Modelling Covid-19 for India

Mohammad Abdullah
B.Tech. (Information Technology and Mathematical Innovations)
Cluster Innovation Centre
University of Delhi, India

ABSTRACT

Coronavirus disease 2019 (COVID-19) is an infectious disease caused by coronavirus 2 (SARS-CoV-2), an extreme acute respiratory syndrome. The disease was first reported in Wuhan, the capital of the Hubei province of China, in December 2019 and has since spread worldwide, resulting in the ongoing coronavirus pandemic of 2019–20. The first case of COVID-19 in India was registered on January 30th 2020. In this paper, I propose the mathematical model for analysing and predicting the number of confirmed cases, future trends and reporting ratios of COVID-19. The model on which I will be working is the SEIRD model.This is an extension of the classic SIR model and simply adds two more compartments, the Exposed parameter to show the population that is exposed and the dead parameter to show the number of people who died of coronavirus.The time dependent R_0 values and resource and age dependent fatality rates are also estimated in the paper.

1. INTRODUCTION

The number of new cases of COVID-19 are increasing day by day around the world. We are noticing a huge surge of population being infected and the whole world is struggling to fight this pandemic. Amidst various data information provided by the state and central government and various agencies in India, it has now become imperative that we implement a certain mathematical model, in our case the SEIRD model to predict the number of people susceptible,exposed,infected, recovered and dead at any time after the first case of the novel Coronavirus in India.
The SIR model is a simple model provided by Kermack and Mckendrick. In epidemiology, there are several models that can model the infectious diseases. SIR is one of them.
In this project, I have modelled COVID-19 using the compartmental model which separates the population into several compartments. I tried to understand the background and also the theory behind the models.This disease spreads from one member to another due to coming in contact to one another. This project will try to get the idea of how quickly the virus spreads, what proportion of the people become infected, what proportion of them dies, what proportion of them are in critical condition and die. Also, modelling would help to predict the future graphs and deduce a solution that can help to stop this spread.

2. MODELLING DESCRIPTION

SEIRD MODEL

SEIRD Model stands for Susceptible, Exposed, Infectious, Recovered and Dead respectively. The SEIRD model is based on the SIR model, an addition of the Exposed compartment and Dead compartment makes it the SEIRD model. In this project I add another compartment of Death, as this virus is even causing deaths.
Detailed description of the variables S, I, E, R and D:
S : Individuals that can become hosts when exposed, by catching infection.
E : Individuals in incubation state, these people cannot spread the virus. This is the period before being infectious.
I : Individuals showing the symptoms of the infection, and can transmit the virus to others.
R : Individuals that are recovered, they were infected but not anymore.
D : Individuals that passed away from the disease.

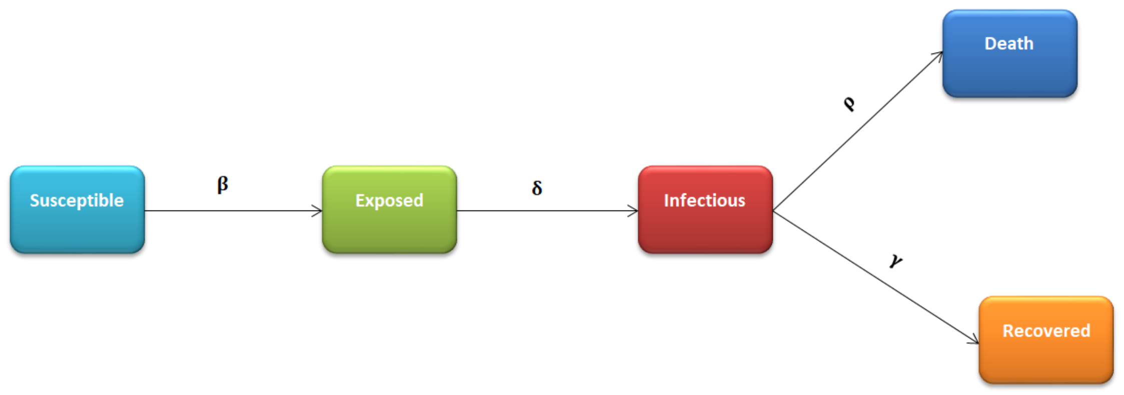


Fig1:SEIRD model with 5 states

The various parameters that is used with this model:
N : Total population of India
D : Number of days an infected person has and during this time can infect others
R_0: Number of people an infected patient infects in total.
α: Fatality rate or probability of dying once infected.
β: Expected number of people infected by an already infected person per day
γ: Proportion of infected patients recovering each day
δ: Length of incubation period
ρ: Rate at which people die or after how many days they die

Rate, Probability and Population are the three parameters to formulate an equation. Thus the equation formulated are:

Change in S(t) in a day is equal to expected amount of people an infected person infects per day(β) * number of people infected (I) * number of people susceptible on day t (S) / total population (N) * 1 (Rate of transition)

$$dS/dt = -\beta * I(t) * S(t)/N \quad ...(1)$$

Change in E(t) in a day is equal to Change in S (dS/dt) - length of incubation period (δ) * number of people exposed (E) *1 (Rate of transition)

$$dE/dt = \beta * I * S/N - \delta * E \quad ...(2)$$

Change in I(t) in a day is equal to length of incubation period (δ) * number of people exposed (E) - rate at which people die (ρ) * fatality rate or the death probability (α) * number of people infected (I) - the proportion of infected recovering per day (γ) * (1 - fatality rate(α)) * number of people infected (I)

$$dI/dt = \delta * E - (1 - \alpha) * \gamma * I - \alpha * \rho * I \quad ...(3)$$

Change in R(t) in a day is equal to the proportion of infected recovering per day (γ) * (1 - fatality rate(α)) * number of people infected (I)

$$dR/dt = \gamma * (1 - \alpha) * I \quad ...(4)$$

Change in D(t) in a day is equal to rate at which people die (ρ) * fatality rate (α) * number of people infected (I)

$$dD/dt = \rho * \alpha * I \quad ...(5)$$

Where,
S(t) : Number of people **susceptible** on day t
E(t) : Number of people **exposed** on day t
I(t) : Number of people **infected** on day t
R(t): Number of people **recovered** on day t
D(t): Number of people **dead** on day t

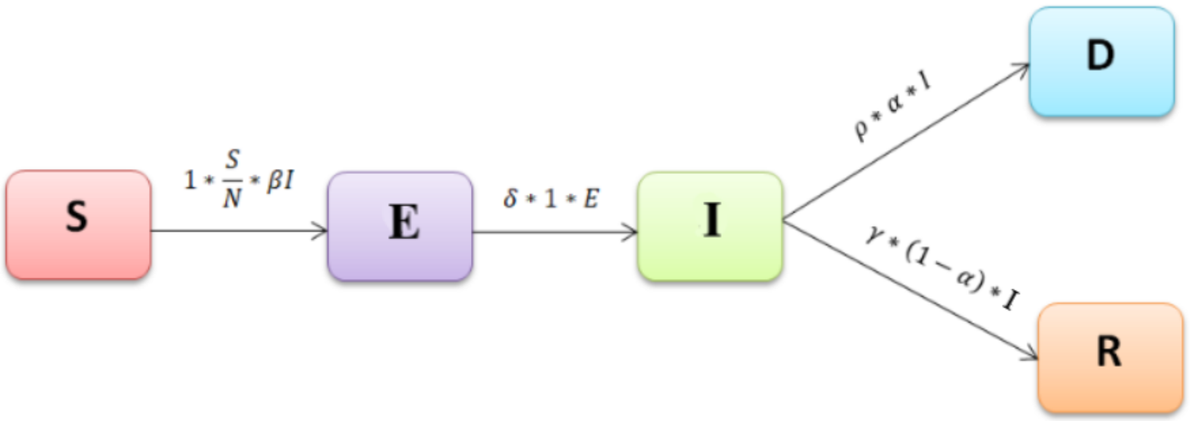


Fig2: Modified SEIR to SEIRD model

These five Equations stated above are used in modelling the COVID-19 Dataset of India.

