Differential equation models

Taras Nagornny, Marcos Lora, Jonathan Cheng

Chem/Eng 231(Mathematical Modeling)

To: Professor Peng Bai

Third May, 2022

Simplified SIRD Model for the COVID19 in China, France and the US

Taras Nagornny, Marcos Lora, Jonathan Cheng

Abstract

For this report, a method is needed to predict the trends of the spread of Covid-19 to try and prevent further spread of such disease. A SIRD diagram is used (susceptible-infected-recovered-dead) because it is relatively simple and somehow effective at explaining data sets simultaneously and doing so allows governments to prepare for Covid-19 and to prevent spread of Covid-19 to neighboring areas. By looking at the trends of the *Susceptible*, *Infected*, *Recovered*, and *Dead* population using the simplified SIRD model trends can be made short-time predictions of the amount of people that would fall in the given category for the following week. However, the model would not work for a longer-range predictions because the parameters are not constant and there are much more than three parameters (f.e: any intentional or not the actions taken by the government affects the parameters). So, a SIRD model is useful because it plots a graph that can find the gradient to find the constant for a given time.

The results we are trying to find are the three parameters in the SIRD diagram which are *Beta*, *Lambda* and *Kd*. We did the calculations for China, France and the US, but we will only primarily discuss China for now in the paper. For the first 60 days of data, the results obtained for China were *Kd* = 0.01307, *Lambda* = 0.003894 and *Beta* = 0.3492. After finding the values for the three parameters for each equation we would then compare them to the values given in a different research paper (the paper from the Moodle written by D. Sen, D. Sen). We would repeat this process for the other two countries (France and USA) to see how accurate our values for the parameters were.

Introduction

Covid-19, a strain of the novel coronavirus, originated in Wuhan, China where a large number of people were initially infected. The rapid spread of the virus has presented a major global health crisis. The deadliness of the virus can be attributed to the respiratory inflammation the virus causes. While death is the least favorable outcome of the pandemic, there are other negative impacts it has had, such as social and economical. The spread of the disease is not governed by the infected number of people alone, so mathematical models need to be developed in order to forecast predict, and analyze other factors of spread. Other factors influencing spread include total population of a country, norms and measures taken by a government, population density, and more.

Furthermore, the crisis of Covid-19 has resulted in a need for a quantitative understanding of the time evolution of the complex and non-linear process. Statistical and mathematical analysis of the data would provide insights on the trend and spread of the disease and can aid in containing the disease as fast as possible. The analysis of the reported data would help in analyzing the spread of the disease and to make future predictions and trends. This would allow many organizations to prepare and plan their method in containing the spread. Also, it would help the government to understand whether the actions taken by them were efficient or not.

In literature there are a few models that can be used to explain the data however, they can be classified into 2 different categories, collective and networked models. For this project the SIRD model (Susceptible-Infected-Recovered-Dead) would be used. However, due to the complex interdependence of many different variables governing the spread of the disease the mathematical model would need to predict the temporal behavior of infected, dead and recovered people. Due to many outside factors that cannot be controlled (Quarantine, Social distancing, and lockdown measures) a simple SIRD model would only give a preliminary understanding of the entire process and would not suffice for more complex processes. The generalized SIRD model is used for the spread of the pandemic considering the fraction of the population that is exposed while quarantined, confined, infected, recovered and dead. It is noteworthy for a proper model a simultaneous corroboration of time evolution with the independent data sets is needed. For the following report, we will analyze the Covid-19 time series data for 3 countries (USA, France, and China) using the Generalized SIRD model.

With new variants and strains of Covid-19 (Delta, Omicron) a mathematical model is needed to predict the trends of the spread of the different variants to try and prevent the spread of it. The pandemic has been happening for at least two years now and it looks like it is not going away anytime soon, and new variants are being discovered so the SIRD model is important in trying to predict different trends and find improved methods on how to prevent spread of the disease. There are many methods that can be used to find the rate of the spread of the disease, but the method used is linear regression by least squares method as it would give the most accurate value.

Methods

$$I + S \xrightarrow{\beta} I \qquad \frac{dS(t)}{dt} = -\beta \frac{I(t)S(t)}{N} \tag{1}$$

$$I \xrightarrow{\lambda} R \qquad \frac{dI(t)}{dt} = \beta \frac{I(t)S(t)}{N} - \lambda I(t) - k_d I(t) \tag{2}$$

$$I \xrightarrow{k_d} D \qquad \frac{dR(t)}{dt} = \lambda I(t) \tag{3}$$

$$\frac{dD(t)}{dt} = k_d I(t) \tag{4}$$

Figure (1) The equations used for the Generalized SIRD model.

Sen, D., & Sen, D (2021)

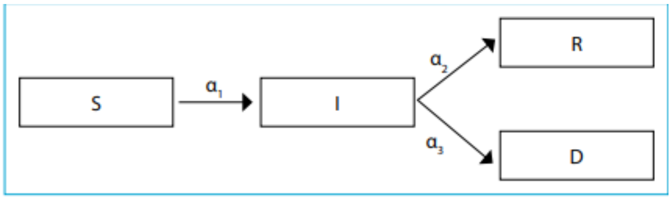


Figure (2) Simplified flowchart showing how to solve each parameter (a1 is beta a2 is lambda and a3 is Kd). It's one of the other ways to solve this system of equations. We did not use it.

The Generalized SIRD model deals with the number of Susceptible, Infected, Recovered and Dead people. This model assumes that a susceptible person who comes in contact with someone infected is prone to get the disease. Also, an infected person has two outcomes: they either recover☺ or they die. Another assumption made by the model is that people who are exposed to the virus are infected immediately with no latent time between the exposure and the infection happening. Outside factors like quarantine are not considered in the model as well. As seen above those are the equations, we are basing the SIRD model on. S(t) is the number of people susceptible to the disease at a given time and is not cumulative, I(t) is the number of people who have been infected at a given time and is not cumulative as well, R(t) is the cumulative number of people who have recovered from the disease, and D(t) is a cumulative number of people who have died from the disease. The parameters of the model represent: *Kd* – death rate due to the disease, *Lambda* – recovery rate, *Beta* – exposure rate.

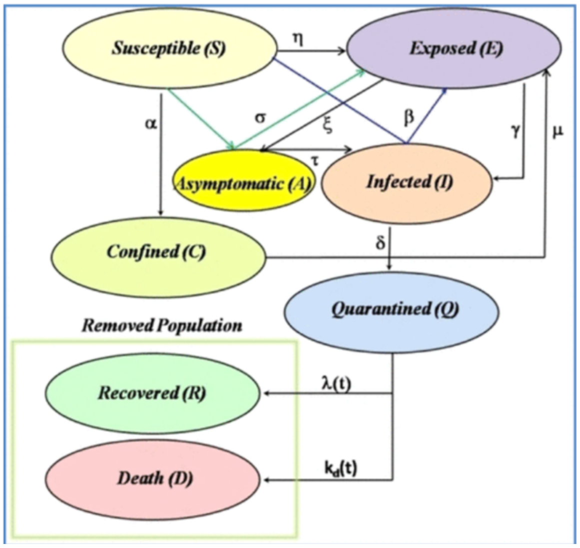


Figure (3) Schematic of SIRD model

$$S \xrightarrow{\alpha} C$$
$$\frac{dS(t)}{dt} = -\alpha S(t) - \beta\frac{I(t)S(t)}{N} - \sigma\frac{S(t)A(t)}{N} - \eta S(t) \tag{5}$$

$$I + S \xrightarrow{\beta} E \tag{6}$$

$$C \xrightarrow{\mu} E \qquad \frac{dA(t)}{dt} = -\tau A(t) + \xi E(t) \tag{7}$$

$$S \xrightarrow{\eta} E \quad S + A \xrightarrow{\sigma} E \qquad \frac{dC(t)}{dt} = \alpha S(t) - \mu C(t) \tag{8}$$

$$E \xrightarrow{\xi} A \quad E \xrightarrow{\gamma} I$$
$$\frac{dE(t)}{dt} = -\gamma E(t) + \beta\frac{I(t)S(t)}{N} + \mu C(t) + \eta S(t) + \sigma\frac{S(t)A(t)}{N} - \xi E(t) \tag{9}$$

$$A \xrightarrow{\tau} I \quad I \xrightarrow{\delta} Q \qquad \frac{dI(t)}{dt} = \tau A(t) + \gamma E(t) - \delta I(t) \tag{10}$$

$$Q \xrightarrow{\lambda(t)} R \qquad \frac{dR(t)}{dt} = \lambda(t)Q(t) \tag{11}$$

$$Q \xrightarrow{k_d(t)} D$$
$$\frac{dQ(t)}{dt} = \delta I(t) - \lambda(t)Q(t) - k_d(t)Q(t) \tag{12}$$

$$\frac{dD(t)}{dt} = k_d(t)Q(t) \tag{13}$$

$$S(t) + C(t) + E(t) + A(t) + I(t) + Q(t) + R(t) + D(t) = N \tag{14}$$

Figure (4) illustrates the relationship between the various parameters and related factors in the Modified SIRD model.

As seen in above (Figure (4)) a lot more parameters need to be found, for this Modified SIRD model However, due to the fact that more factors were considered the graphs should be more accurate and allow for more accurate predictions. The method for collecting the data and finding the constants for a given time is different from our generalized SIRD model (we need to use fsolve and ode45 in MATLAB)

So, as motioned above, there are two methods to find the parameters for this system of ODE. The first method involves MATLAB operators such as ode45 (or ode23) and fsolve, and the second method is graphical. We decided to use the graphical method because other groups decided to use ode and fsolve method. Also, it seemss beneficial for us to be able to visualize the data.

Our plan of solving Generalized SIRD model is:

1) Arrange the given data. The given data (for China and France) is cumulative and is separated by regions. For our calculations we need the data for the whole country (no separations by region). Also, the values for I(t) and S(t) should not be cumulative. Sorting the data is 70% of the code. I used different loops and matrixes operations.
2) Find the derivative of a given data sets with respect to time using the gradient function in MATLAB. *(Please, see the picture below)*

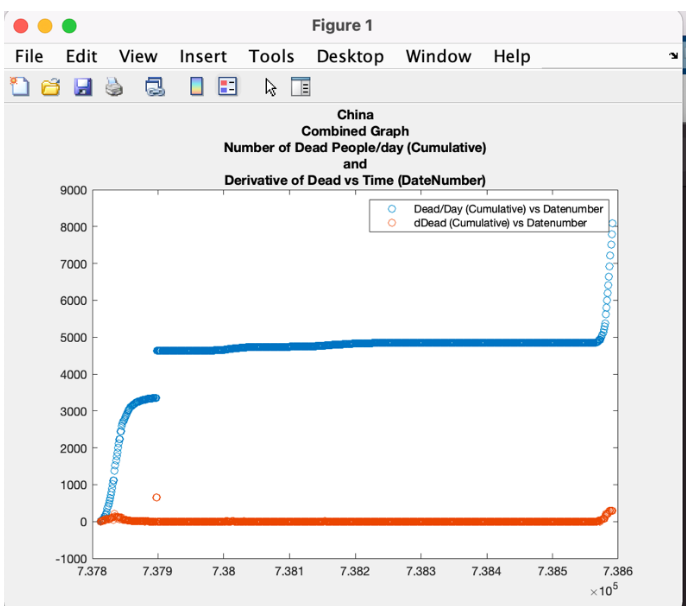


Figure (5) graph for number of dead people per day and derivative of the number of dead people vs time.

3) Using the equations (1), (3), (4) for Generalized SIRD model from the paper (Figure 1). Plot the derivatives against the data. For example, Derivative of Dead people (cumulative) against the number of Infected people (not cumulative). (*Please, see the picture below)*

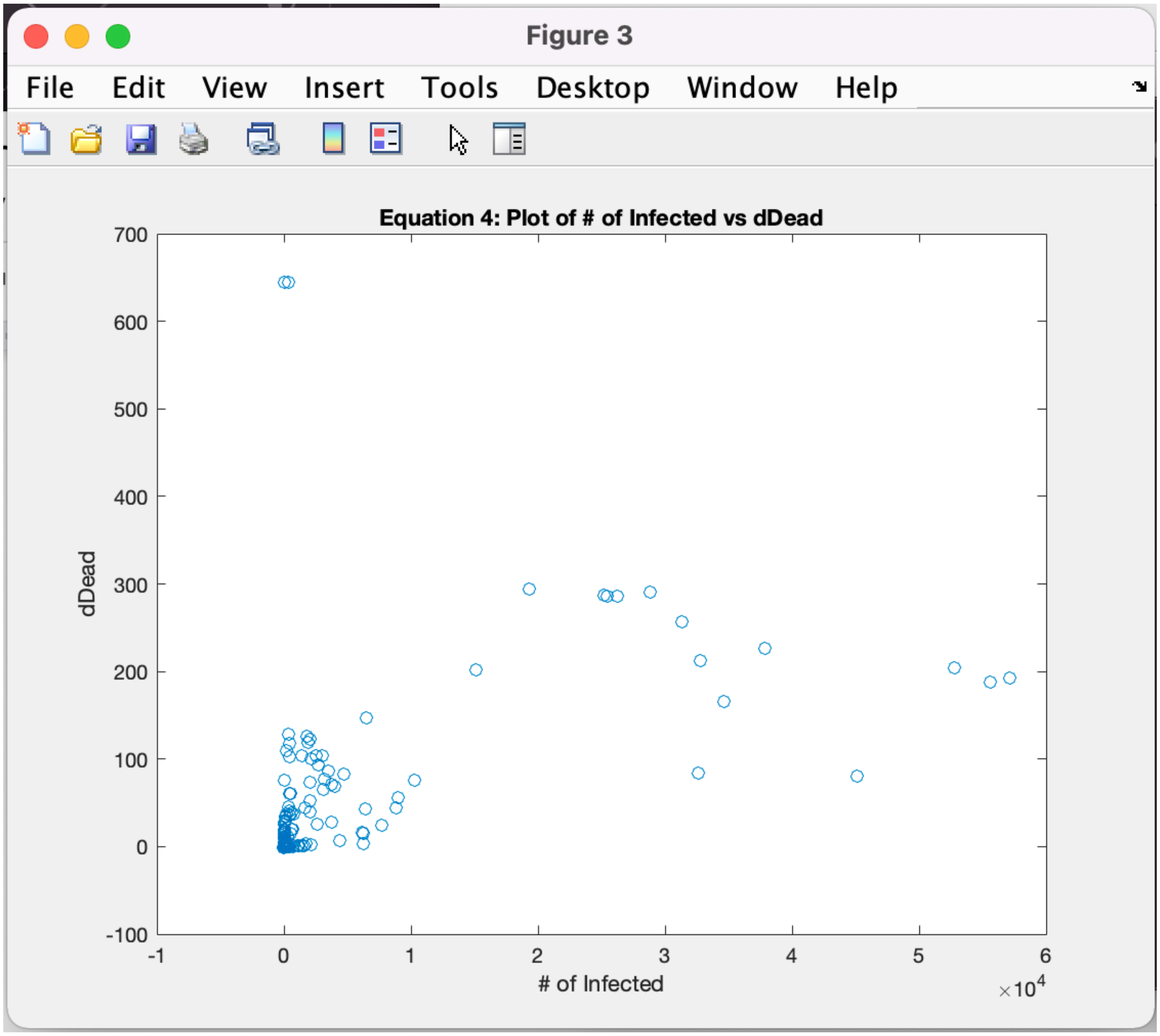


Figure (6) Graph for the number of people infected vs the derivative of the number of dead people.

