TIME SERIES

MARÍA HERNÁNDEZ RUBIO

TIME SERIES

AGENDA

- **What is a time series?**
- **TS decomposition**
  - `code`
- **Forecasting**
  - As a Machine Learning problem
  - With Exponential Smoothing
  - As a stochastic problem: AR, MA, ARMA, ARIMA, ARCH, GARCH
  - Selecting the model that best fits the data
- **How good is our model?**
  - `code`
- **Additional material**
  - Clustering
  - TS Forecasting with Deep Learning
  - Final remarks
  - TS in R and Python
  - Bibliography and resources

ABOUT ME…

María Hernández Rubio

Mathematics & Computer Science, UAM
MsC Computational Intelligence, UAM

Senior Data Scientist at BBVA Data & Analytics

Joined BBVA in 2011, Smart Cities, external consultant
Joined Beeva (now BBVA Next Technologies) in 2013
Joined BBVA D&A in 2014

- Urban Analysis, C360, RecSys
- (Non-) Customer Intelligence
- Smart Replies (NLP)

What is a time series?

Variables that depend on time

TIME SERIES

Time series: variable that depends on time.
Collection of data points collected at constant time intervals.

- It can be univariate or multivariate. We will focus on univariate TS.
- But what makes a TS different from, say a regular regression problem? There are 2 things:
  - It is **time dependent**. So the basic assumption of a linear regression model that the observations are independent doesn't hold in this case.
  - Along with an increasing or decreasing trend, most TS have some form of **seasonality trends**, i.e. variations specific to a particular time frame.
- Very central in Econometrics.
- Objectives:
  - **Understand** or model the behaviour of the economic magnitude
  - Predict or **forecast the future.**

TIME SERIES EXAMPLES

Economics:

- Share price measured each day / hour.
- Unemployment rate measured each month.
- IPC or Euríbor

Metheorology:

- Daily rainfall, temperature
- Hourly wind speed

Demography:

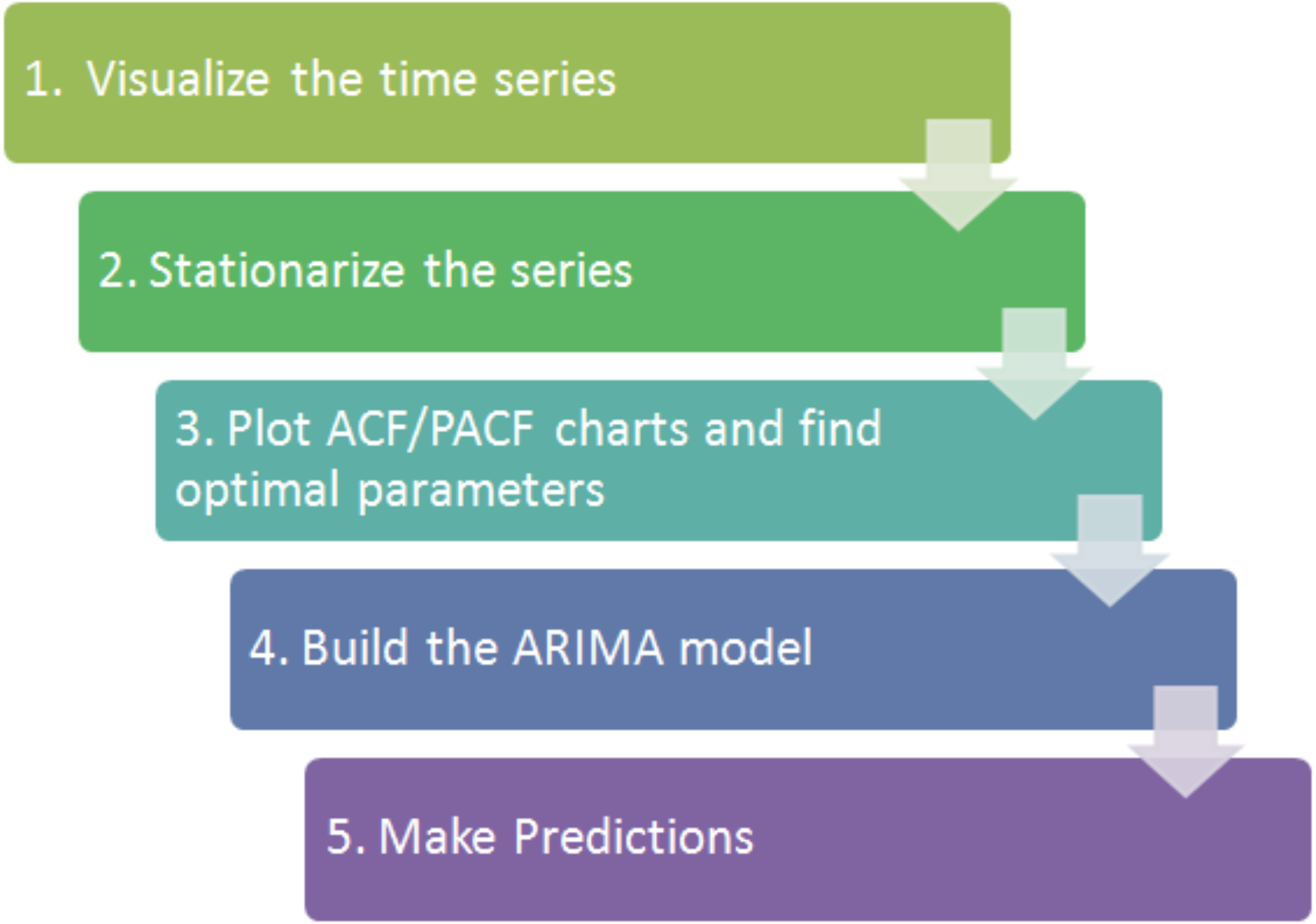
- Population of a city/province/country in successive years

Marketing:

- Sales in successive months

Exercise: name a few!

TIME SERIES ANALYSIS



Exploration

Preprocessing

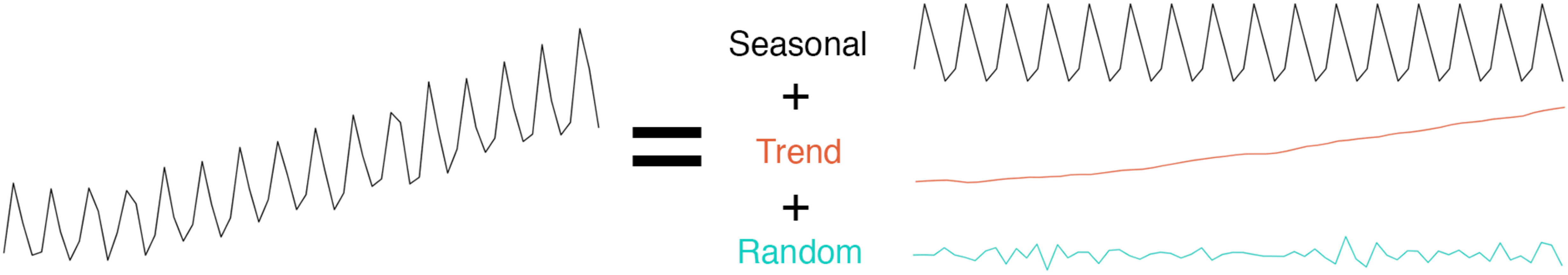
Modelling & hyperparameter tunning

Predict

Source: analyticsvidhya blog

Time Series Decomposition

Computing trend, seasonality and noise



Time Series Decomposition

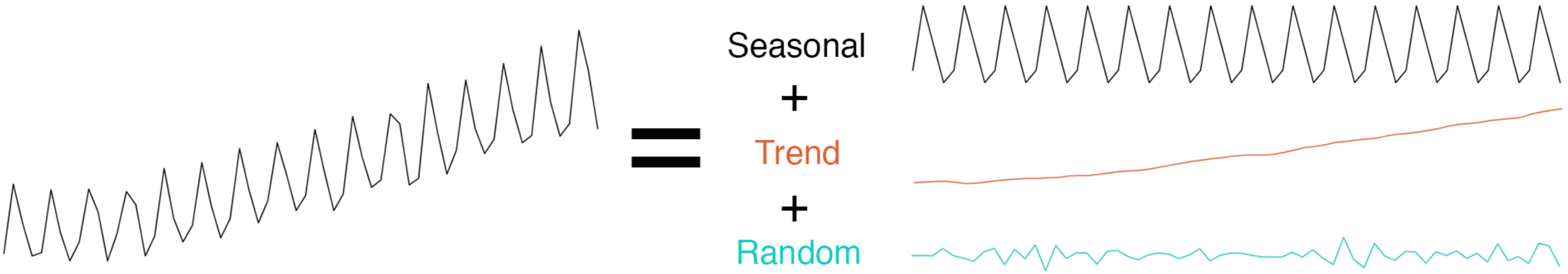
Computing trend, seasonality and noise

TS DECOMPOSITION

$$y_t = T_t + C_t + S_t + e_t$$

- T_t: Trend
- C_t: cyclical component
- S_t: seasonality
- e_t: error/random term

Note: the trend and cycle component are usually combined in the trend-cycle component (T_t)



TS DECOMPOSITION

Additive decomposition

$$y_t = T_t + C_t + S_t + e_t$$

Multiplicative decomposition

$$y_t = T_t \cdot C_t \cdot S_t \cdot e_t$$

T_t: Trend, C_t: cyclical component, S_t: seasonality, e_t: error/random term

Note: the trend and cycle component are usually combined in the trend-cycle component (T_t)

- In the **additive** model the seasonal component does look constant over time; whereas in the **multiplicative** model the seasonal effect increases as the time series increases in magnitude. Examples:
  - beer production (additive)
  - air passengers (multiplicative)
- *Remark*: taking the log over the multiplicative model we get an additive model:

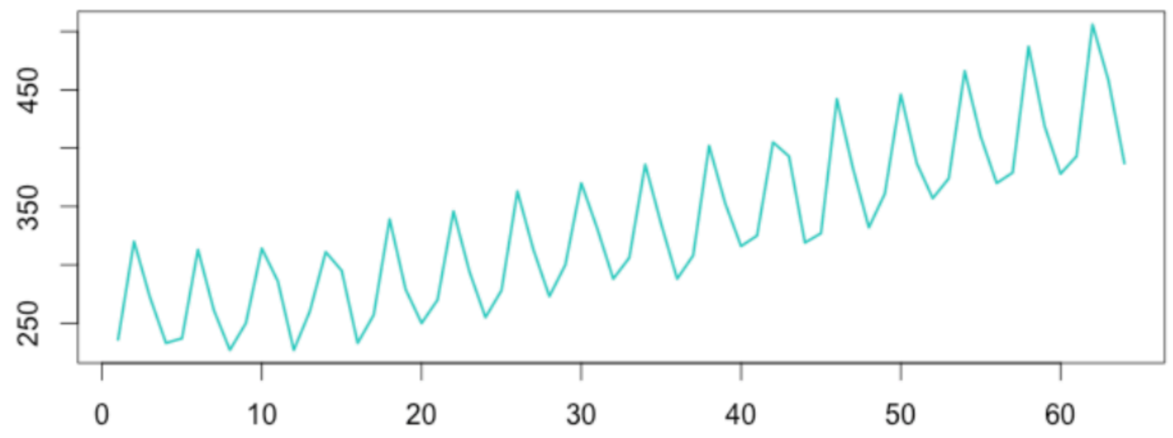
  $\log(y_t) = \log(T_t) + \log(S_t) + \log(e_t)$

Step-by-step example of additive vs multiplicative model

TS DECOMPOSITION

Additive decomposition

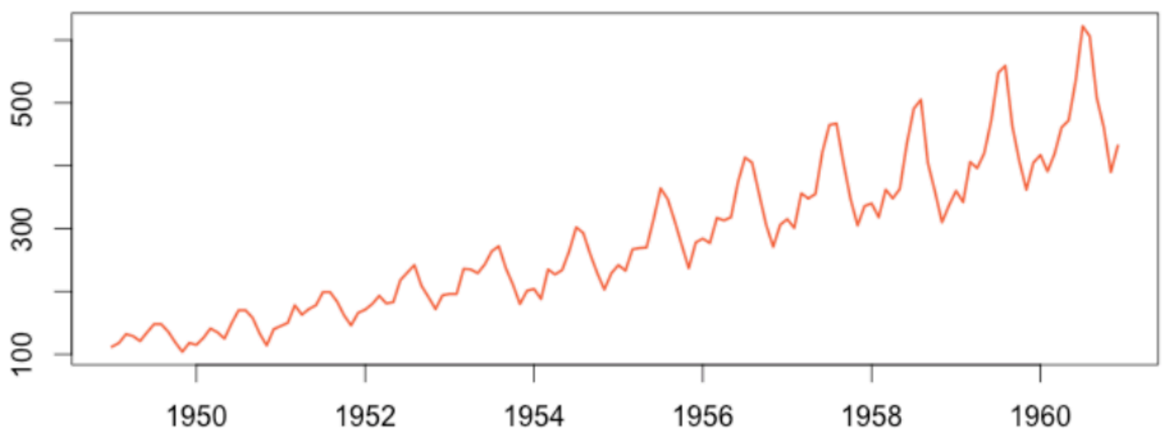
$$y_t = T_t + C_t + S_t + e_t$$



Australian beer production – The seasonal variation looks constant; it doesn't change when the time series value increases. We should use the **additive model.**

Multiplicative decomposition

$$y_t = T_t \cdot C_t \cdot S_t \cdot e_t$$



Airline Passenger Numbers – As the time series increases in magnitude, the seasonal variation increases as well. Here we should use the **multiplicative model.**

Step-by-step example of additive vs multiplicative model

TS DECOMPOSITION

Classical decomposition (ETS)

1. Compute **trend-cycle** component with m-MA (moving average, m=2k+1) → T_t
2. Calculate the detrended series: $y_t – T_t$
3. Estimate the **seasonal** component by averaging over all values of that season (ex. Monthly) → S_t
4. Calculate the **errors** as $y_t – T_t – S_t$

For multiplicative model, simply divide instead of subtracting.

