S\_I\_R V2 01.09.2021

Elvis Kirui

2021-09-12

# R Programming

## loading Relevant packages and Data Set

#Import relevant packages  
  
## tidyverse includes readr, ggplot2, dplyr, forcats, tibble, tidyr, purrr, stringr  
library(tidyverse)   
library(readxl)  
library(janitor)  
library(lubridate)  
## loading the csv data set  
  
setwd('F:\\Documents\\ElvisKip\\Sir\_Models 06.08.2021')  
  
#setwd('C:/Users/atele/Desktop/Reinp/ElvisKip/SIR')  
  
S\_I\_R\_V2 <- read\_excel("S\_I\_R V2.xlsx", sheet = "Sheet1")%>%  
 clean\_names()%>%  
 select(-6,-7)%>%  
mutate(population = gsub(",", "", population))%>%  
mutate(population = as.numeric(population))  
  
  
View(S\_I\_R\_V2)

## Sir Models

### Building a two compartment model in R

The mean infectious period for covid- 19 is 14 days.

The deSolve package in R contains functions to solve initial value problems of a system of first-order ordinary differential equations (‘ODE’).

Step 1: Define the number of people in the infected and recovered cohort, the recovery rate gamma (γ=1/14) and the follow up duration depending on the month.

Step 2: Specify the model function

In the model function, specify the differential equations. The function has three input arguments :

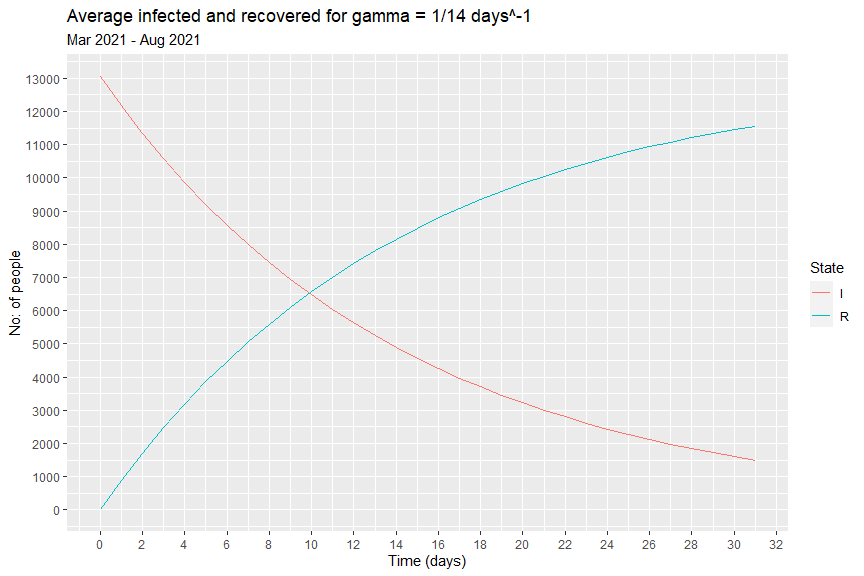
* time: The time-points at which we want to solve the model at
* state:It stores the number of people in each compartment at every time point
* parameters: The names and values of model parameters

Step 3 : Solve the model using ode() in deSolve package

Step 4: Plot the output using ggplot2

#Load required libraries  
library(deSolve)  
library(reshape2)  
library(plotly)

# Model inputs  
  
initial\_values1<-c(I=S\_I\_R\_V2$number\_of\_infected[S\_I\_R\_V2$x2020 == "Total"]/18, R=0)  
  
#initial\_values1<-c(I=sum(Owid\_Covid\_Kenya$cases\_new)/18, R=0)  
  
parameters1<-c(gamma=round(1/14,2))  
  
  
  
# Time points  
  
time1=seq(from=0,t=31,by=1)  
  
# SIR model function  
  
sir\_model1 <-function(time1,state,parameters1){  
 with(as.list(c(state,parameters1)),{  
 dI=-gamma\*I  
 dR=gamma\*I  
 return(list(c(dI,dR)))  
 }  
)   
}  
  
#Solving the differential equation  
  
output1<-as.data.frame(ode(y=initial\_values1,func = sir\_model1,parms=parameters1,times = time1))  
  
