COVID-19 Modelling

March 30, 2020

1 SECIRD Model

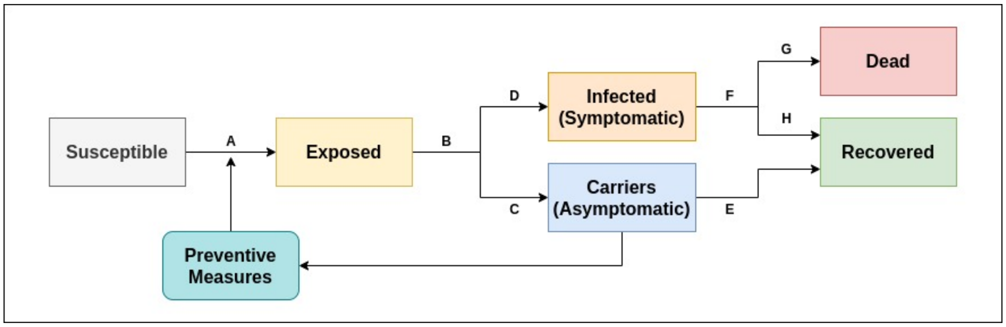
The SECIRD model is an extension to the SEIR deterministic model for modelling the spread of an infectious disease. In it, a population is broken into the following non-overlapping groups corresponding to stages of the disease:

- **Susceptible (S).** The subpopulation susceptible to acquire the disease.
- **Exposed (E).** The subpopulation that has been infected with the virus, but not yet in an infective state capable of transmitting the virus to others.
- **Carrier (C).** The subpopulation that has been infected with the virus but are symptomatic while being capable of infecting others.
- **Infectious (I).** The subpopulation that has acquired the virus and can infect others, and can also possibly die.
- **Recovered (R).** The subpopulation that has recovered from infection and presumed to be no longer susceiptible to the disease.
- **Dead (D).** The subpopulation that suffers disease-induced death.

A above model for the spread of an infectious disease in a uniform population is given by the deterministic SECIRD equations

$$
\begin{aligned}
\frac{dS}{dt} &= -(1-u)\frac{\beta S K}{N} \\
\frac{dE}{dt} &= (1-u)\frac{\beta S K}{N} - \frac{\alpha E}{N} \\
\frac{dC}{dt} &= \frac{\sigma \alpha E}{N} - \frac{\gamma C}{N} \\
\frac{dI}{dt} &= \frac{(1-\sigma)\alpha E}{N} - \frac{\gamma I}{N} \\
\frac{dR}{dt} &= \frac{\delta \gamma I}{N} + \frac{\gamma C}{N} \\
\frac{dD}{dt} &= \frac{(1-\delta)\gamma I}{N} \\
u &= z\left(1 - \frac{C}{N}\right) \\
N &= S + E + C + I + R + D
\end{aligned}
$$

The rate processes are modeled as follows.



SECIRD Model

- $(1-u)\frac{\beta SK}{N}$ is the rate at which susecptible population encounters the infected population resulting in trasmission of the disease. S is the size of the susceptible population. β is a the model parameters with units of 1/day. K is the probability of disease transmission in contact between a susceptible and infectious/carrier subject.
- u describes the effectiveness on any public health interventions to control transmission of the disease. $u = 0$ corresponds to no effective public health interventions, $u = 1$ implies total elimination of disease transmission. u is also dependent upon the C, the population of carriers. z represents the effectiveness of preseventive measures.
- αE is the rate at which exposed population becomes infective, where E is the size of the exposed population. The average period of time in the exposed state is the incubation period of the disease, and equal to $\frac{1}{\alpha}$.
- σ represents the probabilityt for an exposed population to become a carrier.
- γ represents the rate at which infected/carrier population recovers and becomes resistent to further infection. The average time period is $\frac{1}{\gamma}$
- I is the size of the infective population.
- δ represents the mortality probability.

In [1]:
```
!pip3 install scipy numpy seaborn matplotlib --user
```

```
Requirement already satisfied: scipy in /home/whatsis/.local/lib/python3.6/site-packages (1.4.1)
Requirement already satisfied: numpy in /home/whatsis/.local/lib/python3.6/site-packages (1.16.1)
Requirement already satisfied: seaborn in /home/whatsis/.local/lib/python3.6/site-packages (0.9.0)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.6/dist-packages (2.2.4)
Requirement already satisfied: pandas>=0.15.2 in /usr/local/lib/python3.6/dist-packages (from seaborn) (0.24.2)
Requirement already satisfied: cycler>=0.10 in /home/whatsis/.local/lib/python3.6/site-packages (from matplotli
Requirement already satisfied: kiwisolver>=1.0.1 in /home/whatsis/.local/lib/python3.6/site-packages (from mat
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /home/whatsis/.local/lib/python3.6/s
Requirement already satisfied: pytz in /usr/lib/python3/dist-packages (from matplotlib) (2018.3)
Requirement already satisfied: python-dateutil>=2.1 in /home/whatsis/.local/lib/python3.6/site-packages (from 
Requirement already satisfied: six>=1.10 in /home/whatsis/.local/lib/python3.6/site-packages (from matplotlib)
Requirement already satisfied: setuptools in /home/whatsis/.local/lib/python3.6/site-packages (from kiwisolver>
```

In [ ]:
```
import numpy as np
from scipy.integrate import odeint
```

```
        import matplotlib.pyplot as plt
        import seaborn as sns

In [95]: def step(c, t, t_social_distancing):
             return 1-c if t >= 10*t_social_distancing else 0

         def deriv(x, t, z, u, alpha, beta, gamma, delta, sigma):
             s, e, c, i, r, d = x
             dsdt = -(1 - u*step(c, t, z)/100) * beta * s * i
             dedt =  (1 - u*step(c, t, z)/100) * beta * s * i - alpha * e
             dcdt =  sigma * alpha * e - gamma * c
             didt =  (1 - sigma) * alpha * e - gamma * i
             drdt =  (1 - delta) * gamma * i + gamma * c
             dddt =  delta * gamma * i
             return [dsdt, dedt, dcdt, didt, drdt, dddt]

In [60]: def run(R0, Mr, As, t_incubation, t_recovery, N, n, t_social_distancing, u_social_distancing):
             # initial number of infected and recovered individuals
             e_initial = n/N
             c_initial = 0.00
             i_initial = 0.00
             r_initial = 0.00
             d_initial = 0.00
             s_initial = 1 - e_initial - i_initial - r_initial - d_initial - c_initial

             # Inititalize variables
             alpha = 1/t_incubation
             gamma = 1/t_recovery
             beta = R0*gamma
             delta = Mr
             sigma = As

             t = np.linspace(0, 350, 350)
             x_initial = s_initial, e_initial, c_initial, i_initial, r_initial, d_initial
             s, e, c, i, r, d = odeint(deriv, x_initial, t,
                                       args=(t_social_distancing, u_social_distancing,
                                             alpha, beta, gamma, delta, sigma)).T
             s0, e0, c0, i0, r0, d0 = odeint(deriv, x_initial, t, args=(0, 0, alpha, beta, gamma, delta, sigma)).T


             # plotting the data
             fig = plt.figure(figsize=(16, 8))
             ax = [fig.subplots()]

             pal = sns.color_palette()

             ax[0].stackplot(t/7, N*s, N*e, N*c, N*i, N*r, N*d, colors=pal, alpha=0.6)
             ax[0].set_xlabel('Weeks after Inital Exposure')
```

```
    ax[0].set_xlim(0, t[-1]/7)
    ax[0].set_ylim(0, N)
    ax[0].legend([
        'Susceptible',
        'Exposed',
        'Carrier (Asymptomatic)',
        'Infectious (symptomatic)',
        'Recovered',
        'Dead'],
        loc='best')
    ax[0].plot(np.array([t_social_distancing, t_social_distancing]), ax[0].get_ylim(), 'r', lw=3)
    ax[0].plot(np.array([0, t[-1]])/7, [N/R0, N/R0], lw=3, label='herd immunity')
    ax[0].annotate("Start of social distancing",
        (t_social_distancing, 0), (t_social_distancing + 1.5, N/10),
        arrowprops=dict(arrowstyle='->'))
    ax[0].annotate("Herd Immunity without social distancing",
        (t[-1]/7, N/R0), (t[-1]/7 - 8, N/R0 - N/5),
        arrowprops=dict(arrowstyle='->'))
    #plt.tight_layout()
    return ax
```

