Motivation:

I recently found population density available from the EU (https://ghsl.jrc.ec.europa.eu/download.php?ds=pop (https://ghsl.jrc.ec.europa.eu/download.php?ds=pop)) and wanted to use it in a project. With the coronavirus pandemic on everyone's mind at the minute, I saw an opportunity to try and do a model of the coronavirus spread in Ireland using this population density information. I have not been able to find any similar analyses online.

Data Reduction:

I was looking to do a simple model since much of the data for the coronavirus I would be using in any case would be quite uncertain given how new the disease was. As a result, I firstly removed all regions other than Ireland (I did my analysis on the entire island) and I decreased the population density resolution from 1km to 4km.

I then assumed that rural areas have negligible probability of contracting the disease due to their isolation, and removed them. I defined a rural area as any region with a population density of less than 2500 people per 16km^2.

The total remaining population was 4074762.8577740956 from the original 6551319.200469393 (which is very close the accepted value for the population). I conducted some checks and found that the population of Galway city after my desampling was 81887.163439935, which is very close once again to the known value. Thus, I was content that this left me with satisfactory results. After this step I had a far more manageable 374 locations with non-negligible population densities, compared to over 200^2 as I had in the beginning.

Model:

Taking inspiration from the Kormack-McKendrick Model as covered in classwork, I have subdivided the population into 5 groups: Infected (I), Susceptible (S), Exposed (E), Dead (D) and Recovered (R) (I toyed with the idea of having a Quarantined (Q) class but this did not add much value i.e. could essentially be achieved by changing existing constants). Recovered people were assumed to be immune to reinfection, thought this could be easily changed. The E class was required simply for the model to be able to fit to data i.e. with no E model, it was found that the $t = 0$ zero infection would be very low and would not fit the data.

For the model, there were a number of decisions to make. I firstly decided to go with a continuous model since this was easiest, should yield acceptable results and could be rethought later if needs be. I needed to simulate the spread of the disease by the movement of people. To do this, I could either change the populations for each time or I could simulate it using a formula. I went with a formula since moving people would have been computationally intensive.

I have seen gravity-like models used to model the movement of people in mobs or the number of burgularies in city regions. Since I wanted to simulate the movement of people without actually changing any of the population densities I had obtained, a model like this struck me as a good idea - I needed far away regions to be less affected but at the same time I needed dense regions to be more affected. Using this model, one obtains that the number of new exposed people in region j due to the infected population in i would be $E_{new}(i \to j) = \alpha \frac{S_j I_i}{r^2}$. In the case that region i infected itself, then $r = 0$ and $E_{new}(i \to i) = \alpha_0 S_i I_i$. I later added functionality for an optional threshold on the possible number of infections by a single person, in case this was required to better fit the data. The number of new infected would just be obtained from the E class i.e. $I_{new} = \kappa E$. Members of the E class could also return to the S class i.e. $S_{new} = \gamma E$. The number of dead and recovered could only come from the I class i.e. $D_{new} = \beta I$ and $R_{new} = \sigma I$.

That is, the entire model is:

$$E_{new}(i \to j) = \alpha \frac{S_j I_i}{r^2}$$

if r=0, then

$$E_{new}(i \to i) = \alpha_0 S_i I_i$$

with an optional threshold for S

$$\begin{aligned} I_{new} &= \kappa E \\ S_{new} &= \gamma E \\ D_{new} &= \beta I \\ R_{new} &= \sigma I \end{aligned}$$

Determination of Constants:

I had a model, but now need to obtain the constants for it. This was quite difficult since there were many differing values depending on the region examined, but I was lucky in that there were many papers online for me to view (WHO's database https://www.who.int/emergencies/diseases/novel-coronavirus-2019/global-research-on-novel-coronavirus-2019-ncov (https://www.who.int/emergencies/diseases/novel-coronavirus-2019/global-research-on-novel-coronavirus-2019-ncov) was particularly helpful). Each constant's determination requires some explanation so I shall do them each in term.

- $\beta \approx 0.001550098$:

The death rate per infection or case fatality rate (CFR) had a great deal of data on it. However, it was recorded to vary greatly from country to country.

From a report of 72 314 Cases from the Chinese Center for Disease Control and Prevention (see https://jamanetwork.com/journals/jama/fullarticle/2762130 (https://jamanetwork.com/journals/jama/fullarticle/2762130), http://weekly.chinacdc.cn/en/article/id/e53946e2-c6c4-41e9-9a9b-fea8db1a8f51 (http://weekly.chinacdc.cn/en/article/id/e53946e2-c6c4-41e9-9a9b-fea8db1a8f51)), the CFR was estimated to be 2.3% in China.

In Italy, a report from 35731 cases by the Istituto Superiore di Sanità (see https://www.epicentro.iss.it/coronavirus/bollettino/Bollettino%20sorveglianza%20integrata%20COVID-19_19-marzo%202020.pdf (https://www.epicentro.iss.it/coronavirus/bollettino/Bollettino%20sorveglianza%20integrata%20COVID-19_19-marzo%202020.pdf)) put the lethality of the virus at 8.5%.

As of March 20, the Center for Disease Control and Prevention in the US reported 15,219 cases and 201 deaths, putting the crude mortality ratio in the region of 1.3%. The CFR should be higher considering deaths do not occur until numerous days after the diagnosis of a case.

In Switzerland, the number of cases reported by l'Office Fédéral de la Santé Publique as of Monday the 23rd of March was 8060, with 66 deaths (see https://www.rts.ch/info/suisse/11185998-plus-de-8000-cas-et-66-deces-en-suisse-selon-un-nouveau-bilan-de-l-ofsp.html (https://www.rts.ch/info/suisse/11185998-plus-de-8000-cas-et-66-deces-en-suisse-selon-un-nouveau-bilan-de-l-ofsp.html)). This puts the crude mortality ratio in the region of 0.8%.

The World Health Organisation (WHO) reported on the 6th of March that the crude mortality ratio (the number of reported deaths divided by the reported cases) was between 3-4% (see https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200306-sitrep-46-covid-19.pdf?sfvrsn=96b04adf_2 (https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200306-sitrep-46-covid-19.pdf?sfvrsn=96b04adf_2)).

Quoting from a report from WHO (https://www.who.int/docs/default-source/coronaviruse/who-china-joint-mission-on-covid-19-final-report.pdf (https://www.who.int/docs/default-source/coronaviruse/who-china-joint-mission-on-covid-19-final-report.pdf)):

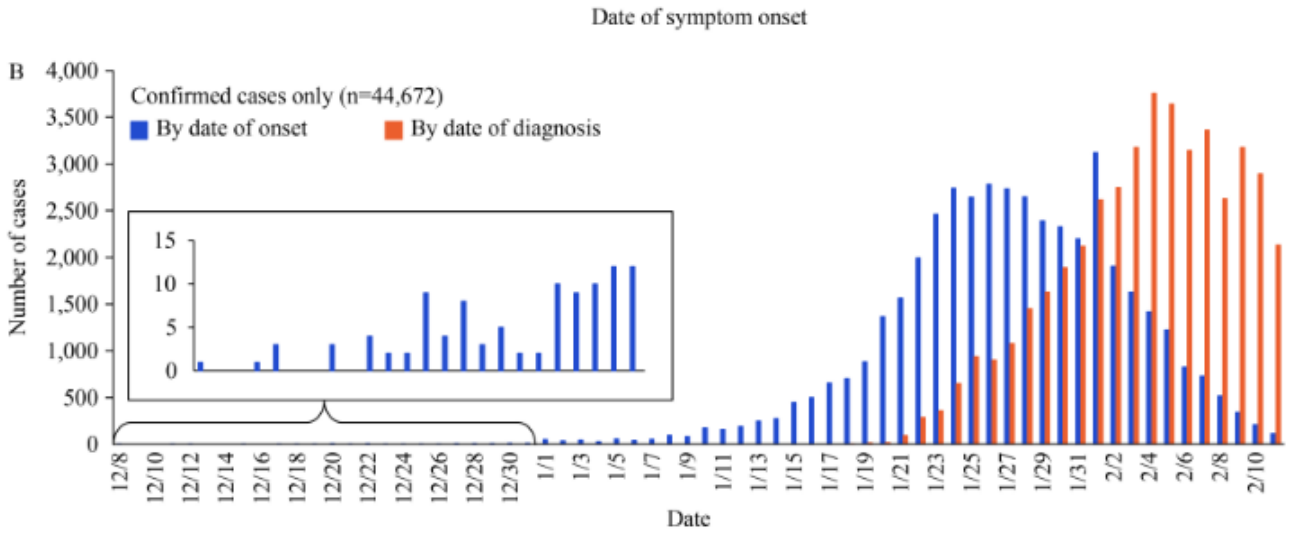
"As of 20 February, 2114 of the 55,924 laboratory confirmed cases have died (crude fatality ratio [CFR2] 3.8%) (note: at least some of whom were identified using a case definition that included pulmonary disease). The overall CFR varies by location and intensity of transmission (i.e. 5.8% in Wuhan vs. 0.7% in other areas in China). In China, the overall CFR was higher in the early stages of the outbreak (17.3% for cases with symptom onset from 1- 10 January) and has reduced over time to 0.7% for patients with symptom onset after 1 February (Figure 4)."