  
#To convert the dataframe to long format  
out\_long1=melt(output1,id="time")   
  
  
 ggplot(data = out\_long1,   
 aes(x = time, y = value, colour = variable)) +   
 geom\_line() +   
 scale\_x\_continuous(n.breaks = 16)+  
 scale\_y\_continuous(n.breaks = 13)+  
 labs(title = "Average infected and recovered for gamma = 1/14 days^-1",  
 subtitle = "Mar 2021 - Aug 2021",  
 x = "Time (days)",  
 y = "No: of people")+  
 scale\_color\_discrete(name="State")



we observe that on a monthly average, half of the infected cohort recovers in around 10 days

### Simple SIR Model

**What are the assumptions of an SIR model?**

* Homogeneous population : Individuals in the same compartment is subject to the same hazards
* Well-mixed population : All susceptible have the same risk of getting infected
* Immunity forever : Individuals who recovered from the disease are immune forever

The last assumption leads to the observation that SIR model is good for scenarios where population doesn’t change and the immunity is stable during the simulation period.

These assumptions give us the following set of differential equations:

dS/dt = -beta \* S \* (I/N)

dI/dt= beta \* S \* (I/N)- gamma \* I

dR/dt= gamma \* I

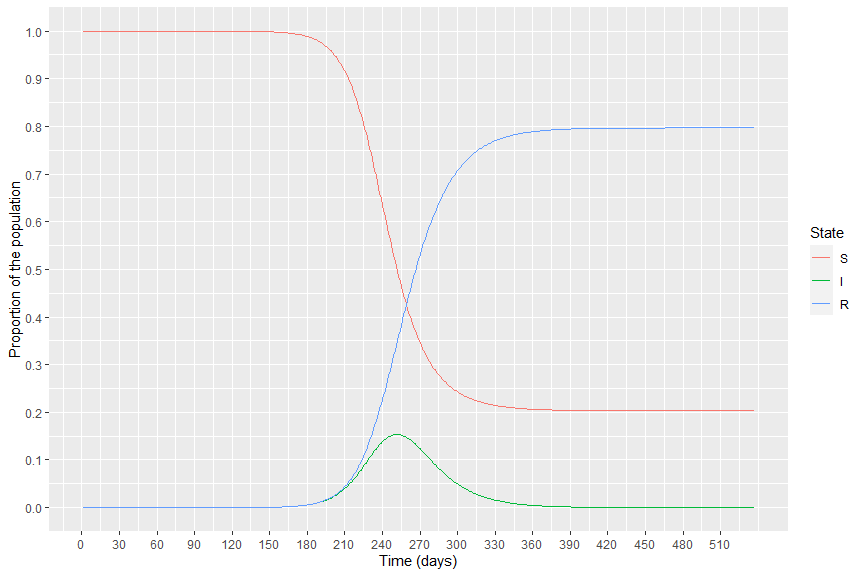
where S,I and R represents the number of susceptible,infected and recovered individuals in each compartment respectively and N=S+I+R.

gamma(γ) is the recovery rate .(**controls transition between I and R**)

beta (β) is the infection rate i.e the average number of secondary infections per unit time. (**controls transition between S and I**)

#The mean infectious period for covid- 19 is 14 days and a person infects one #person on average every 7 days.  
  
# Model inputs  
  
initial\_state\_values=c(S=as.numeric(S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"])-1,  
 I=1,R=0)  
parameters=c(gamma=round(1/14,2),beta=round(1/7,2))  
  
# Time points  
  
time=seq(from=1,  
 to=length(seq(as.Date("2020/03/13"), as.Date("2021/08/31"), by=1))  
 ,by=1)

# SIR model function   
  
sir\_model2 <- function(time,state,parameters){  
 with(as.list(c(state,parameters)),{  
 N=S+I+R  
 dS=-beta\*S\*(I/N)  
 dI= beta\*S\*(I/N)-gamma\*I  
 dR= gamma\*I  
   
 return(list(c(dS,dI,dR)))  
 }  
 )  
}  
  
  
#Solving the differential equations  
output2<-as.data.frame(ode(y=initial\_state\_values,func = sir\_model2,parms=parameters,times = time))%>%  
 mutate(S = format(S, scientific = F, digits = 0))%>%  
 mutate(I = format(I, scientific = F, digits = 0))%>%  
 mutate(R = format(R, scientific = F, digits = 0))%>%  
 mutate(across(c(2:4), as.numeric))  
  
  
out\_long2 <- melt(output2,id="time")  
# To plot the proportion of susceptible, infected and recovered individuals over time  
ggplot(data = out\_long2,   
 aes(x = time,   
 y = value/S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"], colour = variable)) +   
 geom\_line() +  
 #scale\_x\_continuous(n.breaks = 20)+  
 scale\_x\_continuous(breaks = seq(0, length(seq(as.Date("2020/03/13"), as.Date("2021/08/31"), by=1)), by = 30))+  
 scale\_y\_continuous(n.breaks = 10)+  
 xlab("Time (days)")+  
 ylab("Proportion of the population")+  
 scale\_color\_discrete(name="State")



