

Deep Learning para series temporales

Part I

Introduction



POLITÉCNICA

UNIVERSIDAD
POLITÉCNICA
DE MADRID



Máster
Deep Learning

Presentation

Introduction



- ▶ Víctor Rodríguez Fernández
victor.rfernandez@upm.es
- ▶ María Inmaculada Santamaría Valenzuela
mi.santamaria@upm.es

Develop a project

- ▶ **Option I:** Develop a project within a challenge.
- ▶ **Option II:** Develop your own project (previously asking teachers to confirm the proposal is adequate).

**Part I: Introduction**

Why am I here?

**Part II: Preprocessing
and analysis**

ETL + First observations

Part III: Forecasting

Predicting the future

Part IV: Supervised

- Supervised ML

**Part V: Non-
supervised**

Other Deep Learning tasks

What are time series?

Identify temporal data



What are time series?

- ▶ **Meals:** kilocalories, grams, ...
- ▶ **Airport:** planes departs, number of passengers waiting
- ▶ **Weather:** temperature readings
- ▶ **Finance:** stock prices
- ▶ **Retail:** sales data
- ▶ **Sports:** goals per minute

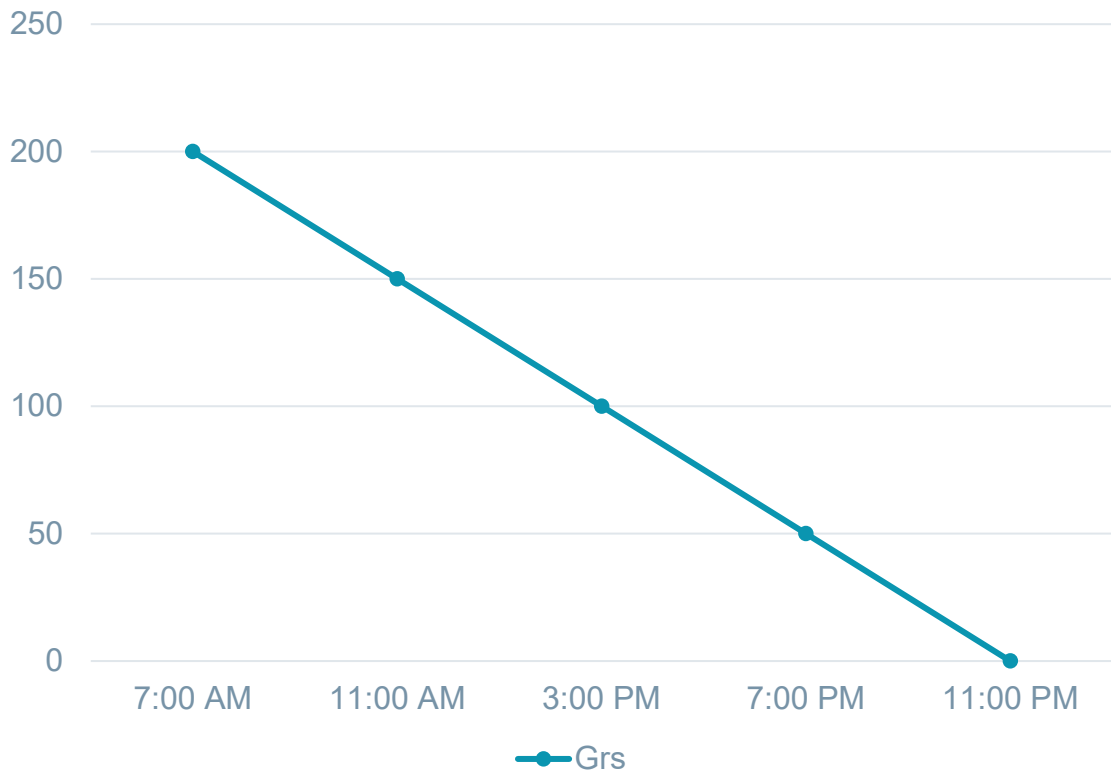
A first time series

Time	Available food (gr)
7:00 AM	200
11:00 AM	150
03:00 PM	100
07:00 PM	50
11:00 PM	0





