Greening Energy Market and Finance

Time series analysis with applications to green energy markets

Dr. Andrea Mazzon,

Ludwig Maximilians Universität München

Project website: http://grenfin.eu

With the support of the Erasmus+ Programme of the European Union

Main contents

What we want to do today

Aim of the lecture of today is to:

- Discuss the evolution of renewable energy prices in the last years/decades, with the support of some graphs;
- Embed this analysis in a more quantitative framework: i.e., using time series;
- Give then an overview / recap of the main features of time series;
- Follow this overview with a practical example: prices of Solar PV modules, from 1983 to 2019;
- For such an example, show the implementation in Python of the analysis we will discuss.

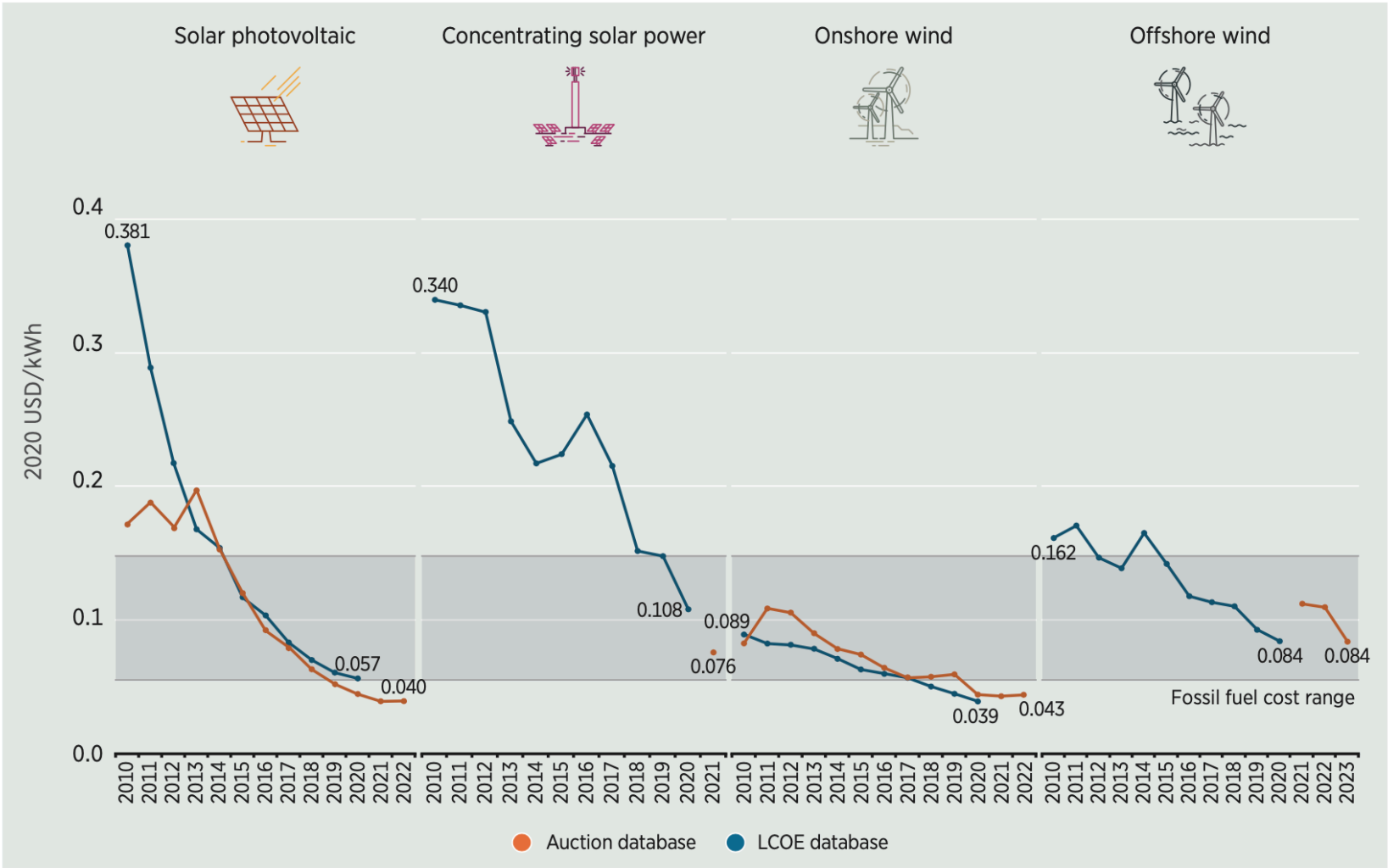
The decay of renewable energy costs in last years

The following slides are based on the report *Renewable power generation costs in 2020* from the International Renewable Energy Agency (IRENA), available at

https://www.irena.org/publications/2021/Jun/Renewable-Power-Costs-in-2020

- In the decade 2010 to 2020, solar and wind power technologies have become more and more competitive.
- The global weighted average of levelized cost of electricity fell:
  - 85% for solar photovoltaics;
  - 68% for concentrating solar power;
  - 56% for onshore wind;
  - 48% for offshore wind,
- Not only renewables energies are now competing with fossil fuels, but are undercutting them.

LCOE and PPA/auction prices for solar PV, onshore wind, offshore wind and CSP, 2010-2023



Source: IRENA Renewable Cost Database

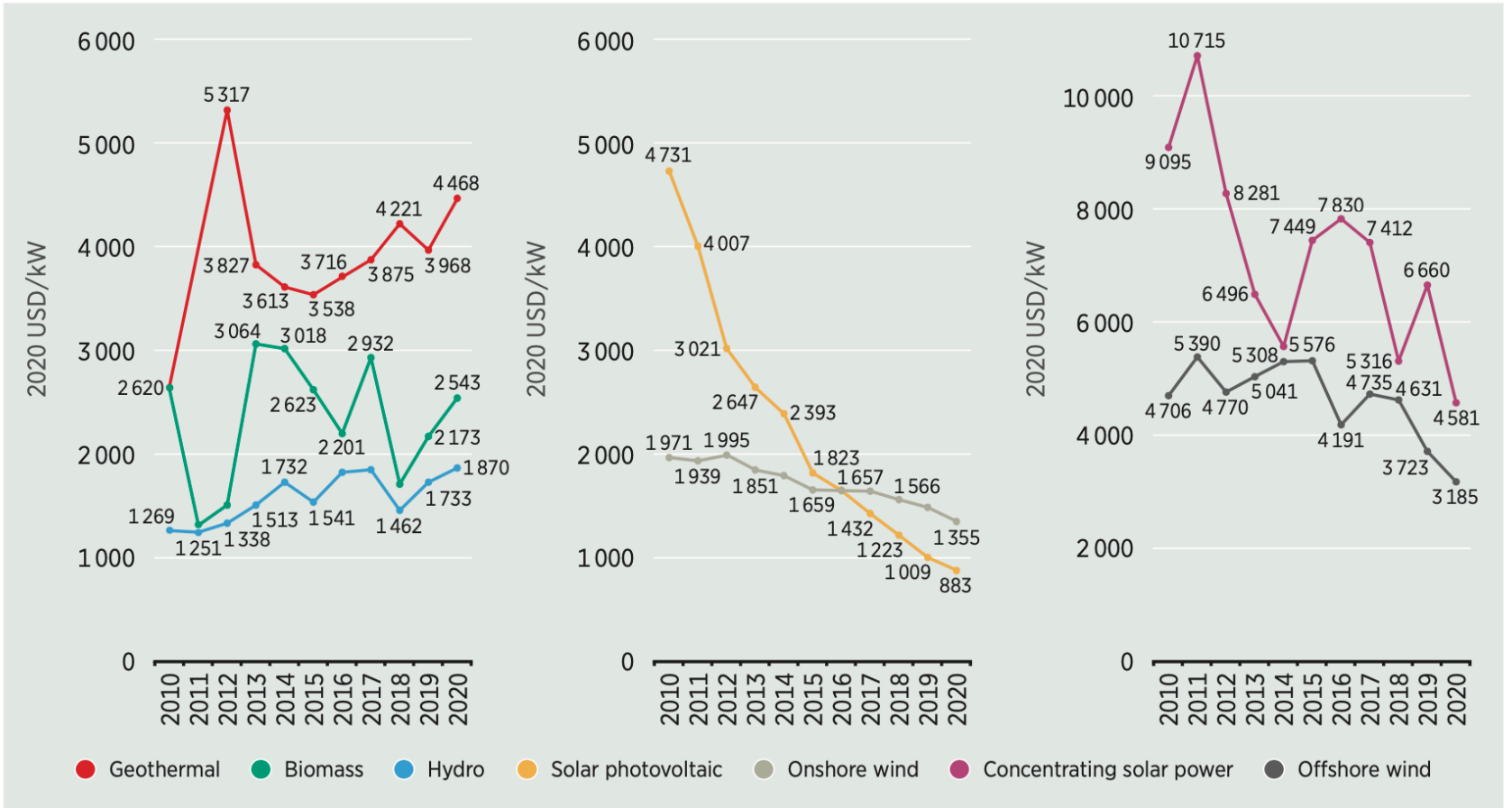
Note: The thick lines are the global weighted average LCOE, or auction values, by year. For the LCOE data, see Figure ES2 note. The band that crosses the entire chart represents the fossil fuel-fired power generation cost range.

Some comparison between coal-fired power plants and renewable energy

In 2021:

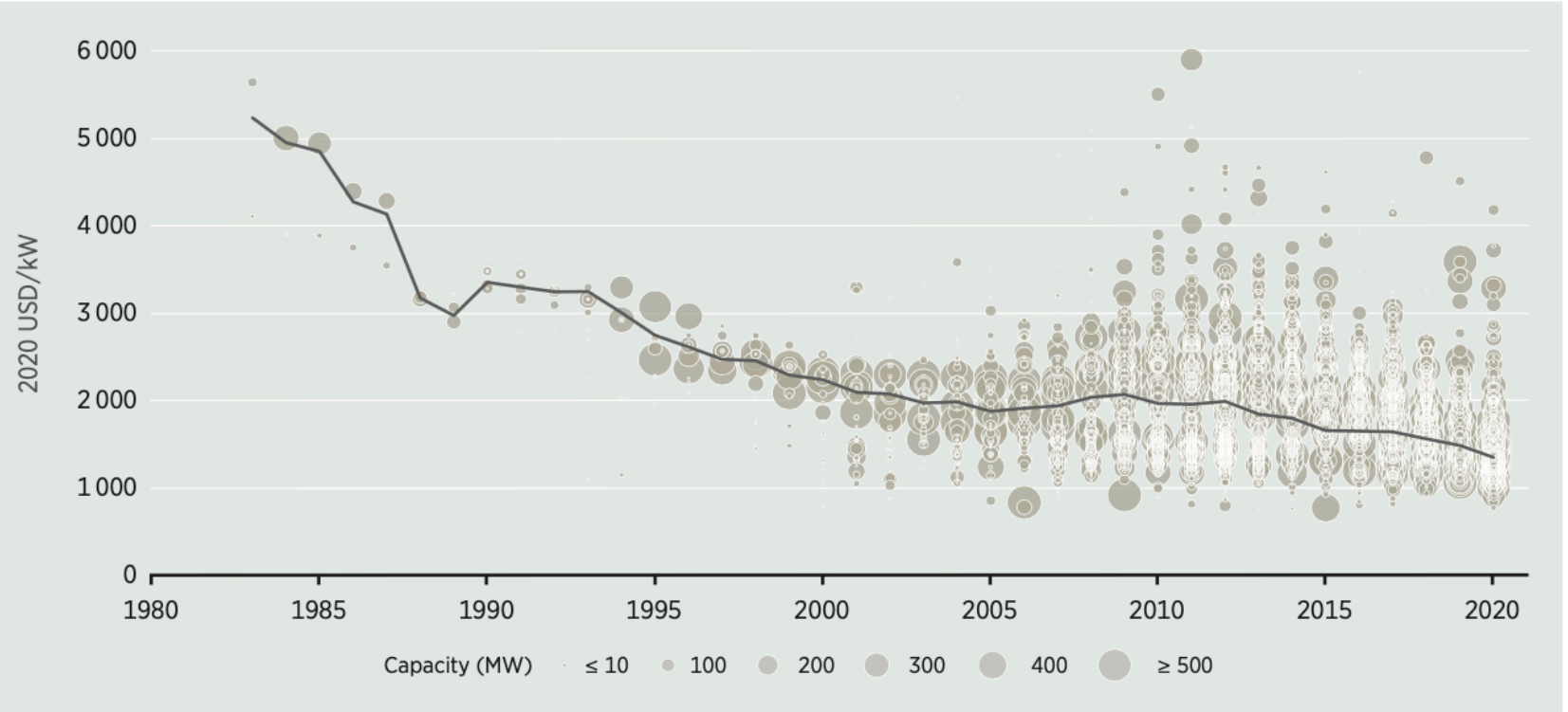
- In Europe, coal-fired power plant operating costs are above the costs of new solar PV and onshore wind (including the cost of CO2 prices).
- In the United States, between 77% and 91% of the existing coal-fired capacity has operating costs that are estimated to be higher than the cost of new solar or wind power capacity.
- In India, the figure is between 87% and 91%.

