ANALYSIS OF KENYA COVID-19 USING SIR MODEL

Group 1

2022-07-02

## loading Relevant packages and Data Set

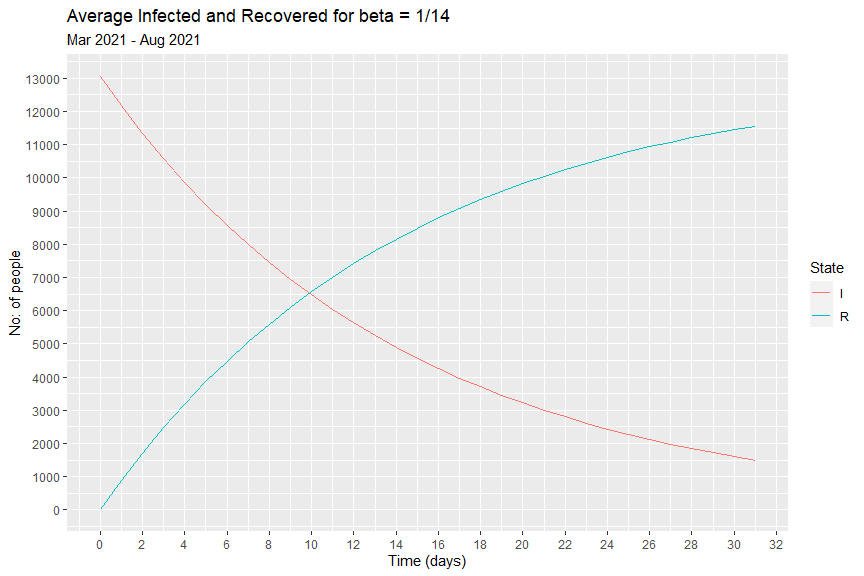
library(tidyverse)   
library(readxl)  
library(janitor)  
library(lubridate)  
## loading the csv data set  
S\_I\_R\_V2 <- read\_excel("S\_I\_R V2.xlsx", sheet = "Sheet1")%>%  
 clean\_names()%>%  
 select(-6,-7)%>%  
mutate(population = gsub(",", "", population))%>%  
mutate(population = as.numeric(population))

## Sir Models

### Building a two compartment model in R

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

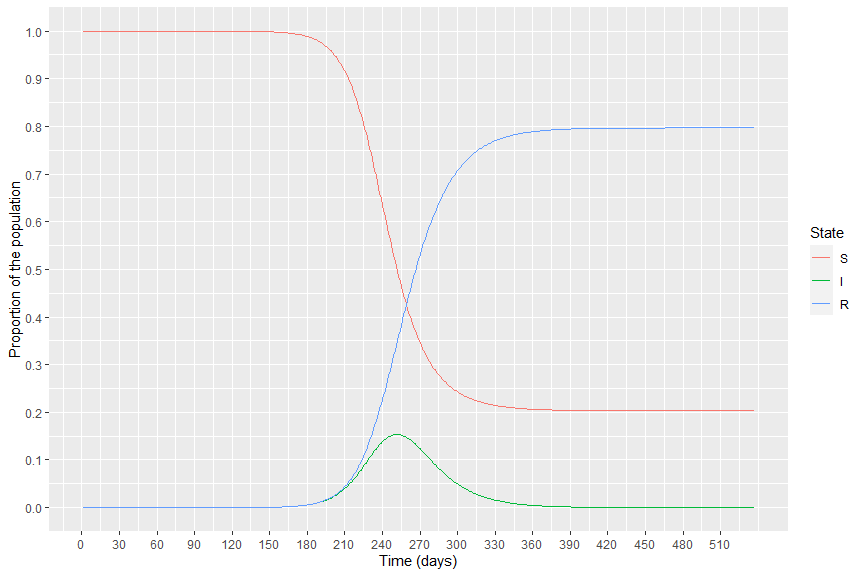
# Model inputs  
  
initial\_values1<-c(I=S\_I\_R\_V2$number\_of\_infected[S\_I\_R\_V2$x2020 == "Total"]/18, R=0)  
  
#initial\_values1<-c(I=sum(Owid\_Covid\_Kenya$cases\_new)/18, R=0)  
  
parameters1<-c(beta=round(1/14,2))  
  
