The Rolling R0 Package

Team Ninja
Department of Applied Mathematics & Statistics
State University of New York at Stony Brook

```r
# import packages
if (!require(tidyverse)) install.packages('tidyverse')
```

```
## Loading required package: tidyverse
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
if (!require(httr)) install.packages('httr')
```

```
## Loading required package: httr
```

```r
if (!require(jsonlite)) install.packages('jsonlite')
```

```
## Loading required package: jsonlite
```

```
##
## Attaching package: 'jsonlite'
```

```
## The following object is masked from 'package:purrr':
##
##     flatten
```

```r
if (!require(fitdistrplus)) install.packages('fitdistrplus')
```

```
## Loading required package: fitdistrplus
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
## Loading required package: survival
```

```r
if (!require(R0)) install.packages('R0')
```

```
## Loading required package: R0
```

```r
if (!require(ggplot2)) install.packages('ggplot2')
if (!require(knitr)) install.packages('knitr')
```

```
## Loading required package: knitr
```

```r
library(tidyverse)
library(httr)
library(jsonlite)
library(fitdistrplus)
library(R0)
library(ggplot2)
library(knitr)
```

```r
# preparation of getting data from https://c3.ai/customers/covid-19-data-lake/
source("c3aidatalake.R")
```

Introduction

In pandemic, the basic reproduction number, denoted by R_0, 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. @Fraser2009 In commonly used infection models, when $R_0 > 1$ the infection will be able to start spreading in a population, but not if $R_0 < 1$. @Fine2011

Methods and Data

We used the cumulative total confirmed cases data from COVID tracking project (https://covidtracking.com) to fit a time varying reproduction number of each states of united states and compare the R_0 before and after some critical date like stay at home order, mask mandate and reopen date and the population data from United States Census to calculate a population weighted R_0 for states grouped by politic leaning and safety policies. We got the confimder cases data and population data from c3.ai data lake (https://c3.ai/customers/covid-19-data-lake/). We got the data of politic leaning of each states by the election results of 2016 elections from New York Times (https://www.nytimes.com/elections/2016/results/president) and the data of mask mandate by each states from AARP (https://www.aarp.org/health/healthy-living/info-2020/states-mask-mandates-coronavirus.html) stay at home order from UStoday (https://www.usatoday.com/storytelling/coronavirus-reopening-america-map/#caseload).

We used Maximum Likelihood method proposed by @White2009 to estimate R_0 for COVID-19 in each states with one month as window size. We used estimated parameters of the serial interval distribution from @Du2020 (mean is 3.96 days and sd is 4.75 days) and the distribution most used to describe serial interval distribution gamma distribution as the setting of generation time.

We also genrated *Rolling* R_0 by rolling window method with one month as window size, plot the *Rolling* R_0 for each states as well as the important policies dates to explore the effect of different policies.

Results

Comparison of R_0 before and after mask mandate

There are 33 states posting mask mandate totally, we use the daily positive increase cases data in one month before mask mandate to estimate the R_0 before mask mandate and the data in one month after mask mandate to estimate the R_0 after mask mandate to estimate R_0 after mask mandate.

```r
mask_date <- read.csv('mask_date.csv') #read_data
stateslist <- c(state.abb , 'DC') #add DC in the states list

result <- NULL
```

```r
for(stateab in stateslist){
  if(mask_date$date[mask_date$geo_value == stateab] != ""){
    #generate statename
    if(stateab == 'DC'){
      statename = "DistrictofColumbia"
    }else{
        statename <- gsub(" ", "", state.name[state.abb == stateab], fixed = TRUE)}
    statename <- paste(statename, 'UnitedStates', sep = '_')

    #get data
    data <- evalmetrics("outbreaklocation",
                        list(
                          spec = list(
                            ids = list(statename),
                            expressions = list("CovidTrackingProject_ConfirmedCases"),
                            interval = "DAY",
                            start = "2020-03-01",
                            end  = "2020-11-13"
                            )
                          )
                        )
    data <- data[, c('dates','data')]

    #change format of dates and data
    #(cumulative to incident and positive case should be postive integer)
    data$dates <- as.Date(data$dates)
    data$data <- data$data - c(0, data$data[1:(length(data$data)-1)])
    data$data[data$data <= 1] <- 1
    data$data <- as.integer(data$data)

    #generate epic data and mask mandate effective date
    #(Here we have a 15 days delay due to the incubation period and test delay)
    epic <- data$data
    effectdate <- as.Date(mask_date$date[mask_date$geo_value == stateab],
                          format = '%m/%d/%y') + 15
    names(epic) <- as.Date(data$dates)

    #generate generation time based on serial interval distribution
    mGT <- generation.time("gamma", c(3.96,4.75))

    #calculate the R0 before and after mask mandate effective date
    R0_before <- estimate.R(epid = epic, GT = mGT, begin = effectdate - 30,
                            end = effectdate, methods=c("ML"))$estimates$ML$R
    R0_after <- estimate.R(epid = epic, GT = mGT, begin = effectdate+1,
                           end = effectdate + 30, methods=c("ML"))$estimates$ML$R
    tmp <- data.frame(state = stateab, R0_before = R0_before, R0_after = R0_after)
    result <- rbind(result,tmp)
  }
}
```

```r
kable(result, caption = 'Comparisoon of R0 before/after mask mandate')
```