1.1 Default Parameters

Parameter	Symbol	Typical
Reproduction number	R_0	2.4
Incubation period (days)	$\tau_{incubation}$	5.1
Recovery period (days)	$\tau_{recovery}$	3.3
Mortality Rate	M_r	0.2
Asymptomatic Rate	A_s	0.2
Population	N	22,09,000
Initial number exposed	n	10
Mitigation by preventive measures	u	0.2
Start of social distancing following exposure (weeks)	t_{sd}	12

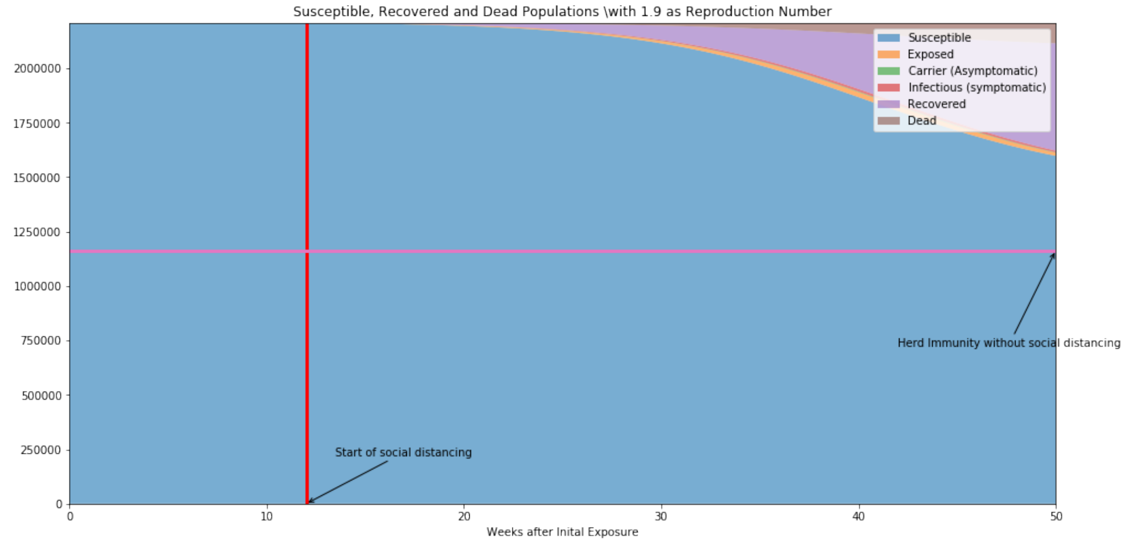
1.2 Variation in Reproduction Number

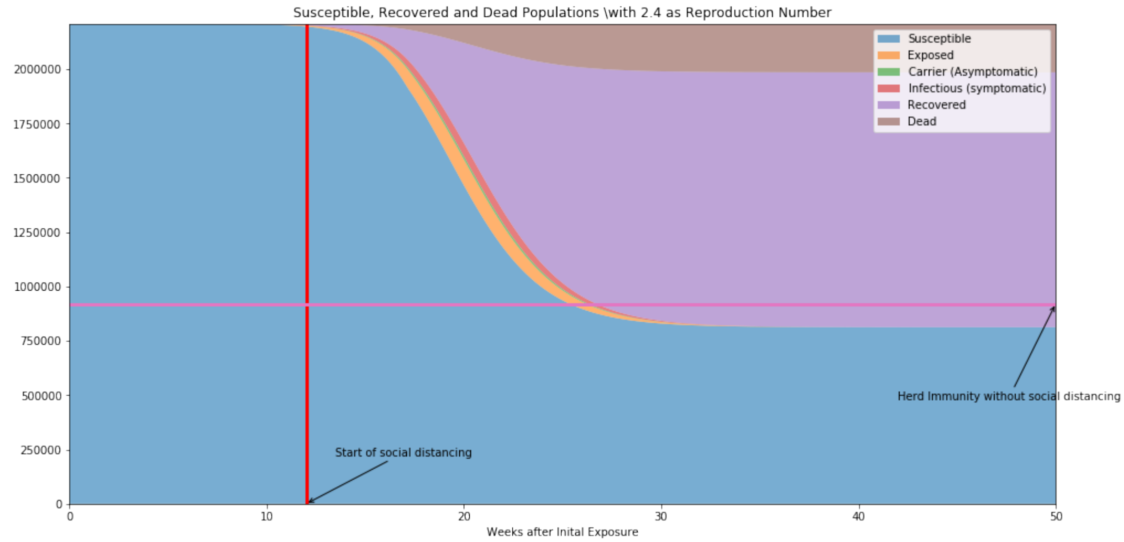
We begin by changing the R_0, reproduction number of the disease at hand. We choose the three values from the expected range of R_0 for COVID-19[1]. In general, R_0 for an infection can be thought of as the expected number of cases directly generated by one case in a population where all individuals are susceptible to infection. Below graphs reveal that change in R_0 leads to exponential change in net infected individuals. However, most of them are able to recover successfully.

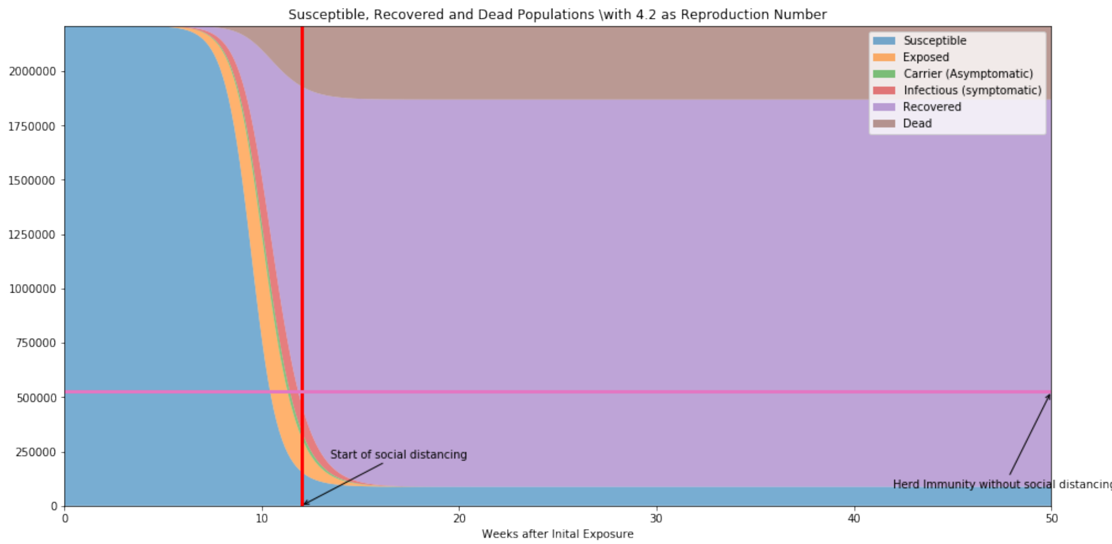
```
In [74]: R0s = [1.9, 2.4, 4.2]

        for R0 in R0s:
            ax = run(R0, 0.2, 0.2, 5.1, 3.3, 2209000, 3, 12, 20)
```

```
ax[0].set_title('Susceptible, Recovered and Dead Populations \
                 with {0:1.1f} as Reproduction Number'.format(R0))
```