Transition Values (From one Compartment to Another):

Rate
Probability
Population

From one compartment to the other, the first transition value is the rate at which the transition takes place. The second one tells us the probability of the population of the first compartment going to the next compartment. The third transition value tells us the actual population of the first compartment going to the next compartment.

Reproduction Number(R_0)

The basic reproduction number (R_0) measures transferability rate of the disease. It helps in determining if the infection will remain constant, spread exponentially or die out. Different values of R_0 have different significance.

$R_0>1$: Every person infects more than one individual, on an average. In this case the disease spread.

$R_0<1$: Every person infects less than one individual, on an average. In this case the disease dies out.

R_0=1: Every person infects exactly one individual. In this case the disease becomes epidemic. It will make the disease move throughout the population.

The basic reproduction number (R_0) can be equated as the ratio of rates over time. If an infected person comes in contact with β individuals per unit time, and the disease has an average infection period as 1/γ, then R_0 = β*D or β/γ (as γ = 1/D) . Also, an assumption is considered here, that every individual that comes in contact with an infectious person becomes infected.

Basically answered, R0 is the average number of people an infected person infects in total.

At first, it is kept constant at 2.75 according to the Indian data which you will see below. But as time passes, the R_0 values keep on changing. It is not constant throughout time.Taking into considerations, the nationwide lockdown which is meant to reduce the number of confirmed cases of Coronavirus, it is very important to vary the R0 value. That's what lockdown is meant for!

To make the above R0 more realistic, we can take R_0 as a logistic function,
Logistic growth functions are used to model real-life quantities whose growth levels off because the rate of growth changes—from an increasing growth rate to a decreasing growth rate

$$R_0(t) = ((R_{0\ start} - R_{0\ end}) / (1+e^{-k(-x=x0)})) + R_{0\ end} \quad \ldots(6)$$

Where,
$R_{0\ start}$ denotes the R_0 value on the first day of the infection,
$R_{0\ end}$ denotes the R_0 value on the last day of infection,
x_0 denotes the lockdown date (lockdowns are imposed to decrease the number of individuals exposed to the virus),
and k plays as the deciding factor of logistic growth rate or steepness of the curve i.e how quickly the R_0 declines.

R_0 for INDIA:

Referring to an IMCR study, Union Health Ministry Joint Secretary Lav Agarwal said the current R_0 for the coronavirus Infection in INDIA is somewhere between 1.5 to 4. Therefore, considering the average value that comes out to be 2.75.

- $R_{0\ avg}$ = 2.75
- $R_{0\ start}$ = 4
- $R_{0\ end}$ =1.5
- x_0 = 53 days (the duration after which current lockdown is implemented in INDIA after the first case.)

Death Probability(α):

The fatality rate or death probability is not constant for most of the population. The average death probability in India stands at 3.4% (fitted best from government sources). The death probability depends on various parameters. In this part, we will see how the death probability is dependent on resource and age groups.

Resource as in, not all people having Coronavirus go to the hospital to treat themselves. Moreover,if the number of people infected are more, the probability of the number of people dying is also affected.

Initially, I have modelled our data according to the national average of 3.4% but later on I modified the formula in accordance to resource and age group dependencies.

The death probability is calculated as follows:

$\alpha(t) = (S * I(t)/N) + \alpha_{optimum}$

Where, S is scaling factor and $\alpha_{optimum}$ is optimal fatality rate. Also, S is a resource dependent scale.
S is some arbitrary constant scaling factor that controls how big of an influence the proportion of infected should have.
A scaling factor of 1 and 0.1 (1 is considered appropriate for this situation) is used in this paper.
We can vary the scaling factor according to our needs. If the death rate is weakening then, s could be less and likewise.

Age group is also an important factor to consider. Not all people of all ages die with equal probabilities. Old age dies with more probability. So alpha optimum could be calculated like this:

$\alpha_{opt} = 0.572*0.004 + 0.345*0.019 + 0.08*0.116 + 0.003*0.219 = 0.01878$
Therefore, $\alpha_{optimum} = 1.878$ %

Where the proportion of population is taken and also the probability of different age groups multiplied with them and summed up as shown above.

Proportion of population in India by age group:

Age	Percentage
00-29	57.2%
30-59	34.5%
60-89	8%
89+	0.3%

αor the death probability by age group in India:

Age	α(in percentage)
00-29	0.4%
30-59	1.9%
60-89	11.6%
89+	21.9

This probability differs depending on the age group. These percentages shown above don't add up to 100. The reason behind this is, these percentages do not represent the share of death by the given age groups. Instead, it represents the risk of dying for a person within an age group, if infected by the virus.The average number of days taken by an infected person detected of virus and dies because of it is estimated to be 19 to 20 days.

β: Number of people getting infected per day in India

Expected number of people that are infected by an already infected person is calculated as follows for our model:
There is no need to put the value of βdirectly because it depends on R_0 and γ. And we have already calculated logistic reproduction number and have the value of γ.
Therefore, we can say:
$\beta_{logistic} = R_0 * \gamma$ or $R_{0\ logistic} * \gamma$

δ: Incubation Period in India

The incubation period of the virus is the time between the exposure and the display of symptoms.
The incubation period is the same for every place, it ranges from 1 to 12.5 days. The median estimate is 5-6 days, but it can even be as long as 14 days.

Therefore, δ=6.

ρ: **Rate at which people die in India**

According to government of India data sources, it takes 19-20 days for people to die after once infected with the virus(In case they are to die ,otherwise they can recover also). It takes the period when the person gets critical and then dies.
ρ=1/No.of days people die after infection
In this case the time is 19.5 days(taking the average), so ρ = 1/19.5

γ: **Recovery Rate in India**

The proportion of recoveries per day is 1/14. As the D (Number of days, an infected person infects others) = 14.
Since, γ= 1/D
Therefore, γ=1/14

D : **Infectious Period in India**

The number of days the infected person spreads the virus to healthy people varies from 8 to 14 days, it can also be longer if the person is not quarantined and he/she is still coming in contact with other people. This period starts 1-3 days before the symptoms develop. In our model we have taken D as 14.

N: **Population of India**
The current population of India is 1,387,297,452 which is used to model Covid-19.