Table 1: Comparisoon of R0 before/after mask mandate

state	R0_before	R0_after
AL	1.1847780	1.160939
AR	1.2058922	1.155672
CA	1.2773798	1.146408
CO	1.2004057	1.106373
CT	1.1337086	1.090830
DE	1.1800764	1.064778
HI	0.9325994	1.058251
IL	1.1890914	1.054268
IN	1.1856516	1.136980
KY	1.3027521	1.178410
LA	1.1760452	1.070191
ME	1.2242999	1.116612
MD	1.2387726	1.136089
MA	1.0735228	1.039419
MI	1.0873704	1.064340
MN	1.1742611	1.158197
MT	1.2430955	1.185647
NV	1.2811453	1.133773
NJ	1.1621245	1.037794
NM	1.0962056	1.204864
NY	1.0826756	1.060544
NC	1.2075433	1.111457
OH	1.1330479	1.159721
OR	1.2141624	1.155951
PA	1.1363983	1.104927
RI	1.1579152	1.058961
TX	1.2600038	1.146206
VT	1.1797399	1.136310
VA	1.0912397	1.225795
WA	1.2179830	1.140000
WV	1.1827857	1.063080
WI	1.1332040	1.259802
DC	1.3299504	1.140629

From the table, there are only 5 out of 33 states, which have mask mandate, R_0 increased after mask mandate the others all decreased. With some materials reading, we found the reason for the increasing of R_0 in this 5 states can be as follow.

1. Some states post mask mandate very earlier, and positive cases is very few, R_0 doesn't work well in this situation, like Hawaii, the month after mask mandate, the new cases are only one or two each day, in this case small number of new cases will affect R_0 very much.

2. Conflict policies, after the government post mask mandate, they also expire the stay at home order, like New Mexico, the mask mandate and stay at home order expire date are both at May 15th, and Virginia, post the mask mandate on May 29, but stay at home order expired a a few days after mask mandate on June 10th.

3. Mass gathering, during one month after the mask mandate make effect, there are some mass gathering due to different reasons, like the protests in June, Virginia and in August, Wisconsin.

```r
kable(result[result$R0_before < result$R0_after,],
      caption = 'States R0 before is less than after mask mandate')
```

Table 2: States R0 before is less than after mask mandate

	state	R0_before	R0_after
7	HI	0.9325994	1.058251
20	NM	1.0962056	1.204864
23	OH	1.1330479	1.159721
29	VA	1.0912397	1.225795
32	WI	1.1332040	1.259802

```r
# wilcoxon test for R0 before and after
shapiro.test(result$R0_before - result$R0_after)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  result$R0_before - result$R0_after
## W = 0.88464, p-value = 0.002179
```

```r
wilcox.test(result$R0_before, result$R0_after, paired = T, alternative = 'greater')
```

```
##
##  Wilcoxon signed rank exact test
##
## data:  result$R0_before and result$R0_after
## V = 450, p-value = 0.0009089
## alternative hypothesis: true location shift is greater than 0
```

With the wilcoxon signed rank test, the R_0 before mask mandate is significantly higher than after mandate (with p-value 0.001365).