# Time points  
  
time1=seq(from=0,t=31,by=1)  
  
# SIR model function  
  
sir\_model1 <-function(time1,state,parameters1){  
 with(as.list(c(state,parameters1)),{  
 dI=-beta\*I  
 dR=beta\*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 beta = 1/14",  
 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

# 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(beta=round(1/14,2),alpha=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=-alpha\*S\*(I/N)  
 dI= alpha\*S\*(I/N)-beta\*I  
 dR= beta\*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 = 1))%>%  
 mutate(I = format(I, scientific = F, digits = 1))%>%  
 mutate(R = format(R, scientific = F, digits = 1))%>%  
 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

#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 | 2.0099 | 0.1436 | ( 1.9286, 2.0913 ) |
| 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 ) |

library(readxl)  
library(zoo)  
monthly\_data <- read\_excel("monthly data.xlsx",   
 col\_types = c("text", "date", "numeric",   
 "numeric", "numeric", "numeric",   
 "text")) %>%   
 select(-c(6,7))  
monthly\_data$Period<-as.yearmon(monthly\_data$Month)

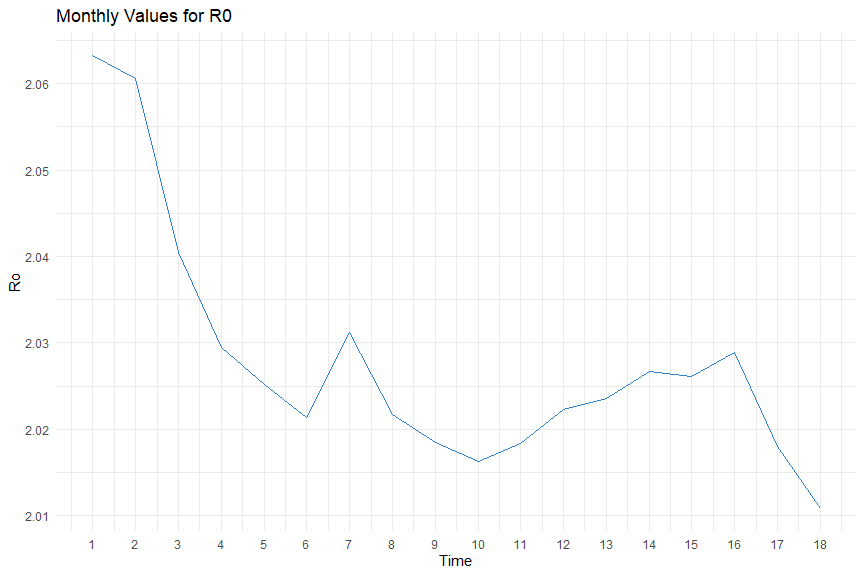
### Computing Ro for monthly values

So=(monthly\_data$Population[1]-monthly\_data$Infected[1]-monthly\_data$Recovered[1])  
n=monthly\_data$Population[19]  
Ro\_Month<-monthly\_data %>%   
 mutate(p=(1-((Population-Infected-Recovered)/Population[Status=="Total"])),  
 theta=(-log(1-p))/Recovered,  
 Ro=Population[Status=="Total"]\*theta,  
 infection\_rate= Ro\*(1/14),  
 ST=Population-Infected-Recovered,  
 UT=Ro\*(ST/So),  
 SE\_theta=((2\*(n/((n-1)^2+0.5)))-((theta^2)\*UT))^0.5/Recovered) %>%   
 slice(-19)  
Rdata<-Ro\_Month %>% select(Status,Month,Ro) %>%   
 mutate(Time=c(1:18))