Global weighted-average total installed costs by technology, 2010-2020



Source: IRENA Renewable Cost Database

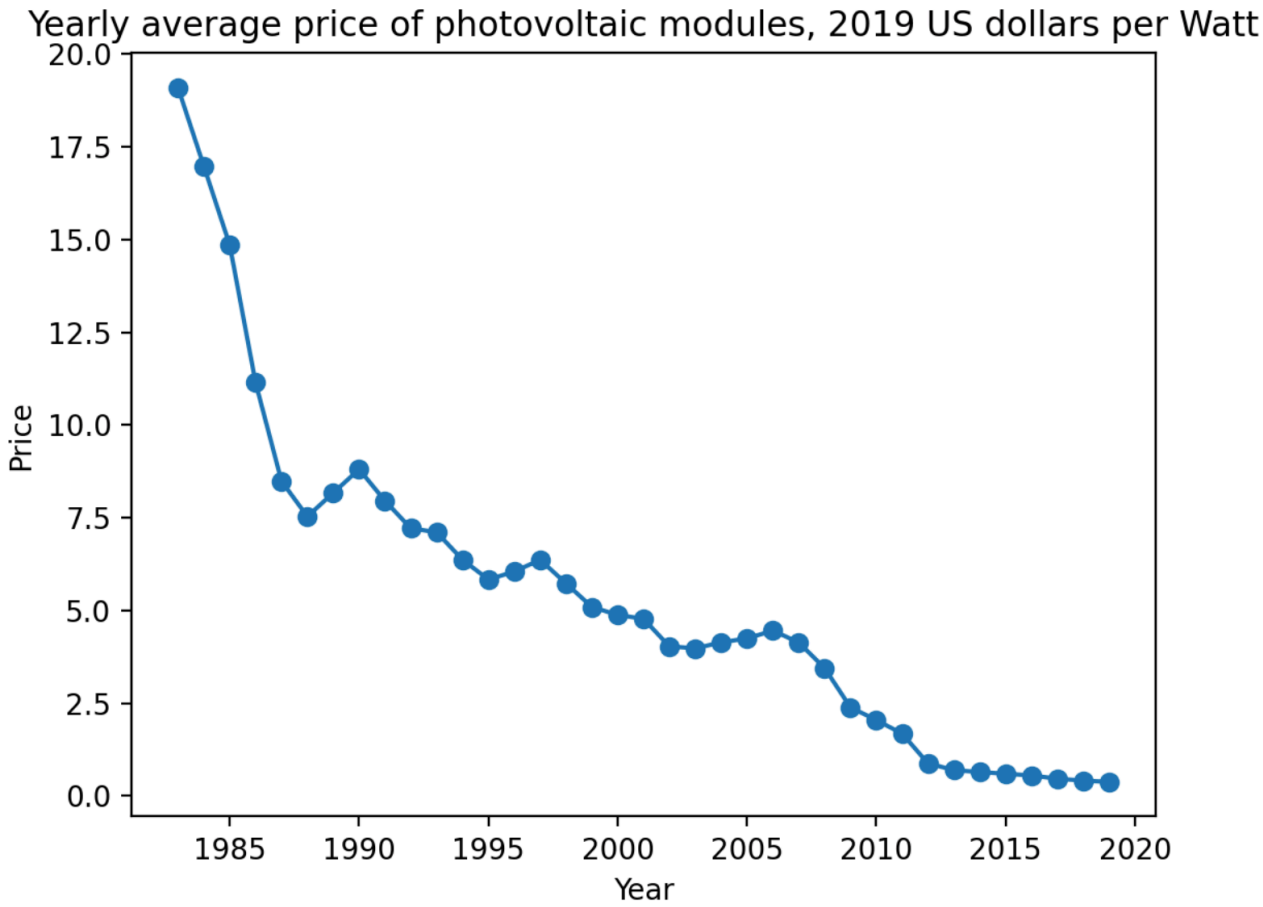
Total installed costs of onshore wind projects and global weighted-average, 1983-2020



Source: IRENA Renewable Cost Database

Solar PV module prices

We now consider a dataset that we are going to use along the whole lecture: Global average price of solar photovoltaic (PV) modules, measured in 2019 US dollars per Watt, from 1983 to 2019.
Source: Our World in Data. The dataset is downloadable also in Excel format at the following link: https://ourworldindata.org/grapher/solar-pv-prices



Towards time series

The figure at the last slide clearly exhibits a decreasing trend.

But: can we analyze it further? More quantitatively? Can we make predictions on the price evolution in next years?

Time series analysis!

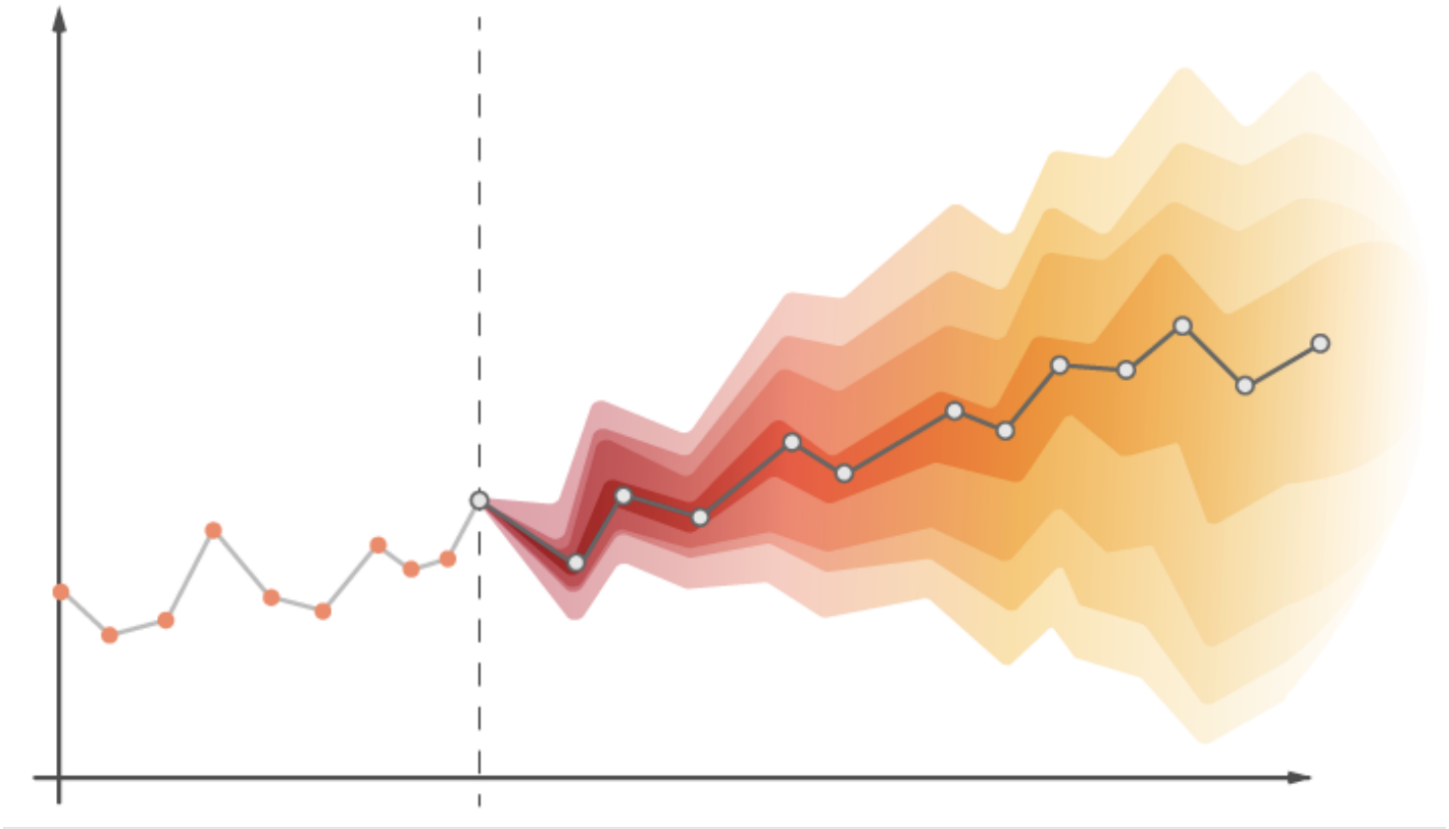
What are time series?

Think about the evolution of prices of Solar PV modules:

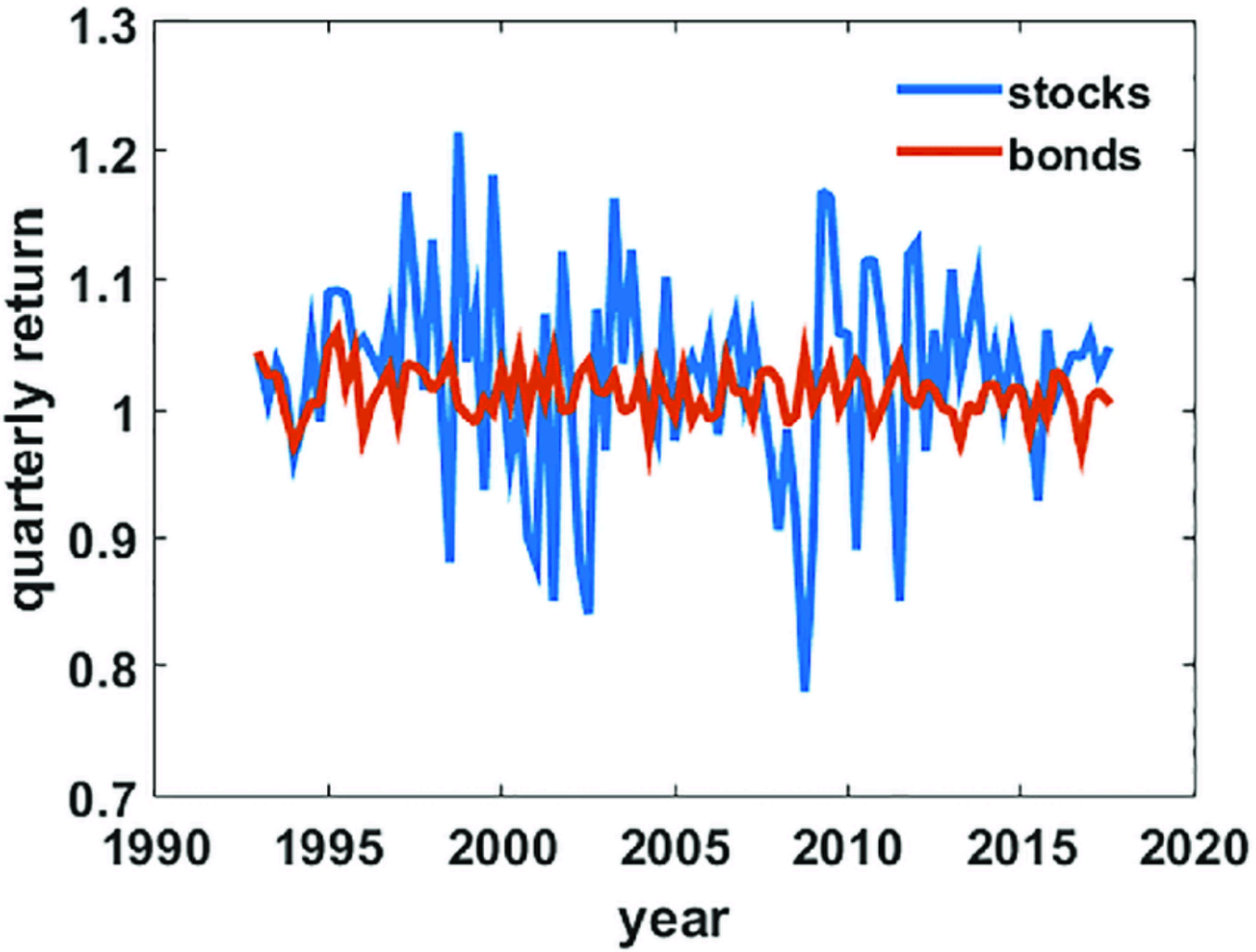
- If you look at the past (last year, last 5 years, last n years) you can observe a list of recorded values, each one corresponding to one day;
- If you think about the future, it is hard to forecast the values they will take. No deterministic phenomenon, but random!
- However, you can observe the past in order to try to forecast the future.
- Very loosely speaking, this is the objective of time series analysis.

Time series: look at the past and try to forecast the future

Suppose your point of observation is the dashed line: you can see one trajectory in the past, but of course more than one trajectory is possible in the future.



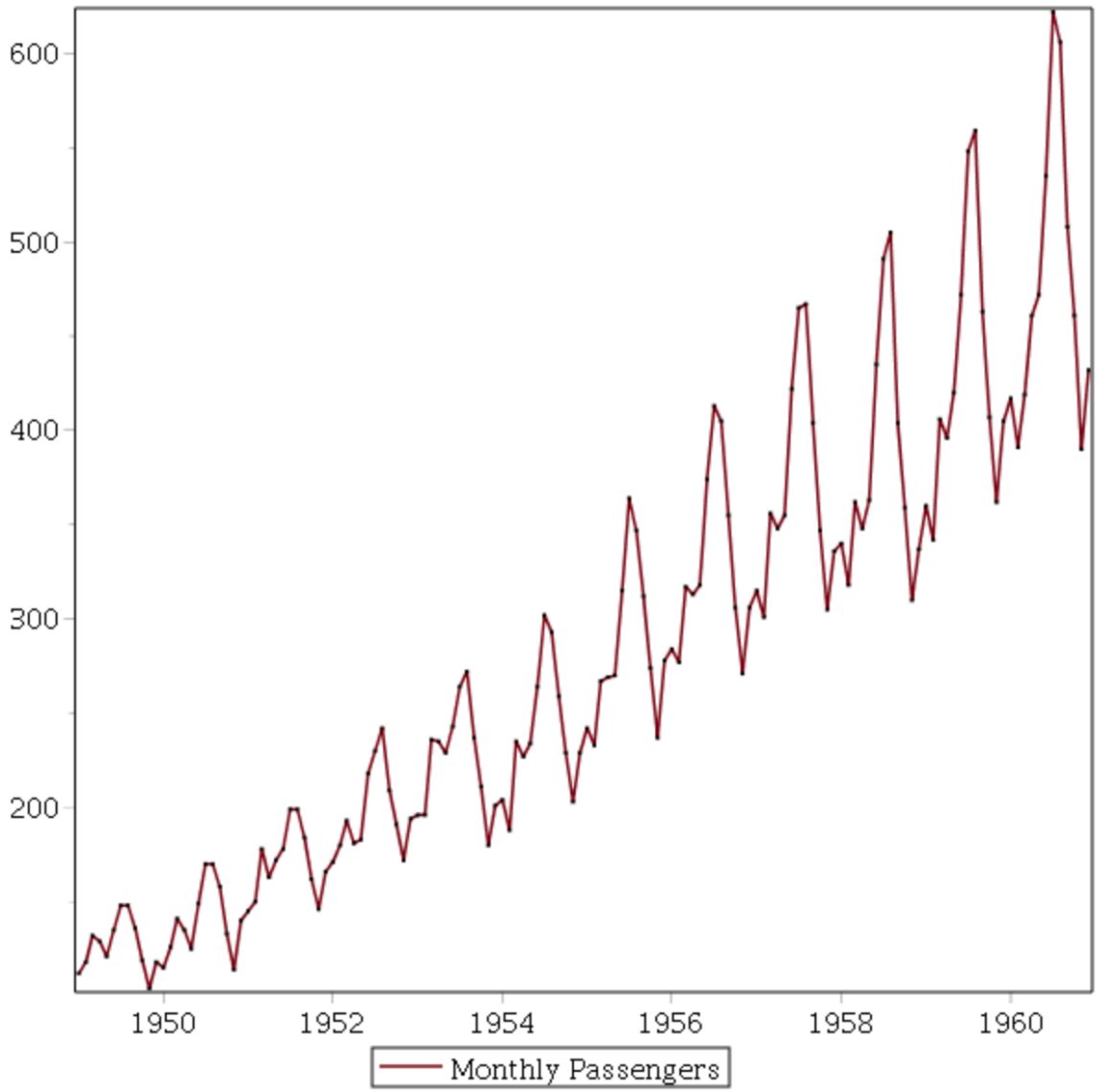
Example of a time series: stock and bond returns



Source: Guy Metcalfe, The Mathematics of Market Timing.

