FIB UPC

STATISTIC MODELLING AND DESIGN OF EXPERIMENT

Simulation of Covid-19 First Wave in Catalonia

Report

Yaëlle Pihan and Marco Carega

June 2022

1. Executive summary

This document reports how we analysed the impact of different factors on Covid-19 First Wave tendencies in Catalonia. We considered specifically the transmission rate, the proportion of persons contained and if some recommendations were made or not to deal with Covid secondary's effects. To do so, we defined a model to represent and simulate Covid-19 First Wave in Catalonia.

2. System Description, Introduction

As introduced before, we aimed to simulate the First Wave of Covid-19 in Catalonia in order to understand the effects of several parameters on its evolution. Thus it is needed to define the First Wave of Covid in Catalonia, that is to say, the period as well as the population concerned.

Catalonia is a territory counting 7653845 inhabitants, where the first case was identified on the 15/2/2020 and the Second Wave Contention began on the 26/10/2020. Therefore we can consider the First wave as the period between these two events.

To keep in mind the reality of the situation, here are several graphs recording the impacts of Covid-19 in Catalonia: https://www.catalannews.com/covid-19/item/coronavirus-in-catalonia-daily-figures-explained-in-graphs.

3. Problem Description

In the context of Covid-19, we all have been (and keep) experiencing the numerous impacts of this pandemic over our lives. In this study we focus on human impacts such as the evolution of the number of person infected, the number of recoveries with or without long term effects and the number of deaths. The objective is, of course, to lower the number of person suffering or dying from this flu, which can be reached by different means.

For example if less people are susceptible to catch the virus, then the global health system would be less overcrowded and more able to take care of infected people, especially the ones at risks. In this continuity, the long term secondary effects could also be taken in charge with recommendations and prescriptions. As a consequence of these observations, we focused on two tasks:

1. Find a way to transform the pics of new Covid cases into a more smoothed evolution. This would provide more time to the health system to adapt to the pandemic (find more reanimation beds, create places to test people, etc...) and ovoid overcrowding.
2. Find a way to lower the number of the persons encountering long term secondary effects.

3.1. Systemic Structural, Systemic Data, and Simplifying Hypotheses

In order to define our model, we established some simplifying, systemic structural and systemic data hypothesis.

3.2. Simplifying hypothesis

- SH1: A person can only catch the virus once.
- SH2: When a new variant of Covid-19 appears, the previous one disappears.
- SH3: All the infected people stay at home.
- SH4: Contained people are not susceptible to be infected.
- SH5: Catalonia is a closed territory, without interaction with others.
- SH6: The probability of catching the virus and recovering or dying does not depend on the state of the health system as well as the revenues of the persons.
- SH7: We don't consider the climatic conditions despite it's potential importance in Covid transmission's rate.

3.3. Systemic Structural hypothesis

- SSH1: The entities are the inhabitants of Catalonia.
- SSH2: The entities evolve from one state to an other over time.
- SSH3: At a day d, an entity is either:
  - Susceptible to catch Covid-19 (1)
  - Exposed to Covid-19 (2)
  - Infected by Covid-19 (3)
  - Recovered from Covid-19 (4.1)
  - or Dead from Covid-19 (4.2) ...and cannot be at an other state.
- SSH4: An entity can only evolve from state 1 to state 4.
- SSH5: When there is a containment, the initial quantity of entities Susceptible is lowered.
- SSH6: The simulation ends when the First Wave is over, that is to say when each state reaches a constant number of entities.

3.4. Systemic Data hypothesis

- SDH1: 7653845 is the number of entities.
- SDH2: At the beginning of the simulation, all the entities minus 1 are Susceptible and 1 entity is Infected.
- SDH3: The default transmission rate is 1,23.
- SDH4: The default incubation period value is 5,5 days.
- SDH5: The default recovery period value is 5 days, the associated default recovery rate is 0,129.
- SDH6: The default fatality rate is 0.002.
- SDH7: By default no one is contained.
- SDH8: The default percentage of person suffering from 2nd effects is 0,85.

These data were taken from the paper *"Modeling Influenza and SARS-CoV-2 interaction, analysis for Catalonia region"* written by our teacher P. Fonseca and his colleagues. Some of these data were slightly modified during the design of experiment process, but the changes are always notified in the report.

4. Model Specification

As announced previously, only five states can characterize the entities:

- Susceptible
- Exposed
- Infected
- Recovered
- Dead

Indeed, we use a SEIRD model to simulate Covid-19 First Wave in Catalonia as P. Fonseca and his colleagues did.

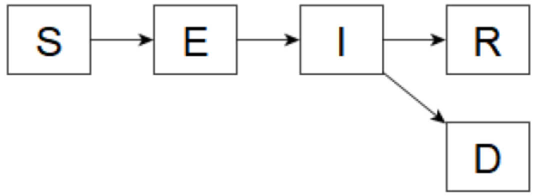


Figura 1: Compartments/Entities states of a SEIRD model

5. Codification

In order to codify our SEIRD model, we used different software.

As a prototype we used an Excel sheet (fig. 2) as a matrix to calculate the number of entities concerned by each state over the time in days. We tried to obtain a realistic evolution, comparing it to an another (and professional) SEIRD model built on Insight Maker

	A	B	C	D	E	F
1	Days	Susceptible S	Exposed E	Infected I	Recovered R	Died D
2	0	8000000	0	1	0	0
3	1	7999999,164	0,836181031	0,871134	0,12757732	0,00129
4	2	7999998,435	0	4,9397796	0,238714263	7999999
5	3	7999994,305	4,130549218	4,3032101	0,868918108	7999999
6	4	7999990,707	0	24,401419	1,417910118	7999999
7	5	7999970,303	20,40397982	21,256906	4,530977722	7999999
8	6	7999952,528	0	120,53751	7,242876821	7999999
9	7	7999851,737	100,7905839	105,00433	22,62072966	7999999
10	8	7999763,936	0	595,42576	36,01690043	8000000
11	9	7999266,067	497,8690359	518,69564	111,9797232	8000000
12	10	7998832,384	0	2941,1986	178,1535224	8000001
13	11	7996373,368	2459,015523	2562,1782	553,3837557	8000005
14	12	7994231,895	0	14527,078	880,2595775	8000008
15	13	7982093,386	12138,50886	12655,032	2733,585272	8000027
16	14	7971535,174	0	71716,773	4348,080337	8000043
17	15	7911780,342	59754,83227	62474,921	13497,51403	8000135

Figura 2: Matrix built on Excel

We obtained with the given parameters (fig. 3), the graph below (fig. 4).