## Estimating Basic reproduction number and infection rate (beta)

### Counting Process

The estimator theta is given by

theta = -In(1-p)/R(T)

s.e(theta) = (n/(S(0)+0.5)+n/(S(0)+0.5)-theta^2\*U(T))^(1/2)/R(T)

R\_o = n \* theta

se\_R\_0 = n \* s.e(theta)

where p is the observed final size (p = 1 − S(T)/n) at the end of the epidemic at time T.

n is the size of the population

S(T) is the size of the susceptible population at time T

S(0) is the size of the susceptible population at time 0

R(T) is the size of recovered individuals at time T

U(T)/R\_eff = R\_o \* S(T)/S(0)

#As at march 2020 - August 2021  
  
p\_Mar20\_Aug21 = 1- ((S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]- S\_I\_R\_V2$number\_of\_infected[S\_I\_R\_V2$x2020 == "Total"]-S\_I\_R\_V2$number\_of\_recovered[S\_I\_R\_V2$x2020 == "Total"])/(S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]))  
  
theta\_Mar20\_Aug21 = -log(1-p\_Mar20\_Aug21)/S\_I\_R\_V2$number\_of\_recovered[S\_I\_R\_V2$x2020 == "Total"]  
  
R\_o\_Mar20\_Aug21 = S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]\*theta\_Mar20\_Aug21  
  
Beta\_Mar20\_Aug21 = R\_o\_Mar20\_Aug21 \* (1/14)  
  
Ut\_Mar20\_Aug21 = R\_o\_Mar20\_Aug21 \* ((S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]- S\_I\_R\_V2$number\_of\_infected[S\_I\_R\_V2$x2020 == "Total"]-S\_I\_R\_V2$number\_of\_recovered[S\_I\_R\_V2$x2020 == "Total"])/(S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]-1))  
  
  
se\_theta\_Mar20\_Aug21 = (((S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]/((S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]-1)^2 + 0.5))+(S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]/((S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]-1)^2+0.5))-((theta\_Mar20\_Aug21^2)\*Ut\_Mar20\_Aug21))^(1/2))/S\_I\_R\_V2$number\_of\_recovered[S\_I\_R\_V2$x2020 == "Total"]  
  
  
se\_R\_o\_Mar20\_Aug21 = S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"] \* se\_theta\_Mar20\_Aug21  
  
R\_o\_Mar20\_Aug21\_interval = paste(R\_o\_Mar20\_Aug21 - 1.96\*se\_R\_o\_Mar20\_Aug21, R\_o\_Mar20\_Aug21 + 1.96\*se\_R\_o\_Mar20\_Aug21)

#As at beginning   
  
p\_beginning = 1- ((S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]- 84-79)/(S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]))  
  
theta\_beginning = -log(1-p\_beginning)/79  
  
R\_o\_beginning = S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]\*theta\_beginning  
  
Beta\_beginning = R\_o\_beginning \* (1/14)

#Total Lockdown  
  
p\_TotalLockdown = 1 - ((S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]- sum(S\_I\_R\_V2$number\_of\_infected[which(S\_I\_R\_V2$status == "Total Lockdown")])-sum(S\_I\_R\_V2$number\_of\_recovered[which(S\_I\_R\_V2$status == "Total Lockdown")]))/(S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]))  
  
theta\_TotalLockdown = -log(1-p\_TotalLockdown)/sum(S\_I\_R\_V2$number\_of\_recovered[which(S\_I\_R\_V2$status == "Total Lockdown")])   
  
R\_o\_TotalLockdown = S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]\*theta\_TotalLockdown  
  
Beta\_TotalLockdown = R\_o\_TotalLockdown \* (1/14)  
  
