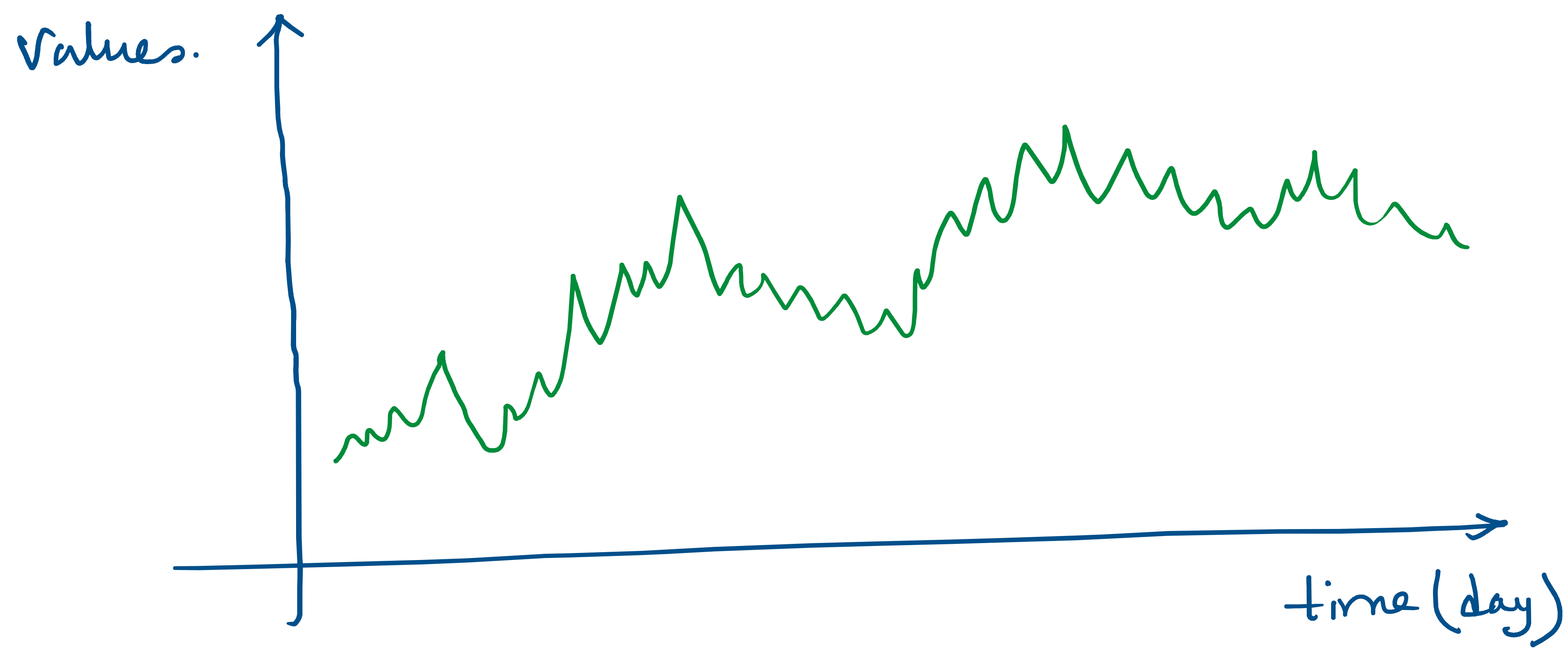
Time Series Resampling :-



During resampling we **compress** the time series

Ex: From daily data we can compress to weekly, monthly, quarterly, annual data.

freq:- day → freq:- week

day 1	y_1
day 2	y_2
day 3	y_3
⋮	
day 7	y_7

$\}$ week 1 → aggregate $(y_1, y_2, \ldots, y_7)$

mean, min, max, std, (sum)

df → pandas dataframe containing time series data.

df.resample(rule).agg()

- mean()
- min()
- max()
- std()
- sum()

Aggregation functions.

The rule parameter describes the frequency with which we can apply aggregation function.
It is passed using an "offset alias".