M. Kendall and A. Stuart (1983) *The Advanced Theory of Statistics*, Vol.3, Griffin. pp. 410–414.

STL decomposition

Decompose a time series into **s**easonal, **t**rend and irregular components using **l**oess.

1. Compute the **seasonal** component by loess smoothing of that season → S_t
2. Loess smooth $y_t – S_t$ to find the **trend** → T_t. Apply it to S_t.
3. Iterate
4. Calculate the **errors** as $y_t – T_t – S_t$

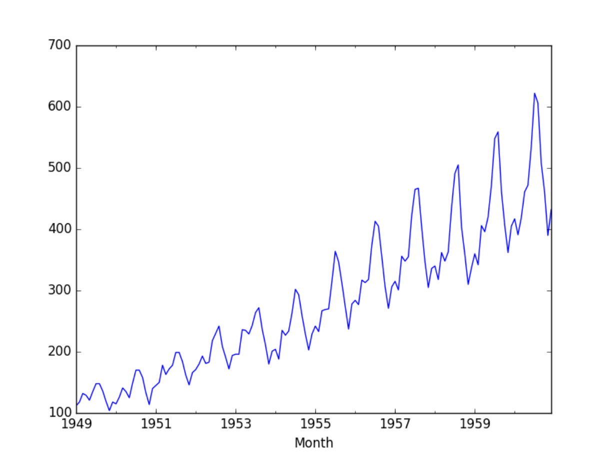
- ✓ The seasonal component may change over time.
- ✓ Robust to outliers (included in the error term).
- x Not valid for multiplicative models.

R. B. Cleveland, W. S. Cleveland, J.E. McRae, and I. Terpenning (1990) STL: A Seasonal-Trend Decomposition Procedure Based on Loess. *Journal of Official Statistics*, **6**, 3–73.

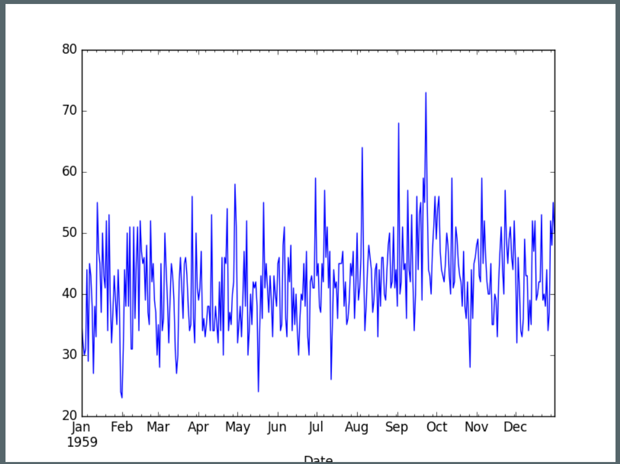
[simplified theory]

[TECH. ASIDE] STATIONARY PROCESS

- A TS or process is said to be **stationary** if the joint distribution of N samples does not change over time, i.e.
- A TS or process is said to be **stationary** if it **statistical properties (mean, variance,…) remain constant over time:**
  - Constant mean
  - Constant variance
  - Autocovariance does not depend on time



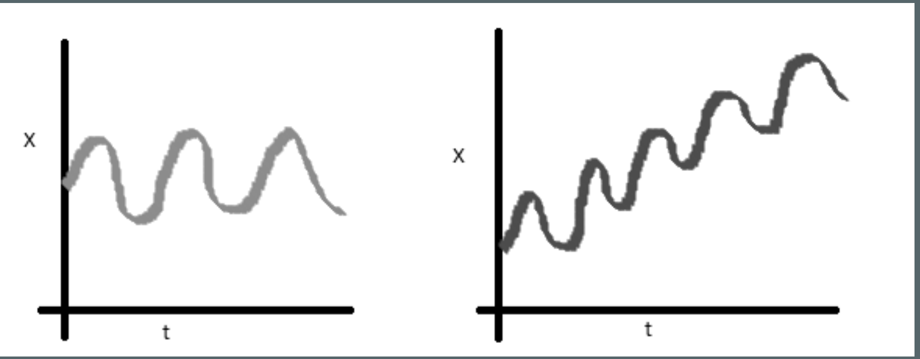
Airline passengers, **non-stationary** process

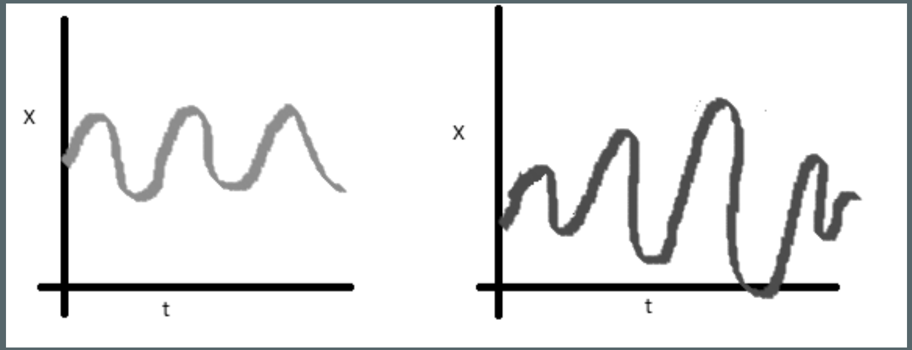


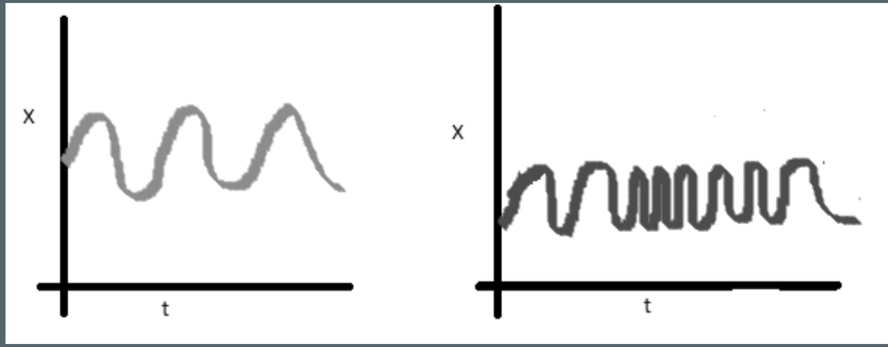
Daily female births, **stationary** process

- Most of the TS models work on the assumption that the TS is stationary. Also, the theories related to stationary series are more mature and easier to implement as compared to non-stationary series.

[TECH. ASIDE] **STATIONARY PROCESS - EXERCISE**



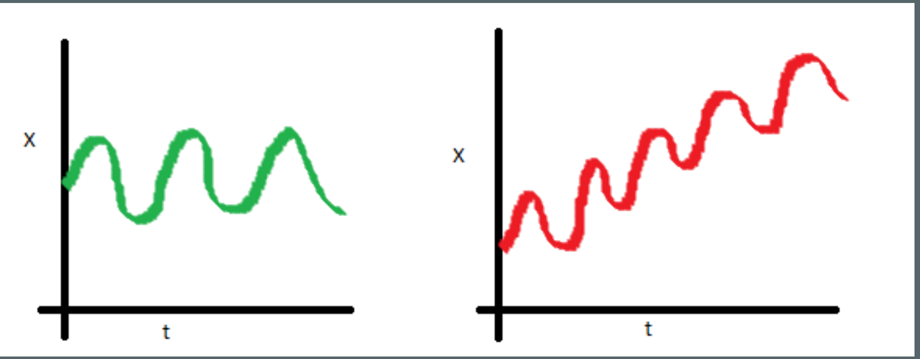




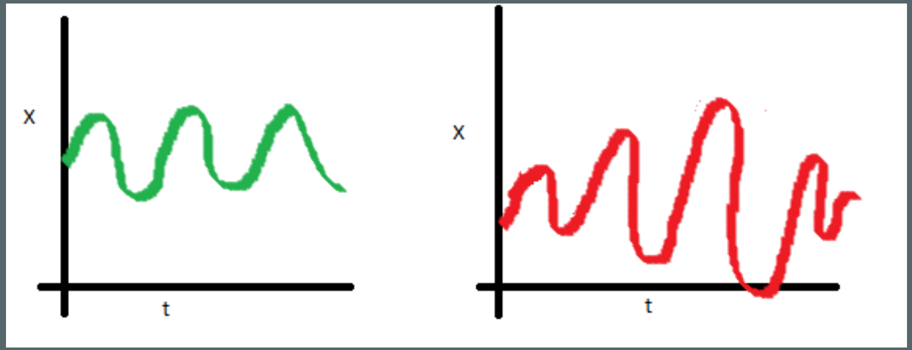
Images taken from here

[simplified theory]

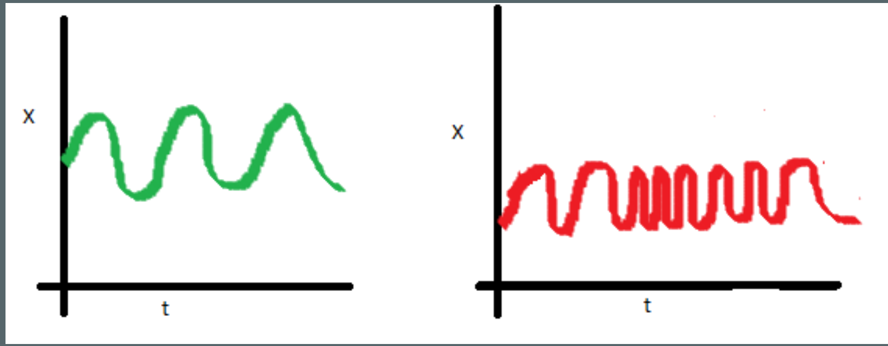
[TECH. ASIDE] STATIONARY PROCESS - EXERCISE



Mean is not constant



Variance is not constant



Covariance of y_t and y_{t+m} is not constant

Images taken from here

[simplified theory]

[TECH. ASIDE] CHECK FOR STATIONARY PROCESS

Visual Inspection

- plot the TS and visually check if there is any kind of trend or seasonality.

Rolling Statistics:

- compute moving average or moving variance and see if it varies with time.
- Split data in N (two/three) partitions and compare mean and variance

Statistical Tests, Augmented Dickey-Fuller Test:

- Null Hypothesis: TS has a unit root and hence is non-stationary.
- If the test statistic (p-value) is small, we can reject the null hypothesis and say that the series is stationary.

[simplified theory]

[TECH. ASIDE] WHY DO WE CARE FOR STATIONARY?

"If we fit a stationary model to data, we assume our data are a realization of a stationary process. So our first step in an analysis should be to check whether there is any evidence of a trend or seasonal effects and, if there is, remove them."

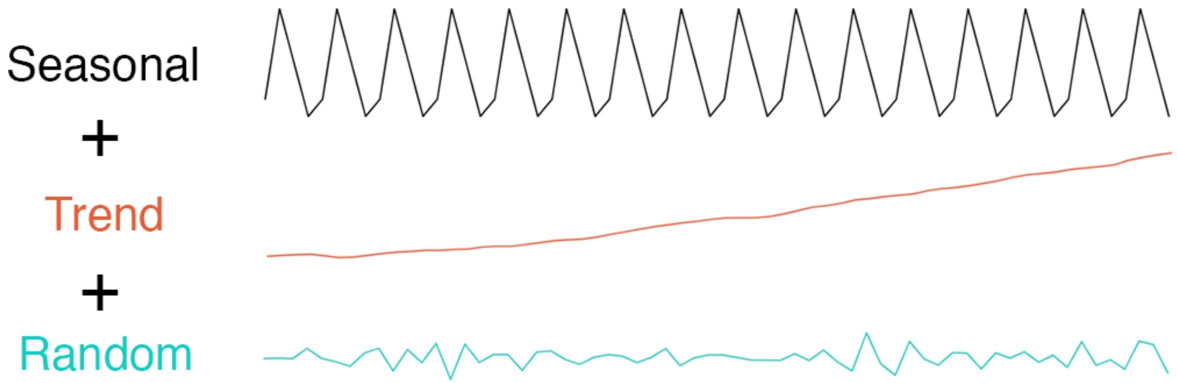
"Introductory Time Series with R", Page 122

Most of traditional TS forecasting models assume data is stationary, so make the TS stationary and train models on the stationary TS.