A second example: towards trend and seasonality

Number of monthly air passengers in thousands from 1949 until 1960. Source: Box, Jenkins, and Reinsel, *Time Series Analysis, Forecasting and Control*



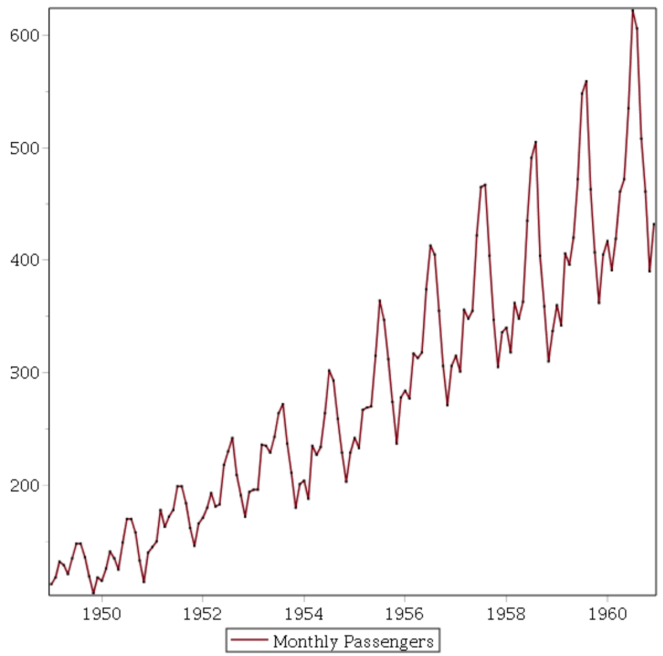
Some goals of time series analysis

When we analyze time series, we might want to:

- Forecast: predict future values of the time series;
- Identify patterns: it is assumed that the data consist of a systematic pattern (usually a set of identifiable components) and random noise (error) which usually makes the pattern difficult to identify.

Trend and seasonality: a first intuition

Think again at the plot we have seen few minutes ago. Here it is again.



One can clearly note:

- An increasing trend;
- A seasonality effect: there are some peaks at some nearly constant time intervals.

Trend, seasonality, cyclicality

Trend

Long-term increase or decrease in the data, which can be linear or non-linear.

Seasonality

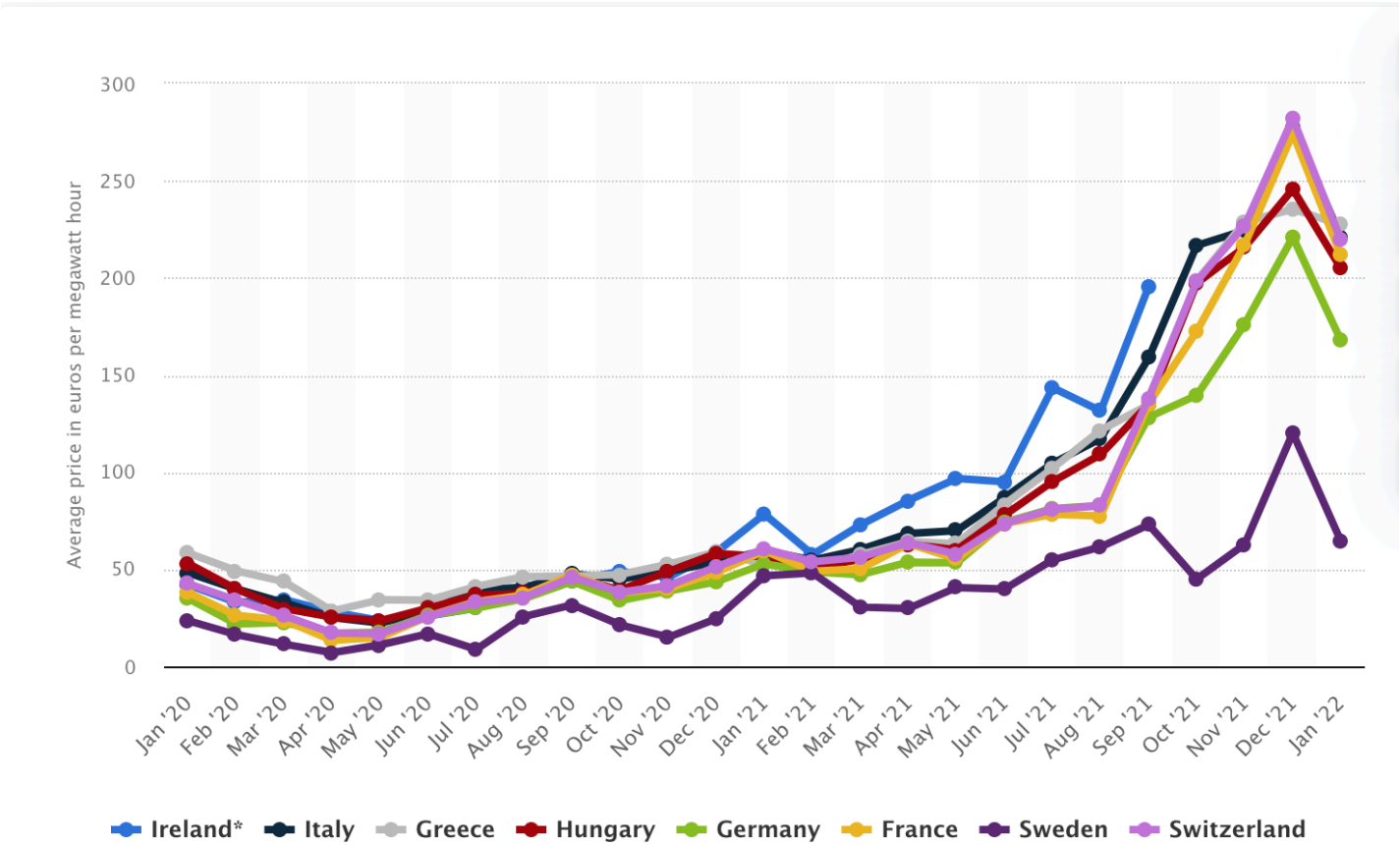
A seasonal pattern occurs when a time series exhibits rises and falls that are of a fixed frequency: for example, the time of the year or the day of the week.

Cyclicality

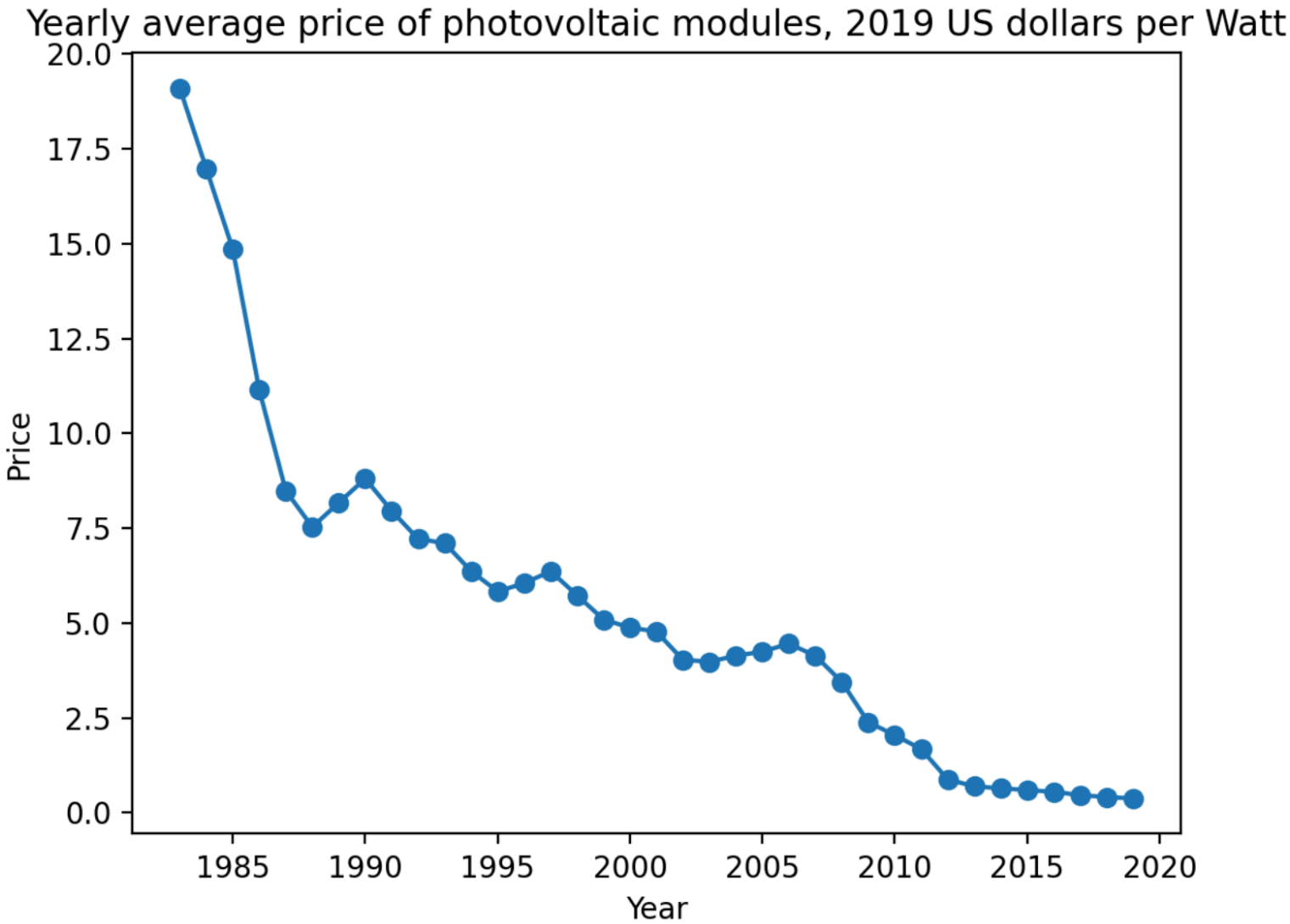
A cycle occurs when a time series exhibits rises and falls that are not of a fixed frequency.

Example of trend: electricity prices

Average monthly electricity wholesale prices in selected countries in the European Union (EU) from January 2020 to January 2022. Source: Statista 2022.



Example of trend: again renewable energy prices, Solar PV module prices



Example of trend renewable energy prices: Solar PV module prices

- Note: the previous plot and the following analysis from this dataset has been produced in Python.
- Commands besides downloading the data and save the prices as an array named `prices`:

```
import matplotlib.pyplot as plt
years = range(1983,2020)
plt.plot(years, prices,'-o')
plt.title('Yearly average price of photovoltaic modules,
2019 US dollars per Watt')
plt.xlabel('Year')
plt.ylabel('Price')
plt.show()
```

Components of a time series

A time series has four components:

- Trend
- Seasonality
- Cyclicality
- Random fluctuations.

One typically tries to express the value of the observations as a function of the four components, and isolate the four components.

How to estimate trend and seasonality?

- Plotting the prices data can already give you a nice insight about trend and seasonality.
- This can provide a very useful base for further analysis (for example, to have an idea of the number of periods in case of seasonality effects).
- What if you want to go more into details? The next topic is how to estimate the trend component with moving averages (MA).
- Once you estimate the trend, you can detect the seasonality from the detrend data just computing the average realizations for each period (for example month, day of the week, quarters, etc).

Moving averages smoothing

- The idea is to estimate the trend at t by computing the average of y close to t.
- Observations that are nearby in time are also likely to be close in value: the average eliminates some of the randomness in the data, leaving a smooth trend component.
- The number m of data (i.e., of times) that are taken into consideration in the average is called the *order* of the average. It is an odd number.
- A moving average of order m is denoted by m-MA.
- An m-MA can be written as

$$\hat{T}_t := \frac{1}{m} \sum_{j=-k}^{k} y_{t+j},$$

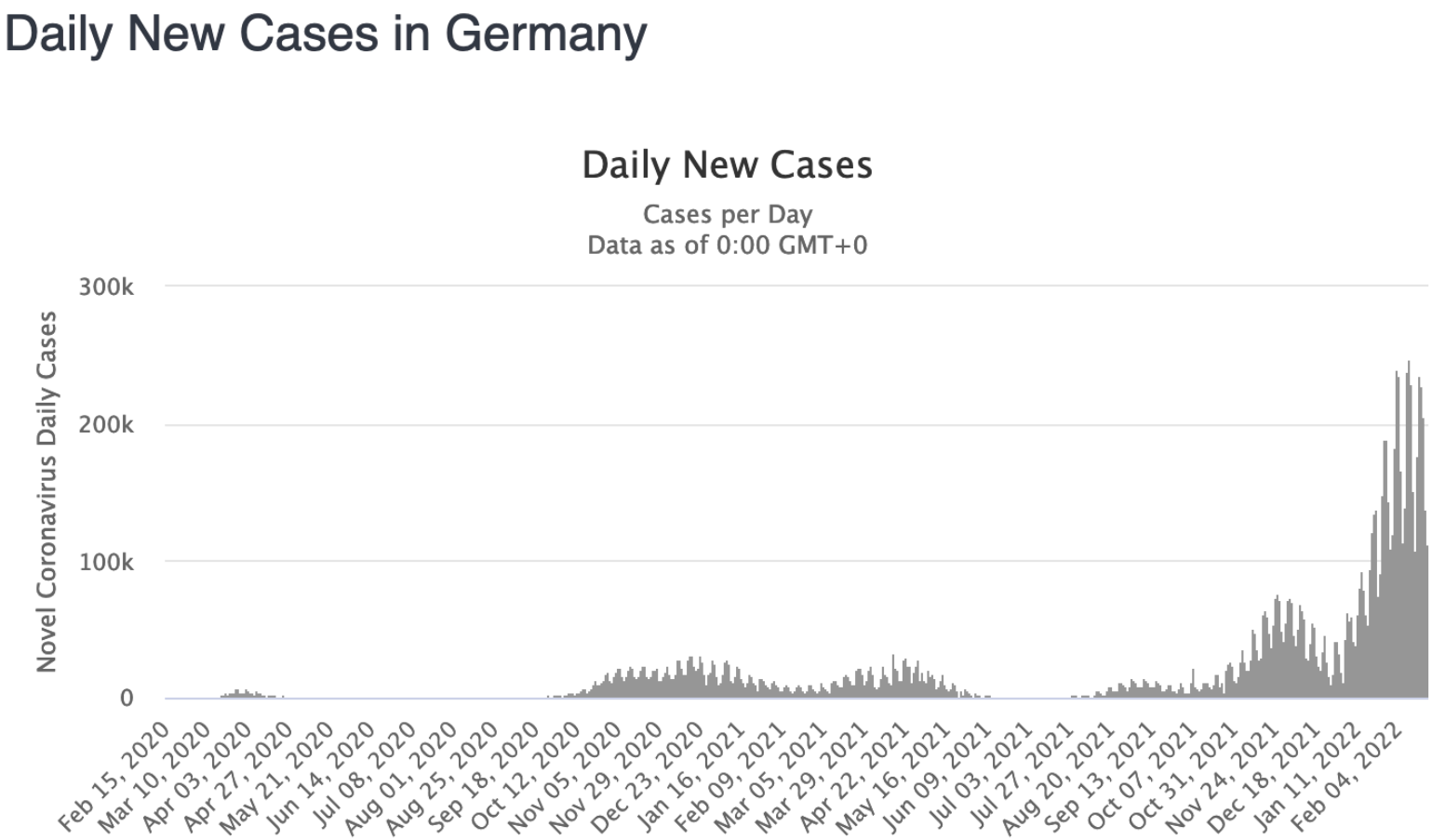
where $k = \frac{m-1}{2}$, so that $m = 2k + 1$.