Comparison R_0 curve of red states versus blue states

We separate the states into three groups, red (Republican), blue (Democratic) and Swing. If in 2016 president election, the proportion vote for Republication is 10 percent higher than Democratic, then we thought the state is red, and similar rule for blue states. The other states are considered as Swing.

```r
# Reading election data
politics <- read.csv('2016_election_result.csv')
politics <- politics[,c('State_ab','Hillary.Clinton.Democratic.percentage',
                        'Donald.Trump.Republican.percentage')]
colnames(politics) <- c('state','blue','red')
politics$red <- as.numeric(sub("%","",politics$red))/100
politics$blue <- as.numeric(sub("%","",politics$blue))/100
politics$group = "Swing"

# separate states with red blue and swing states with the election result
politics$group[(politics$red - politics$blue) > 0.1] = "Red"
politics$group[(politics$blue - politics$red) > 0.1] = "Blue"
politics <- politics[,c('state', 'group')]
politics <- politics[politics$state %in% c(state.abb,'DC'),]
redvsblue <- data.frame(red = politics$state[politics$group == 'Red'],
                        blue = c(politics$state[politics$group == 'Blue'],
```

```
                                  rep("", sum(politics$group == 'Red') -
                                        sum(politics$group == 'Blue'))),
                         swing = c(politics$state[politics$group == 'Swing'],
                                   rep("", sum(politics$group == 'Red') -
                                         sum(politics$group == 'Swing'))))
kable(redvsblue, caption = 'States group')
```

Table 3: States group

red	blue	swing
AL	CA	AZ
AK	CT	CO
AR	DE	FL
ID	DC	GA
IN	HI	IA
KS	IL	ME
KY	MD	MI
LA	MA	MN
MS	NJ	NV
MO	NY	NH
MT	OR	NM
NE	RI	NC
ND	VT	OH
OK	WA	PA
SC		TX
SD		VA
TN		WI
UT		
WV		
WY		

```
#Separate the states in mask group and no mask group as their mask mandate
maskgroup <- c()
nomaskgroup <- c()
for(stateab in stateslist){
  if(!mask_date$date[mask_date$geo_value == stateab] == ""){
    maskgroup <- c(maskgroup, stateab)
  }else{
    nomaskgroup <- c(nomaskgroup,stateab)
  }
}

# add space value for print
maskvsnomask <- data.frame(mask = maskgroup,
                           nomask = c(nomaskgroup,
                                      rep("", length(maskgroup) - length(nomaskgroup))))
kable(maskvsnomask, caption = 'mask mandate states vs no mask mandate states')
```

Table 4: mask mandate states vs no mask mandate states

mask	nomask
AL	AK

mask	nomask
AR	AZ
CA	FL
CO	GA
CT	ID
DE	IA
HI	KS
IL	MS
IN	MO
KY	NE
LA	NH
ME	ND
MD	OK
MA	SC
MI	SD
MN	TN
MT	UT
NV	WY
NJ	
NM	
NY	
NC	
OH	
OR	
PA	
RI	
TX	
VT	
VA	
WA	
WV	
WI	
DC	

With these two tables, there are 14 out of 14 blue states had mask mandate, 11 out of 17 swing states had mask mandate, but only 8 out of 20 red states had mask mandate.

We used daily rolling window with window size is 30 days, to compare the population weighted R_0 of Red, Blue and Swing states.

```
#Calculate rolling R0 for each states
R0_series <- NULL
startdate <- as.Date('2020-03-01')
i = 1:(as.Date('2020-11-13') - as.Date('2020-03-01') - 30)
date <- startdate + 29 + i
R0_series <- data.frame(date = date)

#calculate each state R0 series
for(stateab in stateslist){
  if(stateab == 'DC'){
    statename = "DistrictofColumbia"
    }else{
    statename <- gsub(" ", "", state.name[state.abb == stateab], fixed = TRUE)
  }
```