A first time series



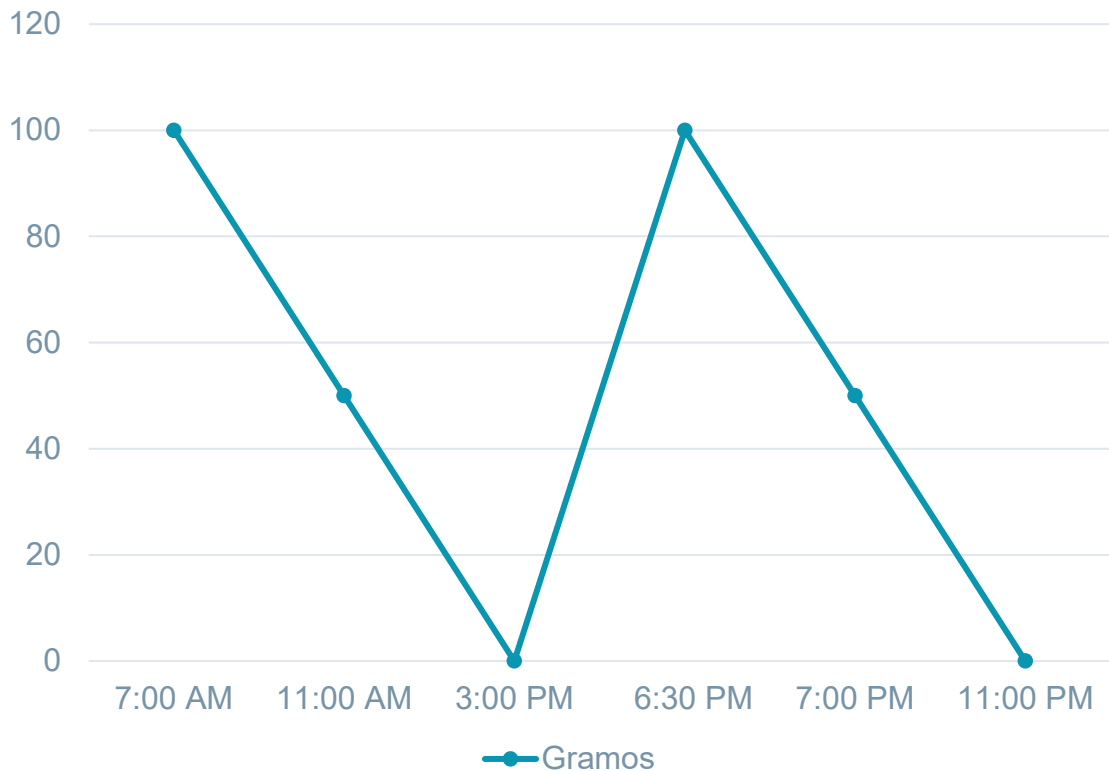
A first time series with refill

Time	Available food (grs)
7:00 AM	100
11:00 AM	50
03:00 PM	0
06:30 PM	100
07:00 PM	50
11:00 PM	0





A first time series with refill



*Stop: What characteristics make
data be a time series?*

?



A first time series

Ordered time sequence
 $t_0 < t_1 < t_2 < t_3 < t_4 < t_5$

Time	Available food (gr)
7:00 AM	100
11:00 AM	50
03:00 PM	0
06:30 PM	100
07:00 PM	50
11:00 PM	0

Time	Data
t_0	Data (real value) 1
t_1	Data 2
t_2	Data 3
t_3	Data 4
t_4	Data 5
t_5	Data 6



A first time series



Variate to analyze
Real values (1, 0.1, -0.27, ...)

Time	Available food (gr)
7:00 AM	100
11:00 AM	50
03:00 PM	0
06:30 PM	100
07:00 PM	50
11:00 PM	0

Time	Data
t0	Data 1
t1	Data 2
t2	Data 3
t3	Data 4
t4	Data 5
t5	Data 6



A first time series

Each data associated to one timestamp
as an index

Time	Available food (gr)
7:00 AM	← 100
11:00 AM	← 50
03:00 PM	← 0
06:30 PM	← 100
07:00 PM	← 50
11:00 PM	← 0

Index	Data
t0	← Data 1
t1	← Data 2
t2	← Data 3
t3	← Data 4
t4	← Data 5
t5	← Data 6



*What datatypes in python can we
use for storing a time series?*

?

Store time series in python

- ▶ List
- ▶ Set
- ▶ Tuple
- ▶ Dictionary
- ▶ String
- ▶ Array
- ▶ Make a class



Store time series in python

▶ List

~~▶ Set~~

- Does not allow repeated values

~~▶ Tuple~~

- I am not going to write 10K elements 1 by 1 as a tuple to define my series. And you cannot append or shorten it for preprocessing.

