**DOCUMENTATION MODEL BUILD**

EXPERIMENT #1 17/08/2022

First parameter setting with ensemble s-curve observed

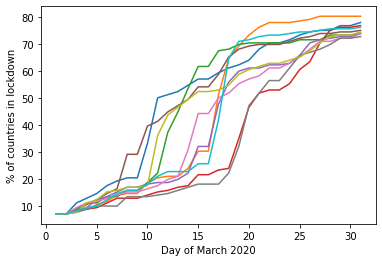
N = 10

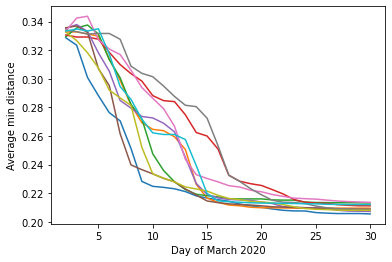
index\_for\_choice = 20 (this is the number of countries, that are taken by every individual country into account)

CountryModel(0.015, 0.12)

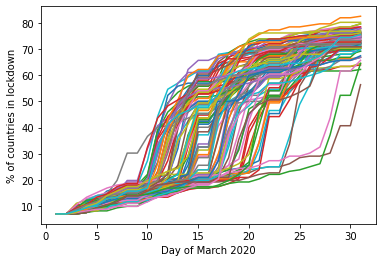
Base\_alert = 0.015% (this is similar to broadcasting influence in typical bass diffusion model)

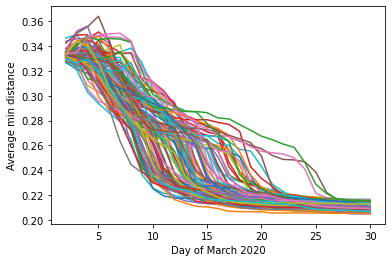
Social\_alert\_threshold = 0.12





N= 100





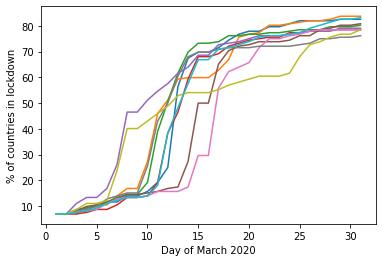
First parameter setting with ensemble s-curve observed

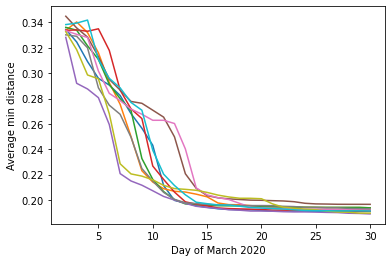
N = 10

Clique\_size = 15 (this is the number of countries, that are taken by every individual country into account)

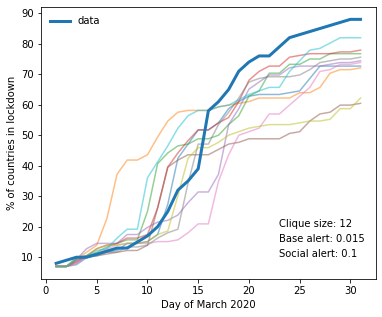
Base\_alert = 0.015 (this is similar to broadcasting influence in typical bass diffusion model)

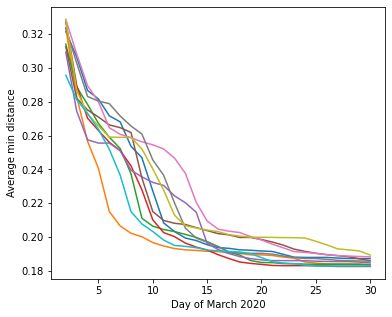
Social\_alert\_threshold = 0.12 (this is the threshold that needs to be passed (from below) so that a country adopts lockdown based on social influence)