### Statistical Analyses of Ro

1. Descriptive Statistics

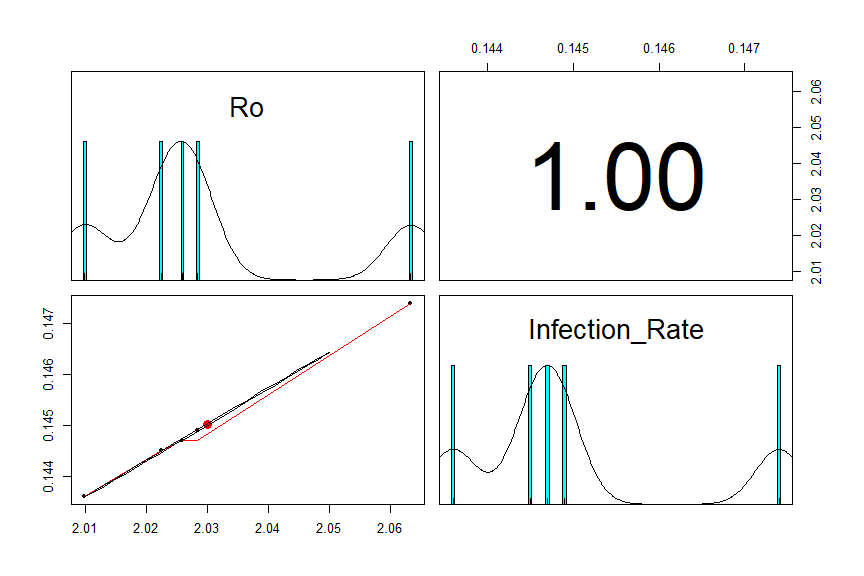
sumarydata<-monthly\_data %>%   
 slice(-19) %>%  
 select(-Period) %>%   
 mutate(Status=as.factor(Status))  
  
sad<-sumarydata %>%   
 group\_by(Status) %>%  
 summarise(Total\_Infected=sum(Infected),  
 Total\_Recovered=sum(Recovered))  
  
ggplot(Rdata) +  
 aes(x = Time, y = Ro) +  
 geom\_line(size = 0.5, colour = "#2E80D0") +  
 scale\_x\_continuous(n.breaks = 18)+  
 labs(x = "Time", y = "Ro", title = "Monthly Values for R0") +  
 theme\_minimal()



cor(counting\_process\_output$infection\_rate\_Beta,counting\_process\_output$Ro)

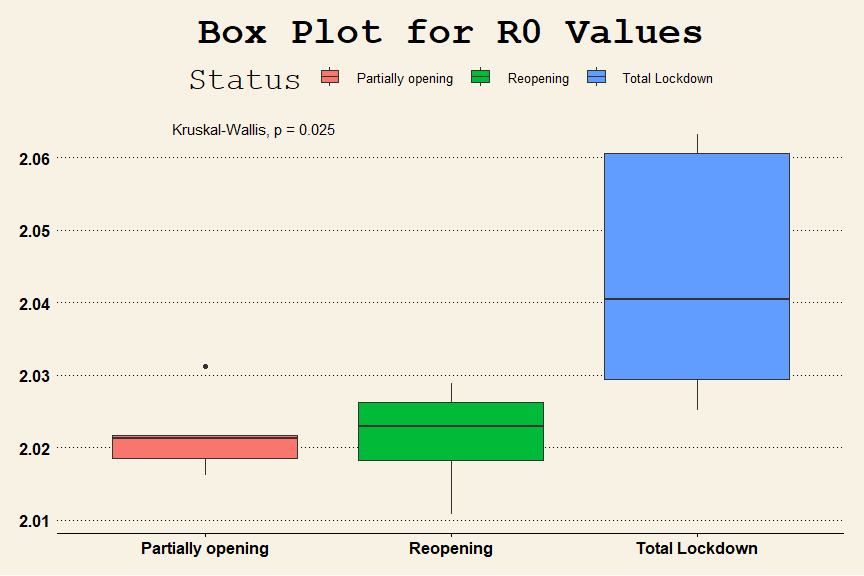
## [1] 0.9999092

library(psych)  
counting\_process\_output %>%   
 select(Ro,infection\_rate\_Beta) %>%   
 mutate(Infection\_Rate=infection\_rate\_Beta) %>%   
 select(-infection\_rate\_Beta) %>%   
 pairs.panels()



1. Comparing means of r0 QN: Were the lock downs significant in reducing the values of Ro?

Rdata<-Ro\_Month %>%   
 select(Status,Month, Ro) %>%   
 slice(-19) %>%   
 mutate(Time=c(1:18))  
write.csv(Rdata, "E:\\4.1\\Project\\SIRModel\\Rdata.csv", row.names = FALSE)  
  
ggplot(Rdata) +  
 aes(x = Status, y = Ro, fill = Status) +  
 geom\_boxplot(shape = "circle") +  
 stat\_compare\_means()+  
 scale\_fill\_hue(direction = 1) +  
 labs(  
 subtitle = ,  
 x = "Status",  
 y = "R0",  
 title = "Box Plot for R0 Values",  
 fill = "Status"  
 ) +  
 ggthemes::theme\_wsj() +  
 theme(plot.title = element\_text(face = "bold", hjust = 0.5))



kruskal.test(Ro~Status, data=Rdata)

