**Equations - Dec 7 2020**

STOCKS

[**'P\_RR'**, 0.2, **'Resources\_RMNCH'** , **None**],  
 [**'L4\_HF'**, 0.2, **'HC\_Financing\_4/5'** , **None**], *#* [**'L2\_HF'**, 0.2, **'HC\_Financing\_2/3'** , **None**], *#* [**'S\_FR'**, 0.2, **'SDR\_Facility\_Repurposing'** , **None**],  
 [**'L4\_HR'**, 0.3, **'HR\_Readiness\_4/5'** , **None**],  
 [**'L2\_HR'**, 0.2, **'HR\_Readiness\_2/3'** , **None**],  
 [**'L4\_DC'**, 0.4, **'Delivery\_Capacity\_4/5'** , **None**],  
 [**'L2\_DC'**, 0.9, **'Delivery\_Capacity\_2/3'** , **None**],  
 [**'P\_D'**, 0.6, **'Performance\_Data'** , **None**],  
 [**'L4\_Q'**, 0.5, **'Quality\_4/5'** , **None**],  
 [**'L2\_Q'**, 0.4, **'Quality\_2/3'** , **None**], *#*

FACTORS THAT INFLUENCE FLOWS

[**'Funding\_MNCH'**, 0.2, **'Funding\_MNCH'** , **None**],  
 [**'Adherence\_budget'**, 0.2, **'Adherence\_budget'** , **None**],  
 [**'Employee\_incentives'**, 0.2, **'Employee\_incentives'** , **None**],  
 [**'Strong\_referrals'**, 0.2, **'Strong\_referrals'** , **None**],  
 [**'Training\_incentives'**, 0.2, **'Training\_incentives'** , **None**],

[**'BL\_Capacity\_factor'**, 20, **'BL Capacity Factor'** , 20 ],

MOTHER FACTORS  
 [**'Initial\_Negative\_Predisp'**, 2, **'Initial Negative Predisp'** , **None**],  
 [**'Health\_outcomes\_\_Predisp'**, 3.2, **'Health outcome -> Predisp hospital'**, 1.0], *#* [**'L4\_Q\_\_Predisp'**, 0.5, **'L4/5 quality -> Predisp hospital'** , 0.2],  
 [**'Predisp\_L2\_L4'**, 2.0, **'Predisp L2'** , 1.0], *#* [**'Health\_const\_0'**, 0.8, **'Initial Health (do not change)'** , **None**], *#* [**'Health\_slope\_0'**, 0.2, **'Variability Initial Health (do not change)'**, **None**],*#* [**'Q\_Health\_multiplier'**, 4., **'Q Health Effect'** , **None**], *#* [**'Q\_Health\_L4\_constant'**, 1.5, **'L4 Health Effect'** , **None**],  
 [**'Q\_Health\_L4\_L2\_difference'**, 1., **'Difference L4-L2 Health Effect'** , **None**],  
 [**'Q\_Health\_L4\_referral\_difference'**, 0.5, **'L2 -> L4 Referral Health Effect'** , **None**],  
 [**'Q\_Health\_Home\_negative'**, 3.0, **'Home Delivery Health Effect'** , **None**], *#* [**'Time\_delay\_awareness'**, 2.0, **'Awareness Delay (months)'** , **None**], *# 24*

TARGETS OF EACH STOCK (LIMITING UPPER BOUND)  
 [**'P\_D\_target'**, 0.7, **'Data Performance target'** , **None**],  
 [**'L4\_HF\_target\_0'**, 0.8, **'L4/5 HC Financing target'** , **None**],  
 [**'L2\_HF\_target\_0'**, 0.6, **'L2/3 HC Financing target'** , **None**],  
 [**'L4\_target\_0'**, 0.9, **'L4 target'** , **None**],  
 [**'L2\_target\_0'**, 0.9, **'L2 target'** , **None**],  
 [**'S\_FR\_target\_0'**, 0.7, **'SDR FR target'** , **None**],  
 [**'S\_T\_target\_0'**, 0.9, **'SDR T target'** , **None**],  
 [**'dL4\_DC\_in\_0'**, 0.2, **'L4 Rate of Capacity Increase'**, **None**],  
 [**'dL2\_DC\_in\_0'**, 0.2, **'L2 Rate of Capacity Decrease'**, **None**],  
 [**'P\_RR\_target\_0'**, 1.0, **'Resources RMNCH target'** , **None**],  
 [**'L4\_DC\_target'**, 0.9, **'L4 Delivery Capacity Target'** , **None**],  
 [**'L2\_DC\_target'**, 0.1, **'L2 Delivery Capacity Target'** , **None**],

BL\_Capacity\_factor = multiplier used for capacity

L4\_D\_Capacity\_Multiplier = multiplier used for capacity of L4 in comparison with L2