1.2.1 References

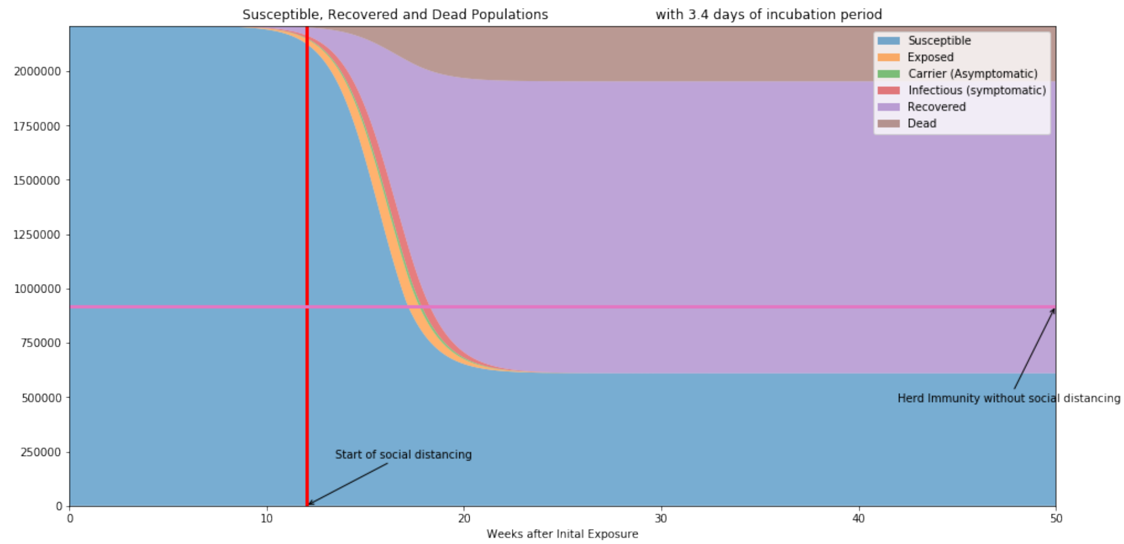
[1] Liu T, Hu J, Kang M, Lin L (January 2020). "Time-varying transmission dynamics of Novel Coronavirus Pneumonia in China". bioRxiv. doi:10.1101/2020.01.25.919787.

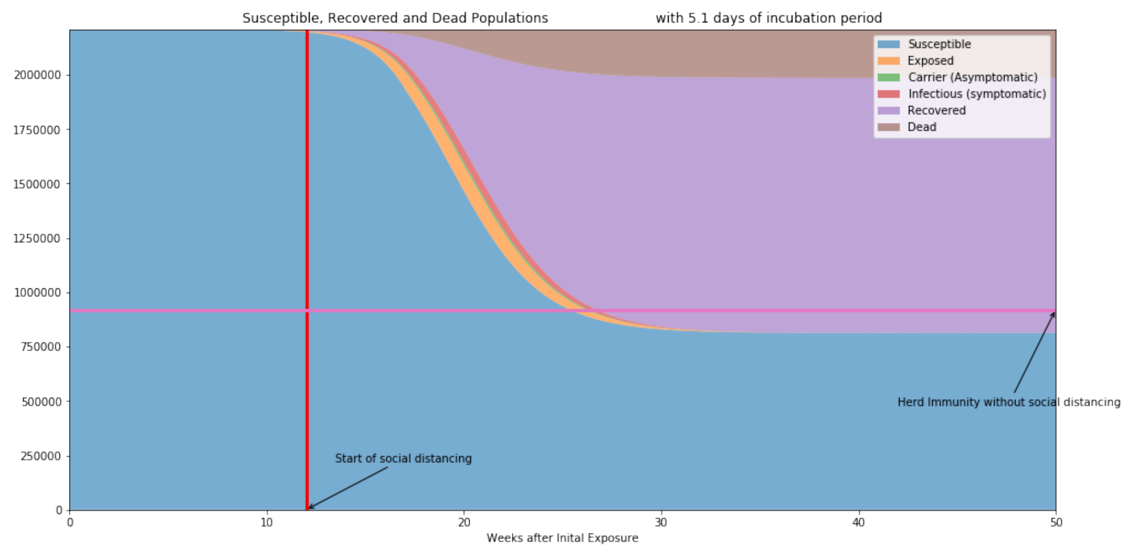
1.3 Variation in Incubation Period

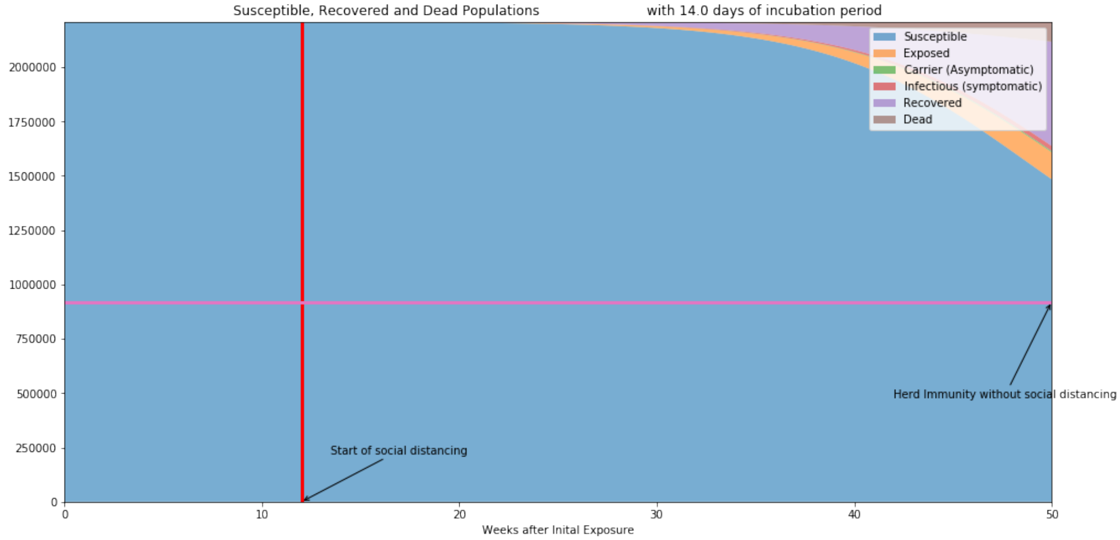
Next, we vary the incubation period for our disease. As we know that the rate of transition from exposed to either of the infected/carrier stage is inversely proportional to $\tau_{incubation}$, the spread of disease gets restricted as its value becomes larger.

In [77]:
```
Incs = [3.4, 5.1, 14.0]

for inc in Incs:
    ax = run(2.4, 0.2, 0.2, inc, 3.3, 2209000, 3, 12, 20)
    ax[0].set_title('Susceptible, Recovered and Dead Populations \
                    with {0:1.1f} days of incubation period'.format(inc))
```


Susceptible, Recovered and Dead Populations with 3.4 days of incubation period
Susceptible
Exposed
Carrier (Asymptomatic)
Infectious (symptomatic)
Recovered
Dead
2000000
1750000
1500000
1250000
1000000
750000
500000
250000
0
Herd Immunity without social distancing
Start of social distancing
0
10
20
30
40
50
Weeks after Inital Exposure


Susceptible, Recovered and Dead Populations with 5.1 days of incubation period
Susceptible
Exposed
Carrier (Asymptomatic)
Infectious (symptomatic)
Recovered
Dead
2000000
1750000
1500000
1250000
1000000
750000
500000
250000
0
Herd Immunity without social distancing
Start of social distancing
0
10
20
30
40
50
Weeks after Inital Exposure

