Time Series specificities

Data analyst classroom trainings

Fabrice Jimenez
Data Scientist & Artificial Intelligence Engineer
Airbus Commercial Aircraft

Time Series: definition and scope

"Usual" dataset

Index	Average price (€)	Distance (km)	Type
categorical	continuous		categorical
restaurant_1	15	2.2	Italian
restaurant_2	10	5.1	Vietnamese
restaurant_3	25	0.4	Spanish
...	...	...	...

Time series dataset

Time	Stock price (€)	Temp (°C)	Website clicks
continuous		continuous	
00:01	15	22.5	4
00:02	14.2	24	6
00:03	15.6	26.5	20
...	...	...	...

Time series = subcategory of what we already know…

➜ Same principles and techniques will apply: scaling, PCA, clustering, statistics, predictive models….

Then what more do I have to learn?

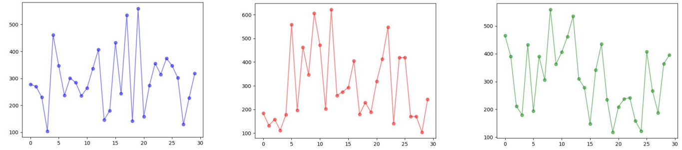
➜ Restricted scope = more possibilities because of:

Relationship between rows Not independent anymore…

All continuous variables

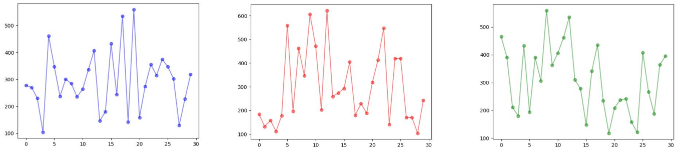
Time Series: for which task?

Classification



Country: *France* *Spain* *France*

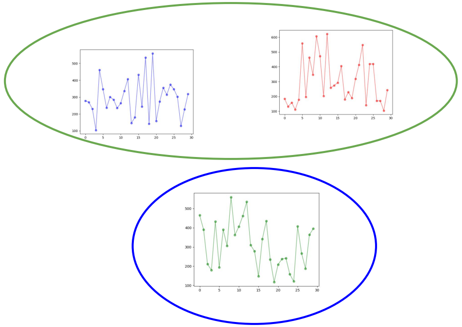
Regression



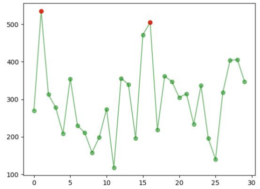
Satisfaction: *8.2* *7.5* *8.6*

Anomaly detection

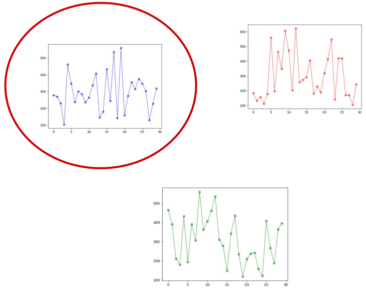
Clustering



Inside time series



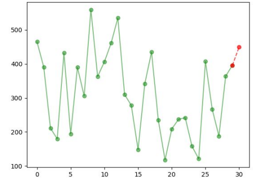
Between time series



Clearly define your question:
inside or between time series is possible for all tasks

For example:

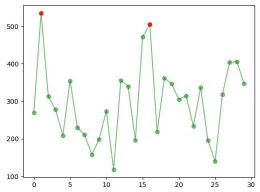
Next point prediction

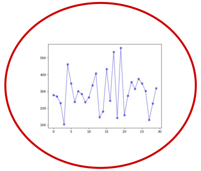


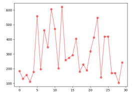
Time Series: for which task?

