# Time Series Introduction

# Ordering

In general machine learning:

- no specific ordering of the data points
- all training points are potentially relevant for the prediction of a new point

In **time series**:

- chronological ordering of the points
- recent points are potentially more relevant than older points for the prediction of a new point

CAMBRIDGE SPARK

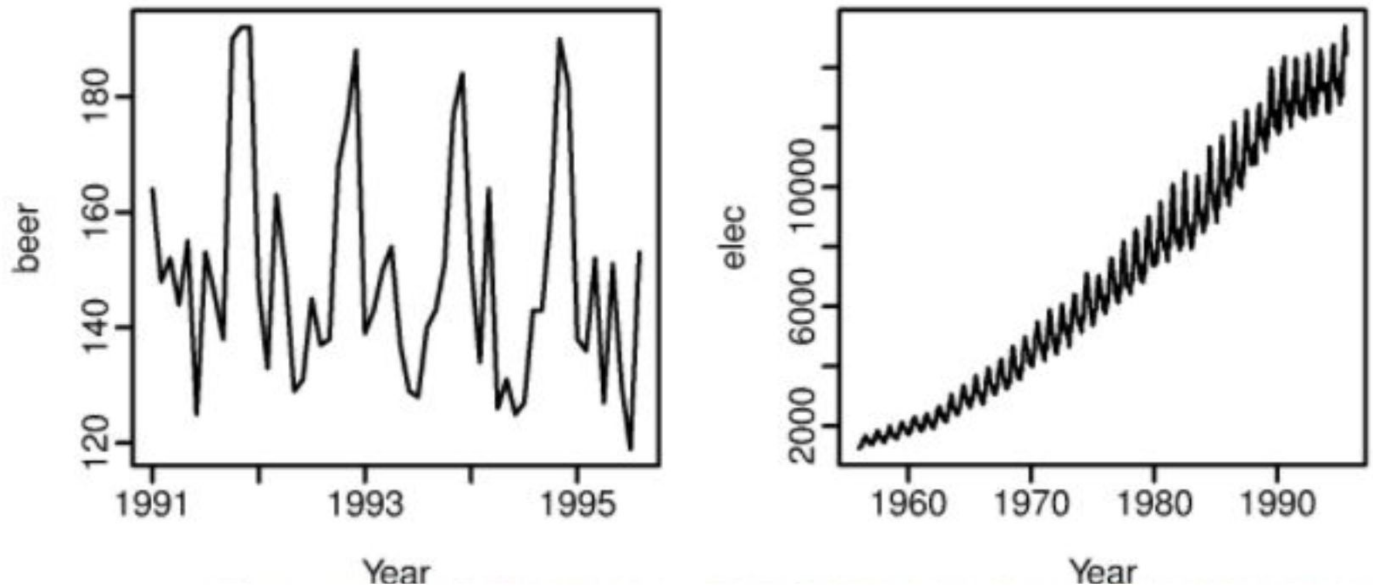
# Patterns in time series

Generally, assume the time series can be decomposed into several elements, each of which we model separately:

- **Trend**: a long term increase or decrease in the data
- **Seasonality**: a cyclic pattern in the data (e.g.: days of the week, quarters of the year)
- **Noise**: a non deterministic element in the data

CAMBRIDGE SPARK

# Seasonality vs. Trend




CAMBRIDGE SPARK

Hands-on session

# time_series_intro-skeleton.ipynb

## >> Importing and visualising Time Series data

CAMBRIDGE SPARK

# Resampling

Resampling involves changing the frequency of your time series observations.

**Upsampling**: increase the frequency of the samples, (e.g. from days to hours)

**Downsampling**: decrease the frequency of the samples (e.g. from days to weeks)

`pandas.DataFrame.resample`

CAMBRIDGE SPARK

# Resampling

We use resampling because we have observations at the wrong *frequency*:

- They may be too granular or not granular enough!

**Upsampling**: this typically requires more care, as we are essentially interpolating between observations to *guess* what the measurements would have been in between.

For example: if we observe data hourly but need measurements every 30 minutes, then one approach could be to simply average the measurements before and after (linear interpolation).

`pandas.DataFrame.resample`

CAMBRIDGE SPARK

# Resampling

We use resampling because we have observations at the wrong *frequency*:

- They may be too granular or not granular enough!

**Downsampling**: this typically *safer* than upsampling, because are aggregating data and reducing the granularity.

For example: we may observe hourly data but only be interested in daily measurements. Then one approach would be to compute the mean over all hourly measurements.