MAKING A TS STATIONARY

Decomposing

- Model the trend and seasonality
- Remove it from the data
- Train models on the residuals.



Differencing

- Take the difference of the observation at time t with the observation at time t-1.
- This is called *lagging*.
- Despite its simplicity, this algorithm usually works very well in practice.
- We may need to compute lag(2).

TS DECOMPOSITION - CODE

R

```
data <- ts(data_no_ts, start, frequency)

adf.test(data, k)

decomp <- decompose(data, type = "multiplicative")
decomp$trend
decomp$seasonal
decomp$random


stl_decomp = stl(data, "periodic")     #does not use loess for seasonal estimation
stl_decomp = stl(data, s.window=13)    #uses loess with window = s.window
# wacht out! Only for additive models

# stl$time.series contains 3 columns: seasonal, trend and remainder
stl_seasonality <- stl_decomp$time.series[,1]
stl_trend <- stl_decomp$time.series[,2]
stl_remain <- stl_decomp$time.series[,3]
```

TS DECOMPOSITION - CODE

Python

```
from statsmodels.tsa.stattools import adfuller

dftest = adfuller(ts, autolag='AIC')
print('Test statistic: {}'.format(dftest[0]))
print('p-value: {}'.format(dftest[1]))
print('Lag: {}'.format(dftest[2]))
print('Number of observations: {}'.format(dftest[3]))
for key, value in dftest[4].items():
    print('Critical Value ({}) = {}'.format(key, value))


from statsmodels.tsa.seasonal import seasonal_decompose

data = data.set_index('Month')
decomposition = seasonal_decompose(data)

trend = decomposition.trend
seasonal = decomposition.seasonal
residual = decomposition.resid
```

Let's code!

ts_01_decomposition_before_class.R

TIME SERIES

AGENDA

- **What is a time series?**
- **TS decomposition**
  - `code`
- **Forecasting**
  - As a Machine Learning problem
  - With Exponential Smoothing
  - As a stochastic problem: AR, MA, ARMA, ARIMA, ARCH, GARCH
  - Selecting the model that best fits the data
- **How good is our model?**
  - `code`
- **Additional material**
  - Clustering
  - TS Forecasting with Deep Learning
  - Final remarks
  - TS in R and Python
  - Bibliography and resources

Forecasting

Predicting what the TS will do in the future

FORECASTING

Ideas?

Last value
Same value of last year
Mean/median of n last values
…

MODELING THE TS AS A MACHINE LEARNING PROBLEM

Model the problem as a **regression problem**, where:

- X_{t+1} is the dependent variable
- $\{X_j\}$ j = 0, …, t are the features

Question: **Does it make sense?**

MODELING THE TS AS A MACHINE LEARNING PROBLEM

Model the problem as a **regression problem**, where:

- X_{t+1} is the dependent variable
- $\{X_j\}$ j = 0, …, t are the features

If there are many TS, applying ML may be meaningful

Does it make sense?

- Regression assumes variables are uncorrelated
- We need data. Ex: think of female births TS.
- …

MODELING THE TS AS A MACHINE LEARNING PROBLEM

Model the problem as a **regression problem**, where:

- $\{y_t\}$ is the dependent (or forecast) variable
- $\{x_t^j\}$, j = 1, …, n are the features, we have n TS
- Each t is a “data point”

Examples:

- Predict US *consumption* based on *personal income*
- Predict US *consumption* based on *personal income*, *personal savings* and *unemployment*.

MODELING THE TS WITH EXPONENTIAL SMOOTHING

Recall - Simple forecasting methods:

- Last value: $y_t = y_{t-1}$
- Mean of past values $y_t = avg(y_{t-j})$ j=1,…, t-1

We can see both approaches as a weighted mean: $y_t = avg(w_i y_{t-i})$

- Last value: w_i=0 for all i except i=1
- Mean of past values: $w_i = 1/(t-1)$ for all I

Exponential smoothing

-make w_i bigger for recent observations and smallest for oldest observations
-make it decrease exponentially as observations come from further in the past

MODELING THE TS WITH EXPONENTIAL SMOOTHING

Exponential smoothing

-make w_i bigger for recent observations and smallest for oldest observations
-make it decrease exponentially as observations come from further in the past

- **Simple Exponential Smoothing:**

  $S_t = \alpha y_t + (1-\alpha)S_{t-1}$ $\qquad$ α=smoothing constant, $0 \leq \alpha \leq 1$

  $S_t = \alpha y_t + (1-\alpha)\,[\alpha y_{t-1}+(1-\alpha)S_{t-2}] = \alpha y_t + \alpha(1-\alpha)y_{t-1} + (1-\alpha)^2 S_{t-2} = \ldots$

  $= \alpha y_t + \alpha(1-\alpha)y_{t-1} + \alpha(1-\alpha)^2 y_{t-2} + \ldots = \Sigma_k \alpha(1-\alpha)^k y_{t-k} + (1-\alpha)^t S_0, \quad k=0,\ldots, t$

Does not work well in practice → other forms of Exponential Smoothing:

- Holt's method: allow for trends
- Winter's method: allow for seasonality

→ **Holt-Winter's method**

MODELING THE TS AS A STOCHASTIC PROCESS

stochastic ≈ random

stochastic process = sequence of random variables X_t

TS is a realization of each random variable

Observations are on discrete time

There is a relation between future and past values

AUTOREGRESSIVE - AR(P)

- **AR(p)**
- MA(q)
- ARMA(p, q)
- ARIMA(p, d, q)
- ARCH(q)
- GARCH(p,q)

$$X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t$$

- X_t variables, φ_i parameters, ε_t white noise*, *c* constant
- AR(1) AR(2)
- Futures values only depend linearly on its own *p* previous values
- ε is also called *error*, *innovation term, shocks, residuals*…
- φ_i corresponds to how much the X_{t-i} contributes to X_t.

*white noise: iid samples from normal distribution with mean 0 and constant variance $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$

AUTOREGRESSIVE - AR(P)

$$X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t$$

$$X_t = c + \varphi_1 X_{t-1} + \varphi_2 X_{t-2} + \cdots + \varphi_p X_{t-p} + \varepsilon_t$$

- X_t variables, φ_i parameters, ε_t white noise*, *c* constant
- AR(1) AR(2)
- Futures values only depend linearly on its own *p* previous values
- ε is also called *error*, *innovation term, shocks, residuals*…
- φ_i corresponds to how much the X_{t-i} contributes to X_t.

*white noise: iid samples from normal distribution with mean 0 and constant variance $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$

[TECH. ASIDE] AUTOCORRELATION FUNCTION (ACF)

Correlation between a signal and a delayed copy of itself, as function of the delay.

$$\rho(\tau) = \mathrm{Cor}(X_t, X_{t+\tau}) = \frac{\sum_{i=1}^{N-\tau}(X_i - \cancel{\bar{X}})(X_{i+\tau} - \cancel{\bar{X}})}{\sum_{i=1}^{N}(X_i - \bar{X})^2}$$

Var(X)

Observations:

- $\rho(0) = \mathrm{Cor}(X_t, X_t) = 1$
- $\rho(\tau) = \mathrm{Cov}(X_t, X_{t+\tau}) / \mathrm{Var}(X_t)$
- We call $\gamma(\tau) = \mathrm{Cov}(X_t, X_{t+\tau})$
- $\gamma(0) = \mathrm{Var}(X_t)$
- → $\rho(\tau) = \gamma(\tau) / \gamma(0)$
- $\rho(\tau) = \rho(-\tau)$ and $\gamma(\tau) = \gamma(-\tau)$
- Sometimes we will compute $\gamma(\tau)$ and call it acf, they will only differ in dividing by the variance.

[TECH. ASIDE] PARTIAL AUTOCORRELATION FUNCTION (PACF)

Correlation between a signal and a lagged version of itself, removing the effect of intermediate lags α(k)

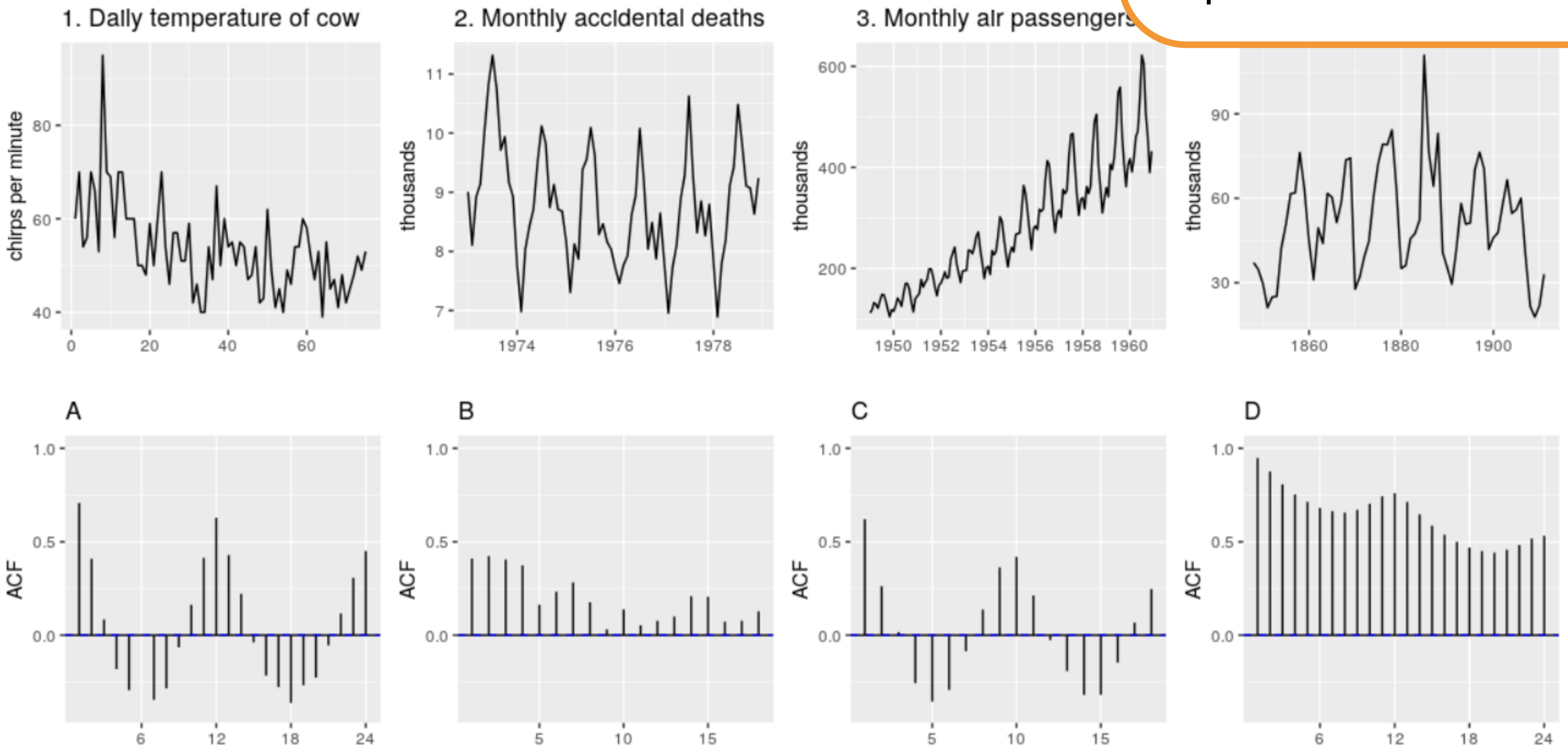
$$\alpha(1) = \mathrm{Cor}(X_t, X_{t+1}) = \rho(1)$$

$$\alpha(k) = \mathrm{Cor}(X_t, X_{t+k} | X_{t+1}, \ldots, X_{t+k-1})$$

EXERCISE

- Match each graph in the first row with its ACF in the s

SOLUTION:
1B – 2A – 3D – 4C
- Trend: ACF small lags is high; slowly decreases
- Seasonality: ACF has high peaks for seasonal lags



From otexts online bok

AR(1)

$$X_t = \varphi X_{t-1} + \varepsilon_t$$

Exercise: Compute mean, variance and autocorrelation function.