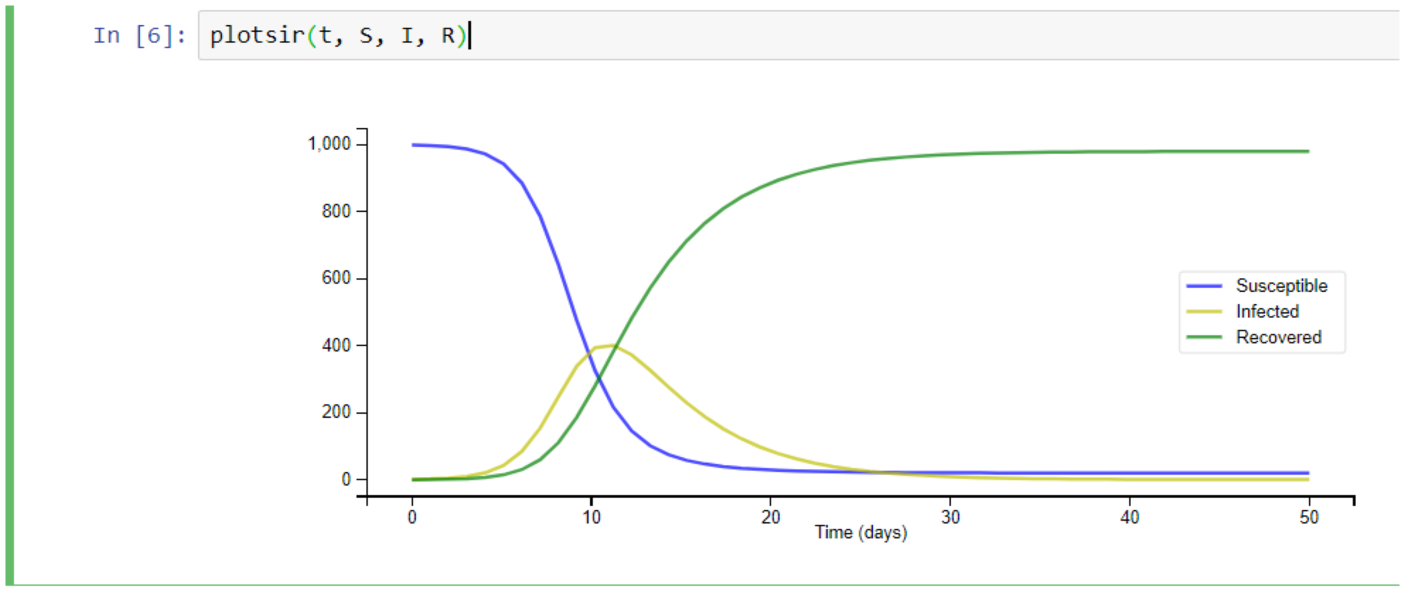
3. DATA ANALYSIS

Our study was based on publicly available data from 30 January 2020. The new verified daily cases reported for India have been taken from the Ministry of Health and Family Welfare India. I tried to approximate the mean values of the major epidemiological parameters,i.e. the basic reproduction number, the case fatality and the case recovery based on the released data. I calibrated the parameters of the SEIRD model to fit the reported data and tried getting the results.

I have considered the following assumptions in this paper:

- The population is divided into several groups,assuming they share the same characteristics within the group.
- No new individual can join the susceptible group because the natality or birth rate has been ignored here and no immigrations of any kind is allowed.
- Next, I assume that if a person has been infected with Covid-19 once, he cannot catch the virus the next time. Either the infected person will jump to the dead or the recovered compartment and not susceptible or exposed.
- Deaths do not change the population to a meaningful extent given the huge population of India.
- R_0 only decreases or remains constant,it does not increase. Hence this implies that we cannot risk loosening preventions such as lockdown or bluntly,other healthcare preventions.
- I am extrapolating from incomplete or preliminary data. So the model might not be exact to the real scenario. Basically, the parameters I have taken are not precise and are from different data sources that have been approximated.
- I presume homogeneity within the population as a whole, that is, I do not find some places to be initial hot spots and others to enforce restrictions sooner and tighter.
- The data can vary with the slightest change in the initial conditions.

3.1 Output of Basic SIR Model in Python (Trial Code)



```
In [6]: plotsir(t, S, I, R)
```

I have taken a population demonstration of count 1000 to test the code for the basic SIR model.
The results were optimistic. I have assumed that each person infects only one person per day. The infection lasts for four days,that is, within this time the person can infect other people with COVID-19. The recovery rate,gamma will be ¼ since (rate of infection is 4 days). One out of 1000 persons is infected and the rest susceptible to the disease!

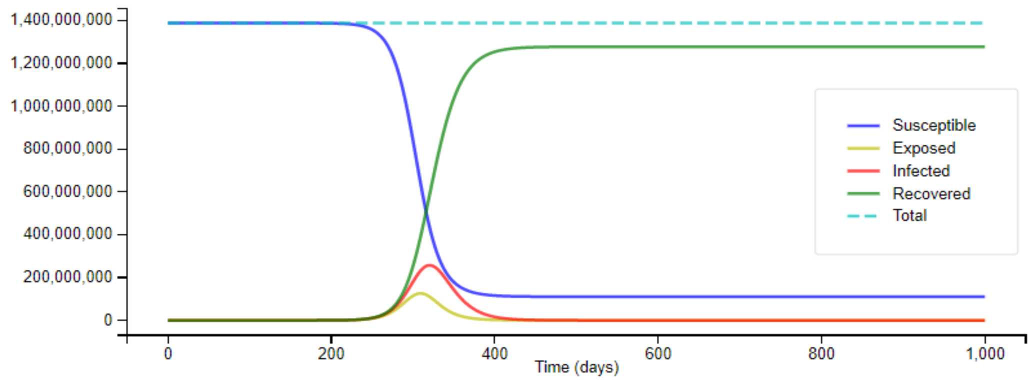
The graph shown above demonstrates the number of Susceptibles,Infected and Recovered at any point of time, that is, after any number of days upto **50 days.**

It must be noticed that at any given time, the sum of susceptibles, infected and recovered is always 1. Since the population is constant.

3.2 Results of Codes in Python for Original Indian Data

Output of The Exposed Compartment

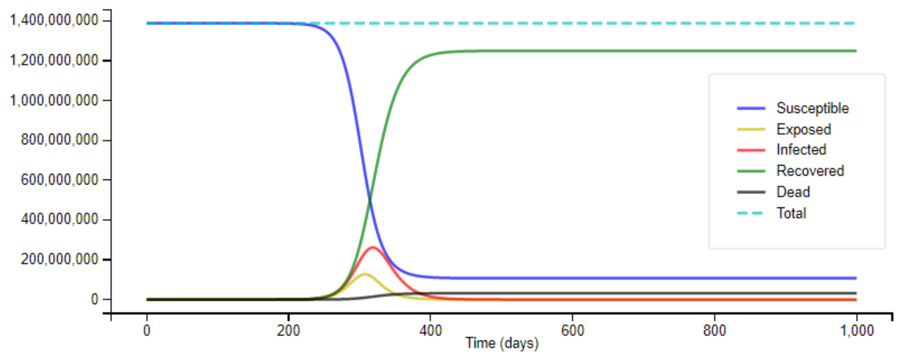
```
In [74]: plotseird(t, S, E, I, R)
```



The graph shows at any given no of days after 30th Jan,2020 the no of Susceptibles, Exposed, Infected and Recovered. At any point of time, if you sum the total of S,E,I,R the values of population will be equal to the total population. The default value of R_0 i.e 2.75 and αi.e 3.4% is taken. The y- axis shows the population of India taken to be 1,387,297,452 in 2020.

Output of the Dead Compartment:

```
In [78]: plotseird(t, S, E, I, R, D)
```



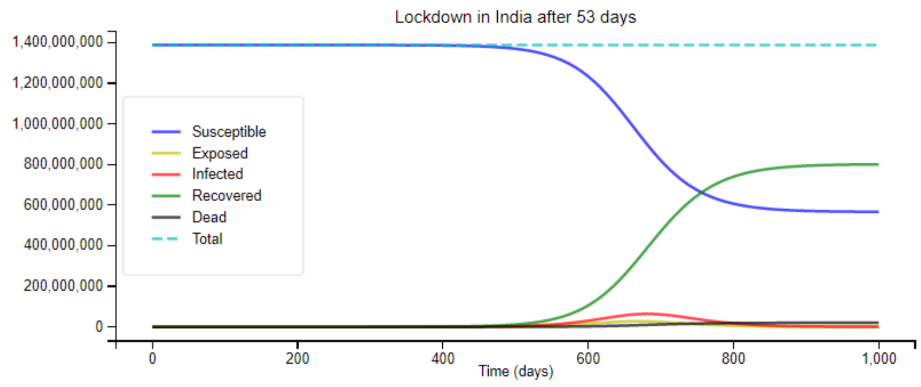
When the dead compartment has been added, the graph looks like this. Note the dead compartment graph line is shown in black.The default value of R_0 i.e 2.75 and αi.e 3.4% is taken. The y- axis shows the population of India taken to be 1,387,297,452 in 2020. The X-axis shows the number of days upto 1000.