There are many more examples I can take but, in short, the CFR varies a great deal from population to population. The primary causes for this are healthcare services and population demographics. For example, one of the main reasons why CFR rate is so high in Italy since the country has an old population. As a result, to find the CFR in Ireland, I had to take population demographics into account. I made the initial assumption that the healthcare services are comparable between Ireland and Italy, Spain, France, Germany, the Netherlands, China and South Korea - the locations for which I successfully obtained data yielding the lethality rate per age group (see http://weekly.chinacdc.cn/en/article/id/e53946e2-c6c4-41e9-9a9b-fea8db1a8f51 (http://weekly.chinacdc.cn/en/article/id/e53946e2-c6c4-41e9-9a9b-fea8db1a8f51), https://link.springer.com/article/10.1007/s00134-020-05991-x (https://link.springer.com/article/10.1007/s00134-020-05991-x), https://www.cdc.go.kr/board/board.es?mid=a30402000000&bid=0030&act=view&list_no=366621&tag=&nPage=1 (https://www.cdc.go.kr/board/board.es?mid=a30402000000&bid=0030&act=view&list_no=366621&tag=&nPage=1), https://www.epicentro.iss.it/coronavirus/bollettino/Bollettino%20sorveglianza%20integrata%20COVID-19_19-marzo%202020.pdf (https://www.epicentro.iss.it/coronavirus/bollettino/Bollettino%20sorveglianza%20integrata%20COVID-19_19-marzo%202020.pdf), https://www.santepubliquefrance.fr/maladies-et-traumatismes/maladies-et-infections-respiratoires/infection-a-coronavirus/documents/bulletin-national/covid-19-point-epidemiologique-du-15-mars-2020 (https://www.santepubliquefrance.fr/maladies-et-traumatismes/maladies-et-infections-respiratoires/infection-a-coronavirus/documents/bulletin-national/covid-19-point-epidemiologique-du-15-mars-2020), https://www.rivm.nl/documenten/epidemiologische-situatie-covid-19-in-nederland-23-maart-2020-0 (https://www.rivm.nl/documenten/epidemiologische-situatie-covid-19-in-nederland-23-maart-2020-0), https://www.ortenaukreis.de/media/custom/2390_5025_1.PDF?1584540677 (https://www.ortenaukreis.de/media/custom/2390_5025_1.PDF?1584540677) and https://www.mscbs.gob.es/profesionales/saludPublica/ccayes/alertasActual/nCov-China/documentos/Actualizacion_53_COVID-19.pdf (https://www.mscbs.gob.es/profesionales/saludPublica/ccayes/alertasActual/nCov-China/documentos/Actualizacion_53_COVID-19.pdf)).

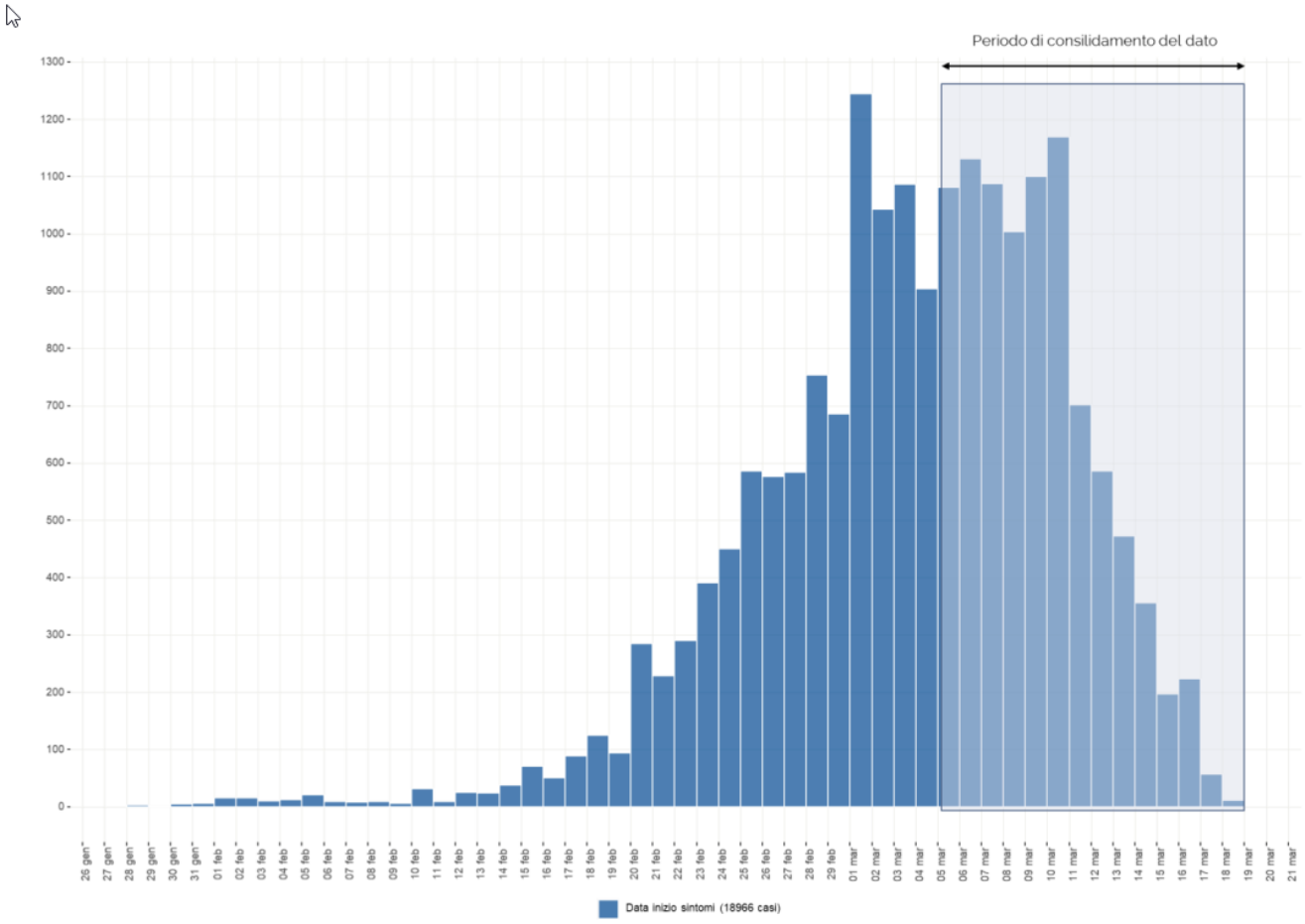
Since I could easily obtain census data (https://www.cso.ie/en/releasesandpublications/ep/p-cp3oy/cp3/assr/ (https://www.cso.ie/en/releasesandpublications/ep/p-cp3oy/cp3/assr/)) to get the age distribution of the population in 2011 and 2016, I decided it was best to make use of this data. Using the census data from 2011 and 2016, I made an estimate of the population demographics in 2020 by assuming the change in an age bracket would the same over the same time period (i.e. four fifths of the percentage decrease from the number of 14 year olds in 2011 to the number of 19 year olds in 2016 would be the same as the percentage change in the number of 14 year olds in 2016 to the number of 18 year olds in 2020). Lastly, I can use my lethality rates per age group to find the overall best value for β for my model.

I ended up doing these in Excel simply because it was quickest. I decided to mostly ignore the data from Asia since the rates for Korea were considerably different to those in Europe and my data for China was quite old. I also ended up ignoring the data from France and the partial German dataset since they simply did not fit the granularity of the other sets and were quite small samples in comparison. The value I come out with using

the amalgamated data from Spain, Italy and the Netherlands and the estimated Irish population demographics for 2020 is 0.022957524 for the CFR. That is, I would expect 2.2957524% of people infected to die from the coronavirus in Ireland. To calculate β, I need a time dependence. As done in http://weekly.chinacdc.cn/en/article/id/e53946e2-c6c4-41e9-9a9b-fea8db1a8f51 (http://weekly.chinacdc.cn/en/article/id/e53946e2-c6c4-41e9-9a9b-fea8db1a8f51), the best way to do this is to look at the person-days observed i.e. sum the number of days each individual has been infected. The person-days observed in the study were 661609, and the distribution of cases looked as follows:

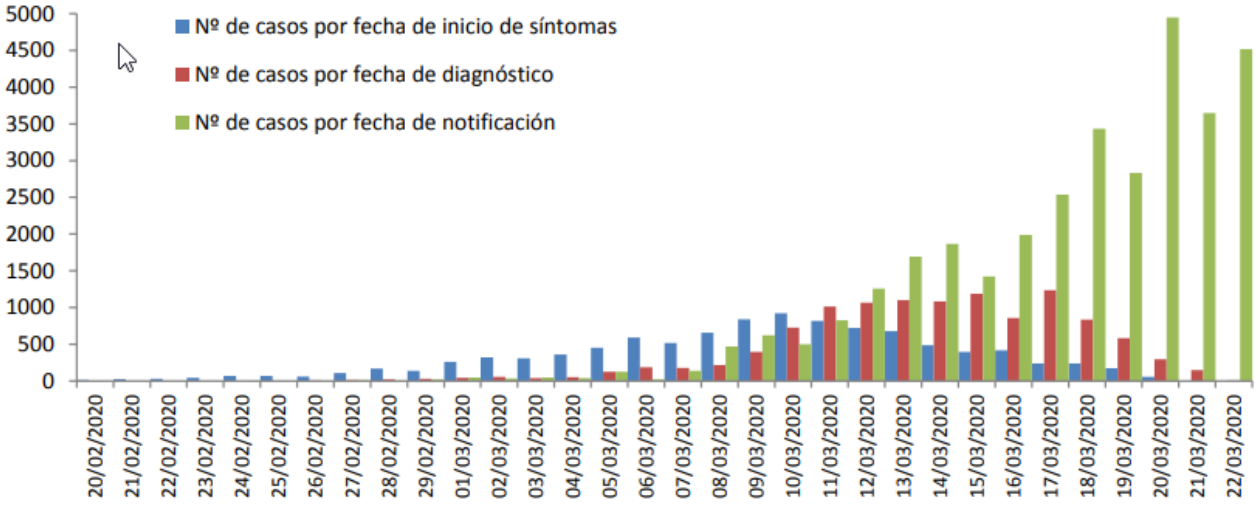


It is normally distributed with a mean date of about the 28th of January. This gives a mean number of person days of 14. The number of cases is 44672, and we duly see that 44672 by 14 is 625408 $\approx$ 661609 (error of 5.5%). We are dealing with approximate values at best in any case. Now, looking at the sources for our amalgamated data will be helpful. The Italian dataset as shown below is also approximately normally distributed.



with a mean date of the approximately the 5th of March, corresponding to a mean of 14 person-days also. The data from Spain tells a very similar story:

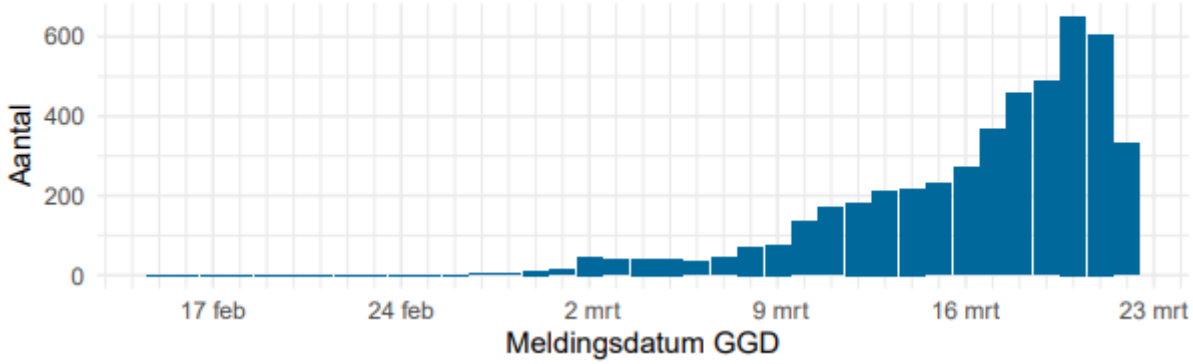
Figura 1. Casos diarios confirmados de COVID-19 por fecha de notificación, diagnóstico e inicio de síntomas.



It has a mean date of about the 9th of March, corresponding to a mean of 14 person-days. The dataset for the Netherlands does not have the date when symptoms begin. It does have the number of new cases, however, and if we look at the distribution:

Aantal bij de GGD'en gemelde COVID-19 patiënten, naar meldingsdatum

Meldingen tot en met 22-03-2020.

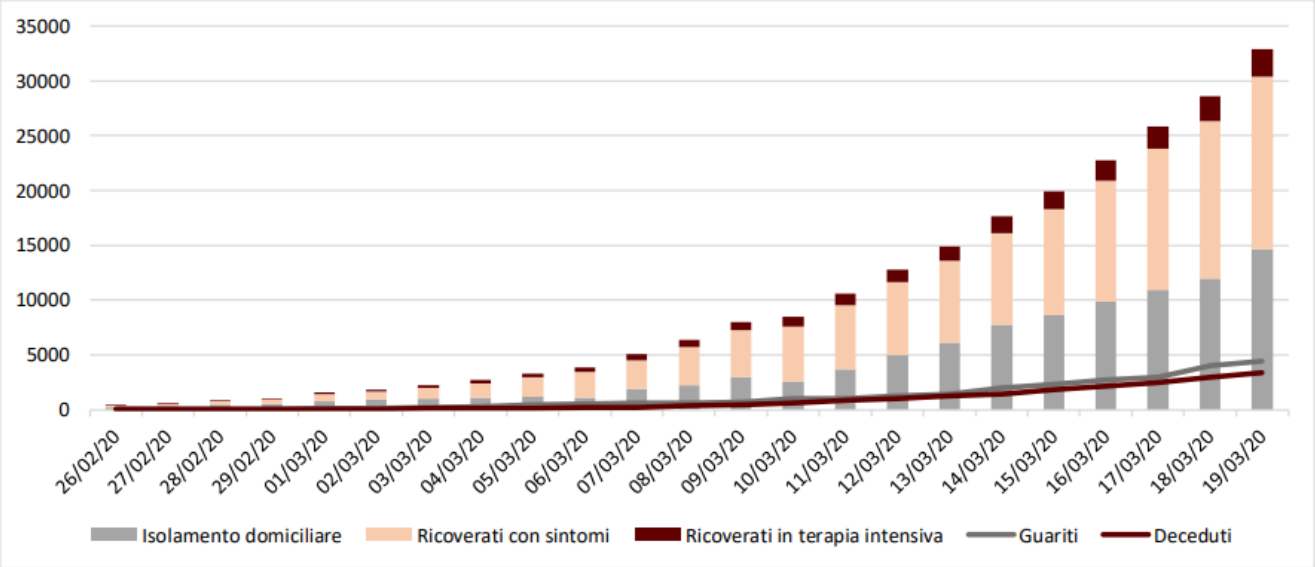


and compare it to that of Italy, Spain and China, we see it is extremely similar.

As a result, I would expect it to have a mean number of about 14 person days also. In short, the amalgamated datasets have the same mean number of person days as the Chinese one and all sets are approximately normally distributed. This means that 44672 cases in any of these regions should mean 661609 person-days. Thus, the CFR we computed i.e. 0.022957524 can now be converted into a time dependent one. The average number of hours per patient is $\frac{661609}{44672} = 14.81037339$. Thus the CFR is actually telling us that 0.022957524 of the infected die with an average of 14.81037339 personal days of being observed. We want the death rate per day, so we obtain this by $\beta \approx \frac{0.022957524}{14.81037339} \approx 0.001550098$ deaths from 100 infected per person-day.

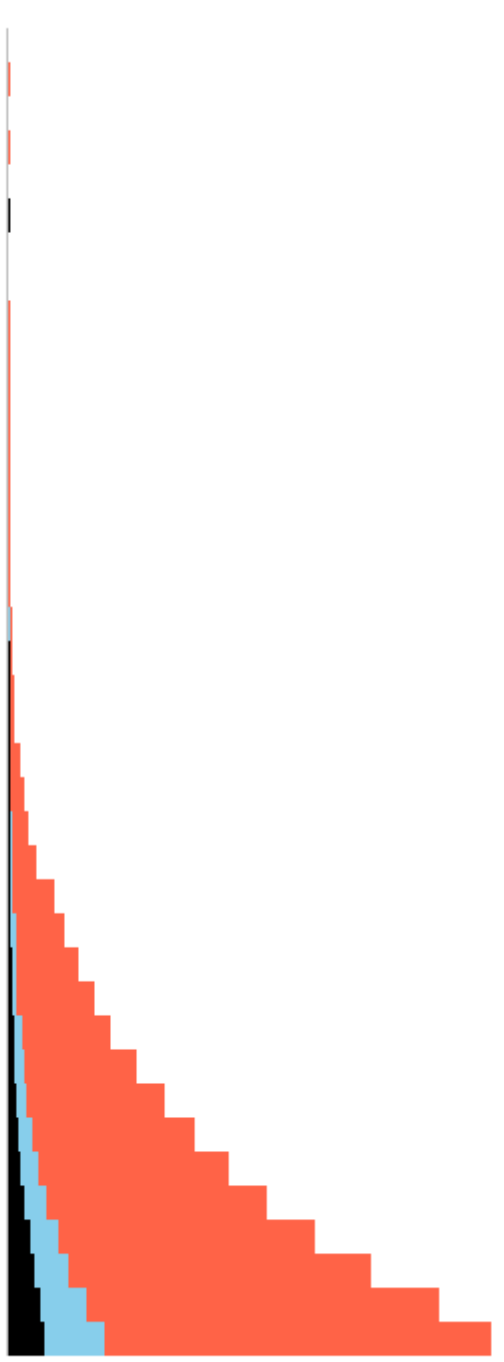
- $\sigma \approx 0.002138126$:

Looking at data in Europe, the percentage of recovered people from the infected per person-day seems to be approximately the same as the percentage that die i.e. $\beta \approx \sigma$. Looking at Italy in particular, we can see the two curves are extremely close:



with $\sigma \approx 1.15\beta$. However, we recall that the β value in Italy was considerably higher than elsewhere, namely 0.00185924. Thus, $\sigma \approx 1.15 \times 0.00185924 \approx 0.002138126$. This is 1.37 times our amalgamated value for β. Looking at data from Spain and China, this seems reasonable.