Ut\_TotalLockdown = R\_o\_TotalLockdown \* ((S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]- sum(S\_I\_R\_V2$number\_of\_infected[which(S\_I\_R\_V2$status == "Total Lockdown")])-sum(S\_I\_R\_V2$number\_of\_recovered[which(S\_I\_R\_V2$status == "Total Lockdown")]))/(S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]-1))  
  
  
se\_theta\_TotalLockdown = (((S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]/((S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]-1)^2 + 0.5))+(S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]/((S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]-1)^2+0.5))-((theta\_TotalLockdown^2)\*Ut\_TotalLockdown))^(1/2))/sum(S\_I\_R\_V2$number\_of\_recovered[which(S\_I\_R\_V2$status == "Total Lockdown")])  
  
  
se\_R\_o\_TotalLockdown = S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"] \* se\_theta\_TotalLockdown  
  
R\_o\_TotalLockdown\_interval = paste(R\_o\_TotalLockdown - 1.96\*se\_R\_o\_TotalLockdown, R\_o\_TotalLockdown + 1.96\*se\_R\_o\_TotalLockdown)

#Partially opening  
  
p\_Partiallyopening = 1 - ((S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]- sum(S\_I\_R\_V2$number\_of\_infected[which(S\_I\_R\_V2$status == "Partially opening")])-sum(S\_I\_R\_V2$number\_of\_recovered[which(S\_I\_R\_V2$status == "Partially opening")]))/(S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]))  
  
theta\_Partiallyopening = -log(1-p\_Partiallyopening)/sum(S\_I\_R\_V2$number\_of\_recovered[which(S\_I\_R\_V2$status == "Partially opening")])   
  
R\_o\_Partiallyopening = S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]\*theta\_Partiallyopening  
  
Beta\_Partiallyopening = R\_o\_Partiallyopening \* (1/14)  
  
Ut\_Partiallyopening = R\_o\_Partiallyopening \* ((S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]- sum(S\_I\_R\_V2$number\_of\_infected[which(S\_I\_R\_V2$status == "Partially opening")])-sum(S\_I\_R\_V2$number\_of\_recovered[which(S\_I\_R\_V2$status == "Partially opening")]))/(S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]- sum(S\_I\_R\_V2$number\_of\_infected[which(S\_I\_R\_V2$status == "Total Lockdown")])-sum(S\_I\_R\_V2$number\_of\_recovered[which(S\_I\_R\_V2$status == "Total Lockdown")])))  
  
  
se\_theta\_Partiallyopening = (((S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]/((S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]- sum(S\_I\_R\_V2$number\_of\_infected[which(S\_I\_R\_V2$status == "Total Lockdown")])-sum(S\_I\_R\_V2$number\_of\_recovered[which(S\_I\_R\_V2$status == "Total Lockdown")]))^2 + 0.5))+(S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]/((S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]- sum(S\_I\_R\_V2$number\_of\_infected[which(S\_I\_R\_V2$status == "Total Lockdown")])-sum(S\_I\_R\_V2$number\_of\_recovered[which(S\_I\_R\_V2$status == "Total Lockdown")]))^2+0.5))-((theta\_Partiallyopening^2)\*Ut\_Partiallyopening))^(1/2))/sum(S\_I\_R\_V2$number\_of\_recovered[which(S\_I\_R\_V2$status == "Partially opening")])  
  
  
se\_R\_o\_Partiallyopening = S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"] \* se\_theta\_Partiallyopening  
  
R\_o\_Partiallyopening\_interval = paste(R\_o\_Partiallyopening - 1.96\*se\_R\_o\_Partiallyopening, R\_o\_Partiallyopening + 1.96\*se\_R\_o\_Partiallyopening)

#Reopening  
  
p\_Reopening = 1 - ((S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]- sum(S\_I\_R\_V2$number\_of\_infected[which(S\_I\_R\_V2$status == "Reopening")])-sum(S\_I\_R\_V2$number\_of\_recovered[which(S\_I\_R\_V2$status == "Reopening")]))/(S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]))  
  
theta\_Reopening = -log(1-p\_Reopening)/sum(S\_I\_R\_V2$number\_of\_recovered[which(S\_I\_R\_V2$status == "Reopening")])   
  
R\_o\_Reopening = S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]\*theta\_Reopening  
  
Beta\_Reopening = R\_o\_Reopening \* (1/14)  
  
