LogEpi: Logistic curves applied to Covid-19

Beatriz Cuyabano and Gabriel Rovere

May 8, 2020

This vignette offers a brief explanation on how the logistic curve is used to model epidemiological events, and provides a set of examples of analysis using the `R` package `LogEpi`[1].

A logistic curve can be used to fit a counting process that starts at zero and ends at a determined number. Such curve has been widely used to describe the growth of a population, the growth of cases in a disease outbreak, or the growth of deaths in a such outbreak. The general logistic function is described as:

$$y = f(x \mid a, b, c) = \frac{a}{1 + \exp\left[(b - x)/c\right]}. \tag{1}$$

The value of y represents the cumulative counts of the event to be described, x represents the days passed after the first occurrence of this event, and a, b, and c are the parameters of the event. To simplify the description, $x = 1$ at the day of the first occurrence.

- a is the total number of occurrences at the stabilization of the event,
- b is the day in which we will observe the maximum new occurrences of the event,
- c is the speed of the process.

The parameters are obtained by minimizing the mean square error (MSE) of the model:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \frac{a}{1 + \exp\left[(b - x_i)/c\right]}\right)^2, \tag{2}$$

$$\hat{a}, \hat{b}, \hat{c} = \textit{argmin}\, \text{MSE}, \tag{3}$$

constraining their values to $a \geq max(y_i)$, $b \geq 1$, and $c > 0$.

Note that this curve does not take into account any demographic or social information. The adjustment is based on the observed numbers provided. If the outbreak is still in its initial stage, projections will be necessary to model the entire outbreak. This will be detailed with examples

[1] `LogEpi` is in its initial versions. Please help us improve our work in this package by reporting any bugs, errors, or suggestions to bia.cdc@gmail.com.

throughout this document. The latest Covid-19 data used for the examples was downloaded directly into R using the LogEpi package, and is provided by Johns Hopkins University (https://systems.jhu.edu/research/public-health/ncov/). The datasets are available at the repository https://github.com/CSSEGISandData/COVID-19. This data contains the historical numbers of confirmed cases, deaths, and recovery, both globally and detailed for all the states of the United States. The examples are based on the data downloaded May 8, 2020.

We will start by loading the package and defining the folder and filenames template to download the Covid-19 data.

```
# load LogEpi package
library(LogEpi)
# define folder and filenames template for the Covid-19 data
tmp.folder <- "Covid19_folder"
tmp.filename <- "Covid19_JHdata"
```

Function `load_JH_db()` dowloads the datasets into the folder defined in object `tmp.folder`, using the object `tmp.filename` as the filenames template.

```
# download the data from the JHU repository
# https://github.com/CSSEGISandData/COVID-19
loc <- load_JH_db(folder=tmp.folder,filename=tmp.filename)
```

As of today (May 8, 2020), the recovery data is not available specifically for the US states, only for the country as a whole, therefore the warning message below is expected.

```
# Warning messages:
# 1: In download.file(info.url,paste0(folder,"/",filename,"_",data.type, : cannot open
#    URL 'https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_
#    19_data/csse_covid_19_time_series/time_series_covid19_recovered_US.csv': HTTP sta
#    tus was '404 Not Found'
# 2: In download.file(info.url,paste0(folder,"/",filename,"_",data.type, : cannot open
#    URL 'https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_
#    19_data/csse_covid_19_time_series/time_series_covid19_recovered_US.csv': HTTP sta
#    tus was '404 Not Found'
```

The output of `load_JH_db()` is a list with two objects: `countries` and `US_states`. These are the lists of places for which the data is available.

```
head(loc$countries)

## [1] "Afghanistan"         "Albania"             "Algeria"
## [4] "Andorra"             "Angola"              "Antigua and Barbuda"

head(loc$US_states)

## [1] "Alabama"         "Alaska"          "American Samoa" "Arizona"
## [5] "Arkansas"        "California"
```

To extract the Covid-19 data from a location of interest, function `extract_covid19_data()` will be used. If the interest is in the data from a country, set argument `data="global"`. If the interest is in the data from a US state, set argument `data="US"`. The output will be a list with three objects: `confirmed`, `deaths`, and `recovered`. For data on states of the US, `recovered` will be `NULL`. Each object has a first column with the name of a province/state/administration within the location, and the subsequent columns are the cumulative reported numbers on the dates indicated as the column's name. If no information is available for sublocations, the first column will be empty and only one row of data will be output. We offer three examples to illustrate.

China

```
dataCovid <- extract_covid19_data("China","global",folder=tmp.folder,filename=tmp.filename)
head(dataCovid$confirmed[,1:8],7)

##   Province/State 1/22/20 1/23/20 1/24/20 1/25/20 1/26/20 1/27/20 1/28/20
## 1          Anhui       1       9      15      39      60      70     106
## 2        Beijing      14      22      36      41      68      80      91
## 3      Chongqing       6       9      27      57      75     110     132
## 4         Fujian       1       5      10      18      35      59      80
## 5          Gansu       0       2       2       4       7      14      19
## 6      Guangdong      26      32      53      78     111     151     207
## 7        Guangxi       2       5      23      23      36      46      51

head(dataCovid$deaths[,1:8],7)

##   Province/State 1/22/20 1/23/20 1/24/20 1/25/20 1/26/20 1/27/20 1/28/20
## 1          Anhui       0       0       0       0       0       0       0
## 2        Beijing       0       0       0       0       0       1       1
## 3      Chongqing       0       0       0       0       0       0       0
## 4         Fujian       0       0       0       0       0       0       0
## 5          Gansu       0       0       0       0       0       0       0
## 6      Guangdong       0       0       0       0       0       0       0
## 7        Guangxi       0       0       0       0       0       0       0

head(dataCovid$recovered[,1:8],7)

##   Province/State 1/22/20 1/23/20 1/24/20 1/25/20 1/26/20 1/27/20 1/28/20
## 1          Anhui       0       0       0       0       0       0       0
## 2        Beijing       0       0       1       2       2       2       4
## 3      Chongqing       0       0       0       0       0       0       0
## 4         Fujian       0       0       0       0       0       0       0
## 5          Gansu       0       0       0       0       0       0       0
## 6      Guangdong       0       2       2       2       2       4       4
## 7        Guangxi       0       0       0       0       0       0       2
```