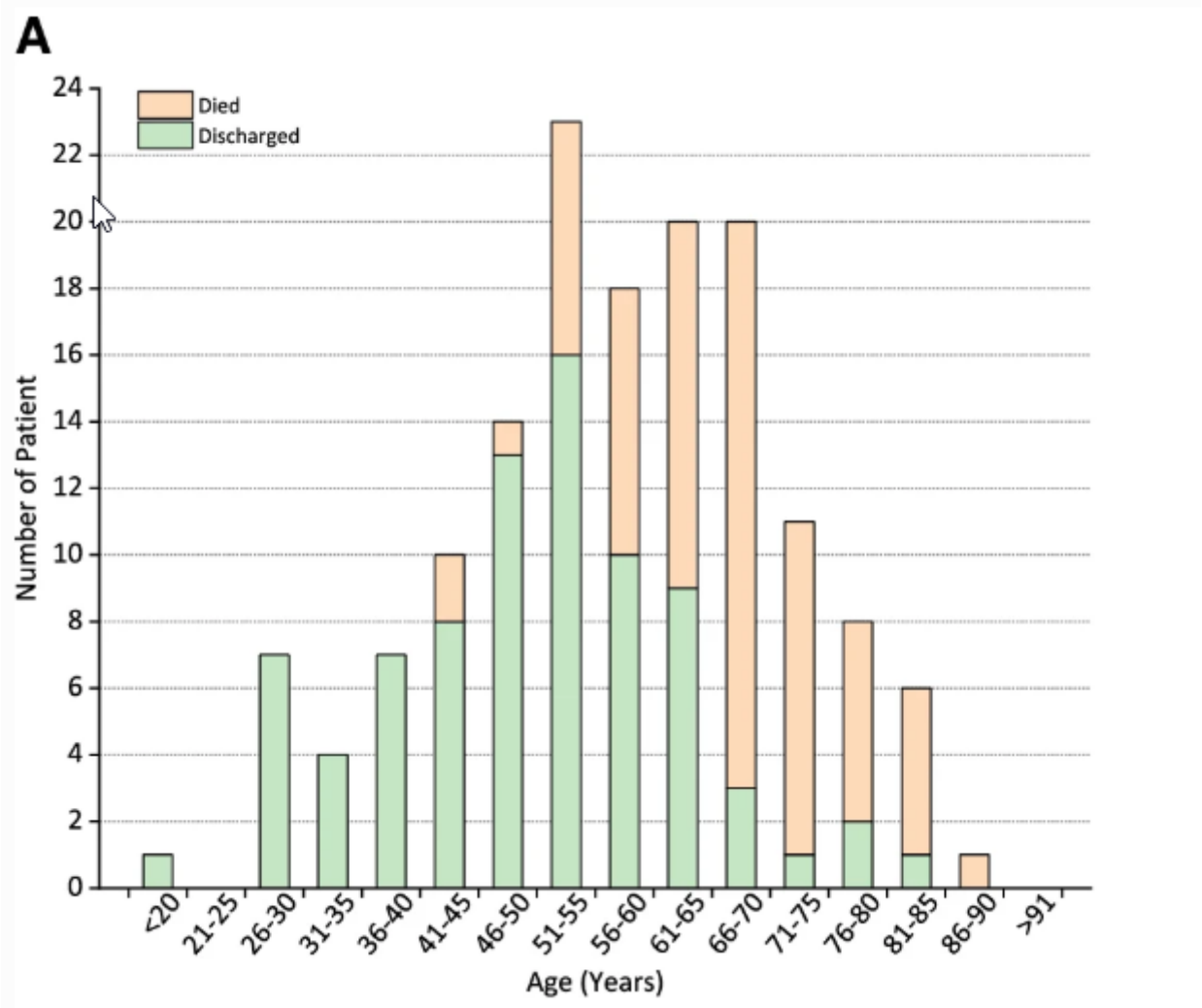
COVID-19 cases in Spain (v·t·e)



Date	# of cases	# of deaths
2020-01-31	1 (n.a.)	
⋮	1 (=)	
2020-02-09	2 (+100%)	
⋮	2 (=)	
2020-02-13	3 (+50%[i])	1 (n.a.)
2020-02-14	3 (=[ii])	1 (=)
⋮	3 (=)	1 (=)
2020-02-24	4 (+33%)	1 (=)
2020-02-25	10 (+150%)	1 (=)
2020-02-26	14 (+40%)	1 (=)
2020-02-27	26 (+86%)	1 (=)
2020-02-28	34 (+31%)	1 (=)
2020-02-29	59 (+74%)	1 (=)
2020-03-01	85 (+44%)	1 (=)
2020-03-02	121 (+42%)	1 (=)
2020-03-03	166 (+37%)	2 (+100%)
2020-03-04	228 (+37%)	2 (=)
2020-03-05	282 (+24%)	3 (+50%)
2020-03-06	401 (+42%)	8 (+167%)
2020-03-07	525 (+31%)	10 (+25%)
2020-03-08	674 (+28%)	17 (+70%)
2020-03-09	1,231 (+83%)	30 (+76%)
2020-03-10	1,695 (+38%)	36 (+20%)
2020-03-11	2,277 (+34%)	55 (+53%)
2020-03-12	3,146 (+38%)	86 (+56%)
2020-03-13	5,232 (+66%)	133 (+55%)
2020-03-14	6,391 (+22%)	196 (+47%)
2020-03-15	7,988 (+25%)	294 (+50%)
2020-03-16	9,942 (+24%)	342 (+16%)
2020-03-17	11,826 (+19%)	533 (+56%)
2020-03-18	14,769 (+25%)	638 (+20%)
2020-03-19	18,077 (+22%)	833 (+31%)
2020-03-20	21,571 (+19%)	1,093 (+31%)
2020-03-21	25,496 (+18%)	1,381 (+26%)
2020-03-22	29,909 (+17%)	1,813 (+31%)
2020-03-23	35,480 (+19%)	2,342 (+29%)
2020-03-24	42,058 (+19%)	2,994 (+28%)
2020-03-25	50,105 (+19%)	3,650 (+22%)
2020-03-26	56,188 (+12%)	4,089 (+12%)

Sources:

- Latest figures: RTVE (regular updates from the Ministry of Health)
- Daily figures: Worldometer



- $\gamma = 0$:

This was assumed to be zero i.e. I assumed that *all* of the exposed class became infected.

- $\kappa \approx 0.18$:

Setting $\gamma = 0$ meant κ was now just the incubation period constant, which was easier to find information on and thus the main reason I set $\gamma = 0$ in the first place. There have been a number of differing estimates for the incubation time. This article from the Journal of Medical Virology https://www.thelancet.com/journals/laninf/article/PIIS1473-3099(20)30144-4/fulltext (https://www.thelancet.com/journals/laninf/article/PIIS1473-3099(20)30144-4/fulltext) put the estimate of the mean incubation time at 5.2 days. https://www.eurosurveillance.org/content/10.2807/1560-7917.ES.2020.25.5.2000062#incubationperioddistribution-1 (https://www.eurosurveillance.org/content/10.2807/1560-7917.ES.2020.25.5.2000062#incubationperioddistribution-1) put it at a considerably higher 6.4 days. This article https://annals.org/aim/fullarticle/2762808/incubation-period-coronavirus-disease-2019-covid-19-from-publicly-reported (https://annals.org/aim/fullarticle/2762808/incubation-period-coronavirus-disease-2019-covid-19-from-publicly-reported) gave that the median incubation time was 5.1 days, and that 97.5% of those who develop symptoms will do so within 11.5 days. I unfortunately cannot utilise the data for the median since I only have one parameter with this model. Lastly, https://www.mdpi.com/2077-0383/9/2/538/htm (https://www.mdpi.com/2077-0383/9/2/538/htm) gave that the mean incubation period was 5.6 days. I have a number of parameters here, and I shall prioritise the mean. The mean of all the given means is 5.7. This becomes 5.6 if I assume a normal distribution, but looking at example plots this seems unfair.