▶ Dictionary

~~▶ String~~

- If I am going to work with data, I don't want a string with the data: I want the data

▶ Array

▶ Make a class



Store time series in python

- ▶ List
- ~~▶ Set~~
- ~~▶ Tuple~~
- ▶ Dictionary
- ~~▶ String~~
- ▶ Array
- ▶ Make a class

Use a predefined widely used class





Store time series in python

	<u>np.ndarray</u>	<u>torch.tensor</u>	<u>pd.Series</u>	<u>pd.DataFrame</u>
Allow saving 1 time series	yes	Yes	Yes	Yes
Allow saving > 1 columns time series	yes	Yes	No	Yes
Allow indexing	Yes	Yes	Yes	Yes
Has index	No	No	Yes	Yes
Compute in GPU or CPU	CPU	CPU/GPU	CPU	CPU



NumPy



PyTorch



pandas

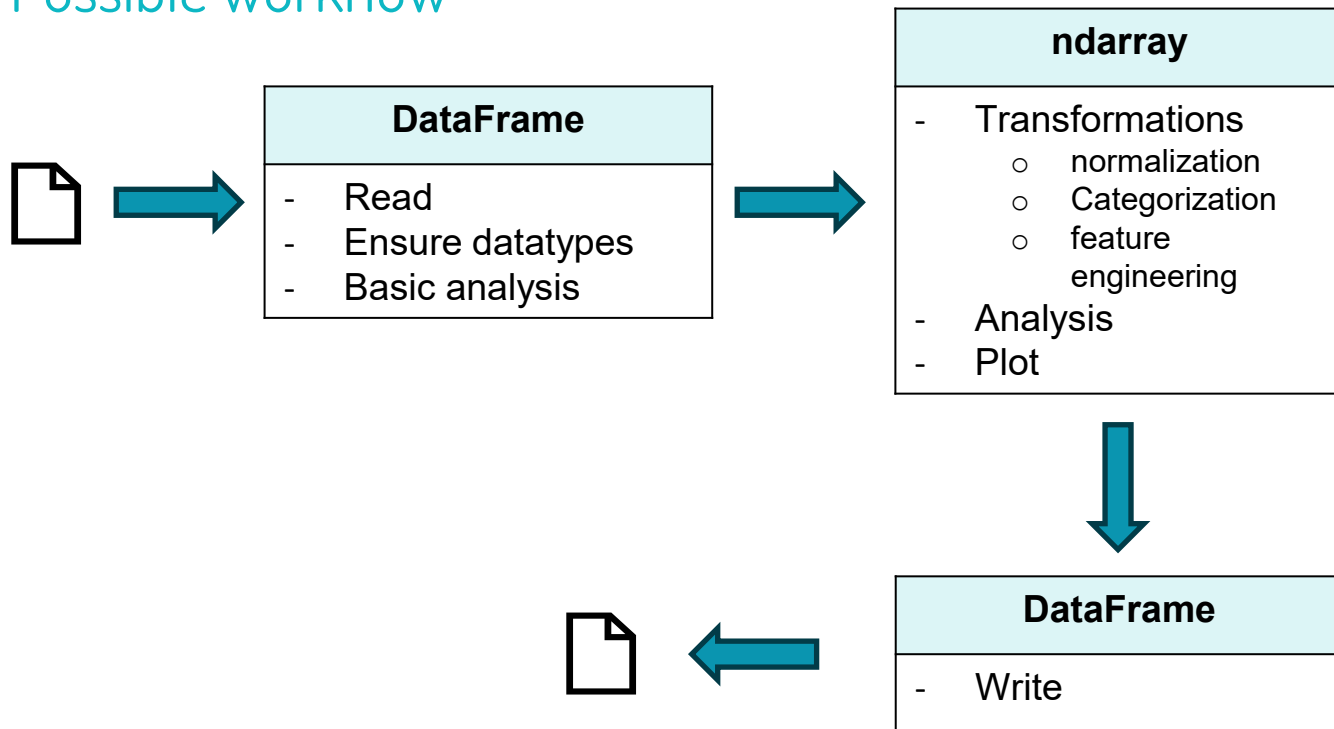


Store time series in python

	<u>np.ndarray</u>	<u>torch.tensor</u>	<u>pd.Series</u>	<u>pd.DataFrame</u>
Mathematic operations	Advanced	Advanced. Optimized for Machine Learning	Basic	Basic
Used for	<ul style="list-style-type: none"> Scientific / statistics calculations Generating time series 	<ul style="list-style-type: none"> Deep Learning. Training in GPU 	Tabular data, 1 column time series	<ul style="list-style-type: none"> Tabular data Multiple column time series Useful for reading and storing time series. Easy conversion with ndarray and tensor
Optimized for	Vectorized numerical operations	High performance computing with auto-differentiation	Reading / storing single series data	<ul style="list-style-type: none"> Reading(/storing time series Tabular data analysis
Compatibility	High with tensors and DataFrames	High with np.ndarray	Easy conversion to/from DataFrames	Easy conversion to/from np.ndarray