Italy

```
dataCovid <- extract_covid19_data("Italy","global",folder=tmp.folder,filename=tmp.filename)
dataCovid$confirmed[,1:8]

##   Province/State 1/22/20 1/23/20 1/24/20 1/25/20 1/26/20 1/27/20 1/28/20
## 1                      0       0       0       0       0       0       0

dataCovid$deaths[,1:8]

##   Province/State 1/22/20 1/23/20 1/24/20 1/25/20 1/26/20 1/27/20 1/28/20
## 1                      0       0       0       0       0       0       0

dataCovid$recovered[,1:8]

##   Province/State 1/22/20 1/23/20 1/24/20 1/25/20 1/26/20 1/27/20 1/28/20
## 1                      0       0       0       0       0       0       0
```

Michigan (US state)

```
dataCovid <- extract_covid19_data("Michigan","US",folder=tmp.folder,filename=tmp.filename)
head(dataCovid$confirmed[,1:8])

##    Admin2 1/22/20 1/23/20 1/24/20 1/25/20 1/26/20 1/27/20 1/28/20
## 1  Alcona       0       0       0       0       0       0       0
## 2   Alger       0       0       0       0       0       0       0
## 3 Allegan       0       0       0       0       0       0       0
## 4  Alpena       0       0       0       0       0       0       0
## 5  Antrim       0       0       0       0       0       0       0
## 6  Arenac       0       0       0       0       0       0       0

head(dataCovid$deaths[,1:8])

##    Admin2 1/22/20 1/23/20 1/24/20 1/25/20 1/26/20 1/27/20 1/28/20
## 1  Alcona       0       0       0       0       0       0       0
## 2   Alger       0       0       0       0       0       0       0
## 3 Allegan       0       0       0       0       0       0       0
## 4  Alpena       0       0       0       0       0       0       0
## 5  Antrim       0       0       0       0       0       0       0
## 6  Arenac       0       0       0       0       0       0       0

head(dataCovid$recovered[,1:8])

## NULL
```

Now, we want to build epidemic tables, with the data displayed in columns. This will be done using function `mkEpiTable()`. By default, the output is a table with cumulative the cumulative numbers of confirmed cases, deaths and recovered, as well as the number of active cases on the dates. Setting `daily=TRUE` will change the output to a table with the numbers reported in each day. The number of active cases is reported equally for both `daily=TRUE` and `daily=FALSE`, as this number is neither cumulative nor new occurences. For locations with detailed data by province/state/administration, data can be narrowed to a specific sublocation, or a set of sublocations. We offer a few examples on two different locations to illustrate.

China

```
dataCovid <- extract_covid19_data("China","global",folder=tmp.folder,filename=tmp.filename)
tbCovid <- mkEpiTable(dataCovid)
head(tbCovid)
```

```
##         date confirmed deaths recovered active
## 1 2020-01-22       548     17        28    503
## 2 2020-01-23       643     18        30    595
## 3 2020-01-24       920     26        36    858
## 4 2020-01-25      1406     42        39   1325
## 5 2020-01-26      2075     56        49   1970
## 6 2020-01-27      2877     82        58   2737
```

```
tbCovid <- mkEpiTable(dataCovid,daily=TRUE)
head(tbCovid)
```

```
##         date confirmed deaths recovered active
## 1 2020-01-22       548     17        28    503
## 2 2020-01-23        95      1         2    595
## 3 2020-01-24       277      8         6    858
## 4 2020-01-25       486     16         3   1325
## 5 2020-01-26       669     14        10   1970
## 6 2020-01-27       802     26         9   2737
```

```
tbCovid <- mkEpiTable(dataCovid,specific="Hubei")
head(tbCovid)
```

```
##         date confirmed deaths recovered active
## 1 2020-01-22       444     17        28    399
## 2 2020-01-23       444     17        28    399
## 3 2020-01-24       549     24        31    494
## 4 2020-01-25       761     40        32    689
## 5 2020-01-26      1058     52        42    964
## 6 2020-01-27      1423     76        45   1302
```

```
tbCovid <- mkEpiTable(dataCovid,specific=c("Hubei","Guangdong"))
head(tbCovid)
```

```
##         date confirmed deaths recovered active
## 1 2020-01-22       470     17        28    425
## 2 2020-01-23       476     17        30    429
## 3 2020-01-24       602     24        33    545
## 4 2020-01-25       839     40        34    765
## 5 2020-01-26      1169     52        44   1073
## 6 2020-01-27      1574     76        49   1449
```