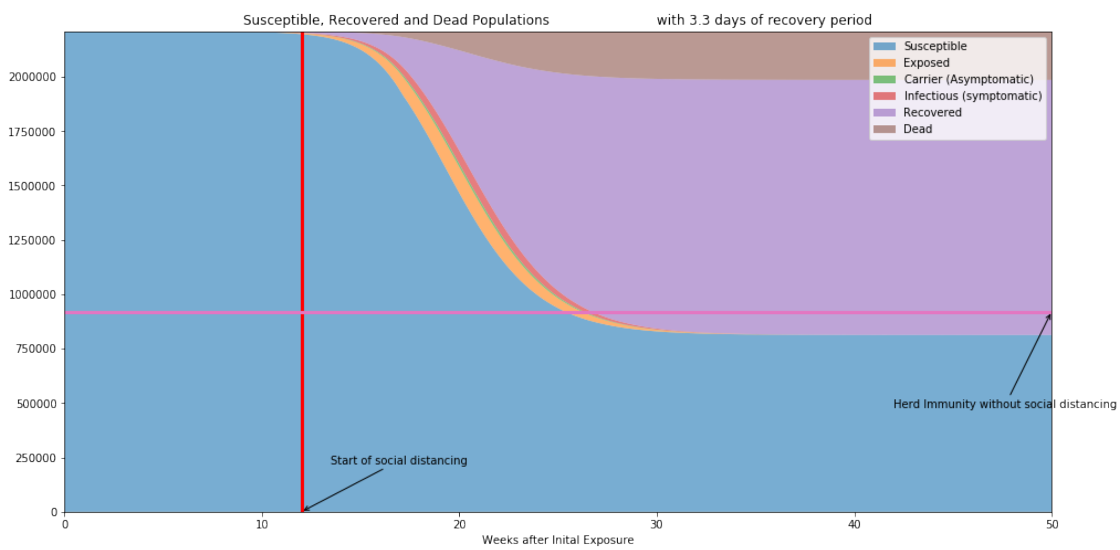


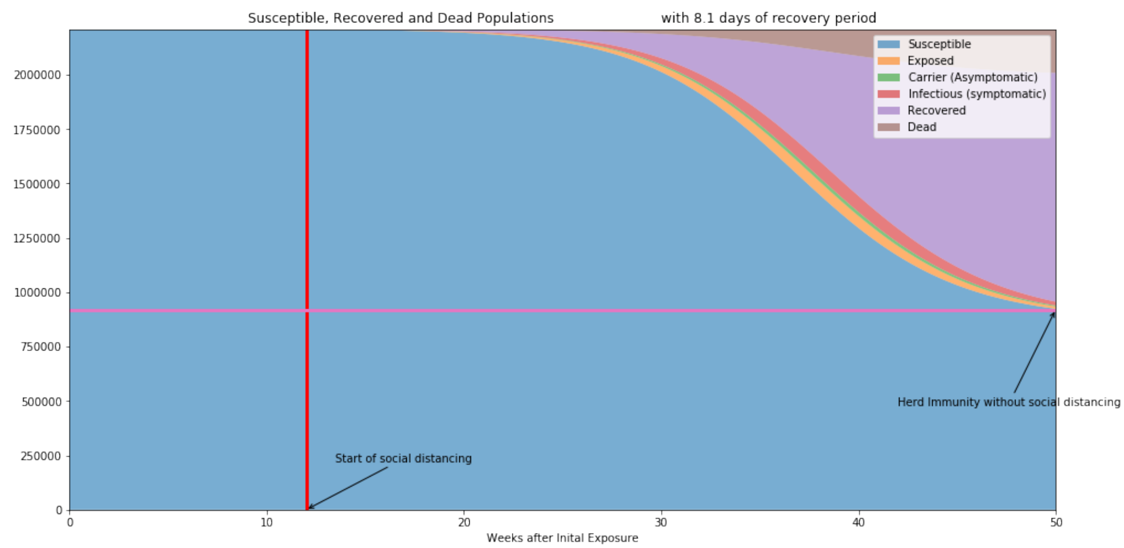
1.4 Variation in Recovery Period

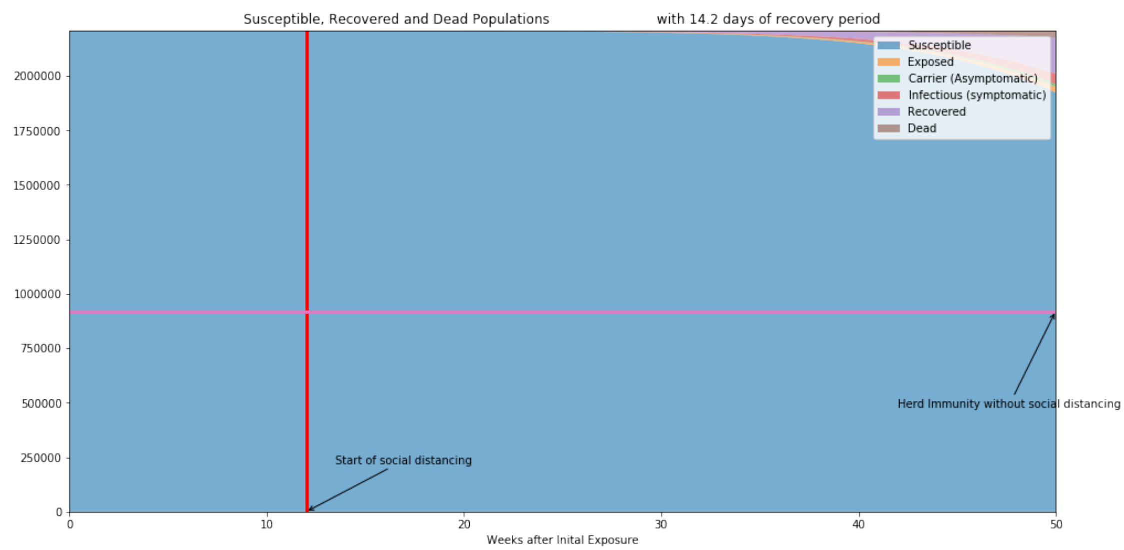
Next, we vary the recovery period for a infected/carrier. As we know the rate of recovery $\tau_{recovery}$ is inversely proportional to γ, and β which are the rate of transition from susceptible to exposed stage and exposed to either of the infected/carrier stage respectively. We see that with larger $\tau_{recovery}$, i.e. with a lower value of γ the spread of disease increases.

In [78]:
```
Recovs = [3.3, 8.1, 14.2]

for rec in Recovs:
    ax = run(2.4, 0.2, 0.2, 5.1, rec, 2209000, 3, 12, 20)
    ax[0].set_title('Susceptible, Recovered and Dead Populations \
                    with {0:1.1f} days of recovery period'.format(rec))
```





