Observables

TheSimulationResult object, provides the following observables (usualy defined as val result=zombieInvasion(...) in the ScalaTask). Some of the observable are time series. In this case, observable can be aggregated via a temporal step. The by : Int = 20 notations stand for the temporal step parameter by default. All time-serie indicators take such a by parameter, which samples the original 500 steps time-serie by this fixed step width (for convenience and size of output data).

Agent-related indicators

- `humansDynamic(by: Int = 20): Array[Int]` sampled time serie (each by time steps) of number of humans
- `walkingHumansDynamic(by: Int = 20): Array[Int]` sampled time serie of number of walking humans
- `runningHumansDynamic(by: Int = 20): Array[Int]` sampled time serie of number of running humans
- `zombiesDynamic(by: Int = 20): Array[Int]` sampled time serie of number of zombies
- `walkingZombiesDynamic(by: Int = 20): Array[Int]` sampled time serie of number of walking zombies
- `runningZombiesDynamic(by: Int = 20): Array[Int]` sampled time serie of number of running zombies

Event-related indicators

- `rescuedDynamic(by: Int = 20): Array[Int]` sampled time serie of number of rescued humans
- `killedDynamic(by: Int = 20): Array[Int]` sampled time serie of killed zombies
- `zombifiedDynamic(by: Int = 20): Array[Int]` sampled time serie of number of zombified humans
- `fleeDynamic(by: Int = 20): Array[Int]` sampled time serie of number of humans fleeing from zombies
- `pursueDynamic(by: Int = 20): Array[Int]` sampled time serie of number of zombies pursuing humans
- `humansGoneDynamic(by: Int = 20): Array[Int]` sampled time serie of number of humans who left the world
- `zombiesGoneDynamic(by: Int = 20): Array[Int]` sampled time serie of number of zombies who left the world

Global indicators

- `totalZombified: Int` total number of zombified humans over the course of the simulation
- `halfZombified: Int` time at which half of humans are zombified
- `peakTimeZombified(window: Int = 20): Int` time at which the zombification is the most intense (smoothed over a window size `window`)
- `peakSizeZombified(window: Int = 20): Int` number of zombification when zombification is the most intense (smoothed over a window size `window`)
- `totalRescued: Int` total number of humans rescued
- `halfTimeRescued: Int` time at which half of the humans have been rescued
- `peakTimeRescued(window: Int = 20): Int` time at which rescue is the most intense (smoothed over a window size window)
- `peakSizeRescued(window: Int = 20): Int` number of rescue at the time of peakTimeRescued

Spatial indicators

- `spatialMoranZombified: Double` spatial autocorrelation of the location of zombification events cumulated over time. Take values between -1 (strongest negative autocorrelation) 0 (no spatial autocorrelation) and 1 (strongest autocorrelation)
- `spatialDistanceMeanZombified: Double` average distance between zombification events
- `spatialEntropyZombified: Double` entropy of zombification events, or how zombification is uniformally distributed across cells ($\in[0;1]$)
- `spatialSlopeZombified: Double` level of aggregation of zombification events, can be interpreted as "clustring" intensity.