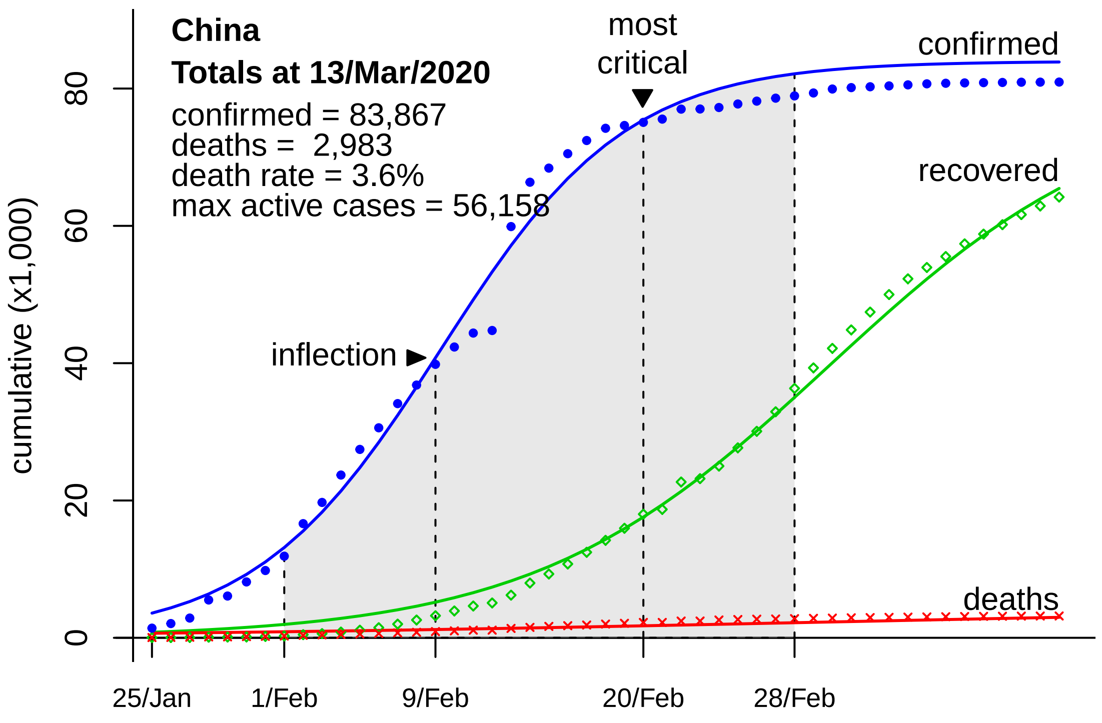
Michigan (US state)

```
dataCovid <- extract_covid19_data("Michigan","US",folder=tmp.folder,filename=tmp.filename)
tbCovid <- mkEpiTable(dataCovid)
head(tbCovid)

##          date confirmed deaths recovered active
## 50 2020-03-11         2      0        NA     NA
## 51 2020-03-12         2      0        NA     NA
## 52 2020-03-13        16      0        NA     NA
## 53 2020-03-14        25      0        NA     NA
## 54 2020-03-15        32      0        NA     NA
## 55 2020-03-16        54      0        NA     NA

head(tbCovid[tbCovid$confirmed > 0,])

##          date confirmed deaths recovered active
## 50 2020-03-11         2      0        NA     NA
## 51 2020-03-12         2      0        NA     NA
## 52 2020-03-13        16      0        NA     NA
## 53 2020-03-14        25      0        NA     NA
## 54 2020-03-15        32      0        NA     NA
## 55 2020-03-16        54      0        NA     NA

tbCovid <- mkEpiTable(dataCovid,specific="Ingham")
head(tbCovid)

##          date confirmed deaths recovered active
## 52 2020-03-13         1      0        NA     NA
## 53 2020-03-14         1      0        NA     NA
## 54 2020-03-15         1      0        NA     NA
## 55 2020-03-16         1      0        NA     NA
## 56 2020-03-17         3      0        NA     NA
## 57 2020-03-18         5      0        NA     NA

head(tbCovid[tbCovid$confirmed > 0,])

##          date confirmed deaths recovered active
## 52 2020-03-13         1      0        NA     NA
## 53 2020-03-14         1      0        NA     NA
## 54 2020-03-15         1      0        NA     NA
## 55 2020-03-16         1      0        NA     NA
## 56 2020-03-17         3      0        NA     NA
## 57 2020-03-18         5      0        NA     NA
```

The next step is to use these tables generated by `mkEpiTable()` to model the logistic curves of the epidemic in the location of interest. We will now use the function `mkEpiCurves()`.

China

```
dataCovid <- extract_covid19_data("China","global",folder=tmp.folder,filename=tmp.filename)
tbCovid <- mkEpiTable(dataCovid)
fitCovid <- mkEpiCurves(tbCovid,plot.title="China")
```

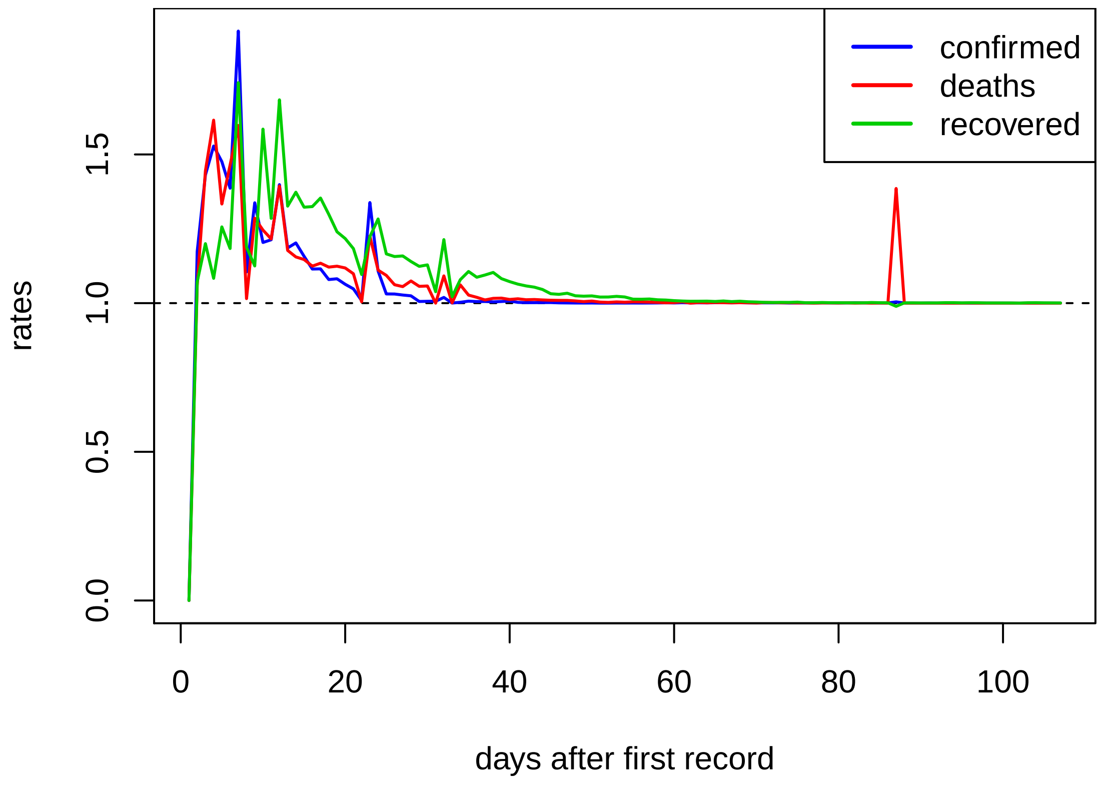


```
# print the parameters of the curves
fitCovid$par

##                   a        b         c
## confirmed 83975.02 19.28183  4.917805
## deaths     4637.00 40.11560 20.148926
## recovered 79338.02 39.84679  7.838458

# print the dates in which the first observation of the epidemic events occurred
fitCovid$day1

##    confirmed       deaths    recovered
## "2020-01-22" "2020-01-22" "2020-01-22"

# print the key dates of the of this epidemic
fitCovid$key_dates

## start_critical     inflection  most_critical    end_critical
##   "2020-02-01"   "2020-02-09"   "2020-02-20"    "2020-02-28"

# rates with which the epidemic event develops
head(fitCovid$rates)
```

```
##         date confirmed   deaths recovered
## 1 2020-01-22  0.000000 0.000000  0.000000
## 2 2020-01-23  1.173358 1.058824  1.071429
## 3 2020-01-24  1.430793 1.444444  1.200000
## 4 2020-01-25  1.528261 1.615385  1.083333
## 5 2020-01-26  1.475818 1.333333  1.256410
## 6 2020-01-27  1.386506 1.464286  1.183673
```