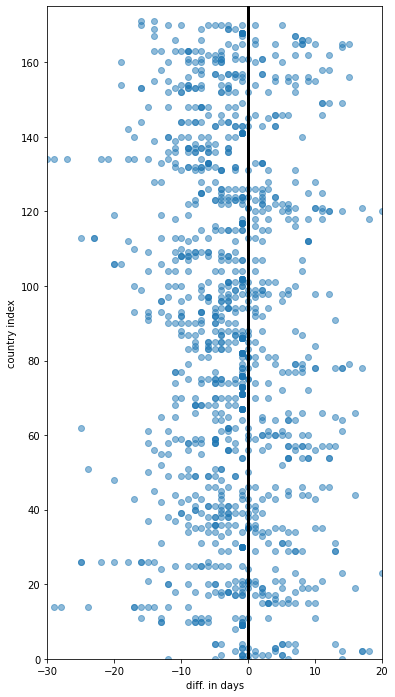


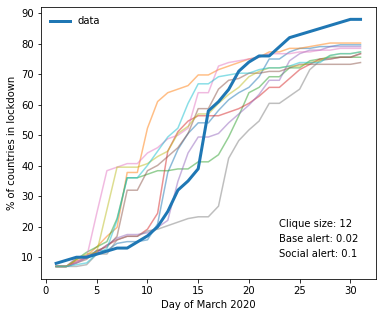


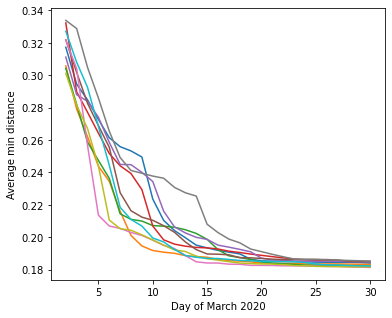
EXPERIMENT #2 18/08/2022

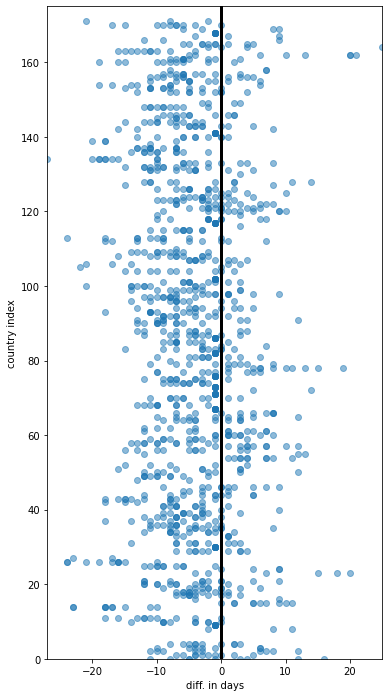


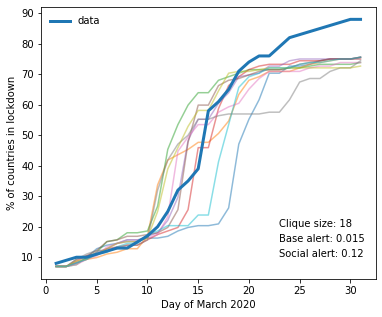


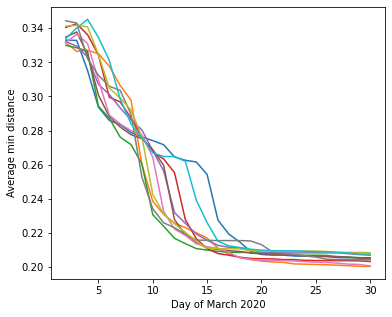


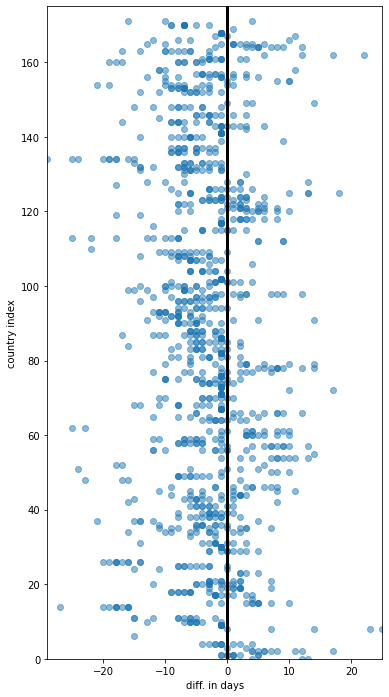








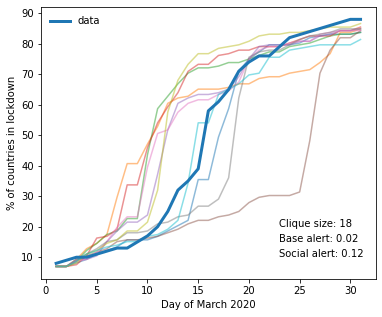


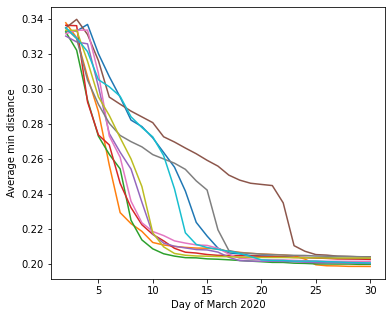


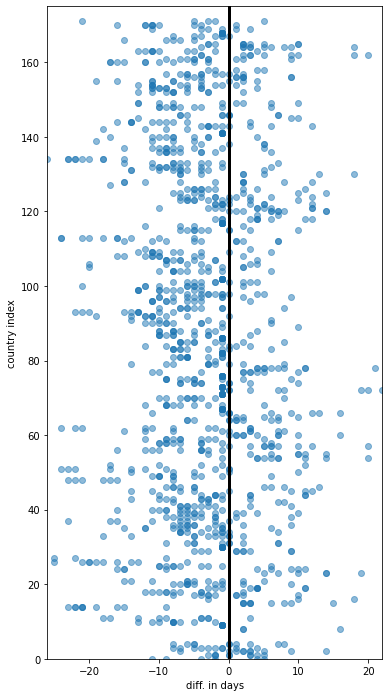
EXPERIMENT #3 18/08/2022

The rule to set the base alert threshold includes now both pop density as well as a weight by democracy