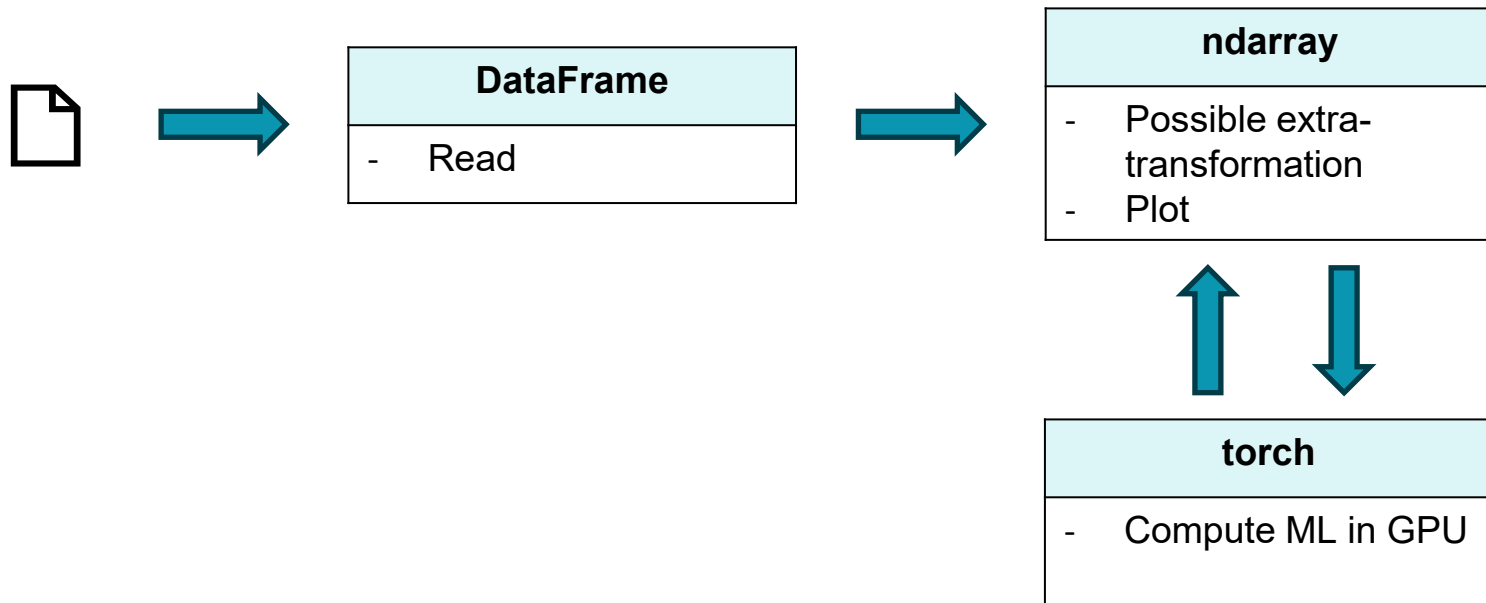
Store time series in python

Possible workflow



Store time series in python

Possible workflow





Store time series in python

```
raw_data_9 = pd.DataFrame({  
    "Time": ["7:00 AM", "11:00 AM", "3:00 PM", "7:00 PM", "11:00 PM"],  
    "Available food (gr)": raw_data  
})  
  
display(raw_data_9.head())
```

	Time	Available food (gr)
0	7:00 AM	200
1	11:00 AM	150
2	03:00 PM	100
3	07:00 PM	50
4	11:00 PM	0

- ▶ Index: 0, 1, 2...
- ▶ Column names: "Time", "Available food (gr)"
- ▶ Column values:
 - ▶ "7:00 AM", "12:00 AM", ...
 - ▶ 200, 150, ...



Store time series in python

```
# Index
<pd.DataFrame>.index

# Columns names
<pd.DataFrame>.columns

# Values by column name
<pd.core.series.Series>.values
<pd.DataFrame>[<column name (str)>].values

# Values by column position
<pd.DataFrame>.iloc[:,<column number>].values

# Values by row position
<pd.DataFrame>.iloc[<row number>, :].values
```



Store time series in python

- ▶ We focus on float data (the most used in time series analysis)

In the next lesson we will see

- ▶ How to preprocess data (ETL)
 - ▶ How to make the time be the index of the time series

*What other time series can we
check?*

?



Other time series examples

INE

Instituto Nacional de Estadística

Castellano

[Electoral Census](#)[Electronic Office](#)[Share](#)