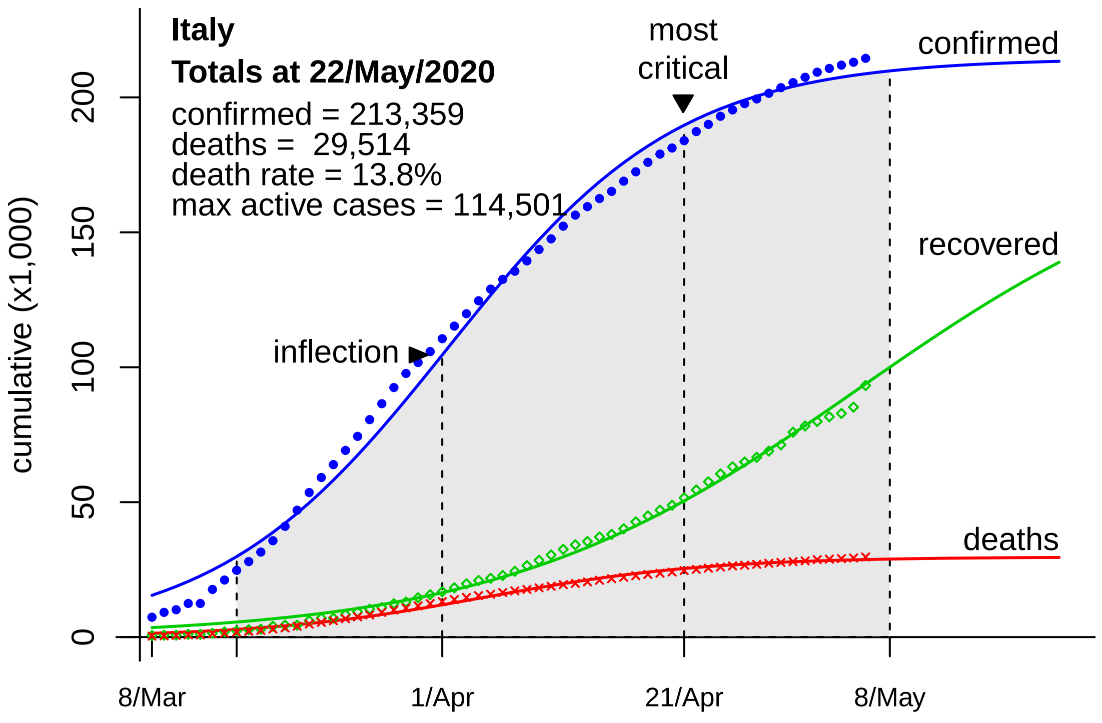
The rates in `fitCovid$rates` are the percentage with which the events develop. For example, if in a day the total number reported of the epidemic event is 100, and 120 on the next day, the rate observed in this next day is $120/100 = 1.2$, an increase of 20%.

```
# plot the rates
par(mar=c(4,4.5,0,1))
xrange <- c(1,nrow(fitCovid$rates))
yrange <- range(fitCovid$rates[,-1])
plot(NA,xlim=xrange,ylim=yrange,xlab="days after first record",ylab="rates")
abline(h=1,lty=2)
lines(1:nrow(fitCovid$rates),fitCovid$rates$confirmed,lwd=1.5,col=4)
lines(1:nrow(fitCovid$rates),fitCovid$rates$deaths,lwd=1.5,col=2)
lines(1:nrow(fitCovid$rates),fitCovid$rates$recovered,lwd=1.5,col=3)
legend("topright",lwd=2,col=c(4,2,3),legend=names(fitCovid$rates)[-1])
```

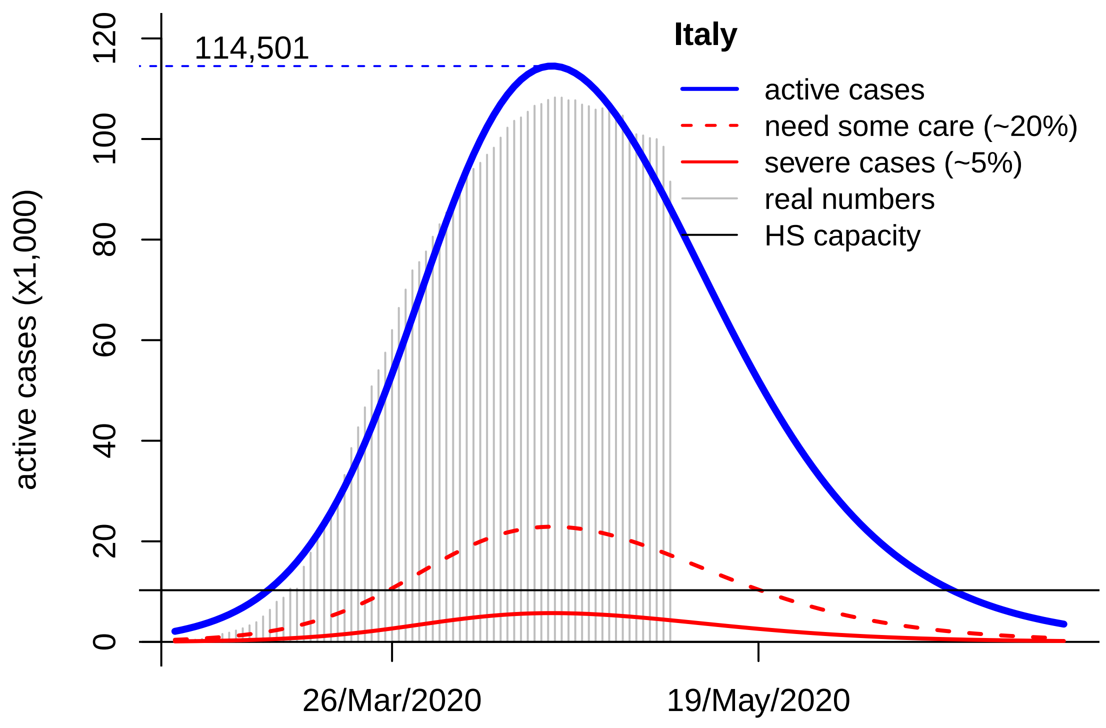


Italy

```
dataCovid <- extract_covid19_data("Italy","global",folder=tmp.folder,filename=tmp.filename)
tbCovid <- mkEpiTable(dataCovid)
fitCovid <- mkEpiCurves(tbCovid,plot.title="Italy")
```