- $E[X_t]$
- $Var[X_t]$
- $Cov[X_t, X_{t+T}]$ and $Corr[X_t, X_{t+T}]$

AR(1)

$$X_t = \varphi X_{t-1} + \varepsilon_t$$

Exercise: Compute <u>mean</u>, variance and autocorrelation function.

$$
\begin{aligned}
E[X_t] &= E[\varphi X_{t-1} + \varepsilon_t] = \varphi E[X_{t-1}] + E[\varepsilon_t] \\
&= \varphi E[X_t] + 0
\end{aligned}
$$

$E[X_{t-1}] = E[X_t]$ since X_t is stationary
$E[eps_t] = 0$ since it is white noise

$$\implies \mu = \varphi \mu$$

$$\implies \mu(1-\varphi) = 0$$

$$\implies \mu = 0 \quad \text{or} \quad \varphi = 1$$

$|\varphi| < 1$ since it is stationary

$$\implies \boxed{E[X_t] = 0}$$

Note: E[Xt] = c if constant is present in definition

AR(1)

$$X_t = \varphi X_{t-1} + \varepsilon_t$$

Exercise: Compute mean, variance and autocorrelation function.

$$
\begin{aligned}
Var[X_t] &= E[X_t^2] - (E[X_t])^2 = E[\varphi^2 X_{t-1}^2 + \varepsilon_t^2 + 2\varphi X_{t-1}\varepsilon_t] - \mu^2 \\
&= \varphi^2 E[X_{t-1}^2] + E[\varepsilon_t^2] + 2\varphi E[X_{t-1}\varepsilon_t] - \mu^2 \\
&= \varphi^2 E[X_{t-1}^2] + \sigma_\varepsilon^2 + 0 - 0 \\
&= \varphi^2 Var[X_{t-1}] + \sigma_\varepsilon^2
\end{aligned}
$$

σ_ε is the standard deviation of ε_t

$E[X_{t-1}\varepsilon_t] = 0$ since ε_t is uncorrelated to previous values

$$\Longrightarrow Var[X_t] = \varphi^2 Var[X_t] + \sigma_\varepsilon^2$$

$Var[X_{t-1}] = Var[X_t]$ since X_t is stationary

$$\Longrightarrow Var[X_t](1-\varphi^2) = \sigma_\varepsilon^2 \Longrightarrow \boxed{Var[X_t] = \frac{\sigma_\varepsilon^2}{1-\varphi^2}}$$

AR(1)

$$X_t = \varphi X_{t-1} + \varepsilon_t$$

Exercise: Compute mean, variance and <u>autocorrelation</u> function.

$$\begin{aligned} Cov[X_t, X_{t+1}] &= E[X_t X_{t+1}] - E[X_t]E[X_{t+1}] \\ &= E[X_t(\varphi X_t + \varepsilon_{t+1})] - E[X_t]E[X_{t+1}] \\ &= \varphi E[X_t^2] + E[X_t \varepsilon_{t+1}] - \mu^2 \\ &= \varphi Var[X_t] \implies \boxed{Cov[X_t, X_{t+1}] = \varphi \frac{\sigma_\varepsilon^2}{1-\varphi^2}} \end{aligned}$$

We're assuming mu=0

$E[X_t \varepsilon_{t+1}] = 0$ since ε_t is uncorrelated to previous values

Exercise: Corr[X_t, X_{t+1}]?

Corr[X_t, X_{t+1}] =φ

AR(1)

$$X_t = \varphi X_{t-1} + \varepsilon_t$$

Exercise: Compute Cov[X_t, X_{t+2}] & Corr[X_t, X_{t+2}] and the general form Cov[X_t, $X_{t+\tau}$] & Corr[X_t, $X_{t+\tau}$]

Solution:

$$Cov[X_t, X_{t+2}] = \varphi^2 \frac{\sigma_\varepsilon^2}{1-\varphi^2} \qquad Corr[X_t, X_{t+2}] = \varphi^2$$

$$\boxed{Cov[X_t, X_{t+\tau}] = \varphi^{|\tau|} \frac{\sigma_\varepsilon^2}{1-\varphi^2}} \qquad \boxed{Corr[X_t, X_{t+\tau}] = \varphi^{|\tau|}}$$

AR(P)

AR(0)

$$X_t = \varepsilon_t$$

AR(1)

$$X_t = \varphi X_{t-1} + \varepsilon_t$$

AR(2)

$$X_t = \varphi_1 X_{t-1} + \varphi_2 X_{t-2} + \varepsilon_t$$

MOVING AVERAGE - MA(Q)

$$X_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} = \mu + \sum_{i=0}^{q} \theta_i \varepsilon_{t-i}$$

- μ series mean, θ_t parameters, ε_t white noise
- MA(1) MA(2)
- Futures values only depend linearly on the current and previous shocks.
- *"linear regression over current and previous shocks"*
- The ACF of MA(q) is zero at lag q+1 and greater. **Exercise** (to do in class)
  - → Empirical way of obtaining maximum lag.

MA(Q)

$$X_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} = \mu + \sum_{i=0}^{q} \theta_i \varepsilon_{t-i}$$

When is it a good model?

- A way to think about the moving average model is that it corrects future forecasts based on errors made on recent forecasts.
- Example: stock market
- More info:
  - https://stats.stackexchange.com/questions/45026/real-life-examples-of-moving-average-processes
  - https://stats.stackexchange.com/questions/107834/under-what-circumstances-is-an-ma-process-or-ar-process-appropriate

MA(1)

$$X_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1}$$

Exercise: Compute E[X_t], Var[X_t], Cov[X_t, X_{t+T}] and Corr[X_t, X_{t+T}]

MA(1)

$$X_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1}$$

Solution

$$E[X_t] = \mu + E[\varepsilon_t] + \theta_1 E[\varepsilon_{t-1}] = \mu + 0 + 0 \implies E[X_t] = \mu$$

$$Var[X_t] = E[X_t^2] - (E[X_t]^2) = E[X_t^2] - 0 = E[(\varepsilon_t + \theta_1 \varepsilon_{t-1})^2]$$

$$= E[\varepsilon_t^2] + \theta_1^2 E[\varepsilon_{t-1}^2] + 2\theta_1 E[\varepsilon_t \varepsilon_{t-1}] = Var[\varepsilon_t] + \theta_1^2 Var[\varepsilon_{t-1}]$$

$$\implies Var[X_t] = \sigma_\varepsilon^2 (1 + \theta_1^2)$$

$E[\varepsilon_t \varepsilon_{t-1}] = Cor[\varepsilon_t, \varepsilon_{t-1}] = 0$ since ε_t is uncorrelated to previous errors

MA(1)

- AR(p)
- **MA(q)**
- ARMA(p, q)
- ARIMA(p, d, q)
- ARCH(q)
- GARCH(p,q)

$$X_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1}$$

Solution

$$Cov[X_t, X_{t+1}] = \ldots = \theta_1 \sigma_\varepsilon^2 \implies Corr[X_t, X_{t+1}] = \frac{\theta_1}{1+\theta_1^2}$$

$$\begin{aligned} Cov[X_t, X_{t+h}] &= E[(\varepsilon_t + \theta_1 \varepsilon_{t-1})(\varepsilon_{t+h} + \theta_1 \varepsilon_{t+h-1})] \\ &= E[\varepsilon_t \varepsilon_{t+h}] + E[\varepsilon_t \varepsilon_{t+h-1}] + \theta_1 E[\varepsilon_t \varepsilon_{t+h}] + \theta_1^2 E[\varepsilon_{t-1} \varepsilon_{t+h-1}] \end{aligned}$$

It is different from 0 when subscripts of epsilon are equal. This happens when h=0 (variance) or h=1 (before),
So $Cov[X_t, X_{t+h}] = 0$ for h>1

ARMA(P,Q)

$$X_t = c + \varepsilon_t + \underbrace{\sum_{i=1}^{p} \varphi_i X_{t-i}}_{\text{AR(p)}} + \underbrace{\sum_{i=1}^{q} \theta_i \varepsilon_{t-i}}_{\text{MA(q)}}$$

- AR component (regressing on past values) + MA component (regressing on error terms)
- ARMA models can be estimated by using the **Box–Jenkins methodology**:
  - Estimate *p* with PACF and *q* with ACF.
  - Estimate the parameters by least squares
  - (more on this in next section)
- **Exercise**: compute $E[X_t]$ for ARMA(p,q)

ARIMA(P, D, Q)

$$X'_t = c + \varepsilon_t + \sum_{i=1}^{p} \varphi_i X'_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}$$

$$X'_t = X_t - X_{t-d}$$

- AR component (regressing on past values)

  \+ MA component (regressing on error terms)

  \+ I component (integrating)
- I(d) means that we need to *differentiate* the TS d times to make it stationary
- ARIMA(p, d, q) = ARMA(p, q) on the lagged TS

ARCH(Q) / GARCH(P,Q)[1]

- ARCH(q): the error variance follows an AR(q) process

$$\varepsilon_t = \sigma_t z_t, \quad z_t \text{ white noise}$$

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2, \quad \alpha_0 > 0,\ \alpha_i \geq 0, \forall i > 0$$

- GARCH(p,q): the error variance follows an ARMA(p,q) process

$$\varepsilon_t | \psi_{t-1} \sim \mathcal{N}(0, \sigma_t^2)$$

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{i=1}^{q} \beta_i \sigma_{t-i}^2$$

Engle, Robert F. (1982). "Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation". *Econometrica*. **50** (4): 987–1007

TS FORECASTING - CODE

R

```
# stats
acf(data)
pacf(data)
model <- arima(data, order = (p,d,q), seasonal)
predict(model, n.ahead)

# forecast
auto.arima(data)
forecast(model, level, h)
```

Python

```
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
plot_acf(series)
plot_pacf(series)

from statsmodels.tsa.arima_model import ARIMA
model = ARIMA(history, order=(5,1,0))
model_fit = model.fit()
output = model_fit.forecast()
yhat = output[0]
```

Selecting the model that best fits the data

BOX-JENKINS METHODOLOGY

"Every stationary process can be approximated by an ARMA process"

Box & Jenkins, 1976

BOX-JENKINS METHODOLOGY

*"Every **stationary** process can be **approximated** by an **ARMA** process"*

Box & Jenkins, 1976

1. Model identification
2. Parameter estimation
3. Model diagnosis
4. Prediction

BOX-JENKINS METHODOLOGY

1. **Model identification**
   - Check for stationarity (seen before):
     - Visual
     - Moving statistics
     - Statistical test: Augmented Dicky-Fuller
     - Check for units roots (new!)
   - Selecting p and q
2. Parameter estimation
3. Model diagnosis
4. Prediction

BOX-JENKINS METHODOLOGY

1. **Model identification. Check stationary with characteristic polynomial (theory)**

- The condition for stationarity of a general AR(p) model is that the roots of $1 - \phi_1 z - \phi_2 z^2 - \ldots - \phi_p z^p = 0$ all lie outside the unit circle.
- A stationary AR(p) model is required for it to have an MA(∞) representation.
- Example 1: Is $y_t = y_{t-1} + u_t$ stationary?

  The characteristic root is 1, so it is a unit root process (so non-stationary)
- Example 2: Is $y_t = 3y_{t-1} - 0.25y_{t-2} + 0.75y_{t-3} + u_t$ stationary?

  The characteristic roots are 1, 2/3, and 2. Since only one of these lies outside the unit circle, the process is non-stationary.

from https://slideplayer.com/slide/4595589/

BOX-JENKINS METHODOLOGY

1. **Model identification. Selecting p and q**

ACF and PAC come to help!

- ACF measures the relation between X_t and X_{t+n}, including X_j for j=t+1,… t+n
- PACF measures the direct relation between X_t and X_{t+n}, excluding X_j for j=t+1,…, t+n-1

Example: AR(p): $X_t = a_1X_{t-1} + a_2X_{t-2} + \ldots + a_pX_{t-p} + \varepsilon_t$.