How to choose the period m?

- Moving averages are used to estimate the trend, and in particular to get rid of seasonality effects.
- For this reason, it makes sense to choose m equal (or multiple of) the periods in the seasonality.
- For example, if you observe a phenomenon that exhibits a weekly seasonality effect, you can choose $m = 7$: this rules out the seasonality effect because at every day you consider the whole week of observations close to that day.

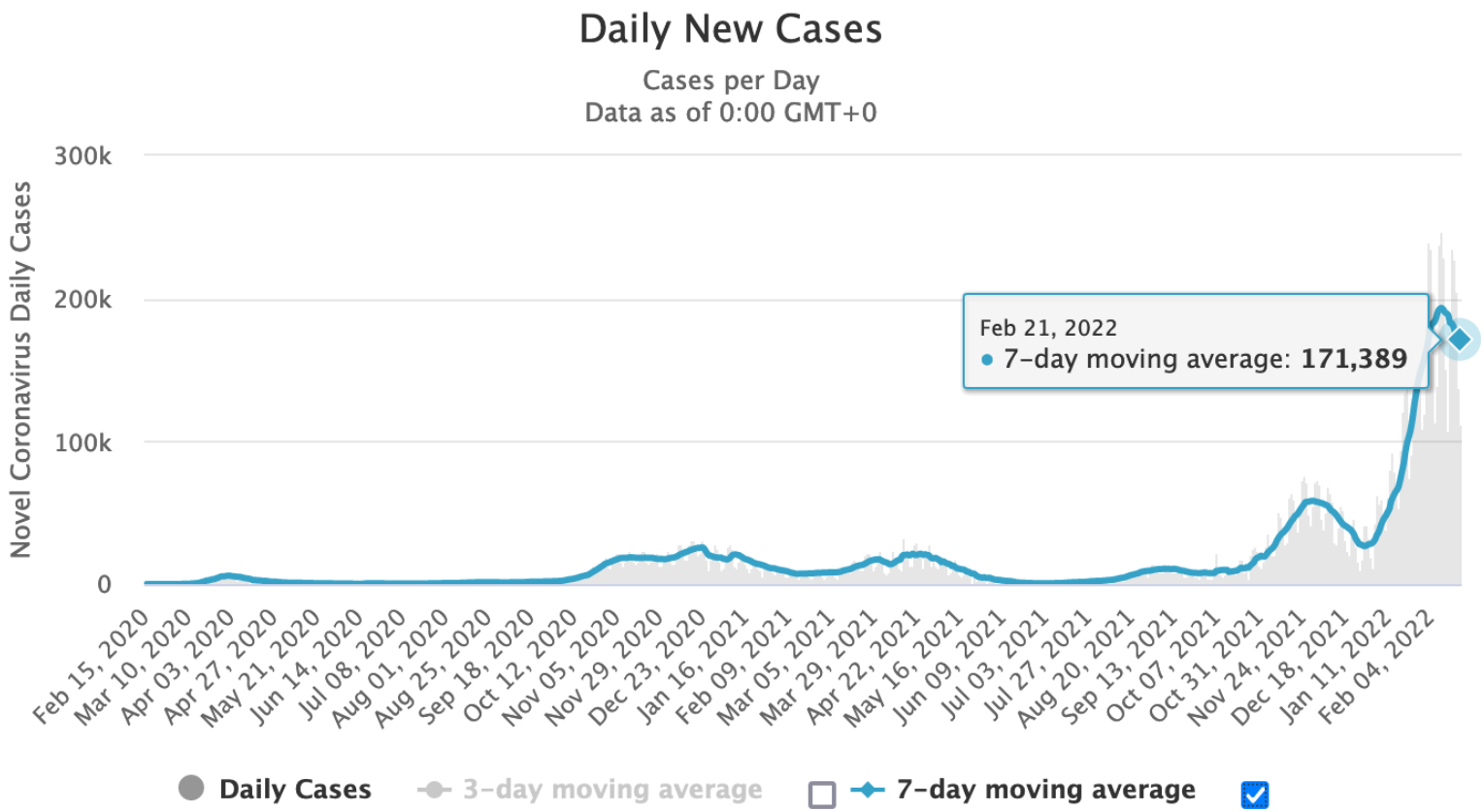
A well known example: Covid 19 daily new cases, Germany

Source: worldofmeters.info



Now, we take a 7-MA

Daily New Cases in Germany



$2 \times m$ moving averages

- We have said two things about the period m: it is odd, and it should be equal to the number of periods in case of seasonality effects.
- So: what if such a number n of periods is even?
- Idea: take $m = n + 1$, and make the first and last observation weight $1/2$ with respect to the others.
- Such MA are called $2 \times m$ MA.
- For example, in case of quarterly data, a nice option is to choose a 2×4 MA:

$$\hat{T}_t = \frac{1}{8} y_{t-2} + \frac{1}{4} y_{t-1} + \frac{1}{4} y_t + \frac{1}{4} y_{t+1} + \frac{1}{8} y_{t+2}$$

With this choice, all the quarters are weighted the same, because $t - 2$ corresponds to the same quarter as $t + 2$.

Weighted moving averages

- $2 \times m$ MA is the most used and famous example of weighted MAs.
- In general, a weighted m-MA can be written as

$$\hat{T}_t := \sum_{j=-k}^{k} a_j y_{t+j},$$

  where $k = \frac{m-1}{2}$ and $\sum_{j=-k}^{k} a_j = 1$.
- Choosing $a_j = \frac{1}{m}$ for all j we are back to the classical MA.

Autocorrelation of a time series

- *Autocorrelation* refers to the correlation of a time series with its own past and future values.
- It can be seen as the similarity between observations as a function of the time lag between them.
- It can also be called *lagged correlation*.
- Positive autocorrelation might be considered as a specific form of a tendency for a system to remain in the same state from one observation to the next.
- Negative autocorrelation is instead typical of mean-reverting processes.

Autocorrelation function

- Let $(y_t)_{t=1,\dots,N}$ be the values we observe of a time series.
- We introduce the autocorrelation function (ACF) r by defining

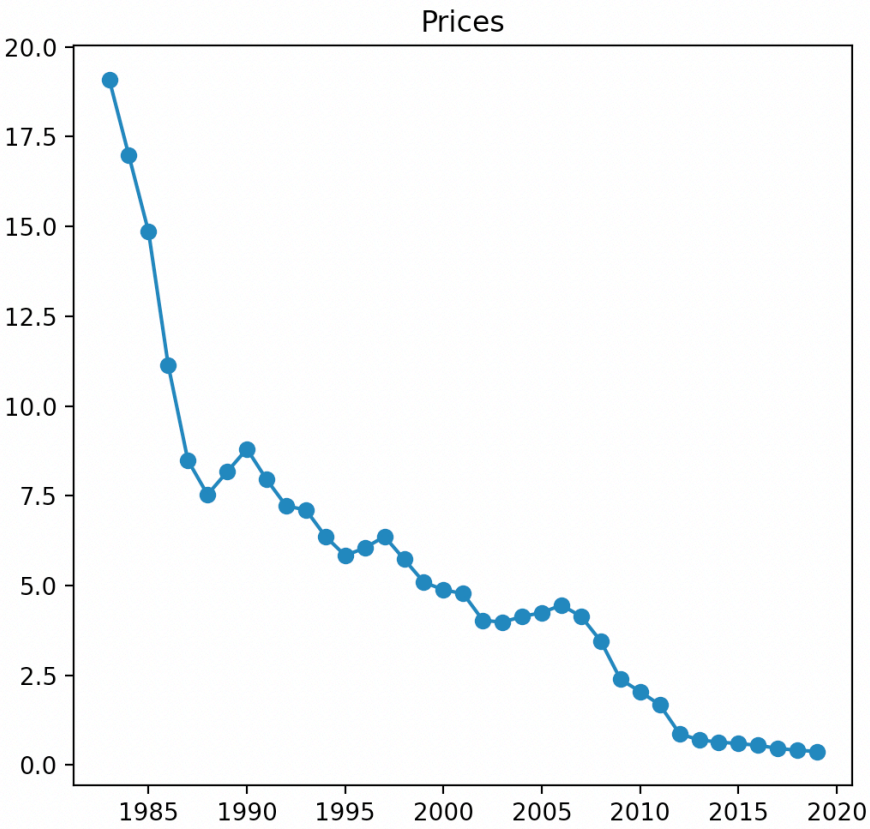
$$r(k) = \frac{\sum_{t=k+1}^{N}(y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum_{t=1}^{N}(y_t - \bar{y})^2},$$

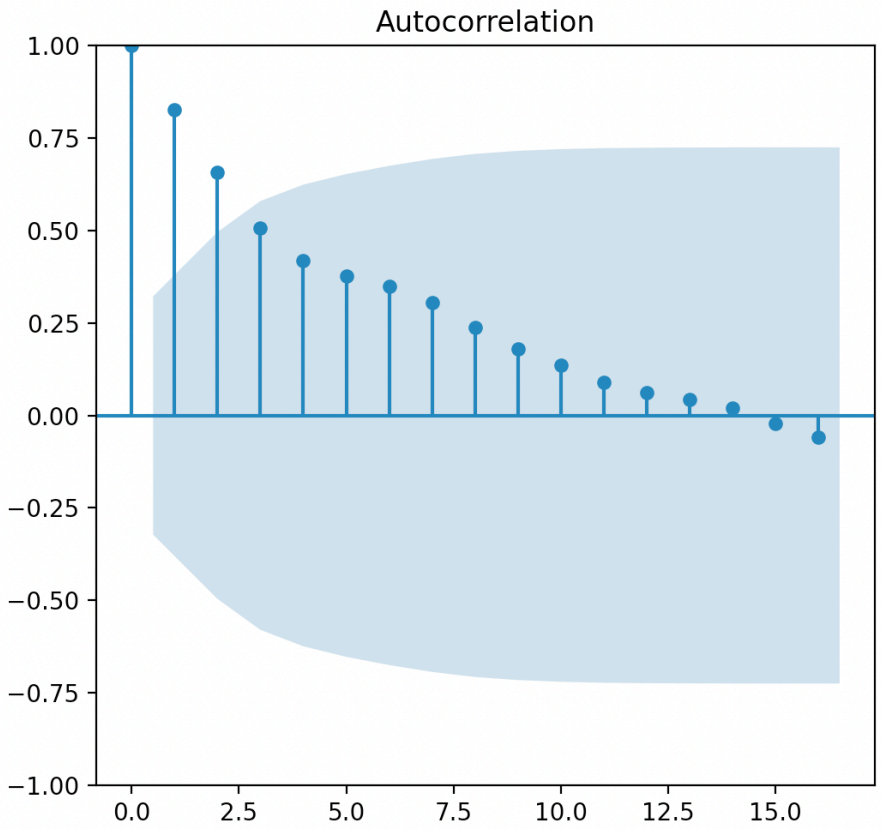
  where $\bar{y} = \frac{1}{N}\sum_{t=1}^{N} y_t$ is the average of the observed data.
- That is: $r(1)$ measures the correlation (i.e. the relationship) between y_t and y_{t-1}, $t = 2, \cdots, N$, $r(2)$ measures the correlation between y_t and y_{t-2}, $t = 3, \dots, N$, and so on.

Detecting trend and seasonality from the ACF

- When a time series has a trend, shorter lags have large positive correlations because observations close in time tend to have similar values.
- When a time series has seasonality, the autocorrelations are larger for lags at multiples of the seasonal frequency than for other lags.
- When a time series has both trend and seasonality, a mixture of both the effects can be observed.
- A good way to look at this is to plot the ACF.

Example: global average price of solar photovoltaic (PV) modules





Python code

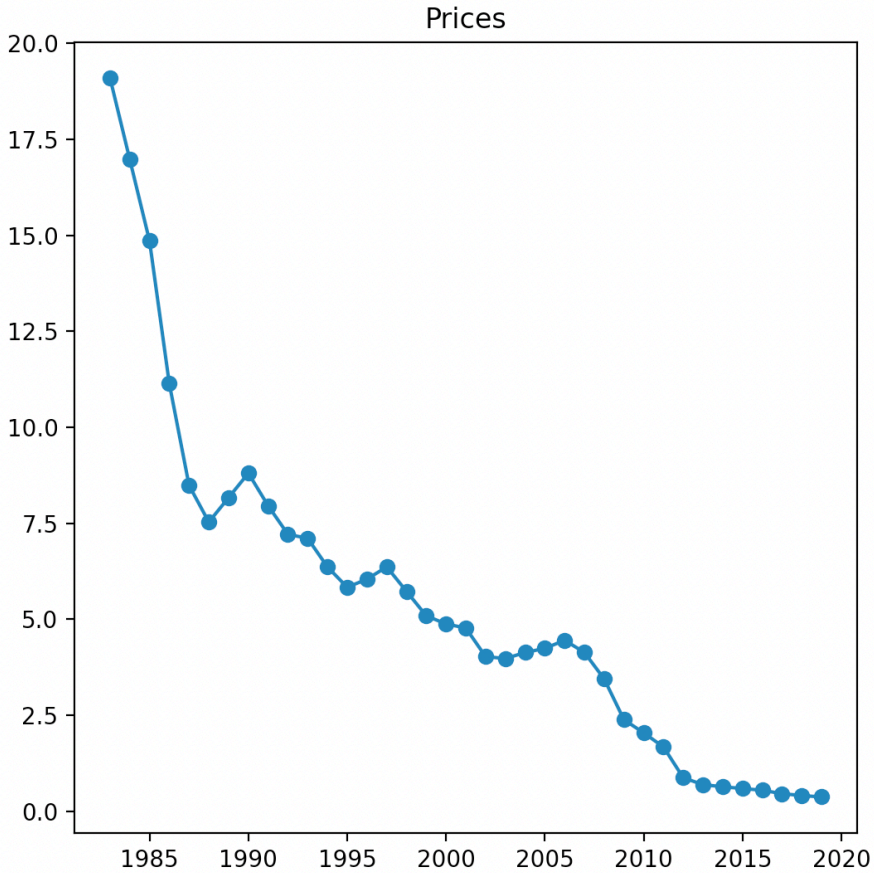
```
from statsmodels.graphics.tsaplots import plot_acf
fig, axes = plt.subplots(1, 2)
axes[0].plot(years, prices, '-o')
axes[0].set_title('Prices')
plot_acf(prices, ax=axes[1])
```