Output of Time Dependent R_0:

- Simple Approach: Single Lockdown
- If Lockdown happens, R_0 falls to 1.5
- If lockdown doesn't happen, R_0 is 4

Real Lockdown after 53 days

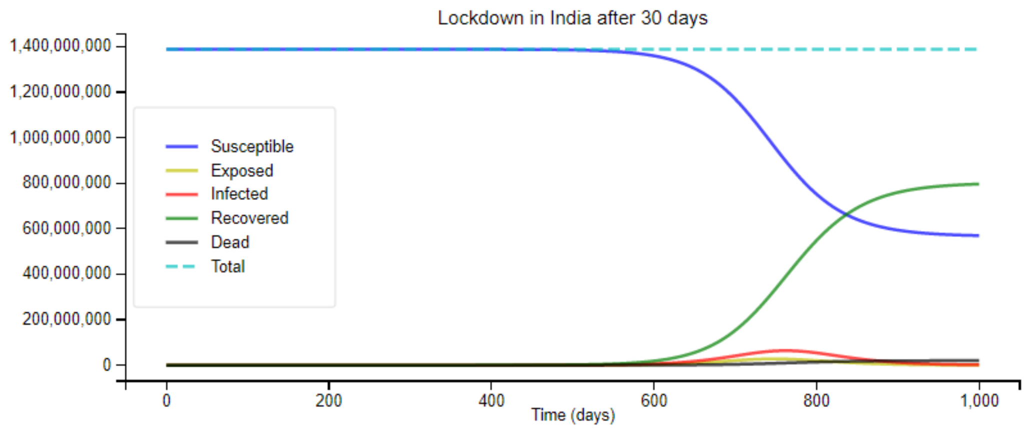
```
In [125]: plotseird(t, S, E, I, R, D, L)
```



Here, the value of R0 changes after lockdown to 1.5 else it is 4.The value of X0 that denotes the duration after which lockdown was implemented i.e 53 days as on 25th March,2020 after 30th January,2020.

Lockdown imagined after 30 days

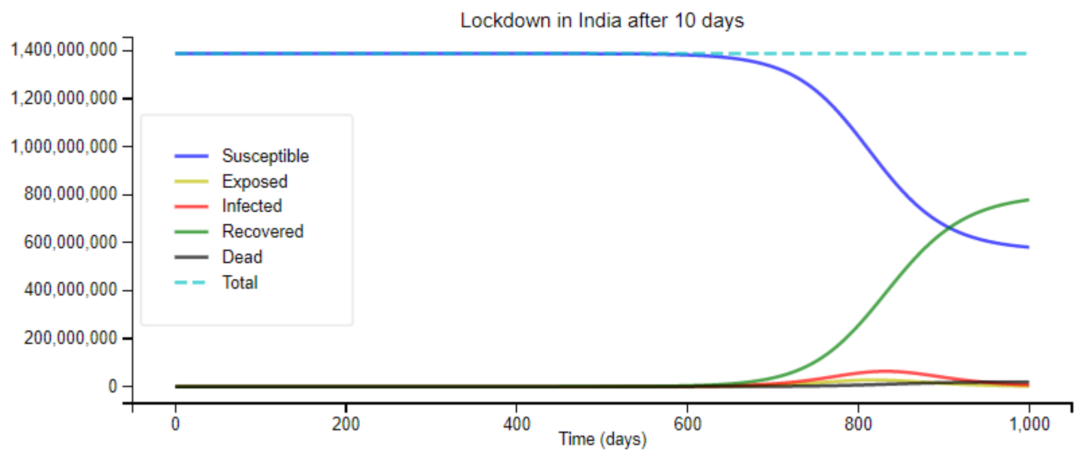
```
In [129]: plotseird(t, S, E, I, R, D, L)
```



This graph is when we suppose that lockdown is implemented after 30 days. Note that upto 600 days the graph is stabilized.

Lockdown imagined after 10 days

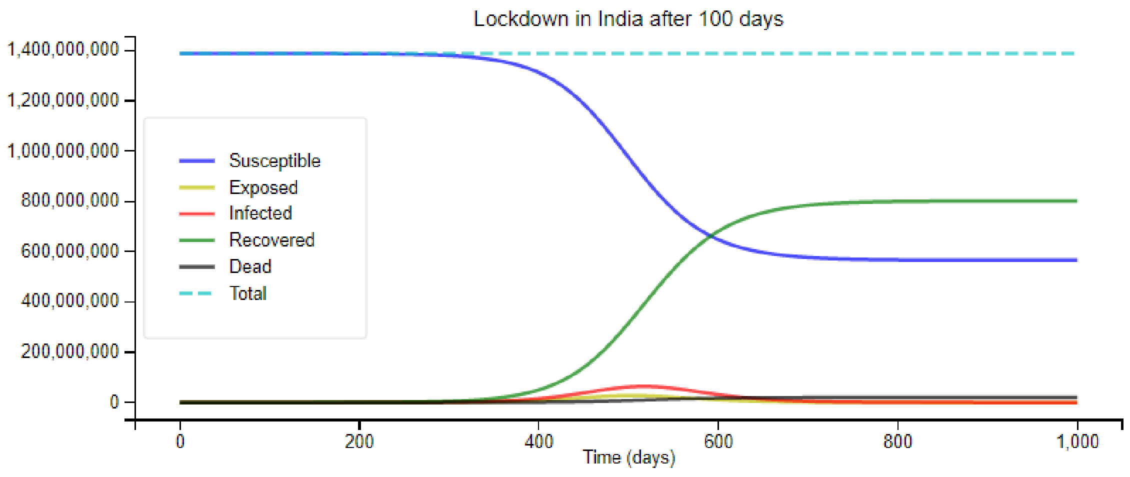
```
In [133]: plotseird(t, S, E, I, R, D, L)
```



This graph is when we suppose that lockdown is implemented after 10 days. Note that upto 700 days the graph is still stable.

Lockdown imagined after 100 days.

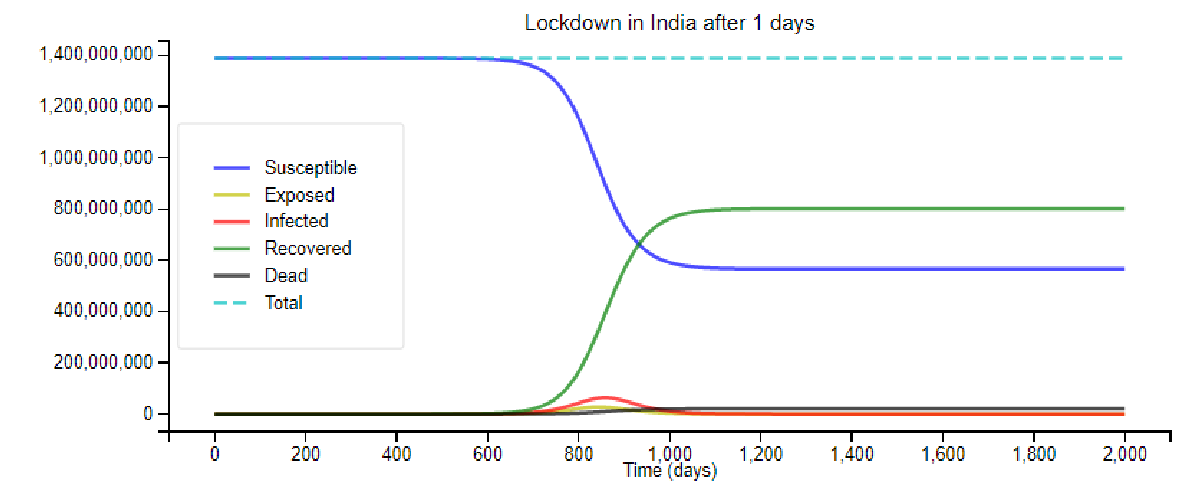
```
In [137]: plotseird(t, S, E, I, R, D, L)
```



This graph is when we suppose that lockdown is implemented after 100 days. Note that upto 350 days the graph is still stable. That means it is the worst scenario among all that the infection starts very early.

Lockdown imagined after 1 day.