SEIRD	Initial states		Constant	Value
S0	8000000	transmission rate	beta	0,8
E0	0	latency rate	alpha	5
I0	1	recovery rate	gamma	0,12887
R0	0	fatality rate	mu	0,01
D0	0	nb pers in catalogn	N	7653845

Figura 3: Constants used

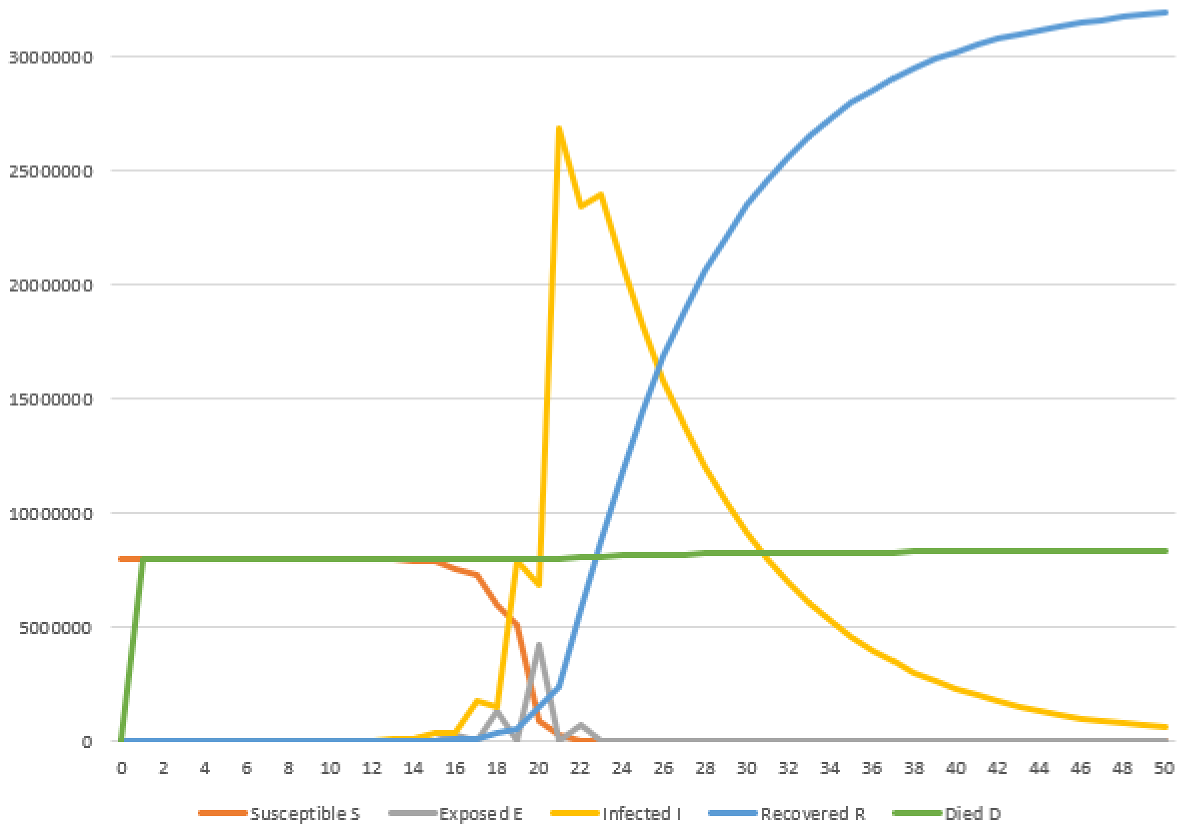


Figura 4: Simulation prototype of the SEIRD model

As Excel was showing some default like the impossibility of keeping a constant number of entities, or the difficulty of having strictly positive results; we decided to build our model on two other tools: one with a Python script and one with Insight Maker.

5.1. Python script

5.1.1. Data

From the prototype we made a Python script where we implemented each formula of the SEIRD model. Considering the day, we compute each value in this way:

$$S' = S - \frac{\beta * S * I}{N}$$

$$E' = E + \frac{\beta * S * I}{N} - \alpha * E$$

$$I' = I + \alpha * E - \gamma * I$$

$$R' = R + \gamma * (1 - \mu) * I$$

$$D' = D + \gamma * \mu * I$$

Moreover we computed the number of contained people C and the percentage of secondary effects L, to do it we used these formulas:

$$C' = M - S$$

$$L' = \nu * R$$

Furthermore, in order to get the right results, we did two more steps:

- In order to get a smoother evolution, we take always the maximum between the current value and the previous one. We applied this reasoning to every value of the SEIRD model.
- We applied the normalization the each value in order to the sum is equal to N. We did it in this way:

$$Nt = S + E + I + R + D$$

$$S = \frac{N * S}{Nt}$$

$$E = \frac{N * E}{Nt}$$

$$I = \frac{N * I}{Nt}$$

$$R = \frac{N * R}{Nt}$$

$$D = \frac{N * D}{Nt}$$

5.1.2. Script

The Python script consists in the implementation of the following classes:

The class *Model* is class computing a scenario, it take some constants as inputs to compute all the values we need for the SEIRD model and to produce a plot;

The class *FullFactorialDesign* is the class that allow us to compute all the scenarios. Considering the class Model, we implemented some simple methods:

execute() is the principal method of this class, it takes the constant values as inputs and compute all the SEIRD values for each day;

update() computes the SEIRD values for a single day;

normalize() allows us to scale the values in order to their sum is equal to the total population N;

plotAndSave() makes a DataFrame using the pandas library, make a plot from this and create an excel file and a csv one with the data we the model generated.

```python
def execute(self,steps,alpha,gamma,mu,omega,M,num):
    if(self.recommendation):
        omega=0.08
    beta=self.beta
    N=(1-self.containedPercent)*M
    Sa=[N-1]
    Ea=[0]
    Ia=[1]
    Ra=[0]
    Da=[0]
    Ca=[M-(N-1)]
    La=[0]
    for i in range(0,steps):
        S,E,I,R,D,L,C=self.update(Sa[i],Ea[i],Ia[i],Ra[i],Da[i],La[i],Ca[i],N,M,alpha,beta,gamma,mu,omega)
        Sa+=[S]
        Ea+=[E]
        Ia+=[I]
        Ra+=[R]
        Da+=[D]
        La+=[L]
        Ca+=[C]
    self.plotAndSave(steps,Sa,Ea,Ia,Ra,Da,La,Ca,num)
```

Figura 5: Function *execute()*

```python
def plotAndSave(self,steps,Sa,Ea,Ia,Ra,Da,La,Ca,num):
    data=pd.DataFrame()
    data.insert(0,"time",np.arange(0,steps+1,1))
    data.insert(0,"L",np.array(La))
    data.insert(0,"C",np.array(Ca))
    data.insert(0,"D",np.array(Da))
    data.insert(0,"R",np.array(Ra))
    data.insert(0,"I",np.array(Ia))
    data.insert(0,"E",np.array(Ea))
    data.insert(0,"S",np.array(Sa))
    data.plot(x="time",y=["S","E","I","R","D","L","C"],kind="line")
    data.to_excel("./covidsim-"+str(num)+".xlsx")
    data.to_csv("./covidsim-"+str(num)+".csv")
```

