CHIME Interactive

Anshul Saxena and Gabriel Odom

4/4/2020

## Setup

We need to load some R packages before we can perform our computing.

library(reticulate)  
library(tidyverse)

The shell script to install the packages we need is in src/install\_python\_packages\_20200403.sh. This creates a conda environment called env\_sflCHIME and activates it. At current, I have been unable to run these commands on Windows.

On Mac, I have a Conda environment created for this project. In the terminal, I type which python and it tells me

which python

/Users/gabrielodom/anaconda3/envs/env\_sflCHIME/bin/python

If it says

/usr/bin/python

(or some other place), then something is wrong.

I specifically want this python environment, so I “point” reticulate to the particular environment that we installed with Anaconda, replete with the version of Python we selected and all associated libraries. (**NOTE**: this line must be executed before any other reticulate functions or Python calls, otherwise it will have no effect.)

use\_python(  
 "/Users/gabrielodom/anaconda3/envs/env\_sflCHIME/bin/python",  
 required = TRUE  
)

We also have some Python libraries to load.

from functools import reduce  
from typing import Tuple, Dict, Any  
import pandas as pd  
import streamlit as st  
import numpy as np  
import altair as alt

IT WORKS!!!!!! (This seriously took a full weekend to figure out, and then I called in the big guns: Tim N. and Athina H. Note that this is not compatible with environments created via Anaconda, at least not by default.)

## Parameters

Now, we want to set our initial parameters. We organise them the same way as on the CHARM app.

### Hospital Parameters

# Regional Population  
S\_default = 2761581  
S = S\_default  
  
# Hospital Market Share  
Penn\_market\_share = 0.15  
  
# Currently Hospitalised Patients (at current hospital?)  
known\_cases = 400  
current\_hosp = known\_cases  
  
# Currently Hospitalised Patients  
known\_infections = 1000  
initial\_infections = known\_infections

### Spread and Contact Parameters

# Doubling Time  
doubling\_time = 6  
  
# Social Distancing (% reduction in social contact)  
relative\_contact\_rate = 0

### Severity Rates

# Hospitalisation (% of total infections)  
hosp\_rate = 0.05  
  
# ICU (% of total infections)  
icu\_rate = 0.02  
  
# Ventilated (% of total infections)  
vent\_rate = 0.01

### Severity Times

# Infectious Days  
recovery\_days = 14.0  
  
# Average hospital length of stay (days)  
hosp\_los = 7  
  
# Average ICU length of stay (days)  
icu\_los = 9  
  
# Average length of stay on a ventilator (days)  
vent\_los = 10

## Directly Estimated Parameters

Given the parameters above, we calculate the following disease spread parameters.

### SIR Initial Values

total\_infections = current\_hosp / Penn\_market\_share / hosp\_rate  
print(total\_infections)

## 53333.333333333336

detection\_prob = initial\_infections / total\_infections  
print(detection\_prob)

## 0.01875

S, I, R = S, initial\_infections / detection\_prob, 0  
print(I)

## 53333.333333333336

intrinsic\_growth\_rate = 2 \*\* (1 / doubling\_time) - 1  
print(intrinsic\_growth\_rate)

## 0.12246204830937302

**NOTE**: the total infections and initial infections are equal at .

### SIR Transition Rates

gamma = 1 / recovery\_days  
# mean recovery rate, gamma, (in 1/days).  
print(gamma)

## 0.07142857142857142

beta = (intrinsic\_growth\_rate + gamma) / S \* (1 - relative\_contact\_rate)  
# Contact rate, beta  
# {rate based on doubling time} / {initial S}  
print(beta)

## 7.021000641949103e-08

r\_t = beta / gamma \* S   
# Current Reproduction Number  
# r\_t is r\_0 after distancing  
print(r\_t)

## 2.7144686763312222

r\_naught = r\_t / (1 - relative\_contact\_rate)  
# Initial Reproduction Number  
print(r\_naught)

## 2.7144686763312222

doubling\_time\_t = 1 / np.log2(beta \* S - gamma + 1)   
# Current Doubling Time  
# doubling time after distancing  
print(doubling\_time\_t)

## 5.999999999999998

As an aside, if I want to access any of these values in R (for instance, ), I can call them as py$r\_naught, which yields 2.7144687.

## Model Specification

We can now specify the SIR model at outset and over time.

# SIR at Outset  
def sir(y, beta, gamma, N):  
 S, I, R = y  
 Sn = (-beta \* S \* I) + S  
 In = (beta \* S \* I - gamma \* I) + I  
 Rn = gamma \* I + R  
 if Sn < 0:  
 Sn = 0  
 if In < 0:  
 In = 0  
 if Rn < 0:  
 Rn = 0  
   
 scale = N / (Sn + In + Rn)  
 return Sn \* scale, In \* scale, Rn \* scale  
  
# Dynamic SIR   
def sim\_sir(S, I, R, beta, gamma, n\_days, beta\_decay = None):  
 N = S + I + R  
 s, i, r = [S], [I], [R]  
 for day in range(n\_days):  
 y = S, I, R  
 S, I, R = sir(y, beta, gamma, N)  
 if beta\_decay:  
 beta = beta \* (1 - beta\_decay)  
 s.append(S)  
 i.append(I)  
 r.append(R)  
   
 s, i, r = np.array(s), np.array(i), np.array(r)  
 return s, i, r

Run the model, assuming no decay in contact rate over the next 30 days.

n\_days = 30  
beta\_decay = 0.0  
  
s, i, r = sim\_sir(S, I, R, beta, gamma, n\_days, beta\_decay = beta\_decay)

**ERROR**: Error in py\_call\_impl(callable, dots$args, dots$keywords) : TypeError: unsupported operand type(s) for +: 'float' and 'type'

We now inspect the results of this model