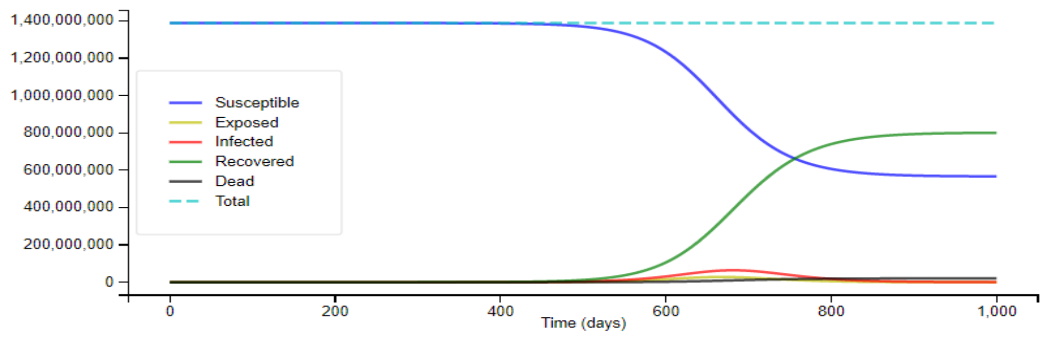
```
In [147]: plotseird(t, S, E, I, R, D, L)
```

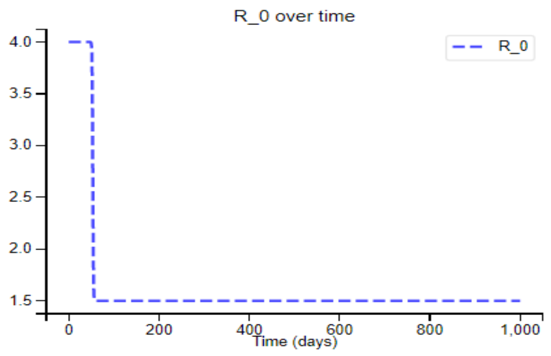


This graph is when we suppose that lockdown is implemented after 1 day. Note that upto 780 days the graph is still stable.

Output of Time Dependent R_0:

- **Advanced Approach: Logistic R_0**



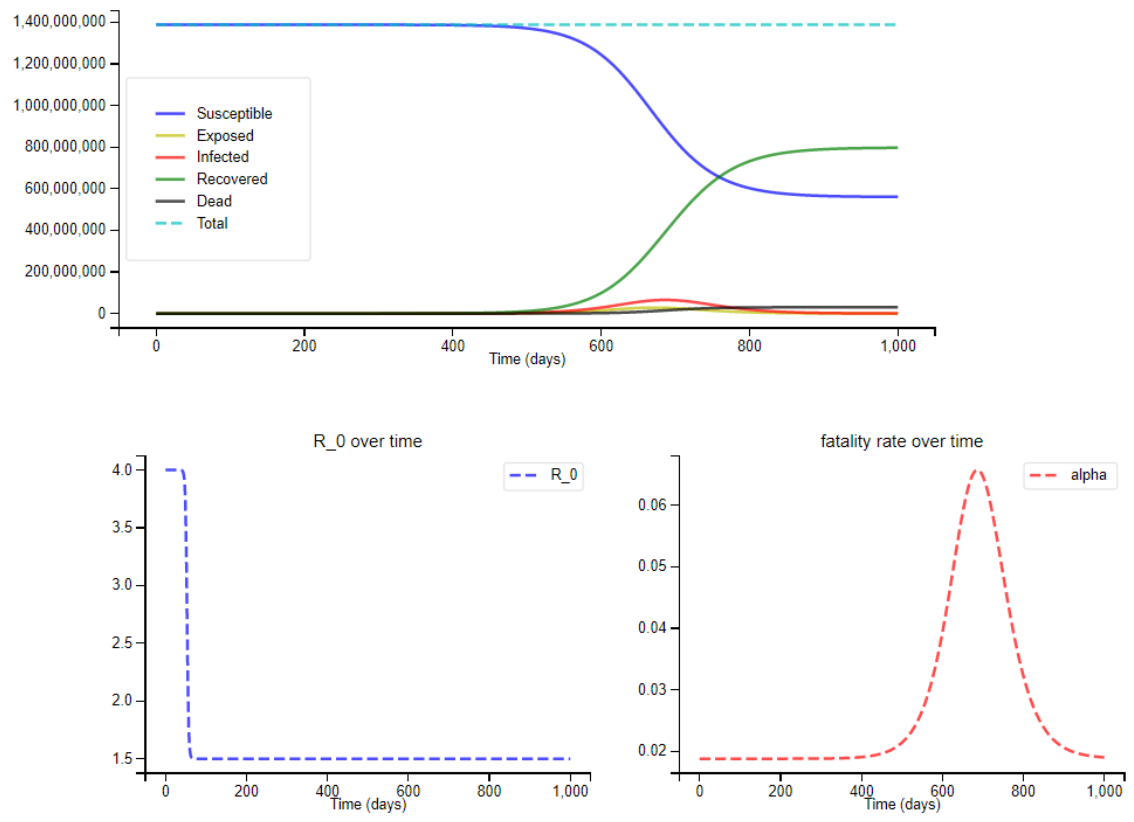


Here, the value of constant R_0 is changed and logistic R_0 is applied. R_0 start and end being 4.0 and 1.5 respectively. The X0 value is the real lockdown value. It means after 53 days in India the lockdown started. This graph shows the S,E,I,R,D after 30th January in no of days after that date for a whooping population of India. Also, the next graph shows how R_0 varies with time.

Output of Resource and Age Dependent Fatality Rate:

- When S=1
- That is, the effect of the infected population is somewhat influential to the fatality rate.

```
In [156]: plotseird(t, S, E, I, R, D, R0=R0_over_time, Alpha=Alpha_over_time)
```



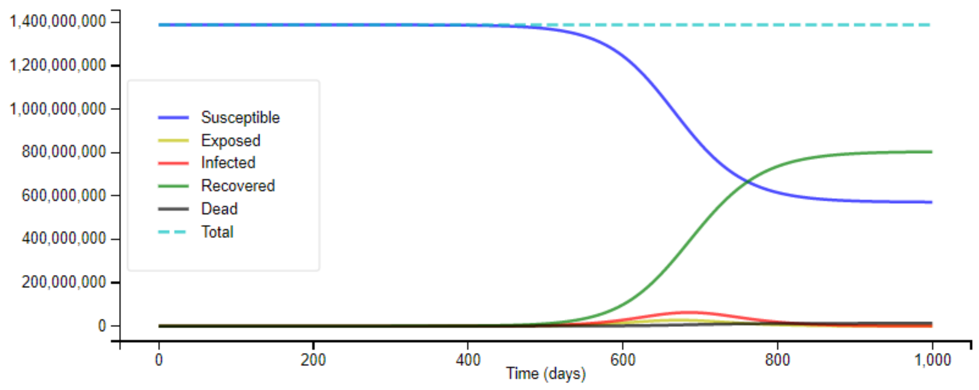
The graph shows the number of S,E,I,R,D in the population with the number of days after the first case of Coronavirus appeared. Now the death probability is not constant at 3.4% but different casualties rate via different age groups is considered along with the resources such as healthcare facilities such as hospital beds, PPE kits,etc on which the scaling factor depends. Here, we have demonstrated that the resources and health care facilities are not good enough in India such that the scaling factor is 1 and thus, a higher death probability.

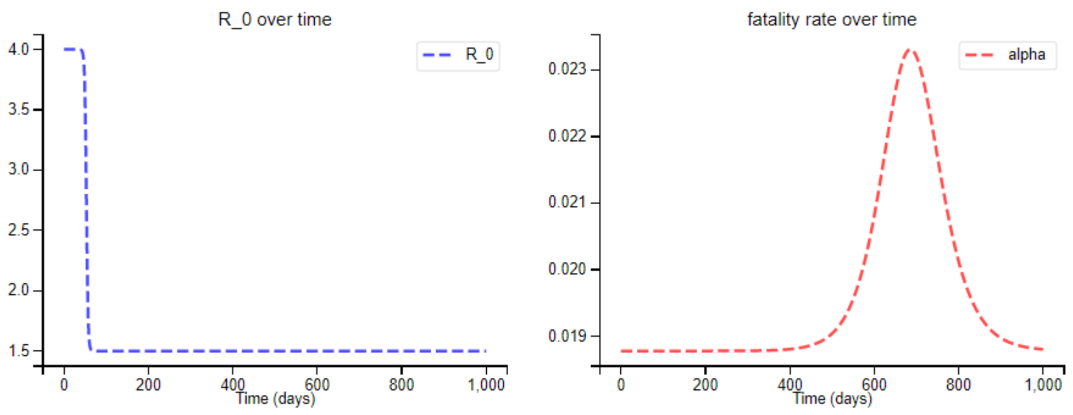
Note the bottom right curve, showing the highest fatality rate of 0.06.