Figura 6: Function *plotAndSave()*

```python
def update(self,S,E,I,R,D,L,C,N,M,alpha,beta,gamma,mu,omega):
    Nt=S+E+I+R+D
    print(S,E,I,R,D,Nt)
    dS=-beta*S*I/N
    dE=beta*S*I/N -alpha*E
    dI=alpha*E-gamma*I
    dR=gamma*(1-mu)*I
    dD=gamma*mu*I
    dL=omega*R
    dC=M-S
    S=max(S+dS,S)
    E=max(E+dE,E)
    I=max(I+dI,I)
    R=max(R+dR,R)
    D=max(D+dD,D)
    L=dL
    #C=max(C+dC,0)
    C=dC
    S,E,I,R,D=self.normalize(S, E, I, R, D, N)
    #C=N*C/Nt
    #C=min(C,N)
    return S,E,I,R,D,L,C
```

Figura 7: Function *update()*

The class *FullFactorialDesign* provides two methods:

getCombination() returns a list of dictionary, each one contains the values of the factors for one possible scenario. For simplicity, we fixed the values of the factors as lists of constants, in a generic case we can reuse this class putting these values as parameters.

execute() takes the constants values as inputs and computes iteratively each possible scenario.

```python
class FullFactorialDesign:
    def __init__(self):
        pass

    def getCombinations(self):
        #betas=[0.42*15,0.02*40]
        betas=[1.3,0.3]
        #betas=[0.42*15*0.1]
        recommends=[False,True]
        containedPercents=[0.5,0.0]
        res=[]
        for containedPercent in containedPercents:
            for recommend in recommends:
                for beta in betas:
                    res+=[{"beta":beta,
                           "recommandation":recommend,
                           "containedPercent":containedPercent
                           }]
        return res


    def execute(self,alpha,gamma,mu,omicron,N,steps):
        combs=self.getCombinations()
        for i in range(0,len(combs)):
            comb=combs[i]
            model=covid.Model(comb)
            model.execute(steps,alpha,gamma,mu,omicron,N,i)
```

Figura 8: RS2

5.1.3. Python Simulation

In order to perform the simulation, we used the constant values we explained before and the following values for the factors:

- the transmission rate β is set to 1.3;
- there is no recommendation;
- the percentage of contained people is zero, so all the population is in the system.

With these parameters, we got the following plot:

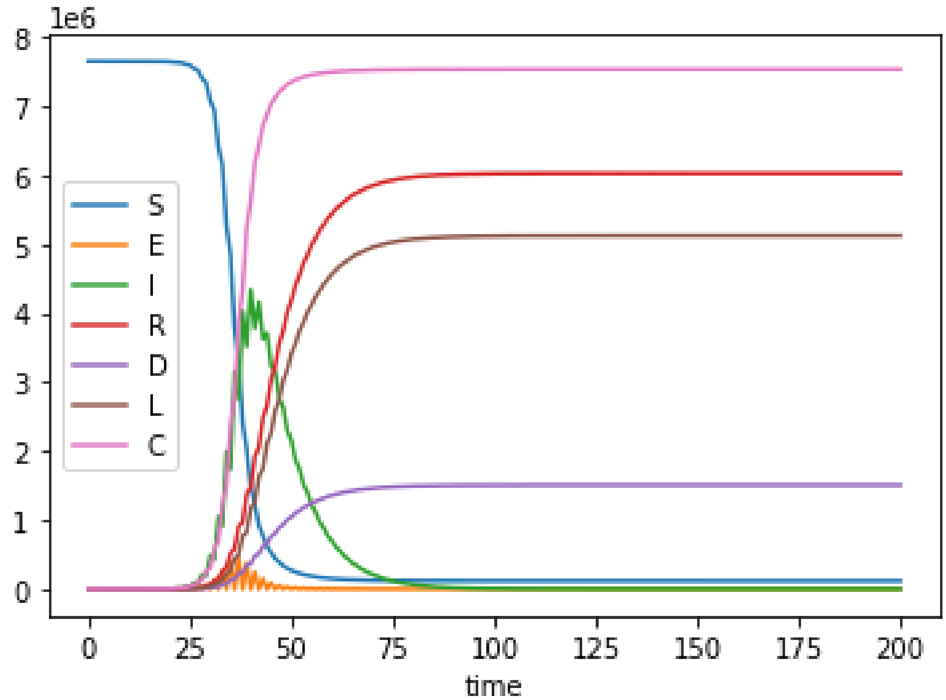


Figura 9: RS2

This model provides better results than the prototype. Indeed, the evolution of the Susceptible entities decreases and concurs with the increasing evolution of the Recovered as well as the Death one. On the other hand, the bells/pics of the Infected and Exposed make sense: their evolution must look like a Gaussian. However we can notice some noise on both evolution, probably due to the approximations done to obtain this simulation. We could model other Covid-19 waves thanks to this script by modifying the input values.

5.2. Insight Maker

5.2.1. Data

With Insight Maker, the formulas used are slightly different from the previous ones.

Indeed, the rates used are usually not in the same magnitude or unity: for example the Recovery rate used for the previous formulas is between 0 and 1; whereas the next formulas rather use a Recovery period measured in days. Here below the formulas used in Insight Maker.

The total number of entities N is defined by:

$$N = S + E + I + R$$

The number of Susceptible entities S is defined by c the percent of contained persons and 7653845 the population in Catalonia such as:

$$S = 7653845 * (1 - c)$$

The process ”Being Exposed”BE from the states Susceptible to Exposed is defined by the Initiation pulse i and the New cases of Infected NC as:

$$BE = i + NC$$

$$\text{where } i = pulse(0, 1, 0{,}25)$$

NC depends on the probability of Meeting a Susceptible person MS and the Transmission rate β, such as:

$$NC = MS * \beta$$

$$\text{where } MS = \tfrac{S}{N}$$

The process ”Becoming infected”BI from the states Exposed to Infected is defined by the number of entities Exposed and the Incubation period α (in days) as:

$$BI = \frac{E}{\alpha}$$

The process ”Fatalities”F from the states Infected to Death is defined by the number of entities Infected, the Recovery period γ (in days) and the Fatality rate μ as:

$$F = \frac{I}{\gamma} * \frac{\mu}{100}$$

On the other hand, the process Recoveries”r from the states Infected to Recovered is defined as:

$$r = \frac{I}{\gamma} * (100 - \frac{\mu}{100})$$

To end with the formulas used in Insight Maker, here is the number of entities encountering second effects despite their recovery L:

$$L = R * l$$

where l is the percent of entities concerned by 2nd effects

5.2.2. Simulation on Insight Maker

On Insight Maker, the simulation is defined by the following schema.