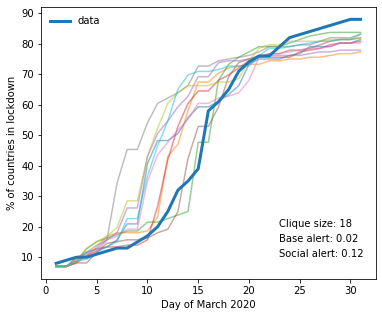
a.alert\_adoption\_threshold = (agent\_data["log\_population\_density\_normalized\_on\_average"][i])\*\*2 \* (1/agent\_data["democracy\_index\_normalized\_on\_average"][i]) \* base\_alert

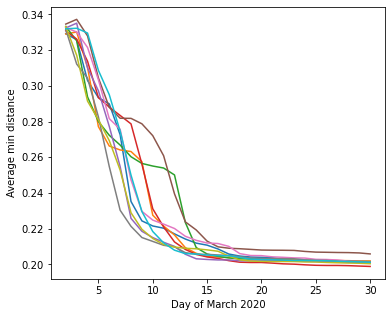


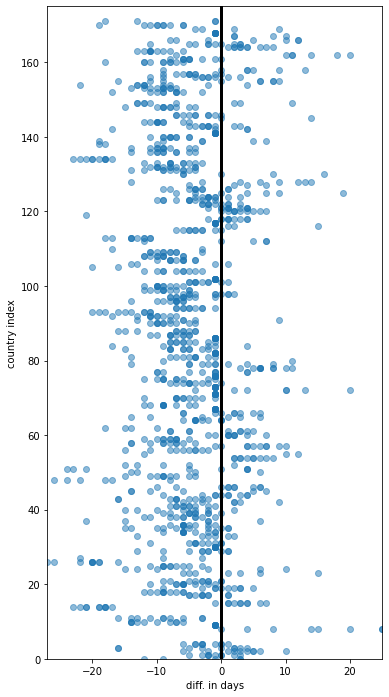


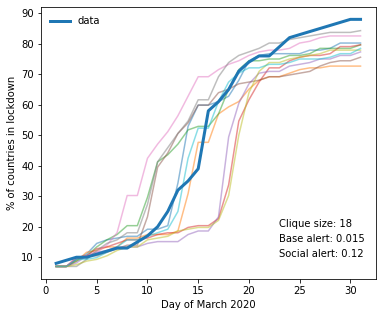


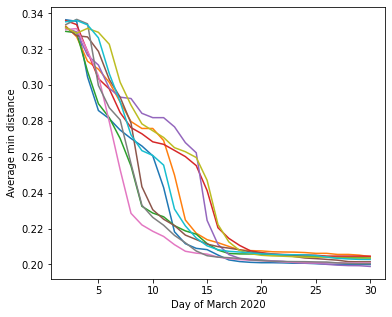
a.alert\_adoption\_threshold = (agent\_data["log\_population\_density\_normalized\_on\_average"][i])\*\*2 \* np.sqrt(1/agent\_data["democracy\_index\_normalized\_on\_average"][i]) \* base\_alert