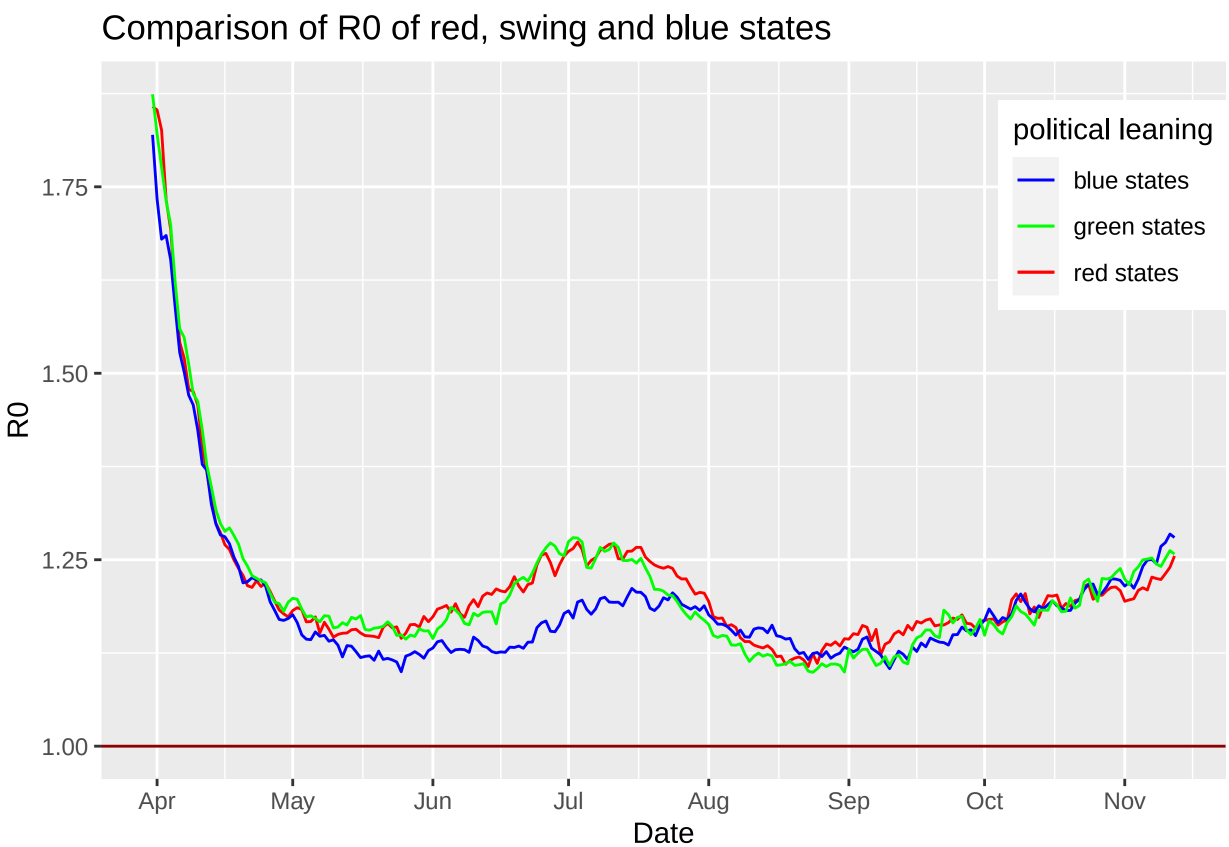
```
  statename <- paste(statename, 'UnitedStates', sep = '_')

  #get data
  data <- evalmetrics("outbreaklocation",
                  list(
                    spec = list(
                      ids = list(statename),
                      expressions = list("CovidTrackingProject_ConfirmedCases"),
                      interval = "DAY",
                      start = "2020-03-01",
                      end  = "2020-11-13"
                      )
                    )
                  )
  data <- data[, c('dates','data')]
  #change format of dates and data
  #(cumulative to incident and positive case should be positive integer)
  data$dates <- as.Date(data$dates)
  data$data <- data$data - c(0, data$data[1:(length(data$data)-1)])
  data$data[data$data <= 1] <- 1
  data$data <- as.integer(data$data)

  #generate epic data
  epic <- data$data
  names(epic) <- as.Date(data$dates)

  #generate generation time based on serial interval distribution
  mGT <- generation.time("gamma", c(3.96,4.75))

  #calculating the R0 for each states by daily rolling window with size 30
  R0 <- c()
  for(i in 1:length(R0_series$date)){
    tmp <- estimate.R(epid = epic, GT = mGT, begin = startdate + i-1,
                    end = startdate + i + 29, methods=c("ML"))$estimates$ML$R
    R0 <- c(R0,tmp)
  }
  R0_series <- transform(R0_series, tmp = R0 )
  colnames(R0_series)[ncol(R0_series)] <- stateab
}
```

```
# get population data
tmppop <- fetch(
  "populationdata",
  list(
    spec = list(
      filter = "contains(parent, 'UnitedStates') &&
      (populationAge == 'Total') &&
      gender == 'Male/Female' &&
      year == 2019 &&
      estimate == 'True'"
      )
    ),
  get_all = TRUE
)
```

```r
# manipulate population data
population <- NULL
for(stateab in stateslist){
  if(stateab == 'DC'){
    statename = "DistrictofColumbia"
  }else{
    statename <- gsub(" ", "", state.name[state.abb == stateab], fixed = TRUE)
  }
  statename <- paste('.*',statename, '_UnitedStates', '.*', sep = '')
  tmp1 <- sum(tmppop$value[grepl(statename, tmppop$id)])
  tmp <- data.frame(state = stateab, population = tmp1)
  population <- rbind(population, tmp)
}
```

```r
# calculate population sum for different political leanings
redp <- 0
bluep <- 0
swingp <- 0
for(stateab in stateslist){
  if(politics$group[politics$state == stateab] == 'Red'){
    redp = redp + population$population[population$state == stateab]
  }
  if(politics$group[politics$state == stateab] == 'Blue'){
    bluep = bluep + population$population[population$state == stateab]
  }
  if(politics$group[politics$state == stateab] == 'Swing'){
   swingp = swingp + population$population[population$state == stateab]
  }
}

# population weighted rolling R0 for red blue and swing states
R0_red <- rep(0,as.Date('2020-11-13') - as.Date('2020-03-01') - 30)
R0_blue <- rep(0,as.Date('2020-11-13') - as.Date('2020-03-01') - 30)
R0_swing <- rep(0,as.Date('2020-11-13') - as.Date('2020-03-01') - 30)
for(stateab in colnames(R0_series)[2:ncol(R0_series)]){
  if(politics$group[politics$state == stateab] == 'Red'){
    R0_red <- R0_red +
      R0_series[,stateab]*population$population[population$state == stateab]/redp
  }else if(politics$group[politics$state == stateab] == 'Blue'){
    R0_blue <- R0_blue +
      R0_series[,stateab]*population$population[population$state == stateab]/bluep
  }else if(politics$group[politics$state == stateab] == 'Swing'){
    R0_swing <- R0_swing +
      R0_series[,stateab]*population$population[population$state == stateab]/swingp
  }
}

# plot for population weighted R0
p = ggplot() +
  geom_line(aes(x = R0_series$date, y = R0_red, color = "red states"))+
  geom_line(aes(x = R0_series$date, y = R0_blue, color = "blue states"))+
  geom_line(aes(x = R0_series$date, y = R0_swing, color = "green states"))+
  ggtitle('Comparison of R0 of red, swing and blue states')+
```

```r
  labs(x = 'Date', y = 'R0')+
  scale_x_continuous(breaks=seq(R0_series$date[1],
                                R0_series$date[length(R0_series$date)], "month")
                    +c(1,0,1,0,1,1,0,1), labels = month.abb[4:11])+
  geom_hline(yintercept = 1, color = "darkred")+
  scale_colour_manual(name = "political leaning",
                      values = c("red states" = 'red',
                                 "blue states" = 'blue',
                                 "green states" = 'green'),
                      guide = "legend") +
  theme(legend.position = c(0.9,0.8))
print(p)
```



