



Good morning 😊  
Notebooks and slides are in:

**[https://github.com/AlineQuadros/teaching\\_data\\_science](https://github.com/AlineQuadros/teaching_data_science)**

**Packages we'll need:**

scipy  
statsmodels  
prophet  
pmdarima  
Plotly

Installation instructions for Prophet (please check):  
**<https://facebook.github.io/prophet/docs/installation.html#python>**





# **TIME SERIES FOR DATA SCIENCE**

DR. ALINE QUADROS  
DATA SCIENCE RETREAT



# Contents

- What are time series?
- What kind of time series properties are there? (*buzz words*)
  - Trend, seasonality
  - Stationarity
  - Autocorrelation
  - Baselines
    - average, naïve, smoothing moving average
- ‘Traditional’ time series analysis:
  - Exponential smoothing (a.k.a ETS models)
  - Autoregressive models (a.k.a. AR models): ARIMA family (ARMA, ARIMA, SARIMA, SARIMAX)
- ML-based time series:
  - Random forests and boosting trees
  - Prophet





# Resources

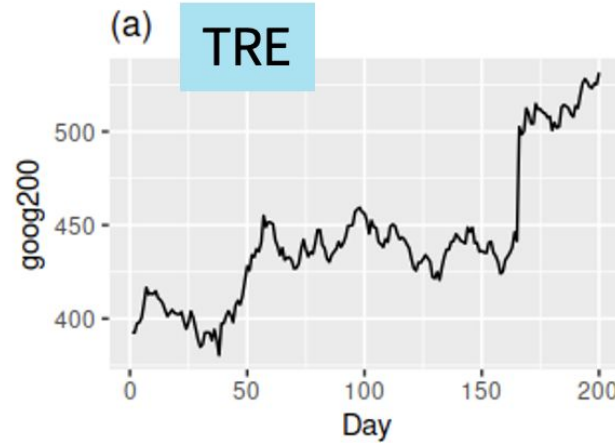
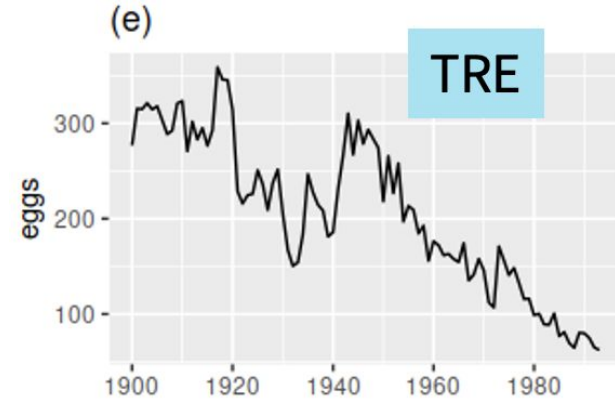
- Free excellent book
  - Hyndman, R.J., & Athanasopoulos, G. (2018) *Forecasting: principles and practice*, 2nd edition, OTexts: Melbourne, Australia. [www.OTexts.com/fpp2](http://www.OTexts.com/fpp2)
  - *Most of the plots used in this presentation come from this book*
- Bastian Kubsch's repository:  
[https://github.com/bkubsch/time\\_series](https://github.com/bkubsch/time_series)
- Online class = <https://online.stat.psu.edu/stat510/lesson/1>
- Exercises:  
[https://github.com/AlineQuadros/Timeseries\\_predictive\\_modelling](https://github.com/AlineQuadros/Timeseries_predictive_modelling)



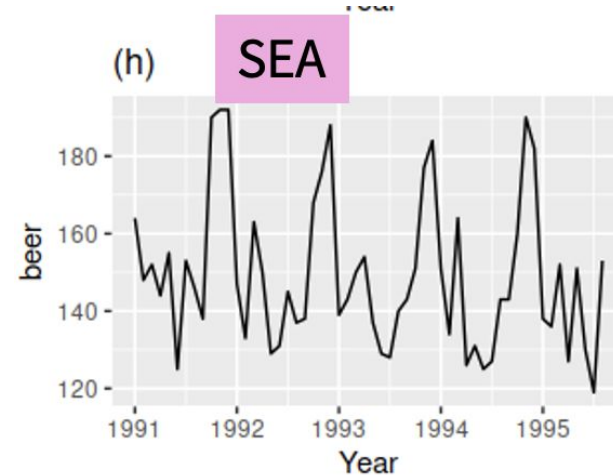
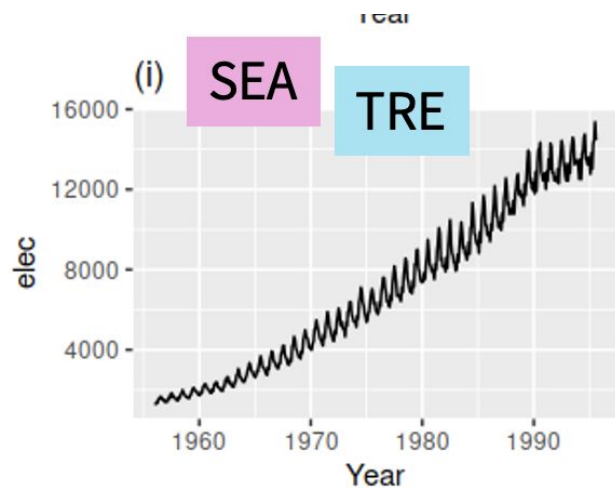