The offset alias is documented in the link:
https://pandas.pydata.org/docs/user_guide/timeseries.html#offset-aliases

Please go through it to understand different kinds of rules to resample the data.

Time Shifting :-

Dates	Values
day 1	y_1
day 2	y_2
day 3	y_3
⋮	⋮
day n	y_n

Time shifting

- Shift-down
- Shift-up.

Shift down:- If someone shifts down the data by '1' day then —

dates	values
day 1	nan (no values)
day 2	y_1
day 3	y_2
day 4	y_3
⋮	⋮
day n	y_{n-1}

If I shift down by 'k' days

dates	values
day 1	nan
day 2	nan
⋮	⋮
day k	nan
day k+1	y_1
day k+2	y_2
⋮	⋮
day n	y_{n-k}

no values for first 'k' days.

Shifting up by 1 day

dates	values
day 1	y_2
day 2	y_3
day 3	y_4
⋮	⋮
day n-1	y_n
day n	NaN.

Shifting up by k days

dates	values
day 1	y_{k+1}
day 2	y_{k+2}
⋮	⋮
day n-k	y_n
day n-k+1	NaN
⋮	⋮
day n	NaN

Time Rolling :-

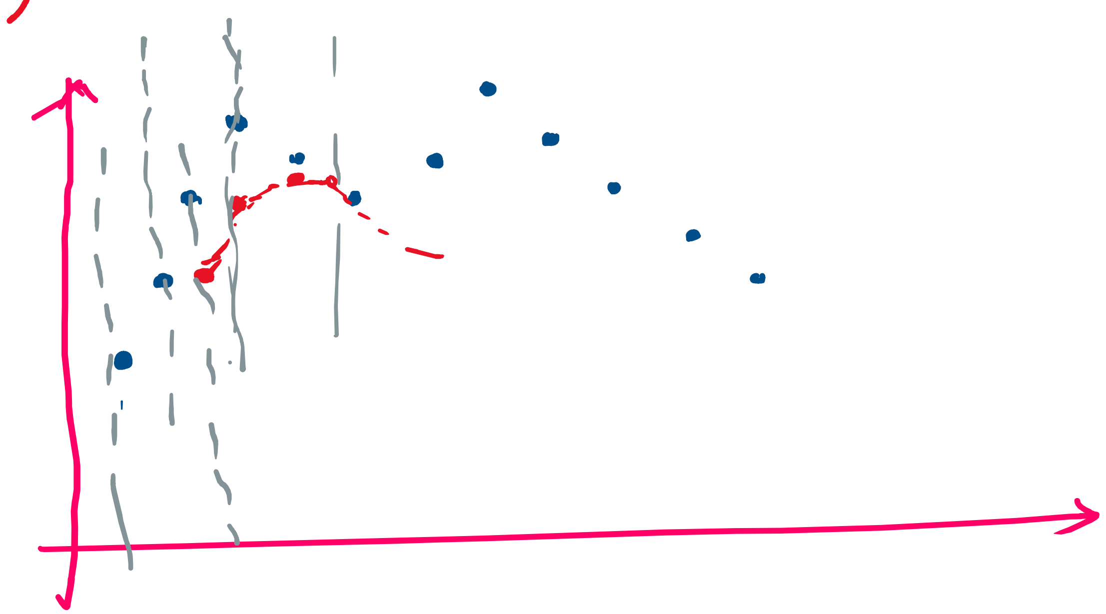
time	values
t_1	y_1
t_2	y_2
t_3	y_3
⋮	⋮
t_n	y_n

Rolling window = period for which the agg function will be applied.

aggregate function = what aggregation method we want to calculate.