`pandas.DataFrame.resample`

CAMBRIDGE SPARK

Hands-on session

# time_series_intro-skeleton.ipynb

>> Resampling

>> Parsing custom date formats

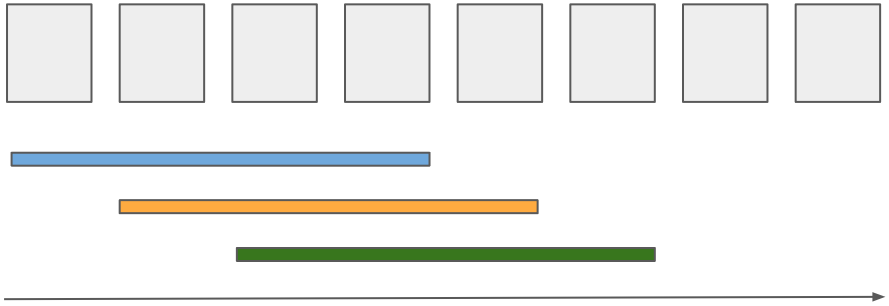
CAMBRIDGE SPARK

# Moving Windows

Big difference in time series: **not all data are equal!**

Moving windows involves applying a function on repeated fixed-width “slices” (window) of the data sliding along in the direction of the data

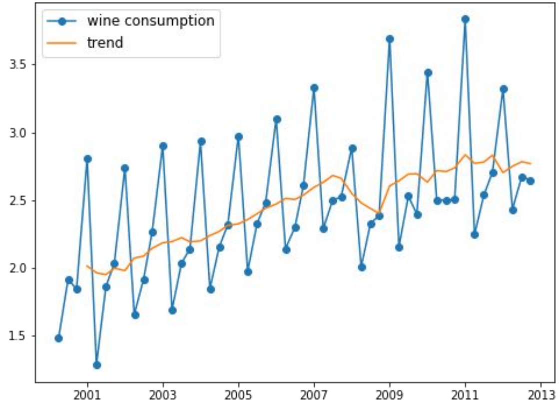
`pandas.DataFrame.rolling`



Window of width 4

# Moving Average

- Better expose the trend of the data
- Creates a new series by calculating the average of fixed-width windows
- The window slides along the time series
- The same principle can be used for moving standard deviation, moving median etc




CAMBRIDGE SPARK

Hands-on session

# time_series_intro-skeleton.ipynb

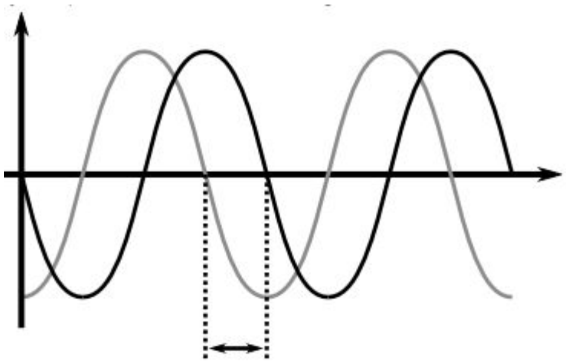
## >> Moving windows

# Shifting Time Series Data

Sometimes we want to shift time series data en masse.

- To remove a known consistent latency
- To help identify causal relationships more easily
- To create lag features

| Index | Value(t) | Value(t-1) |
|---|---|---|
| t | 50 | 40 |



`pandas.DataFrame.shift`

CAMBRIDGE SPARK

Hands-on session

# time_series_intro-skeleton.ipynb

## >> Shifting Time Series Data

CAMBRIDGE SPARK

# Differencing

Differencing is a method of transforming a time series dataset.

Differencing is performed by subtracting a previous observation from the current observation. Here with a lag of 1:

```
difference(t) = observation(t) - observation(t-1)
```

Using differencing can help remove the trend (lag of 1) and seasonality (lag of m) and expose the noise in the time series.

```
pandas.DataFrame.diff
```

CAMBRIDGE SPARK

Hands-on session

# time_series_intro-skeleton.ipynb

## >> Differencing

CAMBRIDGE SPARK

# Autocorrelation

Autocorrelation is the correlation (similarity) of a time series with a lagged version of itself.

Ex: take values `[1:10]` then values `[5:15]` how similar are these two sequences of values?

It helps expose the **seasonality** structure of the data.

`pandas.DataFrame.autocorr`

CAMBRIDGE SPARK

Hands-on session

# time_series_intro-skeleton.ipynb

## >> Autocorrelation

CAMBRIDGE SPARK

# Takeaways

- We often model time series data as consisting of a **trend** and **seasonal** component, together with some observational noise
- Often it may be helpful to resample the data in order to align it with our objectives. Upsampling requires more care than downsampling
- **Not all data are created equal!** We may discard past observations using methods such as sliding windows
- **Differencing** is helpful in removing some of the trend and seasonality in the data