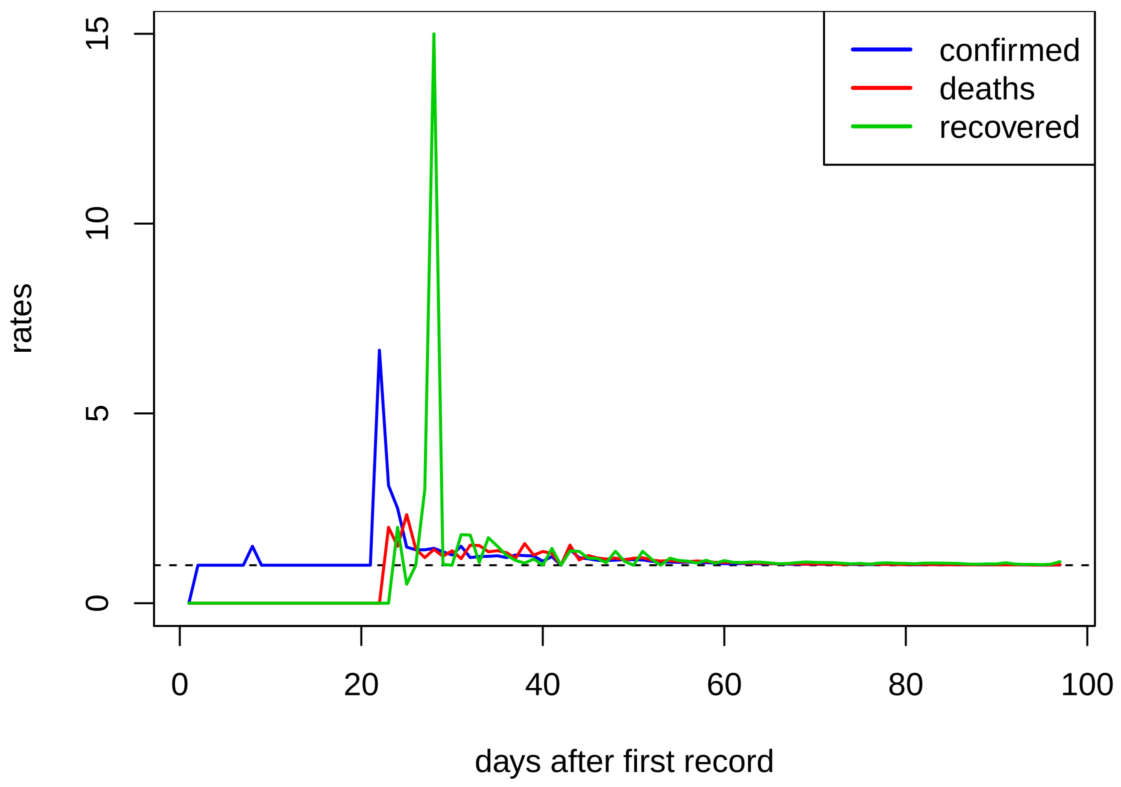


```
# extract information about the health system, along wih demographic data
head(CountryData) # data obtained from the WHO website

##                country  pop2018 HospBeds Physicians Nurses
## 1          Afghanistan 37172386      0.5     0.2840 0.3200
## 2              Albania  2866376      2.9     1.1998 3.5998
## 3              Algeria 42228429      1.9     1.8300 2.2400
## 4              Andorra    77006      2.5     3.3333 4.0128
## 5               Angola 30809762      0.8     0.2149 1.3123
## 6  Antigua and Barbuda    96286      3.8     2.7647 3.1176

hs <- CountryData[CountryData$country == "Italy",3:5]
hs <- hs*CountryData[CountryData$country == "Italy","pop2018"]/1000
names(hs) <- c("beds","phys","nurs")
hs

##        beds     phys     nurs
## 99 205466.4 247351.3 354689.3

# plot the curve of active cases accounting for the proportions of cases that
# require some medical care and the proportion of cases that are critical
# this proportions can be changed by addind the argument 'hospital.cases'
# if there is no information about the health system, just omit the argument
mkEpiPlot(tbCovid,fitCovid,type="active",healthsystem=hs,plot.title="Italy")
```