Output of Resource and Age Dependent Mortality Rate:

- When S = 0.1
- That is, the effect of the infected population is not much. Less no.of people are infected.
- The resource dependent scale lowers the death rate (S=0.1)

```
In [160]: plotseird(t, S, E, I, R, D, R0=R0_over_time, Alpha=Alpha_over_time) #s=0.1
```





The graph shows the number of S,E,I,R,D in the population with the number of days after the first case of Coronavirus appeared. Now the death probability is not constant at 3.4% but different casualties rate via different age groups is considered along with the resources such as healthcare facilities such as hospital beds, PPE kits,etc on which the scaling factor depends. Here, we have demonstrated that the resources and health care facilities are good enough in India such that the scaling factor is 0.1 and thus, a higher death probability.

Note the bottom right curve, showing the highest fatality rate of 0.023.

Conclusions

In this paper , I proposed a SEIRD model for the analysis of the COVID - 19 outbreak in India. In our new formulation, the coefficient of infection rate was adaptively modelled as the inverse function of time, taking into account the restrictions imposed by the Indian Government on social life on 25th March 2020. The results obtained by fitting the data of various regions of India , available since 30th January 2020, indicate a very good fit to the data and predict the behaviour of individuals who are infected.But the predictions will accordingly change due to real time change in daily data.However the predictions made using the current epidemiological models in the current work will be invalid if there are reported violation of quarantine norms.

First I tried to implement the basic SIR model to fetch the results which were optimistic. I then added the different compartments and tried fitting it into the model to predict the outcomes. The following modifications are developed and implemented: a 'dead' state for individuals who have passed away from the disease, a 'exposed' state for individuals who have contracted the disease but are not yet infectious, a 'time dependent R_0' values allowing us to model quarantines, lockdowns and 'resource and age dependent fatality rates' that will allow us to model overcrowded hospitals, resources needed and the difference between the age-wise casualties.

The graph shows that after 320 days in India, the graph stabilises that there is not a significant number of confirmed Coronavirus cases when the logistic R_0 and resource and age related fatality rates are not imposed.

But when the logistic reproduction number is implemented,assuming at the end of 1000 days the R0 end value will be 1.5 and when resource and age-wise fatality rates are implemented , then after 550 days the curve stabilises.

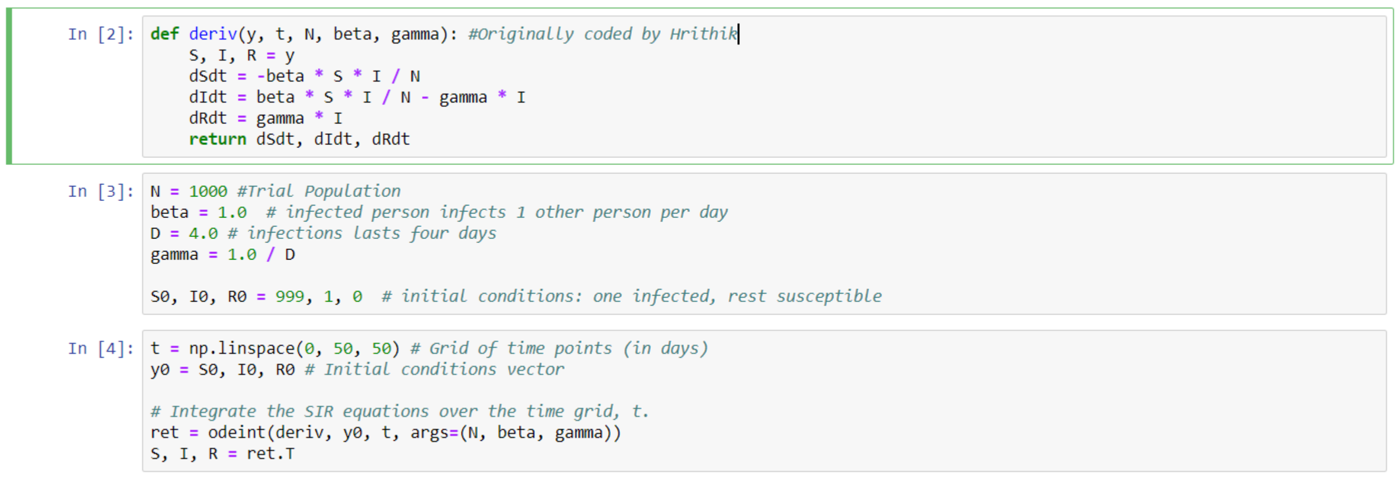
This is just an insight into the Indian data of Coronavirus. The real data might differ from what has been perceived.

References

- Christian Hubbs 2020, *Social Distancing to slow the Coronavirus,* Towards Data Science, Medium, viewed 20 April 2020, <https://towardsdatascience.com/social-distancing-to-slow-the-coronavirus-768292f04296>.
- Plos One 2020, *Data-based analysis, modelling and forecasting of the COVID-19 outbreak,* viewed 22 April 2020*,* <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0230405>.
- Worldometers n.d., *Coronavirus*, viewed 19 April 2020, <https://www.worldometers.info/coronavirus/country/india>.
- Jennifer Ciarochi 2020, *How COVID-19 and Other Infectious Diseases Spread: Mathematical Modeling*, Triple Byte, viewed 12 April 2020, <https://triplebyte.com/blog/modeling-infectious-diseases>.
- E. Cheynet 2020, *Generalized SEIR Epidemic Model (fitting and computation)*, MathWorks, viewed 4 May 2020,<https://in.mathworks.com/matlabcentral/fileexchange/74545-generalized-seir-epidemic-model-fitting-and-computation>.
- Ed Fontes 2020, *Modeling the Spread of COVID-19 with COMSOL Multiphysics®*, Comsol, viewed 25 April 2020, <https://www.comsol.com/blogs/modeling-the-spread-of-covid-19-with-comsol-multiphysics>
- Theory meets Practice 2020, *Flatten the COVID-19 curve*, viewed 27 March 2020, <https://staff.math.su.se/hoehle/blog/2020/03/16/flatteningthecurve.html>

Appendix

Basic SIR Model Trial Code:



```
In [2]: def deriv(y, t, N, beta, gamma): #Originally coded by Hrithik
            S, I, R = y
            dSdt = -beta * S * I / N
            dIdt = beta * S * I / N - gamma * I
            dRdt = gamma * I
            return dSdt, dIdt, dRdt
```