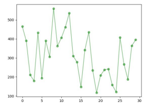
Clearly define your question:
inside or between time series is possible for all tasks

Inside time series | *Between time series*





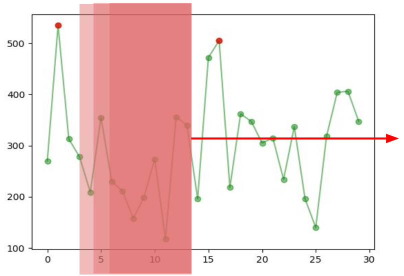




Not very specific to time series

Main focus of this class

Each row is a data point: usual techniques apply, but how to take advantage of rows relationships?



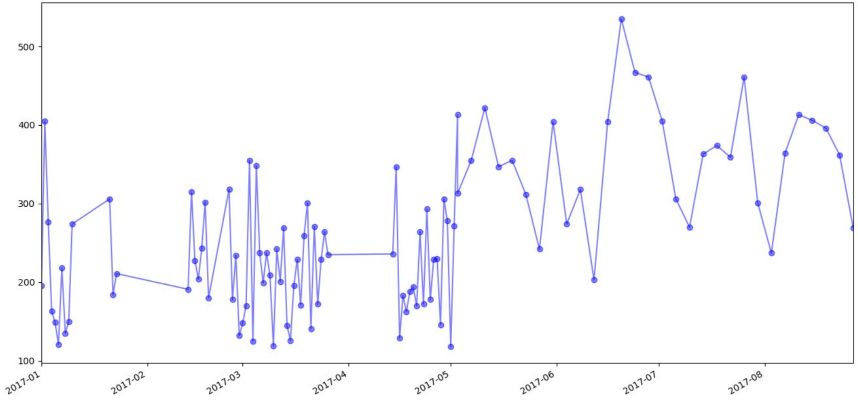
Rolling window principle

Analysis between windows

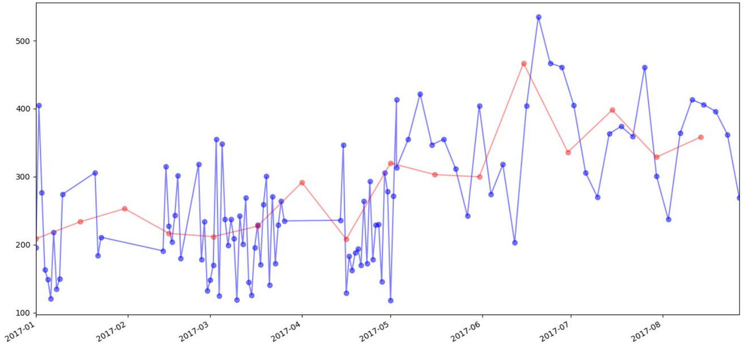
= group of successive points = smaller time series

Preprocessing: what to be careful with?

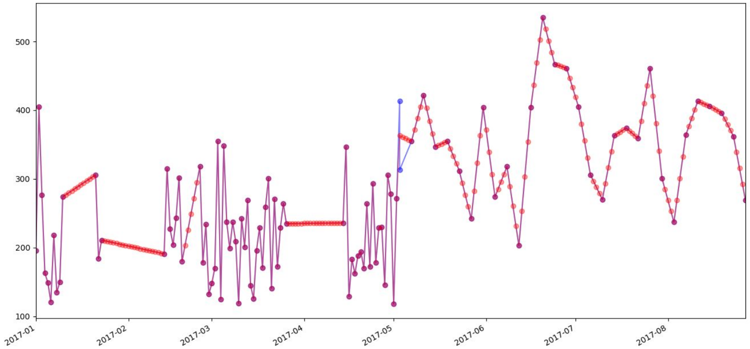
Sometimes, time index is not regular
Irregular sampling, missing values...



Subsampling



Oversampling



Problems?