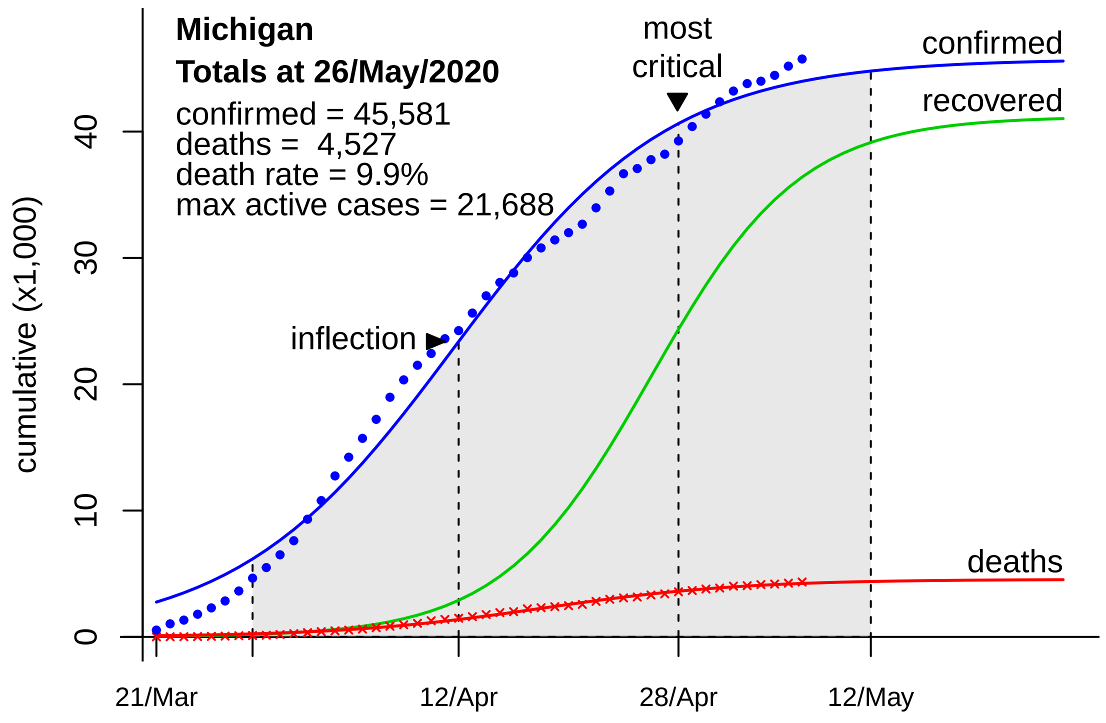
```
# print the parameters of the curves
fitCovid$par

##                a        b         c
## confirmed 214457 62.47745  9.588516
## deaths     29684 44.60913  9.194025
## recovered 184773 74.54253 14.861656

# print the dates in which the first observation of the epidemic events occurred
fitCovid$day1

##    confirmed       deaths    recovered
## "2020-01-31" "2020-02-21" "2020-02-22"

# print the key dates of the of this epidemic
fitCovid$key_dates

## start_critical     inflection  most_critical   end_critical
##   "2020-03-15"   "2020-04-01"   "2020-04-21"   "2020-05-08"

# rates with which the epidemic event develops
head(fitCovid$rates)

##         date confirmed deaths recovered
## 1 2020-01-31         0      0         0
## 2 2020-02-01         1      0         0
## 3 2020-02-02         1      0         0
## 4 2020-02-03         1      0         0
## 5 2020-02-04         1      0         0
## 6 2020-02-05         1      0         0
```

```r
# plot the rates
par(mar=c(4,4.5,0,1))
xrange <- c(1,nrow(fitCovid$rates))
yrange <- range(fitCovid$rates[,-1])
plot(NA,xlim=xrange,ylim=yrange,xlab="days after first record",ylab="rates")
abline(h=1,lty=2)
lines(1:nrow(fitCovid$rates),fitCovid$rates$confirmed,lwd=1.5,col=4)
lines(1:nrow(fitCovid$rates),fitCovid$rates$deaths,lwd=1.5,col=2)
lines(1:nrow(fitCovid$rates),fitCovid$rates$recovered,lwd=1.5,col=3)
legend("topright",lwd=2,col=c(4,2,3),legend=names(fitCovid$rates)[-1])
```



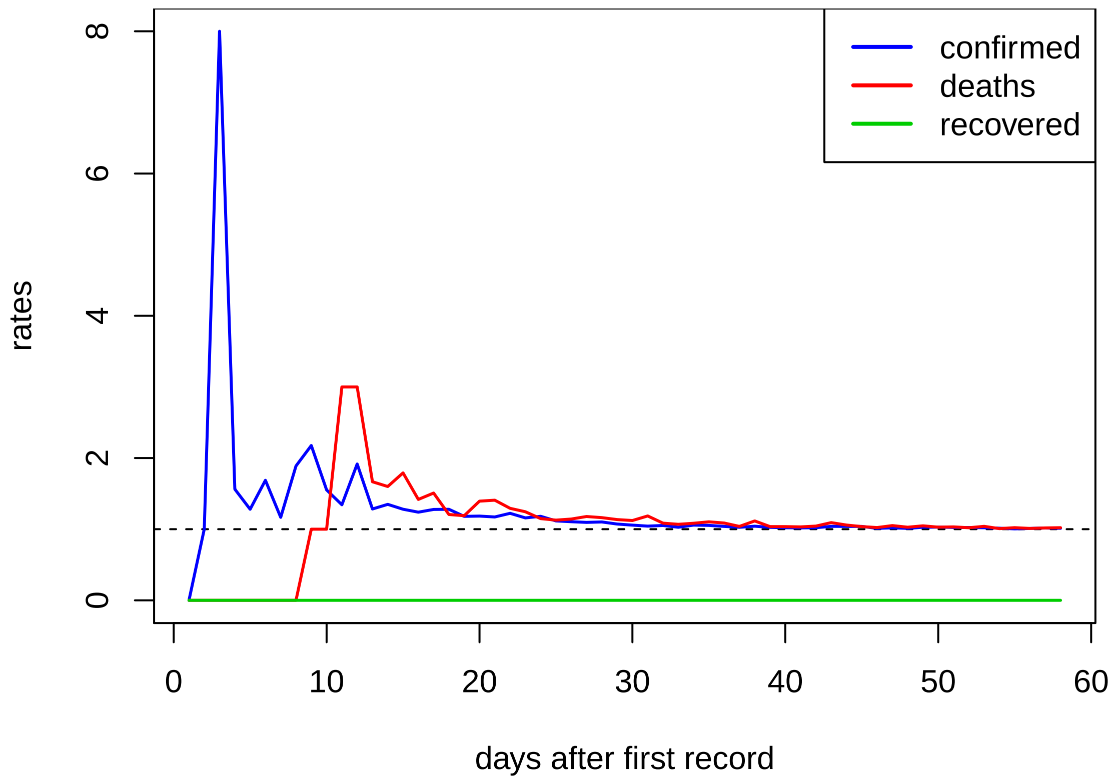
Michigan (US state)

```r
dataCovid <- extract_covid19_data("Michigan","US",folder=tmp.folder,filename=tmp.filename)
tbCovid <- mkEpiTable(dataCovid)
fitCovid <- mkEpiCurves(tbCovid,plot.title="Michigan")
```