4) Do the linear regression of the obtained values from step 4. The slope of the graph is going to be a parameter. For linear regression we used the pre-installed Matlab app called "Curve Fitting". (Please, see the picture below)

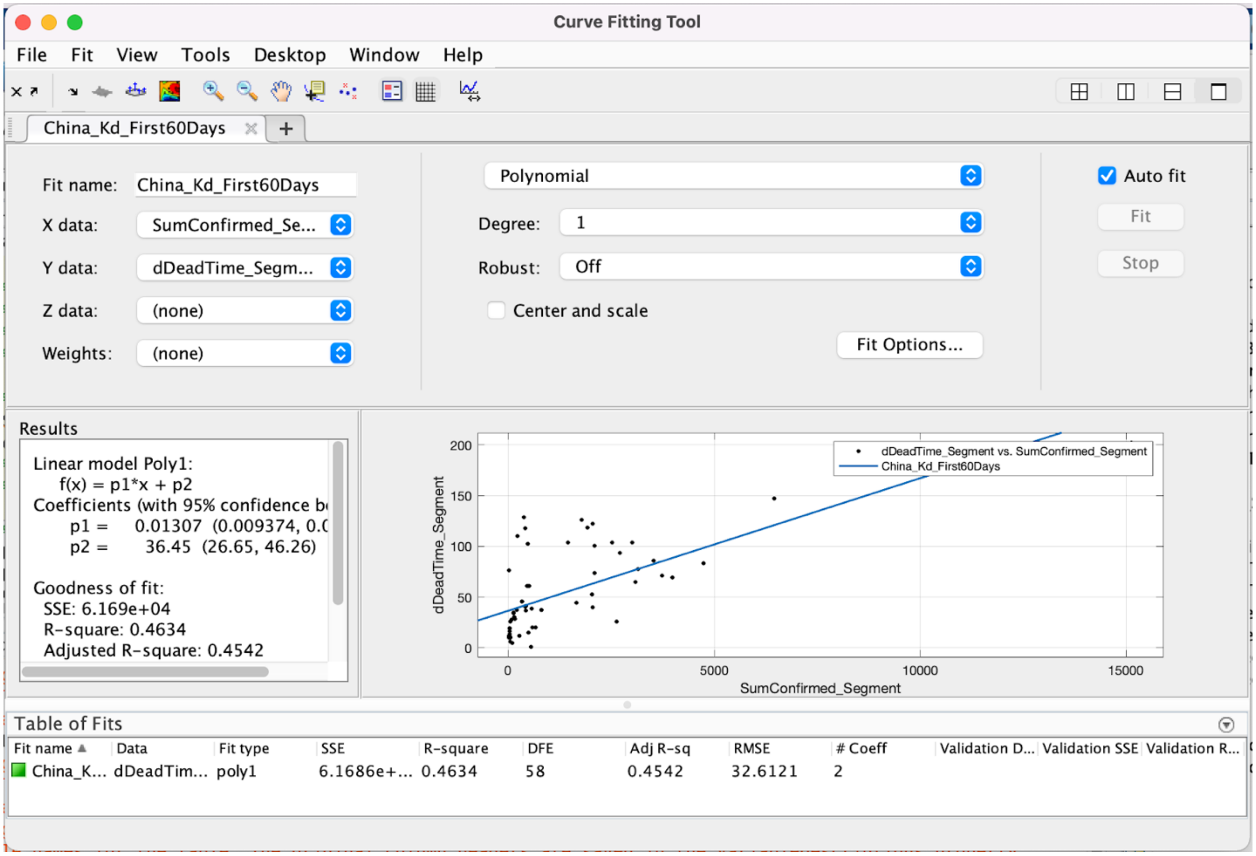


Figure (7) Graph of the equation 4. Linear regression. The slope is the Kd value

<u>Note</u>: we used only the first 60 days out of all 780 days to make the linear regression analysis and find the values of the parameters. We decided to use the first 60 days because we were interested in parameters for the beginning of the pandemic. We acknowledge that the values of the parameters for the different, future time segments are going to be different as we can see on the graphs (the slopes are changing).

As seen in the *Figure 5* in methods which is the number of dead people per day (blue line) after a couple of months the cumulative number of deaths becomes constant which means that people do not die anymore.

Results and discussion

We used a simple SIRD model to find the values of the three parameters. Using the first 60 days time segment the values and linear regression the parameters for given three countries are:

For first 60 Days time Segment	Kd	Lambda	Beta
China	0.01307	0.003894	-0.3492
France	0.04624	0.01903	0.04365
USA	0.01899	0.01874	-0.2903

Another 60 Days time Segment	Kd	Lambda	Beta
China	0.06082	16.81	-0.1754
France	10.15	0.2174	-0.5054
USA	165	-0.3305	83.63

We decided to use the data for China to analyze the values of our parameters. Why? We tossed the coin. Moreover, we decided that the paper would be too broad if we insert and describe the graphs for each country. It would make no sense. Although I expect the values of parameters to be different for each country due to local health regulations; the idea behind is similar.

These parameters represent the actions taken by the government and the efficiency of these actions. As we can see from the table above the values of Kd and Beta are very different. Kd is a recovery rate parameter. The bigger Lambda = 16.81 corresponds to the later time segment and is larger than the Lambda = 0.003894 corresponding to the first days of pandemic. Parameter Lambda defines the recovery rate. As we can expect the recovery rate should increase significantly as time goes, because the government have more time to take some actions, and there are more people to recover.

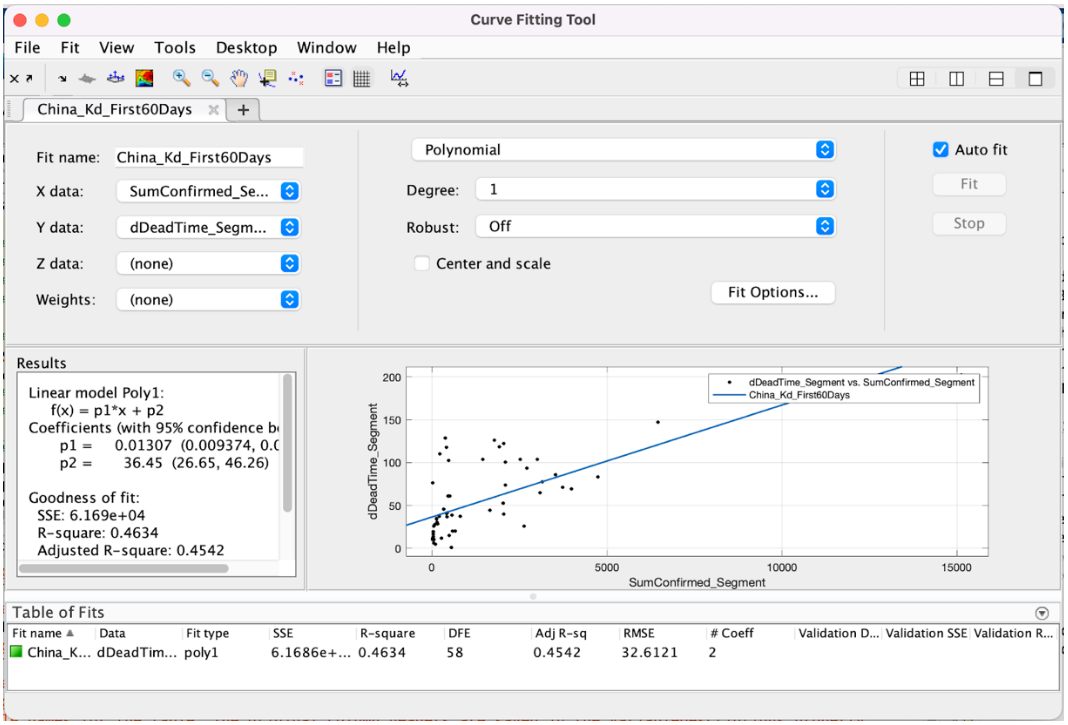


Figure (8) The Value for Kd for China for the first 60 days.

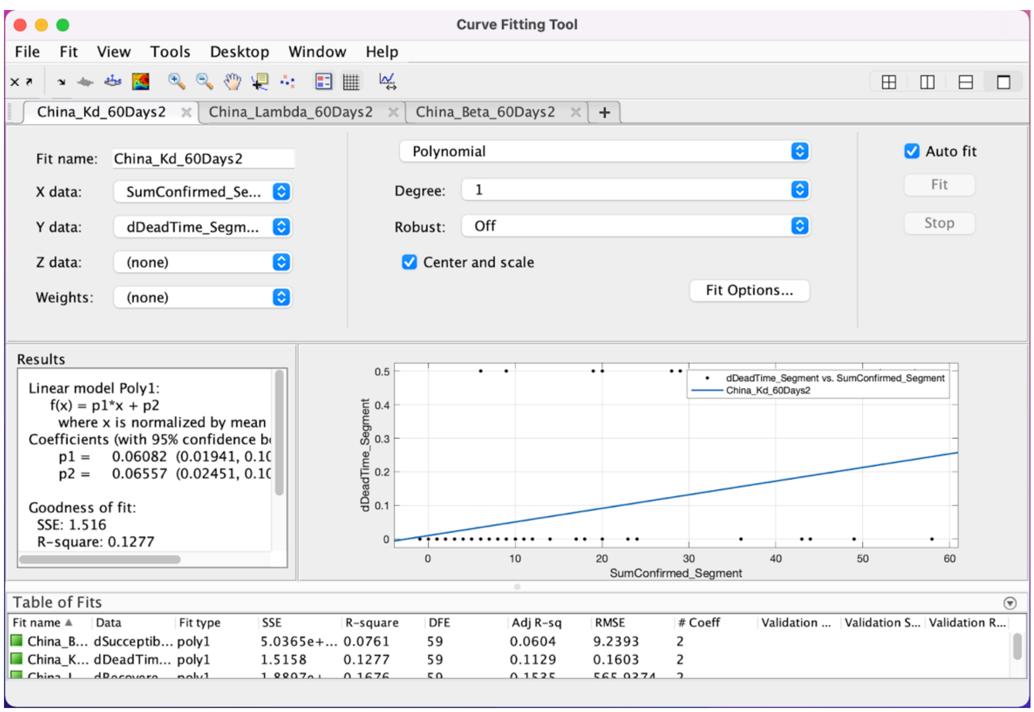


Figure (9) Value for Kd for China for 60 days in the middle.

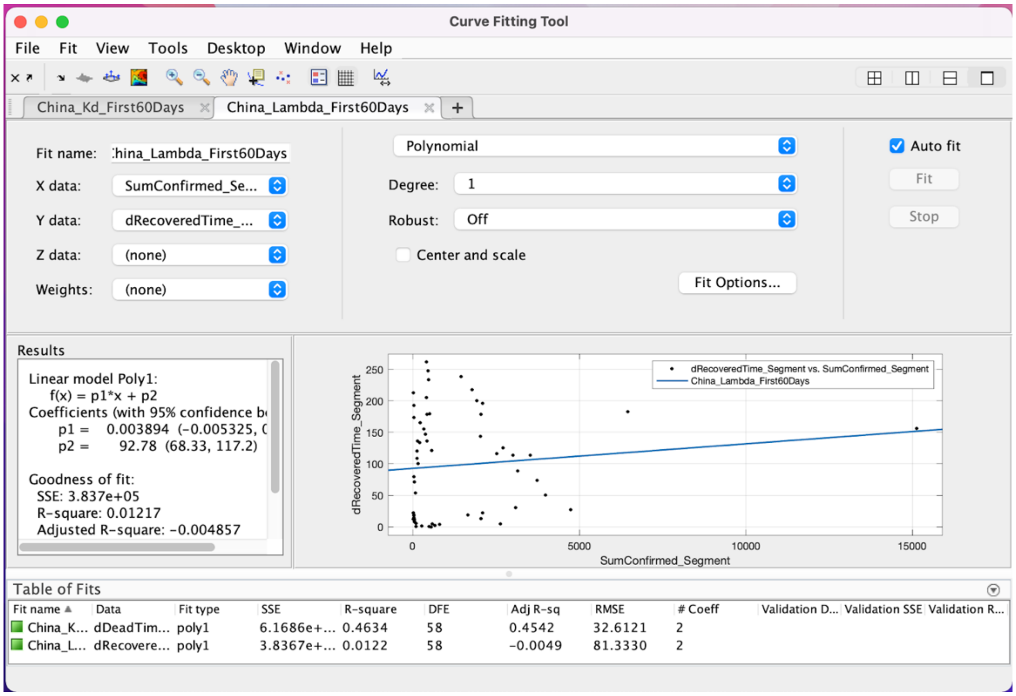


Figure (10) Value for Lambda for China first 60 days.

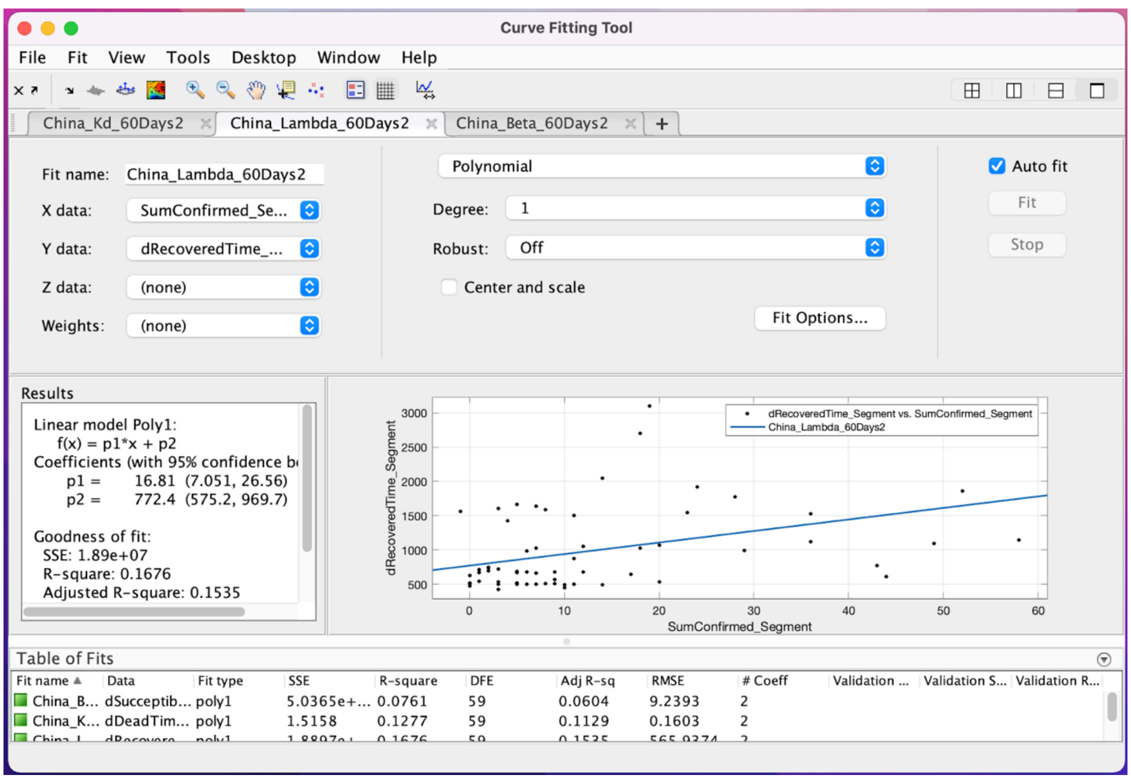


Figure (11) above: Value for Lambda for China for 60 days in the middle.