- ➔ Time intervals between rows are not comparable
- ➔ Can be interesting to keep it, but be careful with the approach

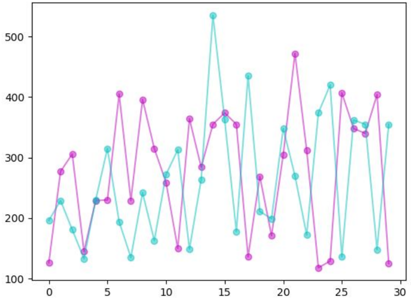
Solutions?

- ➔ Resampling: modify index to have regular timestamps
- ➔ Grouping and interpolation strategy to define: be careful!

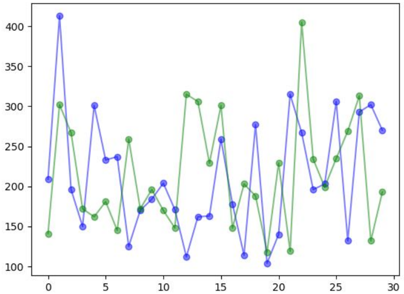
Transformation steps

Input: groups of points with n variables (n time series)

Before applying any technique:
How can we transform the data to take advantage of its structure?



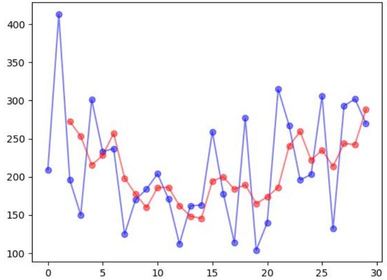
sample 1



sample 2

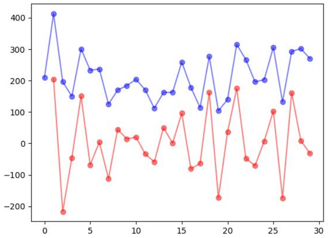
1D: for each time series at a time

I want to remove noise

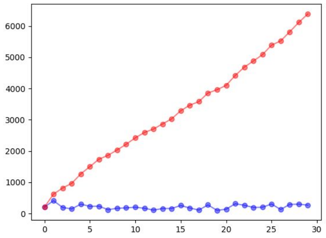


Rolling mean / median
Statistical filter (Hodrick-Prescott...)
Frequency filter (low pass, wavelet...)

I want to study the dynamics / evolution

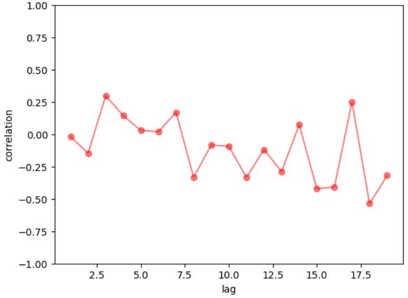


Derivative, differentiation
➜ makes the series stationary (independent of 1st point)



Cumulated sum
➜ aggregates history in value of current point

I want to study periodicity

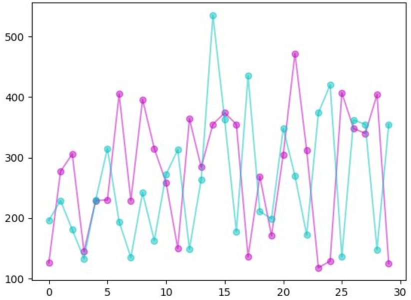


Autocorrelation: Pearson correlation of the series with itself shifted by a lag
➜ high correlation = identify period duration

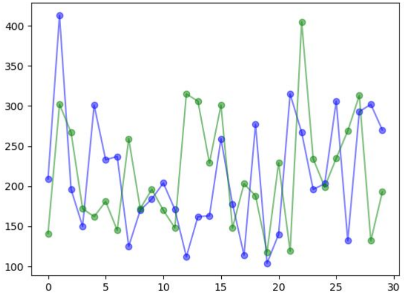
Transformation steps

Input: groups of points with n variables (n time series)

Before applying any technique:
How can we transform the data to take advantage of its structure?



sample 1

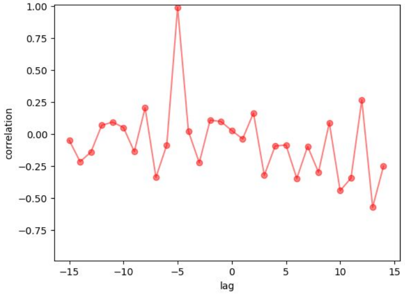


sample 2

Careful with scaling!