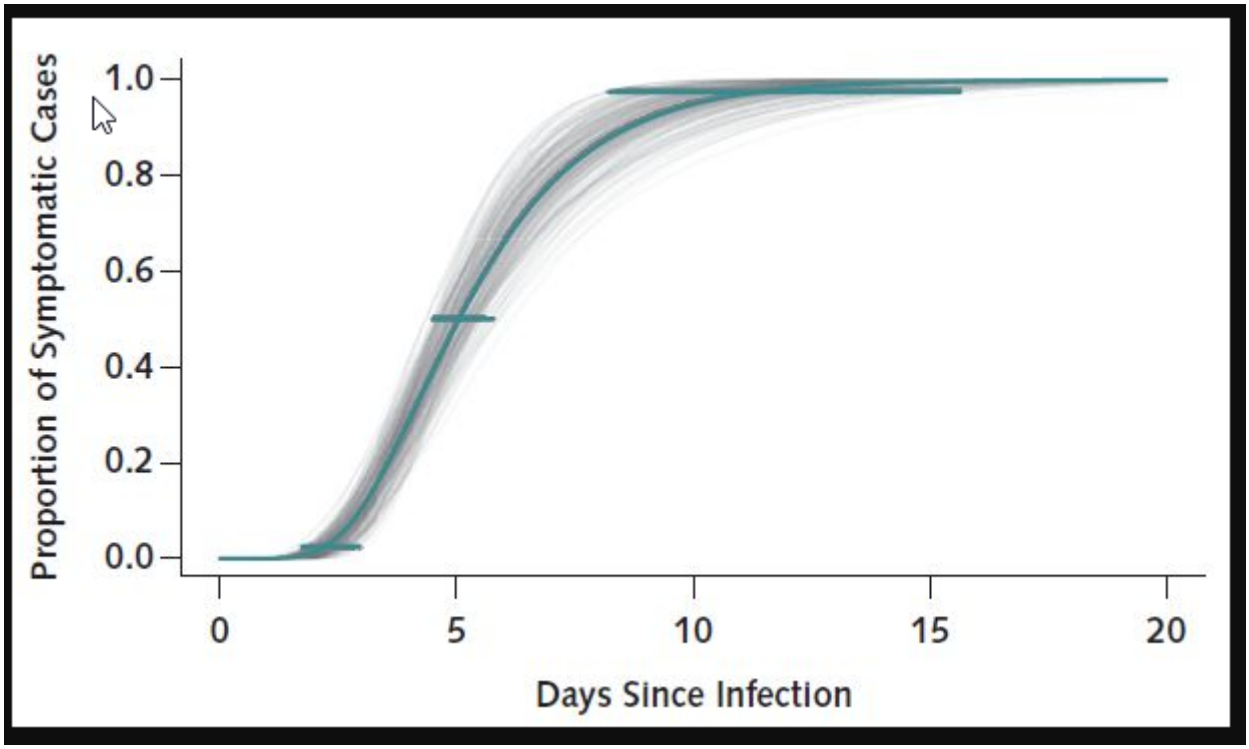
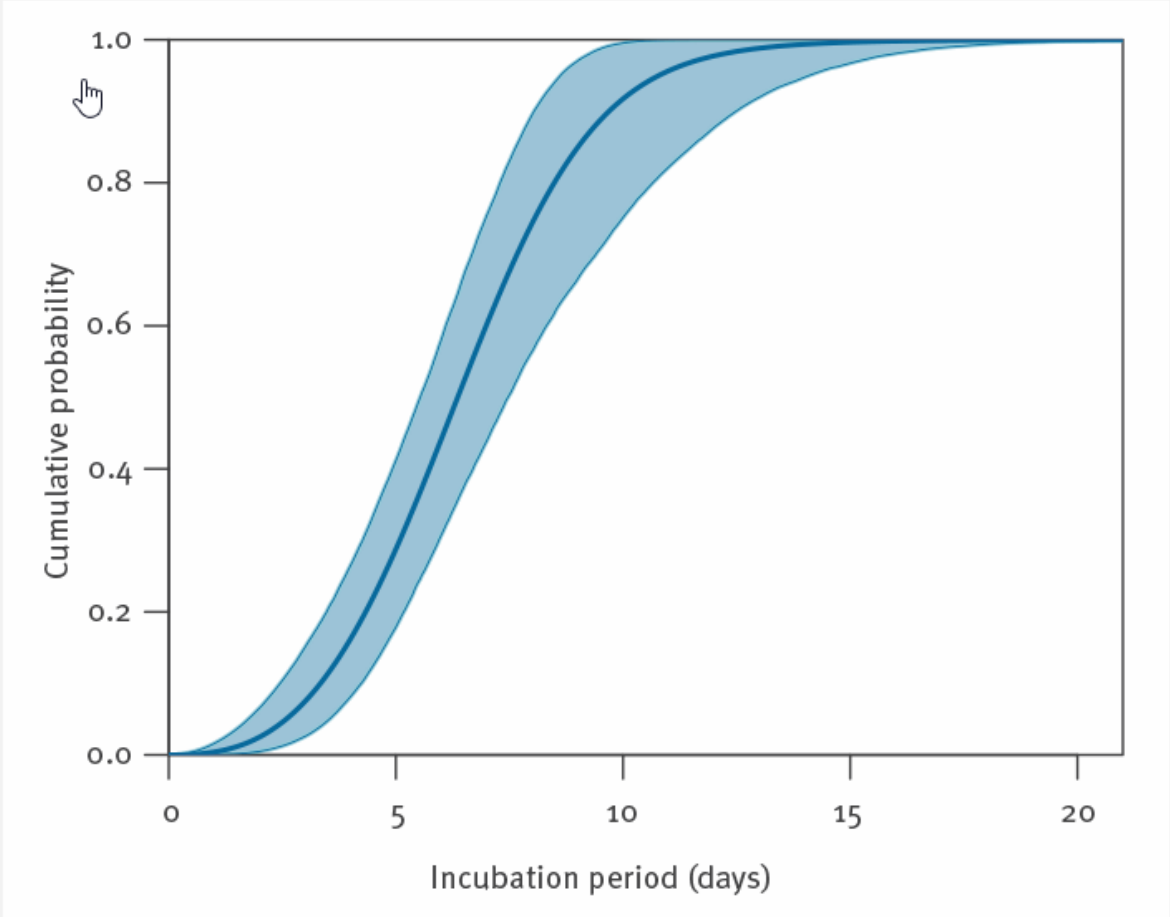
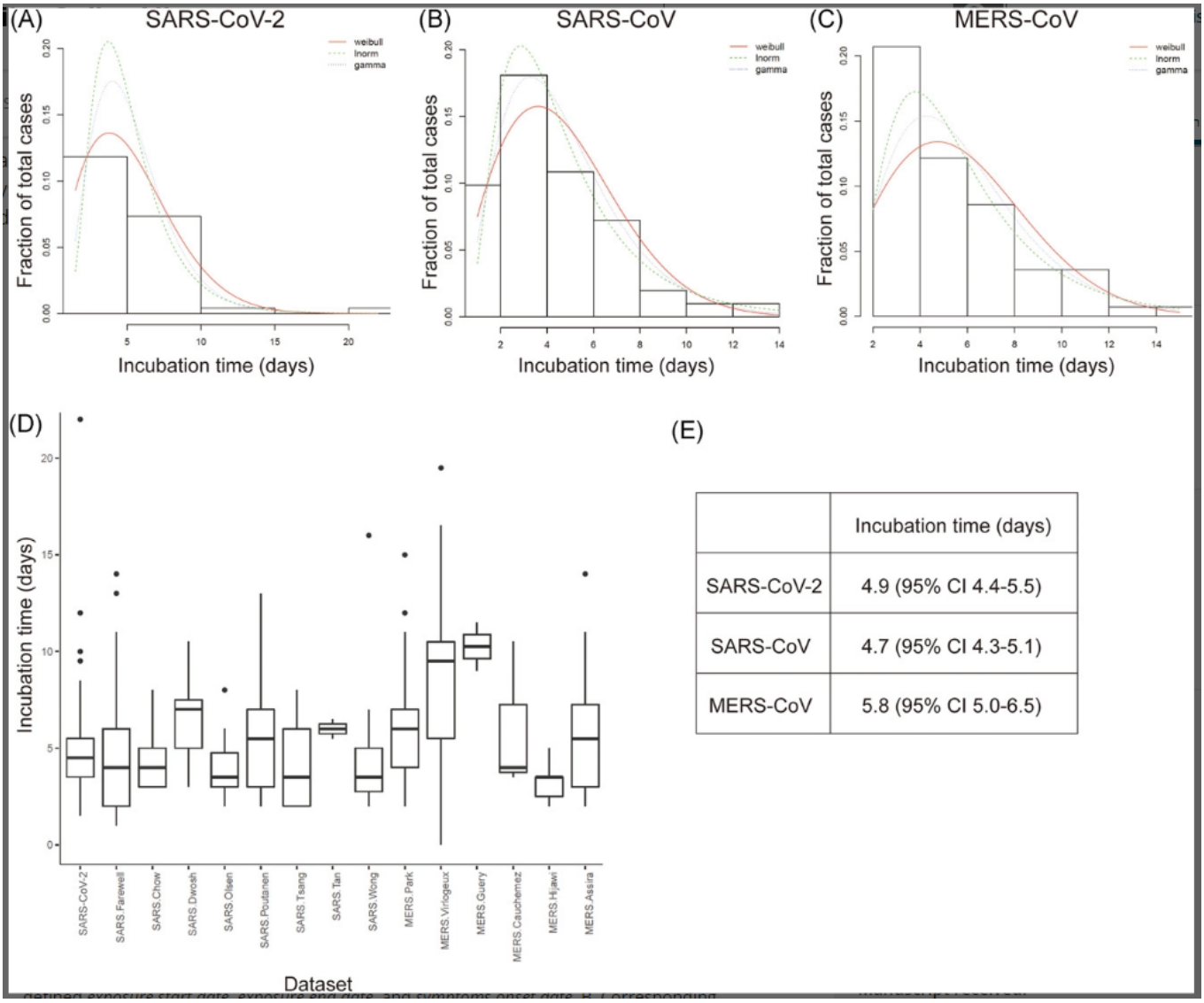