Ut\_Reopening = R\_o\_Reopening \* ((S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]- sum(S\_I\_R\_V2$number\_of\_infected[which(S\_I\_R\_V2$status == "Reopening")])-sum(S\_I\_R\_V2$number\_of\_recovered[which(S\_I\_R\_V2$status == "Reopening")]))/(S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]- sum(S\_I\_R\_V2$number\_of\_infected[which(S\_I\_R\_V2$status == "Total Lockdown"| S\_I\_R\_V2$status == "Partially opening")])-sum(S\_I\_R\_V2$number\_of\_recovered[which(S\_I\_R\_V2$status == "Total Lockdown"|S\_I\_R\_V2$status == "Partially opening")])))  
  
  
se\_theta\_Reopening = (((S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]/((S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]- sum(S\_I\_R\_V2$number\_of\_infected[which(S\_I\_R\_V2$status == "Total Lockdown"| S\_I\_R\_V2$status == "Partially opening")])-sum(S\_I\_R\_V2$number\_of\_recovered[which(S\_I\_R\_V2$status == "Total Lockdown"|S\_I\_R\_V2$status == "Partially opening")]))^2 + 0.5))+(S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]/((S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"]- sum(S\_I\_R\_V2$number\_of\_infected[which(S\_I\_R\_V2$status == "Total Lockdown"| S\_I\_R\_V2$status == "Partially opening")])-sum(S\_I\_R\_V2$number\_of\_recovered[which(S\_I\_R\_V2$status == "Total Lockdown"|S\_I\_R\_V2$status == "Partially opening")]))^2+0.5))-((theta\_Reopening^2)\*Ut\_Reopening))^(1/2))/sum(S\_I\_R\_V2$number\_of\_recovered[which(S\_I\_R\_V2$status == "Reopening")])  
  
  
se\_R\_o\_Reopening = S\_I\_R\_V2$population[S\_I\_R\_V2$x2020 == "Total"] \* se\_theta\_Reopening  
  
R\_o\_Reopening\_interval = paste(R\_o\_Reopening - 1.96\*se\_R\_o\_Reopening, R\_o\_Reopening + 1.96\*se\_R\_o\_Reopening)

counting\_process\_output <- tibble( status = c("Beginning", "Mar2020-Aug2021", "Total Lockdown", "Partially opening", "Reopening" ),  
 Ro = round( c(R\_o\_beginning, R\_o\_Mar20\_Aug21 ,  
 R\_o\_TotalLockdown, R\_o\_Partiallyopening, R\_o\_Reopening),4),  
 infection\_rate\_Beta = round(c(Beta\_beginning, Beta\_Mar20\_Aug21 ,  
 Beta\_TotalLockdown, Beta\_Partiallyopening, Beta\_Reopening), 4),  
 Ro\_95CI = c( NA,  
 paste("( ", round(R\_o\_Mar20\_Aug21 - 1.96\*se\_R\_o\_Mar20\_Aug21,4), ", ", round(R\_o\_Mar20\_Aug21 + 1.96\*se\_R\_o\_Mar20\_Aug21,4), " )", sep = ""),  
 paste("( ", round(R\_o\_TotalLockdown - 1.96\*se\_R\_o\_TotalLockdown,4), ", ", round(R\_o\_TotalLockdown + 1.96\*se\_R\_o\_TotalLockdown,4), " )", sep = ""),  
 paste("( ", round(R\_o\_Partiallyopening - 1.96\*se\_R\_o\_Partiallyopening,4), ", ", round(R\_o\_Partiallyopening + 1.96\*se\_R\_o\_Partiallyopening,4), " )", sep = ""),  
 paste("( ", round(R\_o\_Reopening - 1.96\*se\_R\_o\_Reopening,4), ", ", round(R\_o\_Reopening + 1.96\*se\_R\_o\_Reopening,4), " )", sep = ""))  
 )  
  
library(knitr)  
kable(counting\_process\_output)

|  |  |  |  |
| --- | --- | --- | --- |
| status | Ro | infection\_rate\_Beta | Ro\_95CI |
| Beginning | 2.0633 | 0.1474 | NA |
| Mar2020-Aug2021 | 2.0304 | 0.1450 | ( 1.9474, 2.1134 ) |
| Total Lockdown | 2.0284 | 0.1449 | ( 1.1078, 2.9491 ) |
| Partially opening | 2.0224 | 0.1445 | ( 1.7616, 2.2833 ) |
| Reopening | 2.0259 | 0.1447 | ( 1.8849, 2.1669 ) |