Time series base



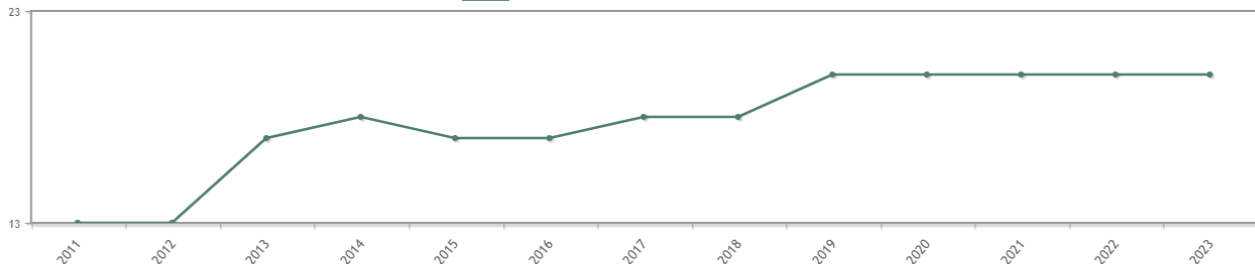
(0) series



Back

INEbase

Type of graph: | Number of latest data to be shown: |



List of series with graphs

☐ All series

☒ APEL243279 - Residents. Absolute value. Residence province. First surname. National Total. AIRE.

<https://ine.es/consul/serie.do?s=136-243279&L=1>



 **MADRID** | Portal de datos abiertos del Ayuntamiento de Madrid

¿Qué estás buscando? 

Tu ciudad más cerca

Gracias a nuestra plataforma de datos abiertos podrás encontrar todos los datos de Madrid que necesitas para tu proyecto

[En portada](#) [Acerca de Datos Abiertos](#) **[Catálogo de datos](#)** [Colabora](#) [Visualiza](#) [Solicitud de reutilización](#)

 [Catálogo de datos](#) > [Conjuntos de datos](#)

Calidad del aire. Datos horarios desde 2001

 [Escuchar](#) 

El Sistema Integral de la Calidad del Aire del Ayuntamiento de Madrid permite conocer en cada momento los niveles de contaminación atmosférica en el municipio. En este conjunto de datos puede obtener la información recogida por las estaciones de control de calidad del aire, con los datos horarios por anualidades desde 2001. (En el año en curso la información se actualizará mensualmente).

Los datos horarios de las magnitudes corresponden a la media aritmética de los valores diezminutales que se registran cada hora.

En este portal también están disponibles otros conjuntos de datos relacionados con la calidad del aire:


- [Calidad del aire: Datos en tiempo real](#)
- [Calidad del aire: Datos diarios desde 2001](#)
- [Calidad del aire: Estaciones de control](#)

Además, puedes encontrar más información sobre estos datos en el Portal de transparencia: [Aire](#)


Calidad del aire. Datos horarios desde 2001 - Portal de datos abiertos del Ayuntamiento de Madrid



Other time series examples








[Communities](#) [My dashboard](#)

 Monash Time Series Forecasting Repository

Published June 11, 2020 | Version 3

[Dataset](#) [Open](#)

Solar Dataset (10 Minutes Observations)

Godaheva, Rakshitha¹ ; Bergmeir, Christoph² ; Webb, Geoff³ ; Hyndman, Rob³ ; Montero-Manso, Pablo⁴ 

Show affiliations

The solar dataset contains approximately 6000 simulated time series representing 5-minute solar power and hourly day-ahead forecasts of photovoltaic (PV) power plants in United States in 2006.

The uploaded dataset contains the aggregated version of a subset of the original dataset used by Lai et al. (2017). It contains 137 time series representing the solar power production recorded per every 10 minutes in Alabama state in 2006.

[Solar Dataset \(10 Minutes Observations\) \(zenodo.org\)](https://zenodo.org/record/3741211/files/solar_dataset.zip)



Other time series examples

- ▶ Residents / year
<https://ine.es/consul/serie.do?s=136-243279&L=1>
- ▶ Air quality data values / hour
[Calidad del aire. Datos horarios desde 2001 - Portal de datos abiertos del Ayuntamiento de Madrid](#)
- ▶ Solar power / 10 minutes
[Solar Dataset \(10 Minutes Observations\) \(zenodo.org\)](#)
- ▶ ECG in High Intensity Exercise Dataset
[ECG in High Intensity Exercise Dataset \(zenodo.org\)](#)
- ▶ Crop yield prediction
[Crop Yield Prediction Dataset \(kaggle.com\)](#)

*Applications in your work / real
world?*

?



Other time series examples

- ▶ Solved incidences/tickets per hour
- ▶ Trading
- ▶ Stock of products in a shop
- ▶ Available memory / execution time



Real-world uses

- ▶ Electronic health record ([pulsus paradoxus](#))
- ▶ Human activity recognition ([HAR using spartphone](#))
- ▶ Cybersecurity ([intrusion detection -> attack prediction](#))
- ▶ Aerospace engineering ([methods & applications for flight](#))
- ▶ Weather forecasting ([kaggle's long-term dataset](#))

How/where can I get data from?

?



