&(1-\hat{\phi}_1 B)(1-B)\cdot y_t = (1+\hat{\theta}_1 B)\varepsilon_t \\
\Leftrightarrow &(y_t - y_{t-1}) = \hat{\phi}_1(y_{t-1} - y_{t-2}) + \varepsilon_t + \hat{\theta}_1 \varepsilon_{t-1} \\
\Leftrightarrow &y_t = y_{t-1} + \hat{\phi}_1 y_{t-1} - \hat{\phi}_1 y_{t-2} + \varepsilon_t + \hat{\theta}_1 \varepsilon_{t-1}
\end{aligned}
$$

For a 1-step-ahead forecast, replace t with $T+1$:

$$
y_{T+1} = y_T + \hat{\phi}_1 y_T - \hat{\phi}_1 y_{T-1} + \underbrace{\varepsilon_{T+1}}_{E(\varepsilon_{T+1})=0} + \hat{\theta}_1 \underbrace{\varepsilon_T}_{\text{estimable by } r_T}
$$

$$
\Rightarrow \hat{y}_{T+1|T} = y_T + \hat{\phi}_1 y_T - \hat{\phi}_1 y_{T-1} + \hat{\theta}_1 r_T
$$

For a multi-step forecast:

$$
\begin{aligned}
\hat{y}_{T+2|T} &= \hat{y}_{T+1|T} + \hat{\phi}_1 \hat{y}_{T+1|T} - \hat{\phi}_1 y_T \\
\hat{y}_{T+h|T} &= \hat{y}_{T+h-1|T} + \hat{\phi}_1 \hat{y}_{T+h-1|T} - \hat{\phi}_1 \hat{y}_{T+h-2} \quad \text{for } h > 2
\end{aligned}
$$

Advantages and Disadvantages of ARIMA models and which data to use them for

+ ARIMA models are particularly effective for short-term forecasts.
+ Through differencing, ARIMA models can handle non-stationary data well.
- ARIMA models do not accurately predict turning points and sudden shifts in time series.
- Finding the optimal order of an ARIMA model involves some level of subjectivity.

Differences and Similarities of ARIMA and ETS models

- ARIMA models time series as stationary processes, while ETS models them as non-stationary processes.
- Purely additive exponential smoothing methods could be considered special cases of the ARIMA models, i.e., can be expressed in ARIMA(p, d, q) form, but there are several ETS models that have no ARIMA counterpart and vice versa.