Rolling window = 3, agg function = mean() } simple moving average with window = 3

time	values (rolling)
t_1	NaN
t_2	NaN
t_3	$\frac{y_1+y_2+y_3}{3}$
t_4	$\frac{y_2+y_3+y_4}{3}$
t_5	$\frac{y_3+y_4+y_5}{3}$
⋮	
t_n	$\frac{y_{n-2}+y_{n-1}+y_n}{3}$



For time rolling with rolling window 'k'

time	values
t_1	NaN
t_2	NaN
t_3	⋮
⋮	NaN.
t_{k-1}	
t_k	$agg(t_1, t_2, \ldots, t_k)$
t_{k+1}	$agg(t_2, t_3, \ldots, t_{k+1})$
⋮	⋮
t_n	$agg(t_{n-k+1}, t_{n-k+2}, \ldots, t_n)$

$agg(--) \rightarrow$ aggregate functions.

mean()

min()

max()

std()

sum().

Time Expanding:—

Only aggregate function is required.

time	values	window	o/p value
t_1	y_1	1	y_1
t_2	y_2	2	$agg(y_1, y_2)$
t_3	y_3	3	$agg(y_1, y_2, y_3)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
t_k	y_k	k	$agg(y_1, y_2, y_3, \ldots, y_k)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
t_n	y_n	n	$agg(y_1, y_2, y_3, \ldots, y_n)$

$agg(\ ) \rightarrow$ aggregate function.

Main components of a time series data :—

time	values
t_1	y_1
t_2	y_2
t_3	y_3
⋮	⋮
t_n	y_n

$y_t \rightarrow$ the value of the time series at 't'th instance.

Additive model

$$\boxed{y_t = \tau_t + S_t + C_t + \epsilon_t}$$

Multiplicative model

$$y_t = \tau_t \times S_t \times C_t + \epsilon_t$$

We will not use.