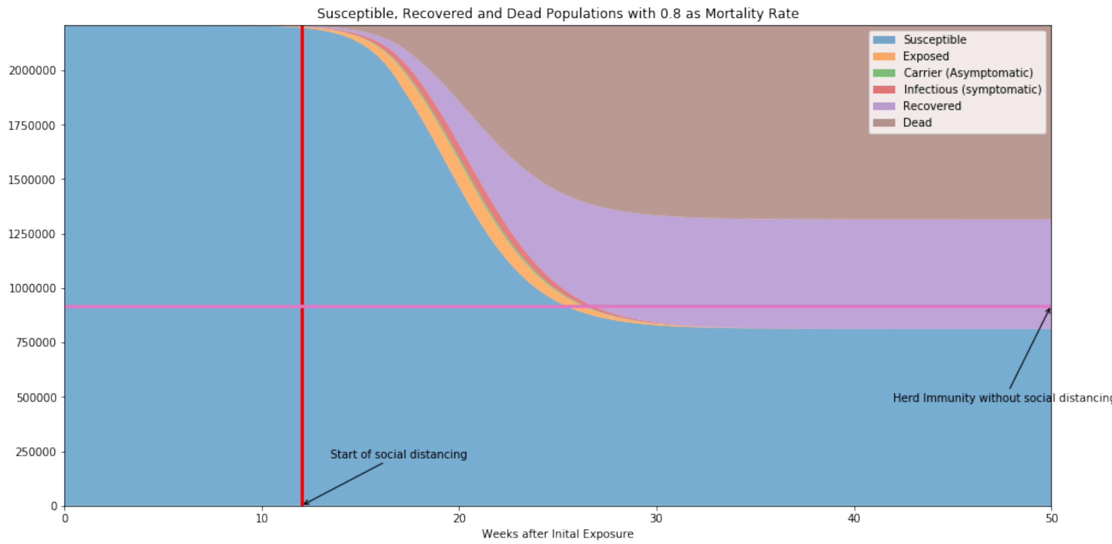


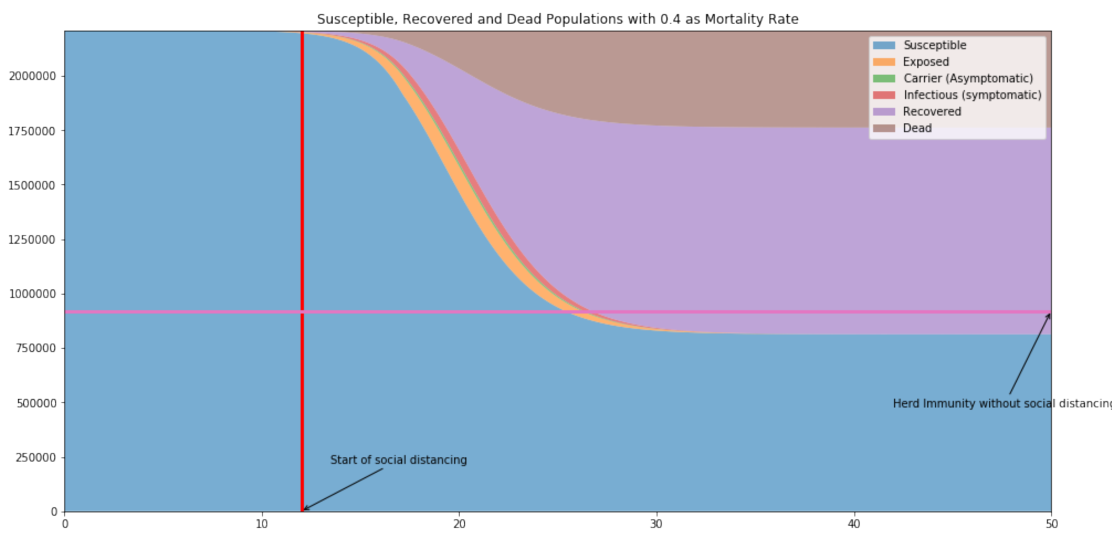
1.5 Variation in Mortality Rate

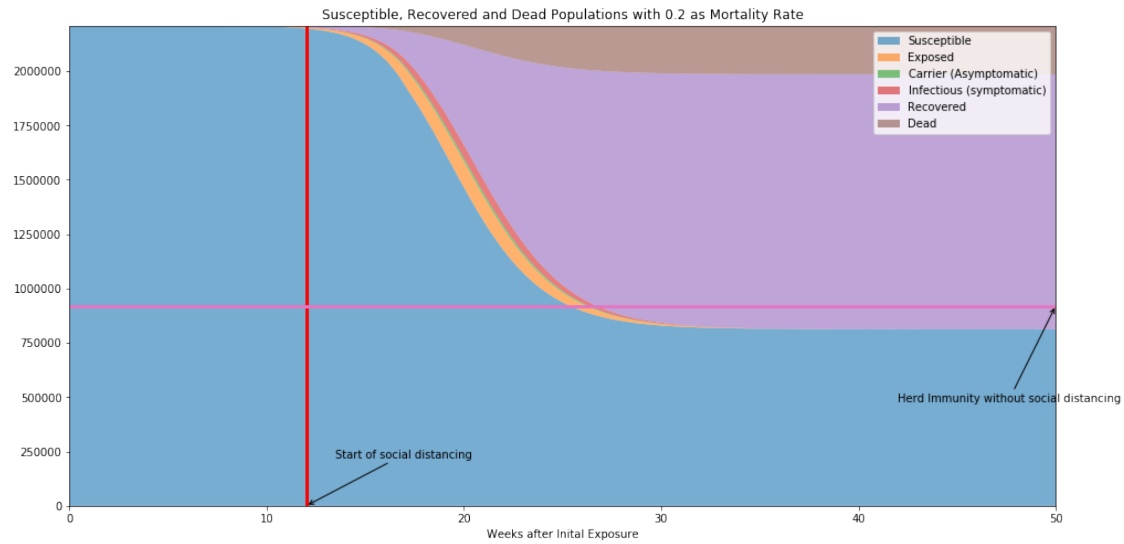
Next, we vary M_r, the mortality rate. As suggested from it name, it only affects the total number of patients dying due to the disease.

In [46]: Rts $=$ [0.8, 0.4, 0.2]

```
for rt in Rts:
    ax = run(2.4, rt, 0.2, 5.1, 3.3, 2209000, 3, 12, 20)
    ax[0].set_title('Susceptible, Recovered and Dead Populations\
                    with {0:1.1f} as Mortality Rate'.format(rt))
```