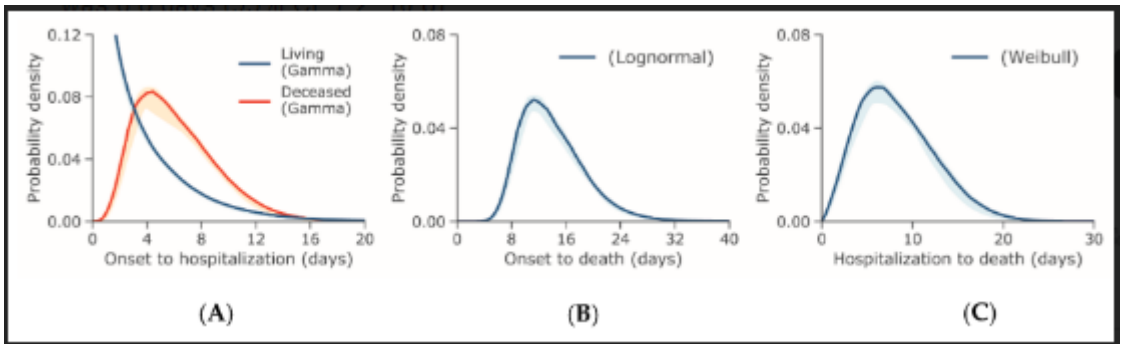
Figure 2. The cumulative density function of the estimated Weibull incubation period distribution for travellers infected with the 2019 novel coronavirus (2019-nCoV) in Wuhan, China, data 20–28 January 2020





	Incubation time (days)
SARS-CoV-2	4.9 (95% CI 4.4-5.5)
SARS-CoV	4.7 (95% CI 4.3-5.1)
MERS-CoV	5.8 (95% CI 5.0-6.5)

The Hospitalisation Period Distribution is as follows:



Thus, I shall focus on the article from the Journal of Clinical Medicine https://www.mdpi.com/2077-0383/9/2/538/htm (https://www.mdpi.com/2077-0383/9/2/538/htm) which gave the closest value to the mean - namely 5.6. It is also worth noting these estimates are all based on the Chinese dataset. By employing $\kappa \approx 0.18$, we obtain a mean of 5.6 and we also have that 90% of all cases develop symptoms with 11.5 days which is close to 97.5% from https://annals.org/aim/fullarticle/2762808/incubation-period-coronavirus-disease-2019-covid-19-from-publicly-reported (https://annals.org/aim/fullarticle/2762808/incubation-period-coronavirus-disease-2019-covid-19-from-publicly-reported). This is the best fit I can come up with without greatly complicating the model by keeping track of the date of exposure.

- $1.6 = 10.2404173w + w_0$, with $\alpha = \frac{w}{19200}$ and $\alpha_0 = \frac{w_0}{19200}$

To obtain an estimate for these constants, I shall focus on the time-varying basic reproduction number (R_t), defined as the mean number of secondary cases generated by a typical infectious individual on each day in a fully susceptible population. An article in the Lancet examining the data from Wuhan (https://www.thelancet.com/journals/laninf/article/PIIS1473-3099(20)30144-4/fulltext (https://www.thelancet.com/journals/laninf/article/PIIS1473-3099(20)30144-4/fulltext)) said that "We estimated that R_t varied during January, 2020, with median values ranging from 1·6 to 2·6 between Jan 1, 2020". From https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Steckbrief.html#doc13776792bodyText2 (https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Steckbrief.html#doc13776792bodyText2), the basic reproduction number should vary between 2.4 and 3.3. This has no time dependence, so I focused on the data from Wuhan. The population density for Wuhan is about 1200 per km^2 in general, and about 6000 per km^2 for urban area. Dublin, by comparison, has a density of 4,811 per km^2 in its urban area. Thus, they are quite comparable and I shall treat them as such.

So, a region of density 1200 per km^2 has a time-varying basic reproduction number with median values between 1.6 and 2.6. Taking the middle value, we have 2.1. That is, in my model: if one of my regions is isolated and has a population of 19200 or higher, then each infected individual infects about 2.1 more each day. I shall assume that this scales linearly with decreasing population. Thus, the question now becomes how many individuals one infected person would infect in a local population with one susceptible individual. This would be $2.1 \times \frac{1}{19200} \approx 0.000109375$ people. Naively, one would say thus $\alpha_0 \approx 0.000109375$, but this would assume that cities only consist of a single 16 km^2 area. This is obviously not the case. My population density data has that Dublin has approximately 57 times this i.e. has an area of around 912 km^2. From https://web.archive.org/web/20121114131842/http://www.cso.ie/en/media/csoie/census/documents/Prelim%20 (https://web.archive.org/web/20121114131842/http://www.cso.ie/en/media/csoie/census/documents/Prelim%20 the urban area is 318 km^2, so we are including a large number of commuter areas i.e. the greater Dublin area.

The boundary of this greater Dublin region stretches to roughly r=8 from its centre. The average distance from the centre is in the region of r=6. We want the sum of all infected in a city due to a single infected person to be 2.1 people. To do this, I shall consider a circle of r=5. If we do that with our granularity, then we obtain:

r	Count
0	1
1	4
2	4
3	4
4	4
5	4
$\sqrt{2}$	4
$\sqrt{5}$	8
$\sqrt{10}$	8
$2\sqrt{2}$	4
$\sqrt{17}$	8
$\sqrt{13}$	8

We want close at least 1.6 people infected, and since the region is will infect outside r=5, we take exactly 1.6. So,

$1.6 = w\left(4 \times 1^{-1} + 4 \times 4^{-1} + 4 \times 9^{-1} + 4 \times 16^{-1} + 4 \times 25^{-1} + 4 \times 2^{-1} + 4 \times 8^{-1} + 8 \times 5^{-1} + \right.$

$\approx 10.2404173w + w_0$

This is our main equation which we will use, and will allow me to vary a parameter to fit our model to the observed data. I shall find some reference values nonetheless, and then we shall find more accurate values by fitting to our data. We still require one more equation to solve. Let us say that if we did enlarge the circle to r=6 then the added infections would be about 0.2 so that the total would reach 1.8, still lower than the estimated upper limit for the median. The added regions would be:

r	Count
6	4
$\sqrt{26}$	8
$2\sqrt{5}$	8
$3\sqrt{2}$	4

So, $0.2 = w\left(4 \times 6^{-1} + 4 \times 18^{-1} + 8 \times 26^{-1} + 8 \times 20^{-1}\right) \approx 1.5965811w$. Thus,
$w = \frac{0.2}{1.5965811} \approx 0.1252676735$

Then, $w_0 = 1.6 - 10.2404173w \approx 1.6 - 10.2404173 \times 0.1252676735 \approx 0.31720674915984826$

The higher we make our value for w_0, the greater the number of infections we will obtain.

We have to scale our values like before, so

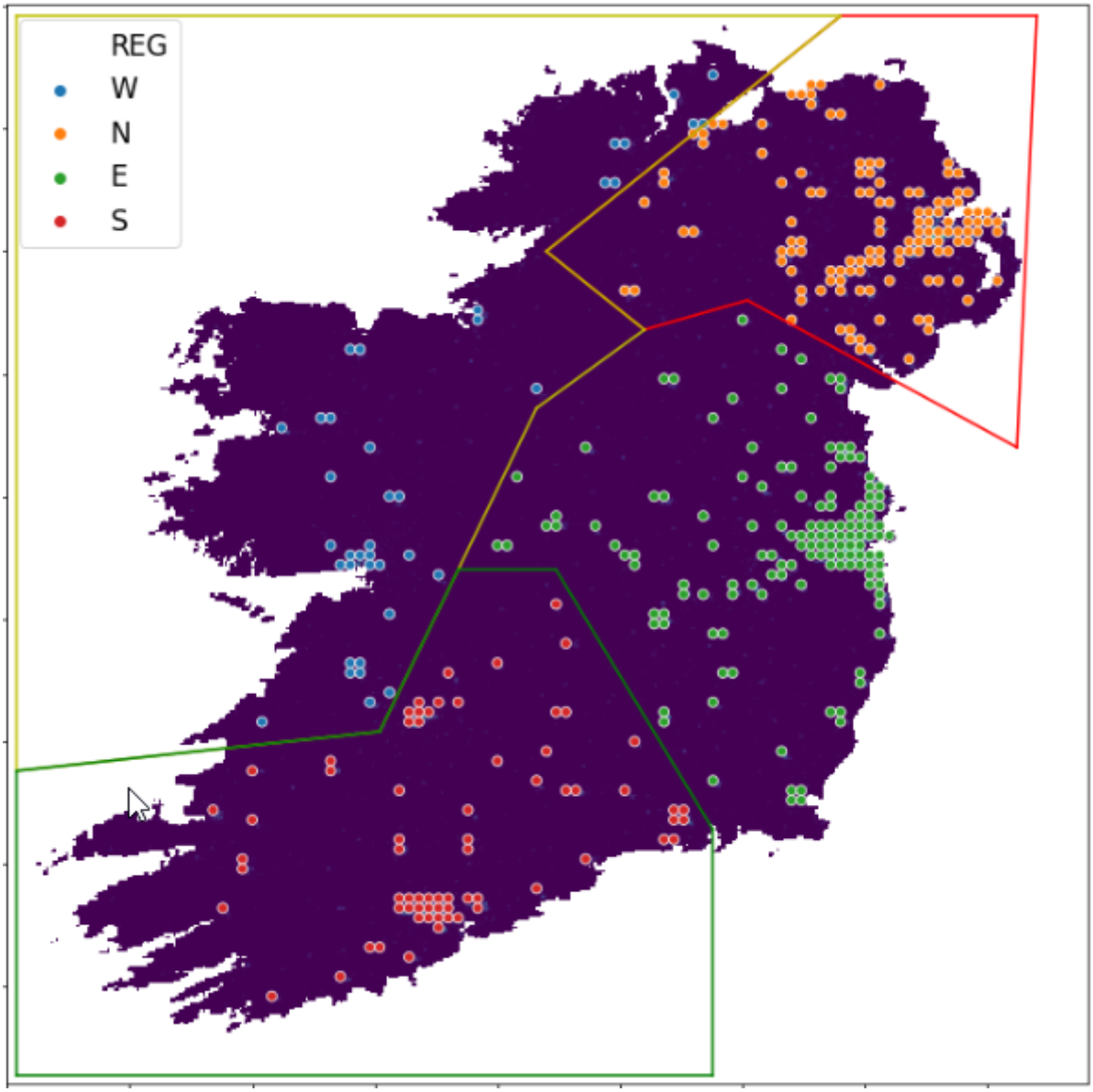
$\alpha_0 \approx 0.31720674915984826 \times \frac{1}{19200} \approx 0.00001652118485207543$