# **TIME SERIES PROPERTIES**

# ○ Trend and seasonality



**Trend:** average changes over time

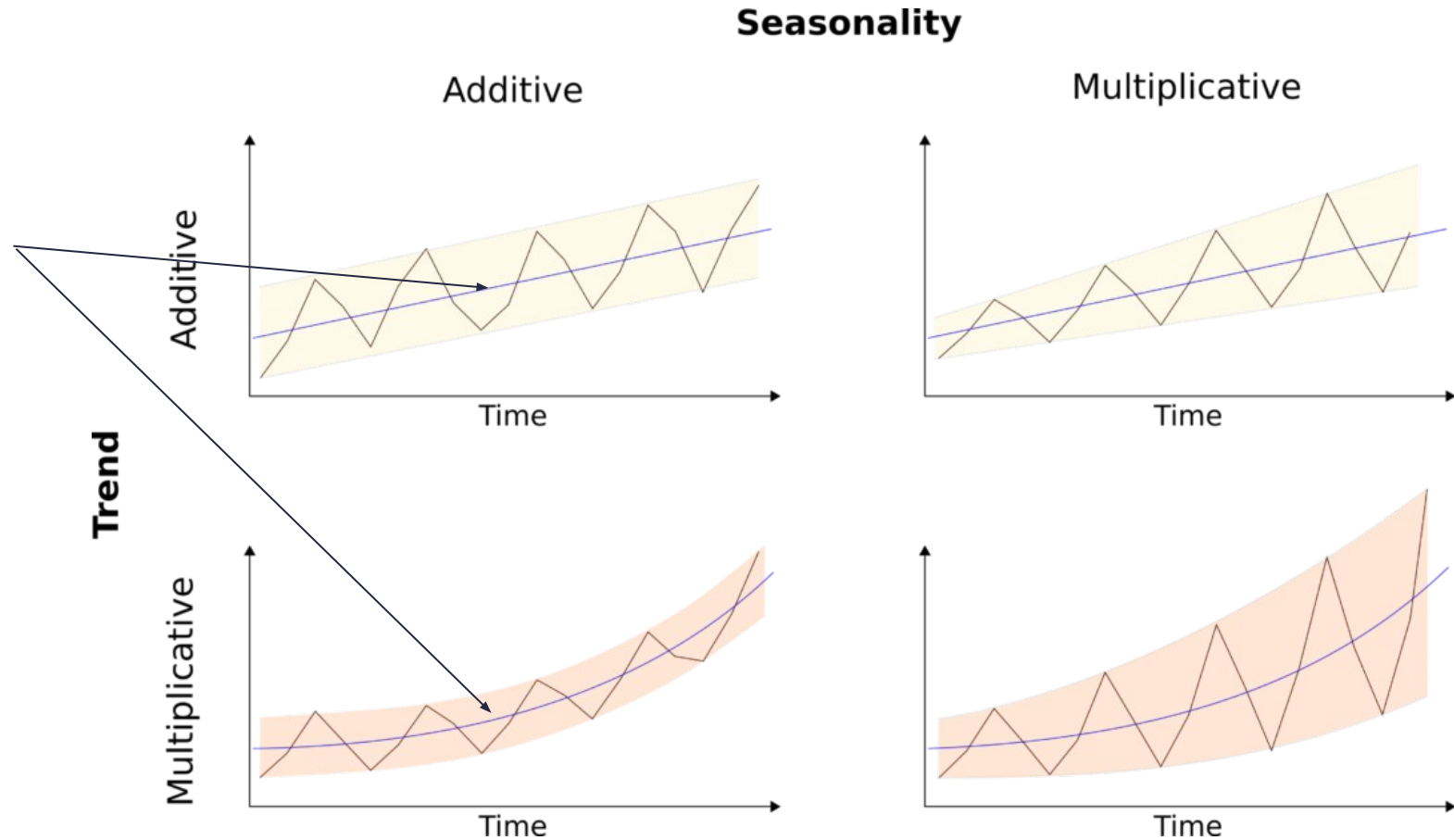


**Seasonality:** variance changes over time



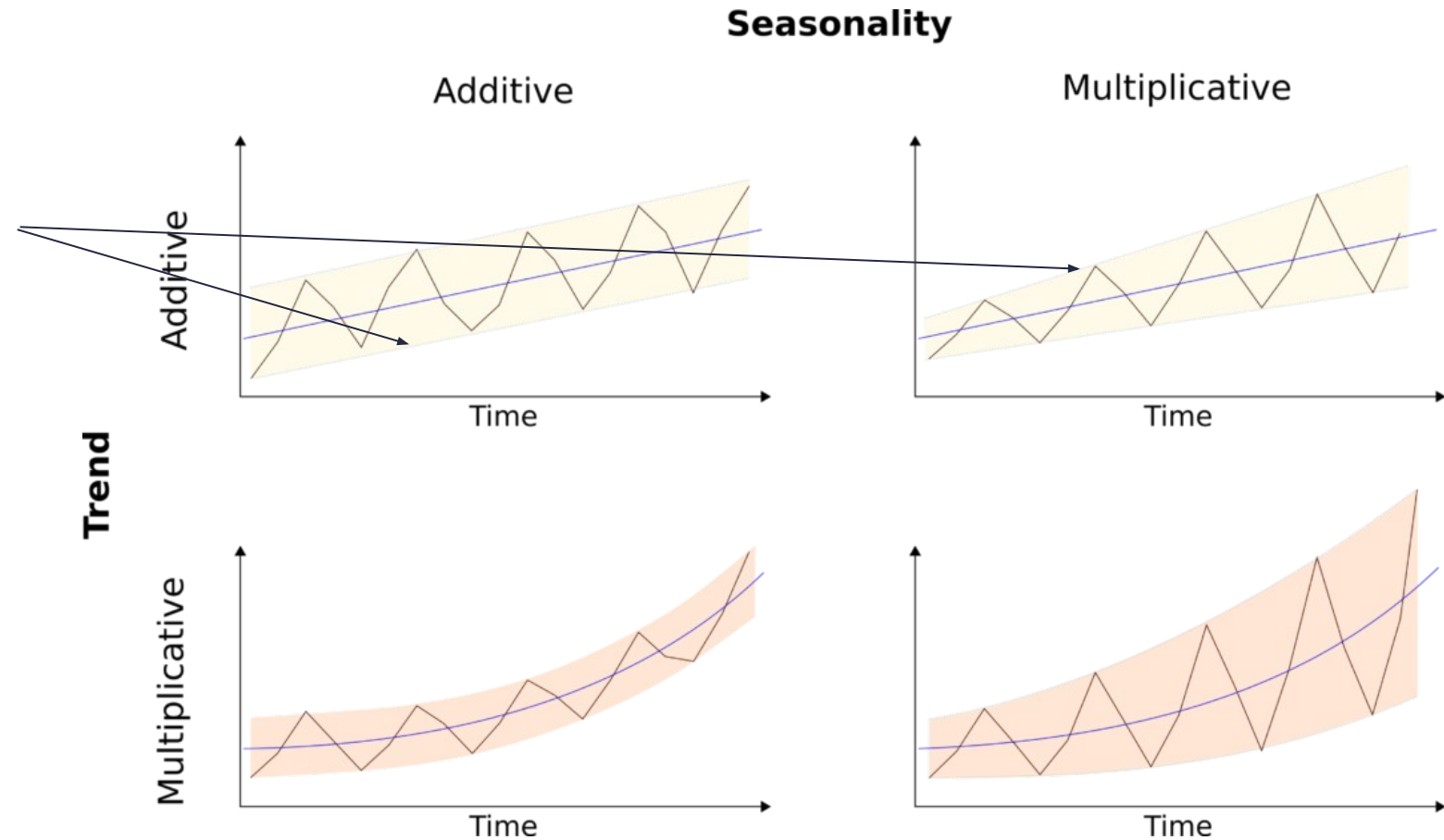
# ○ Trend and seasonality can be additive or multiplicative