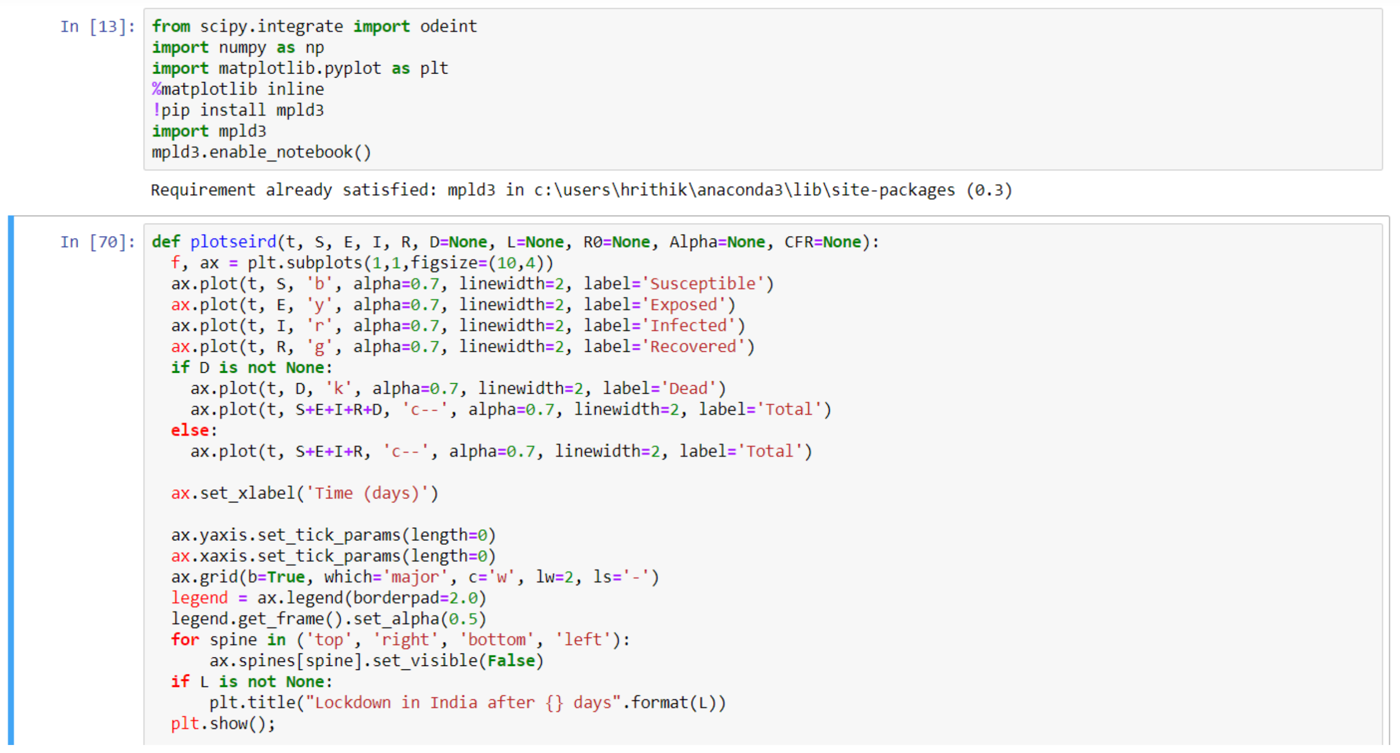
```
In [3]: N = 1000 #Trial Population
        beta = 1.0  # infected person infects 1 other person per day
        D = 4.0 # infections lasts four days
        gamma = 1.0 / D

        S0, I0, R0 = 999, 1, 0  # initial conditions: one infected, rest susceptible
```

```
In [4]: t = np.linspace(0, 50, 50) # Grid of time points (in days)
        y0 = S0, I0, R0 # Initial conditions vector

        # Integrate the SIR equations over the time grid, t.
        ret = odeint(deriv, y0, t, args=(N, beta, gamma))
        S, I, R = ret.T
```

Implementation of SEIRD Model:



```
In [13]: from scipy.integrate import odeint
         import numpy as np
         import matplotlib.pyplot as plt
         %matplotlib inline
         !pip install mpld3
         import mpld3
         mpld3.enable_notebook()

         Requirement already satisfied: mpld3 in c:\users\hrithik\anaconda3\lib\site-packages (0.3)
```

```
In [70]: def plotseird(t, S, E, I, R, D=None, L=None, R0=None, Alpha=None, CFR=None):
           f, ax = plt.subplots(1,1,figsize=(10,4))
           ax.plot(t, S, 'b', alpha=0.7, linewidth=2, label='Susceptible')
           ax.plot(t, E, 'y', alpha=0.7, linewidth=2, label='Exposed')
           ax.plot(t, I, 'r', alpha=0.7, linewidth=2, label='Infected')
           ax.plot(t, R, 'g', alpha=0.7, linewidth=2, label='Recovered')
           if D is not None:
             ax.plot(t, D, 'k', alpha=0.7, linewidth=2, label='Dead')
             ax.plot(t, S+E+I+R+D, 'c--', alpha=0.7, linewidth=2, label='Total')
           else:
             ax.plot(t, S+E+I+R, 'c--', alpha=0.7, linewidth=2, label='Total')

           ax.set_xlabel('Time (days)')

           ax.yaxis.set_tick_params(length=0)
           ax.xaxis.set_tick_params(length=0)
           ax.grid(b=True, which='major', c='w', lw=2, ls='-')
           legend = ax.legend(borderpad=2.0)
           legend.get_frame().set_alpha(0.5)
           for spine in ('top', 'right', 'bottom', 'left'):
               ax.spines[spine].set_visible(False)
           if L is not None:
               plt.title("Lockdown in India after {} days".format(L))
           plt.show();
```

```python
if R0 is not None:
  # sp1
  ax1 = f.add_subplot(121)
  ax1.plot(t, R0, 'b--', alpha=0.7, linewidth=2, label='R_0')

  ax1.set_xlabel('Time (days)')
  ax1.title.set_text('R_0 over time')
  # ax.set_ylabel('Number (1000s)')
  # ax.set_ylim(0,1.2)
  ax1.yaxis.set_tick_params(length=0)
  ax1.xaxis.set_tick_params(length=0)
  ax1.grid(b=True, which='major', c='w', lw=2, ls='-')
  legend = ax1.legend()
  legend.get_frame().set_alpha(0.5)
  for spine in ('top', 'right', 'bottom', 'left'):
    ax.spines[spine].set_visible(False)

if Alpha is not None:
  # sp2
  ax2 = f.add_subplot(122)
  ax2.plot(t, Alpha, 'r--', alpha=0.7, linewidth=2, label='alpha')

  ax2.set_xlabel('Time (days)')
  ax2.title.set_text('fatality rate over time')
  # ax.set_ylabel('Number (1000s)')
  # ax.set_ylim(0,1.2)
  ax2.yaxis.set_tick_params(length=0)
  ax2.xaxis.set_tick_params(length=0)
  ax2.grid(b=True, which='major', c='w', lw=2, ls='-')
  legend = ax2.legend()
  legend.get_frame().set_alpha(0.5)
  for spine in ('top', 'right', 'bottom', 'left'):
    ax.spines[spine].set_visible(False)

  plt.show();
```

Programming the Exposed-Compartment

```
In [71]: def deriv(y, t, N, beta, gamma, delta): #Originally coded by Hrithik
             S, E, I, R = y
             dSdt = -beta * S * I / N
             dEdt = beta * S * I / N - delta * E
             dIdt = delta * E - gamma * I
             dRdt = gamma * I
             return dSdt, dEdt, dIdt, dRdt
```

```
In [72]: N = 1387297452 #Population of India as of 2020
         D = 14.0 # infections lasts fourteen days
         gamma = 1.0 / D
         delta = 1.0 / 6.0  # incubation period of six days
         R_0 = 2.75
         beta = R_0 * gamma  # R_0 = beta / gamma, so beta = R_0 * gamma
         S0, E0, I0, R0 = N-1, 1, 0, 0  # initial conditions: one exposed
```

```
In [73]: t = np.linspace(0, 1000, 1000) # Grid of time points (in days)
         y0 = S0, E0, I0, R0 # Initial conditions vector

         # Integrate the SIR equations over the time grid, t.
         ret = odeint(deriv, y0, t, args=(N, beta, gamma, delta))
         S, E, I, R = ret.T
```

```
In [74]: plotseird(t, S, E, I, R)
```