$\alpha \approx 0.1252676735 \times \frac{1}{19200} \approx 0.000006524357995$

Initial Conditions:

- **I**:

Data is readily available online in relation to the number of new cases recorded each day in both the Republic of Ireland and Northern Ireland (N). Furthermore, the a large amount of the data from the Republic of Ireland has regional granularity - with the regions being South (S), West, East (E) and North-West. Considering the West and North-West are by far the most sparsely populated regions, I combined these two into one region which I just called "West" (W). The boundaries of these regions were not well documented so I had to approximate them. The boundaries I used are shown below:



For data from the Republic of Ireland which did not have regional granularity, I estimated the location by assuming the likelihood of the case being in a certain location only depended on the population density there and then generating random numbers to simulate the cases. Regions with larger populations were favoured, so most of these unspecified datapoints ended up in the East of the country, which is not unrealistic.

Finally, the dead and recovered number had to be subtracted from the number of cases.

- **D**:

The number of deaths in the country is also readily available and was used.

- **R**:

This data is also similarly available.

- **S**:

This was calculated by taking the entire remaining population (roughly 4 million).

- **E**:

The initial conditions for the Exposed Population were hardest to pinpoint. They were essentially found by backtracking from a known datapoint i.e. fixing E such that the next number of infected would be equal to the known datapoint.

We now have all the data necessary to execute our model.

In [1]:

```
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import pandas as pd
import cv2
from PIL import Image
import tifffile as tiff
import matplotlib.animation as animation
import seaborn as sns
from IPython import display

mpl.rcParams.update({'font.size':17}) #make font bigger to match figsize
```

In [37]:

```
W=[0,0,0,0,0,0,4,6,7,7,7,8,10,10,12,14,19,28,30,38]
E=[0,0,1,1,1,2,2,6,7,8,9,9,12,13,24,33,54,79,120, 168]
S=[0,0,0,0,0,0,0,1,4,4,5,7,12,20,31,40,53,59,70,83]
R=[0,0,1,1,1,2,6,13,18,19,21,24,34,43,70,90,129,169,223,292, 366, 557, 683, 785, 906, 1
125, 1329]#, 1564] #republic
N=[1,1,1,1,1,1,3,3,4,7,12,12,16,18,20,29,34,45,52,62,68,77,86,104,128,148,172]
A=np.array(R)+np.array(N)
a=A
a
```

Out[37]:

```
array([   1,    1,    2,    2,    2,    3,    9,   16,   22,   26,   33,
         36,   50,   61,   90,  119,  163,  214,  275,  354,  434,  634,
        769,  889, 1034, 1273, 1501])
```

In [3]:

```
def prochain(idf, alpha=0.1252676735/19200, alpha_0=0.31720674915984826/19200, kappa=0.
18, gamma=0, beta=0.001550098, sigma=0.002138126, threshold=6000*16):
    """Given the current state and various probabilities, predicts the next state"""
    df=idf.copy()
    df1=idf.copy()

    df['I_new']=0
    df['E_new']=0
    df['D_new']=0
    df['R_new']=0
    df['S_new']=0

    df['i_contrib_D']=beta*df['I']
    df['i_contrib_R']=sigma*df['I']
    df['s_contrib_E']=0

    infected=df[df['I']>0]

    for i in infected.index.values: #for each infected
        #print(i)
        for j in range(df.shape[0]): #for each region
            if i==j: #i.e. i to i or j to j
                avail_S=np.where(infected['S'][i]>threshold, threshold, infected['S'][i
])
                df['s_contrib_E'][j]=infected['s_contrib_E'][i]+alpha_0 * infected['I']
[i] * avail_S
            else: #i.e. i to j
                x=df['x'][i]-df['x'][j]
                y=df['-y'][i]-df['-y'][j]
                r_2=(abs(x)**2+abs(y)**2)
                avail_s=np.where(df['S'][j]>threshold, threshold, df['S'][j])
                df['s_contrib_E'][j]=df['s_contrib_E'][j]+infected['I'][i] * 1/r_2 * al
pha * avail_s

    df['s_contrib']=df['s_contrib_E']
    df['i_contrib']=df['i_contrib_D']+df['i_contrib_R']

    df['s_contrib_E']=np.where(df['s_contrib']>df['S'], df['S'], df['s_contrib'])
    df['i_contrib_D'], df['i_contrib_R']=np.where(df['i_contrib']>df['I'],[df['I']*df[
'i_contrib_D']/df['i_contrib'], df['I']*df['i_contrib_R']/df['i_contrib']], [df['i_cont
rib_D'], df['i_contrib_R']])

    df['s_contrib']=df['s_contrib_E']
    df['i_contrib']=df['i_contrib_D']+df['i_contrib_R']


    df['E_new']=df['E_new']+df['s_contrib_E']
    df['D_new']=df['D_new']+df['i_contrib_D']
    df['R_new']=df['R_new']+df['i_contrib_R']
    df['I_new']=kappa*df['E']
    df['S_new']=gamma*df['E']

    df1['I']=df['I']+df['I_new']-df['i_contrib']
    df1['E']=df['E']+df['E_new']
    df1['D']=df['D']+df['D_new']
    df1['R']=df['R']+df['R_new']
    df1['S']=df['S']+df['S_new']-df['s_contrib']

    return df1
```

In [4]:

```
inits=pd.read_csv(r'O:\Desktop\WORLDMAP\INITIAL_CONDS_16.csv', sep=',', index_col=0)
df=pd.read_csv(r'O:\Desktop\WORLDMAP\IRELANDMAP.csv', index_col=0)
inits['I'].sum()
```

Out[4]:

```
275
```

Next datapoint is known to be 354, so we require $354 - 275 = \kappa E_0 \implies E_0 = \frac{79}{\kappa} = \frac{79}{0.18} \approx 439$. I spread these exposed people about the population, assuming the chance of an exposed person being in a location is proportional to the number of people there.

Trying out a number of values, it looks like w=0.15 is quite a good approximation.