**Trend:** Look at the **blue lines** and how it changes over time



# ○ Trend and seasonality can be additive or multiplicative

**Seasonality:** Look at the filled areas and how their shape changes over time

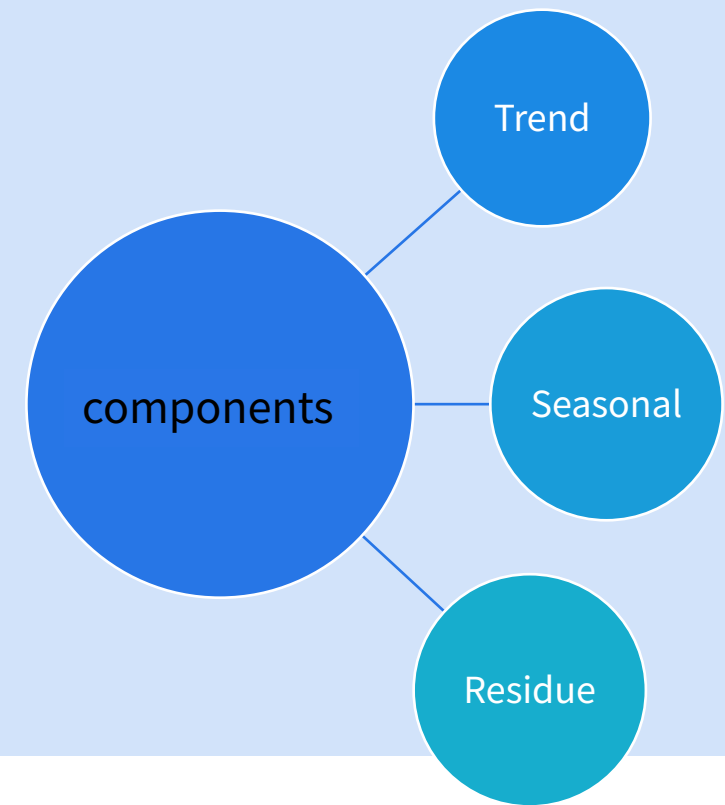




# Time series decomposition

**Additive** = the components add to each other

**Multiplicative** = the effects are multiplied



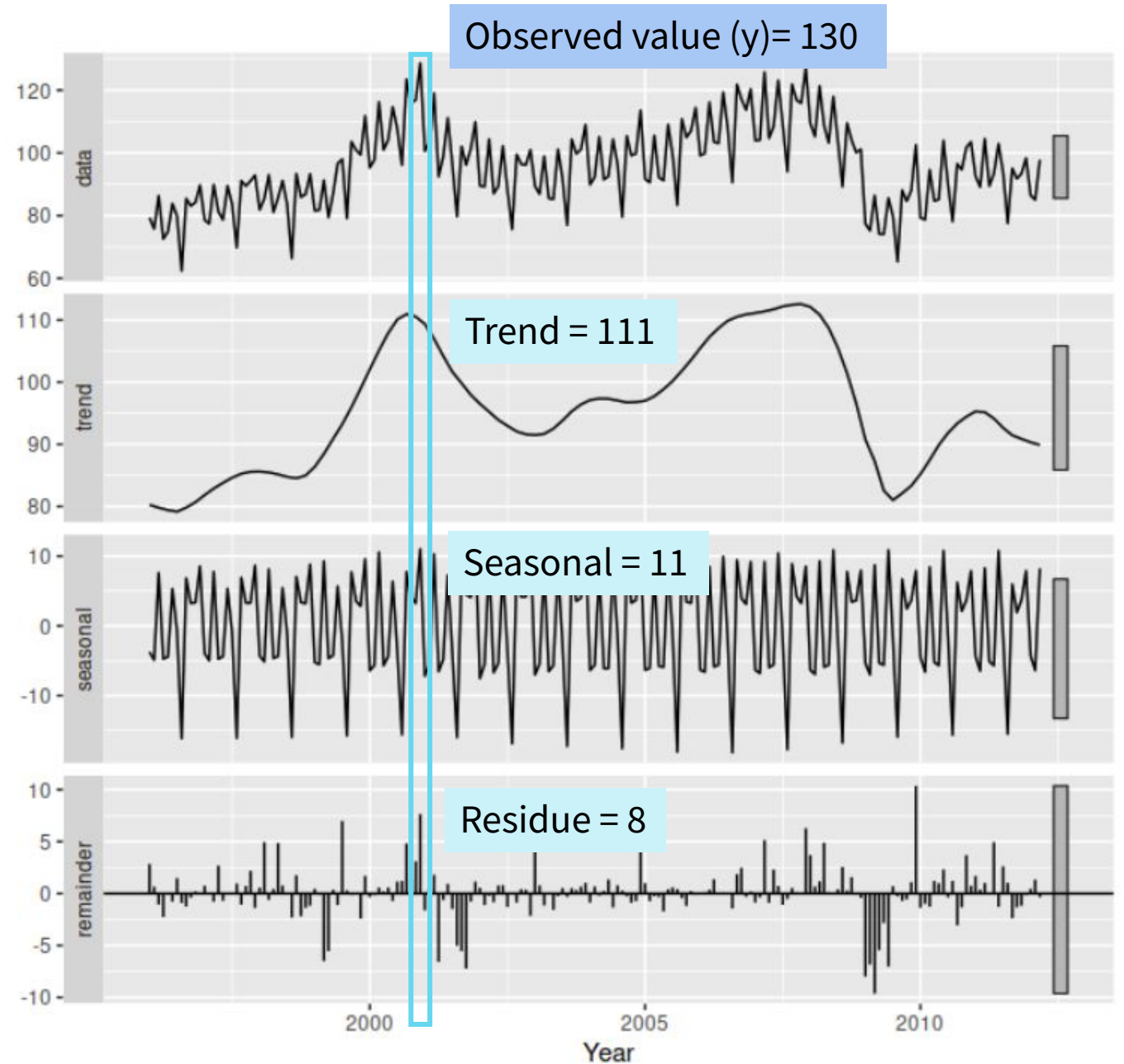
# Time series decomposition

## Example of additive decomposition:

Look at the value at time 2001:

The observed value (130) is decomposed as:

$$130 = 111 \text{ (trend)} + 11 \text{ (seasonal)} + 8 \text{ (residue)}$$



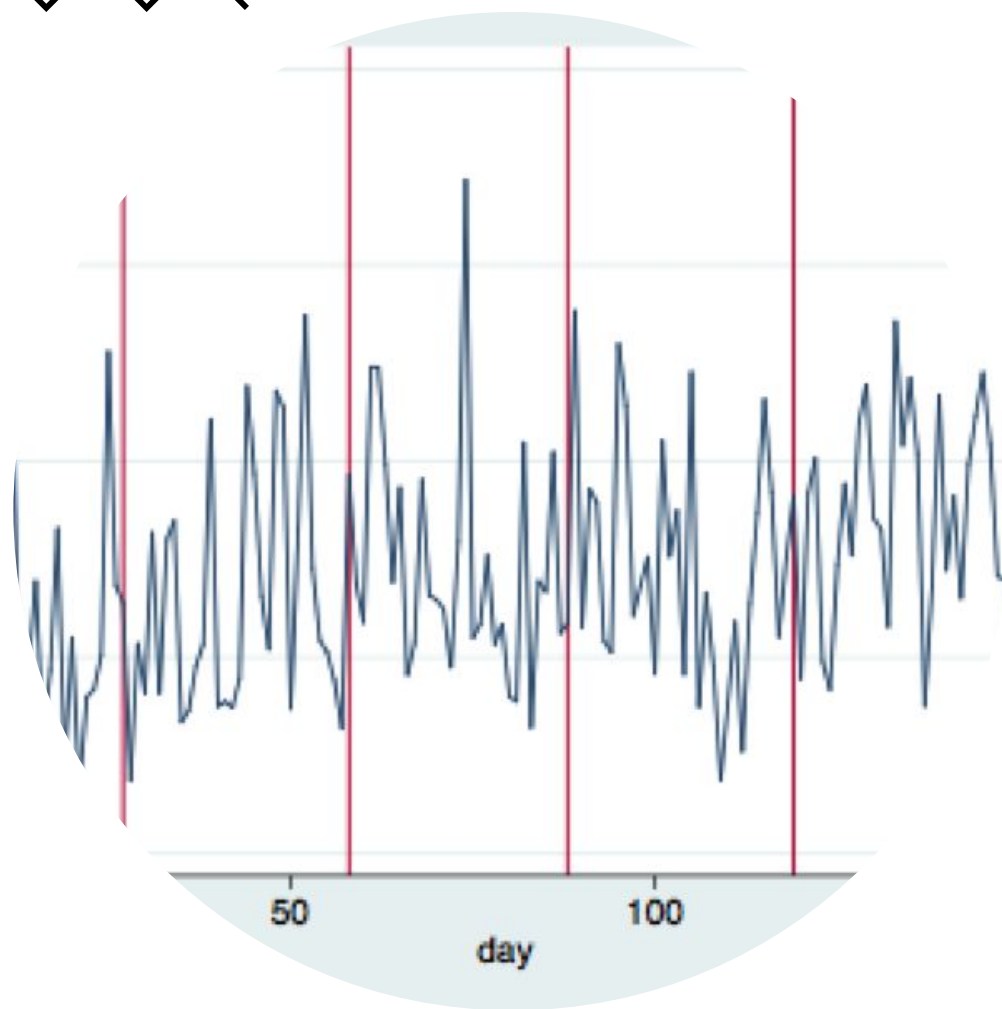


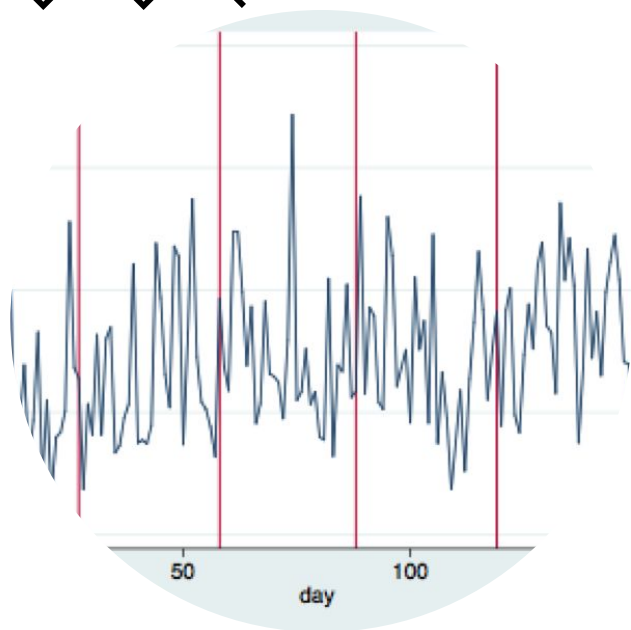
# Stationarity

- Stationary data is data without trends or seasonality
- In stationary data, the mean and variance of the time series don't change over time
- **Meaning, even though the data has a time axis, its values do not depend on the time component itself**