From the picture, blue states has lower R_0 From May to August, since most blue states poste mask mandate at that time however after the a few weeks as well as the new semester started, the pandemic are more severe both in all states again.

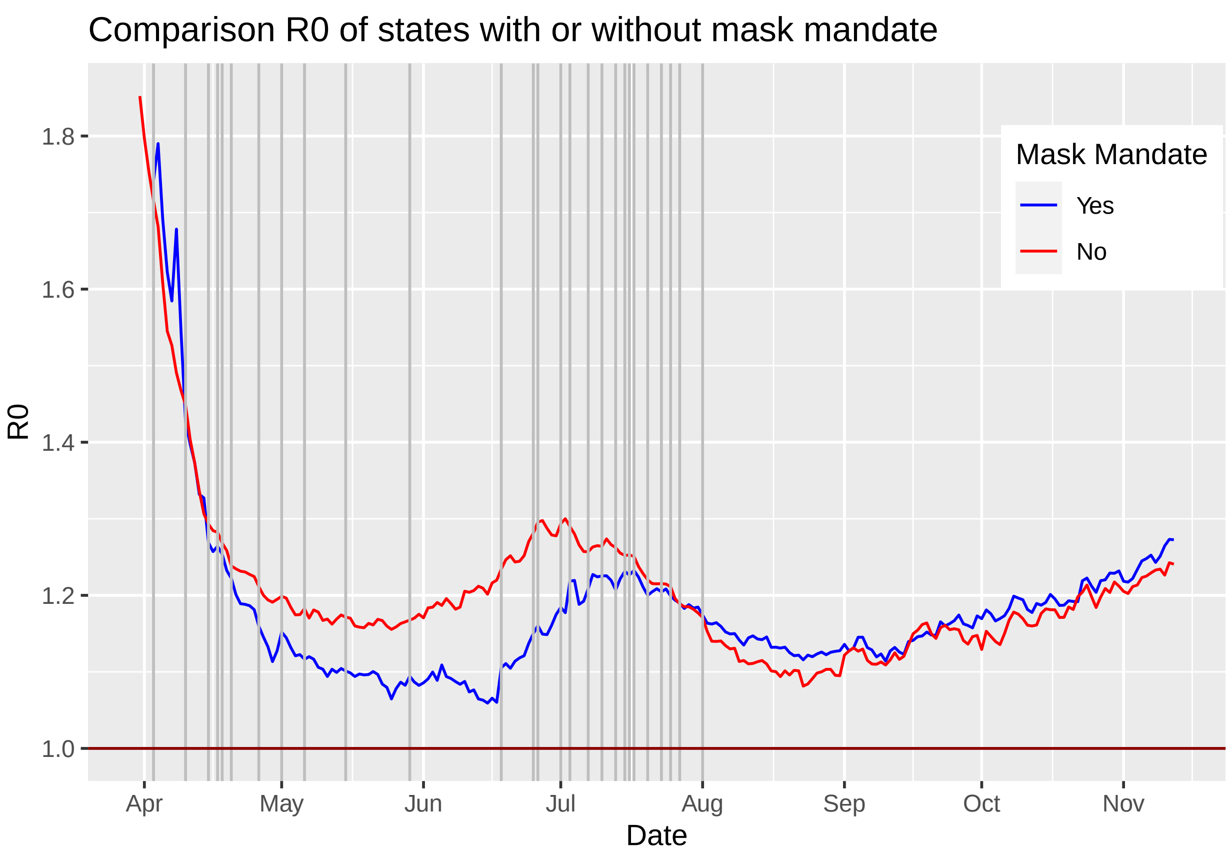
Comparison of states have or do not have mask mandate