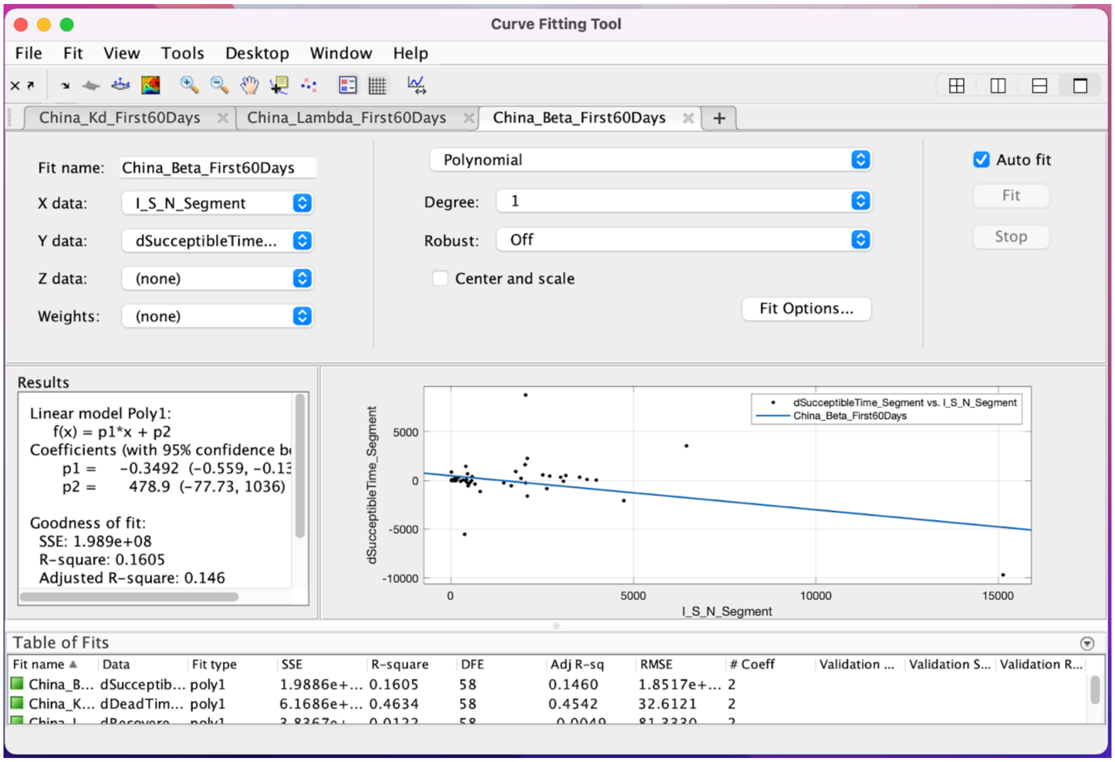


Figure (12) Value for beta in China for first 60 days.

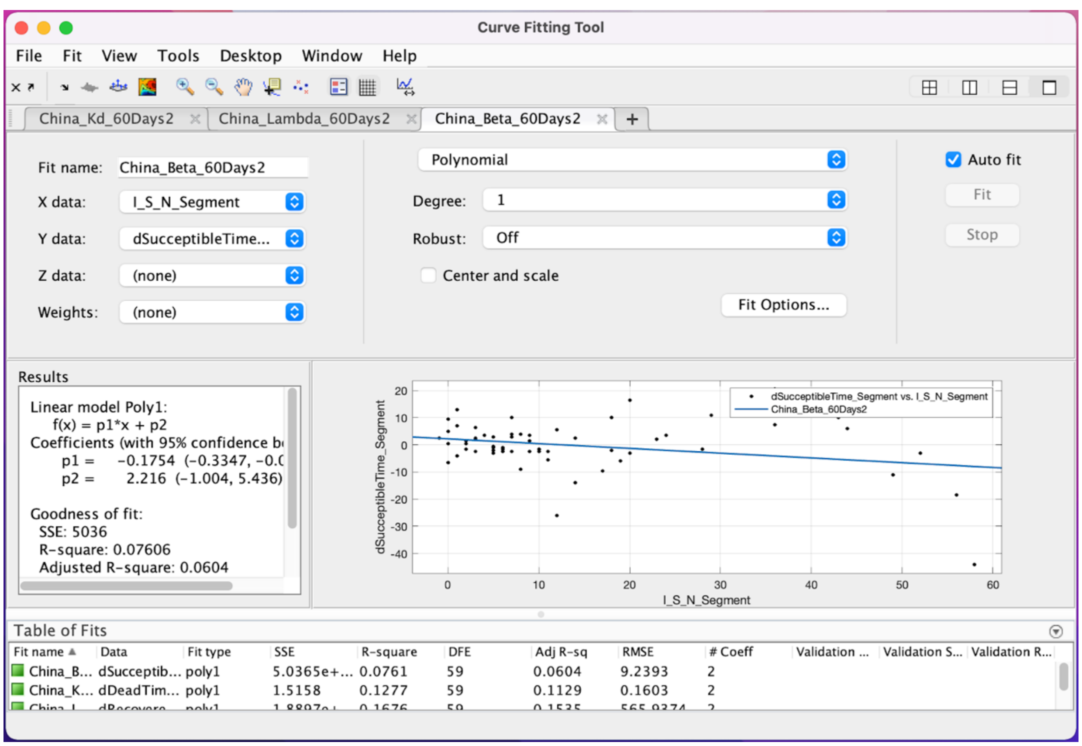


Figure (13) Value for beta in China for 60 days in the middle.

Values from the given in Moodle paper for comparison (Sen D & Sen D = (Sen D)^2:):

	country						
	China	India			Italy	the United States	France
		overall	Maharashtra	Gujarat			
population ($\times 10^7$)	143.93	138.73	12.2	6.5	6.046	33.1	6.527
population density (per km^2)	153	382	365	308	206	36	119
α (day^{-1})	0.080	0.0025	0.0046	0.00149	0.0194	0.01215	0.003709
β (day^{-1})	0.912	0.180	0.2237	0.1132	7.567	7.6678	6.09694
μ (day^{-1})	6.675×10^{-8}	8.152×10^{-6}	9.284×10^{-6}	1.092×10^{-5}	2.278×10^{-6}	3.514×10^{-3}	2.3655×10^{-13}
η (day^{-1})	1.068×10^{-7}	1.529×10^{-8}	7.689×10^{-8}	2.207×10^{-6}	9.180×10^{-7}	1.489×10^{-2}	9.2679×10^{-17}
σ (day^{-1})	0.166	9.217×10^{-6}	1.00×10^{-10}	2.510×10^{-7}	1.463×10^{-3}	1.00×10^{-15}	1.00×10^{-17}
τ (day^{-1})	4.802	4.221×10^{-7}	1.061×10^{-7}	4.215×10^{-6}	1.109×10^{-4}	9.1234×10^{-4}	1.0×10^{-17}
ξ (day^{-1})	1.097	2.200×10^{-10}	5.3046×10^{-9}	0.0153	0.263	1.535×10^{-3}	2.2366×10^{-2}
γ (day^{-1})	0.720	0.023	0.0665	0.0629	0.021	9.8076×10^{-5}	0.02282
δ (day^{-1})	0.105	0.076	0.067	0.0711	0.077	9.7942×10^{-3}	0.0639
λ_0 (day^{-1})	0.216	0.6079	0.308	8.8107	0.157	3.1848	4.0796
λ_1 (day^{-1})	0.055	1.00×10^{-8}	3.487×10^{-8}	0.0086	0.025	1.1072×10^{-2}	0.0597
k_{d0} (day^{-1})	0.035	0.0267	0.1255	1.9142	0.779	3.52629	4.2616
k_{d1} (day^{-1})	0.078	0.0166	0.0167	0.02419	0.061	2.718×10^{-2}	0.06927
initial infected	1	1	1	1	1	1	1
initial exposed	1	1	1	1	1	1	1
initial confined	0	0	0	0	0	0	0
initial quarantined	1	1	1	1	1	1	1

Figure (14)

Comparing the values found by us and found by the research group we can easily notice that the values of the parameters *Kd* and *Lambda* are usually close to each other. But the values of *Beta* are different. We have negative *Beta* values, but scientists have the positive *Beta* values. We think there are only two possible explanations for that:

1. We made a mistake in the code, because the data for the number of Susceptible people, which is required to find the *Beta* parameter, was not given and we had to calculate it by ourselves using the given data and the total population.
2. We used the SIRD model with only 3 parameters and 4 equations, when scientists used the Modified SIRD model with 10 parameters and 10 equations.

I think in the majority of cases the Modified SIRD model would estimate the COVID spread much better compared to our simple SIRD simply because it takes into account more parameters.

We could not solve the Modified SIRD model using our graphical method because the equations there are much more complicated and are beyond the simple linear regression analysis. The modified SIRD model can be solved using the analytic (ode, fsolve).

Also, when we tried to find the parameters we noticed that the value of the parameter depends on the time segment you are taking.

We must take in account that the results calculated in the research paper were for a bigger time period 250-300 days for China, and 2500 days for USA so the values for the three parameters would be different to the one we calculated because our time period is a lot smaller so the results would vary a lot more. Having multiple time graphs for different time periods allows us to not only compare the parameters in different time segments and it also allows us to compare the actions that are taken by the government at different time periods to attempt to stop the spread of Covid-19, it would also allow us to see how effective the method is at stopping the spread of the disease has slowed down exponentially or if it still.

Conclusion

A simple and standard SIRD model was modified to analyze the evolution of Covid-19 pandemic for three countries. It is found to be capable for the most part of explaining the spread of the disease for all data sets simultaneously (*Infected*, *Recovered* and *Dead*) for the countries used. Due to the fact that many outside factors were ignored the simple SIRD diagram did have errors when calculating the three parameters when compared to Sen,D Sen,D's research paper. Despite having similar values for *Kd* and *Lambda* the value we calculated for beta was negative while in the research paper. So many improvements could be made to make sure the values for all the parameters were better.

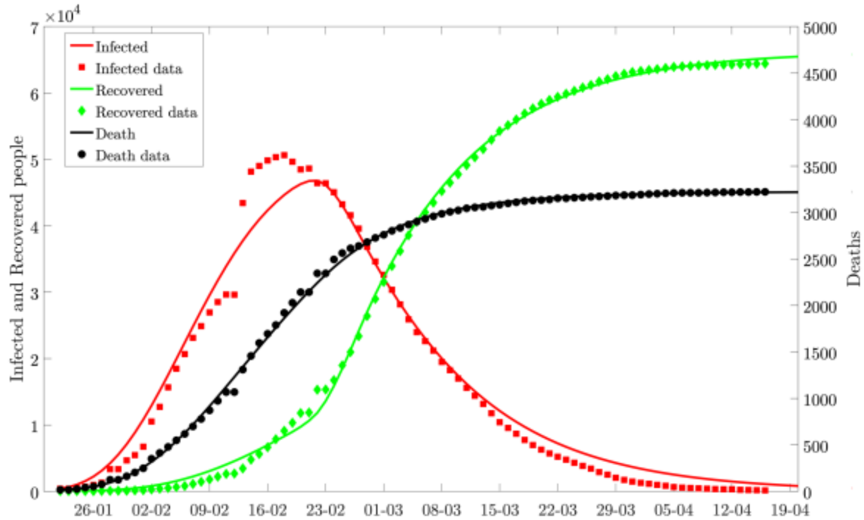


Figure (15) A complex SIRD model used for the outbreak of Covid-19 in China.

Caccavo, D.

There were improvements that could have been made while using the SIRD model, firstly due to the fact that a simple SIRD model was used many assumptions were made. So when the graphs were made the trends that were seen were not realistic due to the fact that latent time has not been considered. As we can see above the more complex SIRD diagram (Figure 2) uses 10 parameters, so a satisfactory description of the data has been published. It is worth mentioning that the data started at day 1 of lockdown so the average number of people meeting would decrease so the number of people who are infected and susceptible to the disease decreases exponentially. The results show that it requires 30 days of quarantine to slow down the spread of the virus. As we can see the increase of parameters would make the analysis of the trends more accurate so better predictions can be made. It can also be seen that not having enough parameters and a lot of outside factors were being ignored so the decay predicted is slower than reality and the predicted values for deaths were also a lot higher than reality. Also, it is not quite correct to compare the data at the same point of time between China and the US. COVID came to the US much later.

References:

1) Al-Raeei, M. (2020). The forecasting of COVID-19 with mortality using SIRD epidemic model for the United States, Russia, China, and the Syrian Arab Republic. *AIP Advances*, *10*(6), 065325. https://doi.org/10.1063/5.0014275

2) Caccavo, D. (2020, April 17). *Chinese and Italian COVID-19 outbreaks can be correctly described by a modified SIRD model* [Review of *Chinese and Italian COVID-19 outbreaks can be correctly described by a modified SIRD model*].

3) Fernández-Villaverde, J., Jones, C., Upenn, S., Gsb, N., Aschenbrenner, L., Auclert, A., Cochrane, J., Tella, S., Ellison, G., Hall, B., Klenow, P., Tonetti, C., Topa, G., & Yashiv, E. (2020). *Estimating and Simulating a SIRD Model of COVID-19 for Many Countries, States, and Cities*. https://web.stanford.edu/~chadj/sird-paper.pdf

4) Nisar, K. S., Ahmad, S., Ullah, A., Shah, K., Alrabaiah, H., & Arfan, M. (2021). Mathematical analysis of SIRD model of COVID-19 with Caputo fractional derivative based on real data. *Results in Physics*, *21*, 103772. https://doi.org/10.1016/j.rinp.2020.103772

5) Sen, D., & Sen, D. (2021). Use of a Modified SIRD Model to Analyze COVID-19 Data. *Industrial & Engineering Chemistry Research*. https://doi.org/10.1021/acs.iecr.0c04754

6) Lounis, Mohamed, and Marwan Al-Raeei. "Estimation of Epidemiological Indicators of COVID-19 in Algeria with an ..." *Estimation of Epidemiological Indicators of COVID-19 in Algeria with an SIRD Model*, Lounis, M., & Al-Raeei, M. (n.d.). *Estimation of epidemiological indicators of COVID-19 in Algeria with an ...* Estimation of

epidemiological indicators of COVID-19 in Algeria with an SIRD model. Retrieved May 3, 2022, from https://www.researchgate.net/profile/Mohamed-Lounis/publication/349443602_Estimation_of_epidemiological_indicators_of_COVID-19_in_Algeria_with_an_SIRD_model/links/603ab9434585158939d58998/Estimation-of-epidemiological-indicators-of-COVID-19-in-Algeria-with-an-SIRD-model.pdf

7) Katella, K. (2022, April 20). *Omicron, Delta, Alpha, and more: What to know about the coronavirus variants*. Yale Medicine. Retrieved May 3, 2022, from https://www.yalemedicine.org/news/covid-19-variants-of-concern-omicron

Appendix (MATLAB Code)

Table of Contents


```
% China %
AllConfirmed =
 readtable('time_series_covid19_confirmed_global_narrow.csv');
AllDead = readtable('time_series_covid19_deaths_global_narrow.csv');
AllRecovered =
 readtable('time_series_covid19_recovered_global_narrow.csv');
```

```
Warning: Column headers from the file were modified to make them valid
 MATLAB
identifiers before creating variable names for the table. The original
 column
headers are saved in the VariableDescriptions property.
Set 'VariableNamingRule' to 'preserve' to use the original column
 headers as
table variable names.
Warning: Column headers from the file were modified to make them valid
 MATLAB
identifiers before creating variable names for the table. The original
 column
headers are saved in the VariableDescriptions property.
Set 'VariableNamingRule' to 'preserve' to use the original column
 headers as
table variable names.
Warning: Column headers from the file were modified to make them valid
 MATLAB
identifiers before creating variable names for the table. The original
 column
headers are saved in the VariableDescriptions property.
Set 'VariableNamingRule' to 'preserve' to use the original column
 headers as
table variable names.
```