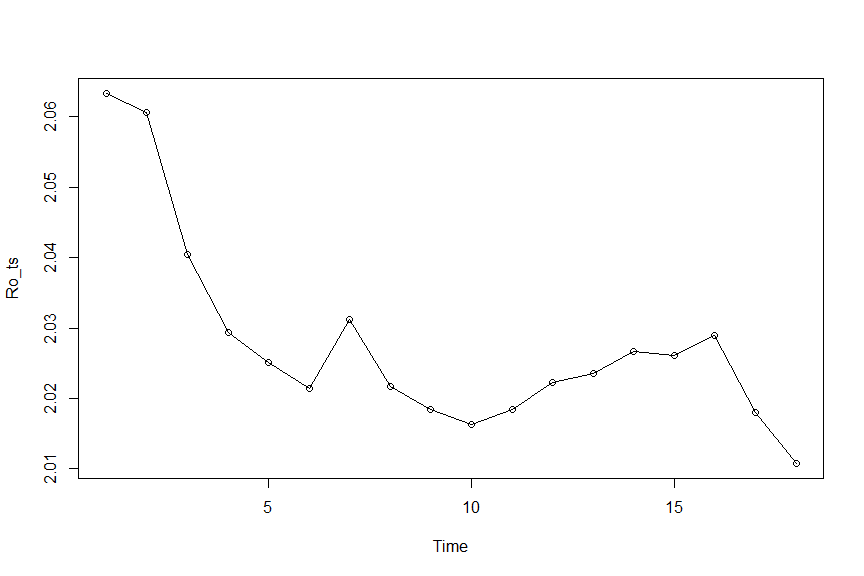
##   
## Kruskal-Wallis rank sum test  
##   
## data: Ro by Status  
## Kruskal-Wallis chi-squared = 7.3904, df = 2, p-value = 0.02484

pairwise.wilcox.test(Rdata$Ro,Rdata$Status, p.adjust.method = "BH")

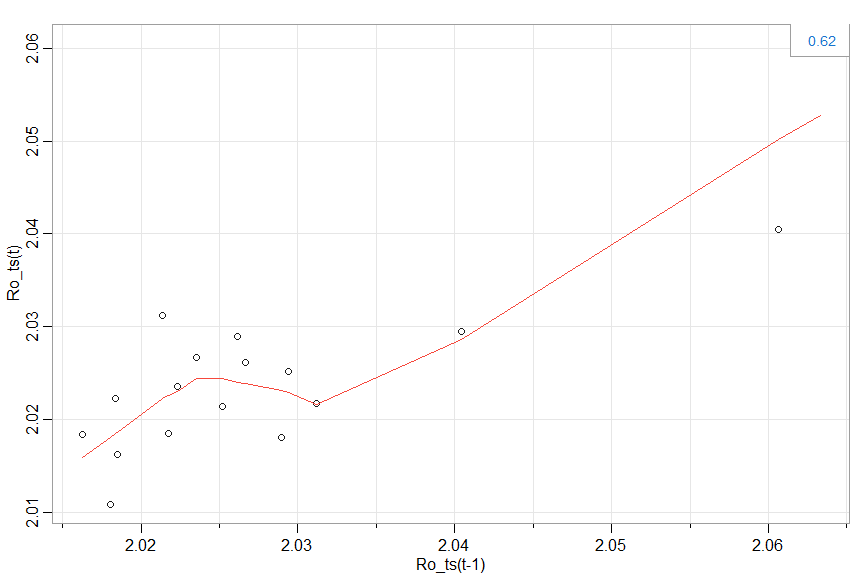
##   
## Pairwise comparisons using Wilcoxon rank sum exact test   
##   
## data: Rdata$Ro and Rdata$Status   
##   
## Partially opening Reopening  
## Reopening 0.833 -   
## Total Lockdown 0.048 0.033   
##   
## P value adjustment method: BH