- PACF(k) will be high for k=1,…, p, but will be 0 afterwards (by definition)

→ Plot PACF and see at what lag it goes down to 0 →p

Example: MA(q): $X_t = \varepsilon_t + b_1\varepsilon_{t-1} + b_2\varepsilon_{t-2} + \ldots + b_q\varepsilon_{t-q}$

- ACF(k) will be 0, k> q (seen in the exercise)

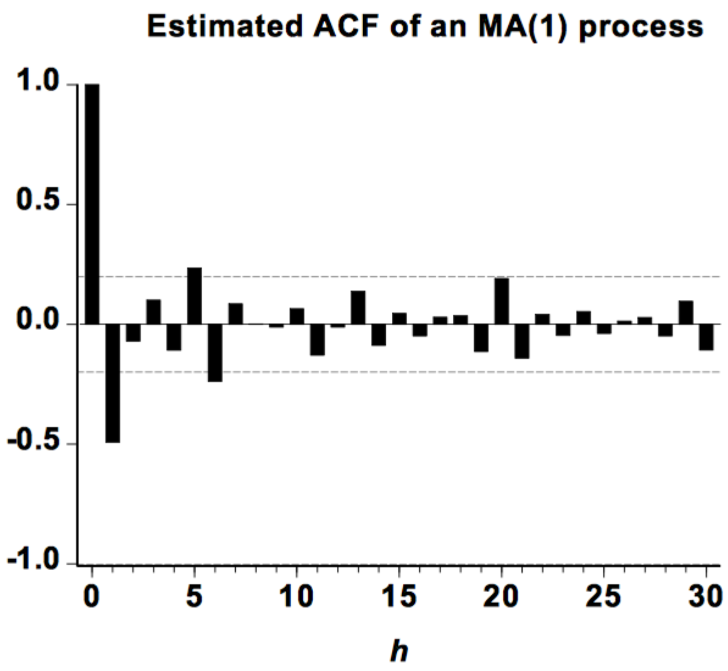
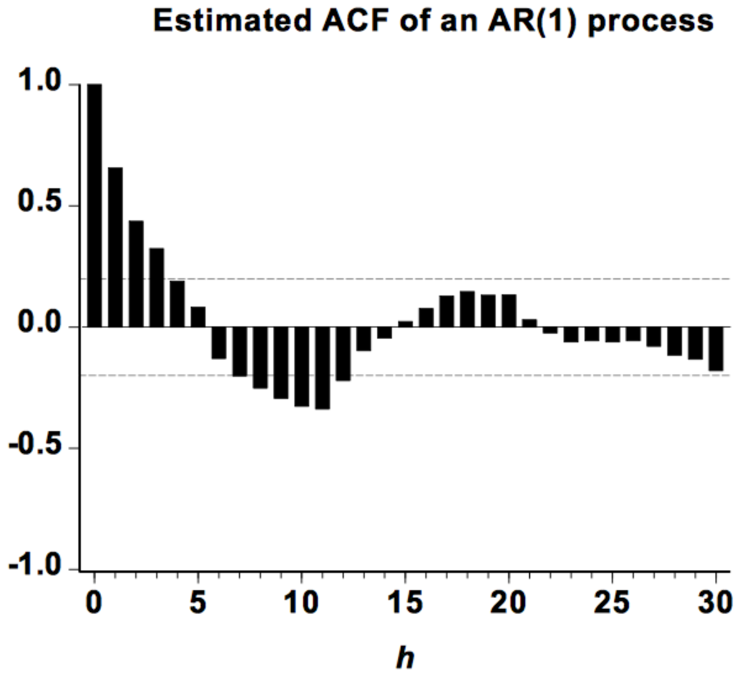
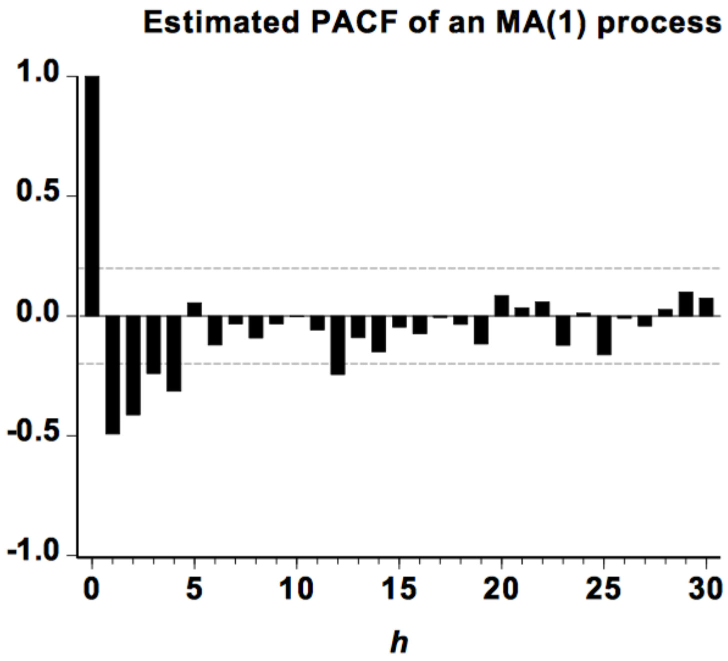
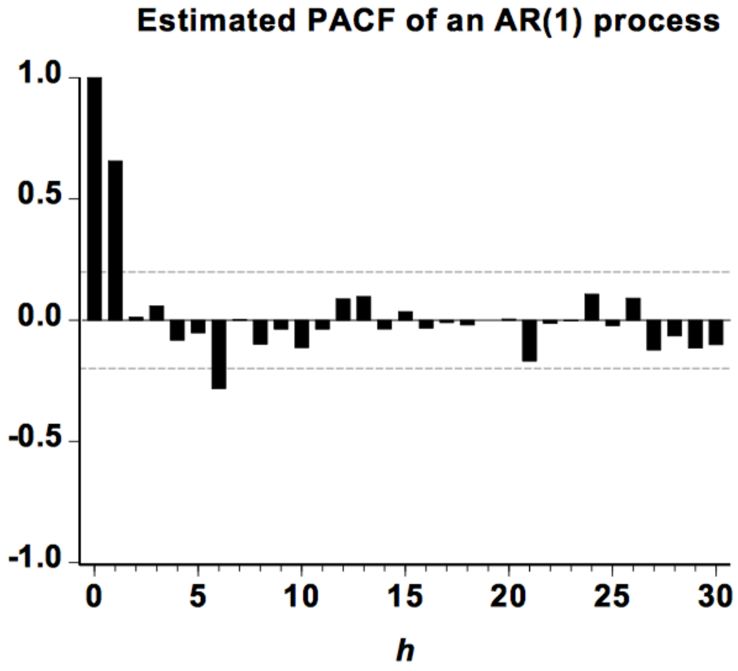
→ Plot ACF and see at what lag it goes down to 0 →q

BOX-JENKINS METHODOLOGY

1. **Model identification. Selecting p and q**

Prozess	ACF ($\rho_X(h)$)	PACF ($\pi_X(h)$)
AR(p)	infinite (dampened exponential or sinusoidal waves	finite $\pi_X(h) = 0$ for $h > p$
MA(q)	finite $\rho_X(h) = 0$ for $h > q$	infinite (dampened exponential or sinusoidal waves
ARMA(p, q)	as AR(p) for $h > q$	as MA(q) for $h > p$

Source: Münster University, Time Series course


Estimated ACF of an MA(1) process
1.0
0.5
0.0
-0.5
-1.0
0
5
10
15
20
25
30
h

Estimated ACF of an AR(1) process
1.0
0.5
0.0
-0.5
-1.0
0
5
10
15
20
25
30
h

Estimated PACF of an MA(1) process
1.0
0.5
0.0
-0.5
-1.0
0
5
10
15
20
25
30
h

Estimated PACF of an AR(1) process
1.0
0.5
0.0
-0.5
-1.0
0
5
10
15
20
25
30
h

AR(1) ~ MA(∞)

$$
\begin{aligned}
X_t &= \varphi_1 X_{t-1} + \varepsilon_t \\
&= \varphi_1(\varphi_1 X_{t-2} + \varepsilon_{t-1}) + \varepsilon_t \\
&= \varphi_1^2 X_{t-2} + (\varphi_1 \varepsilon_{t-1} + \varepsilon_t) \\
&= \varphi_1^2(\varphi_1 X_{t-3} + \varepsilon_{t-2}) + (\varphi_1 \varepsilon_{t-1} + \varepsilon_t) \\
&= \varphi_1^3 X_{t-3} + (\varphi_1^2 \varepsilon_{t-2} + \varphi_1 \varepsilon_{t-1} + \varepsilon_t) \\
&= \dots \implies X_t = \sum_{k=0}^{\infty} \varphi_1^k \varepsilon_{t-k}
\end{aligned}
$$

MA(Q) ~ AR(∞)

Exercise

BOX-JENKINS METHODOLOGY

1. **Model identification. Selecting p and q**

Remarks:

- High peak in PACF at lag 1 should imply AR(1) component. However, this could also imply that the series should be differenced before modeling it.
- The autocorrelation function for an ARMA process will display combinations of behavior derived from the AR and MA parts, but for lags beyond q, the acf will simply be identical to the individual AR(p) model.
- It is possible for an AR term and an MA term to cancel each other's effects, so if a mixed AR-MA model seems to fit the data, also try a model with one fewer AR term and one fewer MA term.

- **There is no unique solution in model specification. We can have more than one suitable model.**

BOX-JENKINS METHODOLOGY

1. Model identification
2. **Parameter estimation**
   - OLS
   - Maximum Likelihood
3. Model diagnosis
4. Prediction

BOX-JENKINS METHODOLOGY

2. **Parameter estimation. AR coefficients with OLS estimation.**

Suppose X_t follows AR(p), then $X_t = \varphi_1 X_{t-1} + \varphi_2 X_{t-2} + \ldots + \varphi_p X_{t-p} + \varepsilon_t$

- At time t, X_j is known for j=0, 1, …, t
- φ_i are unkown

→ consider X_t as a regression problem on X_j (j={t-1, …, t-p}) and estimate φ_i

Problem?

- X_t is correlated with X_j
  - Solution: Neusser, 2006: For AR(p) models, the OLS estimators are consistent and asymptotically efficient.

BOX-JENKINS METHODOLOGY

2. **Parameter estimation. MA coefficients with Maximum Likelihood estimation.**

Suppose X_t follows MA(q), then $X_t = \varepsilon_t + \theta_1\varepsilon_{t-1} + \dots + \theta_q\varepsilon_{t-q}$

- We can't regress on ε_j since those can't be observed
- Estimate the parameters θ_j through maximum likelihood on the joint probability on $\{X_t\}$

BOX-JENKINS METHODOLOGY

1. Model identification
2. Parameter estimation
3. **Model diagnosis**
   - Metrics and residuals (next section)
4. Prediction

BOX-JENKINS METHODOLOGY

1. Model identification
2. Parameter estimation
3. Model diagnosis
4. **Prediction**

BOX-JENKINS METHODOLOGY

*"Every **stationary** process can be **approximated** by an **ARMA** process"*
Box & Jenkins, 19762

Remarks:

- We need the process to be **stationary**. "In the economic and social fields, real series are never stationary however much differencing is done", Commandeur & Koopman, 2007.
- How good is **approximate**?
- If we have an ARIMA process, we just need to first differentiate and later apply ARMA.

How good is our model?

Time series evaluation

METRICS

It is a **regression** problem, so we can use regression metrics:

- MAE (Mean Average Error)
- RMSE (Root Mean Squared Error)
- MPE (Mean Percentage Error)
- MAPE (Mean Absolute Percentage Error)
- R^2
- AIC / BIC (Akaike Information Criterion / Bayesian Information Criterion)

RESIDUALS DIAGNOSTIC

If the model is a good fit, we hope the residuals to be **white noise**:

- Follow a **normal distribution** (mean = 0)
  - Shapiro Test (H0: X is normally distributed) (→ pvalue < 0.05 reject the null → data are not normally distributed)
- **Uncorrelated** (acf = 0)
  - Ljung-Box Test (H0: data are independently distributed) (→ pvalue < 0.05 reject the null → data are not independent)

MODEL EVALUATION SETUP

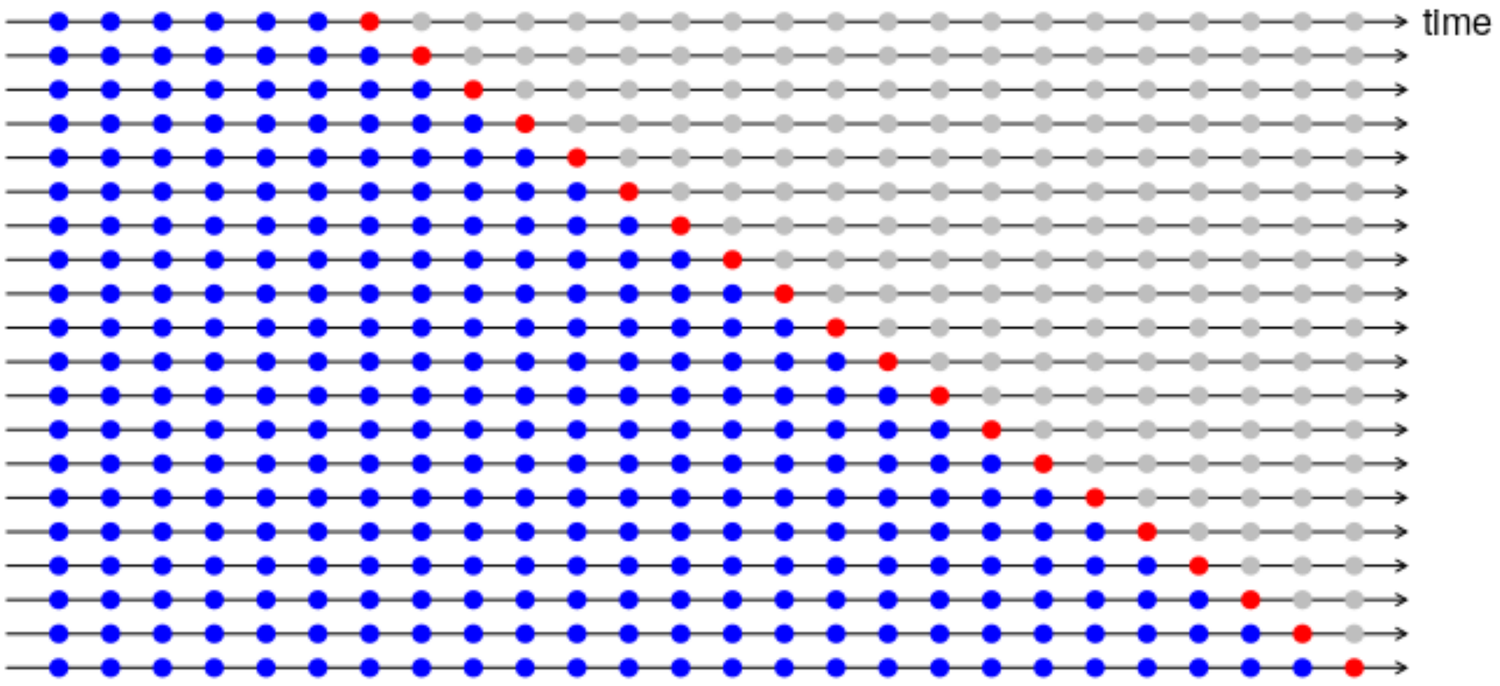
Option 1: standard setup

- N total points
- Train set: k first points (k << N)
- Test set: N-k lastest points