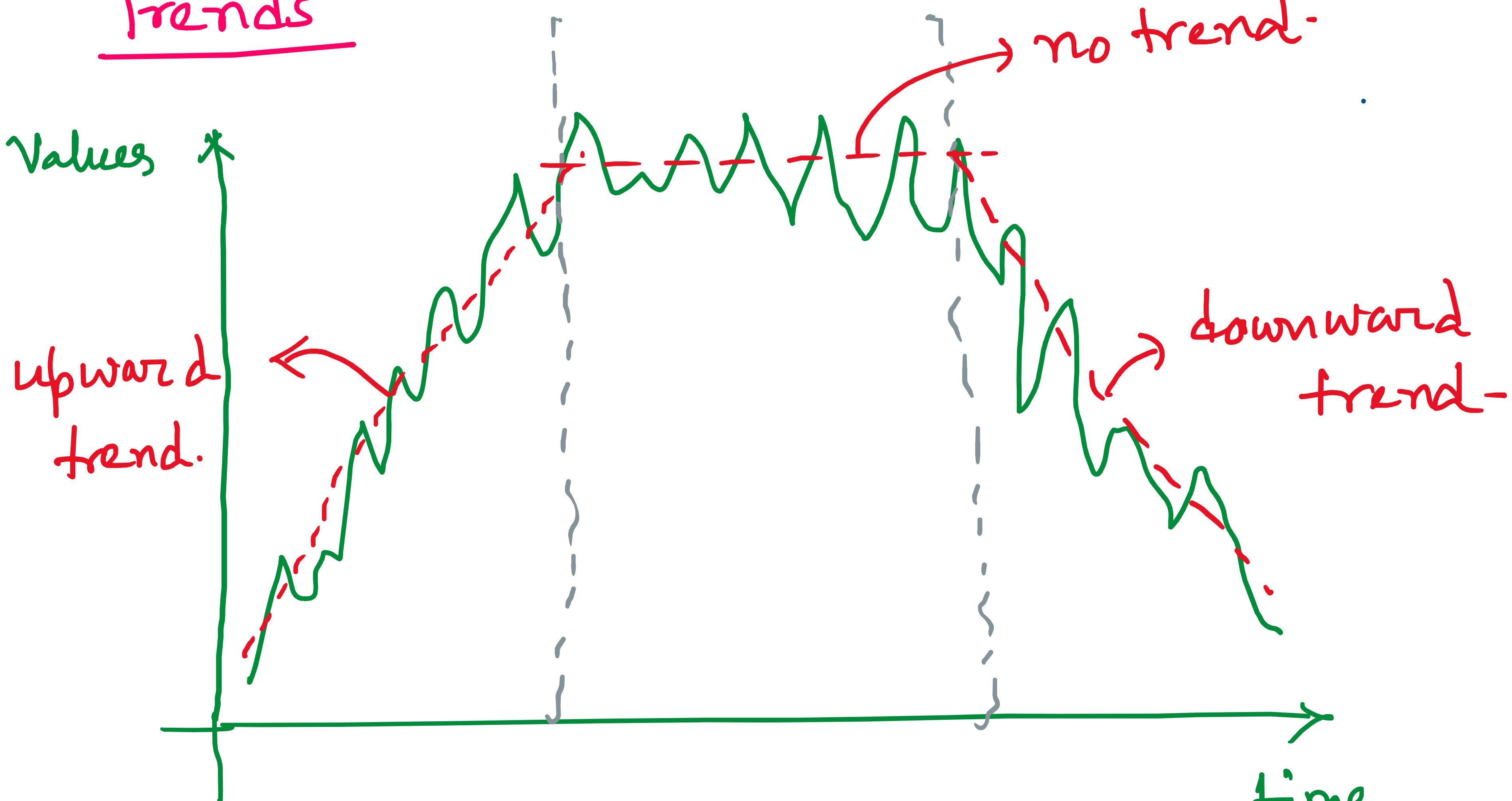
$\tau_t \rightarrow$ trend component

$S_t \rightarrow$ Seasonality component (specific period, usually shorter)

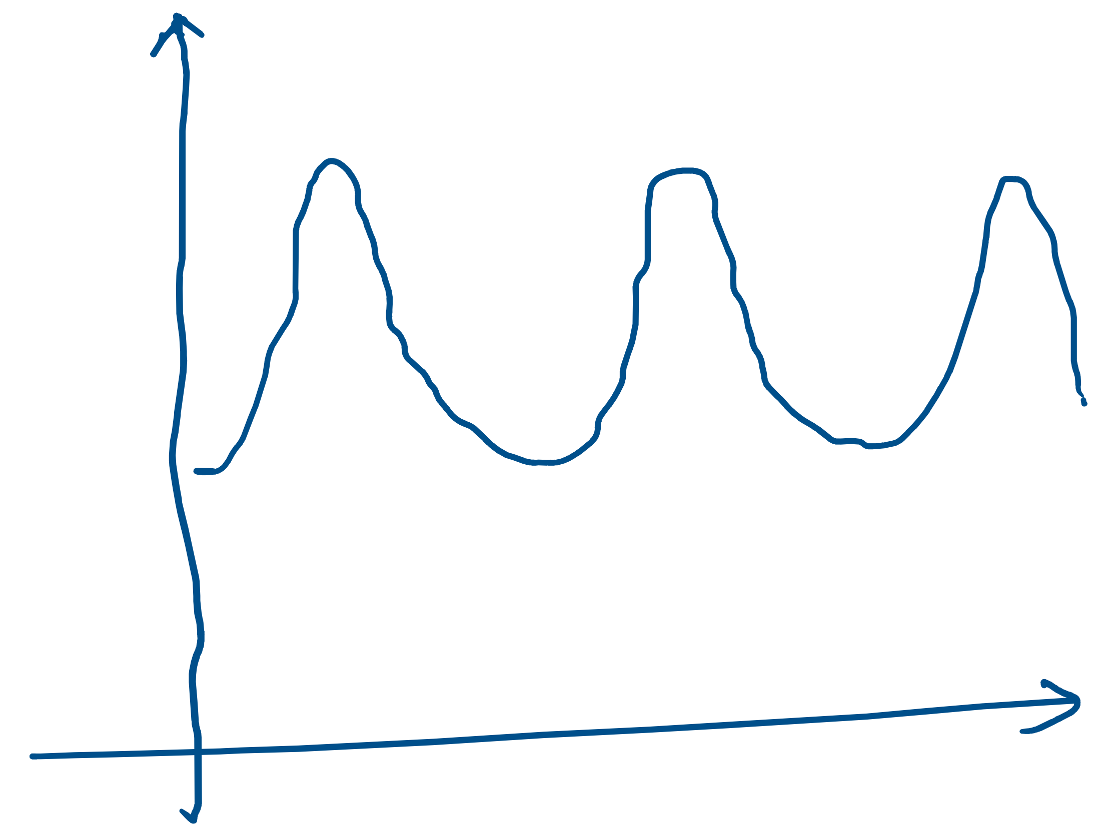
$C_t \rightarrow$ cyclical component (no specific period, usually longer).