Set Your Conditions, please

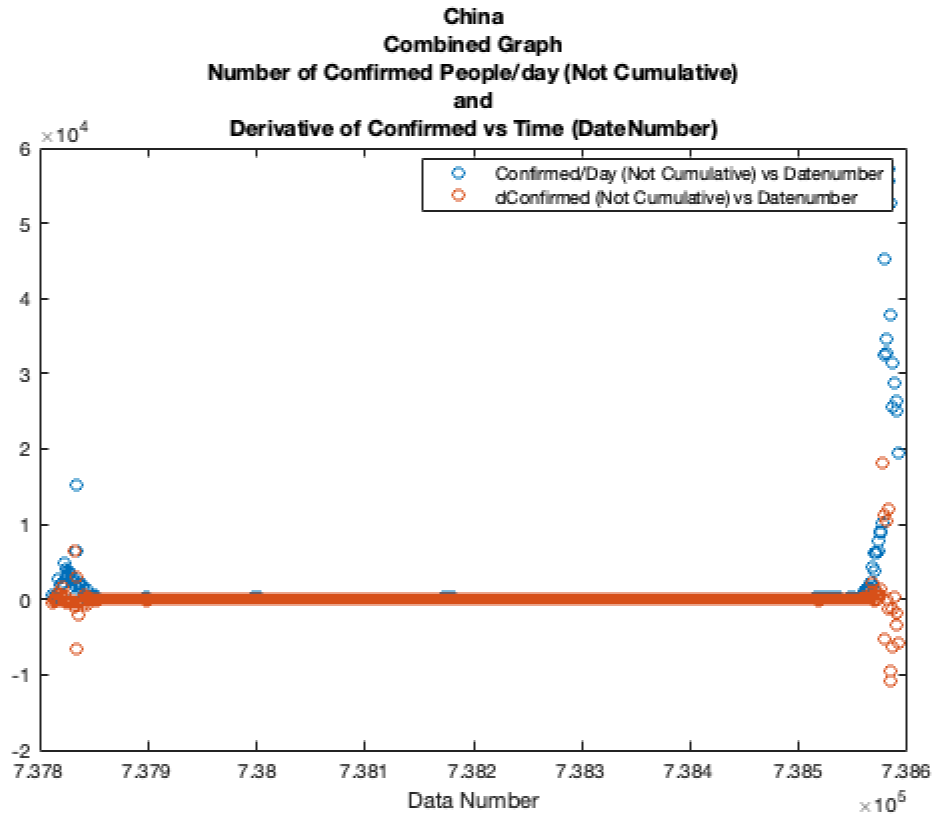
```
TotalPopulation = 1.4*10^9; % Assume we tested that amount of people.
```

```
TimeSegmentLength = 60;% Days
TimeSegmentStartingDay = 100;

ChinaLowerBorderRange = 46021; %46021
ChinaUppperBorderRange = 72540; %72540
```

Confirmed = Infected

```
ConfirmedChina =
 flip(AllConfirmed{ChinaLowerBorderRange:ChinaUppperBorderRange,6}); %
 Range of Confirmed in China {46021:72540,6}
ConfirmedChinaTime =
 flip(AllConfirmed{ChinaLowerBorderRange:ChinaUppperBorderRange,5}); %
 Range of Confirmed in China
DateNumber = datenum(ConfirmedChinaTime); % Time Range in Data Number

Unique = unique(DateNumber); % Number of unique days within which the
 measurements were taken.

[~,~,ix] = unique(DateNumber);
UniqueRepetitions = mean(accumarray(ix,1)); % How many times each
 DateNumber repeats

Length = length(Unique); % Number of Unique Days the measurements were
 taken
x = nan(UniqueRepetitions,Length-1);

for i=1:Length-1
r = DateNumber(1)+i;
a = find(DateNumber == r); % Index
x(:,i) = a;
end
x = [(find(DateNumber == r-(Length-1))) x]; % Indexes for 34
 provinces ; 780 days

w = 0;
for i=1:Length
    w = w+1;
    k = ConfirmedChina(x(:,w));
    h(:,i) = k; % Actual number of Confirmed in each province of China
end

fckConfirmed = reshape(h,[UniqueRepetitions,Length]);
SumConfirmedCumulative = (sum(fckConfirmed))'; % Cumulative # of
 Confirmed/day in China (no more separation on provinces).

y1 = 0;
for i=1:Length-1
    y1 = y1+1;
    u1 = SumConfirmedCumulative(y1+1)-SumConfirmedCumulative(y1);
    j1(:,i) = u1;
end
j1 = [SumConfirmedCumulative(1) j1(1:Length-1)];
```

```
fckConfirmed2 = reshape(j1,[1,Length]);
SumConfirmed = (fckConfirmed2)'; % Number of Infected(Confirmed)
 people/day in China at a particular time (t). Not cumulative!

dConfirmedTime = gradient(SumConfirmed,Unique); % Derivative of the
 data to time

figure(1)
plot(Unique,SumConfirmed,'o');% Plot of Confirmed/Day vs Time
title({'China','Combined Graph','Number of Confirmed People/
day (Not Cumulative)', 'and', 'Derivative of Confirmed vs Time
 (DateNumber)'});
hold on
plot(Unique,dConfirmedTime,'o'); % Plot of Derivative of Data vs Time
legend({'Confirmed/Day (Not Cumulative) vs Datenumber','dConfirmed
 (Not Cumulative) vs Datenumber'})
xlabel('Data Number')
hold off
```



Dead (Done)

```
DeadChina =
 flip(AllDead{ChinaLowerBorderRange:ChinaUppperBorderRange,6}); % Data
 for # of Dead People in China {46021:72540,6}
```

```
DeadChinaTime =
 flip(AllDead{ChinaLowerBorderRange:ChinaUppperBorderRange,5}); % Data
 for dates (month/date/year) of Dead People in China
DateNumber = datenum(DeadChinaTime); % Time (month/date/year) Range in
 Data Number

Length = length(Unique); % Number of (Unique Data Numbers) in our
 Range = # of days the measurements were taken
x = nan(UniqueRepetitions(1),Length-1); % 34x779 Matrix of NaN

for i=1:Length-1
r = DateNumber(1)+i;
a = find(DateNumber == r); % Indexes (in DataNumber) for each Data
 Number. Each datanumber repeats 34 times - there are 34 provinces.
x(:,i) = a;
end
x = [(find(DateNumber == r-(Length-1))) x]; % Fills the matrix
 (34x780) with Indexes for each DataNumber (780 DataNumbers as total).
 780 Columns = 780 Days of Measurements. 34 Raws = 34 Provinces.

wDead = 0;
for i=1:Length
    wDead = wDead+1;
    kDead = DeadChina(x(:,wDead)); % Take each column of x
 individually = take each day individually. And use this Indexes to
 find the #of associated people.
    hDead(:,i) = kDead; % Actual values of Dead (Now we have the
 matrix (34x780) with the actual number of Dead people instead of
 Indexes).
end

fckDead = reshape(hDead,[UniqueRepetitions,Length]); % Something with
 syntaxis
SumDeadCumulative = (sum(fckDead))'; % Cumulative of Dead/day in China
 (no more separation on provinces).

dDeadTime = gradient(SumDeadCumulative,Unique); % Derivative of the
 data (Dead People) with respect to time

% % %
figure(2)
plot(Unique,SumDeadCumulative,'o');% Plot of Dead/Day vs Time
title({'China','Combined Graph','Number of Dead People/day
 (Cumulative)','and','Derivative of Dead vs Time (DateNumber)'})
hold on
plot(Unique,dDeadTime,'o'); % Plot of Derivative of Data vs Time
legend({'Dead/Day (Cumulative) vs Datenumber','dDead (Cumulative) vs
 Datenumber'})
xlabel('Data Number')
hold off

% % %
figure(3)
```

```
plot(SumConfirmed, dDeadTime,'o'); % Plot of # of Infected vs dDead.
 "sum1" is Infected people/day
title('Equation 4: Plot of # of Infected vs dDead');
xlabel('# of Infected')
ylabel('dDead')
```

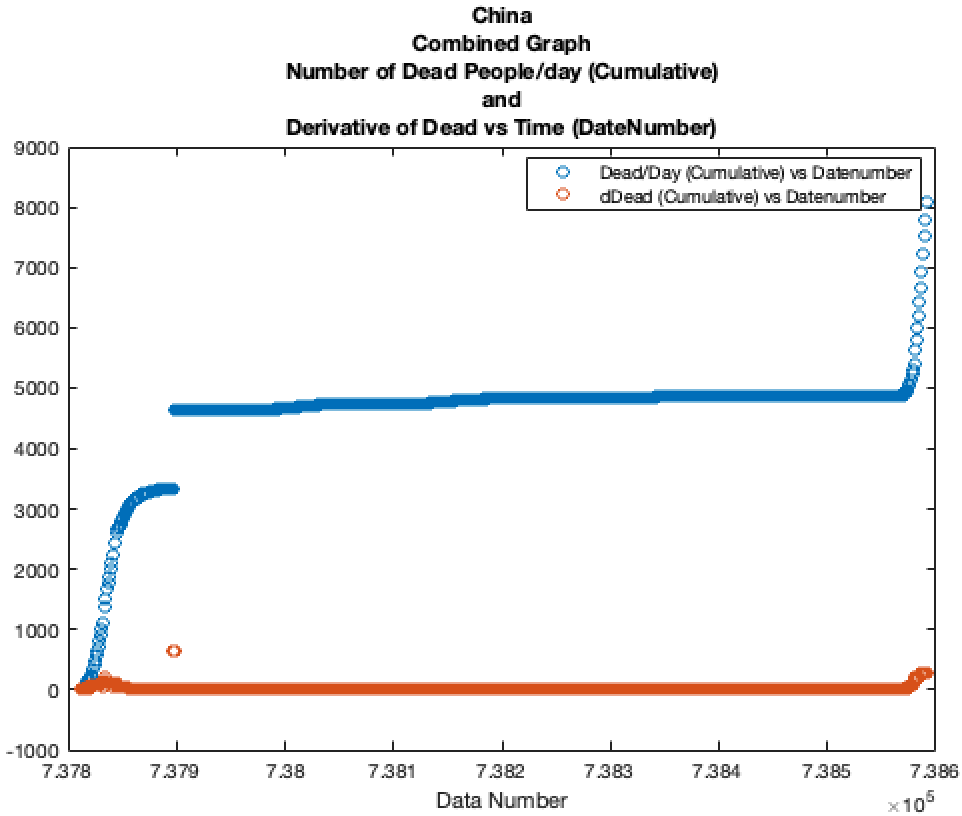
Note: the slope of each individual linear segment is going to be a

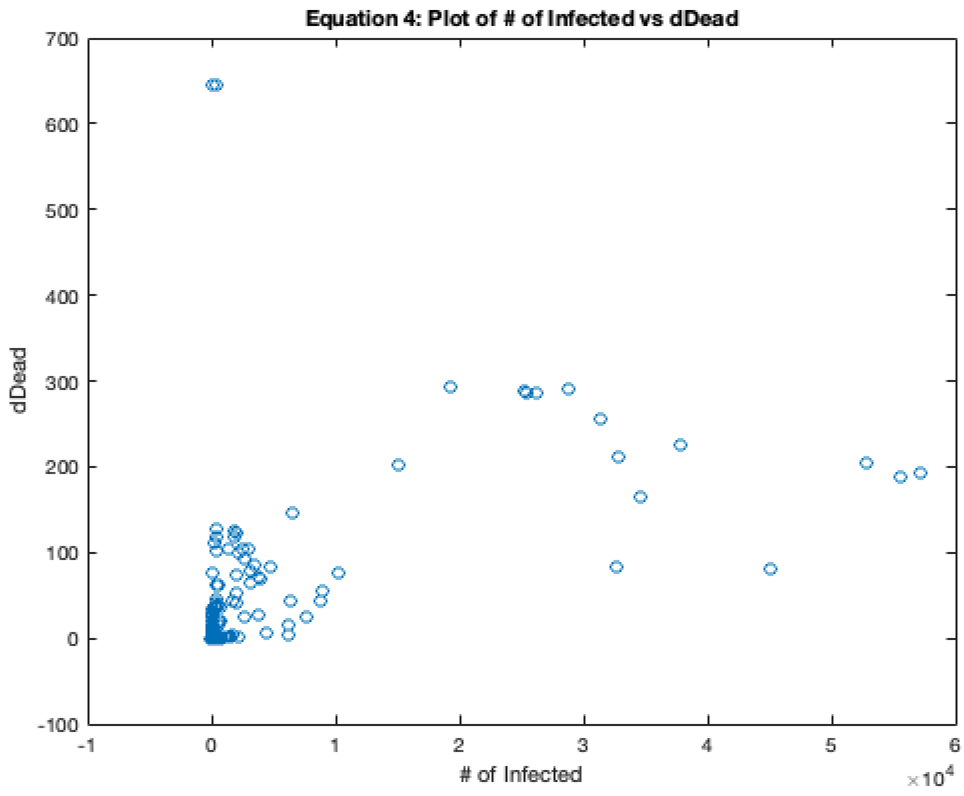
value of Kd. I'm going to take one of the date ranges where the slope behaves linearly and use its Kd value to try to predict the function behaviour for future days.

```
% % %
SumConfirmed_Segment = (SumConfirmed(TimeSegmentStartingDay:
(TimeSegmentLength+TimeSegmentStartingDay)));
dDeadTime_Segment = (dDeadTime(TimeSegmentStartingDay:
(TimeSegmentLength+TimeSegmentStartingDay)));
```

Note: Use the "Curve Fitting App (power = 1) to find the slope (p1), which is going to be equal to Kd

for this segment(1:60) Kd = 0.01307"





Recovered

```
RecoveredChina =
 flip(AllRecovered{ChinaLowerBorderRange:ChinaUppperBorderRange,6}); %
 Data for # of Dead People in China {46021:72540,6}
RecoveredChinaTime =
 flip(AllRecovered{ChinaLowerBorderRange:ChinaUppperBorderRange,5}); %
 Data for dates (month/date/year) of Dead People in China
DateNumber = datenum(RecoveredChinaTime); % Time (month/date/year)
 Range in Data Number

Length = length(Unique); % Number of (Unique Data Numbers) in our
 Range = # of days the measurements were taken
x = nan(UniqueRepetitions(1),Length-1); % 34x779 Matrix of NaN

for i=1:Length-1
r = DateNumber(1)+i;
a = find(DateNumber == r); % Indexes (in DataNumber) for each Data
 Number. Each datanumber repeats 34 times - there are 34 provinces.
x(:,i) = a;
end
x = [(find(DateNumber == r-(Length-1))) x]; % Fills the matrix
 (34x780) with Indexes for each DataNumber (780 DataNumbers as total).
 780 Columns = 780 Days of Measurements. 34 Raws = 34 Provinces.
```

```
wRecovered = 0;
for i=1:Length
    wRecovered = wRecovered+1;
    kRecovered = RecoveredChina(x(:,wRecovered)); % Take each column
 of x individually = take each day individually. And use this Indexes
 to find the #of associated people.
    hRecovered(:,i) = kRecovered; % Actual values of Dead (Now we have
 the matrix (34x780) with the actual number of Dead people instead of
 Indexes).
end

fckRecovered = reshape(hRecovered,[UniqueRepetitions,Length]); %
 Something with syntaxis
SumRecoveredCumulative = (sum(fckRecovered))'; % Cumulative of
 Recovered/day in China (no more separation on provinces).

dRecoveredTime = gradient(SumRecoveredCumulative,Unique); % Derivative
 of the data (Dead People) with respect to time

% % %
figure(4)
plot(Unique,SumRecoveredCumulative,'o');% Plot of Dead/Day vs Time
title({'China','Combined Graph','Number of Recovered People/day
 (Cumulative)','and','Derivative of Recovered (Cumulative) vs Time
 (DateNumber)'})
hold on
plot(Unique,dRecoveredTime,'o'); % Plot of Derivative of Data vs Time
legend({'Recovered/Day vs Datenumber','dRecovered vs Datenumber'})
xlabel('Data Number')
hold off

% % %
figure(5)
plot(SumConfirmed, dRecoveredTime,'o');% Plot of # of Infected vs
 dDead. "sum1" is Infected people/day
title({'China','Equation 3: Plot of # of Infected (Not Cumulative) vs
 dRecovered (Cumulative)'});
xlabel('# of Infected (Not Cumulative)')
ylabel('dRecovered')
```