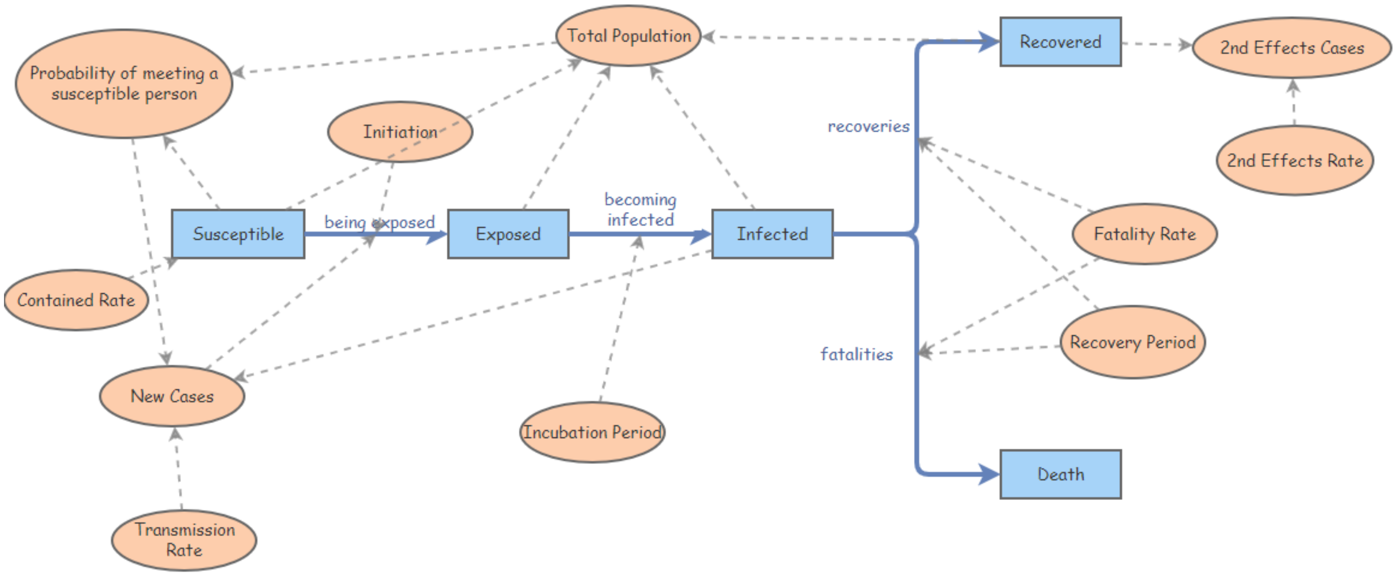


Figura 10: Simulation model on Insight Maker

The model present a specific legend:

- The blue boxes are the different states of the entities.
- The blue arrows are the processes to transfer a certain quantity of entities from one state to an other.
- The pink bubbles are either variables or fixed values.
- The grey dotted arrows show the dependencies between of each element with others.

After a simulation with the following initial values:

- The Transmission rate $\beta = 1{,}23$
- The Incubation period $\alpha = 5{,}5$
- The Recovery period $\gamma = 7{,}76$
- The Fatality rate $\mu = 0{,}85$
- The percent of contained entities $c = 0$
- The percent of entities with 2nd effects $l = 0{,}5$

We obtain a graph like this and can export the corresponding data.

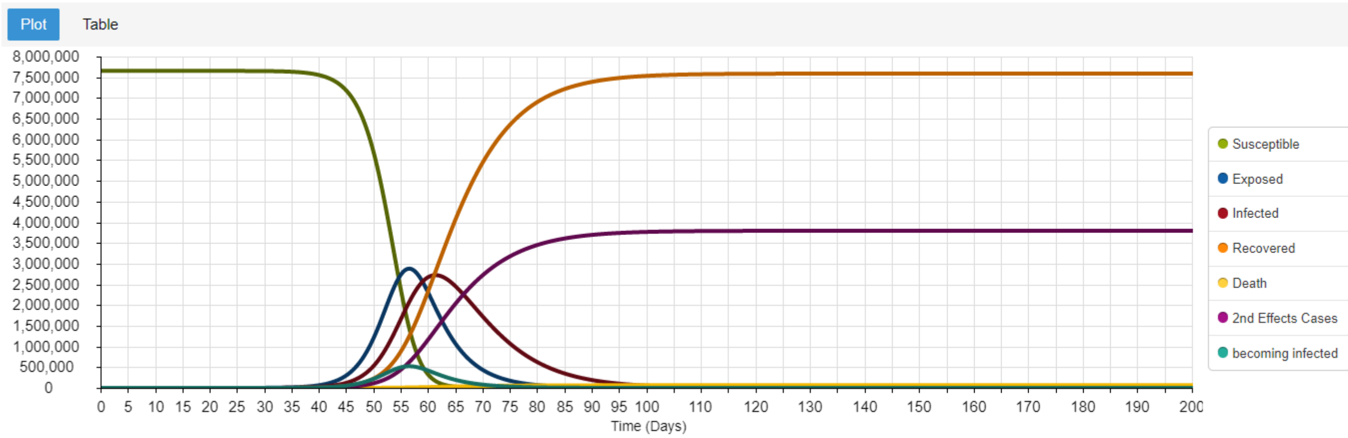


Figura 11: Results of Simulation

This result is very satisfying in comparison to the prototype: every evolution is logical and can become pretty realistic with modifications of some variables. Indeed we ca see a drastic decrease of the number of susceptible entities when the number of infected increases as well as an increasing evolution of the number of recovered entities, that reaches its maximum when the infection wave is over. Note that the simulation step is 0,25 instead of 1 to limit the noise fingered out with Python model.

Moreover, our simulation on Insight Maker allows to modify easily these variables thanks to slider input buttons. Meaning also that we could try to simulate other Covid-19 waves (different period and place) easily with this tool.

6. Definition of the Experimental Framework

6.1. Definition of the DOE

As reminder, the goal of our project, is to transform the pic of new cases into a more smoothed and low evolution. This would give more time to the health system to adapt to the sanitary crisis and potentially lower the number of deaths and people with secondary effects.

In order to do so, we took into consideration two factors that might have important effects on the evolution of infected people per day: the Transmission rate β and the percentage of Contained people c (which is in practice, the proportion of people that is not in the system).

On one hand, at its maximum transmission rate, Covid-19 was at 1.3 and at its lowest, it was at 0.3. On the other hand, a containment at its maximum makes stay at home 65 % of the population, the rest of the population keeps working. Obviously, when there is no containment, 0 % of the population stays at home.

We also aim to lower the number of persons dealing with secondary effects despite their recovery. In our system, this number depends on the number of persons that recovered from

Covid-19. Therefore, it depends on β and c.

However it also depends on l which is the percent of recovered persons that have secondary effects. According to Yang Gao *et al.* in their paper: *"The Short- and Long-Term Clinical, Radiological and Functional Consequences of COVID-19"*; this percent is currently at 85 % after six months if we consider the fatigue effect but can be lowered at 8 % if some recommendations were made (see the table below).