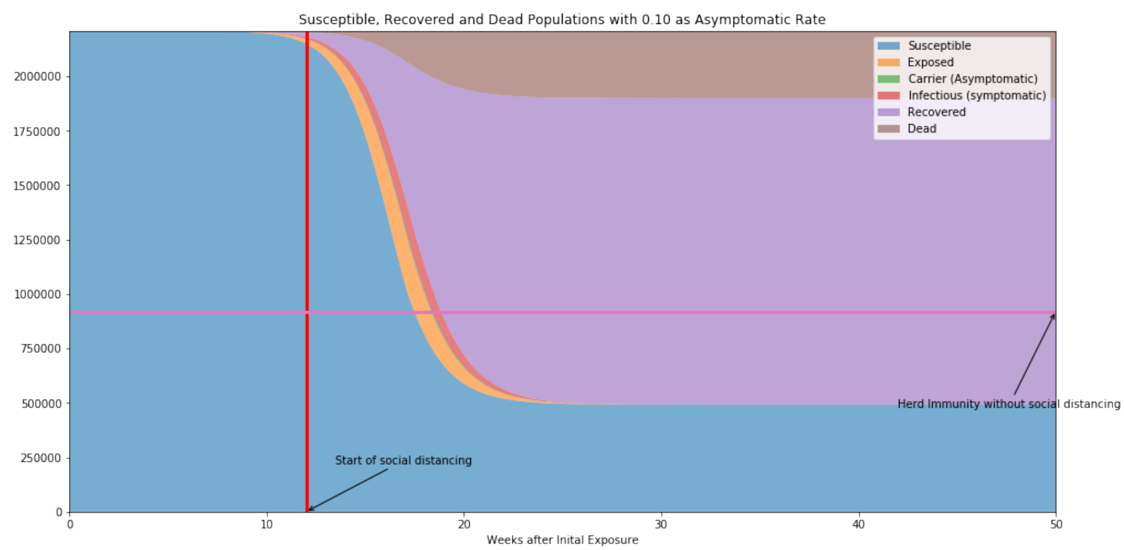


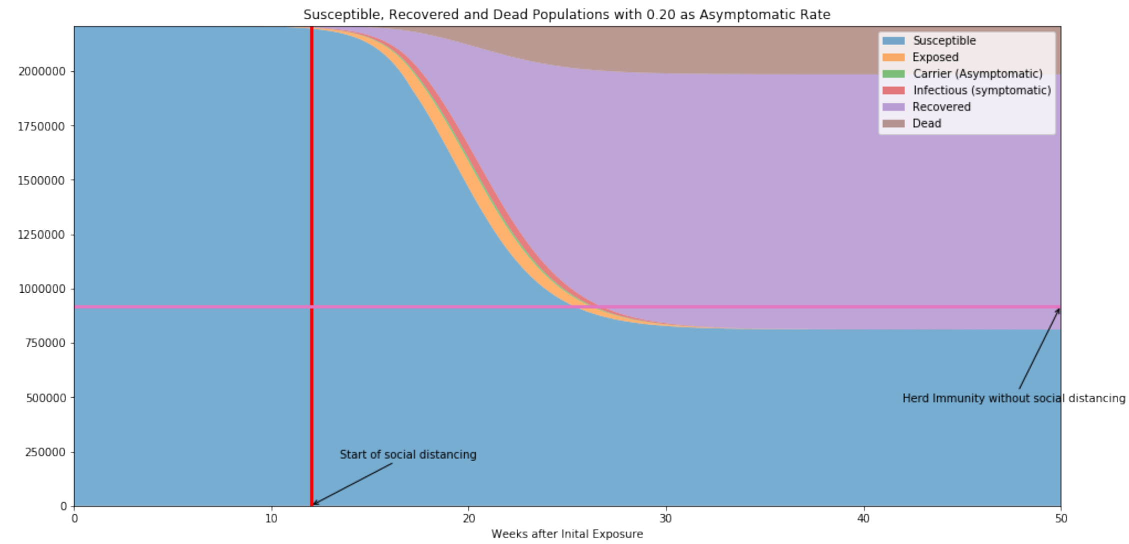
1.6 Variation in Asymptomatic Rate

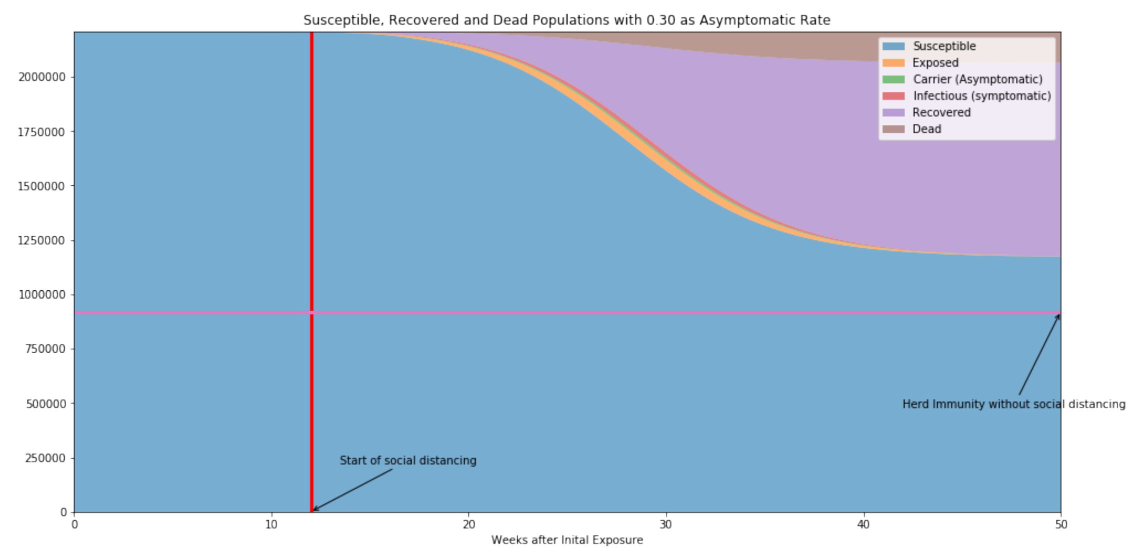
As we increase the asymptomatic rate, my inital guess was that it should have increased the disease spread but somehow it actually decreases the spread. Will have to look more into it. :(

In [102]:
```
Rts = [0.1, 0.2, 0.3]

for rt in Rts:
    ax = run(2.4, 0.2, rt, 5.1, 3.3, 2209000, 3, 12, 20)
    ax[0].set_title('Susceptible, Recovered and Dead Populations with {0:1.2f} as Asymptomatic Rate'.form
```





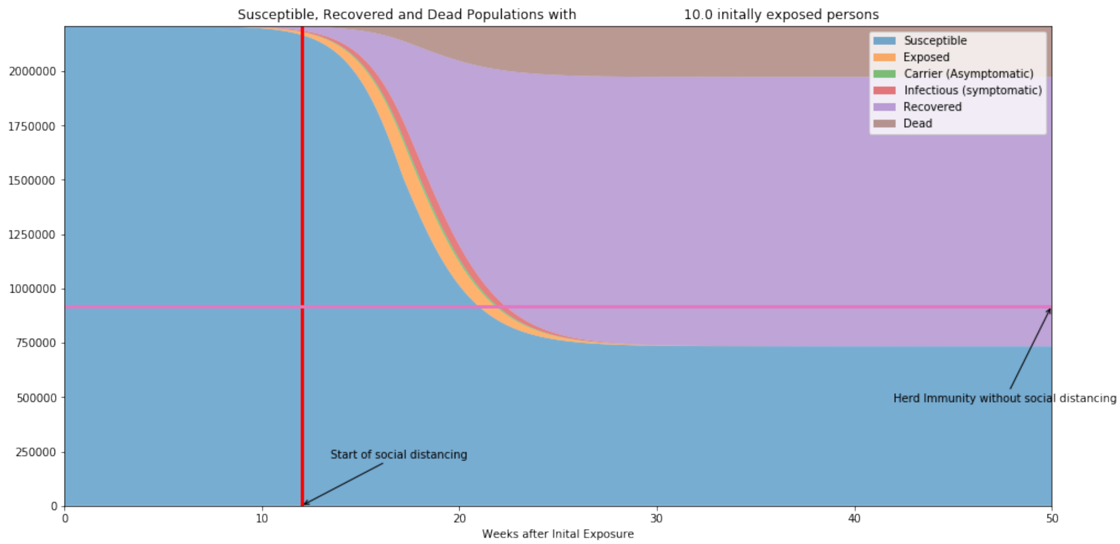


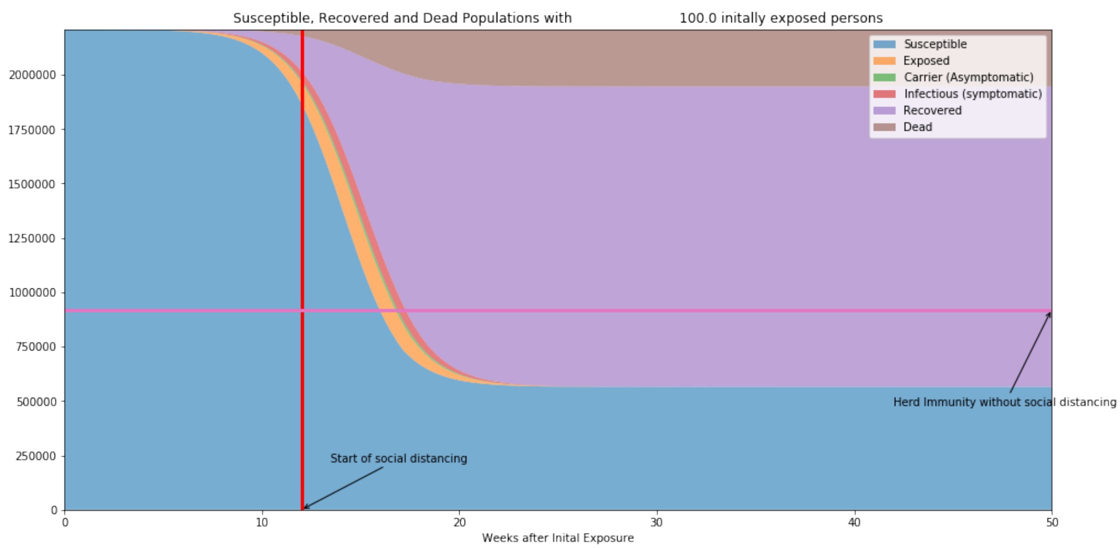
1.7 Variation in Initial Numbers Exposed