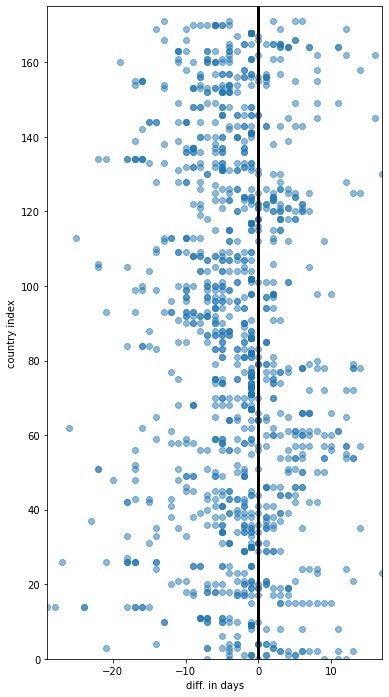




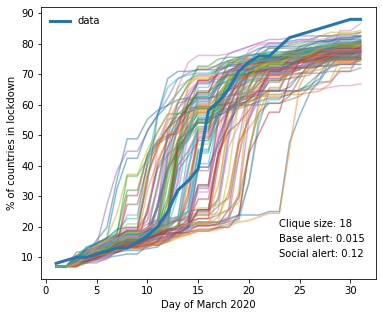


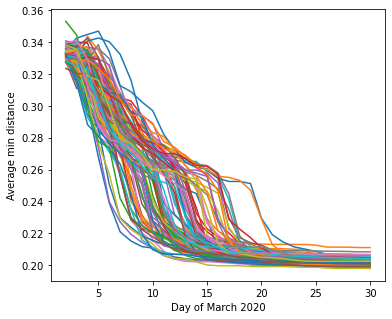


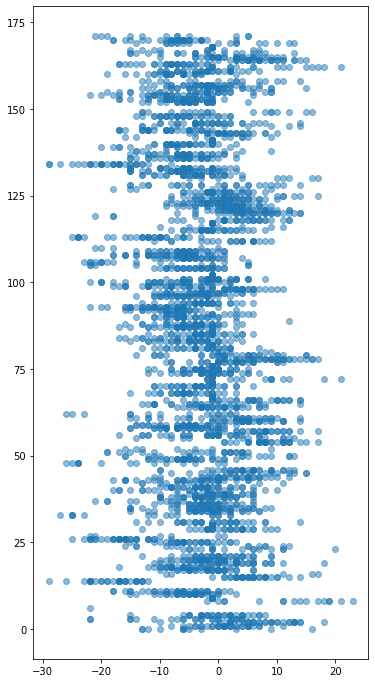




Same as one above but for N = 100







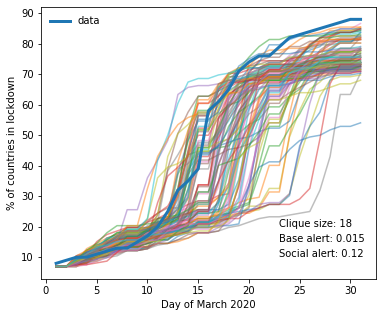
EXPERIMENT #4 18/08/2022

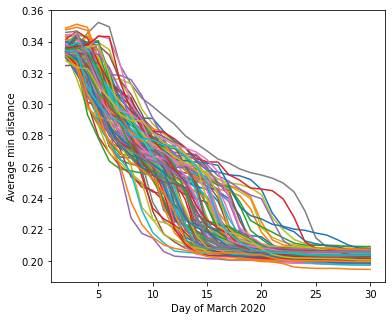
New rule implement square root now influence of democracy on social base threshold. To give autocracies more social reaction affinity. (which as seen below delays lockdowns too much)

a.alert\_adoption\_threshold = (agent\_data["log\_population\_density\_normalized\_on\_average"][i])\*\*2 \* np.sqrt(1/agent\_data["democracy\_index\_normalized\_on\_average"][i]) \* base\_alert

#### but the more democratic a country, the more sensitive they are to the influence of others

a.social\_adoption\_threshold = np.sqrt(agent\_data["democracy\_index\_normalized\_on\_average"][i]) \* social\_base\_threshold



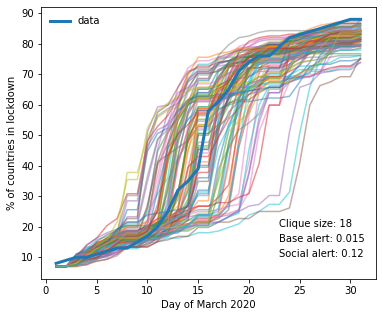


EXPERIMENT #5 18/08/2022

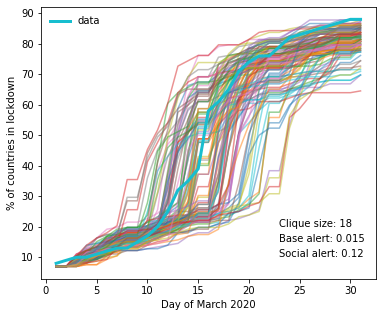
a.alert\_adoption\_threshold = (agent\_data["log\_population\_density\_normalized\_on\_average"][i])\*\*2 \* (1/agent\_data["democracy\_index\_normalized\_on\_average"][i]) \* base\_alert

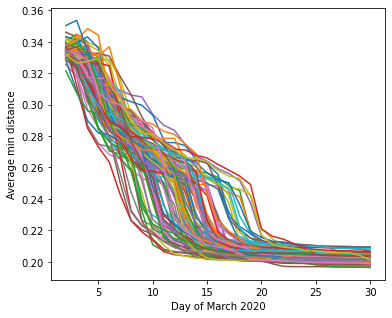
#### but the more democratic a country, the more sensitive they are to the influence of others

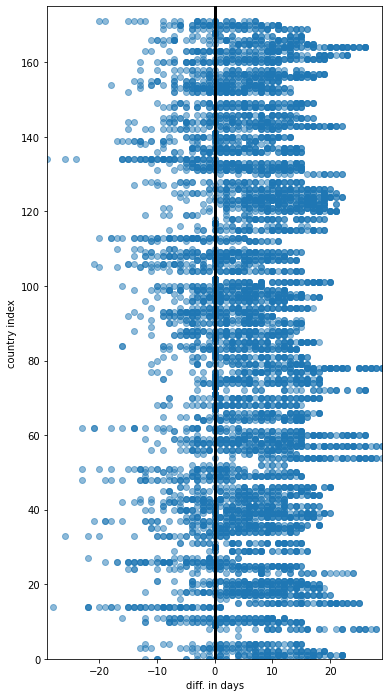
a.social\_adoption\_threshold = (agent\_data["democracy\_index\_normalized\_on\_average"][i]) \* social\_base\_threshold

