

StatesDataConfig:

1.- Data structure

Class that represents a particular geotemporal configuration. It stores which locations and times will have a desired correlation + monotonic pattern. It contains the following instance variables:

#REGARDING GEOGRAPHY

locations_total = 48

locations_noise_perc = 0

location_forced = None -> location that must has the desired pattern

location_range_forced = None -> location range (north, south, etc) that must have the desired pattern

#REGARDING TIME

times_total = 9

times_noise_perc = 0

time_forced = None

time_range_forced = None

It generates a dataframe with the following columns:

- State, name, x, y: state code, name and x,y values for its centroid.
- x_1-x_times_total, y_1-y_times_total: 2*times_total columns with the data values of the two variables that can be correlated: x and y.
- Pearson_r: referenced pearson of the initial data generated. It should be updated later but it is not done at the moment.
- N, S, W, E: columns that indicate with a one if a particular state belongs to a location_range.
- notnoise_geo: indicates if a particular state follows a particular pattern (=1) or not (=0)

It has one row per state + notnoise_time, that indicates if a particular time follow a particular pattern (=1) or not (=0).

A pattern will only be applied in which both columns of notnoise are equal to 1 at the same.

Example:

#REGARDING GEOGRAPHY

locations_total = 6

locations_noise_perc = 0.33

location_forced = None

location_range_forced = N

#REGARDING TIME

times_total = 6

times_noise_perc = 0.33

time_forced = None

time_range_forced = [2,4]

State	x1	x2	x3	x4	x5	x6	y1	y2	y3	y4	y5	y6	North	notnoise_geo
S1													1	1
S2													1	1
S3													1	1
S4													0	0
S5													0	0
S6													0	0
notnoise_time	0	1	1	1	0	0	0	1	1	1	0	0		

In the above example:

- notnoise_time has ones in times 2, 3 and 4 (columns highlighted with yellow).
- not noise_geo has ones in states S1-S3, which match with the north. Rows highlighted with blue.

Therefore a pattern will be applied to the intersection of both, which is the zone in green.

2.- Add not noisy distractors to original configuration:

Rules:

- Always has 1/3 of noise, for both locations and time.
- When there is a range, location or time range, noise is applied outside the range.
- There are no contiguous rules. I.e., it is not necessary that a distractor is near a real notnoise data.

In the following diagram lighter colors indicate distractors.

State	x1	x2	x3	x4	x5	x6	y1	y2	y3	y4	y5	y6	North	notnoise_geo
S1													1	1
S2													1	1
S3													1	1
S4													0	0
S5													0	0
S6													0	0
notnoise_time	0	1	1	1	1	0	0	1	1	1	1	0		

CorrelatedDataGenerator:

1.- Data generation algorithm

Given a StatesDataConfig generates correlated data. It has the following instance variables:

geotemp_config = config -> instance of StatesDataConfig

r = 0

monotonic = False

type_monotonic = None -> 1 indicates increasing, 0 indicates decreasing

r_exception = 0

monotonic_exception = False

type_monotonic_exception = None.

S3													1	1
S4													0	0
S5													1	0
S6													0	0
notnoise_time	0	1	1	1	1	0	0	1	1	1	1	0		

2.- Mirror data generation algorithm

Geographical shuffle:

Case A: There is no location_range (zone) assigned.

All locations are shuffled, ensuring that the location_forced follows the desired pattern (ie, notnoise_geo = 1).

Case B: There is location_range (zone) assigned.

Locations inside the delimited location_range are shuffled among them. Locations outside a location_range are shuffled among them.

Temporal shuffle:

Case A: There is no time_range (period) assigned and no monotonic behavior

All times are shuffled, ensuring that the time_forced follows the desired pattern (ie, notnoise_time = 1).

Case B: There is time_range (period) assigned and no monotonic behavior.

Times inside the delimited time_range are shuffled among them. Times outside a time_range are shuffled among them.

Case C: There is monotonic behavior.

Mirror it: if it is increasing, make it decreasing and vice versa.