$\epsilon_t \rightarrow$ Error component (noise).

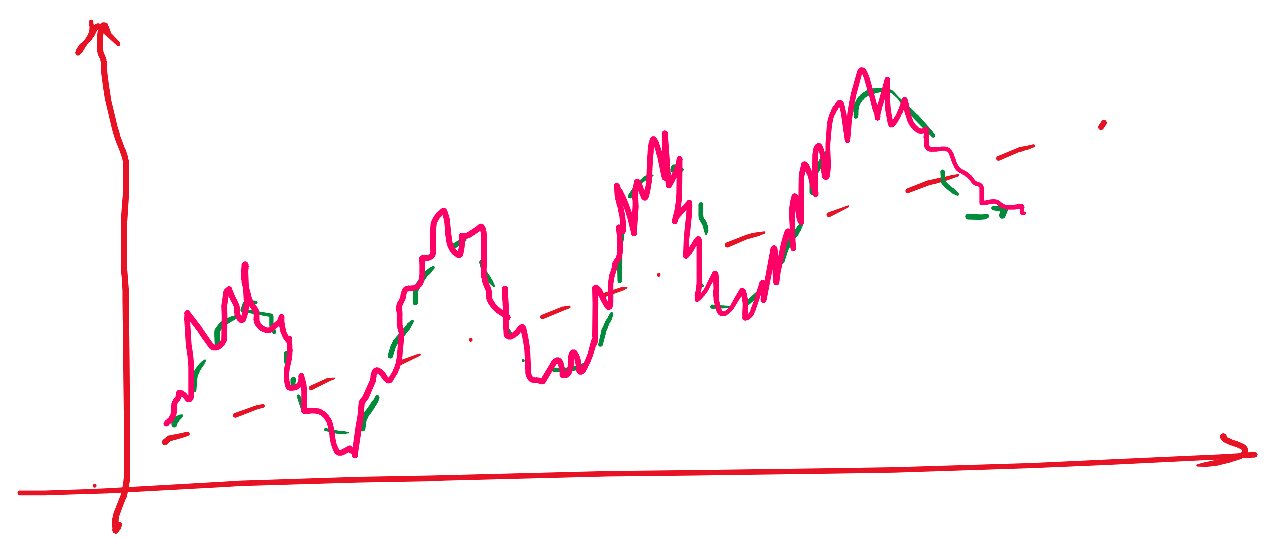
Trends



Seasonality

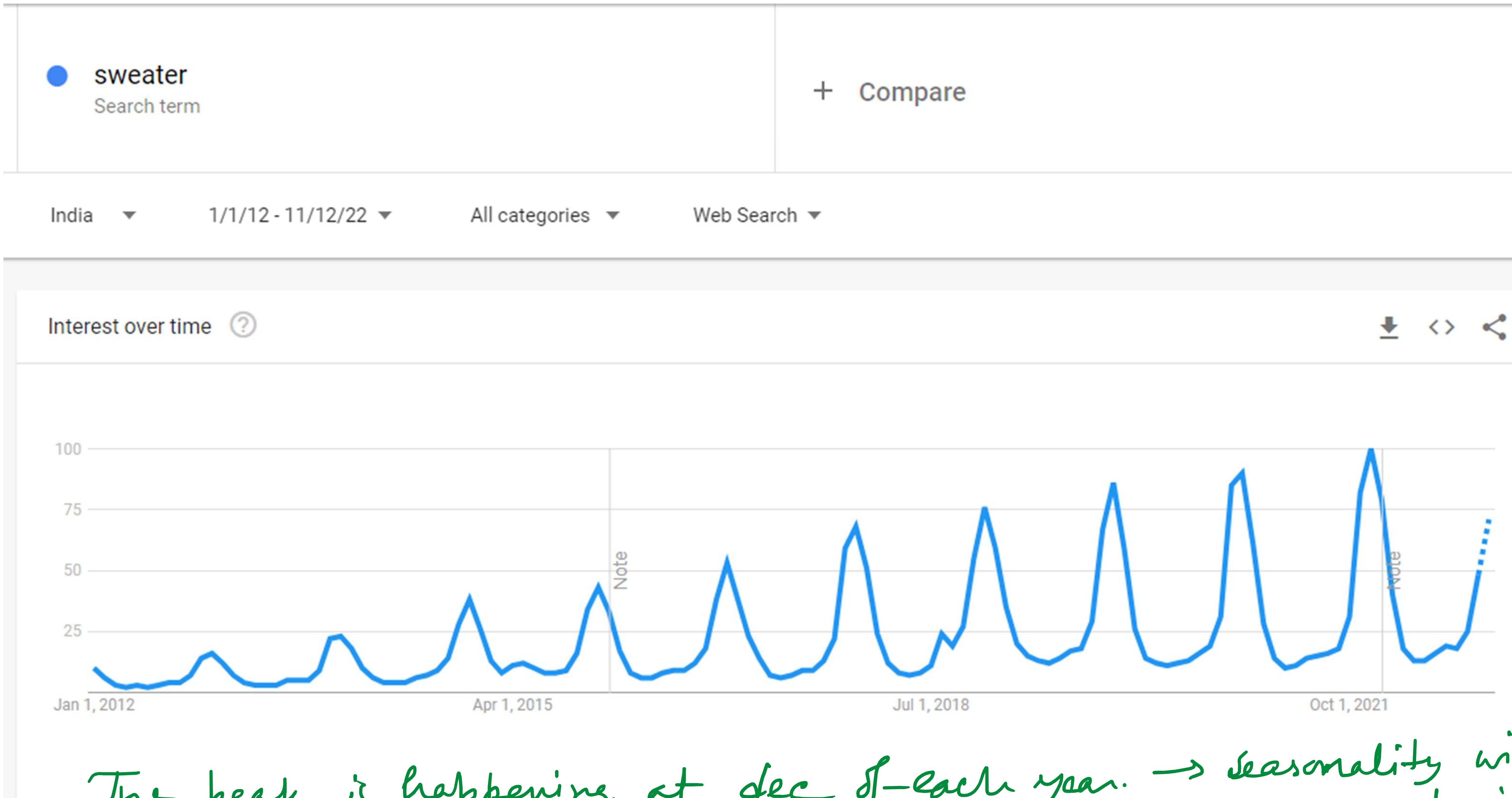


Trend & Seasonality



Example of Seasonal Component in time series

Following is the trend (from google search) of sweater over last 10 years.



The peak is happening at dec of each year. → seasonality with period = 1 year.

Example of cyclical data :-

The cycles are not of specific time period (longer lasting than seasonal)

Market Summary > S&P 500

3,992.93

+3,853.40 (2,761.70%) ↑ all time

11 Nov, 5:06 pm GMT-5 • Disclaimer

+ Follow

1D | 5D | 1M | 6M | YTD | 1Y | 5Y | Max

Open	3,963.72	Low	3,944.82	52-wk high	4,818.62
High	4,001.48	Prev close	3,956.37	52-wk low	3,491.58

Time Series Decomposition

Time series decomposition is the method by which we can break a time series into its constituent components.

There are various methods :-

(1) Hodrick - Prescott filter : H-P filter

$y_t = \tau_t + c_t$ → This algorithm decomposes the time series into trend component & cyclical component.

(2) ETS (Error - Trend - Seasonality) decomposition

$y_t = E_t + \tau_t + S_t$ → This algorithm decomposes the time series into error, trend & seasonal components.