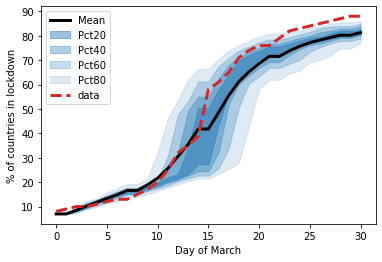


EXPERIMENT #6 19/08/2022

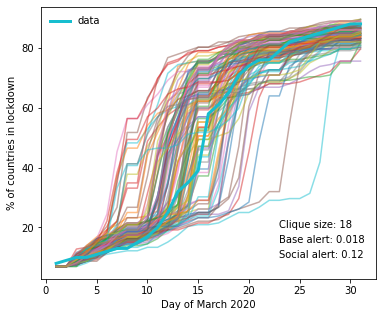


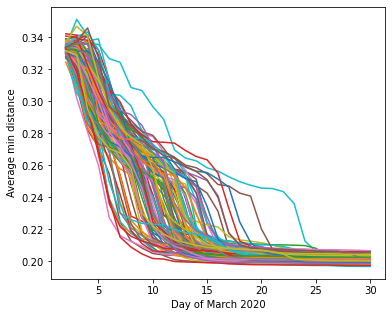


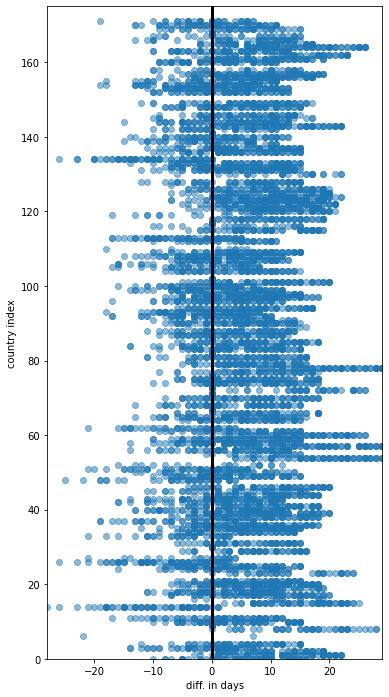


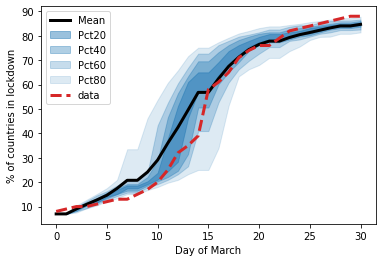


N = 109









EXPERIMENT #7 19/08/2022

Question is

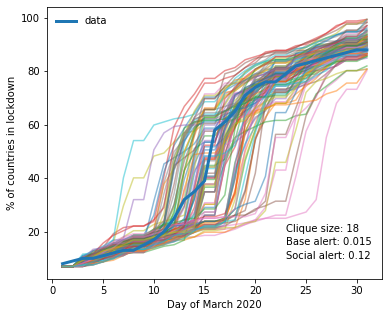
With “global” level mechanism . Yes because overall system state ( the number of lockdowned acountries) influences agent level variable. On the other hand one can easily imagine (and in fact they do anyway already) that countries count the other lockdowned countries.

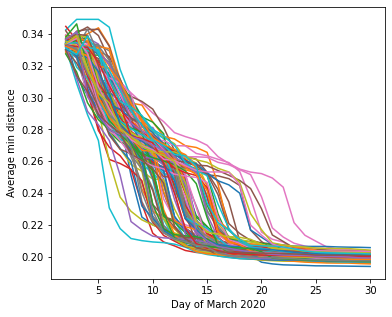
In def step() in agent class

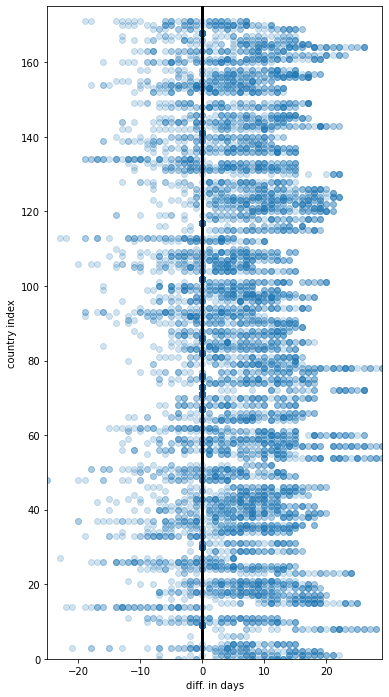
self.social\_adoption\_threshold = self.social\_adoption\_threshold + 0.000015\*np.exp(0.04\*len(list\_of\_lockdown\_countries))

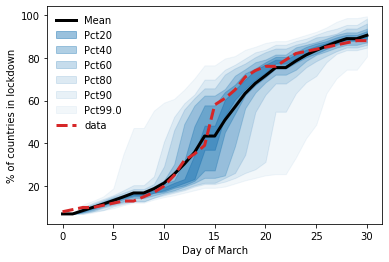
a\*e^(b\*number\_of\_countries)