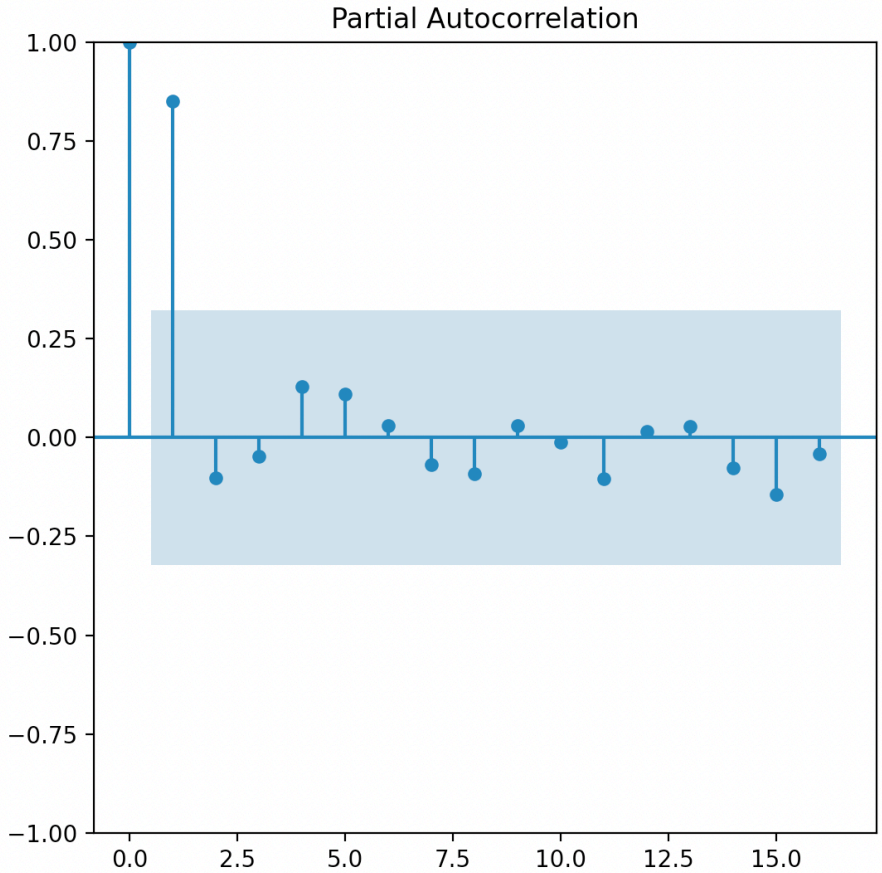
Partial autocorrelation function (PACF)

- Say we want to measure the correlation between y_t and y_{t-2}.
- If y_t and y_{t-1} are correlated, then y_{t-1} and y_{t-2} are also correlated (same lag).
- Then y_t and y_{t-2} might be correlated simply because they are both connected to y_{t-1}, rather than because of any new information contained in y_{t-2} that could be used in forecasting y_t!
- Goal of using the partial autocorrelation function is to solve this issue.
- A partial autocorrelation at lag k, $k \geq 2$, describes the correlation between y_t and y_{t-k}, after removing the effect of the correlation of y_t with $y_{t-1}, \ldots, y_{t-k+1}$.
- More formally, the PACF r^p is defined by

$$r^p(k) = \frac{\text{Covariance}(y_t, y_{t-k} | y_{t-1}, \ldots, y_{t-k+1})}{\sqrt{\text{Variance}(y_{t-k} | y_{t-1}, \ldots, y_{t-k+1}) \text{Variance}(y_t | y_{t-1}, \ldots, y_{t-k+1})}}$$

Example: global average price of solar photovoltaic (PV) modules





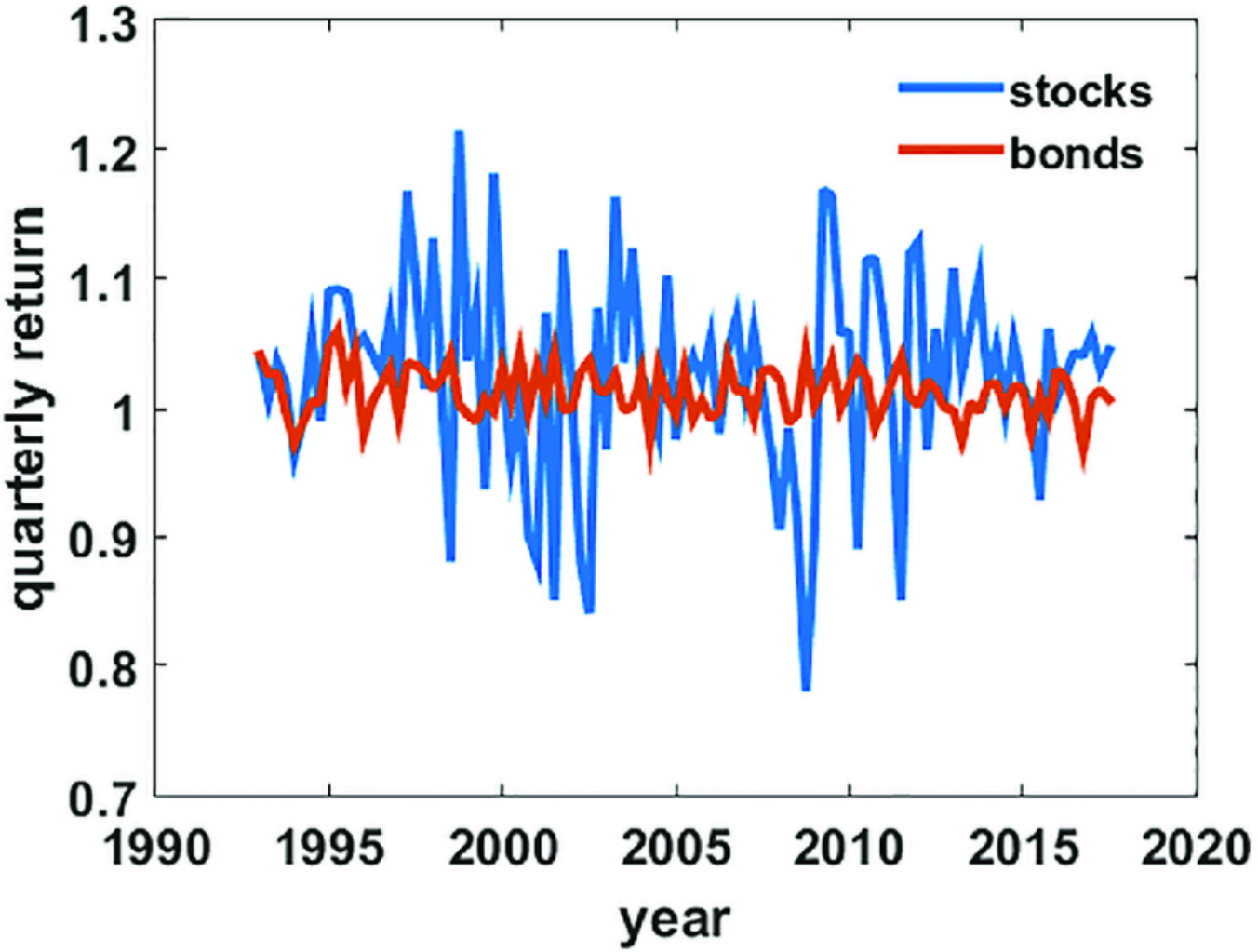
Python code

```
from statsmodels.graphics.tsaplots import plot_pacf
fig, axes = plt.subplots(1, 2)
axes[0].plot(years, prices, '-o')
axes[0].set_title('Prices')
plot_pacf(prices, ax=axes[1])
```

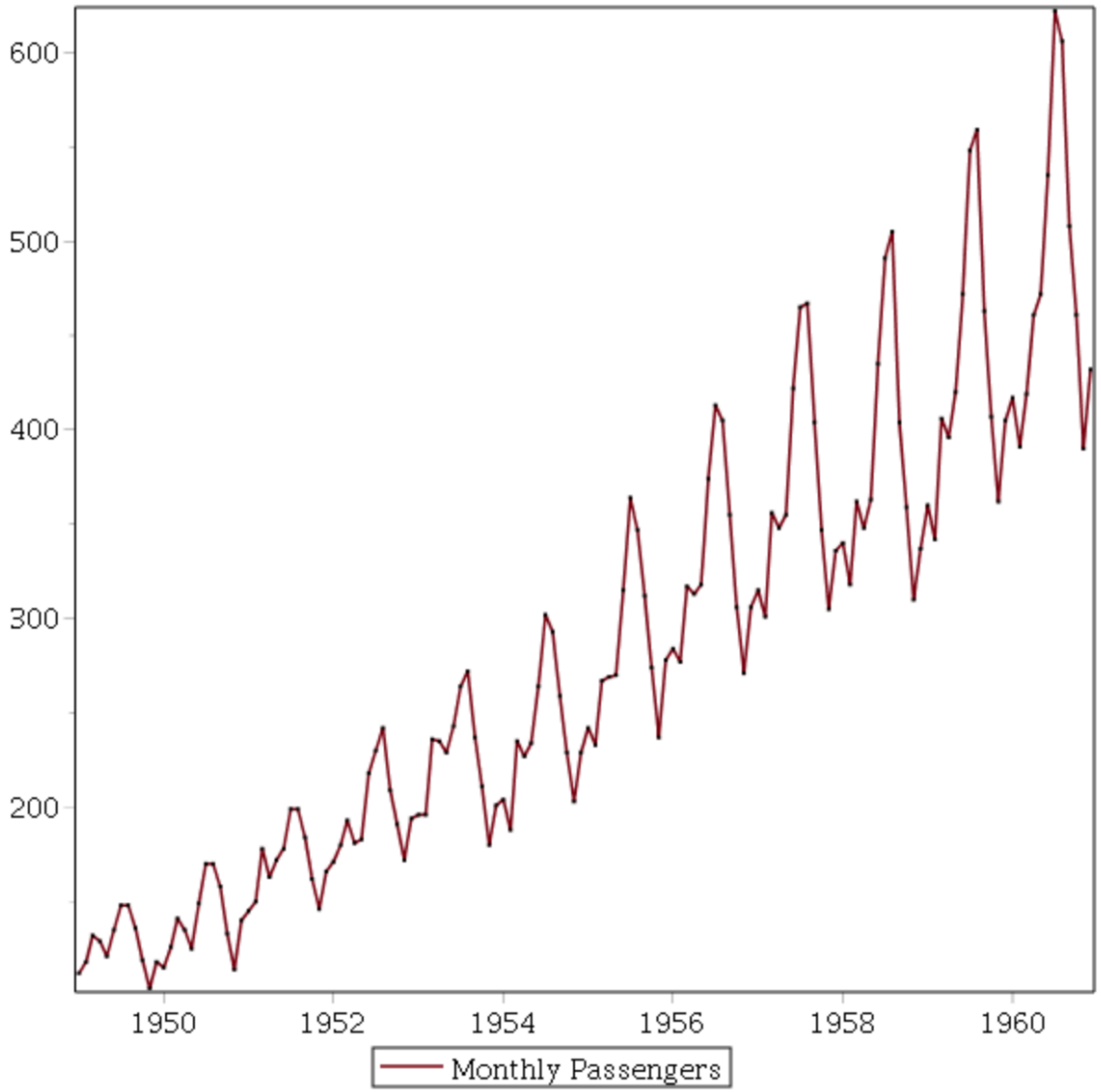
Definition of stationarity

- A time series is *stationary* if its statistical properties (for example mean, variance, autocorrelation, etc.) are all constant over time.
- A stationary time series has no predictable patterns.
- Time series with trends, or with seasonality, are not stationary.
- Note: a time series with cyclic behaviour (but with no trend or seasonality) is stationary! The cycles are not of a fixed period, so we cannot predict the future up and downs (on average).
- Time plots typically show a stationary time series to be roughly horizontal (although some cyclic behaviour is possible), with constant variance.

Stationary time series we have already seen: stock and bond returns



A non-stationary time series we have already seen: number of monthly airplanes passengers



Do not confuse non-stationarity with autocorrelation!

A time series can be stationary and have non-zero autocorrelation, and be non-stationary and have zero autocorrelation.

Example 1

Let X be a random variable with standard normal distribution $\mathcal{N}(0, 1)$. Then the series $y_t = X$ for any t:

- Is of course stationary;
- It has autocorrelation 1 for all the lags.

Example 2

Consider now a sequence of independent random variables X_i with normal distribution $\mathcal{N}(0, i)$ for any i. Then the series $y_t = X_t$ for any t:

- Is non-stationary (the variance depends on time);
- It has autocorrelation 0 for all the lags (the random variables are independent).

Unit roots

- Consider time series of the form

$$y_t = \alpha y_{t-1} + \epsilon_t, \tag{1}$$

  where ϵ is the so called *error term*, supposed to be stationary (white noise).
- A *unit root* is said to exist in a time series y if $\alpha = 1$ in (2).
- Such a series is stationary if and only if $|\alpha| < 1$ in (2).
- That is, if a unit root is present, the time series is not stationary.

Dickey-Fuller test

- Consider again
$$y_t = \alpha y_{t-1} + \epsilon_t.$$
- The Dickey-Fuller test is a way to determine whether the above time series has a unit root (in this case, it is not stationary). How does it work?
- From the equation above, we have
$$y_t - y_{t-1} = \alpha y_{t-1} + \epsilon_t - y_{t-1} = \beta y_{t-1} + \epsilon_t,$$
with $\beta = \alpha - 1$.
- Introduce now the first difference operator Δ defined by $\Delta y_t := y_t - y_{t-1}$. Then we have
$$\Delta y_t = \beta y_{t-1} + \epsilon_t,$$
and a unit root is present if $\beta = 0$.
- So the test boils down to a regression test with null hypothesis $\beta = 0$.

The three version of the Dickey-Fuller test

There are three most well known versions of the test:

1. Test with white noise only:
$$\Delta y_t = \beta y_{t-1} + \epsilon_t,$$
2. Test with constant:
$$\Delta y_t = c_1 + \beta y_{t-1} + \epsilon_t,$$
where $c_1 \in \mathbb{R}$.
3. Test with constant and deterministic time trend:
$$\Delta y_t = c_1 + c_2 t + \beta y_{t-1} + \epsilon_t,$$
where $c_1, c_2 \in \mathbb{R}$.
4. The Augmented Dickey-Fuller adds lagged differences Δy_{t-k} to these models.

Example: global average price of solar photovoltaic (PV) modules

- From the previous plots, it is clear that the time series of PV modules cannot be stationary: clear decreasing trend, significant autocorrelation values also for large lags.
- One can also perform an Augmented Dickey-Fuller unit root test with Python, to corroborate this strong feeling.
- The null hypothesis of the ADF test is that the time series is non-stationary. So, if the p-value of the test is bigger than the significance level (0.05) one we infer that the time series is indeed not stationary.
- We get a p-value of 0.644256: we cannot reject the null hypothesis!
- One can also look at the Augmented Dickey Fuller statistics, defined as $\frac{\hat{\gamma}}{SE(\hat{\gamma})}$: if it is smaller ("more negative") than some critical values, one can reject the null hypothesis. This is not true in our case.