We used daily rolling window with window size of 30 days, to compare the population weighted R_0 of states have mask mandate and do not have mask mandate. At first, all the states are in no mask group, then as the the time forward, after the state post mask mandate, then it will belong to mask group.

```r
# sort mask mandate state by their mandate date
mask_state <- mask_date[!mask_date$date == "",]
mask_state$date <- as.Date(mask_state$date, format = '%m/%d/%y')
mask_state <- mask_state[order(mask_state$date),]
```

```r
# at first all of the states didn't post mask mandate
nomaskgroup <- stateslist
maskgroup <- c()

# Calculate the population weighted rolling R0 for the two groups
j = 1
R0m <- c()
R0nm <- c()
for(i in R0_series$date){
  # while the end of the window came to the mask mandate time, move the states
  #from no mask group to mask group
  while((i == mask_state$date[j]) & j <= nrow(mask_state)){
    nomaskgroup <- nomaskgroup[!(nomaskgroup == mask_state$geo_value[j])]
    maskgroup <- c(maskgroup, mask_state$geo_value[j])
    j = j + 1
  }
  tmpnm <- 0
  tmpm <- 0
  tmpnmp <- sum(population$population[population$state %in% nomaskgroup])
  tmpmp <- sum(population$population[population$state %in% maskgroup])
  for(stateab in stateslist){
    if(stateab %in% nomaskgroup){
      tmpnm <- tmpnm + R0_series[R0_series$date == i, stateab]*
        population$population[population$state == stateab]/tmpnmp
    }else{
      tmpm <- tmpm + R0_series[R0_series$date == i, stateab]*
        population$population[population$state == stateab]/tmpmp
    }
  }
  R0nm <- c(R0nm, tmpnm)
  if(tmpmp != 0) R0m <- c(R0m, tmpm) else R0m <- c(R0m, NA)
}
maskvsnomaskR0 <- data.frame(date = rep(R0_series$date,2), R0 = c(R0nm,R0m),
                             mask = rep(c('no mask','mask'),
                                        each = length(R0_series$date)))

# plot rolling R0 curve for these two groups, with grey line is the date of mask
# mandate of each state
p = ggplot(data = maskvsnomaskR0,  aes(x = date, y = R0),) +
  geom_line(aes(color = factor(mask)))+
  geom_vline(xintercept = mask_state$date, color = 'grey')+
  scale_x_continuous(breaks=seq(R0_series$date[1],
                                R0_series$date[length(R0_series$date)], "month")+
                       c(1,0,1,0,1,1,0,1),
                     labels = month.abb[4:11])+
  geom_hline(yintercept = 1, color = "darkred")+
  scale_color_manual(name = "Mask Mandate",
                     values = c('no mask' = 'red', 'mask' = 'blue'),
                     labels = c('no mask' = 'No', 'mask' = 'Yes')) +
  theme(legend.position = c(0.9,0.8))+
  labs(title = "Comparison R0 of states with or without mask mandate", x = 'Date')
print(p)
```