Target RR is related to Funding\_MNCH

Demand (L\_demand), Capacity (L\_D\_Capacity)

L2\_D\_Capacity[t] = L2\_DC[t] \* BL\_Capacity\_factor  
L4\_D\_Capacity[t] = L4\_DC[t] \* BL\_Capacity\_factor \* L4\_D\_Capacity\_Multiplier  
L2\_demand = logistic(num\_deliver[t,1] / (L2\_D\_Capacity[t]))  
L4\_demand = logistic(num\_deliver[t,2] / (L4\_D\_Capacity[t]))  
  
P\_RR\_target = P\_RR\_target\_0 \* logistic([Funding\_MNCH, 3])  
dP\_RR\_in = P\_D[t] need to subtract a constant to allow for decrease in RR  
  
L2\_HF\_target = L2\_HF\_target\_0 \* P\_RR[t] \* logistic([Adherence\_budget, 2])  
L4\_HF\_target = L4\_HF\_target\_0 \* P\_RR[t] \* logistic([Adherence\_budget, 2])  
dL2\_HF\_in = P\_RR[t] *# coefficients of these three dStock\_in terms add up to 1*dL4\_HF\_in = P\_RR[t]  
L2\_target\_combined\_0 = L2\_target\_0 \* L2\_HF[t] *# combined targets of L2\_HR and L2\_S =0.9\*target of L2\_HF*L2\_HR\_target = L2\_target\_combined\_0 \*

logistic([Employee\_incentives, Strong\_referrals, Training\_incentives, 3])  
dL2\_HR\_in = L2\_HF[t]  
L4\_target\_combined\_0 = L4\_target\_0 \* L4\_HF[t]  
L4\_HR\_target = L4\_target\_combined\_0 \*

logistic([Employee\_incentives, Strong\_referrals, Training\_incentives, 3])  
dL4\_HR\_in = L4\_HF[t]  
S\_FR\_target = S\_FR\_target\_0 \* P\_RR[t] \*

logistic([Employee\_incentives, Strong\_referrals, Training\_incentives, 3])  
S\_T\_target = S\_T\_target\_0 \* P\_RR[t] \*

logistic([Employee\_incentives, Strong\_referrals, Training\_incentives, 3])  
dS\_FR\_in = P\_RR[t]  
dS\_T\_in = P\_RR[t]  
  
dL2\_DC\_in = dL2\_DC\_in\_0 \* S\_FR[t] *# target < stock so need to reverse sign here*dL4\_DC\_in = dL4\_DC\_in\_0 \* S\_FR[t]  
  
L2\_Q\_target = L2\_HR\_target / L2\_target\_combined\_0 \* logistic([Strong\_referrals, -9\*L2\_demand,5])  
L4\_Q\_target = L4\_HR\_target / L4\_target\_combined\_0 \* logistic([Strong\_referrals, -9/2\*L4\_demand,5])  
dL2\_Q\_in = L2\_HR[t]  
dL4\_Q\_in = L4\_HR[t]

l2\_quality\_avg = window\_average(L2\_Q, t+1, time\_window, linear\_weights=**True**)  
l4\_quality\_avg = window\_average(L4\_Q, t+1, time\_window, linear\_weights=**True**)  
L2\_4\_health\_outcomes\_avg = window\_average(P\_D, t+1, time\_window, linear\_weights=**True**)

L2\_net\_capacity = 1 - (L2\_deliveries + 0) / L2\_D\_Capacity[t] *# add 1 to see if one more can be delivered*mother.increase\_age(l4\_quality\_avg, l2\_quality\_avg, L2\_4\_health\_outcomes\_avg, L2\_net\_capacity, mothers, t)

neg\_HO\_t = window\_average(neg\_HO, t+2, time\_window, linear\_weights=**True**, x1=num\_deliver,

x0=(1-P\_D[0]))  
deliveries\_t = window\_average(num\_deliver, t+2, time\_window, linear\_weights=**True**)  
num\_deliver\_avg[t+1,:] = window\_average(num\_deliver, t+2, time\_window, linear\_weights=**True**)  
neg\_HO\_avg[t+1,:] = neg\_HO\_t  
  
l2\_quality = L2\_Q[t+1]  
l4\_quality = L4\_Q[t+1]

**if** np.prod(deliveries\_t[:2]) == 0:  
 L2\_4\_health\_outcomes = P\_D[0] *# use home value***else**:  
 *# don't include neg\_HO\_t[3] = total neg HO* L2\_4\_health\_outcomes = 1 - sum(neg\_HO\_t[1:3]) / sum(deliveries\_t[1:3])

L\_health\_outcomes = np.ones(3) \* P\_D[0]  
**for** k **in** range(3):  
 **if** deliveries\_t[k] > 0:  
 L\_health\_outcomes[k] = 1 - neg\_HO\_t[k] / deliveries\_t[k]

elif:

L\_health\_outcomes[k] = P\_D[0]  
  
P\_D[t+1] = L2\_4\_health\_outcomes  
  
prob\_l4, prob\_l2, logit\_health\_l4, logit\_health\_l2, logit\_health\_l4\_l2, logit\_health\_l0 = \  
 get\_prob\_logit\_health(B, l4\_quality, l2\_quality, L2\_4\_health\_outcomes, B[**'Health\_const\_0'**])

**Mother Class**

self.logit\_health = Health\_const\_0 + Health\_slope\_0 \* (np.random.uniform(-1, 1, 1))  
self.logit\_health\_BL = self.logit\_health  
self.\_health = logistic(self.logit\_health)

prob\_l4, prob\_l2, logit\_health\_l4, logit\_health\_l2, logit\_health\_l4\_l2, logit\_health\_l0 = \  
 get\_prob\_logit\_health(self.B, l4\_quality, l2\_quality, health\_outcomes, self.logit\_health)  
  
rand = np.random.uniform(0, 1, 1)  
**if** prob\_l4 > rand:  
 self.\_facility = 2  
 self.logit\_health = logit\_health\_l4 *# average initial self.logit\_health is Health\_const\_0***else**:  
 rand = np.random.uniform(0, 1, 1)  
 **if** prob\_l2 > rand:  
 **if** L2\_net\_capacity > 0.0: *# if there is room* self.\_facility = 1  
 self.logit\_health = logit\_health\_l2  
 **else**: *# otherwise, go to level 4/5, but not as healthy* self.\_facility = 2  
 self.logit\_health = logit\_health\_l4\_l2  
 **else**:  
 self.\_facility = 0  
 self.logit\_health = logit\_health\_l0

**if** logistic([self.logit\_health]) < np.random.uniform(0, 1, 1):  
 self.\_delivery = -1  
**else**:  
 self.\_delivery = 1  
  
**if** logistic([self.logit\_health, 1]) < np.random.uniform(0, 1, 1):  
 self.baby\_delivery = -1  
**else**:  
 self.baby\_delivery = 1

Health\_outcomes\_\_Predisp = B[**'Health\_outcomes\_\_Predisp'**]  
L4\_Q\_\_Predisp = B[**'L4\_Q\_\_Predisp'**]  
  
Predisp\_L2\_L4 = B[**'Predisp\_L2\_L4'**] *# 1*Q\_Health\_multiplier = B[**'Q\_Health\_multiplier'**] *# 6*Q\_Health\_L4\_constant = B[**'Q\_Health\_L4\_constant'**] *# 1.5*Q\_Health\_L4\_L2\_difference = B[**'Q\_Health\_L4\_L2\_difference'**] *# 1*Q\_Health\_L4\_referral\_difference = B[**'Q\_Health\_L4\_referral\_difference'**] *# 0.5*Q\_Health\_Home\_negative = B[**'Q\_Health\_Home\_negative'**] *# 0.5*Initial\_Negative\_Predisp = B[**'Initial\_Negative\_Predisp'**] *# 0*logit\_predisp\_l4 = L\_Q\_\_Predisp \* l4\_quality \  
 + Health\_outcomes\_\_Predisp \* health\_outcomes\_l4 \  
 - Initial\_Negative\_Predisp  
logit\_predisp\_l2\_nl4 = L\_Q\_\_Predisp \* l2\_quality \  
 + Health\_outcomes\_\_Predisp \* health\_outcomes\_l2 \  
 - Initial\_Negative\_Predisp  
logit\_predisp\_l2\_l4 = logit\_predisp\_l4 + Predisp\_L2\_L4  
  
prob\_l4 = logistic([logit\_predisp\_l4 - 2])  
prob\_l2\_nl4 = logistic([logit\_predisp\_l2\_nl4 - 2])   
  
*# logit\_health\_BL = Health\_const\_0 + Health\_slope\_0 \* (np.random.uniform(-1, 1, 1)) # not used for initialization*logit\_health\_l4 = logit\_initial + Q\_Health\_multiplier \* (l4\_quality - 1 / 2) + Q\_Health\_L4\_constant  
logit\_health\_l2 = logit\_initial + Q\_Health\_multiplier \* (l2\_quality - 1 / 2) + \  
 Q\_Health\_L4\_constant - Q\_Health\_L4\_L2\_difference  
logit\_health\_l4\_l2 = logit\_initial + Q\_Health\_multiplier \* (l4\_quality - 1 / 2) + \  
 Q\_Health\_L4\_constant - Q\_Health\_L4\_referral\_difference  
logit\_health\_l0 = logit\_initial - Q\_Health\_Home\_negative