- **How to detect non-stationarity?**

- Plots, ACF plots
- Hypothesis tests (AD Fuller test)





# Differencing

- How to deal with non-stationary data?  
Use the differences between observations

Stationary vs Non-Stationary Data - Google Stocks





# Autocorrelation

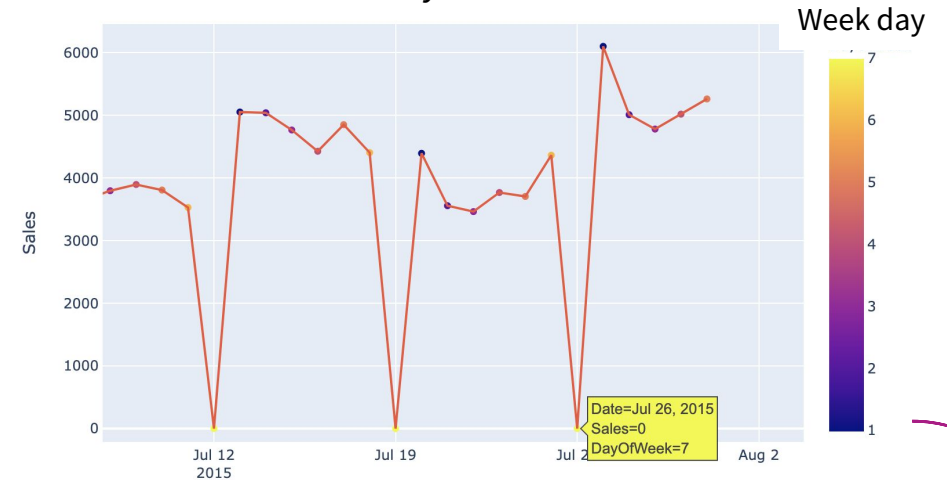
Autocorrelation = correlation with **itself** in the past

The presence of a strong, significant autocorrelation with a fixed interval is a very strong indicator of seasonality

Example:

*Rossman's stores are closed on sundays, and sales pick up on Mondays*

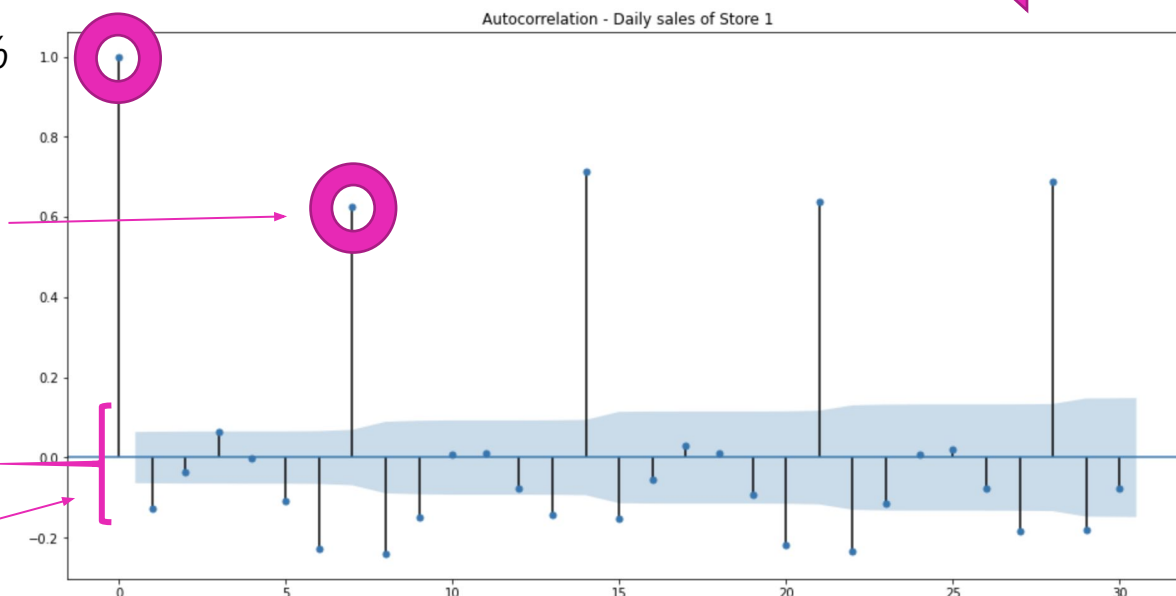
Rossman Dataset – Daily sales of store 1