Persistent symptoms	4-12 weeks	6 months	1 year or longer
Patients with at least 1 symptom	41.0-94.0%	37.0-81.0%	45.0-60.9%
Common symptoms			
Fatigue	17.0-84.8%	8.0-85.0%	10.0-60.9%
Dyspnea	5.5-91.5%	11.9-42%	2.7-48%
Chest pain	0.2-42%	0-21.0%	0-15.8%
Cough	2.0-40.3%	2.1-24.0%	3.0-29.0%
Hair loss	13.3-28.6%	2.5-26.3%	11.0-36.2%
Anxiety or depression	4.3-45.1%	17.39-26.7%	3.3-42.0%
Sleep disorder (e.g. insomnia)	3.6-53.6%	5.42-35%	10.7-43.3%
Impaired memory or poor concentration	8.1-41.0% (up to several months)		

Figura 12: Proportion of patients with persistent symptoms at the short and long terms

Moreover, we chose to do a two level design of experiment: each factor is defined by two possible values, known as the negative and positive cases. The three factors and their values are given below (the first value is the negative case, the second one is the positive case):

1. The transmission rate $\beta = [1{,}3; 0{,}3]$,
2. The percentage of Contained people $c = [0; 0{,}65]$
3. The percentage of recovered people with 2nd effects $l = [0{,}85; 0{,}08]$

An other aspect to consider is the number of replications needed. As our model does not contain any randomness, only one replication was needed.

To conclude, with 3 factors and a 2 level design we have $2^3 = 8$ scenarios. This number is low and so, for this reason, we were able to do a full factorial design instead of a fractional one.

6.2. Application of the DOE

We summarized the 8 scenarios to simulate in the following table:

We realised the 8 scenarios on both models. The plots and tables obtained will be given with this document as well as the Excel sheet on which we summarized the results.

6.3. Results of the DOE

In order to compare the results between each other, we considered two indicators:

	beta	c	I
1	-	-	-
2	+	-	-
3	-	+	-
4	+	+	-
5	-	-	+
6	+	-	+
7	-	+	+
8	+	+	+

Figura 13: 8 scenarios

- The maximum number of Infected people per day (RS1).
- The maximum number of Recovered people with 2nd effect per day (RS2).

Thus, the scenarios with the minimum numbers are supposed to be the best scenarios and the ones with the maximum numbers, the worst. Below are the results RS1 with Insight Maker. We can notice that the worst result was obtained with scenario 1 (2723104 infected persons in one day) and the one with the best result with scenario 8 (310431 infected persons in one day).

	beta	C	R	RS1
1	-	-	-	2723104
2	+	-	-	886941
3	-	+	-	952996
4	+	+	-	310431
5	-	-	+	2723104
6	+	-	+	886941
7	-	+	+	952996
8	+	+	+	310431
The max is with 1:				2723104
The min is with 8				310431

Figura 14: RS1

In the case of RS2 (still with Insight Maker), we obtained a similar result: with the scenario 1, 6450261 persons have 2nd effects whereas with the scenario 2 it's only 183182 persons.

	beta	C	R	RS2
1	-	-	-	6450261
2	+	-	-	5560859
3	-	+	-	2257591
4	+	+	-	1946307
5	-	-	+	607083
6	+	-	+	523375
7	-	+	+	212479
8	+	+	+	183182
The max is with 1:				6450261
The min is with 8				183182

Figura 15: RS2

The plots of both scenarios are given below.

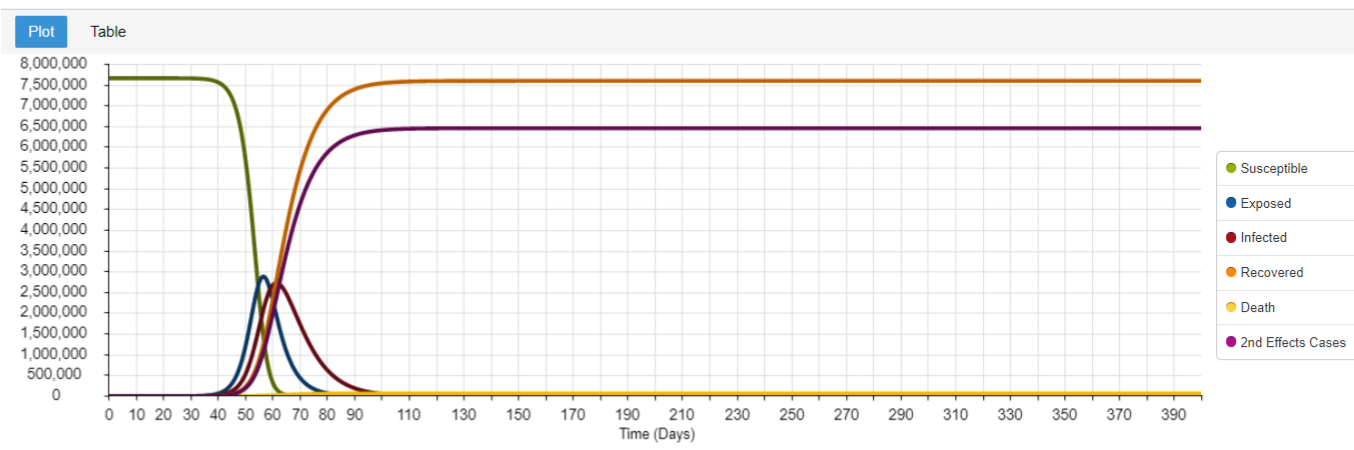


Figura 16: Scenario 1, worst case

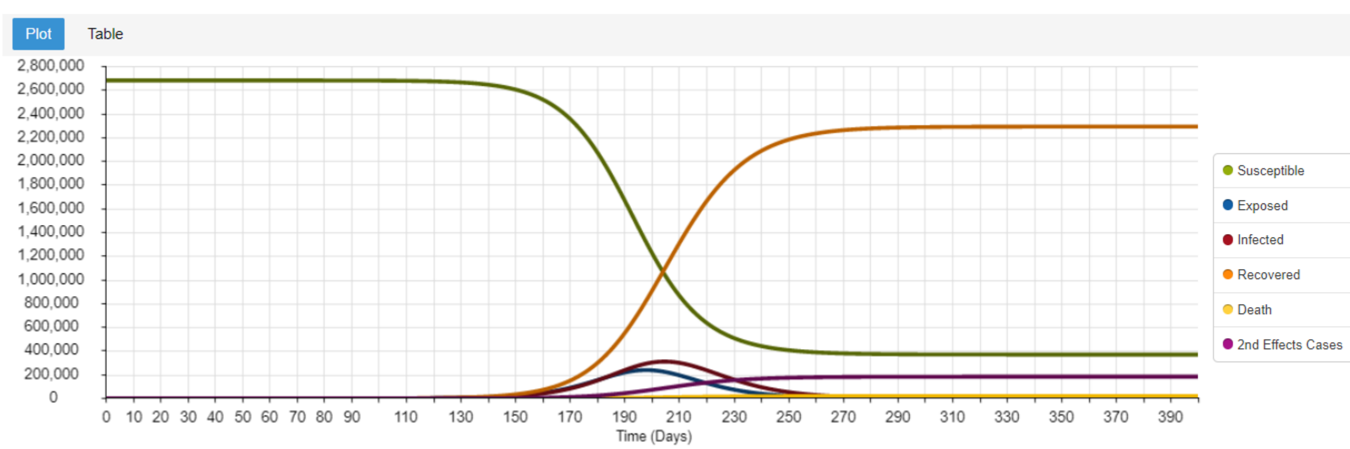


Figura 17: Scenario 8, best case

As you can see, the goals were achieved: the evolution of the number of infected people per day is not a sharp pic anymore in the scenario 8: less people are infected, and this 'smoother pic' arrives later and last on a longer period. Which is a more affordable situation for the health system, giving it time to adapt itself. As for the evolution of recovered people with 2nd effects, the maximum level reached is considerably lowered.