Python code

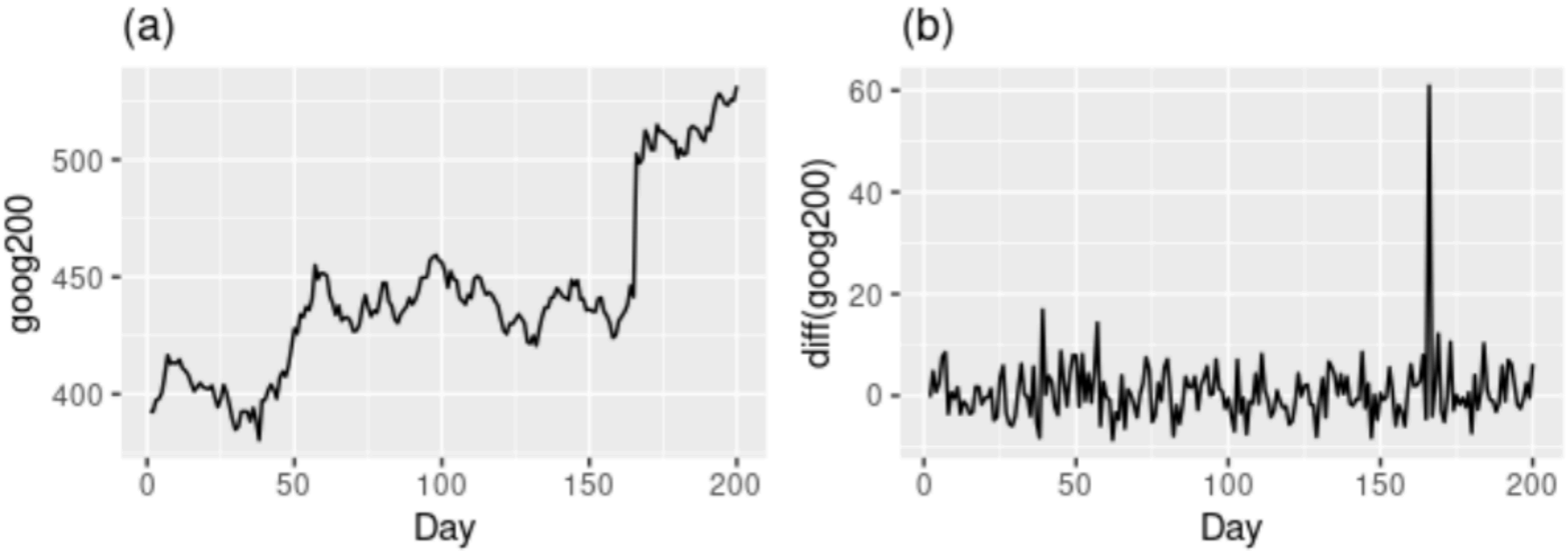
```
from statsmodels.tsa.stattools import adfuller
resultsAdfTest = adfuller(prices)
print('p-value: %f' % resultsAdfTest[1])
print()
print('ADF Statistic: %f' % resultsAdfTest[0])
for key,value in resultsAdfTest[4].items():
print('%f%s Critical Value: %f' %f' %f' % (key, value))
```

Why to transform a non-stationary time series into a stationary one

- A stationary series is relatively easy to predict: we simply predict that its statistical properties will be the same in the future as they have been in the past.
- For this reason, if we note that a time series is non-stationary, we want to apply some transformation in order to get a stationary time series.
- Then, once we predict the transformed series to have same properties in the future as in the past, we transform back and get the properties of the original time series.
- The most used transformation is called *differencing*.

Differencing: a first intuition

The Google stock price is non-stationary, but the daily changes are stationary. Source: Rob J Hyndman and George Athanasopoulos, *Forecasting: Principles and Practice (2nd ed)*.



Differencing

- Even if a time series is non-stationary, the series $\bar{y}$ of its changes, defined by

$$\bar{y}_t = y_{t+1} - y_t, \quad t = 1, \ldots, T-1,$$

  can be stationary.
- This is typically true if the original time series exhibits trend but not seasonality.
- If y has seasonality of period m but no trend, the series

$$\hat{y}_t = y_{t+m} - y_t, \quad t = 1, \ldots, T-m,$$

  is typically stationary.
- If y has both trend and seasonality, it is necessary to take both a seasonal difference and a first difference (i.e., $y_{t+1} - y_t$ as above) to obtain stationary data.
- In case of strong trends, a double differencing might be necessary.

Definition of autoregressive model

- Idea: we forecast the next value of a time series using a linear combination of its past values.
- Formalized:

$$y_t = c + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \cdots + \varphi_p y_{t-p} + \epsilon_t, \tag{2}$$

  where:
  - p is called the *order* of the autoregressive model;
  - the parameters $\varphi_1, \varphi_2, \ldots, \varphi_p \in \mathbb{R}$, possibly satisfying some stationarity conditions (more on this later);
  - ϵ_t is the error component, the realization at time t of a white noise time series: a time series that shows no autocorrelation.
- We refer to y defined in (2) as an AR(p) model: autoregressive model of order p.
- We usually suppose autoregressive models to be stationary.
- In this case, some constraints on the values of the parameters are required.

AR(1) models

- An AR(1) model is defined as

$$y_t = c + \varphi_1 y_{t-1} + \epsilon_t.$$

- It is stationary if and only if $-1 < \varphi_1 < 1$.
- The behaviour of an AR(1) model depends on φ_1 and c. In particular:
  - if $\varphi_1 = 0$, it is white noise (no autocorrelation);
  - if $\varphi_1 = 1$, it equivalent to a random walk (with drift if $c \neq 0$);
  - if $\varphi < 0$, it is mean reverting: it oscillates around the mean.

AR(2) models

- An AR(2) model is defined as

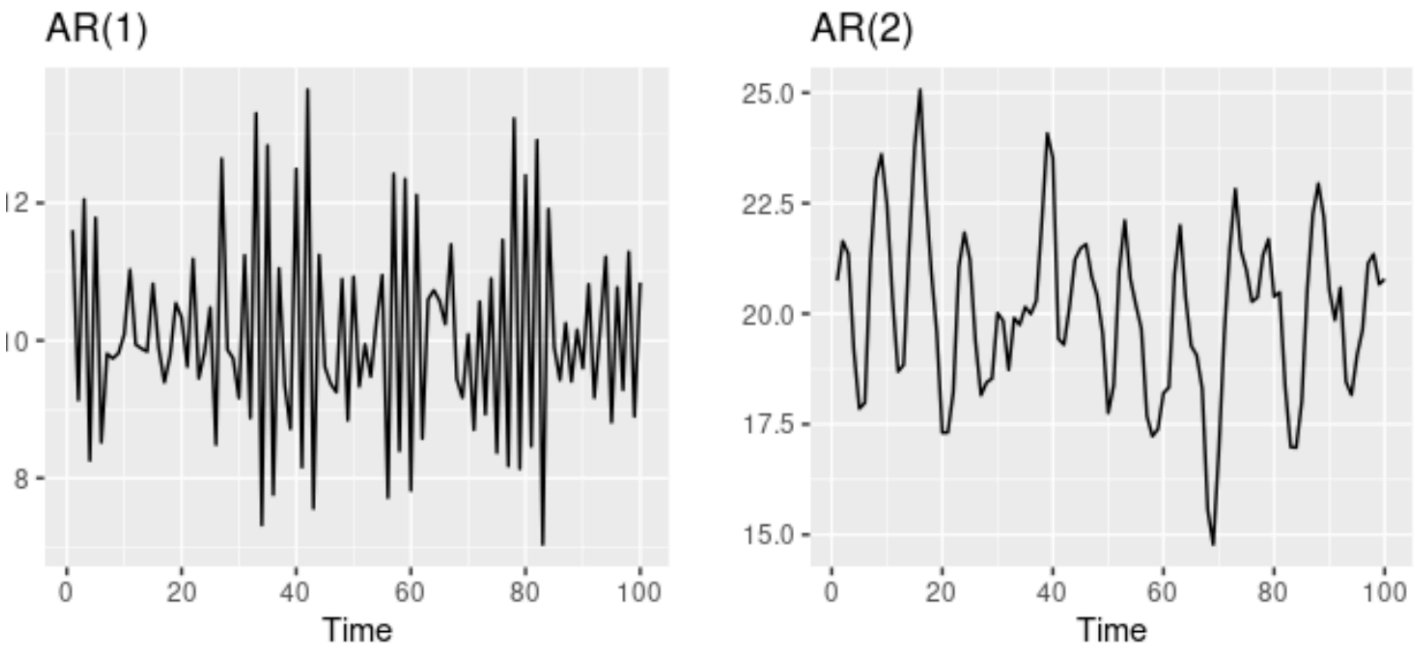
$$y_t = c + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \epsilon_t.$$

- It is stationary if and only if $-1 < \varphi_2 < 1$, $\varphi_1 + \varphi_2 < 1$, $\varphi_2 - \varphi_1 < 1$.
- Stationarity conditions for orders strictly bigger than 2 are much more complex to derive. They can be computed with the help of the most common programming tools, like R and Python.

Plot of an AR(1) and of an AR(2) model

Source: Rob J Hyndman and George Athanasopoulos, *Forecasting: Principles and Practice (2nd ed)*.

- On the left, AR(1) with $y_t = 18 - 0.8y_{t-1} + \epsilon_t$.
- On the right, AR(2) with $y_t = 8 + 1.3y_{t-1} - 0.7y_{t-2} + \epsilon_t$.



Definition of moving average model

- Idea: we forecast the next value of a time series using a linear combination of the past errors.
- Formalized:

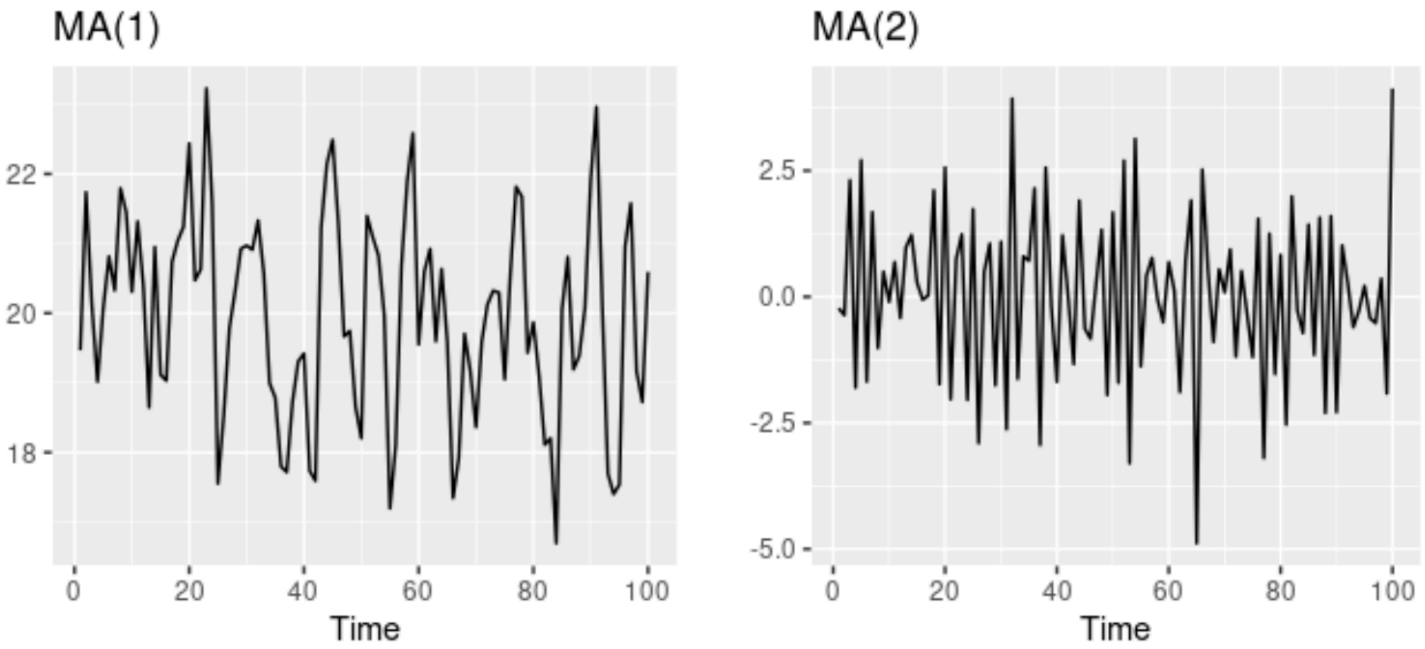
$$y_t = c + \epsilon_t + \psi_1 \epsilon_{t-1} + \psi_2 \epsilon_{t-2} + \cdots + \psi_q \epsilon_{t-q}, \tag{3}$$

  where:
  - q is the *order* of the moving average model;
  - the parameters $\psi_1, \psi_2, \ldots, \psi_q \in \mathbb{R}$, maybe satisfying some conditions (invertibility conditions);
  - $\epsilon_t, \epsilon_{t-1}, \ldots, \epsilon_{t-q}$ are realizations of a white noise time series: a time series that shows no autocorrelation.
- We refer to y defined in (3) as a MA(q) model: a moving average model of order q.
- Note that each value of y can be thought as a weighted moving average of the past q errors.

Plot of a MA(1) and of an MA(2) model

Source: Rob J Hyndman and George Athanasopoulos, *Forecasting: Principles and Practice (2nd ed)*.

- On the left, MA(1) with $y_t = 20 + \epsilon_t + 0.8\epsilon_{t-1}$.
- On the right, MA(2) with $y_t = \epsilon_t - \epsilon_{t-1} + 0.8\epsilon_{t-2}$.



Non-seasonal ARIMA models: a combination between MA and AR models with differencing

- ARIMA is an acronym for Auto Regressive Integrated Moving Average: it is a model for the differenced time series y' where $y'_t := y_t - y_{t-d}$, $d \geq 1$.
- In particular, an ARIMA(p, d, q) model can be written as

$$y'_t = c + \varphi_1 y'_{t-1} + \cdots + \varphi_p y'_{t-p} + \psi_1 \epsilon_{t-1} + \cdots + \psi_q \epsilon_{t-q} + \epsilon_t,$$

  with $y'_t := y_t - y_{t-d}$.
- Note here that on the right hand side there are both lagged values of y and lagged errors. In particular, we have:
  - an autoregressive part with order p;
  - a moving average part with order q.
- The same stationarity and invertibility conditions that are used for autoregressive and moving average models also apply to an ARIMA model.

Particular cases of non-seasonal ARIMA models

Have in mind that an ARIMA(p, d, q) model can be written as

$$y'_t = c + \varphi_1 y'_{t-1} + \cdots + \varphi_p y'_{t-p} + \psi_1 \epsilon_{t-1} + \cdots + \psi_q \epsilon_{t-q} + \epsilon_t.$$

Some examples:

- ARIMA$(0, 0, 0)$: white noise
- ARIMA$(0, 1, 0)$: random walk (with drift if $c \neq 0$)
- ARIMA$(p, 0, 0)$: autoregressive of order p
- ARIMA$(0, 0, q)$: moving average of order q.