MODEL EVALUATION SETUP

Option 2: averaging over the history (cv)

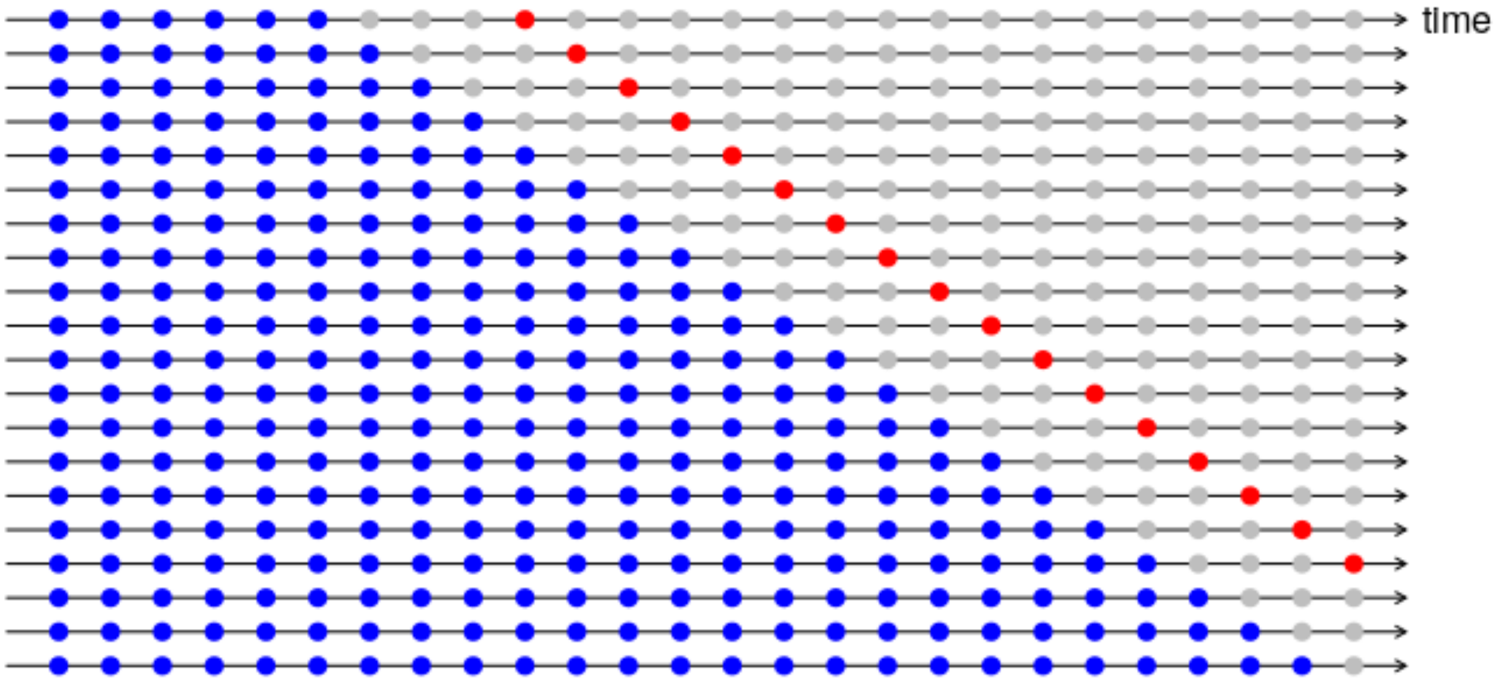
- N total points
- Train set: select k first points from $k = k_0$ to $k=N-1$
- Test set: point k+1



MODEL EVALUATION SETUP

Option 2: averaging over the history (cv)

- N total points
- Train set: select k first points from $k = k_0$ to $k=N-1$
- Test set: point $k+n_0$



BOX-JENKINS METHODOLOGY - SUMMARY

1. Plot series
   - If variance is not constant (multiplicative model) → take logarithm to make it additive
2. If there is a trend (i.e. non stationary) or non stationary for other reason → make it stationary:
   - Differentiate → I(d)
   - Decompose and take random part
3. Check it is stationary with Augmented Dickey-Fuller test
4. Plot ACF and PACF to obtain p, q in ARMA(p,q)
   - Propose several models and test each of them
5. Select the most adequate model:
   - AIC / BIC
   - Residuals
   - Check for significance of the parameters
   - Other regression metrics
6. Forecast

(EXTRA) SEASONAL ARIMA

- We can eliminate seasonality by taking difference with the lagged time series.
- But we can also eliminate seasonality by means of **seasonal differences**.
- There is seasonal dependence when the value of the time series at time T can be used to estimate value of ts at time T+S.
- Seasonal Arima → generalization of ARIMA for ts with **regular dependence** and **seasonal dependence**.

$$\text{ARIMA } (p, d, q) \times (P, D, Q)_S$$

- How are ACFs and PACFs of seasonal arima?:
  - We have to look only at lags S*k
  - ACF: peaks at S*k lag means MA(k) seasonal component.
  - PACF: peaks at S*k lag means AR(k) seasonal component.

SOME RULES FOR TS MODELLING

Identifying the order of differencing and the constant:

- Rule 1: If the series has positive autocorrelations out to a high number of lags (say, 10 or more), then it probably needs a higher order of differencing.
- Rule 2: If the lag-1 autocorrelation is zero or negative, or the autocorrelations are all small and patternless, then the series does *not* need a higher order of differencing. If the lag-1 autocorrelation is -0.5 or more negative, the series may be overdifferenced. **BEWARE OF OVERDIFFERENCING.**
- Rule 3: The optimal order of differencing is often the order of differencing at which the standard deviation is lowest. (Not always, though. Slightly too much or slightly too little differencing can also be corrected with AR or MA terms. See rules 6 and 7.)
- Rule 4: A model with <u>no</u> orders of differencing assumes that the original series is stationary (among other things, mean-reverting). A model with <u>one</u> order of differencing assumes that the original series has a constant average trend (e.g. a random walk or SES-type model, with or without growth). A model with <u>two</u> orders of total differencing assumes that the original series has a time-varying trend (e.g. a random trend or LES-type model).
- Rule 5: A model with <u>no</u> orders of differencing normally includes a constant term (which allows for a non-zero mean value). A model with <u>two</u> orders of total differencing normally does <u>not</u> include a constant term. In a model with <u>one</u> order of total differencing, a constant term should be included if the series has a non-zero average trend.

https://people.duke.edu/~rnau/411arim3.htm

SOME RULES FOR TS MODELLING

Identifying the numbers of AR and MA terms:

- Rule 6: If the partial autocorrelation function (PACF) of the differenced series displays a sharp cutoff and/or the lag-1 autocorrelation is positive--i.e., if the series appears slightly "underdifferenced"--then consider adding one or more AR terms to the model. The lag beyond which the PACF cuts off is the indicated number of AR terms.
- Rule 7: If the autocorrelation function (ACF) of the differenced series displays a sharp cutoff and/or the lag-1 autocorrelation is negative--i.e., if the series appears slightly "overdifferenced"--then consider adding an MA term to the model. The lag beyond which the ACF cuts off is the indicated number of MA terms.
- Rule 8: It is possible for an AR term and an MA term to cancel each other's effects, so if a mixed AR-MA model seems to fit the data, also try a model with one fewer AR term and one fewer MA term--particularly if the parameter estimates in the original model require more than 10 iterations to converge. **BEWARE OF USING MULTIPLE AR TERMS AND MULTIPLE MA TERMS IN THE SAME MODEL.**
- Rule 9: If there is a unit root in the AR part of the model--i.e., if the sum of the AR coefficients is almost exactly 1--you should reduce the number of AR terms by one and increase the order of differencing by one.
- Rule 10: If there is a unit root in the MA part of the model--i.e., if the sum of the MA coefficients is almost exactly 1--you should reduce the number of MA terms by one and reduce the order of differencing by one.
- Rule 11: If the long-term forecasts* appear erratic or unstable, there may be a unit root in the AR or MA coefficients.

https://people.duke.edu/~rnau/411arim3.htm

SOME RULES FOR TS MODELLING

Identifying the seasonal part of the model:

- Rule 12: If the series has a strong and consistent seasonal pattern, then you must use an order of seasonal differencing (otherwise the model assumes that the seasonal pattern will fade away over time). However, never use more than one order of seasonal differencing or more than 2 orders of total differencing (seasonal+nonseasonal).
- Rule 13: If the autocorrelation of the appropriately differenced series is positive at lag s, where s is the number of periods in a season, then consider adding an SAR term to the model. If the autocorrelation of the differenced series is negative at lag s, consider adding an SMA term to the model. The latter situation is likely to occur if a seasonal difference has been used, which should be done if the data has a stable and logical seasonal pattern. The former is likely to occur if a seasonal difference has not been used, which would only be appropriate if the seasonal pattern is not stable over time. You should try to avoid using more than one or two seasonal parameters (SAR+SMA) in the same model, as this is likely to lead to overfitting of the data and/or problems in estimation.

https://people.duke.edu/~rnau/411arim3.htm

Let’s code!

ts_02_forecasting_before_class.R

Additional material

Clustering, Deep Learning, TS in practice, coding, references,…

MULTIVARIATE TIME SERIES

$\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ Time series measured as time t

Example, for n=2:

$\mathbf{x}_1 = x_{11}, x_{12}, x_{13}, \ldots, x_{1(t-1)}, x_{1t,}$

$\mathbf{x}_2 = x_{21}, x_{22}, x_{23}, \ldots, x_{2(t-1)}, x_{2t}$

Example: temperature, wind speed and precipitations each day.

Objective: we want to model and forecast all the series at the same time, assuming each of them affect the other.

Vector ARIMA
(VARMA / VARIMA)

ARMAV

Vector Autoregression
(VAR)

Example: Vector Autoregressive – VAR(1) with 2 variables

$$x_{1,t} = c_1 + \phi_{11,1} x_{1,t-1} + \phi_{12,1} x_{2,t-1} + e_{1,t}$$
$$x_{2,t} = c_2 + \phi_{21,1} x_{1,t-1} + \phi_{22,1} x_{2,t-1} + e_{2,t}$$

$$\begin{pmatrix} x_{1,t} \\ x_{2,t} \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} + \begin{pmatrix} \phi_{11,1} & \phi_{12,1} \\ \phi_{21,1} & \phi_{22,1} \end{pmatrix} \begin{pmatrix} x_{1,t-1} \\ x_{2,t-1} \end{pmatrix} + \begin{pmatrix} e_{1,t} \\ e_{2,t} \end{pmatrix} \qquad \mathbf{x}_t = \mathbf{c} + \boldsymbol{\Phi}_\mathbf{1} \mathbf{x}_{t-1} + \varepsilon_t$$

Example: Vector Autoregressive – VAR(2) with 2 variables

$$\begin{pmatrix} x_{1,t} \\ x_{2,t} \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} + \begin{pmatrix} \phi_{11,1} & \phi_{12,1} \\ \phi_{21,1} & \phi_{22,1} \end{pmatrix} \begin{pmatrix} x_{1,t-1} \\ x_{2,t-1} \end{pmatrix} + \begin{pmatrix} \phi_{11,2} & \phi_{12,2} \\ \phi_{21,2} & \phi_{22,2} \end{pmatrix} \begin{pmatrix} x_{1,t-2} \\ x_{2,t-2} \end{pmatrix} + \begin{pmatrix} e_{1,t} \\ e_{2,t} \end{pmatrix}$$

$$\mathbf{x}_t = \mathbf{c} + \boldsymbol{\Phi}_\mathbf{1} \mathbf{x}_{t-1} + \boldsymbol{\Phi}_\mathbf{2} \mathbf{x}_{t-2} + \varepsilon_t$$

MULTIVARIATE TIME SERIES

Modeling:

- Parameters are estimated using OLS.
- How to determine number of lags?
  - Trial-and-error. Metrics (residuals, AIC, BIC,…)
  - VARselect() function in vars package in R

Stationarity:

- If the TSs are non-stationary, take differences on the data, then fit a VAR model.
- Cointegration: “the TSs are related and needs the same number of differencing”
  - Johansen’s test on the original TS.