Note: the slope of each individual linear segment is going to be a

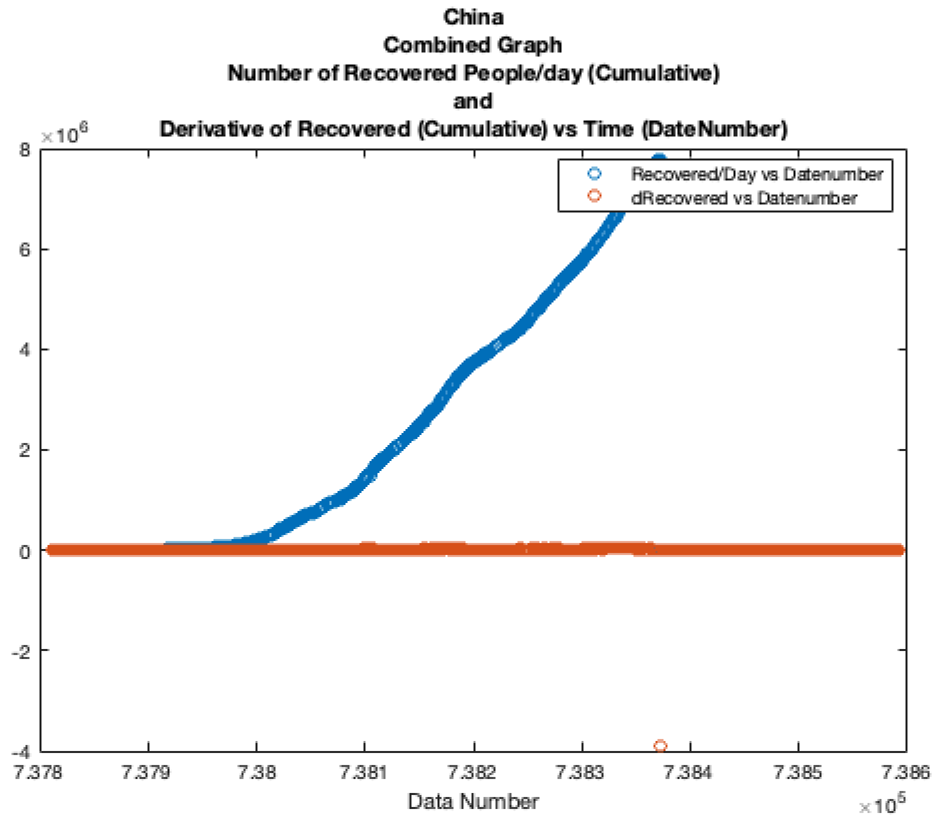
value of Kd. I'm going to take one of the date ranges where the slope behaves linearly and use its Kd value to try to predict the function behaviour for future days.

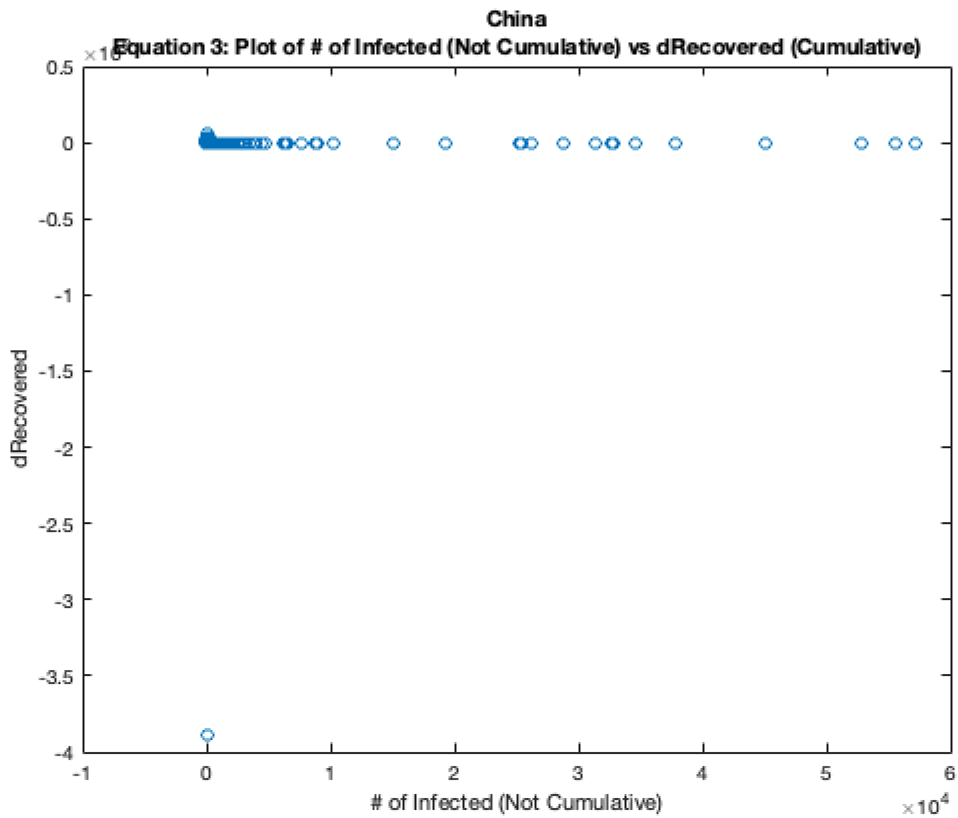
```
SumConfirmed_Segment = (SumConfirmed(TimeSegmentStartingDay:
(TimeSegmentLength+TimeSegmentStartingDay)));
dRecoveredTime_Segment = (dRecoveredTime(TimeSegmentStartingDay:
(TimeSegmentLength+TimeSegmentStartingDay)));
```

Note: Use the "Curve Fitting App (power = 1) to find the slope (p1),

which is going to be equal to Lambda

for this segment(1:60) Lambda = 0.003894"





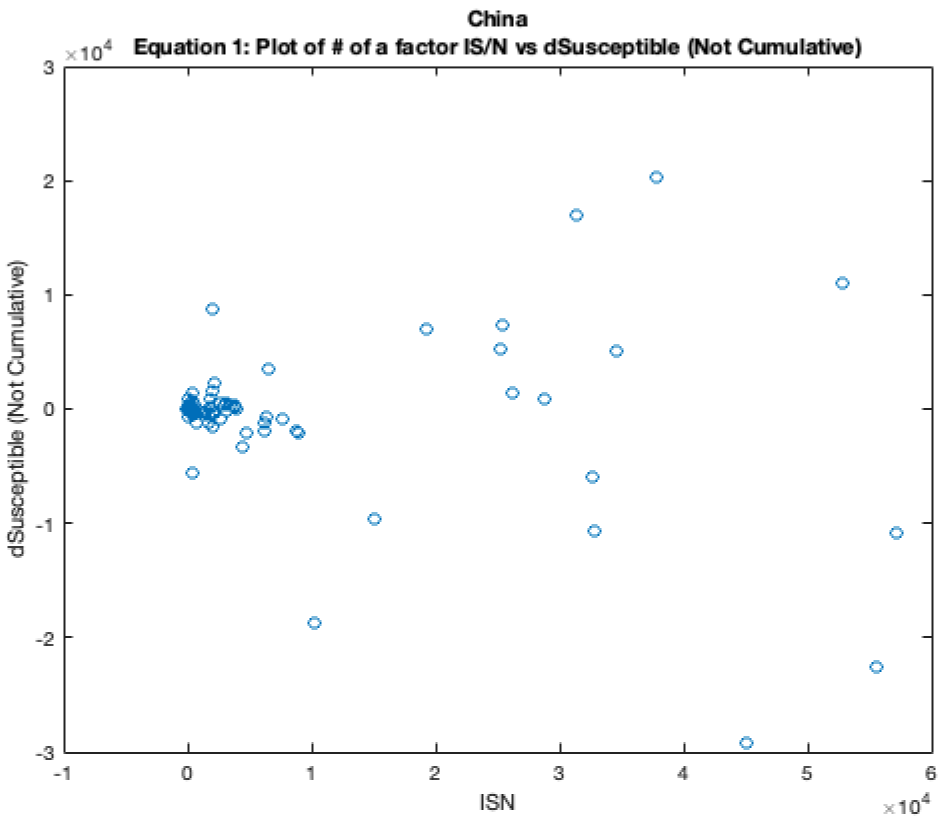
Succeptible

```
y2 = 0;
for i=1:Length-1
    y2 = y2+1;
    u2 = SumDeadCumulative(y2+1)-SumDeadCumulative(y2);
    j2(:,i) = u2;
end
j2 = [SumDeadCumulative(1) j2(1:Length-1)];

fckDead2 = reshape(j2,[1,Length]);
SumDead = (fckDead2)'; % Number of Dead people/day in China at a
 particular time (t). Not cumulative!


y3 = 0;
for i=1:Length-1
    y3 = y3+1;
    u3 = SumRecoveredCumulative(y3+1)-SumRecoveredCumulative(y3);
    j3(:,i) = u3;
end
j3 = [SumRecoveredCumulative(1) j1(1:Length-1)];

fckRecovered2 = reshape(j3,[1,Length]);
```

```
SumRecovered = (fckRecovered2)'; % Number of Recovered people/day in
 China at a particular time (t). Not cumulative!
```

Total Population

```
TotalPopulation = TotalPopulation.*ones(Length,1);
SumSucceptible = TotalPopulation-SumDead-SumRecovered-SumConfirmed; %
 # of succeptible people every day.

dSucceptibleTime = gradient(SumSucceptible,Unique); % Not Cumulative

SumSucceptible_Segment = (SumSucceptible(TimeSegmentStartingDay:
(TimeSegmentLength+TimeSegmentStartingDay)));
dSucceptibleTime_Segment = (dSucceptibleTime(TimeSegmentStartingDay:
(TimeSegmentLength+TimeSegmentStartingDay)));

I_S_N = (SumConfirmed.*SumSucceptible)./TotalPopulation;

figure(6)
plot(I_S_N,dSucceptibleTime,'o');
title({'China','Equation 1: Plot of # of a factor IS/N vs dSusceptible
 (Not Cumulative)'});
xlabel('ISN')
ylabel('dSusceptible (Not Cumulative)')

%Beta = -0.3492

I_S_N_Segment = (I_S_N(TimeSegmentStartingDay:(TimeSegmentLength
+TimeSegmentStartingDay)));
```



Published with MATLAB® R2020b

Code For France is Below

Table of Contents


```
% France %
AllConfirmed =
 readtable('time_series_covid19_confirmed_global_narrow.csv');
AllDead = readtable('time_series_covid19_deaths_global_narrow.csv');
AllRecovered =
 readtable('time_series_covid19_recovered_global_narrow.csv');

TotalPopulation = 6.527*10^7; % Assume we tested that amount of
 people.
TimeSegmentLength = 60; % Days
TimeSegmentStartingDay = 1;

FranceLowerBorderRange = 93601;
FranceUppperBorderRange = 102960;
```

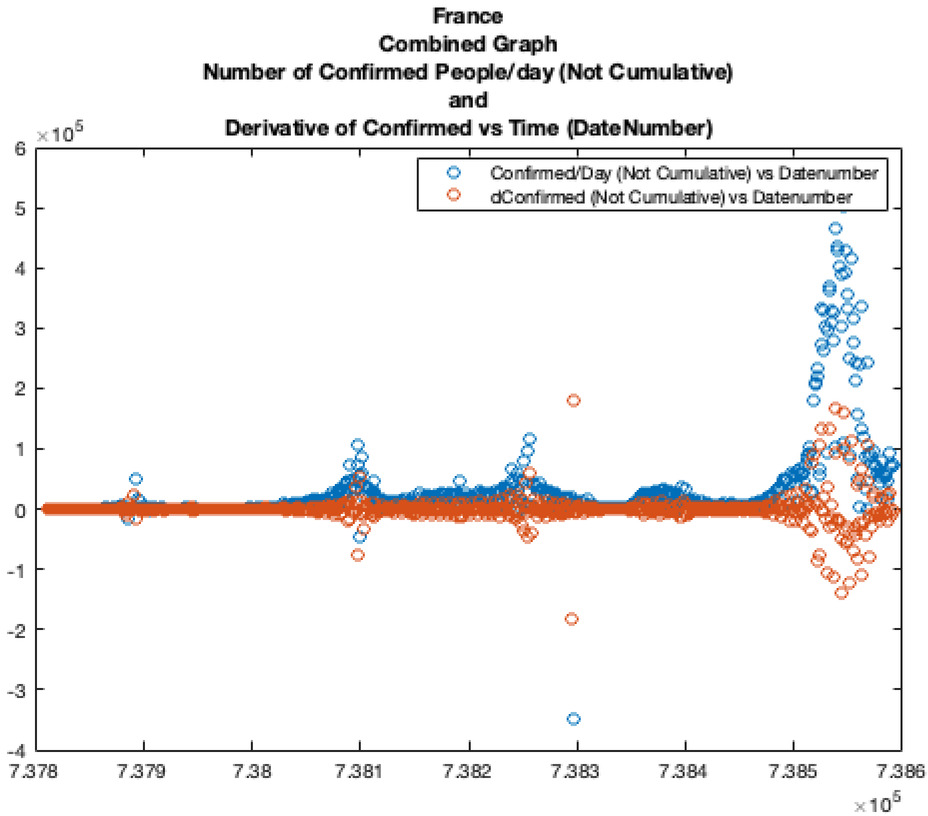
```
Warning: Column headers from the file were modified to make them valid
 MATLAB
identifiers before creating variable names for the table. The original
 column
headers are saved in the VariableDescriptions property.
Set 'VariableNamingRule' to 'preserve' to use the original column
 headers as
table variable names.
Warning: Column headers from the file were modified to make them valid
 MATLAB
identifiers before creating variable names for the table. The original
 column
headers are saved in the VariableDescriptions property.
Set 'VariableNamingRule' to 'preserve' to use the original column
 headers as
table variable names.
Warning: Column headers from the file were modified to make them valid
 MATLAB
identifiers before creating variable names for the table. The original
 column
headers are saved in the VariableDescriptions property.
```

```
Set 'VariableNamingRule' to 'preserve' to use the original column
 headers as
table variable names.
```