The other plots resulting from the six other scenarios are combinations of the given ones.

6.4. Yates algorithm

The last step of our DOE was to calculate the impact of each factor as well as their interactions on our models.

To do so, we implemented Yates algorithm in Python and ran it. The input values used are the previous maximums (RS1 and RS2) from Insight Maker model.

	beta	c	l	RS1	Effects	
1	-	-	-	2723104	1218368	mean
2	+	-	-	886941	-1239364	beta
3	-	+	-	952996	-1173309	C
4	+	+	-	310431	596799	beta*C
5	-	-	+	2723104	0	R
6	+	-	+	886941	0	beta*R
7	-	+	+	952996	0	C*R
8	+	+	+	310431	0	beta*C*R

Figura 18: Yates results, effects of each factors and their interaction on RS1

As we can see, the average of the maximum of infected people per day (RS1) is of 1218368. We can also notice that both β and c, as well as their interaction, have important impacts on RS1 (effects far from 0). However, the impact of l as well as its interactions with the other factors is equal to 0. Meaning, RS1 does not depends on this factor. This result is rather logical: the number maximum of infected people per day does not depend on the percentage of recovered people with secondary symptoms.

	beta	c	l	RS2	Effects	
1	-	-	-	6450261	2 217 642	mean
2	+	-	-	5560859	-328422,75	beta
3	-	+	-	2257591	-2135504,75	C
4	+	+	-	1946307	158132.25	beta*C
5	-	-	+	607083	-3672224,75	R
6	+	-	+	523375	271920.25	beta*R
7	-	+	+	212479	1768106,25	C*R
8	+	+	+	183182	-120926,75	beta*C*R

Figura 19: Yates results, effects of each factors and their interactions on RS2

The results of Yates algorithm applied on RS2 show that in average, the maximum number of recovered persons with secondary symptoms is of 2217642. As for the impacts of each factor and their interactions on RS2, we can see that there are all important (far from 0). Meaning that RS2 is strongly dependent on β, c, and l.

7. Model Validation

Regarding the validation of the model, we made some visualization tests comparing the plots of our model with the one on Insight Maker: we adjusted the constant values in order to get the best fit!

Moreover we performed the same test to compare the Python model with the InsightMaker one, this allowed us to see the evolution of each formula and possible error caused by the approximation or other factors.

We performed a desk checking to evaluate the containment C and the secondary effects L. In this case our expert was the professor that tells to us if the evolution of these values were similar to the real ones.

Another way to validate the model would be to perform a statistical analysis comparing the real data with the simulated ones, so computing measurements of accuracy like RMSE, R2 and MAE or making some plots (like a boxplot) in order to check it.

Finally we could do the black box test, which is a way to check how the model is similar to the real one. In this way we can observe how the model behaves under the same conditions of the real system, so we compare the outputs in order to say if the results are similar or not.

8. Results and Conclusions

To conclude, the goal of this project was to lower the human impact of Covid-19 First Wave in Catalonia. We specifically focused on two tasks: lowering the pic of infected people per day into a more smoothed evolution, and lowering the number of recovered people that would suffer from secondary symptoms.

In order to do so, we designed from a prototype two simulation models: one in Python and the other with Insight Maker. After some first calibrations of our models, we built a full design of experiment with height scenarios and studied the impacts of the chosen factors on the results. We found a scenario much better than the others that responded to the problems: scenario 8.

To end with, some validation, verification and calibration processes would still need to be applied before publishing our models. We proposed some of them that we found appropriated to our case. We also keep in mind that our two models really simplify the reality of Covid pandemic and that many other parameters would need to be taken into consideration for a realistic model.