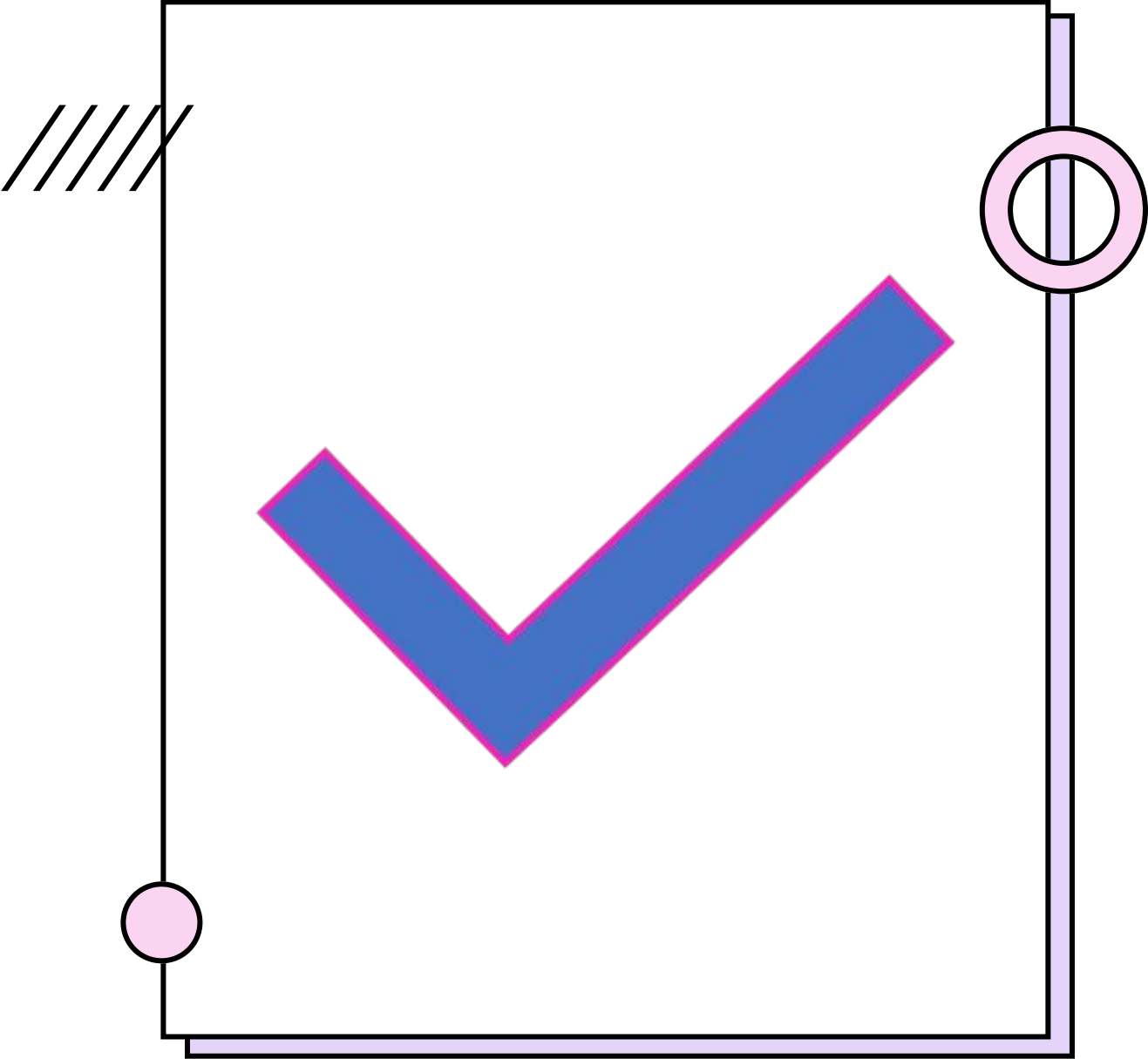
*Correlation with itself is always 1, or 100%*

*Recurrent very strong correlation (65%) at 7 days intervals*

*The shaded area indicates the 95% confidence intervals.  
The null hypothesis is that autocorrelation=0*



# BASELINES



# ○ Simplest possible forecast (baselines)

Simplest representation of trend, seasonality and autocorrelation

In practice, they will be used as a reference to compare our candidate models

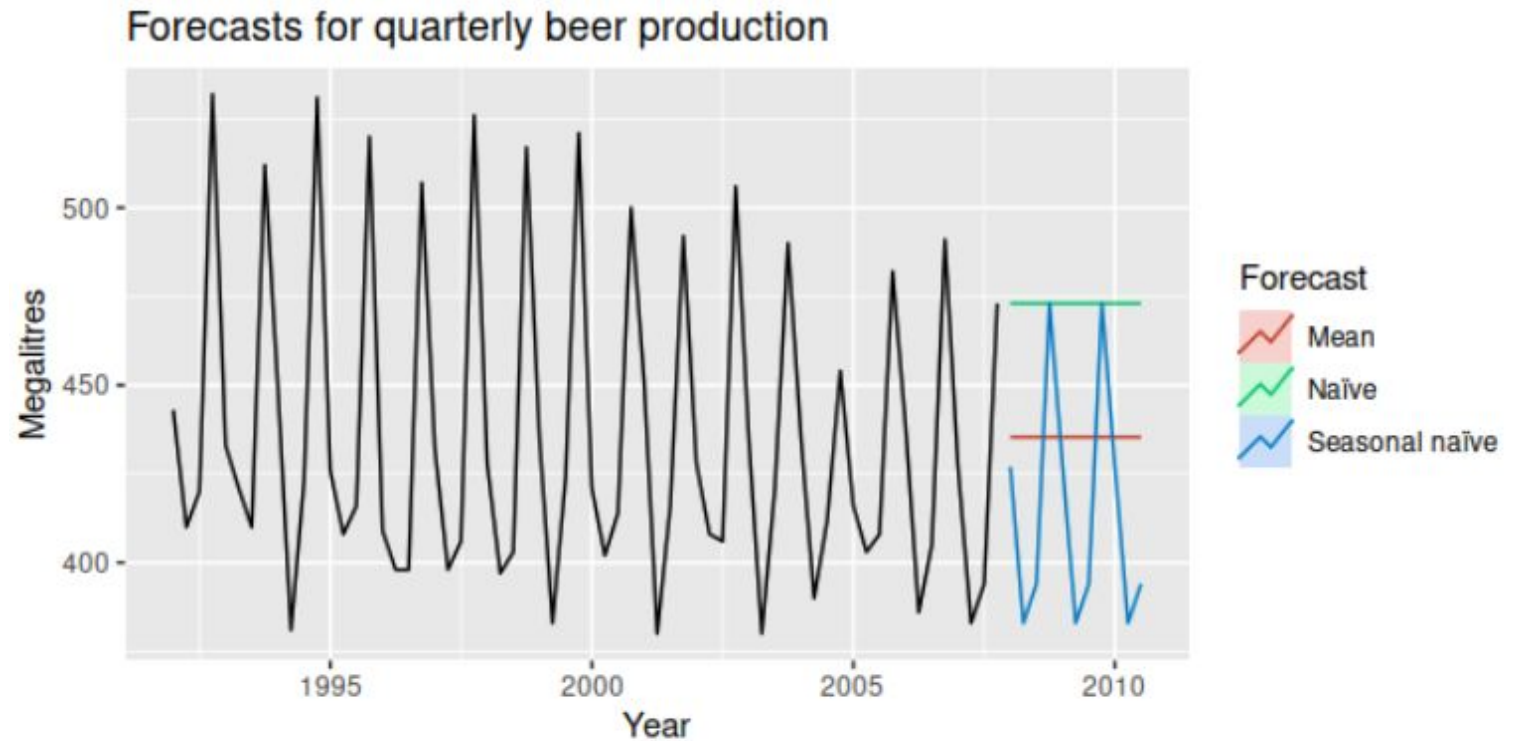



Figure 3.1: Forecasts of Australian quarterly beer production.