Confirmed = Infected

```
ConfirmedFrance =
 flip(AllConfirmed{FranceLowerBorderRange:FranceUppperBorderRange,6}); %
 Range of Confirmed in France {93601:102960,6}
ConfirmedFranceTime =
 flip(AllConfirmed{FranceLowerBorderRange:FranceUppperBorderRange,5}); %
 Range of Confirmed in France
DateNumber = datenum(ConfirmedFranceTime); % Time Range in Data Number

Unique = unique(DateNumber); % Number of unique days within which the
 measurements were taken.

[~,~,ix] = unique(DateNumber);
UniqueRepetitions = mean(accumarray(ix,1)); % How many times each
 DateNumber repeats

Length = length(Unique); % Number of Unique Days the measurements were
 taken
x = nan(UniqueRepetitions,Length-1);

for i=1:Length-1
r = DateNumber(1)+i;
a = find(DateNumber == r); % Index
x(:,i) = a;
end
x = [(find(DateNumber == r-(Length-1))) x]; % Indexes for 12 regions;
 780 days

w = 0;
for i=1:Length
    w = w+1;
    k = ConfirmedFrance(x(:,w));
    h(:,i) = k; % Actual number of Confirmed in each province of
 France
end

fckConfirmed = reshape(h,[UniqueRepetitions,Length]);
SumConfirmedCumulative = (sum(fckConfirmed))'; % Cumulative # of
 Confirmed/day in France (no more separation on provinces).

y1 = 0;
for i=1:Length-1
    y1 = y1+1;
    u1 = SumConfirmedCumulative(y1+1)-SumConfirmedCumulative(y1);
    j1(:,i) = u1;
end
j1 = [SumConfirmedCumulative(1) j1(1:Length-1)];
```

```
fckConfirmed2 = reshape(j1,[1,Length]);
SumConfirmed = (fckConfirmed2)'; % Number of Infected(Confirmed)
 people/day in France at a particular time (t). Not cumulative!

dConfirmedTime = gradient(SumConfirmed,Unique); % Derivative of the
 data to time

figure(1)
plot(Unique,SumConfirmed,'o');% Plot of Confirmed/Day vs Time
title({'France','Combined Graph','Number of Confirmed People/
day (Not Cumulative)', 'and', 'Derivative of Confirmed vs Time
 (DateNumber)'});
hold on
plot(Unique,dConfirmedTime,'o'); % Plot of Derivative of Data vs Time
legend({'Confirmed/Day (Not Cumulative) vs Datenumber','dConfirmed
 (Not Cumulative) vs Datenumber'})
hold off
```



Dead (Done)

```
DeadFrance =
 flip(AllDead{FranceLowerBorderRange:FranceUppperBorderRange,6}); %
 Data for # of Dead People in France {93601:102960,6}
DeadFranceTime =
 flip(AllDead{FranceLowerBorderRange:FranceUppperBorderRange,5}); %
 Data for dates (month/date/year) of Dead People in France
```

```
DateNumber = datenum(DeadFranceTime); % Time (month/date/year) Range
 in Data Number

Length = length(Unique); % Number of (Unique Data Numbers) in our
 Range = # of days the measurements were taken
x = nan(UniqueRepetitions(1),Length-1); % 12x779 Matrix of NaN

for i=1:Length-1
r = DateNumber(1)+i;
a = find(DateNumber == r); % Indexes (in DataNumber) for each Data
 Number. Each datanumber repeats 12 times - there are 12 provinces.
x(:,i) = a;
end
x = [(find(DateNumber == r-(Length-1))) x]; % Fills the matrix
 (12x780) with Indexes for each DataNumber (780 DataNumbers as total).
 780 Columns = 780 Days of Measurements. 12 Raws = 12 Provinces.

wDead = 0;
for i=1:Length
    wDead = wDead+1;
    kDead = DeadFrance(x(:,wDead)); % Take each column of x
 individually = take each day individually. And use this Indexes to
 find the #of associated people.
    hDead(:,i) = kDead; % Actual values of Dead (Now we have the
 matrix (12x780) with the actual number of Dead people instead of
 Indexes).
end

fckDead = reshape(hDead,[UniqueRepetitions,Length]); % Something with
 syntaxis
SumDeadCumulative = (sum(fckDead))'; % Cumulative of Dead/day in
 France (no more separation on provinces).

dDeadTime = gradient(SumDeadCumulative,Unique); % Derivative of the
 data (Dead People) with respect to time

% % %
figure(2)
plot(Unique,SumDeadCumulative,'o');% Plot of Dead/Day vs Time
title({'France','Combined Graph','Number of Dead People/day (Not
 Cumulative)','and','Derivative of Confirmed vs Time (DateNumber)'})
hold on
plot(Unique,dDeadTime,'o'); % Plot of Derivative of Data vs Time
legend({'Dead/Day (Cumulative) vs Datenumber','dDead (Cumulative) vs
 Datenumber'})
hold off

% % %
figure(3)
plot(SumConfirmed, dDeadTime,'o'); % Plot of # of Infected vs dDead.
 "sum1" is Infected people/day
title('Equation 4: Plot of # of Infected vs dDead');
xlabel('# of Infected')
ylabel('dDead')
```

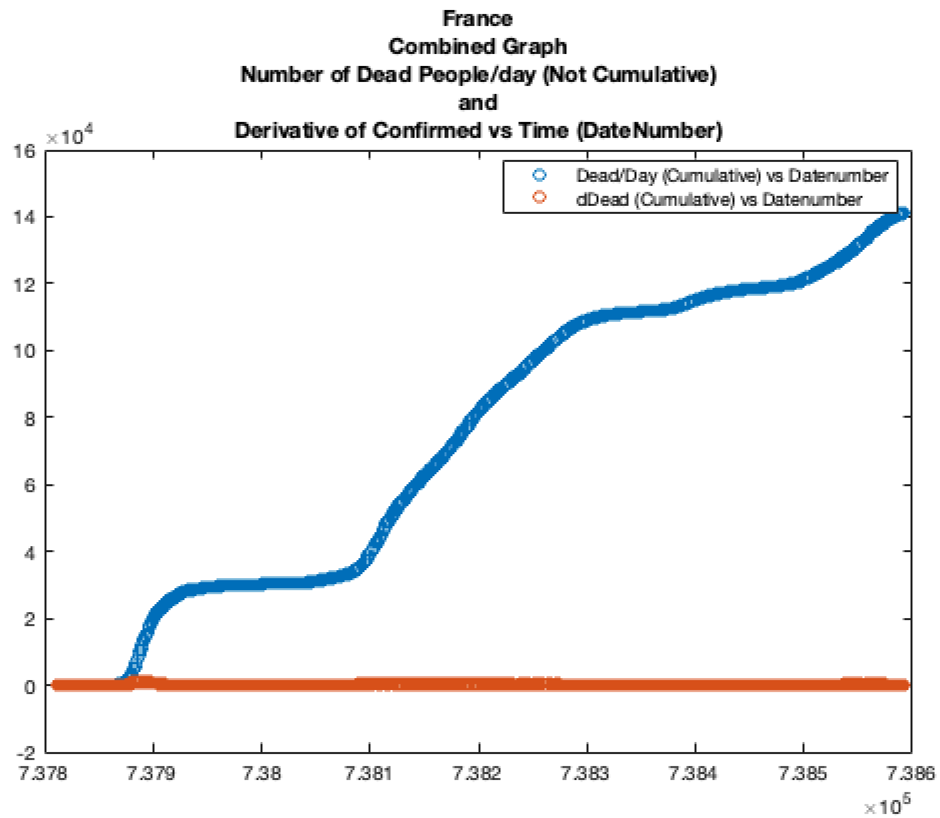
Note: the slope of each individual linear segment is going to be a

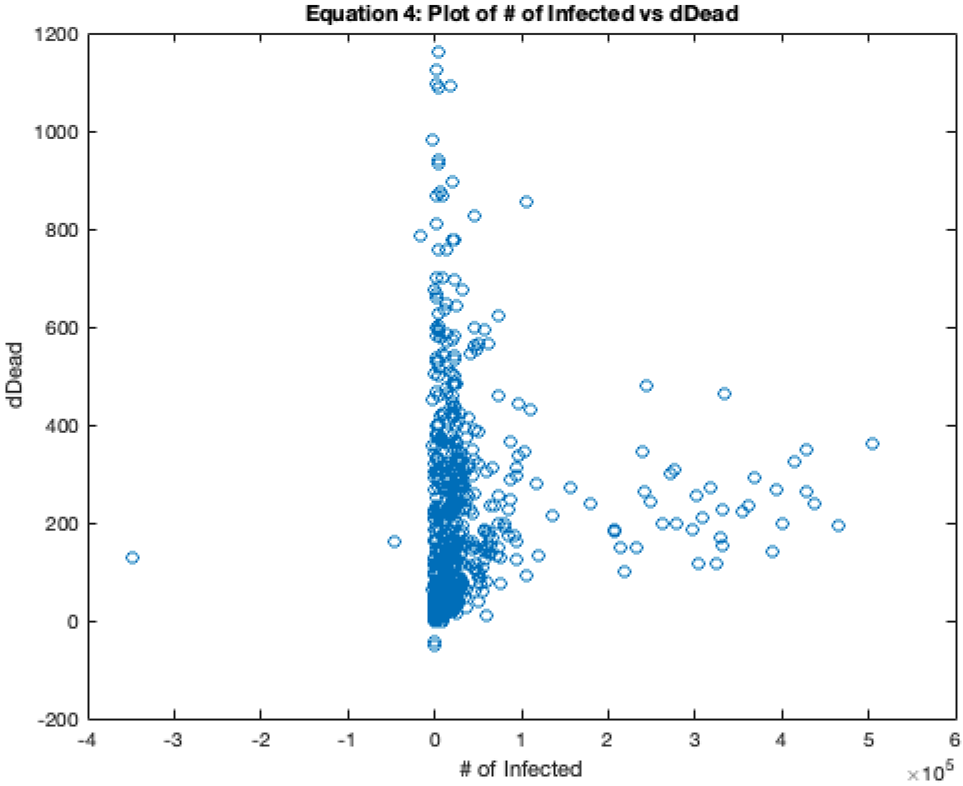
value of Kd. I'm going to take one of the date ranges where the slope behaves linearly and use its Kd value to try to predict the function behaviour for future days.

```
% % %
SumConfirmed_Segment = (SumConfirmed(1:TimeSegmentLength));
dDeadTime_Segment = (dDeadTime(1:TimeSegmentLength));
```

Note: Use the "Curve Fitting App (power = 1) to find the slope (p1), which is going to be equal to Kd

for this segment(1:60) Kd = 0.04624"





Recovered

```
RecoveredFrance =
 flip(AllRecovered{FranceLowerBorderRange:FranceUppperBorderRange,6}); %
 Data for # of Dead People in France {93601:102960,6}
RecoveredFranceTime =
 flip(AllRecovered{FranceLowerBorderRange:FranceUppperBorderRange,5}); %
 Data for dates (month/date/year) of Dead People in France
DateNumber = datenum(RecoveredFranceTime); % Time (month/date/year)
 Range in Data Number

Length = length(Unique); % Number of (Unique Data Numbers) in our
 Range = # of days the measurements were taken
x = nan(UniqueRepetitions(1),Length-1); % 12x779 Matrix of NaN

for i=1:Length-1
r = DateNumber(1)+i;
a = find(DateNumber == r); % Indexes (in DataNumber) for each Data
 Number. Each datanumber repeats 34 times - there are 34 provinces.
x(:,i) = a;
end
x = [(find(DateNumber == r-(Length-1))) x]; % Fills the matrix
 (12x780) with Indexes for each DataNumber (780 DataNumbers as total).
 780 Columns = 780 Days of Measurements. 12 Raws = 12 Provinces.
```

```
wRecovered = 0;
for i=1:Length
    wRecovered = wRecovered+1;
    kRecovered = RecoveredFrance(x(:,wRecovered)); % Take each column
 of x individually = take each day individually. And use this Indexes
 to find the #of associated people.
    hRecovered(:,i) = kRecovered; % Actual values of Dead (Now we have
 the matrix (12x780) with the actual number of Dead people instead of
 Indexes).
end

fckRecovered = reshape(hRecovered,[UniqueRepetitions,Length]); %
 Something with syntaxis
SumRecoveredCumulative = (sum(fckRecovered))'; % Cumulative of
 Recovered/day in France (no more separation on provinces).

dRecoveredTime = gradient(SumRecoveredCumulative,Unique); % Derivative
 of the data (Dead People) with respect to time

% % %
figure(4)
plot(Unique,SumRecoveredCumulative,'o');% Plot of Dead/Day vs Time
title({'France','Combined Graph','Number of Recovered People/day
 (Cumulative)','and','Derivative of Recovered (Cumulative) vs Time
 (DateNumber)'})
hold on
plot(Unique,dRecoveredTime,'o'); % Plot of Derivative of Data vs Time
legend({'Recovered/Day vs Datenumber','dRecovered vs Datenumber'})
hold off

% % %
figure(5)
plot(SumConfirmed, dRecoveredTime,'o');% Plot of # of Infected vs
 dDead. "sum1" is Infected people/day
title({'France','Equation 3: Plot of # of Infected (Not Cumulative) vs
 dRecovered (Cumulative)'});
xlabel('# of Infected (Not Cumulative)')
ylabel('dRecovered')
```

Note: the slope of each individual linear segment is going to be a

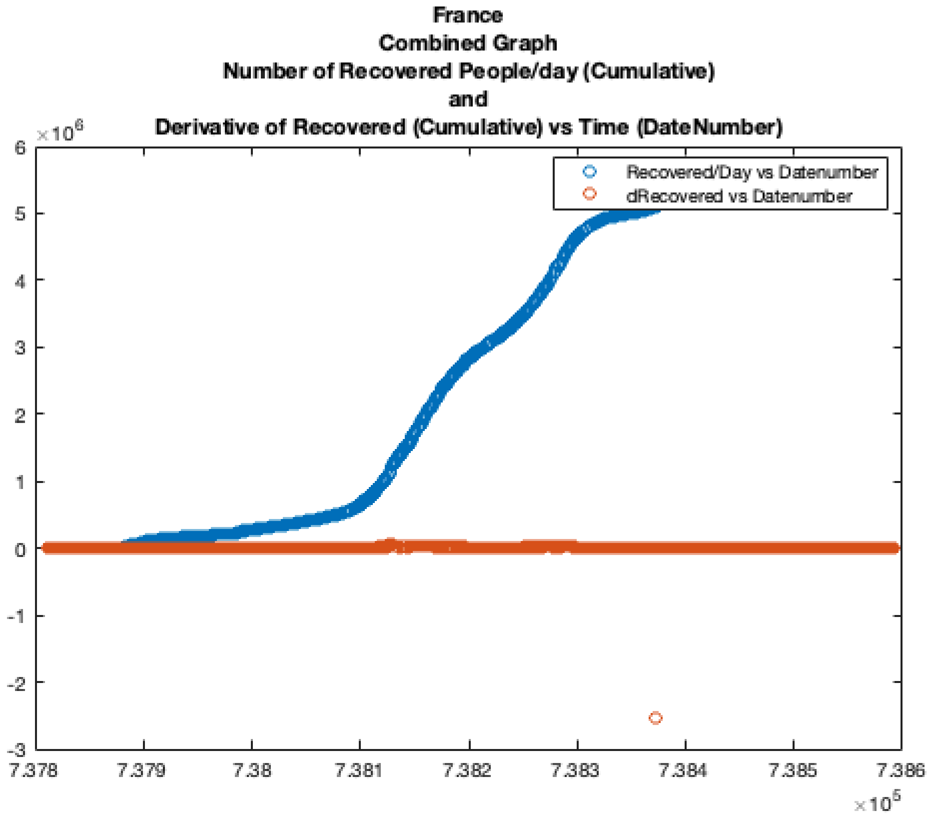
value of Kd. I'm going to take one of the date ranges where the slope behaves linearly and use its Kd value to try to predict the function behaviour for future days.