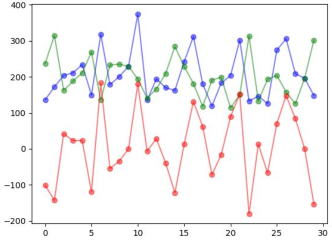
nD: for multiple time series at a time

I want to align 2 time series

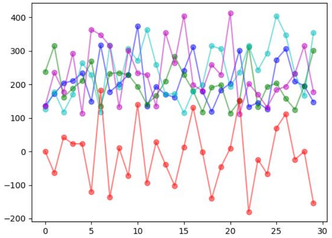


Cross-correlation
➔ correlation of 1 time series with another shifted by a lag
➔ high correlation = shift to apply

I want to study the relative behavior of 2 or more time series

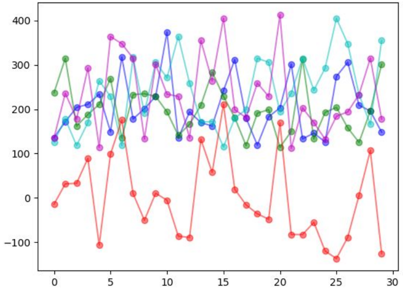


Pointwise difference of 2 time series
➔ relative behavior of the 2



Pointwise difference of 1 time series with mean / median of others
➔ relative behavior of 1 w.r.t the group

I want to aggregate all time series

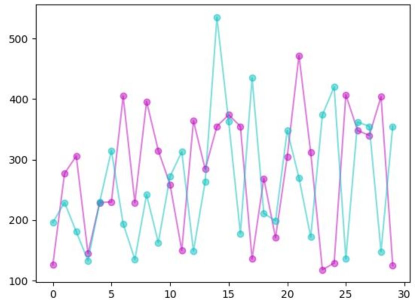


PCA: we can keep 1 or more component
➔ careful with scaling!
➔ careful with interpretation!

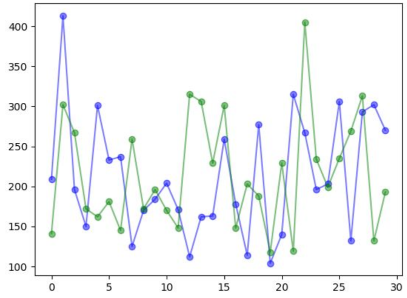
Feature Engineering

Input: groups of points with n variables (n time series)
➜ we have transformed our dataset according to steps above, we consider our time series are relevant for analysis and aggregate all information we need

Goal: summarize each time series in a set of p features in order to apply classical statistics and machine learning approaches



sample 1



sample 2

Features dataset

	ts1_F1	ts1_F2	...	ts1_Fp	ts2_F1	ts2_F2	...	ts2_Fp
Sample 1	48.5	-7.2	...	9.3	8.5	-4.3	...	0.8
Sample 2	32.9	8.6	...	-1.3	14.7	22.2	...	4.7

Why building features?
Why not directly applying known techniques on the relevant time series?

- ➜ Our task: for each sample we need to assign a value (float or category)
- ➜ Samples might not have the same length...
- ➜ The interesting pattern might happen at different time steps (delay...)

We summarize the behavior of time series in time for each sample

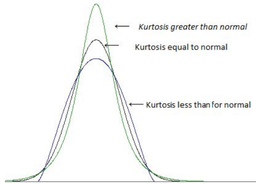
Feature Engineering: which features?