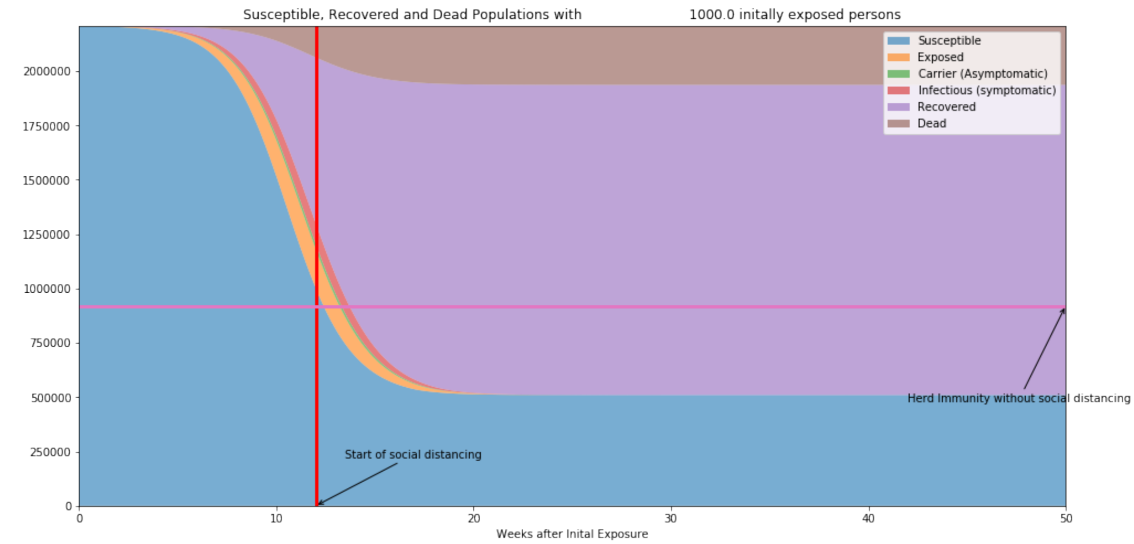
With increase in initial number of exposures, spread of disease increases.

In [76]:
```
Incs = [10, 100, 1000]

    for inc in Incs:
        ax = run(2.4, 0.2, 0.2, 5.1, 3.3, 2209000, inc, 12, 20)
        ax[0].set_title('Susceptible, Recovered and Dead Populations with \
                        {0:1.1f} initally exposed persons'.format(inc))
```


Susceptible, Recovered and Dead Populations with 10.0 initally exposed persons
Susceptible
Exposed
Carrier (Asymptomatic)
Infectious (symptomatic)
Recovered
Dead
Herd Immunity without social distancing
Start of social distancing
Weeks after Inital Exposure
0
250000
500000
750000
1000000
1250000
1500000
1750000
2000000
10
20
30
40
50


Susceptible, Recovered and Dead Populations with 100.0 initally exposed persons
Susceptible
Exposed
Carrier (Asymptomatic)
Infectious (symptomatic)
Recovered
Dead
Herd Immunity without social distancing
Start of social distancing
Weeks after Inital Exposure
0
250000
500000
750000
1000000
1250000
1500000
1750000
2000000
10
20
30
40
50

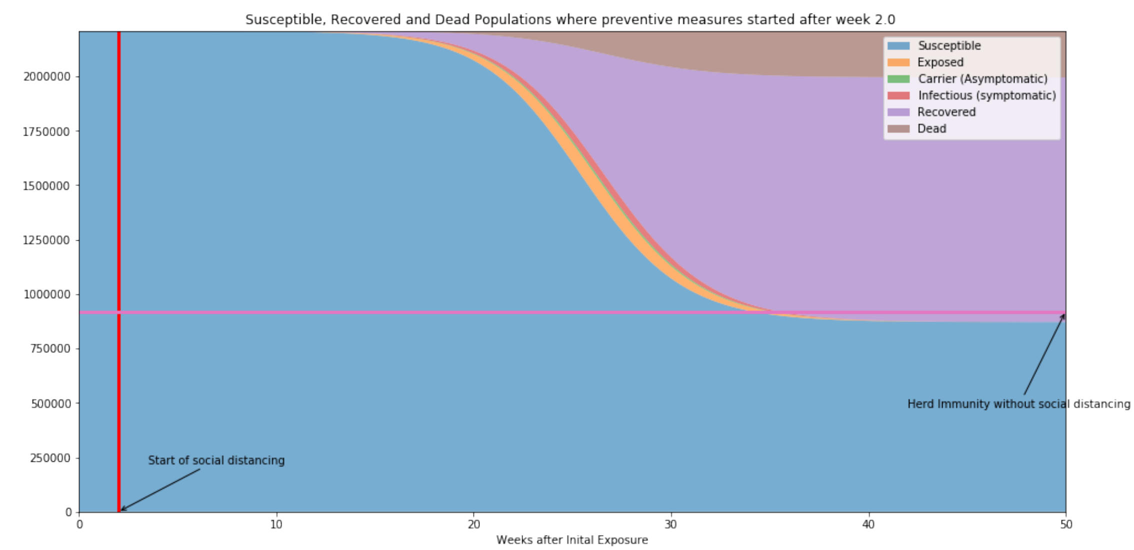


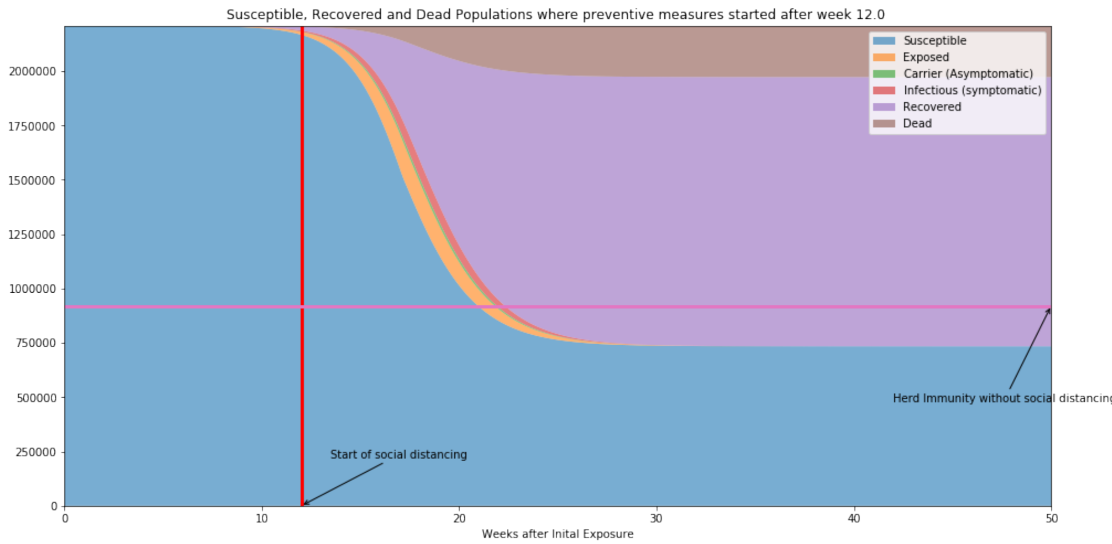
1.8 Variation in the starting preventive measures

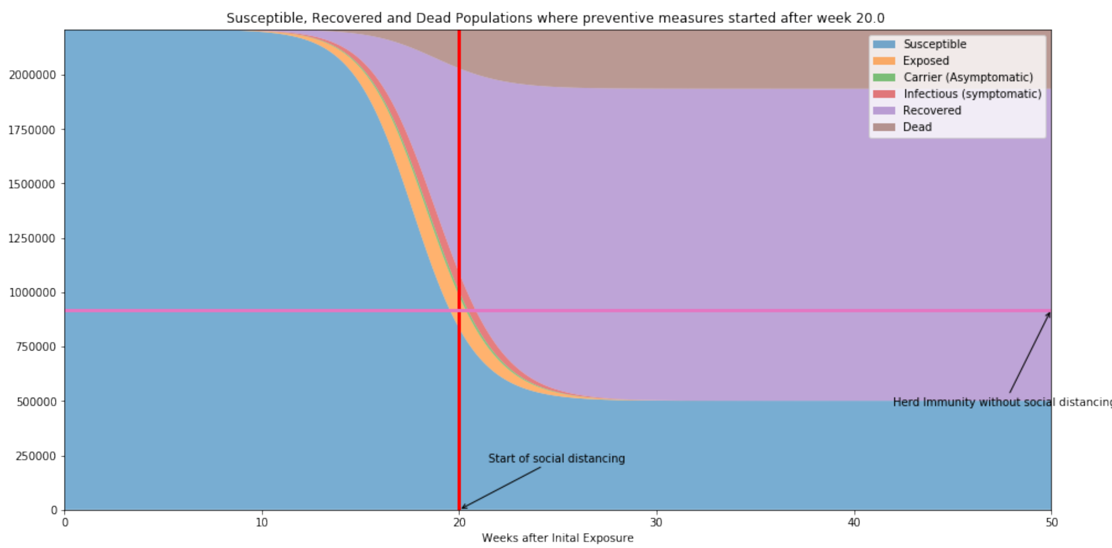
Earlier the preventive memasures are put in place, lesser is the spread.