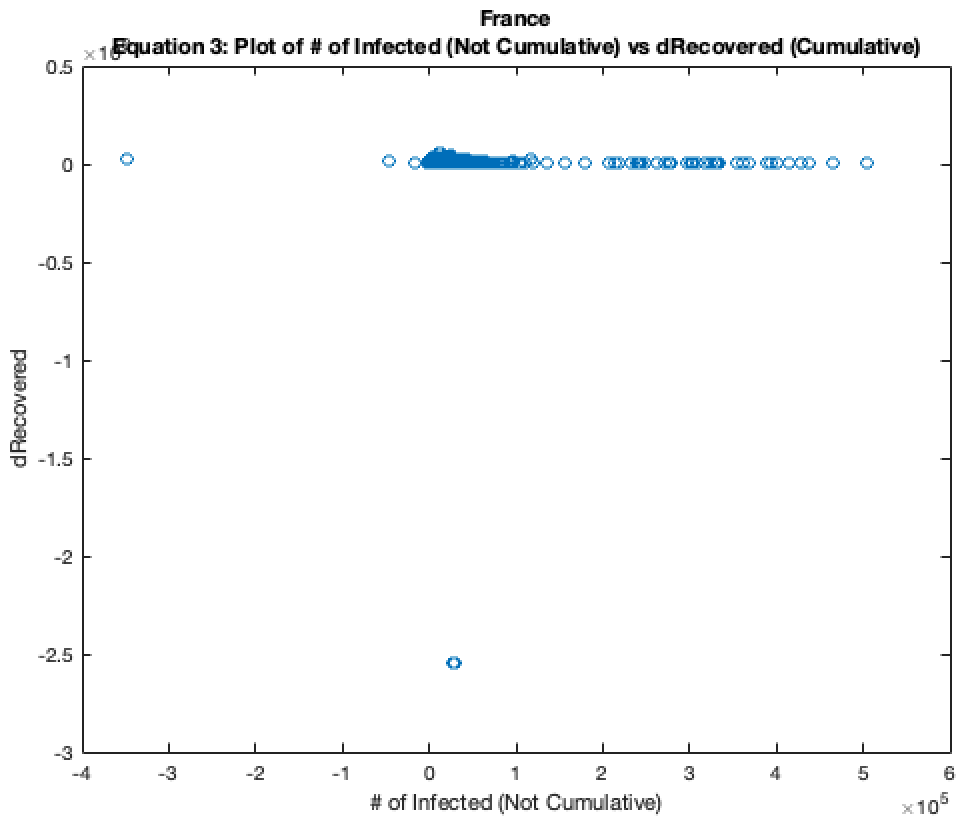
```
SumConfirmed_Segment = (SumConfirmed(1:TimeSegmentLength));
dRecoveredTime_Segment = (dRecoveredTime(1:TimeSegmentLength));
```

Note: Use the "Curve Fitting App (power = 1) to find the slope (p1),

which is going to be equal to Lambda

for this segment(1:60) Lambda = 0.01903"





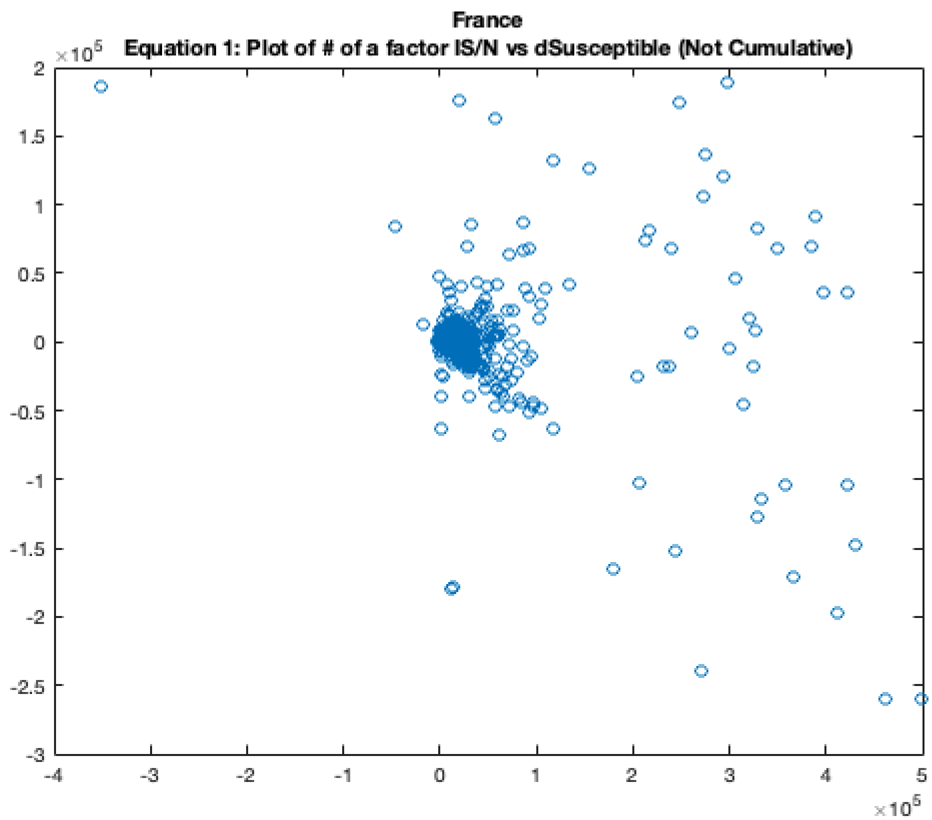
Succeptible

```
y2 = 0;
for i=1:Length-1
    y2 = y2+1;
    u2 = SumDeadCumulative(y2+1)-SumDeadCumulative(y2);
    j2(:,i) = u2;
end
j2 = [SumDeadCumulative(1) j2(1:Length-1)];

fckDead2 = reshape(j2,[1,Length]);
SumDead = (fckDead2)'; % Number of Dead people/day in France at a
 particular time (t). Not cumulative!


y3 = 0;
for i=1:Length-1
    y3 = y3+1;
    u3 = SumRecoveredCumulative(y3+1)-SumRecoveredCumulative(y3);
    j3(:,i) = u3;
end
j3 = [SumRecoveredCumulative(1) j1(1:Length-1)];

fckRecovered2 = reshape(j3,[1,Length]);
```

```
SumRecovered = (fckRecovered2)'; % Number of Recovered people/day in
 France at a particular time (t). Not cumulative!
```

Total Population

```
TotalPopulation = TotalPopulation.*ones(Length,1);
SumSucceptible = TotalPopulation-SumDead-SumRecovered-SumConfirmed; %
 # of succeptible people every day.

dSucceptibleTime = gradient(SumSucceptible,Unique); % Not Cumulative

SumSucceptible_Segment = (SumSucceptible(1:TimeSegmentLength));
dSucceptibleTime_Segment = (dSucceptibleTime(1:TimeSegmentLength));

I_S_N = (SumConfirmed.*SumSucceptible)./TotalPopulation;

figure(6)
plot(I_S_N,dSucceptibleTime,'o');
title({'France','Equation 1: Plot of # of a factor IS/N vs
 dSusceptible (Not Cumulative)'});

% Beta = -0.2734

I_S_N_Segment = (I_S_N(1:TimeSegmentLength));
```



Published with MATLAB® R2020b

Code For the USA is Below

Table of Contents

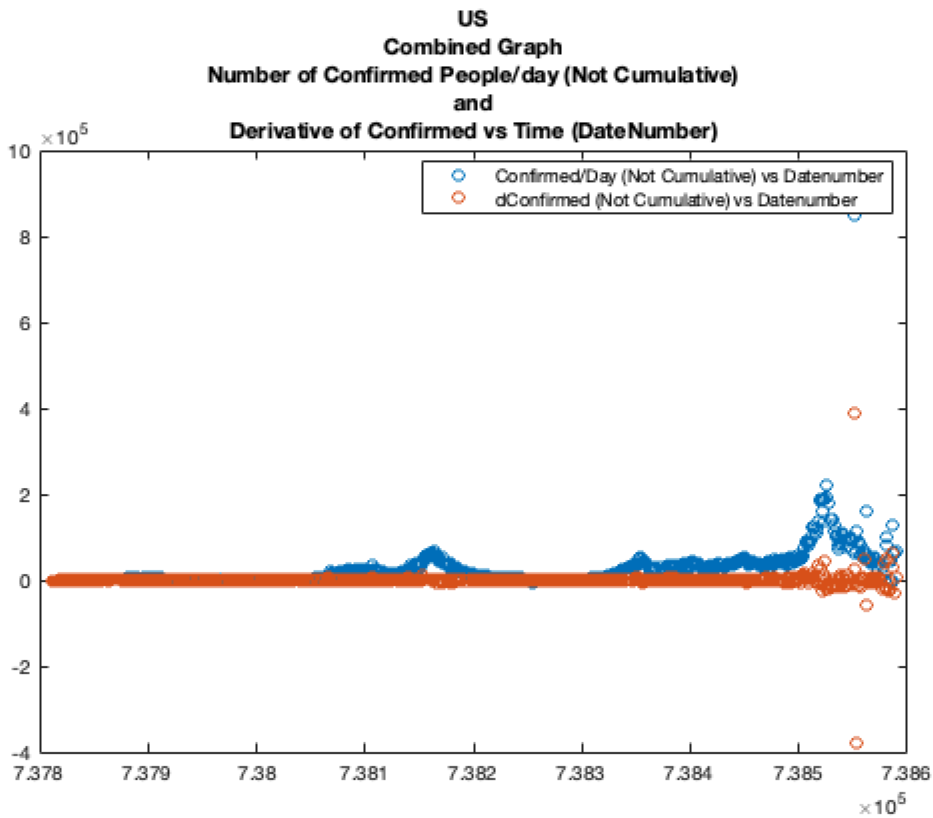

```
% US %
AllConfirmed =
 readtable('time_series_covid19_confirmed_global_narrow.csv');
AllDead = readtable('time_series_covid19_deaths_global_narrow.csv');
AllRecovered =
 readtable('time_series_covid19_recovered_global_narrow.csv');

TotalPopulation = 33.1*10^7; % Assume we tested that amount of people.
TimeSegmentLength = 60; % Days
TimeSegmentStartingDay = 1;

USLowerBorderRange = 202023; %202023
USUppperBorderRange = 202800; %202800
```

```
Warning: Column headers from the file were modified to make them valid
 MATLAB
identifiers before creating variable names for the table. The original
 column
headers are saved in the VariableDescriptions property.
Set 'VariableNamingRule' to 'preserve' to use the original column
 headers as
table variable names.
Warning: Column headers from the file were modified to make them valid
 MATLAB
identifiers before creating variable names for the table. The original
 column
headers are saved in the VariableDescriptions property.
Set 'VariableNamingRule' to 'preserve' to use the original column
 headers as
table variable names.
Warning: Column headers from the file were modified to make them valid
 MATLAB
identifiers before creating variable names for the table. The original
 column
headers are saved in the VariableDescriptions property.
```

```
Set 'VariableNamingRule' to 'preserve' to use the original column
 headers as
table variable names.
```

Confirmed = Infected

```
SumConfirmedCumulative =
 flip(AllConfirmed{USLowerBorderRange:USUppperBorderRange,6}); % Range
 of Confirmed in France {93601:102960,6}
ConfirmedUSTime =
 flip(AllConfirmed{USLowerBorderRange:USUppperBorderRange,5}); % Range
 of Confirmed in France
DateNumber = datenum(ConfirmedUSTime); % Time Range in Data Number
Unique = unique(DateNumber); % Number of unique days within which the
 measurements were taken.
Length = length(Unique); % Number of Unique Days the measurements were
 taken

y1 = 0;
for i=1:Length-1
    y1 = y1+1;
    u1 = SumConfirmedCumulative(y1+1)-SumConfirmedCumulative(y1);
    j1(:,i) = u1;
end
j1 = [SumConfirmedCumulative(1) j1(1:Length-1)];

fckConfirmed2 = reshape(j1,[1,Length]);
SumConfirmed = (fckConfirmed2)';

dConfirmedTime = gradient(SumConfirmed,Unique); % Derivative of the
 data to time

figure(1)
plot(Unique,SumConfirmed,'o');% Plot of Confirmed/Day vs Time
title({'US','Combined Graph','Number of Confirmed People/day
 (Not Cumulative)', 'and', 'Derivative of Confirmed vs Time
 (DateNumber)'});
hold on
plot(Unique,dConfirmedTime,'o'); % Plot of Derivative of Data vs Time
legend({'Confirmed/Day (Not Cumulative) vs Datenumber','dConfirmed
 (Not Cumulative) vs Datenumber'})
hold off
```



Dead (Done)

```
SumDeadCumulative =
 flip(AllDead{USLowerBorderRange:USUppperBorderRange,6}); % Data for #
 of Dead People in France {93601:102960,6}
DeadUSTime =
 flip(AllDead{USLowerBorderRange:USUppperBorderRange,5}); % Data for
 dates (month/date/year) of Dead People in France
DateNumber = datenum(DeadUSTime); % Time (month/date/year) Range in
 Data Number

dDeadTime = gradient(SumDeadCumulative,Unique); % Derivative of the
 data (Dead People) with respect to time

% % %
figure(2)
plot(Unique,SumDeadCumulative,'o');% Plot of Dead/Day vs Time
title({'US','Combined Graph','Number of Dead People/day (Not
 Cumulative)','and','Derivative of Confirmed vs Time (DateNumber)'})
hold on
plot(Unique,dDeadTime,'o'); % Plot of Derivative of Data vs Time
legend({'Dead/Day (Cumulative) vs Datenumber','dDead (Cumulative) vs
 Datenumber'})
hold off
```

```
%%
figure(3)
plot(SumConfirmed, dDeadTime,'o'); % Plot of # of Infected vs dDead.
 "sum1" is Infected people/day
title('Equation 4: Plot of # of Infected vs dDead');
xlabel('# of Infected')
ylabel('dDead')
```

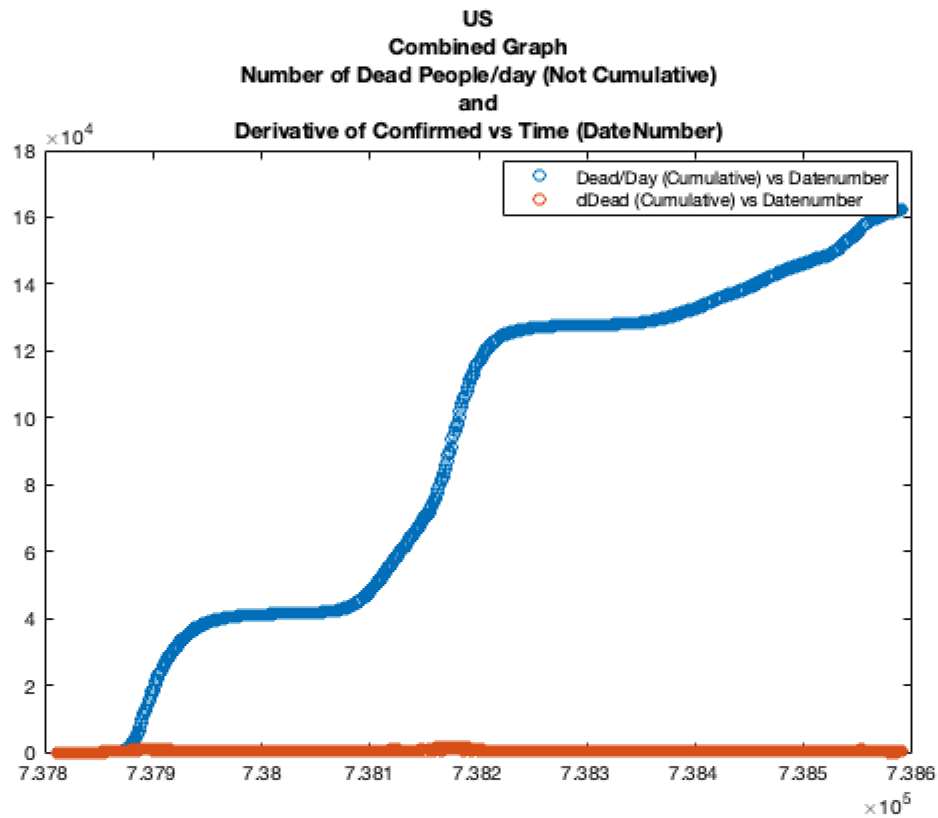
Note: the slope of each individual linear segment is going to be a