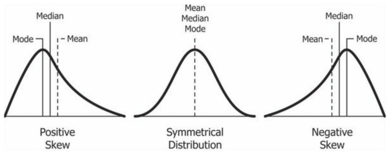
Of course, for each time series in the sample, we can compute classical statistics: mean, standard deviation, min, max, median, quartiles…
➜ But also more complex features according to what you are looking for:

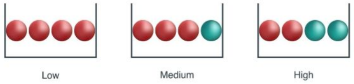
Energy-related

Sum, sum of absolute values
Energy $x_1^2 + x_2^2 + \ldots + x_n^2$
Quadratic mean $Q = \sqrt{\dfrac{x_1^2 + x_2^2 + \ldots + x_n^2}{n}}$

➜ Careful when summing values: not normalized by the length of the time series! Longer = higher…







Distribution-related

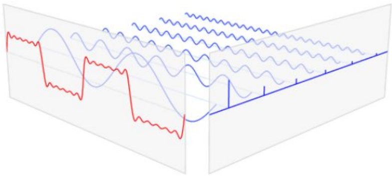
Kurtosis: how spread out is my distribution?

Skewness: how symmetrical is my distribution?

Entropy: how random is my distribution?

Dynamic-related

All features on derivative or absolute derivative (obtained at transformation step)
Number of mean crossings
Number of peaks (defined with (rolling) mean and std or median and mad)



Frequency-related

Fourier coefficients with FFT
Wavelet decomposition with approximation and detail coefficients (with choice of wavelet):
Example: noise energy = energy or quadratic mean of wavelet detail coefficients

Synthesis

Formulate the problem

What task? What scale?

Using raw points is enough

Necessity to exploit time series structure

Regular approach

Unsupervised methods

Supervised methods

Preprocessing

Transformation steps

Feature Engineering

Method validation

Interpretation of results

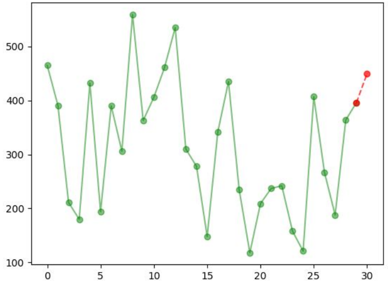
Reformulate the problem

Change some steps

Find a decision model

Example

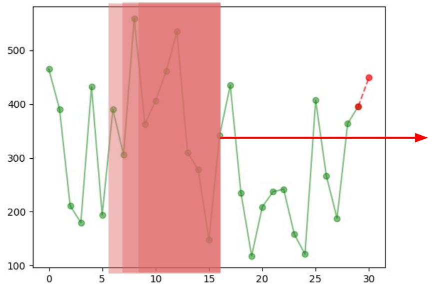
Predict the next point!



Formulate the problem

Regression problem:
Input = group of successive points
Output = value (of the following point)

As we don't have labeled samples, we can only use this time series: we use the rolling window strategy:



Each window = 1 sample
Following point = label output

Preprocessing

We use resampling with mean to have regular timestamps

Transformation steps

We are interested by global dynamics of time series: we smooth the signal with rolling mean, and compute the derivative to be independent of first point

Method validation

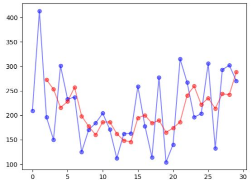
We use mean square error to measure accuracy of our predictions: not so good!
Let's try to change:
➔ smoothing: too strong?
➔ add or change features?
➔ linear regression maybe too simple?

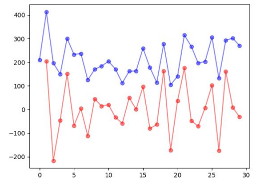
Regular approach

With features and output values for each window, we train a linear regression

Feature engineering

For each rolling window, we compute mean, median, std, min and max





Time Series specificities: application

It’s time to build your own time series analysis pipeline

Main interest = adopt workflow and mindset, make justified choices and implement them

Questions?