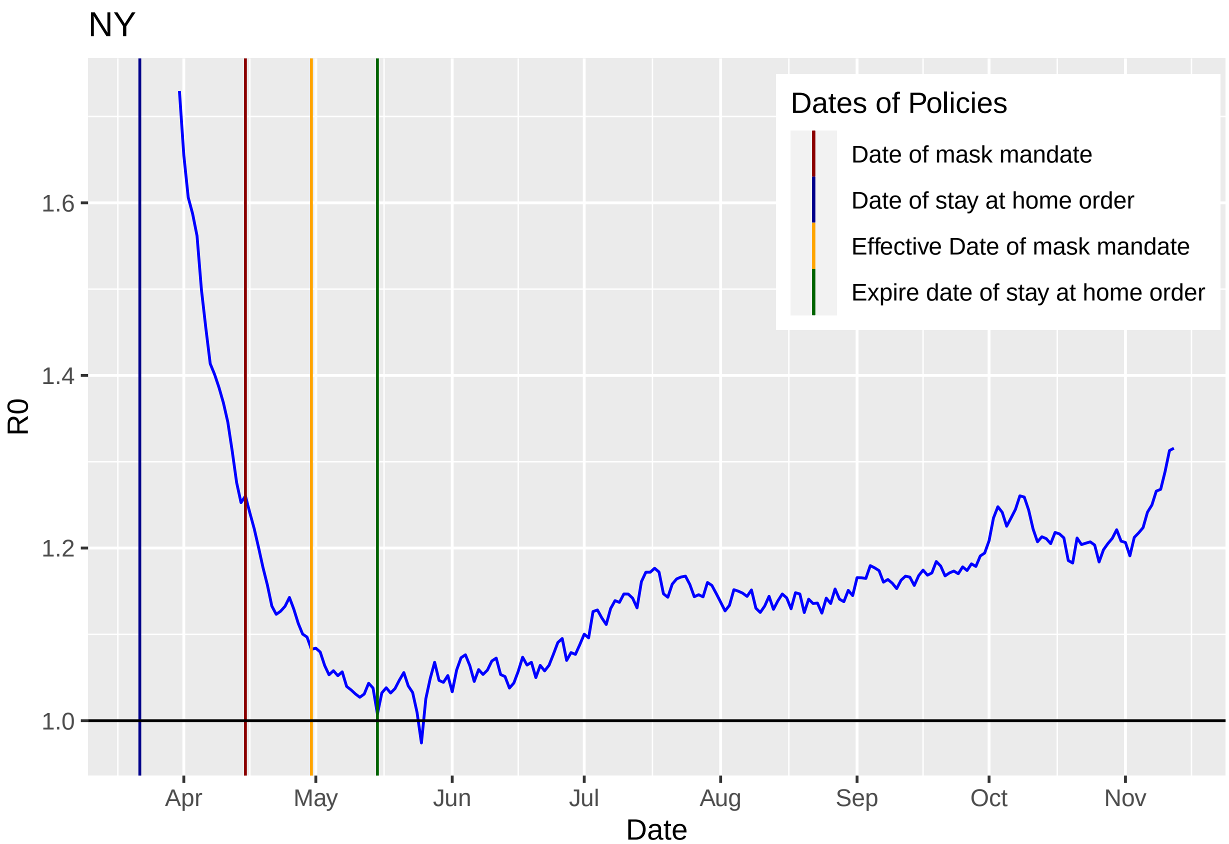
```
## Warning: Removed 3 row(s) containing missing values (geom_path).
```