Real-world uses

- ▶ Own data:
 - ▶ Machines
 - ▶ person making actions
 - ▶ smartwatch
- ▶ Synthetic datasets
 - ▶ Toy (Stumpy): [mSTAMP \(MSTUMP\) Toy Data · TD Ameritrade/stumpy Wiki · GitHub](#)
- ▶ Databases
 - ▶ Kaggle, Google Dataset Search, Ine, private datasets

A bit of theory

Basic definitions

Number of variates

Univariate (1 feature)

Time	Available food (grs)
7:00 AM	100
11:00 AM	50
03:00 PM	0
06:30 PM	100
07:00 PM	50
10:00 PM	0

Multivariate (> 1 feature)

Time	Refilled food (grs)	Eaten food (grs)
7:00 AM	100	0
11:00 AM	0	50
03:00 PM	0	50
06:30 PM	100	0
07:00 PM	0	50
10:00 PM	0	50



Number of variates

Toy dataset

Index	T1	T2	T3
0	0.565117	0.637180	0.741822
1	0.493513	0.629415	0.739731
2	0.469350	0.539220	0.718757
3	0.444100	0.577670	0.730169
4	0.373008	0.570180	0.752406

Tourist number

Date (Unicode format)	Tourist number (total tourist number visiting the island)
33604	8414
33635	9767
33664	13805
33695	12987
33725	32190



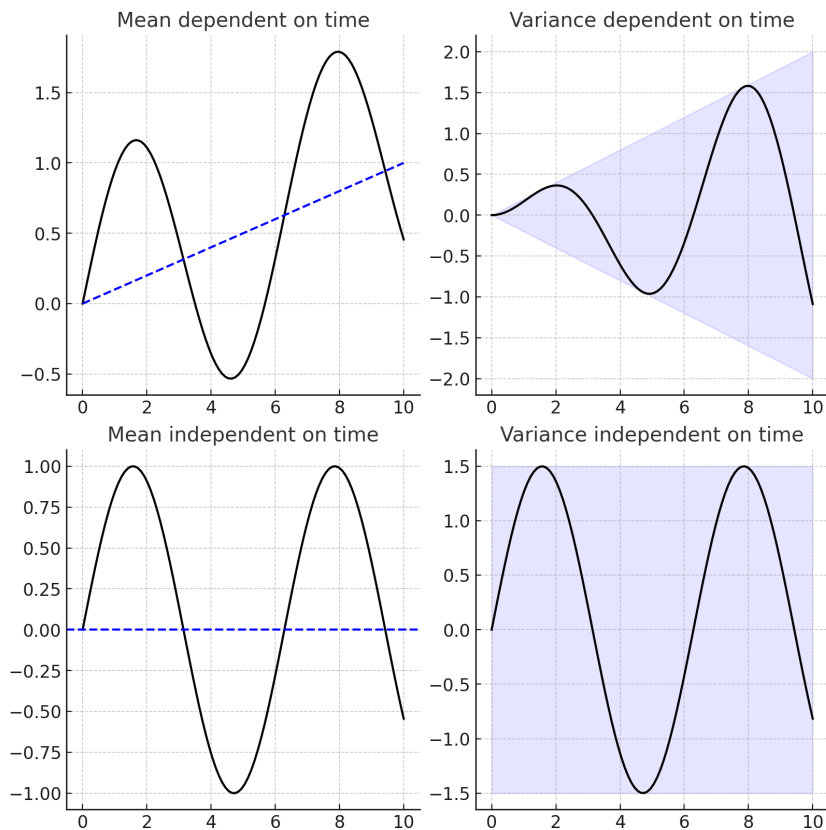
40

Stationarity

A stationary time series *mean* and *variance* doesn't change.