N = 100





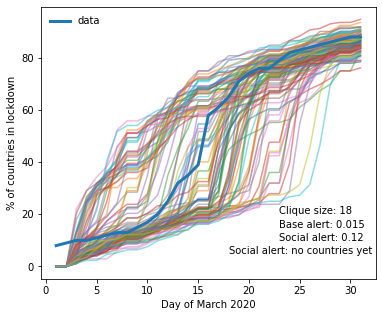


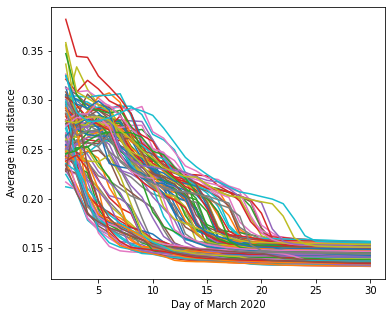


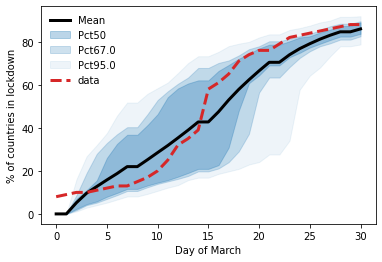
Experiment #8 22/08/2022

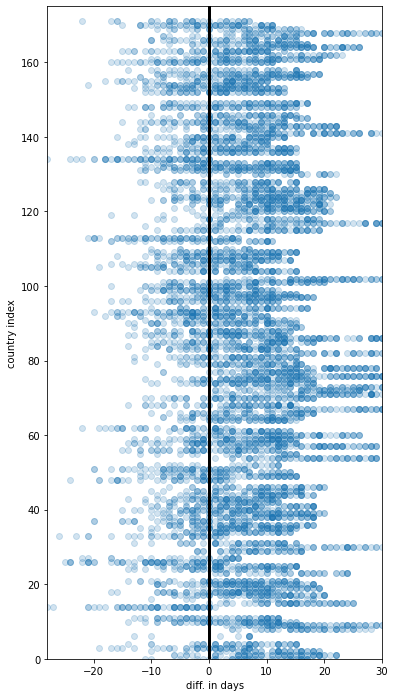
0 countries have adopted as initial conditions