In [29]:

```
w=0.15 #reference is 0.1252676735
w_0=1.6-10.2402173*w

print(w, w_0)

if w_0<=0:
    print("error")
else:
    data=prochain(inits, w/19200, w_0/19200)
    data1=prochain(data, w/19200, w_0/19200)
    data2=prochain(data1, w/19200, w_0/19200)
    data3=prochain(data2, w/19200, w_0/19200)
    data4=prochain(data3, w/19200, w_0/19200)
    data5=prochain(data4, w/19200, w_0/19200)
    data6=prochain(data5, w/19200, w_0/19200)
    data7=prochain(data6, w/19200, w_0/19200)
    data8=prochain(data7, w/19200, w_0/19200)
    data9=prochain(data8, w/19200, w_0/19200)
    data10=prochain(data9, w/19200, w_0/19200)
```

```
0.15 0.06396740500000031

C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:29: Setti
ngWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:23: Setti
ngWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

In [30]:

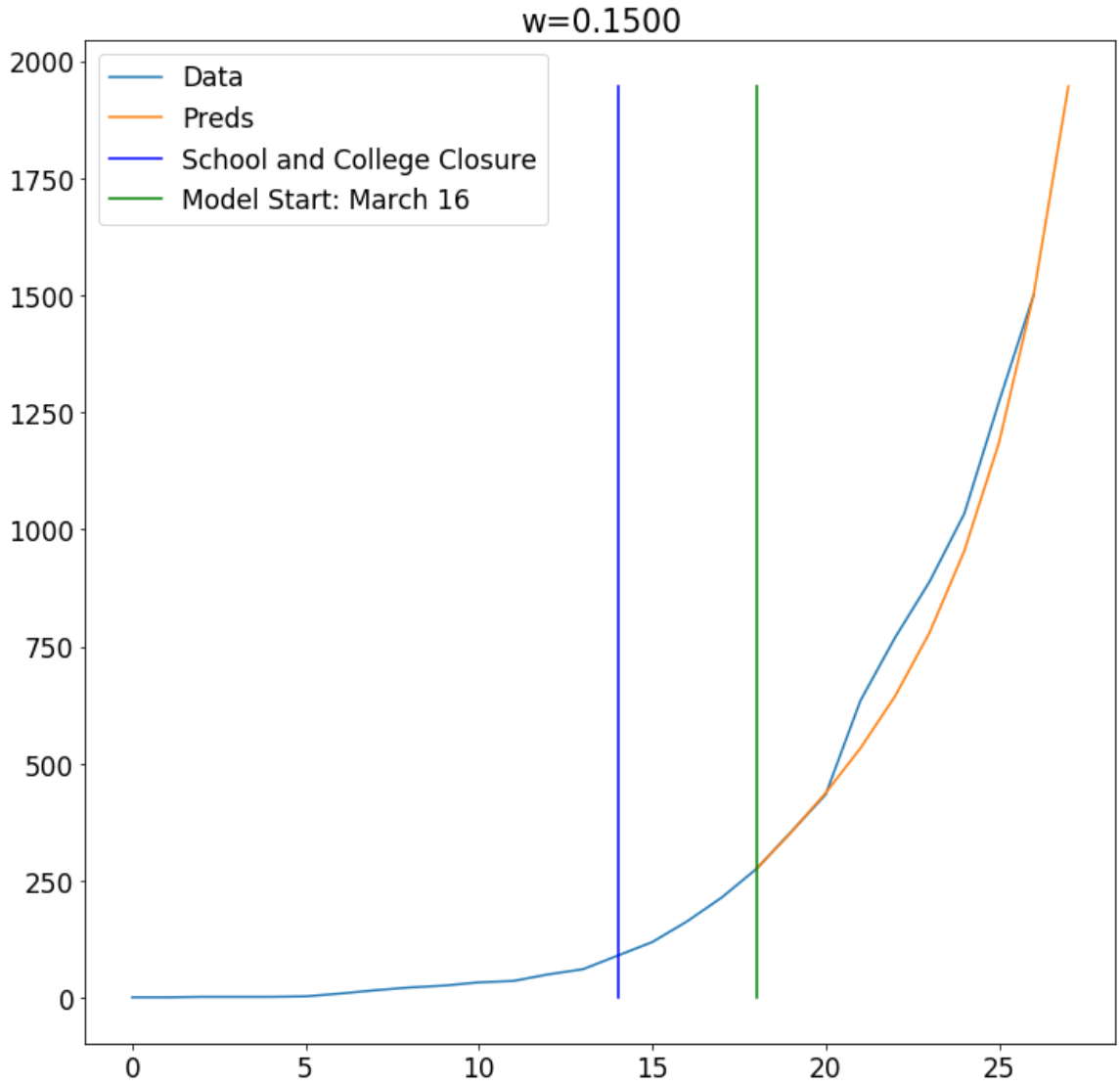
```
def analysis(var, comp=a):
    print(a)
    print(str(inits['I'].sum())+" - "+str(a[18]))
    print(str(data['I'].sum())+" - "+str(a[19]))
    print(str(data1['I'].sum())+" - "+str(a[20]))
    print(str(data2['I'].sum())+" - "+str(a[21]))
    print(str(data3['I'].sum())+" - "+str(a[22]))
    print(str(data4['I'].sum())+" - "+str(a[23]))
    print(str(data5['I'].sum())+" - "+str(a[24]))
    print(str(data6['I'].sum())+" - "+str(a[25]))
    print(str(data7['I'].sum())+" - "+str(a[26]))
    print(str(data8['I'].sum())+" - "+str("N/A"))
    print(str(data9['I'].sum())+" - "+str("N/A"))
    print(str(data10['I'].sum())+" - "+str("N/A"))

    preds=[]
    preds.append(inits[var].sum())
    preds.append(data[var].sum())
    preds.append(data1[var].sum())
    preds.append(data2[var].sum())
    preds.append(data3[var].sum())
    preds.append(data4[var].sum())
    preds.append(data5[var].sum())
    preds.append(data6[var].sum())
    preds.append(data7[var].sum())
    preds.append(data8[var].sum())
    preds.append(data9[var].sum())
    preds.append(data10[var].sum())

    fig, ax = plt.subplots(figsize=(12, 12))
    ax.plot(range(0,27), a, label="Data")
    ax.plot(range(18, 28), preds[0:-2], label="Preds")
    ax.plot(np.ones(100)*14, np.linspace(0, max(preds[0:-2]), 100), 'b-', label="School
and College Closure")
    ax.plot(np.ones(100)*18, np.linspace(0, max(preds[0:-2]), 100), 'g-', label="Model
 Start: March 16")
    ax.set_title("w={0:.4f}".format(w))
    ax.set_xlabel("Days Since First Case")
    ax.set_ylabel("Number of Infected")
    plt.legend()
```

With w=0.15, the fit is quite good. I'll analyse how good our prediction is below.

In [31]:

```
analysis('I')
```

```
[   1    1    2    2    2    3    9   16   22   26   33   36   50   61
   90  119  163  214  275  354  434  634  769  889 1034 1273 1501]
275 - 275
353.00573840000004 - 354
437.5637741634954 - 434
532.6099409500949 - 634
643.8655561832443 - 769
780.2708357861559 - 889
954.3330221631094 - 1034
1185.713228206775 - 1273
1502.460052221385 - 1501
1946.198642997741 - N/A
2578.2006264538695 - N/A
3485.2516450265666 - N/A
```



The model is predicting $1946.198642997741 - 1502.460052221385 \approx 444$ cases today. This is an upper estimate, since we can see that the slope of my graph is increasing faster than the data would indicate.

In [32]:

```
#find the number of dead on the second day
data7['D'].sum()
```

Out[32]:

```
8.002167160057349
```

The number of dead is in fact 9, so we are very close. We next look at the individual regions and see how the model is predicting compared to what we estimated from government data.

In [39]:

```
print("WEST:", data[data['REG']=="W"]['I'].sum(),"vs", W[-1])
print('ERROR: {0:.3f}%'.format(100*(data[data['REG']=="W"]['I'].sum()-W[-1])/W[-1]))
print("SOUTH:", data[data['REG']=="S"]['I'].sum(),"vs", S[-1])
print('ERROR: {0:.3f}%'.format(100*(data[data['REG']=="S"]['I'].sum()-S[-1])/S[-1]))
print("EAST:", data[data['REG']=="E"]['I'].sum(),"vs", E[-1])
print('ERROR: {0:.3f}%'.format(100*(data[data['REG']=="E"]['I'].sum()-E[-1])/E[-1]))
print("NORTH:", data[data['REG']=="N"]['I'].sum(),"vs", N[19])
print('ERROR: {0:.3f}%'.format(100*(data[data['REG']=="N"]['I'].sum()-N[19])/N[19]))
```

```
WEST: 36.105665056 vs 38
ERROR: -4.985%
SOUTH: 83.24182432000002 vs 83
ERROR: 0.291%
EAST: 160.153724896 vs 168
ERROR: -4.670%
NORTH: 73.50452412800001 vs 62
ERROR: 18.556%
```

So we see that we are doing an OK job, though we are over-representing cases in the North.

I have population density data, so I can make useful animations showing the propogation of the disease.

In [40]:

```
Writer = animation.writers['ffmpeg']
writer = Writer(fps=20, metadata=dict(artist='Me'), bitrate=1800)

#saving space, putting animation in separate file
#fig = plt.figure(figsize=(10,10))
#plt.scatter([], [])

datas=[inits,data,data1,data2,data3,data4,data5,data6,data7,data8,data9,data10]

var='I'

def animate(i):
    p=plt.imshow((df**0.5)**2, cmap="copper")
    p=plt.title("Day {0}: {1} Infected".format(i,int(datas[i]['I'].sum())))
    p=plt.axis('off')
    p1=plt.scatter(datas[i][datas[i]['REG']=="W"]['4x'],datas[i][datas[i]['REG']=="W"][
'-4y'],s=datas[i][datas[i]['REG']=="W"][var], c='red')
    p2=plt.scatter(datas[i][datas[i]['REG']=="S"]['4x'],datas[i][datas[i]['REG']=="S"][
'-4y'],s=datas[i][datas[i]['REG']=="S"][var], c='magenta')
    p3=plt.scatter(datas[i][datas[i]['REG']=="E"]['4x'],datas[i][datas[i]['REG']=="E"][
'-4y'],s=datas[i][datas[i]['REG']=="E"][var], c='orange')
    p4=plt.scatter(datas[i][datas[i]['REG']=="N"]['4x'],datas[i][datas[i]['REG']=="N"][
'-4y'],s=datas[i][datas[i]['REG']=="N"][var], c='purple')
    p=plt.legend((p1, p2, p3, p4),
           ('WEST: {0:.0f}'.format(datas[i][datas[i]['REG']=="W"][var].sum()), 'SOUTH:
{0:.0f}'.format(datas[i][datas[i]['REG']=="S"][var].sum()), 'EAST: {0:.0f}'.format(data
s[i][datas[i]['REG']=="E"][var].sum()), 'NORTH: {0:.0f}'.format(datas[i][datas[i]['REG'
]=="N"][var].sum())),
               fontsize=16)

#commenting these out in order to save space
# anim = animation.FuncAnimation(fig, animate, frames=len(datas), repeat=True, interval
=700)
# video=anim.to_html5_video()
# html=display.HTML(video)
# display.display(html)
# plt.close()
```

In [41]:

```
fig2 = plt.figure(figsize=(10,10))
plt.scatter([], [])

anim = animation.FuncAnimation(fig2, animate, frames=len(datas), repeat=True, interval=
700)
anim.save('covid19_prediction_25_03_20.mp4')
```