9. Annexes

9.1. Scenarios Insight Maker

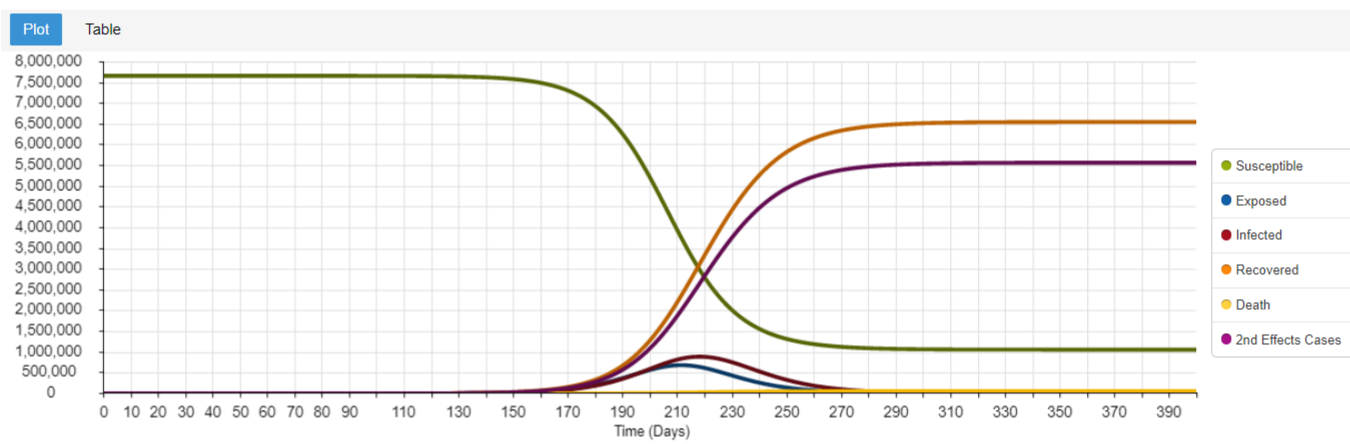


Figura 20: Scenario 2

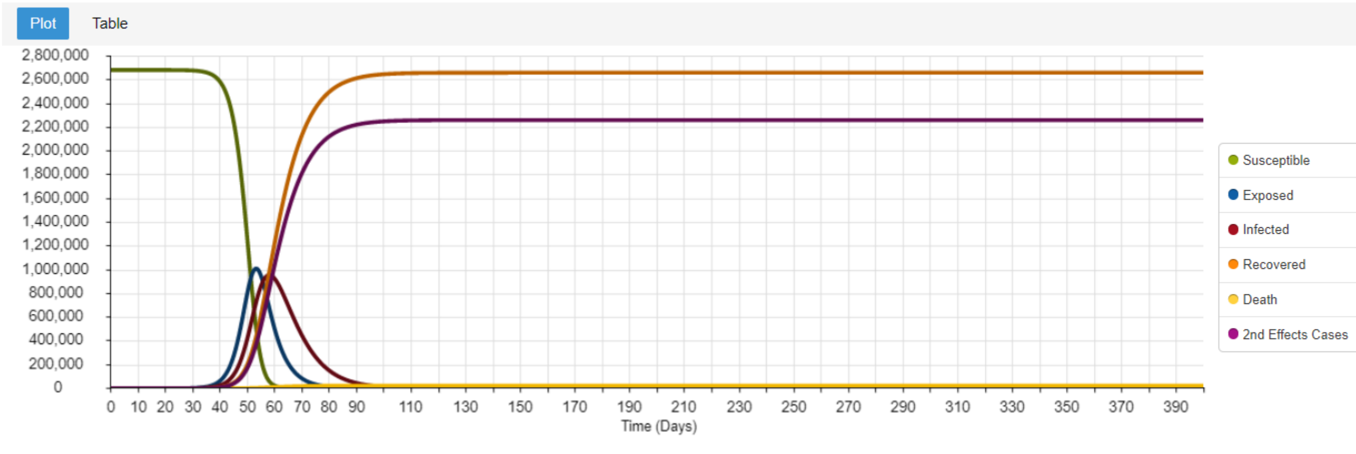


Figura 21: Scenario 3

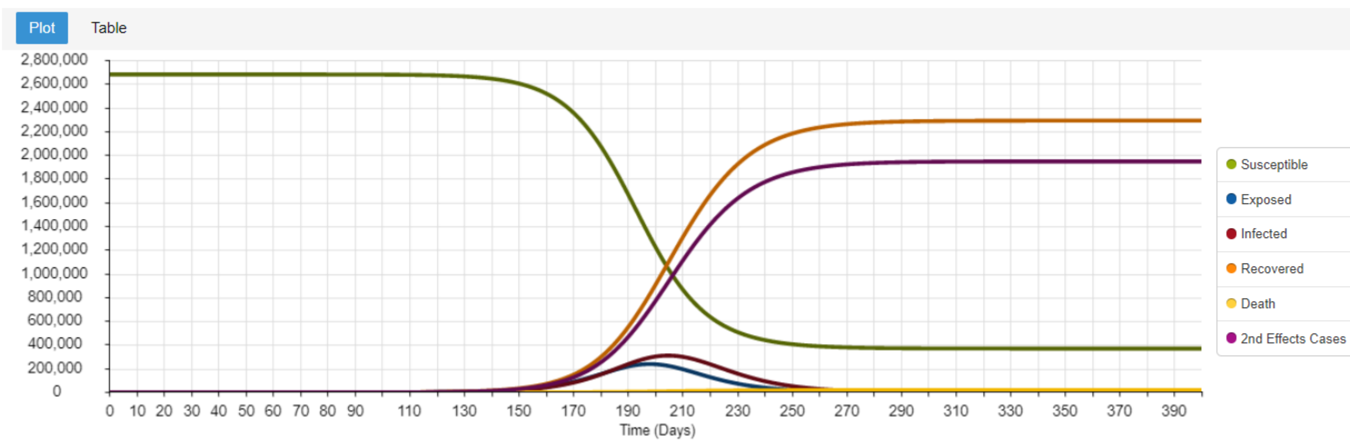


Figura 22: Scenario 4

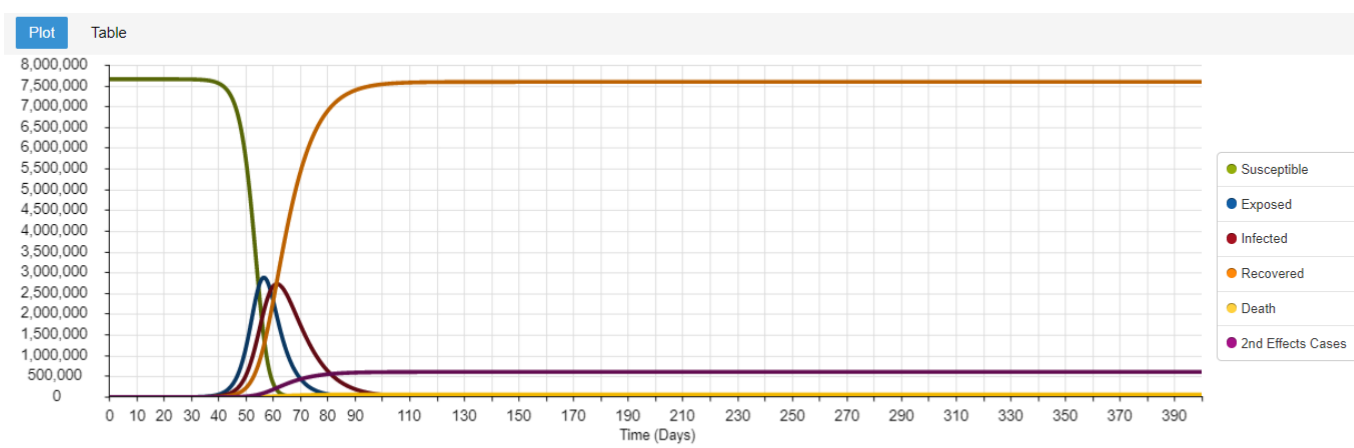


Figura 23: Scenario 5

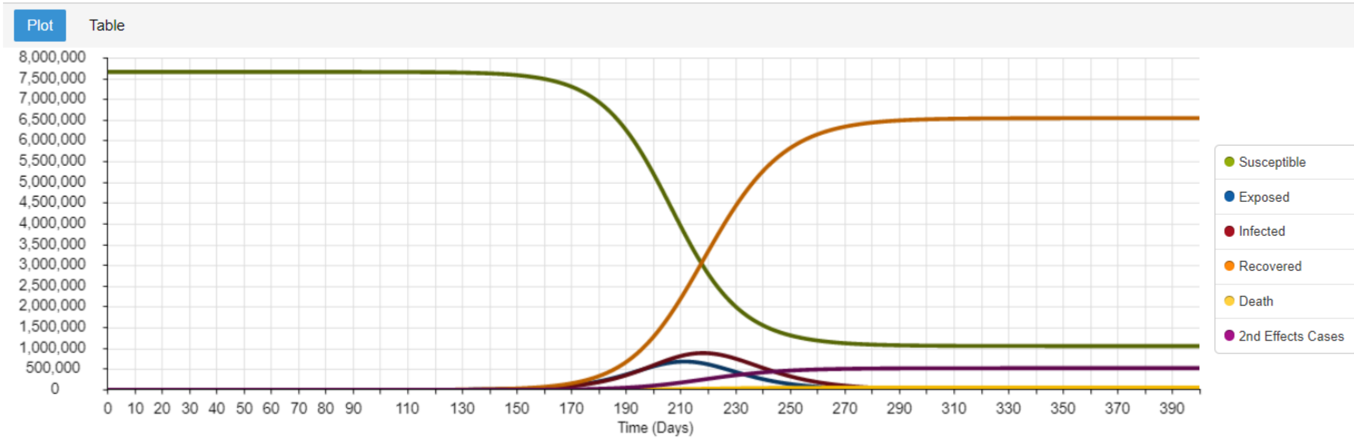


Figura 24: Scenario 6

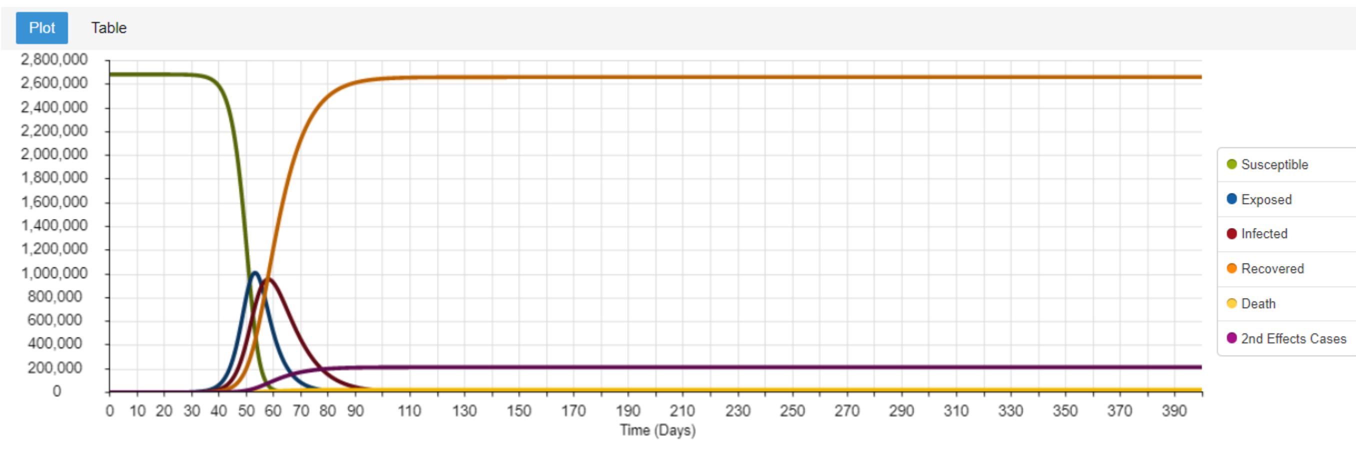
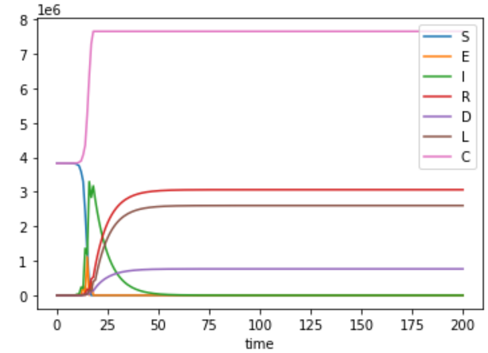


Figura 25: Scenario 7