N = 100



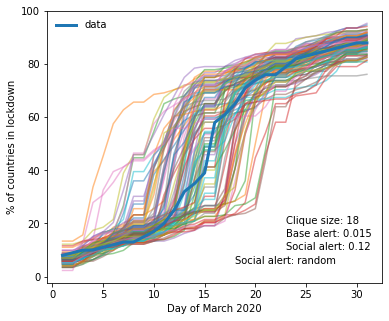


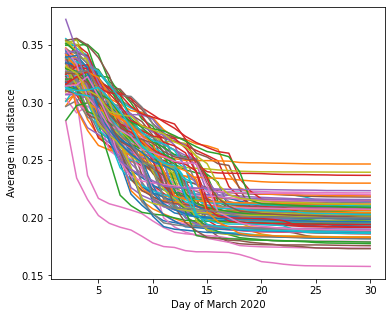


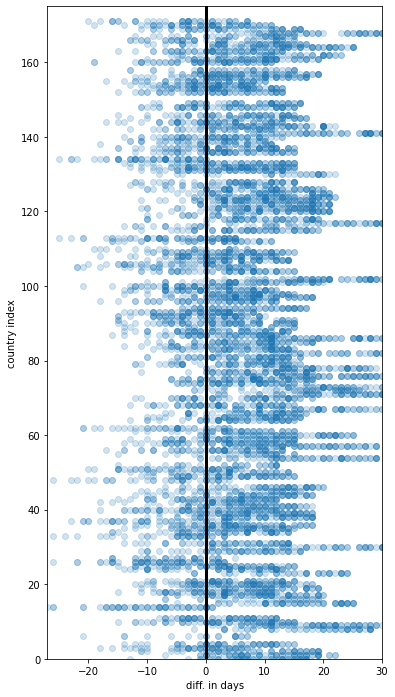


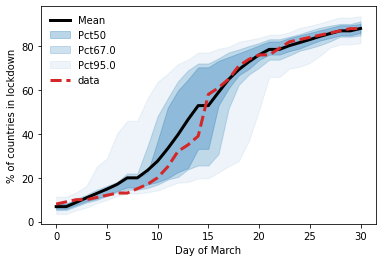
Experiment #9 22/08/2022

And importantly random initial conditions but with fixed density 0.07

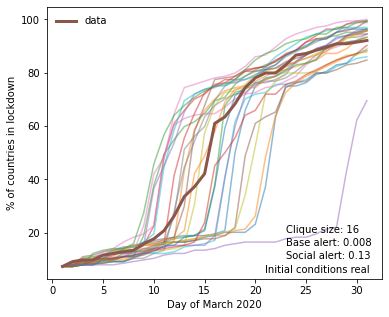


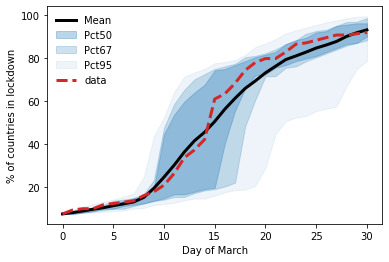


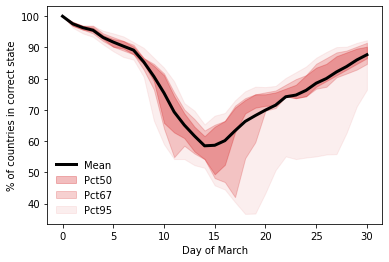




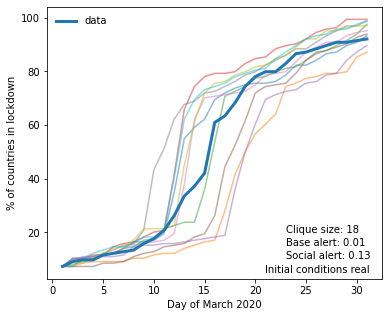
Experiment #10 26.08.2022







Experiment #11 26.08.2022

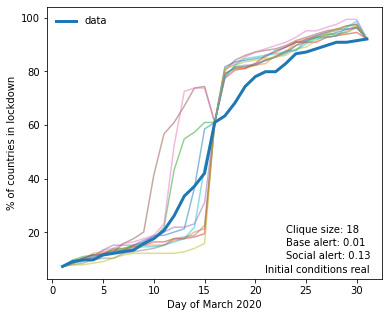


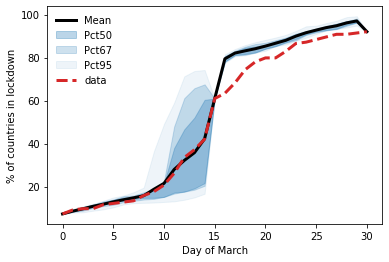
self.alert\_adoption\_threshold = self.alert\_adoption\_threshold + (self.alert\_adoption\_threshold + 0.00003\*np.exp(0.044\*len(list\_of\_lockdown\_countries)) - self.alert\_adoption\_threshold)

Experiment #12 26.08.2022

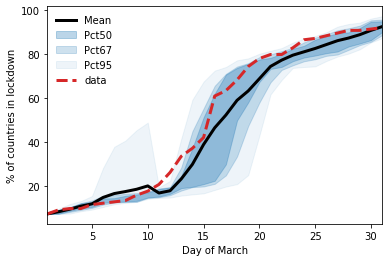
Parameter self.data\_update == "yes"

With direct data updates here after 15 steps in the middle, the model is corrected to the exact actual state. From then on however, it does not do a good job of predicting the actual curve. Even though before the mean was surpisingly correct.

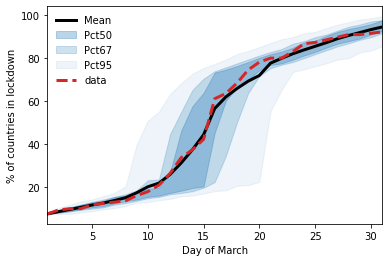


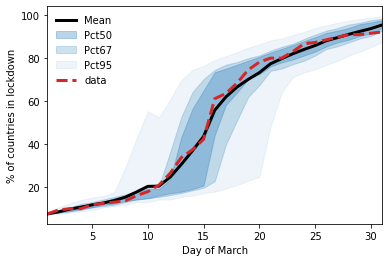


Experiment #13 PARTICLE FILTER ONE 12.10.2022

No of particles = 10, weird behaviour or working data assimilation?

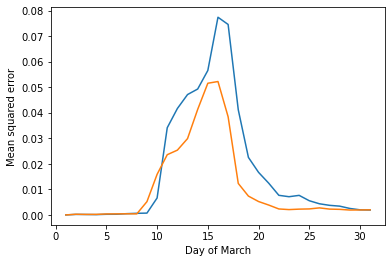
No of particles = 100 not sure variance reduction as good as hoped for, perhaps need to change/improve error metric