MULTIVARIATE TIME SERIES - CODE

R

```
# packages
# marima, ARMAX, TSA::arimax, dse, MTS, MARSS, arima(…, xreg)

vars::VARselect()
vars::VAR(data, p=3, type="const")
```

Python

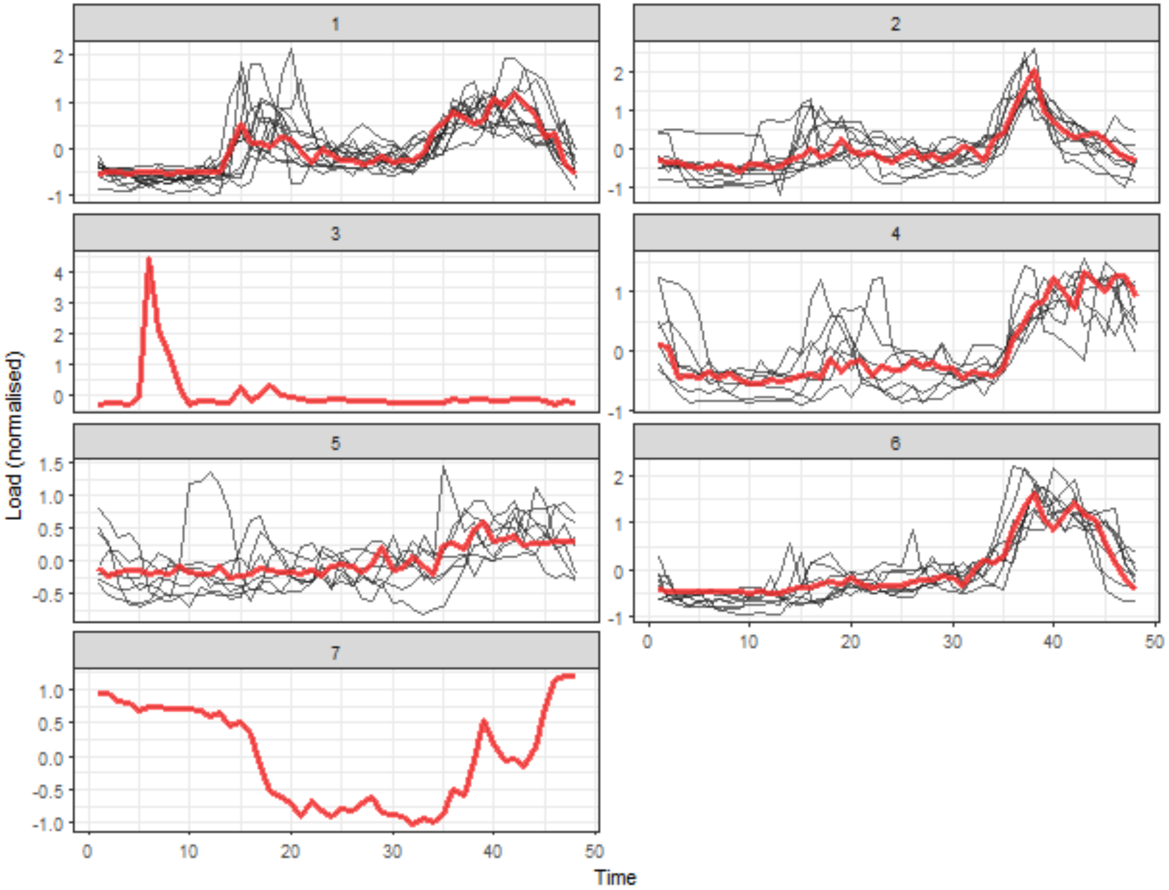
```
#checking stationarity
from statsmodels.tsa.vector_ar.vecm import coint_johansen
coint_johansen(data,-1,1).eig

#fit the model
from statsmodels.tsa.vector_ar.var_model import VAR
model = VAR(endog=train)
model_fit = model.fit()

# make prediction on validation
prediction = model_fit.forecast(model_fit.y, steps=len(valid))
```

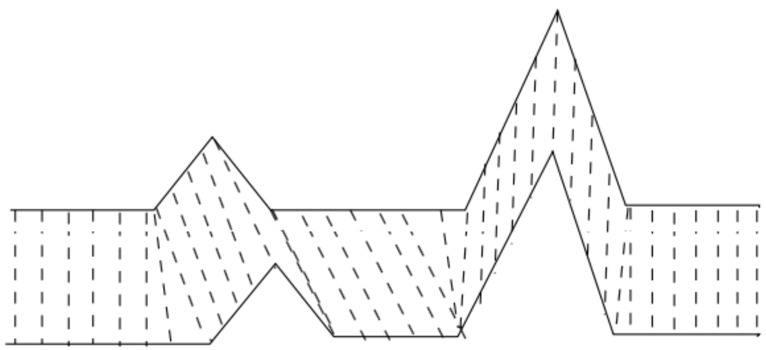
CLUSTERING TS

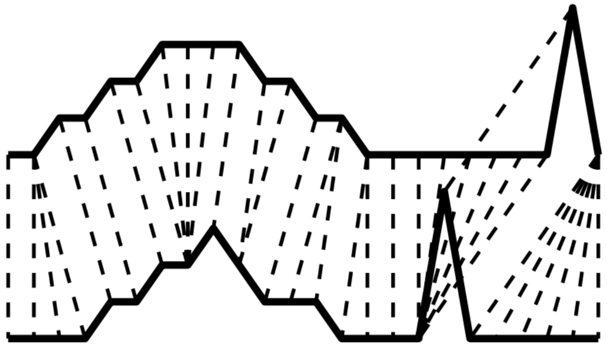
- It might be useful when there are many time series.
- For example, we can:
  - cluster the TS,
  - take the *signature* of each cluster, and
  - fit a model (i.e. ARIMA) on the signature.
- It is a standard clustering problem, but now the features represent points in time.
- (possible) problem: high dimensionality
  - Variants: compute TS metrics and cluster on those features
- Hierarchical clustering is frequently used.
- What distance metrics?
- What if TS are not aligned? (features + DTW)



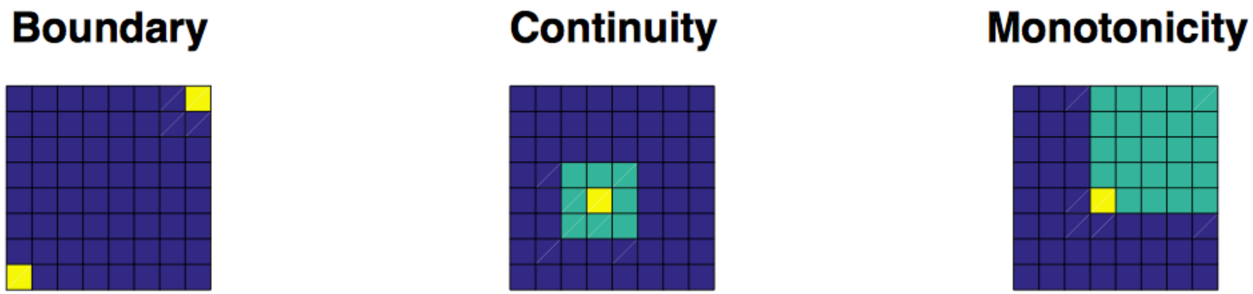
DTW – DYNAMIC TIME WARPING

Algorithm to measure the similiarty between two time series that might not be aligned.





- Start by constructing n × m matrix D, where
  - $D(i,j) = d(x_i, y_i)$,
  - $d(x_i,y_i)=|x_i - y_i|$ or $d(x_i,y_i)= \mathrm{sqrt}((x_i - y_i)^2)$.
- We want $d_{DTW}(x, y) = \min \Sigma_{i=1}^{k} D(w_i)$



Images: Wikipedia & University of Manchester

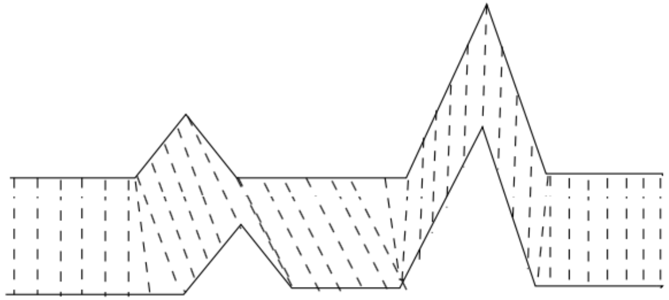
DTW – DYNAMIC TIME WARPING

Pros:

- Allows for TS of different lengths

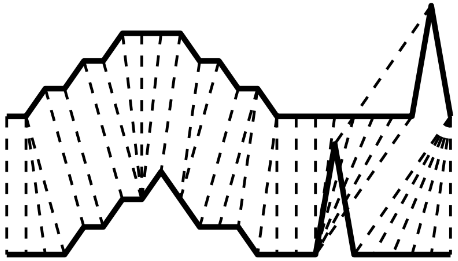
Cons:

- Does not satisfy triangle inequality
- Slow to compute (O(nm) in time and complexity)
  - There are ways to speed it, with restrictions.



Uses:

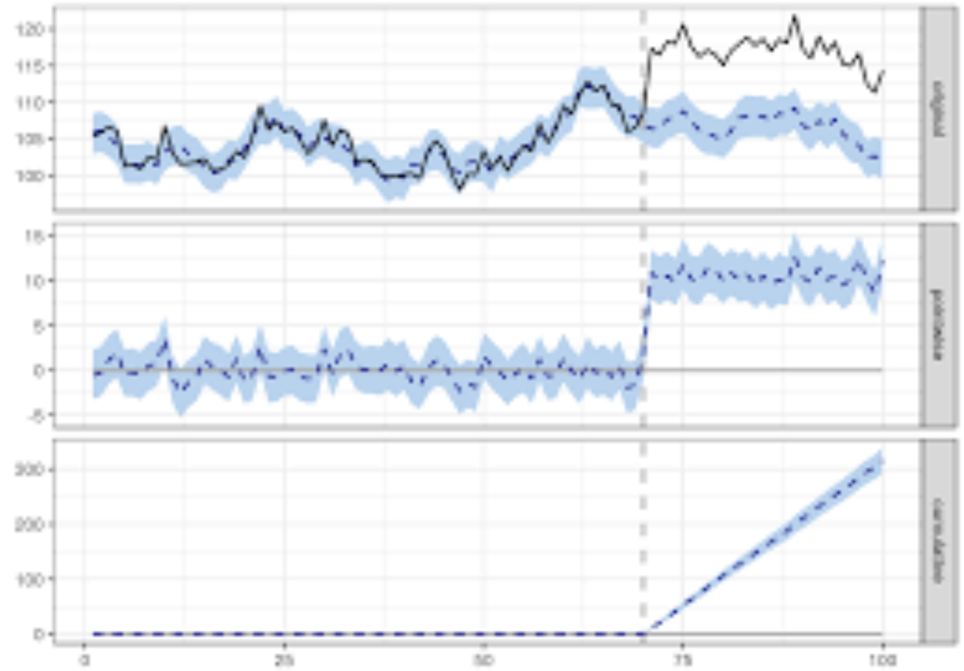
- Walking at different speeds
- Speech recognition (different speeds).
- Querying for similar TS



FORECASTING WITH CAUSAL IMPACT

Google's library for bayesian TS forecasting

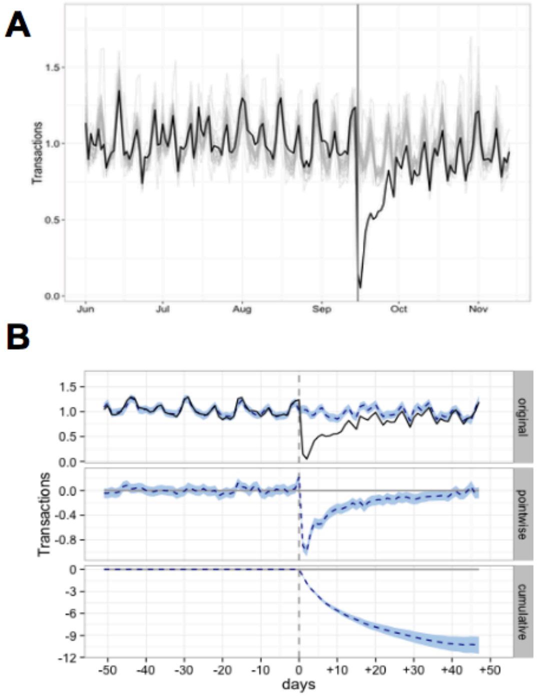
- Built for estimating the impact of a marketing campaign (or A/B test)
- We need many TS, with only one of them being impacted by the treatment.
- Uses a bayesian approach.
- Library (R).
- Paper.