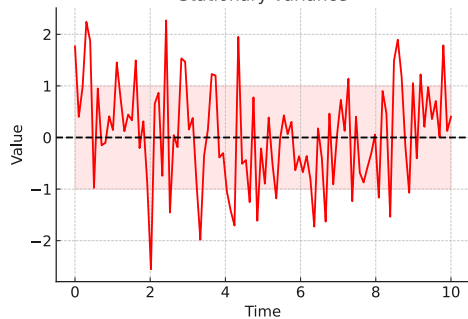
Master
Deep Learning



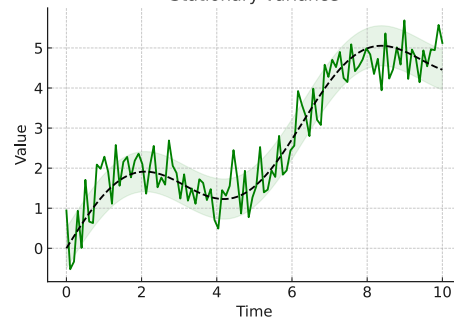
Stationarity



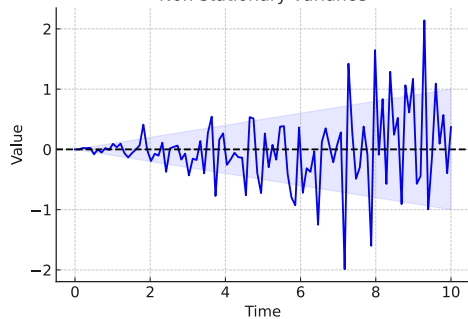
Stationary mean
Stationary variance



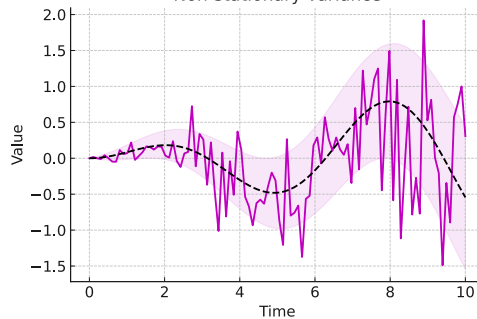
Non-stationary mean
Stationary variance



Stationary mean
Non-stationary variance



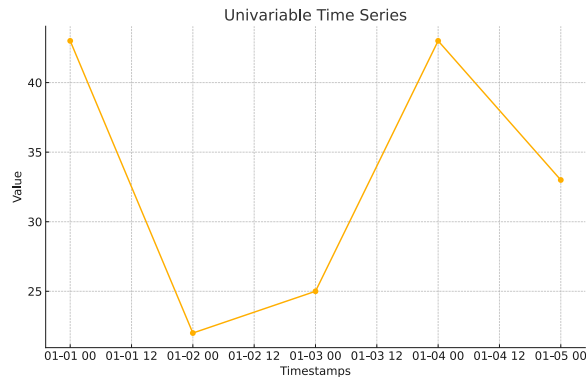
Non-stationary mean
Non-stationary variance





Number of
timestamps

Timestamps	Value
2023-01-01	43
2023-01-02	22
2023-01-03	25
2023-01-04	43
2023-01-05	33

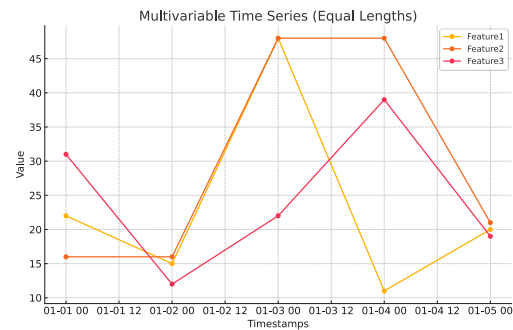


length ?



And...
multivariate?

Timestamps	V1	V2	V3
2023-01-01	16	25	41
2023-01-02	46	47	42
2023-01-03	36	47	26
2023-01-04	40		42
2023-01-05	15		20

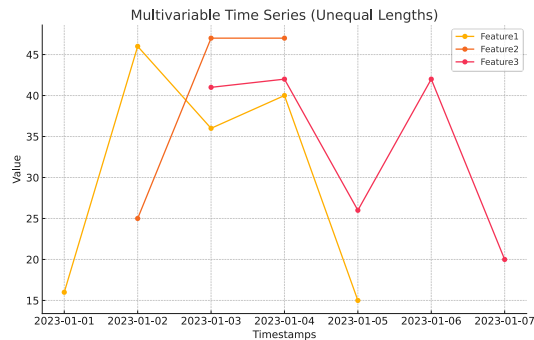


length ?



And...
multivariate?

Timestamps	V1	V2	V3
2023-01-01	22	16	31
2023-01-02	15	16	12
2023-01-03	48	48	22
2023-01-04	11	48	39
2023-01-05	20	21	19



length ?

Spacing



	Time	Available food (gr)
4h	7:00 AM	200
4h	11:00 AM	150
4h	03:00 PM	100
4h	07:00 PM	50
4h	11:00 PM	0

Evenly spaced

	Time	Available food (grs)
4h	7:00 AM	100
4h	11:00 AM	50
3h 30 min	03:00 PM	0
30 min	06:30 PM	100
30 min	07:00 PM	50
4h	11:00 PM	0

Non evenly spaced



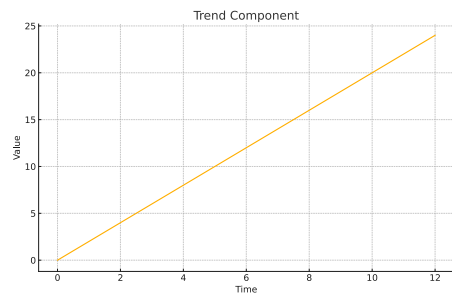
Classical (additive) decomposition



$$x(t) = T(t) + S(t) + I(t)$$

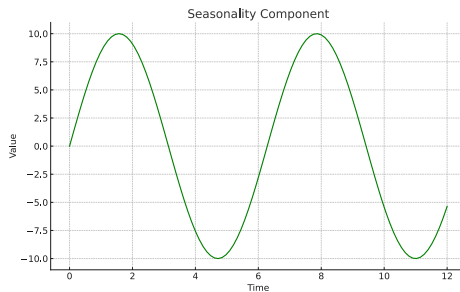
- ▶ $x(t)$: time series “x” data at index position “t”
- ▶ T: trend component.
- ▶ S: seasonality component
- ▶ I: Irregular/noise/random component