# **EXPONENTIAL SMOOTHING**

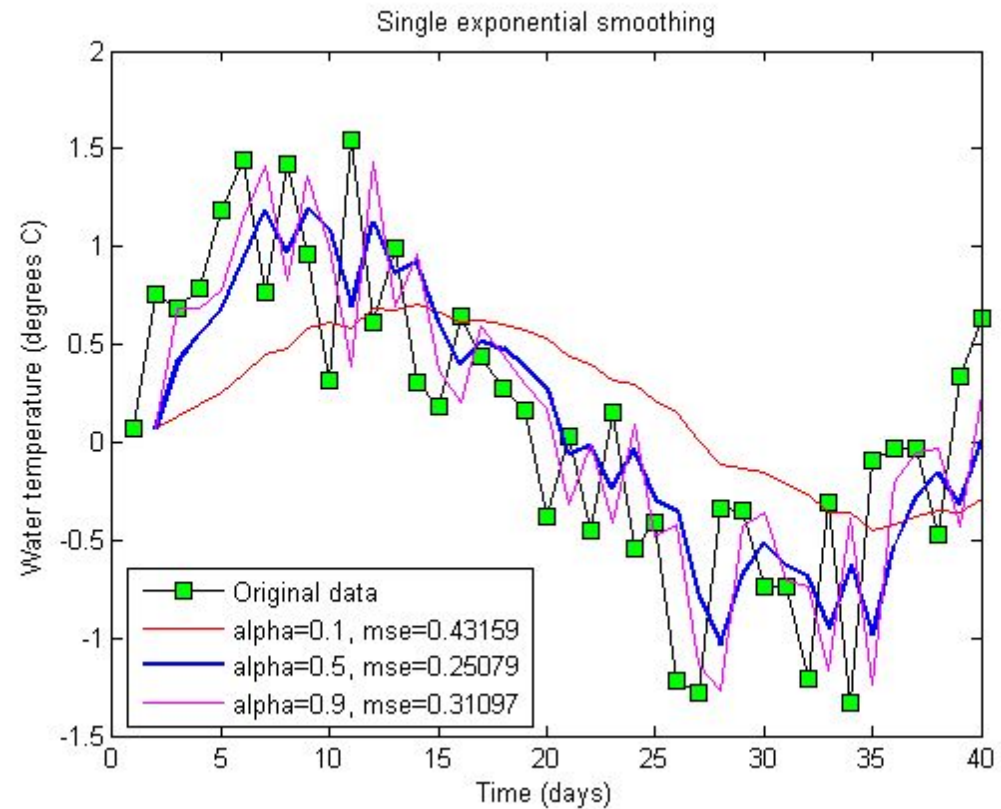


# Exponential Smoothing

Exponential smoothing methods are **weighted averages of past observations**

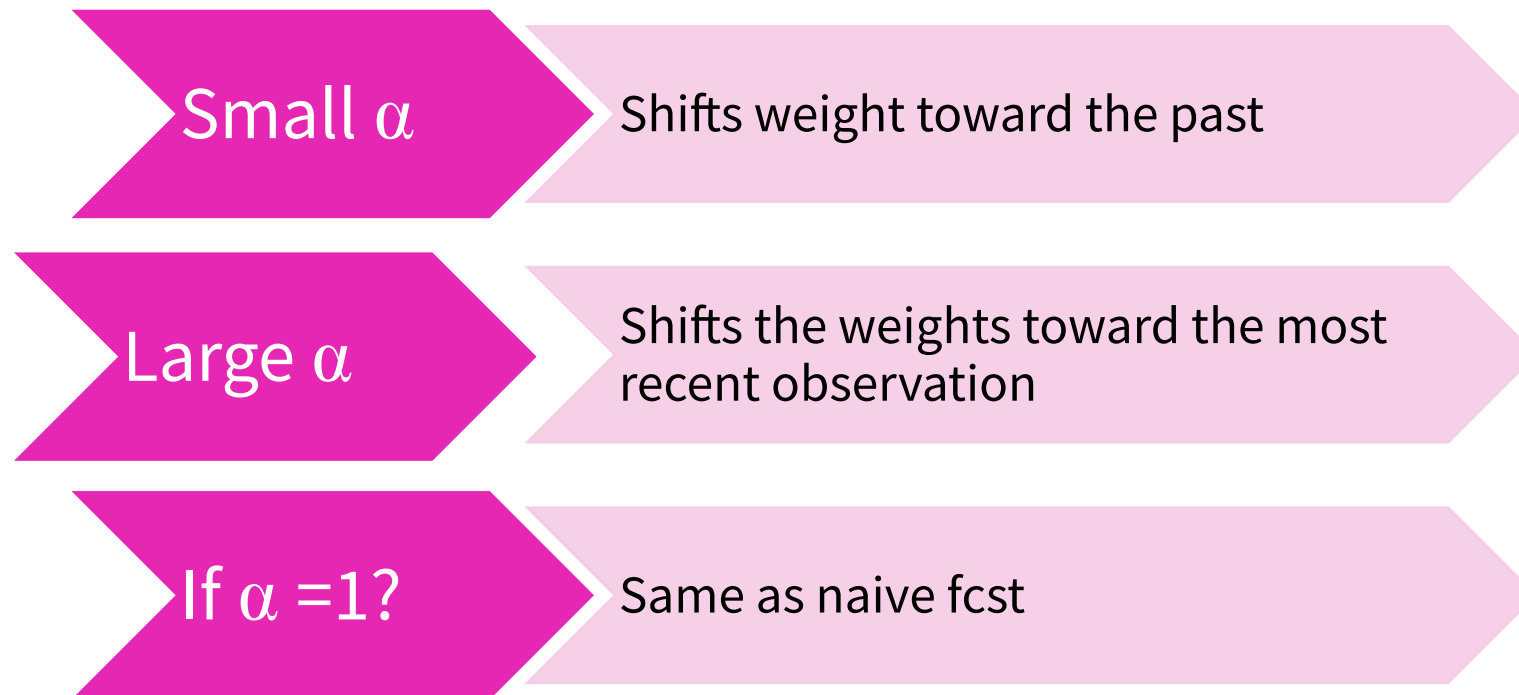
The weights decrease exponentially as the observations get older

The degree of decrease is given by parameter **alpha**.