1. Building a time series model

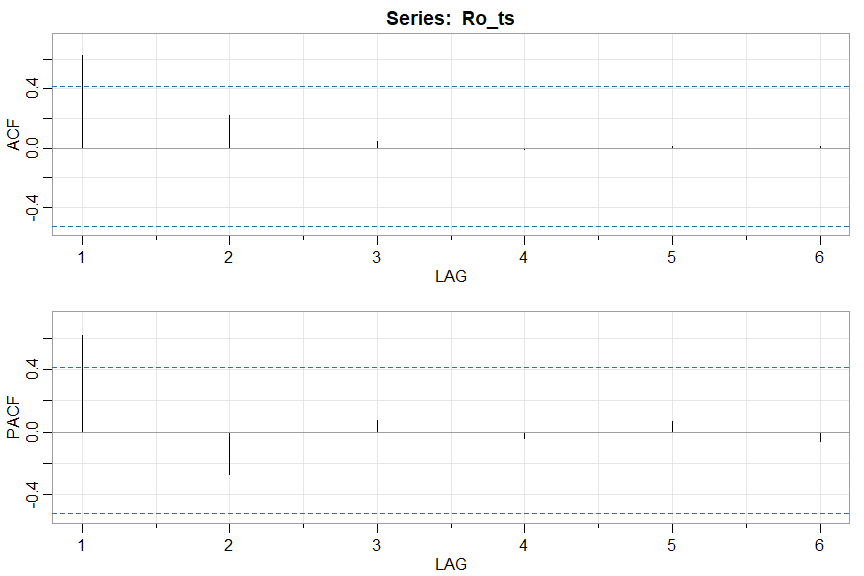
library(astsa)  
Ro\_ts<-ts(Rdata$Ro)  
plot(Ro\_ts, type="o")

 -There is a general downward trend -no visible seasonality -no constancy of variance -There are no outliers in the series

lag1.plot(Ro\_ts,1)



acf2(Ro\_ts)

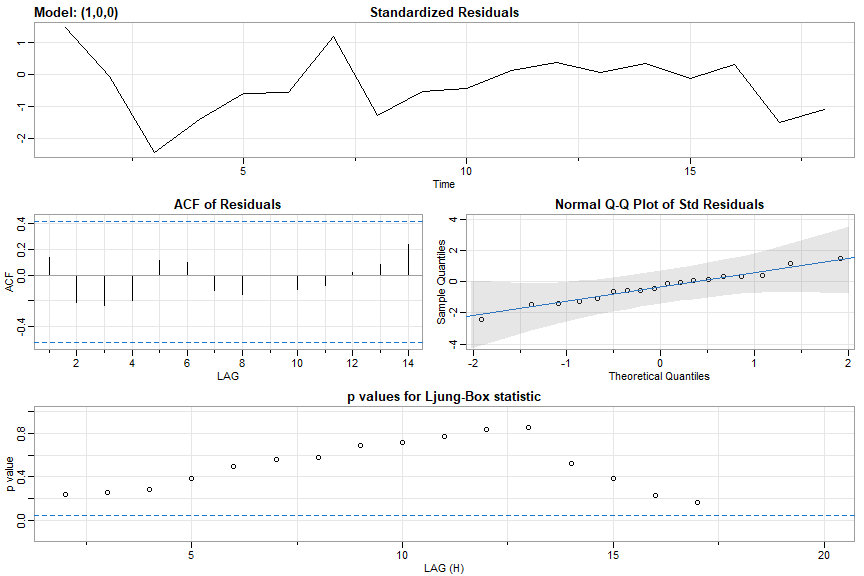


## [,1] [,2] [,3] [,4] [,5] [,6]  
## ACF 0.62 0.22 0.05 -0.01 0.01 0.01  
## PACF 0.62 -0.28 0.08 -0.04 0.07 -0.06

-The acf plot tappers towards zero and the Pacf plot has one significant spike and shuts off the rest of the lags. -This is an AR(1) model.This model predicts future values based on the past values.

sarima(Ro\_ts,1,0,0,)