Classical (additive) decomposition



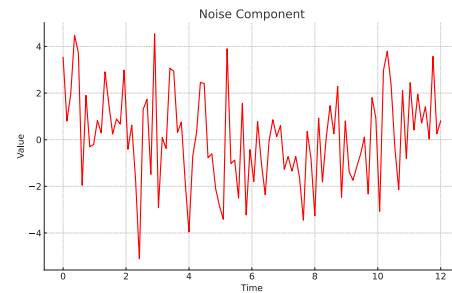
T

+



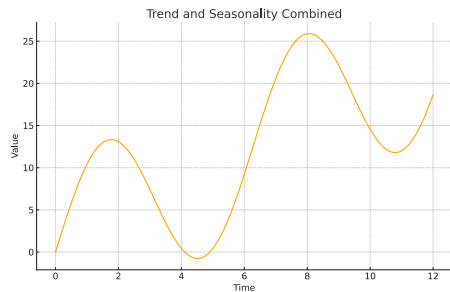
S

+

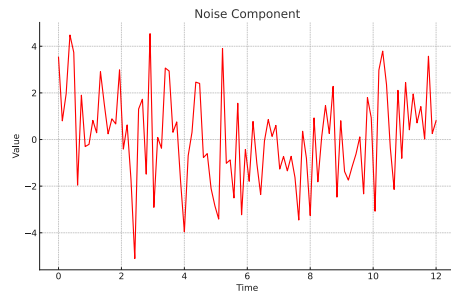


I

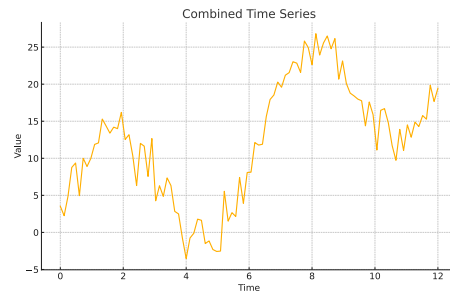
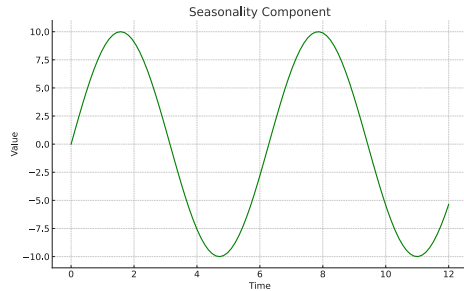
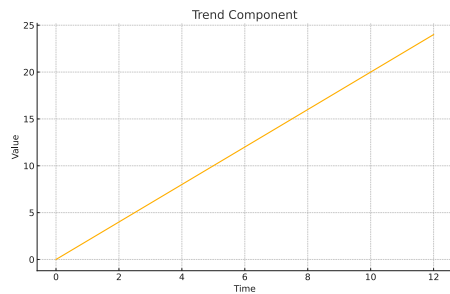
Classical (additive) decomposition



+



=

 $T+S$ I $X(t)$ 

By hand

?

By hand: check learned

- ▶ Check cat dataset length (both datasets)
- ▶ Plot two univariate time series: one stationary, one non-stationary
- ▶ Mark in those series which would be the length
- ▶ Plot a multivariate time series with same lengths
- ▶ Plot a multivariate time series with different lengths

Google Collab
01_Introduction.ipynb

?

Summary

Summary of the lesson



What this we just learned?

- ▶ What is a time series
- ▶ Where can we get time series from
- ▶ Types of time series
- ▶ The problem of evenly/non-evenly distribution
- ▶ Lenth
- ▶ Classical (additive decomposition): trend, seasonality, irregular/noise

Google Collab
01_Introduction_exercises.ipynb

?



What is the next step?

- ▶ Given a time series...
 - ▶ How to Extract + Transform + Load the data
 - ▶ Basic EDA (Exploratory Data Analysis)
 - ▶ Preprocessing techniques (upgrading ETL)

To be continued...

Questions? mi.santamaria@upm.es

Deep Learning para series temporales

Part I

Introduction



POLITÉCNICA

UNIVERSIDAD
POLITÉCNICA
DE MADRID



Máster
Deep Learning