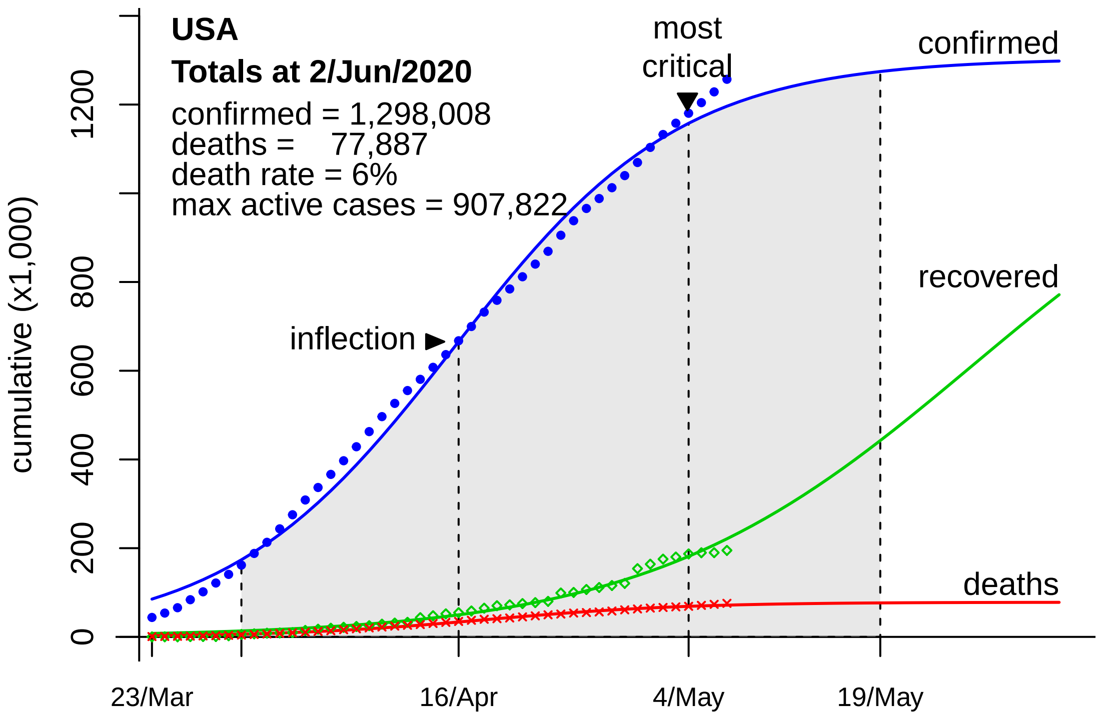
```
# print the parameters of the curves
fitCovid$par

##                   a        b        c
## confirmed 45745.000 32.64963 7.881597
## deaths     4552.732 32.10768 7.314301
## recovered 41192.268 33.00000 5.424665

# print the dates in which the first observation of the epidemic events occurred
fitCovid$day1

##    confirmed       deaths    recovered
## "2020-03-11" "2020-03-18" "2020-03-25"

# print the key dates of the of this epidemic
fitCovid$key_dates

## start_critical     inflection  most_critical   end_critical
##   "2020-03-28"   "2020-04-12"   "2020-04-28"   "2020-05-12"

# rates with which the epidemic event develops
head(fitCovid$rates)

##         date confirmed deaths recovered
## 1 2020-03-11    0.0000      0         0
## 2 2020-03-12    1.0000      0         0
## 3 2020-03-13    8.0000      0         0
## 4 2020-03-14    1.5625      0         0
## 5 2020-03-15    1.2800      0         0
## 6 2020-03-16    1.6875      0         0
```

```
# plot the rates
par(mar=c(4,4.5,0,1))
xrange <- c(1,nrow(fitCovid$rates))
yrange <- range(fitCovid$rates[,-1])
plot(NA,xlim=xrange,ylim=yrange,xlab="days after first record",ylab="rates")
abline(h=1,lty=2)
lines(1:nrow(fitCovid$rates),fitCovid$rates$confirmed,lwd=1.5,col=4)
lines(1:nrow(fitCovid$rates),fitCovid$rates$deaths,lwd=1.5,col=2)
lines(1:nrow(fitCovid$rates),fitCovid$rates$recovered,lwd=1.5,col=3)
legend("topright",lwd=2,col=c(4,2,3),legend=names(fitCovid$rates)[-1])
```



United States

```
dataCovid <- extract_covid19_data("US","global",folder=tmp.folder,filename=tmp.filename)
tbCovid <- mkEpiTable(dataCovid)
fitCovid <- mkEpiCurves(tbCovid,plot.title="USA")
```

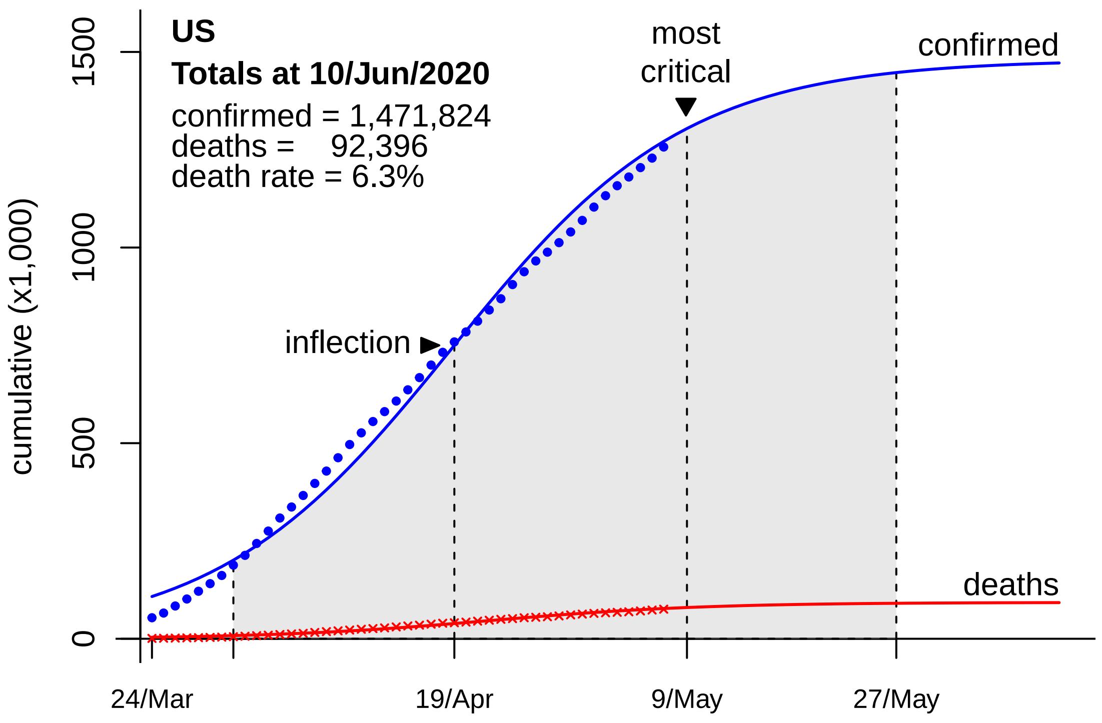


```
# print the parameters of the curves
fitCovid$par

##                    a         b         c
## confirmed 1304286.05  85.63558  8.883872
## deaths      78145.88  50.22634  7.847138
## recovered 1226140.17 108.27220 12.729248
```

We can work with projections to better understand the situation in the US. In this case, we will generate two scenarios for this projection, one projecting seven days, and another projecting fourteen days.