Seasonal ARIMA models

- We have only considered non-seasonal data when looking at ARIMA models.
- But what if we have to handle seasonality?
- We can include seasonal terms in the ARIMA models we have seen so far.
- A seasonal ARIMA (SARIMA) model is denoted by ARIMA$(p, d, q)(P, D, Q)_m$:
  - m is the seasonality period;
  - (P, D, Q) are the equivalent of (p, d, q) in the seasonality term.

SARIMA models, formalized

How to write a SARIMA model

An ARIMA$(p, d, q)(P, D, Q)_m$ model can be written as

$$\Phi(B^m)\varphi(B)\left(y_t - y_{t-d} - (y_{t-m} - y_{t-d-m})\right) = \Psi(B^m)\psi(B)\epsilon_t,$$

where:

- $B^m y_t = y_{t-m}$ and $By_t = y_{t-1}$;
- $\Phi(B^m) = 1 - \Phi_1 B^m - \cdots - \Phi_P B^{Pm}$;
- $\varphi(B) = 1 - \varphi_1 B - \cdots - \varphi_q B^p$;
- $\Psi(B^m) = 1 - \Psi_1 B^m - \cdots - \Psi_Q B^{Qm}$;
- $\psi(B) = 1 - \psi_1 B - \cdots - \psi_q B^q$;

Example

An ARIMA$(0, 1, 1)(0, 1, 1)_4$ model can be written as

$$\begin{aligned} y_t - y_{t-1} - (y_{t-4} - y_{t-5}) &= (1 - \Psi_1 B^4)(1 - \psi_1 B)\epsilon_t \\ &= (1 - \Psi_1 B^4)(\epsilon_t - \psi_1 \epsilon_{t-1}) \\ &= \epsilon_t - \psi_1 \epsilon_{t-1} - \Psi_1 \epsilon_{t-4} + \psi_1 \Psi_1 \epsilon_{t-5}. \end{aligned}$$

Exogenous variables

- It is possible to include in our model also the (linear) effect of r exogenous variables.
- That is, we add to our model the linear term

$$\sum_{k=1}^{r} \alpha_k X_{tk},$$

where X_{tk} is the value at time t of the k-th exogenous variable.

Our goals

- Let's say we want to model a time series y with an ARIMA(p, d, q) model, i.e.,

$$y'_t = c + \varphi_1 y'_{t-1} + \cdots + \varphi_p y'_{t-p} + \psi_1 \epsilon_{t-1} + \cdots + \psi_q \epsilon_{t-q} + \epsilon_t,$$

  with $y'_t := y_t - y_{t-d}$.
- We have to get values for:
  - the order of differencing d;
  - the orders p and q of the autoregressive and moving average parts, respectively;
  - the parameters φ_j, $j = 1, \ldots, p$ and ψ_i, $i = 1, \ldots, q$.
- We want to do this in such a way that fits well our time series.
- The scheme is:
  - First get d (looking at autocorrelations);
  - then get p and q (ACF and PACF plots, AIC criterion);
  - finally get the parameters (Maximum likelihood estimation).
- More details in the very next slides.

Identifying the differencing order

- We can let d be the lowest order of differencing such that the differenced series is stationary.
- In particular:
  - the original series is stationary $\longrightarrow$ no order of differencing is needed;
  - the original series has a constant average trend $\longrightarrow$ one order of differencing is needed;
  - the original series has a time varying average trend $\longrightarrow$ two orders of differencing are needed;
  - usually, no more than two orders of differencing are needed.
- An indicator of stationarity can be an autocorrelation function plot which decays fairly rapidly to zero, either from above or below.
- If instead the autocorrelations of the differenced time series are positive out to a high number of lags (say 7/8 or more), then an higher order of differencing is needed.
- Practical rule of thumb: If the first lag autocorrelation is smaller than -0.5 this may mean the series has been *overdifferenced*.

Identifying p and q

We see two ways to estimate the orders p (autoregressive part) and q (moving average part) of our ARIMA model:

- ACF and PACF plots;
- AIC/BIC criterion.

Identifying p and q with ACF and PACF plots

- Recall that:
  - An ACF plot shows the correlations between y and its k-lagged values for $k = 1, 2, \ldots$.
  - A PACF plot shows the correlations between y and its k-lagged values for $k = 1, 2, \ldots$, after removing the effects of lags $1, 2, \ldots, k-1$.
- The ACF and PACF plots can be helpful in determining the value of p or q if the data are from an ARIMA$(p, d, 0)$ or ARIMA$(0, d, q)$ model.
- We can say that y is an ARIMA$(p, d, 0)$ if we see that:
  - the ACF is exponentially decaying or sinusoidal;
  - there is a significant spike at lag p in the PACF, but none beyond lag p.
- We can say that y is an ARIMA$(0, d, q)$ if we see that:
  - the PACF is exponentially decaying or sinusoidal;
  - there is a significant spike at lag q in the ACF, but none beyond lag p.

AIC and BIC

Definition

The Akaike information criterion (AIC) value for a general model is

$$AIC = 2k - 2\ln(L),$$

where k is the number of parameters in the model and L is the value of the likelihood function of the model: the joint probability of the observed data as a function of the model's parameters.

Definition

The Bayesian information criterion (BIC) value for a general model is

$$BIC = k\ln(n) - 2\ln(L),$$

where k is the number of parameters in the model, n the total number of data we have for the model and L is the value of the likelihood function of the model.

Idea

Choose the model that minimizes AIC or BIC.

Choosing p and q with AIC or BIC

AIC for ARIMA models

The AIC value for an ARIMA(p, d, q) model is

$$AIC(p, q) = 2(p + q + k + 1) - 2\ln(L),$$

where $k = 1$ if $c \neq 0$, $k = 0$ if $c = 0$.

BIC for ARIMA models

The BIC value for an ARIMA(p, d, q) model where a number T of values is observed is

$$BIC(p, q) = (p + q + k + 1)\ln(T) - 2\ln(L),$$

where $k = 1$ if $c \neq 0$, $k = 0$ if $c = 0$.

Estimating the parameters of ARIMA(p, d, q) models

- As a final step, one needs to estimate the parameters $c, \varphi_1, \ldots, \varphi_p, \psi_1, \ldots, \psi_q$.
- These are typically estimated by maximum likelihood estimation (MLE).
- This technique finds the values of the parameters which maximise the probability of obtaining the data that we observe.

Testing an ARIMA model

- Once we have identified the values of the order of an ARIMA model (i.e., p, d, q) and its parameters, we have to test the forecasts we get with our model.
- We can do this by inspecting how well a model performs on new data, i.e., data that were not used when fitting the model.
- The size of the test set is typically about 20% of the total sample.
- The *forecast error* at time t is defined as

$$e_t := y_t - \hat{y}_t,$$

where y_t is the true value of the time series at t and $\hat{y}_t$ is the value forecasted by the ARIMA model.

- Call $\mathcal{T}^{test}$ the train set and let $|\mathcal{T}^{test}|$ be the number of its elements. Then the two most common measures used to test an ARIMA model are:

$$\text{mean absolute error: } \frac{1}{|\mathcal{T}^{test}|} \sum_{t \in \mathcal{T}^{test}} |e_t|$$

and

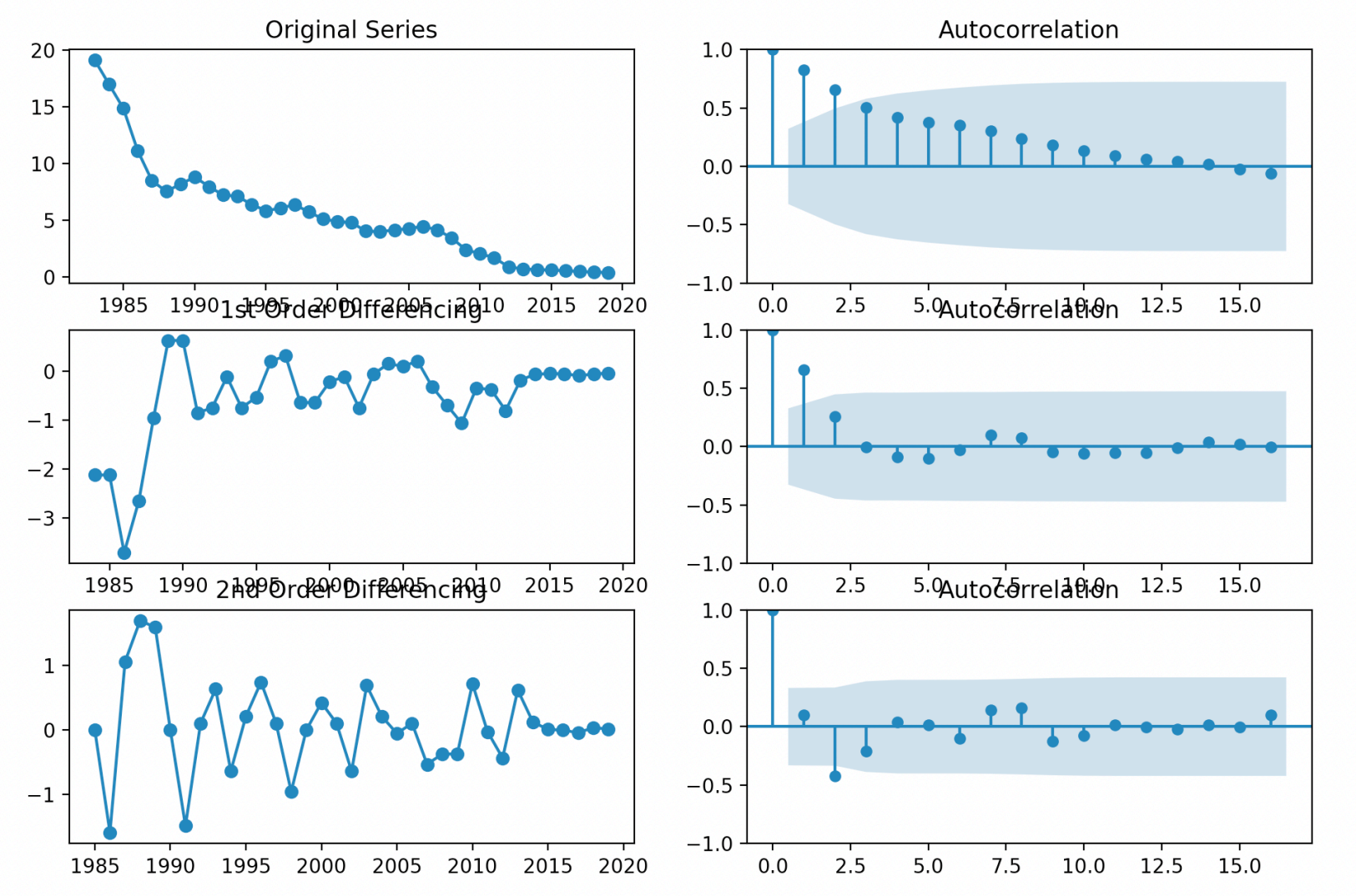
$$\text{root mean squared error: } \sqrt{\frac{1}{|\mathcal{T}^{test}|} \sum_{t \in \mathcal{T}^{test}} (e_t)^2}.$$

Example: Solar PV module prices

- We now want to apply the procedure described before to the data of the solar PV module prices.
- As we have already seen, the series is not stationary (decreasing trend, significantly positive autocorrelations also for large lags in the ACF plot, ADF test).
- Then, we have to come up with a strictly positive differencing order d.
- We look at the ACF plots: remember we want to choose the smallest order for which the differenced series looks *fairly* stationary.
- That is, the smallest order for which the differenced series has significantly positive autocorrelations only until small lags.

Example: Solar PV module prices. Choosing the differencing order

We choose $d = 1$: the third lag is already very close to zero. Another possible choice would have also been $d = 2$.