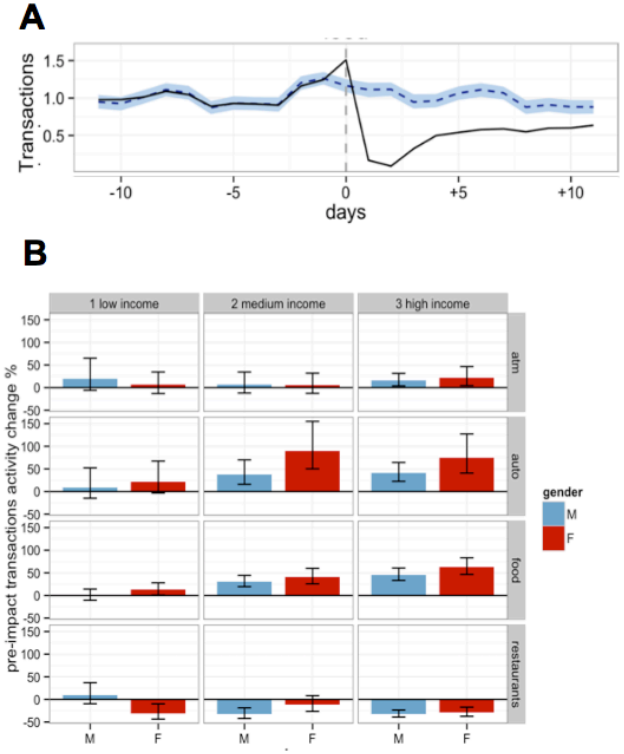


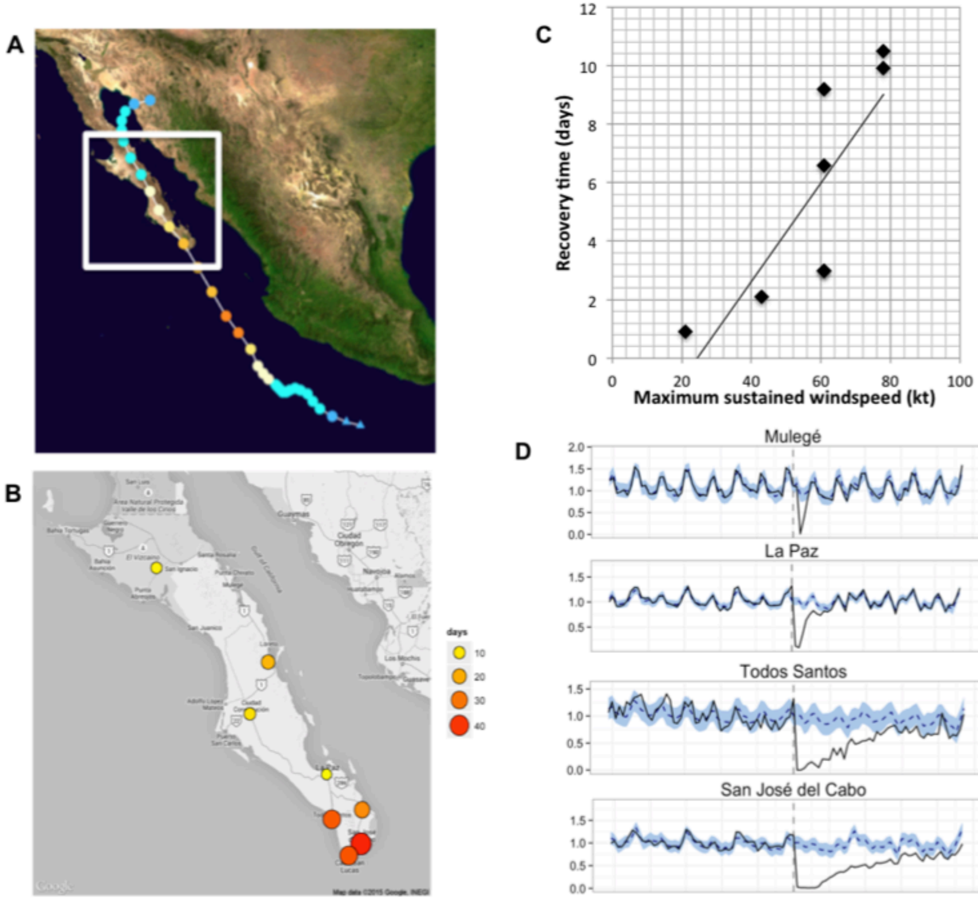
FORECASTING WITH CAUSAL IMPACT

Example: Measuring the economic impact of hurricane Odile in Mexico.

- Paper
- Web: https://www.bbvadata.com/odile/



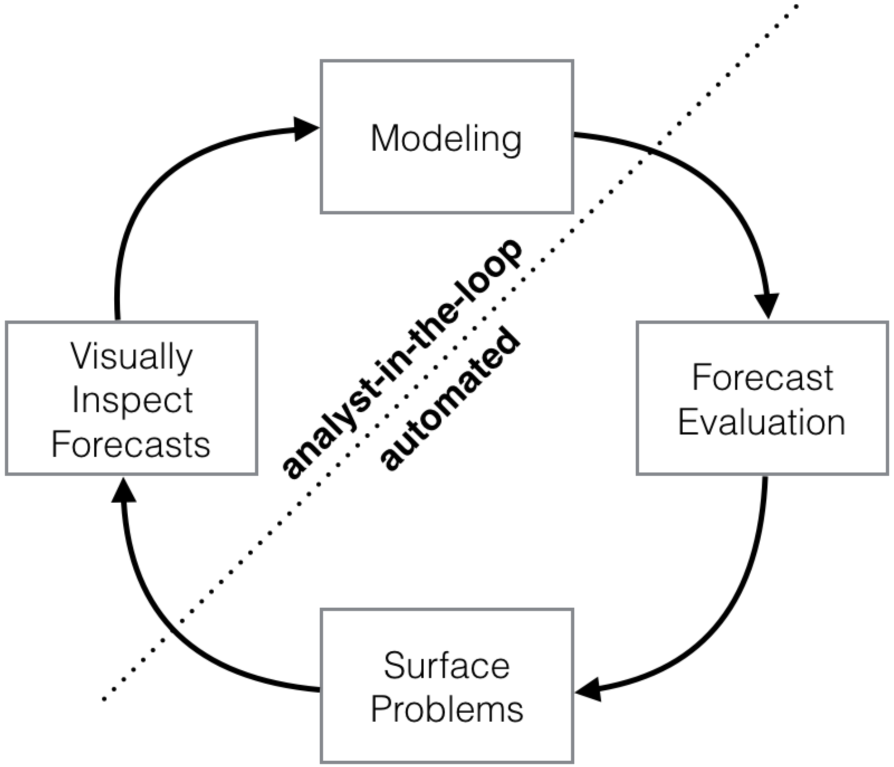




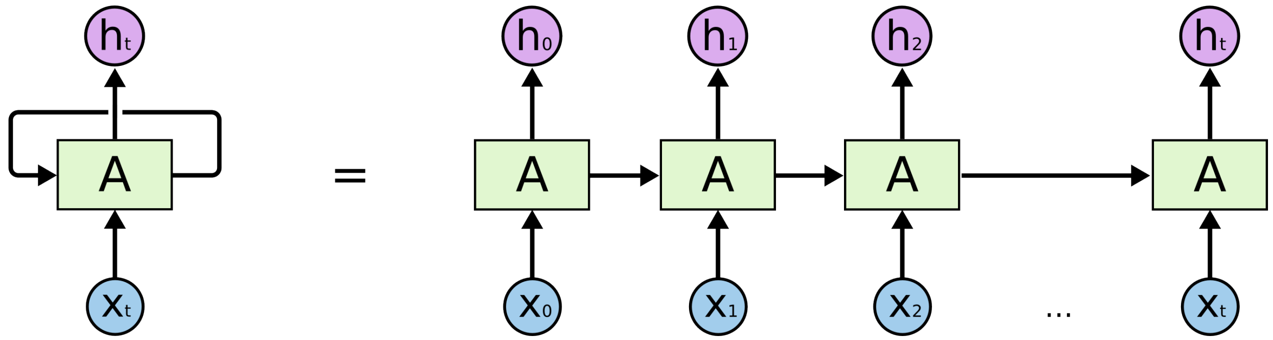
PROPHET

Facebook's library for TS forecasting

- Additive model with non linear trend
- "Forecasting at scale"
- Github: https://github.com/facebook/prophet
- R and Python library
- "Prophet is optimized for the business forecast tasks we have encountered at Facebook"



TS FORECASTING WITH DEEP LEARNING (DL)



- Supervised approach
- RNN, LSTM have a structure that naturally fits Time Series data: future values depend on past values.
- Create each sample with:
  - Output variable: y = x (t)
  - Input / feature variables: x(t-1), x(t-2),…
  - Useful when having multiple ts

TS IN REAL WORLD

We can have two types of problems:

A. One (or few) time series that we want to understand and forecast

B. A lot of time series that we want to forecast each of them.

Case A is what we've been seen so far, modelling with SARIMA and VARIMA models.

Case B is what you can expect on companies where you have multiple customers.

- Do you model an ARIMA model for each customer? 8M ts → auto.arima / holt.winters…
- Baseline (simple) models: last year, last month, mean and median of last N months
- Cluster the TS and use the ts belonging to each group as data for a Regression problem.
- …
- There is not "an approach to rule them all", it's on Data Scientist's hand to find the best modelling strategy.

THE BACKSHIFT OPERATOR

- Very common notation that we have avoid using it in the lecture.
- Lag operator L^i (B^i) means "Take the i-lag of time series" X_{t-i}
- ARMA(p, q)

$$X_t = c + \varepsilon_t + \sum_{i=1}^{p} \varphi_i X_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}$$

$$\left(1 - \sum_{i=1}^{p} \phi_i L^i\right) X_t = \left(1 + \sum_{i=1}^{q} \theta_i L^i\right) \varepsilon_t$$

ETS FOR FORECASTING

- Don't mix it with ets decomposition!!!
- Some readings with code
  - https://stats.stackexchange.com/questions/220299/stl-gives-seasonal-component-but-ets-and-auto-arima-choose-nonseasonal-mo
  - ETS function R
  - ETS and its friends in R
  - Arima vs ETS

TS IN R

Time Series

- https://cran.r-project.org/web/views/TimeSeries.html
- https://a-little-book-of-r-for-time-series.readthedocs.io/en/latest/
- Forecasting: Principles and Practices: https://otexts.com/fpp2/

Multivariate Time Series:

- https://little-book-of-r-for-multivariate-analysis.readthedocs.io/en/latest/
- VAR, SVAR and SVEC models: https://cran.r-project.org/web/packages/vars/vignettes/vars.pdf

Examples

- https://www.analyticsvidhya.com/blog/2015/12/complete-tutorial-time-series-modeling/
- http://r-statistics.co/Time-Series-Analysis-With-R.html

TS IN PYTHON

- Good practice: use the date component as index
- datetime indexing:
  - ts['2018-03-22'] // ts[datetime(2018, 03, 22)]
  - ts['2018']
  - ts['2018-03-20':]
- rolling.mean() rolling.std
- series.shift(), eries.diff()
- pd.plotting.autocorrelation_plot(diet);

- https://www.analyticsvidhya.com/blog/2016/02/time-series-forecasting-codes-python/
- https://towardsdatascience.com/an-end-to-end-project-on-time-series-analysis-and-forecasting-with-python-4835e6bf050b
- https://jakevdp.github.io/PythonDataScienceHandbook/03.11-working-with-time-series.html
- https://www.datacamp.com/community/tutorials/time-series-analysis-tutorial
- https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/
- https://machinelearningmastery.com/gentle-introduction-autocorrelation-partial-autocorrelation/
- Multivariate TS: https://www.analyticsvidhya.com/blog/2018/09/multivariate-time-series-guide-forecasting-modeling-python-codes/

SOME CONCLUSIONS

Name a few!

BIBLIOGRAPHY AND RESOURCES

- *Whittle, P. (1951). Hypothesis Testing in Time Series Analysis. Almquist and Wicksell. Whittle, P. (1963). Prediction and Regulation. English Universities Press.* *ISBN 0-8166-1147-5.*
- Paul S. P. Cowpertwait and Andrew V. Metcalfe. 2009. ”Introductory Time Series with R (1st ed.)”. Springer Publishing Company, Incorporated.
- R. B. Cleveland, W. S. Cleveland, J.E. McRae, and I. Terpenning (1990) STL: A Seasonal-Trend Decomposition Procedure Based on Loess. *Journal of Official Statistics*, **6**, 3–73.
- “Introduction to Time Series and Forecasting”, Brockwell and Davis
- Time Series Decomposition, Otexts
- https://anomaly.io/seasonal-trend-decomposition-in-r/
- ARIMA models for time series forecasting, Duke University. https://people.duke.edu/~rnau/411home.htm
- UC3M Time Series course: http://halweb.uc3m.es/esp/Personal/personas/amalonso/esp/tsa.htm
  - Time Series Clustering: http://halweb.uc3m.es/esp/Personal/personas/amalonso/esp/ASDM-C02-clustering.pdf
  - Seasonal Arima: http://halweb.uc3m.es/esp/Personal/personas/amalonso/esp/TSAtema6.pdf
- Clustering TS: comparing TS clustering algorithms in R: https://cran.r-project.org/web/packages/dtwclust/vignettes/dtwclust.pdf
- DL for TS forecasting: https://machinelearningmastery.com/deep-learning-for-time-series-forecasting/
- https://slideplayer.com/slide/4595589/
- Dynamic Time Warping: http://www.maths.manchester.ac.uk/~mbbx2se2/Docs/Dynamic_time_warping(Steven_Elsworth).pdf

Time series.

María Hernández, @maria_hr