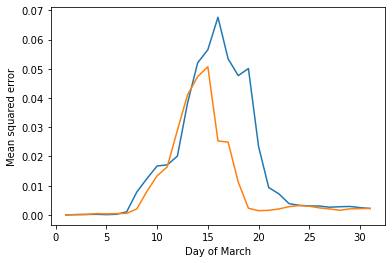


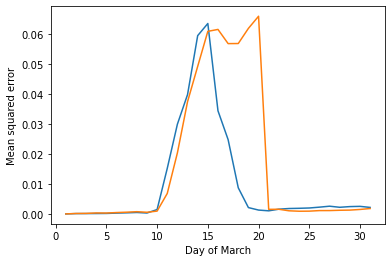


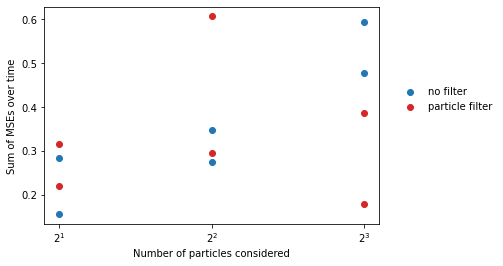
Experiment #14: 17/10/2022

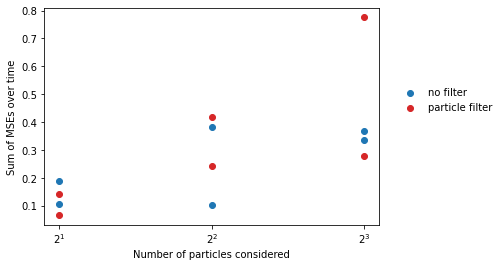
Particle filter MSE over time vs. model alone: 10 simulations

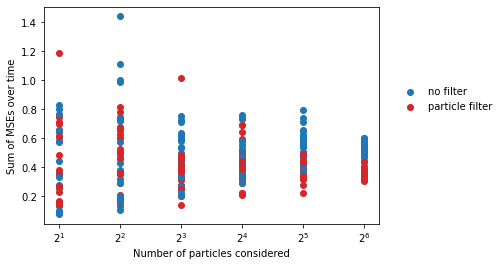












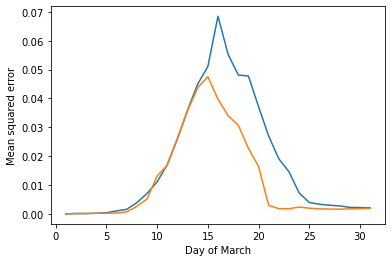
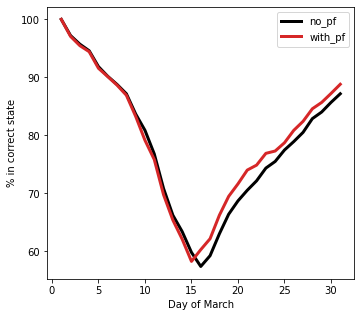
Regression line and more particles

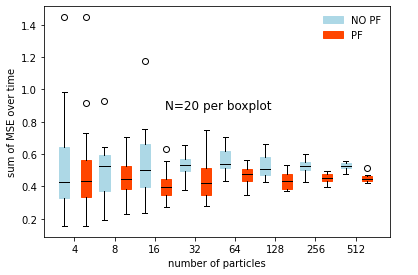
One true trajectory

64 extremely few particles, yess.. but 20\*64 not….

21/11/2022

Microvalidity is weird and should be improved more through particle filter but it doesn’t really here with n= 100

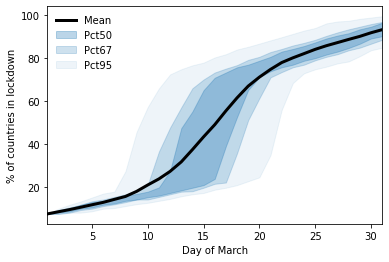


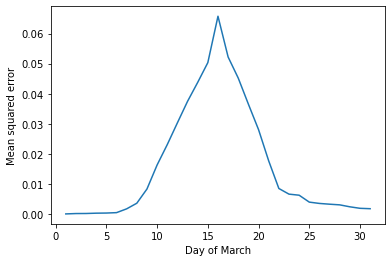


The graph above was wrong since the axes of overlapping graph were not correctly scaled same to each other

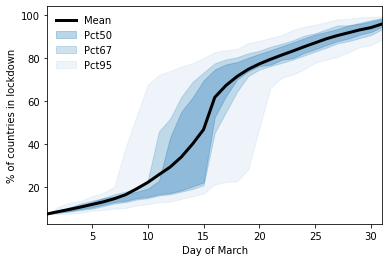
FIRST TIME EXPERIMENTING WITH DA WiNDOW Size systematically

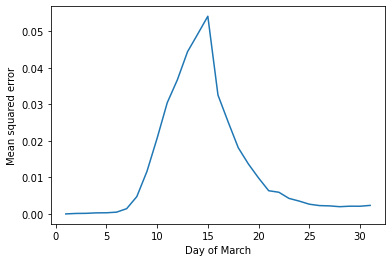
Interesting see difference between no da and da at t = 15



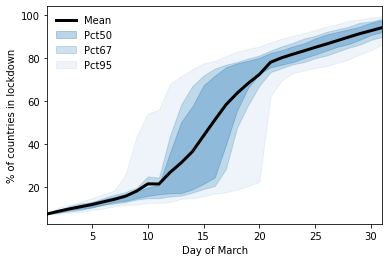


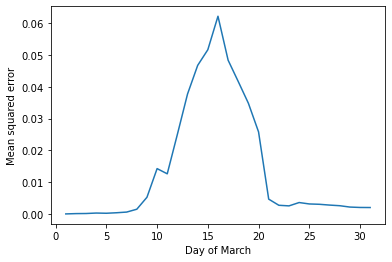
Vs..





However then with DA at t = 10, 20 and 30





With da after every time step