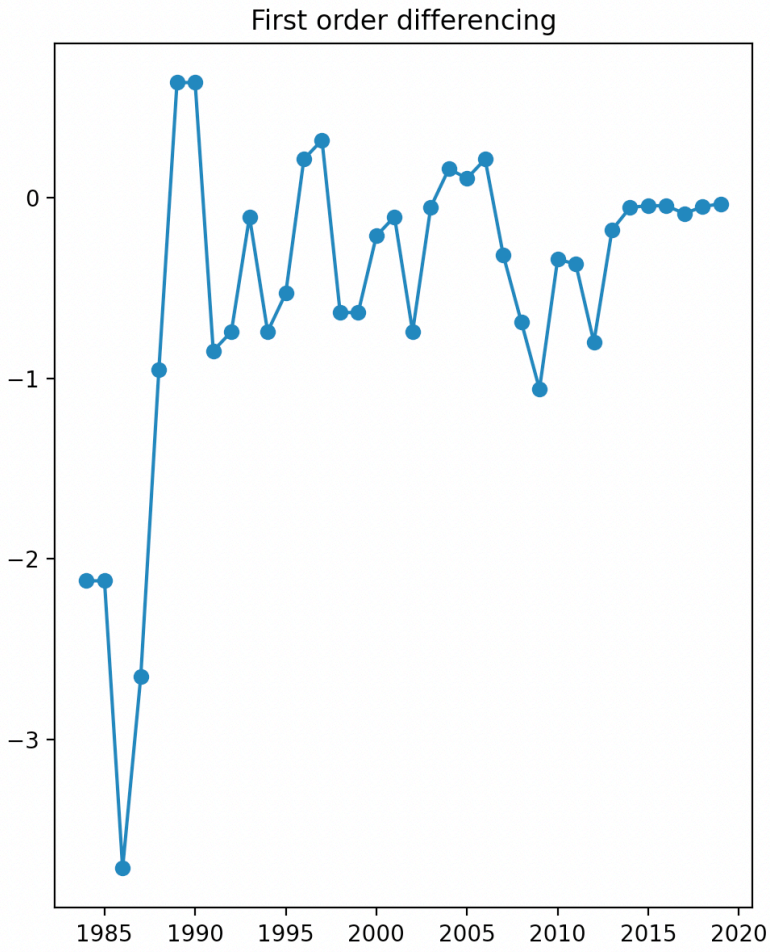


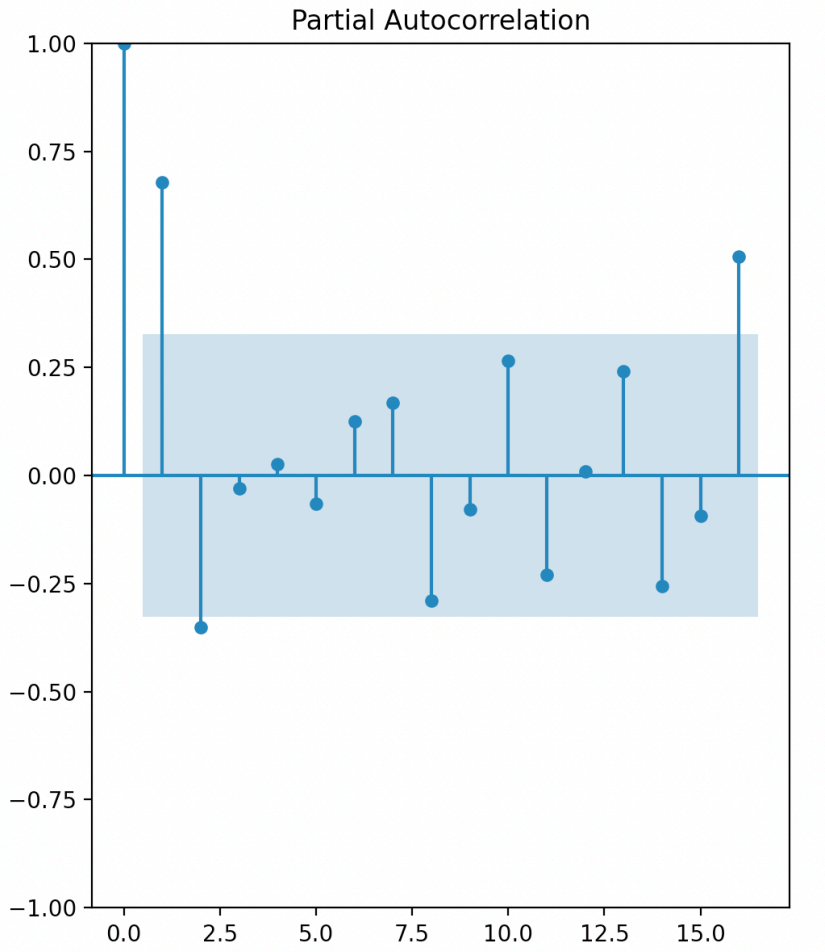
Example: Solar PV module prices. Choosing the differencing order. Code

```
from statsmodels.graphics.tsaplots import plot_acf
import matplotlib.pyplot as plt
years = range(1983,2020)
yearsForFirstDiff = range(1984,2020)
yearsForSecondDiff = range(1985,2020)
fig, axes = plt.subplots(3, 2)
axes[0, 0].plot(years, prices)
axes[0, 0].set_title('Original Series')
plot_acf(prices, ax=axes[0, 1])
axes[1, 0].plot(yearsForFirstDiff, diff(prices))
axes[1, 0].set_title('1st Order Differencing')
plot_acf(diff(prices), ax=axes[1, 1])
axes[2, 0].plot(yearsForSecondDiff, diff(diff(prices)))
axes[2, 0].set_title('2nd Order Differencing')
plot_acf(diff(diff(prices)), ax=axes[2, 1])
plt.show()
```

Example: Solar PV module prices. Choosing the autoregressive order p

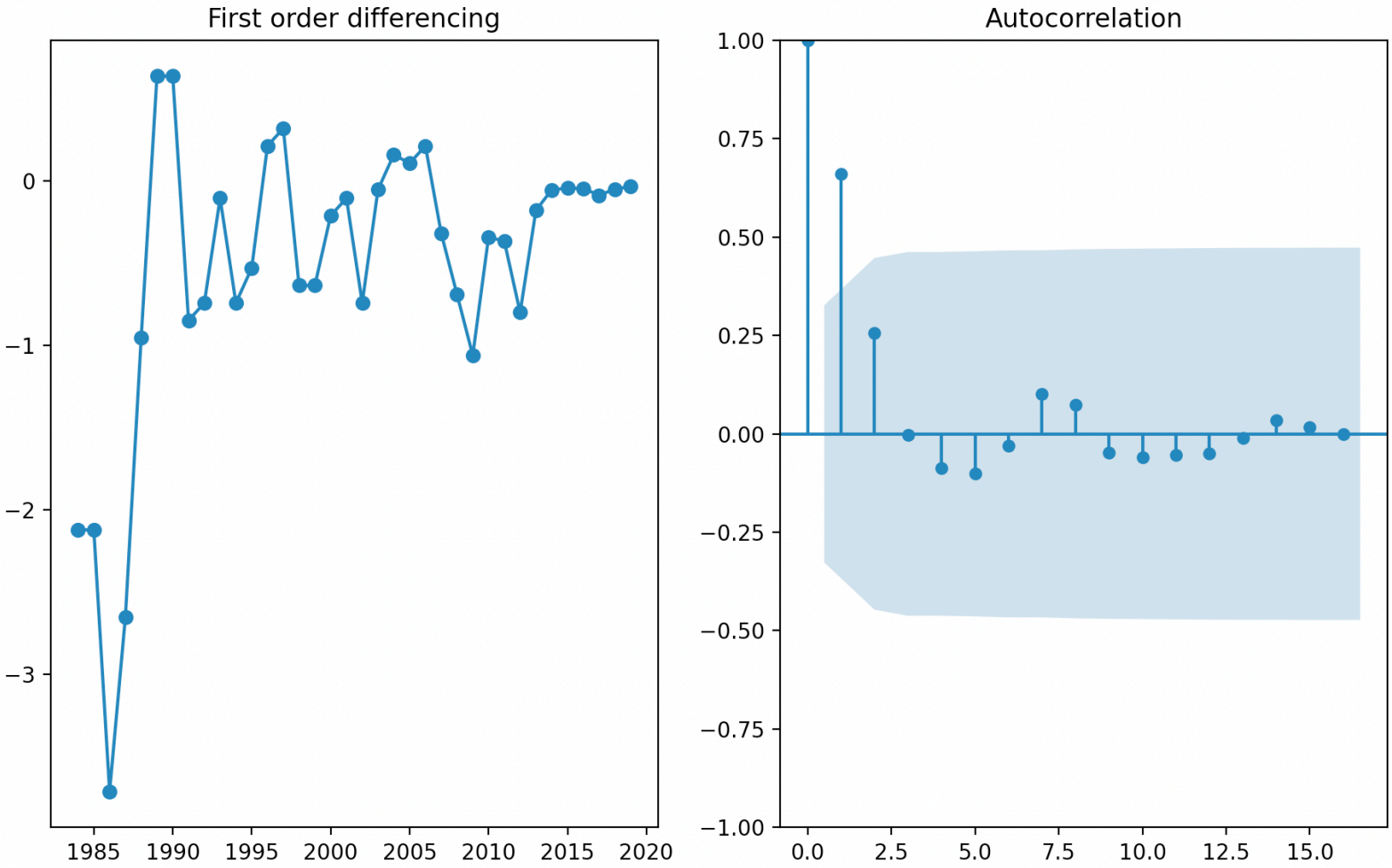
Look at the PACF plot of the differenced time series: there is a significant spike at lag 1, but none beyond lag 1: we choose $p = 1$.





Example: Solar PV module prices. Choosing the MA order q

Look at the ACF plot of the differenced time series: there is a significant spike at lag 1, but none beyond lag 1 (the second one is below the significance area border): we choose $q = 1$.



Example: Solar PV module prices. Recap and AIC test

- For our dataset on the Solar PV module prices, we have then chosen an ARIMA$(1, 1, 1)$ model.
- Actually, we can try to let Python itself find our parameters d, p, q, via AIC criterion.
- We can do this by typing `import pmdarima as pmd` and then

```
autoarima_model = pmd.auto_arima(prices,start_p=1,
                  start_q=1,trace=True)
```

- Here `start_p, start_q` are just the values from which we want to start to test the parameters.
- In the next slide, we can see the output: also proceeding in this way the best choice seems to be ARIMA$(1, 1, 1)$.

Example: Solar PV module prices. Results of AIC test

```
Performing stepwise search to minimize aic
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=73.366, Time=0.13 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=97.404, Time=0.08 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=76.351, Time=0.05 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=76.278, Time=0.04 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=106.069, Time=0.02 sec
 ARIMA(2,1,1)(0,0,0)[0] intercept   : AIC=75.314, Time=0.07 sec
 ARIMA(1,1,2)(0,0,0)[0] intercept   : AIC=75.294, Time=0.08 sec
 ARIMA(0,1,2)(0,0,0)[0] intercept   : AIC=74.531, Time=0.05 sec
 ARIMA(2,1,0)(0,0,0)[0] intercept   : AIC=74.668, Time=0.04 sec
 ARIMA(2,1,2)(0,0,0)[0] intercept   : AIC=77.268, Time=0.15 sec
 ARIMA(1,1,1)(0,0,0)[0]             : AIC=74.534, Time=0.04 sec

Best model:  ARIMA(1,1,1)(0,0,0)[0] intercept
Total fit time: 0.766 seconds
```

Example: Solar PV module prices. Getting the parameters of our ARIMA$(1, 1, 1)$ model

- Once we have chosen d, p and q, we have to estimate the parameters φ_1 and ψ_1 such that we can express our series as

$$y'_t = \varphi_1 y'_{t-1} + \psi_1 \epsilon_{t-1} + \epsilon_t,$$

  with $y'_t := y_t - y_{t-1}$.
- In Python, we can write

```
from statsmodels.tsa.arima.model import ARIMA
```

  and then

```
model = ARIMA(prices, order=(1,1,1))
model_fit = model.fit()
print(model_fit.summary())
```

- In the next slide, we can see what we get: `ar.L1` and `ma.L1` are the AR term φ_1 and the MA term ψ_1, respectively, whereas `sigma2` is the variance of the error term.

Example: Solar PV module prices. Getting the parameters of our ARIMA$(1, 1, 1)$ model

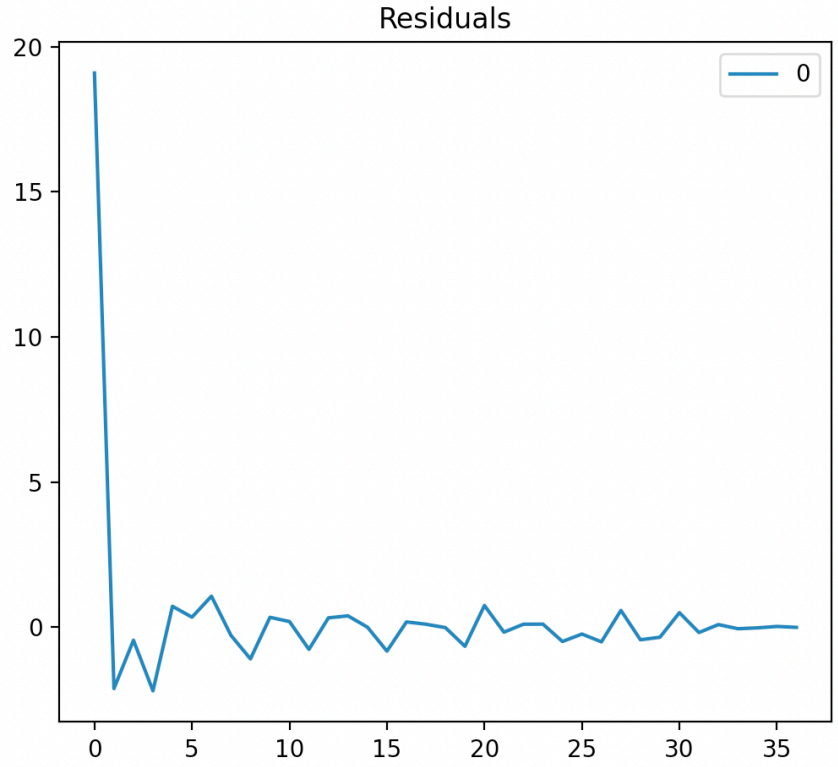
SARIMAX Results

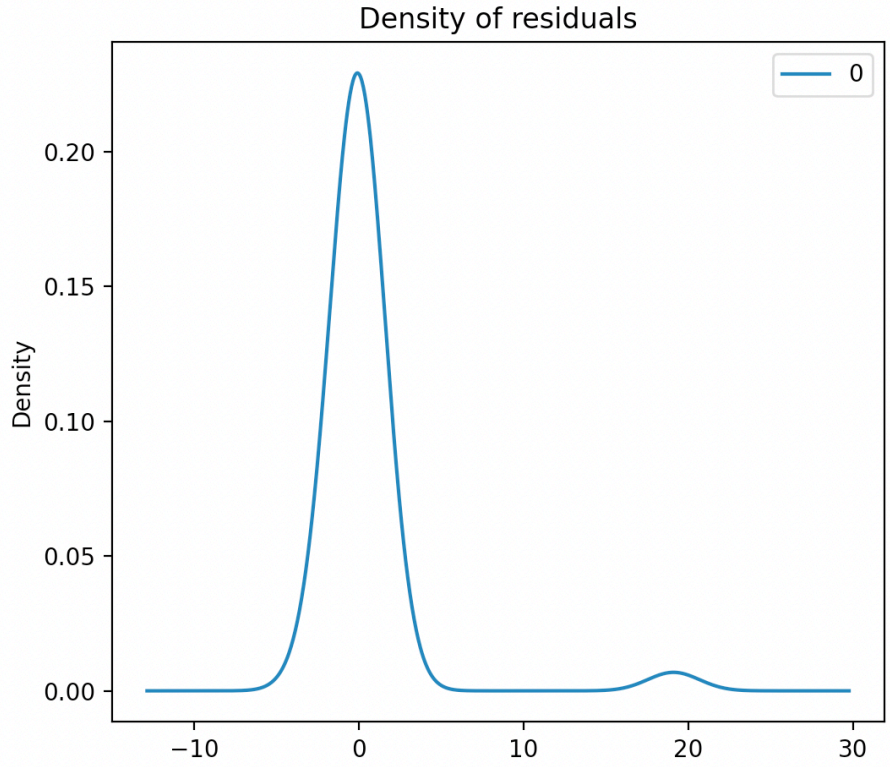
Dep. Variable:	y	No. Observations:	37
Model:	ARIMA(1, 1, 1)	Log Likelihood	-34.267
Date:	Tue, 07 Jun 2022	AIC	74.534
Time:	08:16:05	BIC	79.285
Sample:	0	HQIC	76.192
	- 37		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.6210	0.156	3.979	0.000	0.315	0.927
ma.L1	0.4995	0.228	2.195	0.028	0.054	0.945
sigma2	0.3788	0.087	4.366	0.000	0.209	0.549

Example: Solar PV module prices. Testing the model looking at the residuals

In order to test our ARIMA$(1, 1, 1)$ model, we can look at the residuals and their density





Example: Solar PV module prices. Testing the model looking at the residuals. Code

```
residuals = pd.DataFrame(model_fit.resid)
fig, ax = plt.subplots(1,2)
residuals.plot(title="Residuals", ax=ax[0])
residuals.plot(kind='kde', title='Density of residuals',
ax=ax[1])
plt.show()
```

Main reference

Rob J Hyndman and George Athanasopoulos, *Forecasting: Principles and Practice (2nd ed)*. OTexts, 2018.

Thank you for your attention!

For any question write to

mazzon@math.lmu.de

Greening Energy Market and Finance

Project website: http://grenfin.eu

With the support of the Erasmus+ Programme of the European Union