# ○ Exponential Smoothing

The way the weights are distributed along time is determined by the smoothing parameter alpha ( $\alpha$ )



# Alpha

How the weights are distributed along time is determined by the smoothing parameter **alpha** ( $\alpha$ )

	$\alpha = 0.2$	$\alpha = 0.4$	$\alpha = 0.6$	$\alpha = 0.8$
$y_T$	0.2000	0.4000	0.6000	0.8000
$y_{T-1}$	0.1600	0.2400	0.2400	0.1600
$y_{T-2}$	0.1280	0.1440	0.0960	0.0320
$y_{T-3}$	0.1024	0.0864	0.0384	0.0064
$y_{T-4}$	0.0819	0.0518	0.0154	0.0013
$y_{T-5}$	0.0655	0.0311	0.0061	0.0003





# Methods

- Simple Exponential Smoothing
  - For time series without a clear trend or seasonality, i. e. for **stationary data**
- Holt-Winter's family of methods
  - Can be applied in time series with trend and/or seasonality
  - Trend and seasonality can be **additive** or **multiplicative**:
    - Additive: magnitude does not change over time
    - Multiplicative: magnitude of trend or length of cycle (seasonality) changes over time
    - How to check? Plotting the data



# ○ Time series scenarios (and appropriate models)

		SEASONALITY		
		<u>None</u>	<u>Constant</u> (Additive)	<u>Increasing</u> (Multiplicative)
TREND	<u>None</u>	(N, N) <b>Simple exponential smoothing</b> <i>(uses just alpha)</i>	(N, A)	(N, M)
	<u>Linear</u> (Additive)	(A, N) <b>Holt's linear method</b>	(A, A) <b>Additive Holt-Winter's</b>	(A, M) <b>Multiplicative Holt-Winter's</b>
	<u>Exponential</u> (Additive damped)	(Ad, N) <b>Additive damped trend method</b>	(Ad, A)	(Ad, M) <b>Holt-Winter's damped</b>





# **AUTO REGRESSIVE MODELS**

# ○ Autoregressive models

- Classic and robust methods. Can be applied in many different situations
- Univariate
- ARIMA = Autoregressive Integrated Moving Average
- ARIMA models describe the autocorrelation in the data
- Can deal with non-stationarity: trend and seasonality (SARIMA)



The notation  $(p, q, d)$

# ARIMA

AUTOREGRESSIVE component  
a.k.a.  $p$

**Linear combination of past values;  
Regression against itself**

DIFFERENCING component  
a.k.a.  $d$

**Helps to stabilise the mean,  
reducing trend and seasonality**

MOVING AVERAGE component  
a.k.a.  $q$

**Weighted moving average of  
past errors**

## VARIATIONS:

NON-SEASONAL ARIMA  
 $ARIMA(p, d, q)$

SEASONAL ARIMA  
 $ARIMA(p, d, q) (P, D, Q)_m$

SARIMAX  
+ features

$m$  = the width of the seasonality,





# ○ Understanding (p, d, q) and m

- **p** = is the AR term. Is the number of lags of Y to use as predictors
- **q** = is the MA term. Is the number of lags to use to get the forecast errors
- **d** determines how many periods to lag before calculating the differences
  - For an Array[10, 4, 2, 9, 34] => d=1 results in [-6, -2, 7, 25]
  - If the data is already stationary, d = 0
- The **m** indicates the number of observations per seasonal cycle, will be used in SARIMA and SARIMAX.





# ARMA

Family of models

**ARMA**

Only handles stationary data

**ARIMA**

Can handle trends by applying differentiation

**SARIMA**

Can handle trends and seasonality

**SARIMAx**

Can handle trends + seasonality + regressors





# **WHAT'S NEXT?**

**ML MODELS/Deep learning for  
time series**

# OVERVIEW OF TIMESERIES METHODS

Components	Only univariate (i. e., one time series at time)	Univariate + regressors (other related timeseries)	Multivariate (i. e., multiple time series for multiple features)
Stationary	ARMA Exponential smoothing (ES) (classical)	SARIMAX Prophet LSTM	Tree algorithms RNN
Trend	ARIMA Exp. Smoothing Holz-winter	SARIMAX Prophet LSTM	RNN
Seasonality	SARIMA Exp. Smoothing Holz-winter	SARIMAX Prophet LSTM	Tree algorithms RNN
Trend+seasonality	SARIMA Exp. Smoothing Holz-winter	SARIMAX Prophet LSTM	RNN
Complex seasonality	TBATS	LSTM Prophet	Tree algorithms RNN



# ○ ML-based time series forecasting

- Tree models (Random forests, boosted trees)
- Prophet (can handle combinations of seasonality (e.g. weekday + yearly))

## Deep learning:

- LSTMs and RNN (next classes)
- AWS SAGEMAKER DEEPAR
  - Based on RNN
  - <https://docs.aws.amazon.com/sagemaker/latest/dg/deepar.html>

Temporal fusion Transformers (you can apply almost all you'll learn about language models to time series forecasting)



# ○ ML and TIME SERIES basics

- Always create a baseline model (naïve, average, moving average) to have a clearer understanding of how well your candidate models are performing
  - They should at least be better than the average
- Use as validation and test sets a period as long as the one you are being asked to predict:
  - Example: If you need to predict sales of next 3 months, use the last 3 months of data as test set, and the 3 months before that for cross-validation (for gridsearch, for example)



# ○ Forecast with Prophet



*Prophet is a procedure for forecasting time series data based on an **additive model** where **non-linear trends** are fit with yearly, weekly, and daily seasonality, plus holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data. Prophet is robust to missing data and shifts in the trend, and typically handles outliers well.*

**Note:** Prophet is optimized for business data, i. e., timeseries on calendar dates. Can account for holidays *out-of-the-box*

<https://facebook.github.io/prophet/>

Based on this publication: <https://peerj.com/preprints/3190/> (super recommended reading)