Programming the Dead-Compartment

```
In [75]: def deriv(y, t, N, beta, gamma, delta, alpha, rho): #Originally coded by Hrithik
             S, E, I, R, D = y
             dSdt = -beta * S * I / N
             dEdt = beta * S * I / N - delta * E
             dIdt = delta * E - (1 - alpha) * gamma * I - alpha * rho * I
             dRdt = (1 - alpha) * gamma * I
             dDdt = alpha * rho * I
             return dSdt, dEdt, dIdt, dRdt, dDdt
```

```
In [76]: N = 1387297452 #Population of India 2020
         D = 14.0 # infections lasts fourteen days
         gamma = 1.0 / D
         delta = 1.0 / 6.0  # incubation period of six days
         R_0 = 2.75   #Reproduction Number
         beta = R_0 * gamma  # R_0 = beta / gamma, so beta = R_0 * gamma
         alpha = 0.034  # 3.4% approx death rate
         rho = 1/19.5  # 19.5 days from infection until death
         S0, E0, I0, R0, D0 = N-1, 1, 0, 0, 0  # initial conditions: one exposed
```

```
In [77]: t = np.linspace(0, 1000, 1000) # Grid of time points (in days)
         y0 = S0, E0, I0, R0, D0 # Initial conditions vector

         # Integrate the SIR equations over the time grid, t.
         ret = odeint(deriv, y0, t, args=(N, beta, gamma, delta, alpha, rho))
         S, E, I, R, D = ret.T
```

```
In [78]: plotseird(t, S, E, I, R, D)
```

Time Dependent R_0:

Simple Approach: Single Lockdown

In [79]:
```
def deriv(y, t, N, beta, gamma, delta, alpha, rho): #Originally coded by Hrithik
    S, E, I, R, D = y
    dSdt = -beta(t) * S * I / N
    dEdt = beta(t) * S * I / N - delta * E
    dIdt = delta * E - (1 - alpha) * gamma * I - alpha * rho * I
    dRdt = (1 - alpha) * gamma * I
    dDdt = alpha * rho * I
    return dSdt, dEdt, dIdt, dRdt, dDdt
```

In [82]:
```
L = 53 #Lockdown after 53 days from the first case of 30th Jan,that is, on 25th March,2020
```

In [83]:
```
N = 1387297452 #Population of India 2020
D = 14.0 # infections lasts fourteen days
gamma = 1.0 / D
delta = 1.0 / 6.0  # incubation period of six days
def R_0(t):
    return 4.0 if t < L else 1.5 #If lockdown happens, then infecting rate becomes lowest i.e 1.5 in India
def beta(t):
    return R_0(t) * gamma

alpha = 0.034  # 3.4% death rate or death probability or risk of dying
rho = 1/19.5  # 19.5 days from infection until death
S0, E0, I0, R0, D0 = N-1, 1, 0, 0, 0  # initial conditions: one exposed
```

In [86]:
```
t = np.linspace(0, 1000, 1000) # Grid of time points (in days)
y0 = S0, E0, I0, R0, D0 # Initial conditions vector

# Integrate the SIR equations over the time grid, t.
ret = odeint(deriv, y0, t, args=(N, beta, gamma, delta, alpha, rho))
S, E, I, R, D = ret.T
```

Advanced Approach: logistic R_0

In [148]:
```
def deriv(y, t, N, beta, gamma, delta, alpha, rho): #Originally coded by Hrithik
    S, E, I, R, D = y
    dSdt = -beta(t) * S * I / N
    dEdt = beta(t) * S * I / N - delta * E
    dIdt = delta * E - (1 - alpha) * gamma * I - alpha * rho * I
    dRdt = (1 - alpha) * gamma * I
    dDdt = alpha * rho * I
    return dSdt, dEdt, dIdt, dRdt, dDdt
```

In [149]:
```
N = 1387297452 #Population of India 2020
D = 14.0 # infections lasts fourteen days
gamma = 1.0 / D
delta = 1.0 / 6.0  # incubation period of six days

R_0_start, k, x0, R_0_end = 4.0, 1.5, 53, 1.5

def logistic_R_0(t):
    return (R_0_start-R_0_end) / (1 + np.exp(-k*(-t+x0))) + R_0_end

def beta(t):
    return logistic_R_0(t) * gamma

alpha = 0.034  # 3.4% death rate or death probability
rho = 1/19.5  # 19.5 days from infection until death
S0, E0, I0, R0, D0 = N-1, 1, 0, 0, 0  # initial conditions: one exposed
```

In [150]:
```
t = np.linspace(0, 1000, 1000) # Grid of time points (in days)
y0 = S0, E0, I0, R0, D0 # Initial conditions vector

# Integrate the SIR equations over the time grid, t.
ret = odeint(deriv, y0, t, args=(N, beta, gamma, delta, alpha, rho))
S, E, I, R, D = ret.T
R0_over_time = [logistic_R_0(i) for i in range(len(t))]  # to plot R_0 over time: get function values
```

In [151]:
```
plotseird(t, S, E, I, R, D, R0=R0_over_time)
```

Resource- and Age-Dependent Fatality Rate

In [153]:
```
def deriv(y, t, N, beta, gamma, delta, alpha_opt, rho): #Originally coded by Hrithik
    S, E, I, R, D = y
    def alpha(t):
        return s * I/N + alpha_opt

    dSdt = -beta(t) * S * I / N
    dEdt = beta(t) * S * I / N - delta * E
    dIdt = delta * E - (1 - alpha(t)) * gamma * I - alpha(t) * rho * I
    dRdt = (1 - alpha(t)) * gamma * I
    dDdt = alpha(t) * rho * I
    return dSdt, dEdt, dIdt, dRdt, dDdt
```

In [154]:
```
N = 1387297452 #Population of India 2020
D = 14.0 # infections lasts fourteen days
gamma = 1.0 / D
delta = 1.0 / 6.0  # incubation period of six days

R_0_start, k, x0, R_0_end = 4.0, 0.5, 53, 1.5

def logistic_R_0(t):
    return (R_0_start-R_0_end) / (1 + np.exp(-k*(-t+x0))) + R_0_end

def beta(t):
    return logistic_R_0(t) * gamma

alpha_by_agegroup = {"0-29": 0.004, "30-59": 0.019, "60-89": 0.116, "89+": 0.219}
proportion_of_agegroup = {"0-29": 0.572, "30-59": 0.345, "60-89": 0.08, "89+": 0.003}
s = 1
alpha_opt = sum(alpha_by_agegroup[i] * proportion_of_agegroup[i] for i in list(alpha_by_agegroup.keys()))

rho = 1/19.5  # 19.5 days from infection until death
S0, E0, I0, R0, D0 = N-1, 1, 0, 0, 0  # initial conditions: one exposed
```

In [154]:
```
N = 1387297452 #Population of India 2020
D = 14.0 # infections lasts fourteen days
gamma = 1.0 / D
delta = 1.0 / 6.0  # incubation period of six days

R_0_start, k, x0, R_0_end = 4.0, 0.5, 53, 1.5

def logistic_R_0(t):
    return (R_0_start-R_0_end) / (1 + np.exp(-k*(-t+x0))) + R_0_end

def beta(t):
    return logistic_R_0(t) * gamma

alpha_by_agegroup = {"0-29": 0.004, "30-59": 0.019, "60-89": 0.116, "89+": 0.219}
proportion_of_agegroup = {"0-29": 0.572, "30-59": 0.345, "60-89": 0.08, "89+": 0.003}
s = 1
alpha_opt = sum(alpha_by_agegroup[i] * proportion_of_agegroup[i] for i in list(alpha_by_agegroup.keys()))

rho = 1/19.5  # 19.5 days from infection until death
S0, E0, I0, R0, D0 = N-1, 1, 0, 0, 0  # initial conditions: one exposed
```

In [155]:
```
t = np.linspace(0, 1000, 1000) # Grid of time points (in days)
y0 = S0, E0, I0, R0, D0 # Initial conditions vector

# Integrate the SIR equations over the time grid, t.
ret = odeint(deriv, y0, t, args=(N, beta, gamma, delta, alpha_opt, rho))
S, E, I, R, D = ret.T
R0_over_time = [logistic_R_0(i) for i in range(len(t))]  # to plot R_0 over time: get function values
Alpha_over_time = [s * I[i]/N + alpha_opt for i in range(len(t))]  # to plot alpha over time
```

In [156]:
```
plotseird(t, S, E, I, R, D, R0=R0_over_time, Alpha=Alpha_over_time)
```