9.2. Scenarios Python script



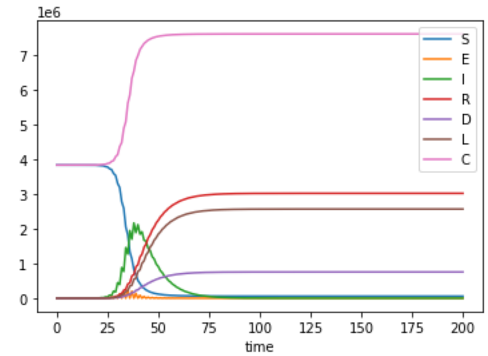


Figura 26: Scenario 2

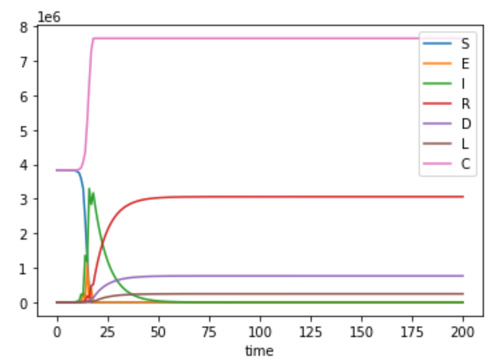


Figura 27: Scenario 3

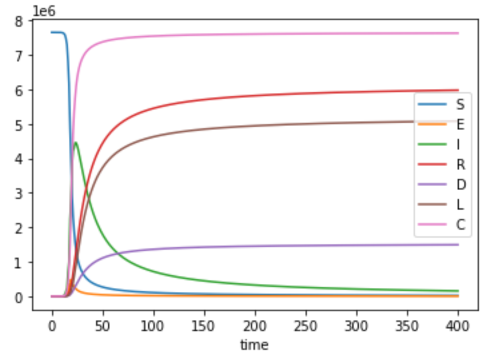


Figura 28: Scenario 4

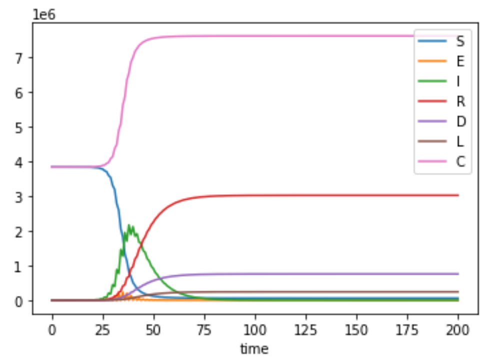


Figura 29: Scenario 5

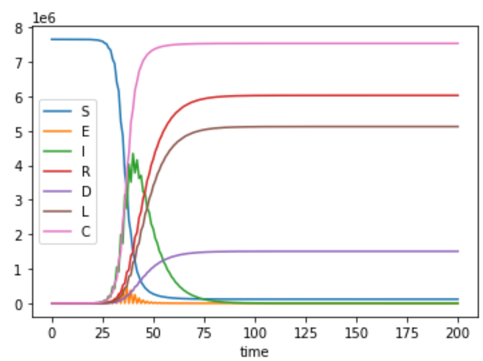


Figura 30: Scenario 6

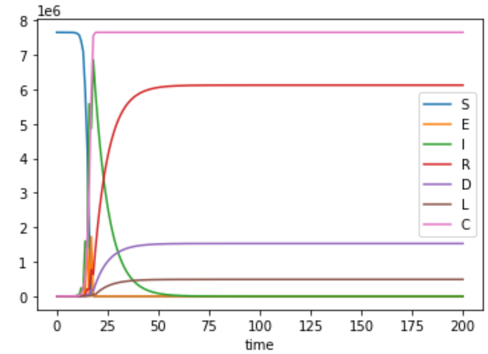


Figura 31: Scenario 7

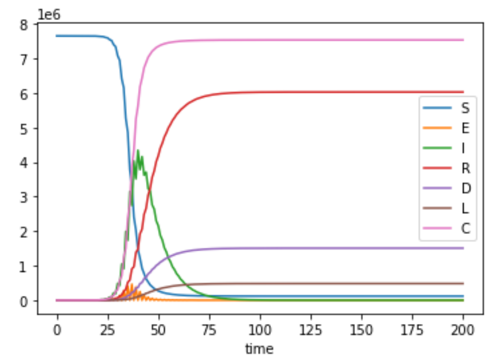


Figura 32: Scenario 8