Plot of each states vs policy days

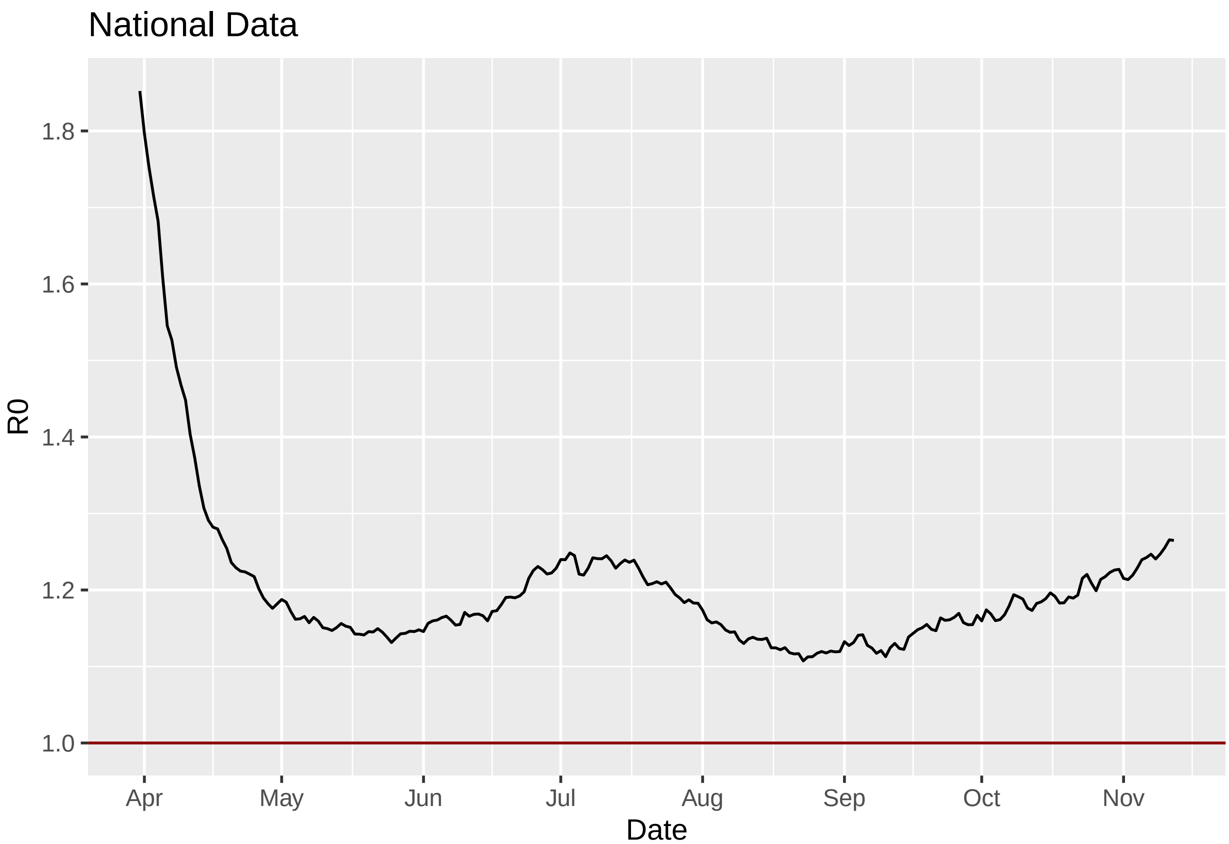
Here we plot the R_0 trend of some states and the date they post some policies like stay at home order, expiration of stay at home order and mask mandate.

```
# Reading data of other policies like stay at home order
stay_date <- read.csv('policy_date.csv', header = T)

# Here we use New York as an example
stateab <- 'NY'

# decide lines color with their political leanings
if(politics$group[politics$state == stateab] == 'Red') color = 'red'
if(politics$group[politics$state == stateab] == 'Blue') color = 'blue'
if(politics$group[politics$state == stateab] == 'Swing') color = 'green'
p <- ggplot() +
  geom_line(aes(x = R0_series$date, y = R0_series[,stateab]), color = color)+
  labs(x = 'Date', y = 'R0', title =  stateab)+
  geom_vline(aes(xintercept = as.Date(mask_date$date[mask_date$geo_value == stateab], format ='%m/%d/%y
   geom_vline(aes(xintercept = as.Date(stay_date$stay_at_home_date[stay_date$geo_value == stateab], for
    geom_vline(aes(xintercept = as.Date(stay_date$stay_at_home_expire_date[stay_date$geo_value == statea
  geom_vline(aes(xintercept = as.Date(mask_date$date[mask_date$geo_value == stateab], format = '%m/%d/%
  scale_x_continuous(breaks=seq(R0_series$date[1], R0_series$date[length(R0_series$date)], "month")+c(1
  geom_hline(yintercept = 1, color = "black") +
  scale_color_manual(name = 'Dates of Policies', values = c('Date of mask mandate' = 'darkred', 'Effect
  theme(legend.position = c(0.8,0.8))
print(p)
```



plot of national data

```r
#generate national rolling R0
national_R0 <- data.frame(date = R0_series$date)
national_pop <- sum(population$population)
tmp <- 0
for(stateab in stateslist){
  tmp <- tmp +
    R0_series[, stateab] *
    population$population[population$state == stateab] / national_pop
}
national_R0$R0 <- tmp
p <- ggplot() +
  geom_line(aes(x = national_R0$date, y = national_R0$R0))+
  labs(x = 'Date', y = 'R0', title =  'National Data')+
  scale_x_continuous(breaks= seq(R0_series$date[1],
                              R0_series$date[length(R0_series$date)], "month")+
                       c(1,0,1,0,1,1,0,1),
                     labels = month.abb[4:11])+
  geom_hline(yintercept = 1, color = "darkred")
print(p)
```


National Data
R0
1.8
1.6
1.4
1.2
1.0
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Date