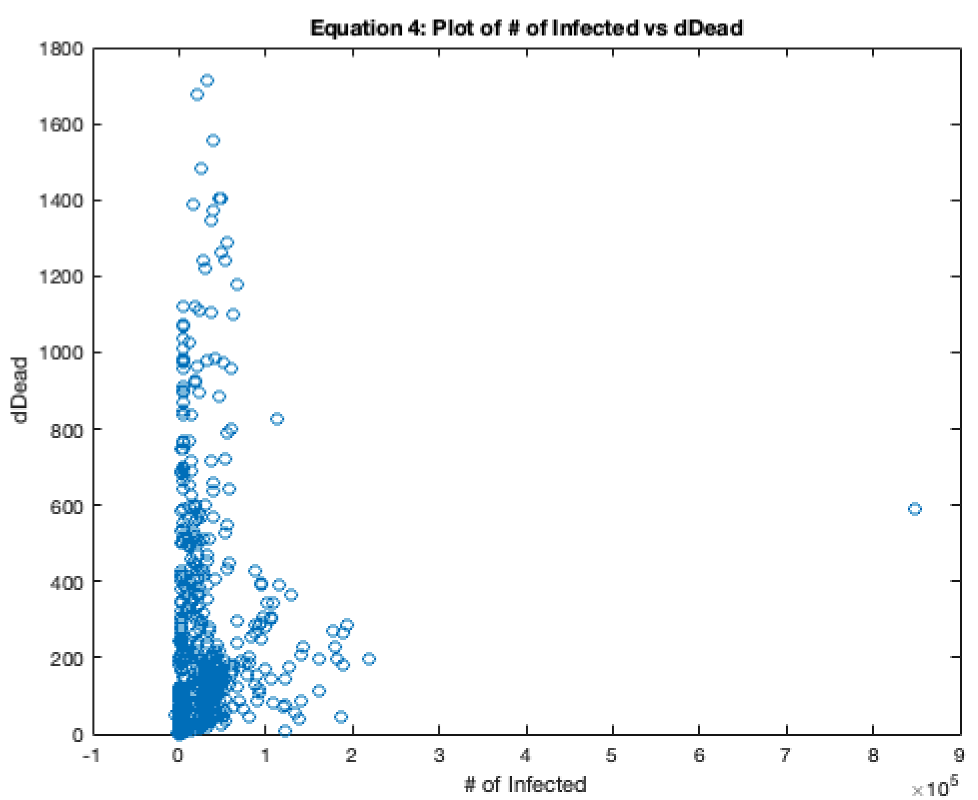
value of Kd. I'm going to take one of the date ranges where the slope behaves linearly and use its Kd value to try to predict the function behaviour for future days.

```
%%
SumConfirmed_Segment = (SumConfirmed(TimeSegmentStartingDay:
(TimeSegmentLength+TimeSegmentStartingDay)));
dDeadTime_Segment = (dDeadTime(TimeSegmentStartingDay:
(TimeSegmentLength+TimeSegmentStartingDay)));
```

Note: Use the "Curve Fitting App (power = 1) to find the slope (p1), which is going to be equal to Kd

for this segment(1:60) Kd = 0.01899"


US
Combined Graph
Number of Dead People/day (Not Cumulative)
and
Derivative of Confirmed vs Time (DateNumber)
×10^4
18
16
14
12
10
8
6
4
2
0
Dead/Day (Cumulative) vs Datenumber
dDead (Cumulative) vs Datenumber
7.378
7.379
7.38
7.381
7.382
7.383
7.384
7.385
7.386
×10^5


Equation 4: Plot of # of Infected vs dDead
dDead
1800
1600
1400
1200
1000
800
600
400
200
0
-1
0
1
2
3
4
5
6
7
8
9
# of Infected
×10^5

Recovered

```
SumRecoveredCumulative =
 flip(AllRecovered{USLowerBorderRange:USUppperBorderRange,6}); % Data
 for # of Dead People in France {93601:102960,6}
RecoveredUSTime =
 flip(AllRecovered{USLowerBorderRange:USUppperBorderRange,5}); % Data
 for dates (month/date/year) of Dead People in France
DateNumber = datenum(RecoveredUSTime); % Time (month/date/year) Range
 in Data Number

Length = length(Unique); % Number of (Unique Data Numbers) in our
 Range = # of days the measurements were taken

dRecoveredTime = gradient(SumRecoveredCumulative,Unique); % Derivative
 of the data (Dead People) with respect to time

% % %
figure(4)
plot(Unique,SumRecoveredCumulative,'o');% Plot of Dead/Day vs Time
title({'US','Combined Graph','Number of Recovered People/day
 (Cumulative)','and','Derivative of Recovered (Cumulative) vs Time
 (DateNumber)'})
hold on
plot(Unique,dRecoveredTime,'o'); % Plot of Derivative of Data vs Time
legend({'Recovered/Day vs Datenumber','dRecovered vs Datenumber'})
hold off

% % %
figure(5)
plot(SumConfirmed, dRecoveredTime,'o');% Plot of # of Infected vs
 dDead. "sum1" is Infected people/day
title({'US','Equation 3: Plot of # of Infected (Not Cumulative) vs
 dRecovered (Cumulative)'});
xlabel('# of Infected (Not Cumulative)')
ylabel('dRecovered')
```

Note: the slope of each individual linear segment is going to be a

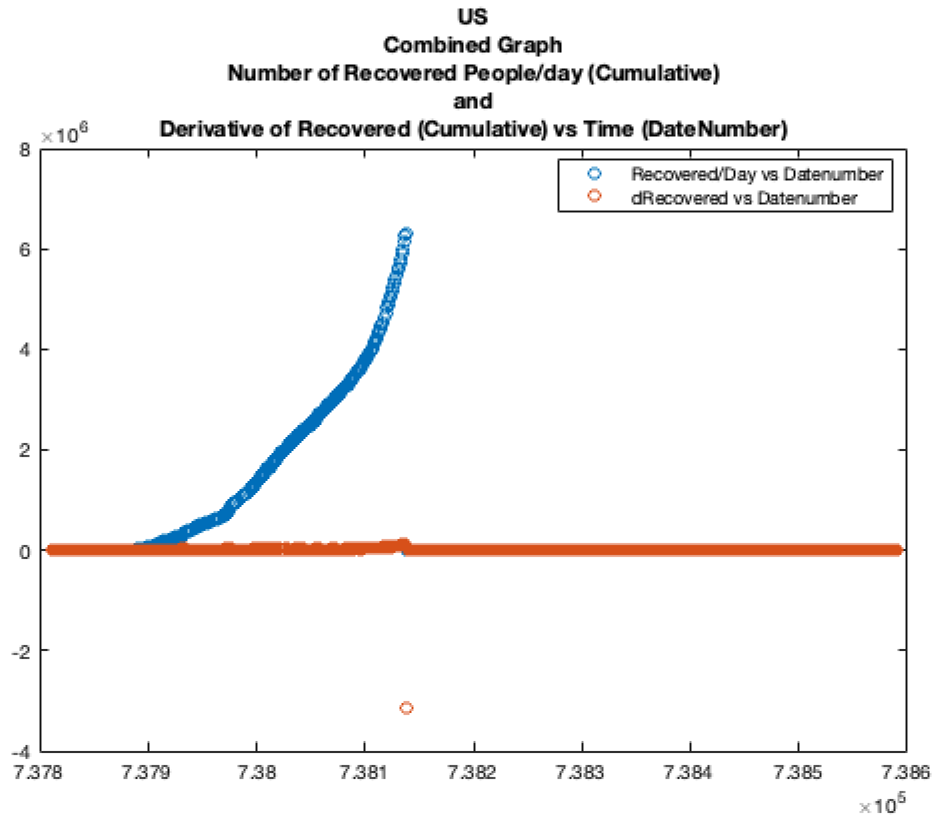
value of Kd. I'm going to take one of the date ranges where the slope behaves linearly and use its Kd value to try to predict the function behaviour for future days.

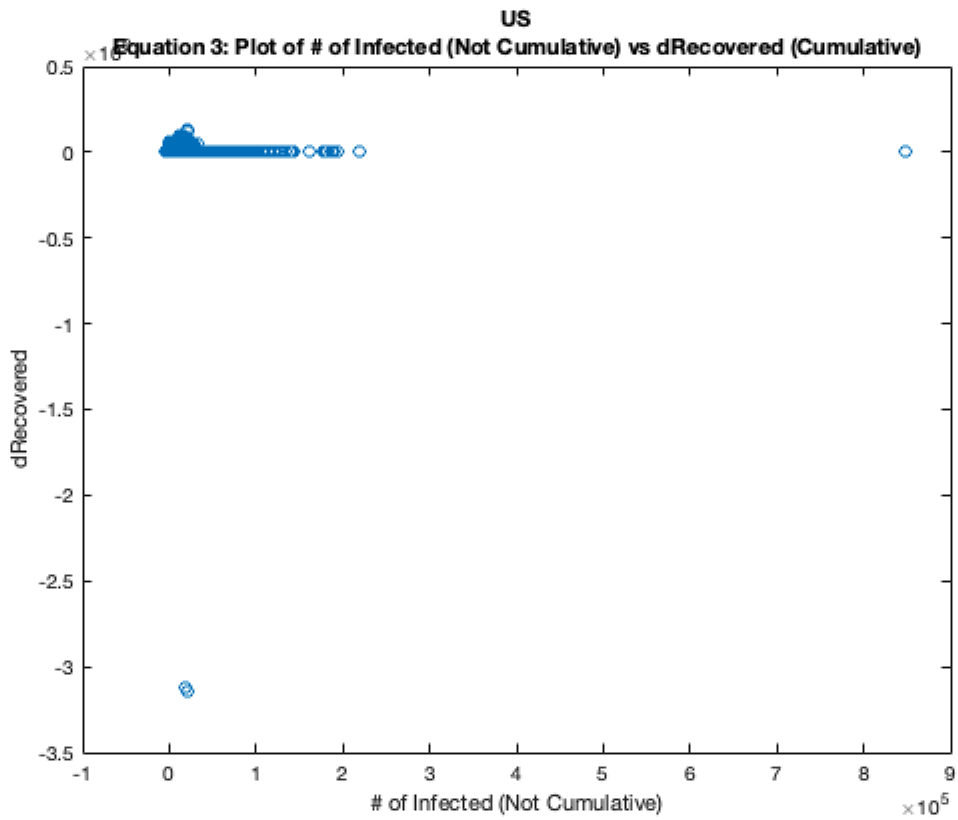
```
SumConfirmed_Segment = (SumConfirmed(TimeSegmentStartingDay:
(TimeSegmentLength+TimeSegmentStartingDay)));
dRecoveredTime_Segment = (dRecoveredTime(TimeSegmentStartingDay:
(TimeSegmentLength+TimeSegmentStartingDay)));
```

Note: Use the "Curve Fitting App (power = 1) to find the slope (p1),

which is going to be equal to Lambda

for this segment(1:60) Lambda = 0.01161"





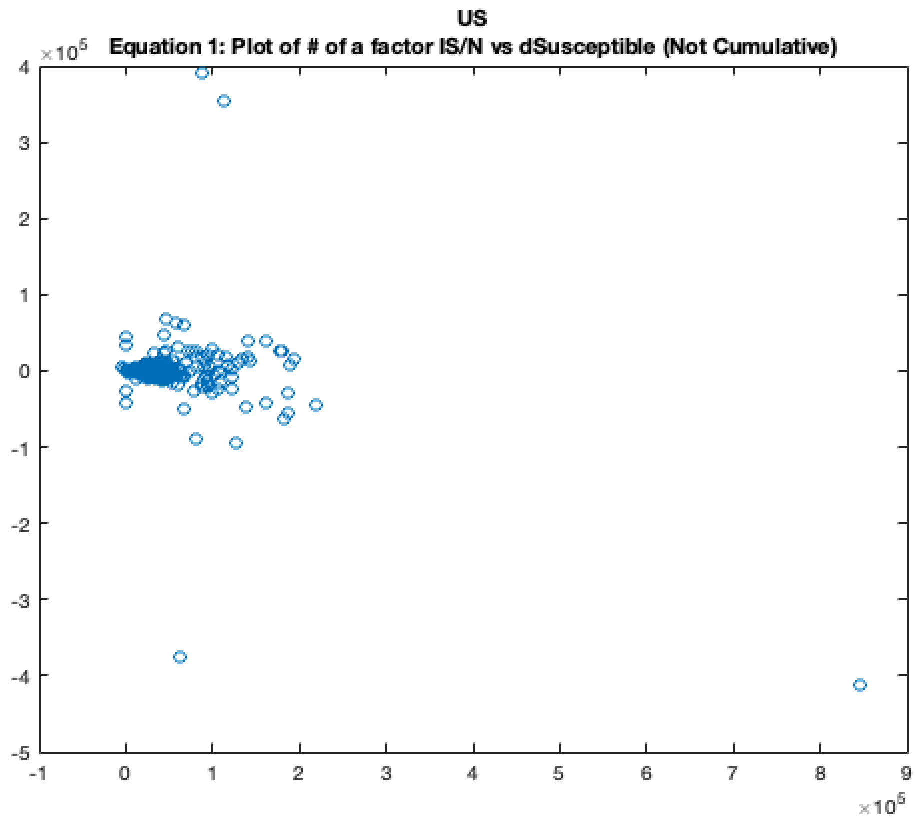
Succeptible

```
y2 = 0;
for i=1:Length-1
    y2 = y2+1;
    u2 = SumDeadCumulative(y2+1)-SumDeadCumulative(y2);
    j2(:,i) = u2;
end
j2 = [SumDeadCumulative(1) j2(1:Length-1)];

fckDead2 = reshape(j2,[1,Length]);
SumDead = (fckDead2)'; % Number of Dead people/day in France at a
 particular time (t). Not cumulative!


y3 = 0;
for i=1:Length-1
    y3 = y3+1;
    u3 = SumRecoveredCumulative(y3+1)-SumRecoveredCumulative(y3);
    j3(:,i) = u3;
end
j3 = [SumRecoveredCumulative(1) j1(1:Length-1)];

fckRecovered2 = reshape(j3,[1,Length]);
```

```
SumRecovered = (fckRecovered2)'; % Number of Recovered people/day in
 France at a particular time (t). Not cumulative!
```

Total Population

```
TotalPopulation = TotalPopulation.*ones(Length,1);
SumSucceptible = TotalPopulation-SumDead-SumRecovered-SumConfirmed; %
 # of succeptible people every day.

dSucceptibleTime = gradient(SumSucceptible,Unique); % Not Cumulative

SumSucceptible_Segment = (SumSucceptible(TimeSegmentStartingDay:
(TimeSegmentLength+TimeSegmentStartingDay)));
dSucceptibleTime_Segment = (dSucceptibleTime(TimeSegmentStartingDay:
(TimeSegmentLength+TimeSegmentStartingDay)));

I_S_N = (SumConfirmed.*SumSucceptible)./TotalPopulation;

figure(6)
plot(I_S_N,dSucceptibleTime,'o');
title({'US','Equation 1: Plot of # of a factor IS/N vs dSusceptible
 (Not Cumulative)'});

% Beta = -0.2405

I_S_N_Segment = (I_S_N(TimeSegmentStartingDay:(TimeSegmentLength
+TimeSegmentStartingDay)));
```



Published with MATLAB® R2020b