In [67]:
```
Incs = [2, 12, 20]

for inc in Incs:
    ax = run(2.4, 0.2, 0.2, 5.1, 3.3, 2209000, 10, inc, 20)
    ax[0].set_title('Susceptible, Recovered and Dead Populations where \
                    preventive measures started after week {0:1.1f} '.format(inc))
```







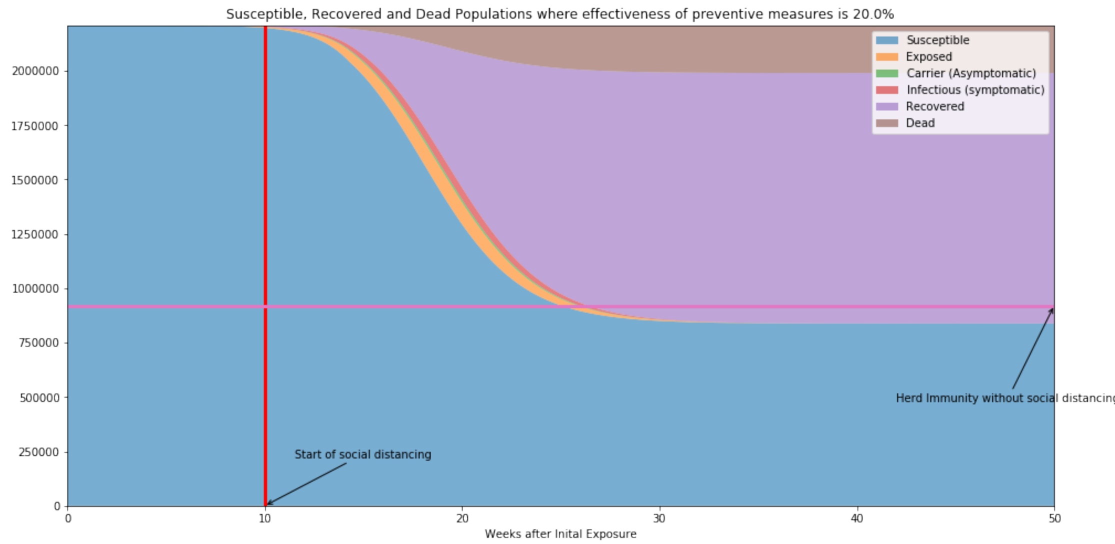
1.9 Variation in the effectiveness of preventive measures

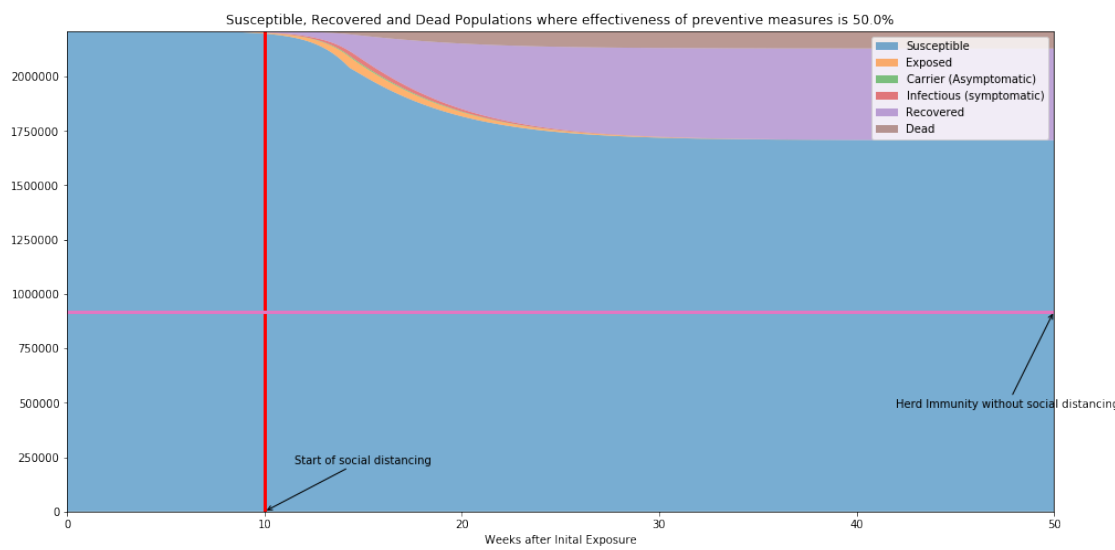
Efficiency of the preventive measure plays a more vital role in controlling the disease spread. However, it doesn't matter if preventuve measures are being set at a much later time, i.e. when sufficient spread has already taken place.

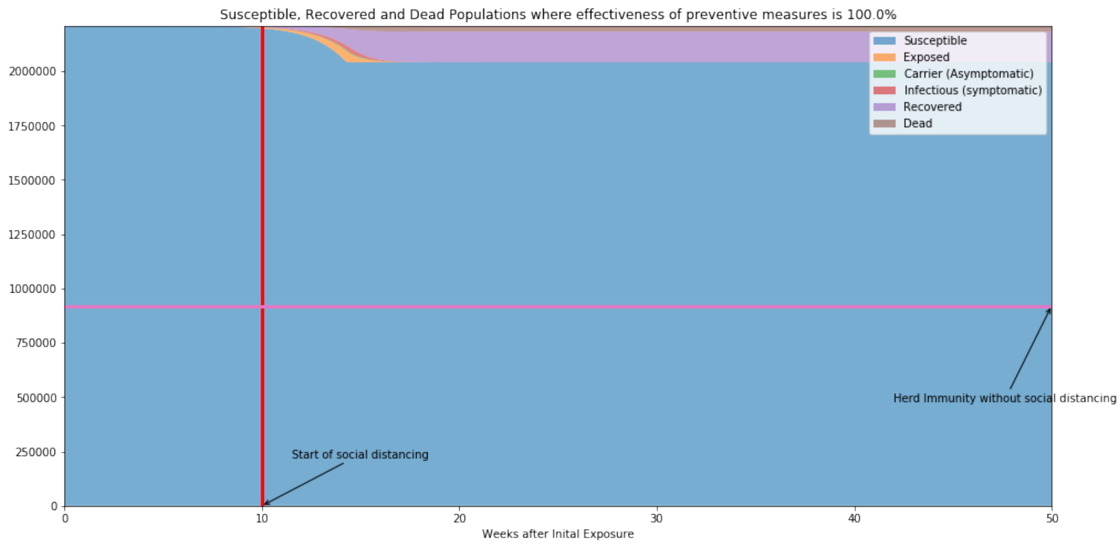
In [72]:
```
Incs = [20, 50, 100]

    for inc in Incs:
        ax = run(2.4, 0.2, 0.2, 5.1, 3.3, 2209000, 10, 10, inc)
```

```
ax[0].set_title('Susceptible, Recovered and Dead Populations where \
                effectiveness of preventive measures is {0:1.1f}%'.format(inc))
```






Susceptible, Recovered and Dead Populations where effectiveness of preventive measures is 100.0%
Susceptible
Exposed
Carrier (Asymptomatic)
Infectious (symptomatic)
Recovered
Dead
2000000
1750000
1500000
1250000
1000000
750000
500000
250000
0
0
10
20
30
40
50
Weeks after Inital Exposure
Start of social distancing
Herd Immunity without social distancing