## initial value -4.501493   
## iter 2 value -4.891232  
## iter 3 value -5.107647  
## iter 4 value -5.126948  
## iter 5 value -5.153667  
## iter 6 value -5.172968  
## iter 7 value -5.172999  
## iter 8 value -5.176362  
## iter 9 value -5.177774  
## iter 10 value -5.177876  
## iter 11 value -5.177877  
## iter 11 value -5.177877  
## final value -5.177877   
## converged  
## initial value -4.658199   
## iter 2 value -4.816887  
## iter 3 value -4.827048  
## iter 4 value -4.831663  
## iter 5 value -4.832773  
## iter 6 value -4.832938  
## iter 7 value -4.832963  
## iter 8 value -4.833002  
## iter 9 value -4.833002  
## iter 10 value -4.833004  
## iter 10 value -4.833004  
## iter 10 value -4.833004  
## final value -4.833004   
## converged



## $fit  
##   
## Call:  
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),   
## xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,   
## optim.control = list(trace = trc, REPORT = 1, reltol = tol))  
##   
## Coefficients:  
## ar1 xmean  
## 0.9294 2.0333  
## s.e. 0.0862 0.0163  
##   
## sigma^2 estimated as 5.676e-05: log likelihood = 61.45, aic = -116.91  
##   
## $degrees\_of\_freedom  
## [1] 16  
##   
## $ttable  
## Estimate SE t.value p.value  
## ar1 0.9294 0.0862 10.7778 0  
## xmean 2.0333 0.0163 124.8727 0  
##   
## $AIC  
## [1] -6.494798  
##   
## $AICc  
## [1] -6.450354  
##   
## $BIC  
## [1] -6.346403

-The standard residuals plot shows no trend which is a good results - The acf plot for their residuals have all the values insignificant indicating that its a good model for prediction. -The normal qq plot shows normality because all the values lies within the line of fit - All ljung box statistics are insignificant.That’s a good result - the values of BIC,AIC and variance are very low. thats a good results.

Coefficients: ar1 xmean 0.9294 2.0333 s.e. 0.0862 0.0163

c=xmean(1-ar1(coefficient)) = 2.0333(1-0.9294) = 0.1436

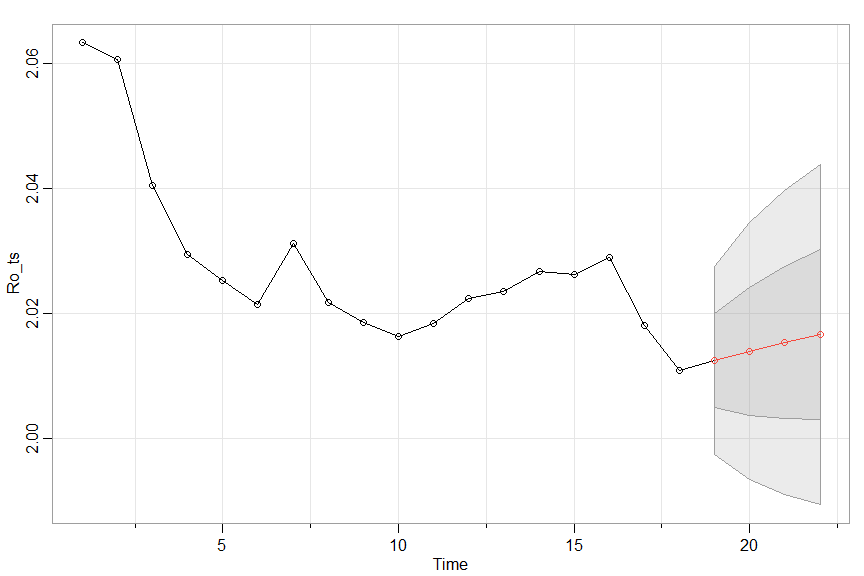
Model

y= 0.1436 + 0.9294Xt-1 + Wt

Estimate SE t.value p.value ar1 0.9294 0.0862 10.7778 0 xmean 2.0333 0.0163 124.8727 0

The p value for the ar1 coefficient is zero showing that the coefficient is significant in the model.

sarima.for(Ro\_ts,4,1,0,0)



## $pred  
## Time Series:  
## Start = 19   
## End = 22   
## Frequency = 1   
## [1] 2.012399 2.013878 2.015253 2.016531  
##   
## $se  
## Time Series:  
## Start = 19   
## End = 22   
## Frequency = 1   
## [1] 0.007533707 0.010284895 0.012170496 0.013590174