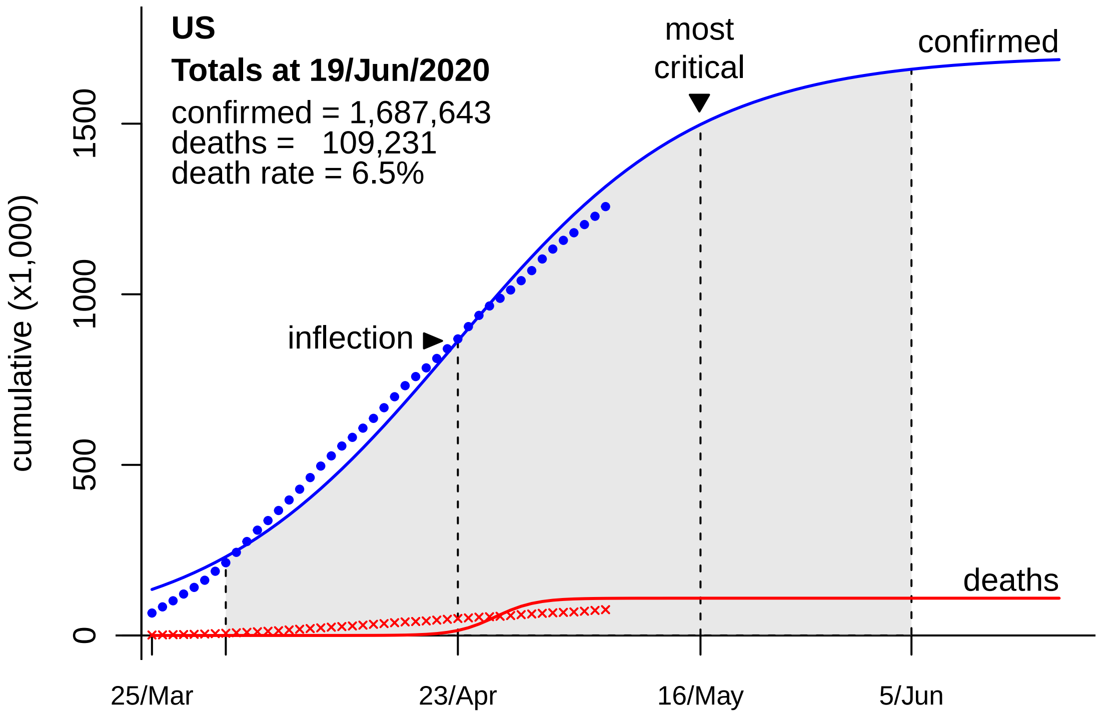
```
#----------------#
# project 7 days #
#----------------#
fitCovid <- mkEpiCurves(tbCovid,plot.title="USA",project=7,plot.curves=FALSE)
# print the parameters of the curves
fitCovid$par

##                   a        b         c
## confirmed 1480229.7 88.72984 10.119361
## deaths      92886.7 54.01576  9.349911
## recovered 1387343.0 89.00000  6.746249

# print the key dates of the of this epidemic
fitCovid$key_dates
```

```
## start_critical     inflection  most_critical    end_critical
##   "2020-03-31"   "2020-04-19"   "2020-05-09"   "2020-05-27"

# print the rates of the projection
unlist(fitCovid$rates_proj)

## confirmed    deaths
##  1.021077  1.026575

# generate plot
mkEpiPlot(tbCovid,fitCovid,plot.title="US",hide.max.active=TRUE,plot.recovered=FALSE)
```



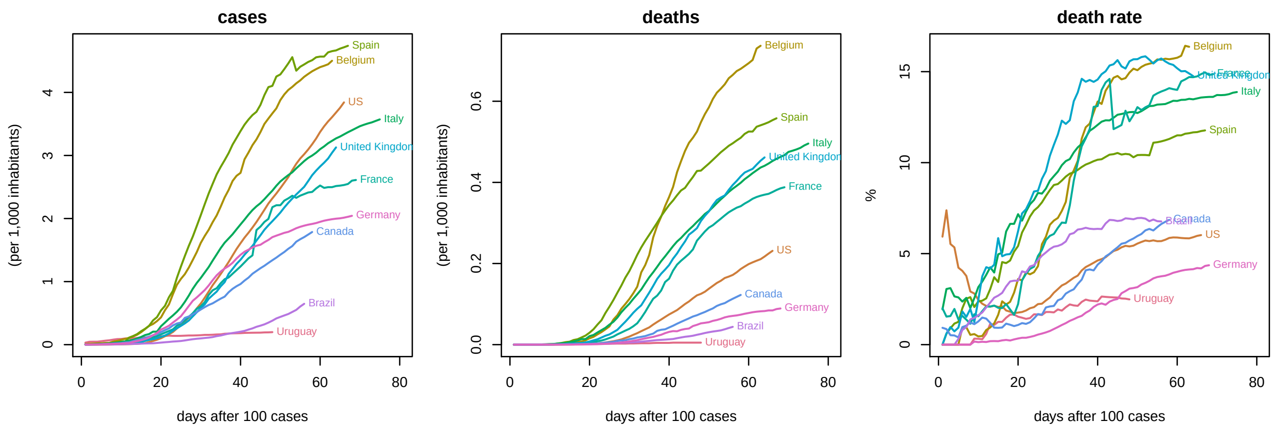
```
#-----------------#
# project 14 days #
#-----------------#
fitCovid <- mkEpiCurves(tbCovid,plot.title="USA",project=14,plot.curves=FALSE)
# print the parameters of the curves
fitCovid$par

##                 a        b         c
## confirmed 1700060 92.63147 11.679142
## deaths     109231 58.55597  1.904956
## recovered 1590829 93.00000  6.979663

# print the key dates of the of this epidemic
fitCovid$key_dates

## start_critical     inflection  most_critical    end_critical
##   "2020-04-01"   "2020-04-23"   "2020-05-16"   "2020-06-05"
```

```
# print the rates of the projection
unlist(fitCovid$rates_proj)

## confirmed    deaths
##   1.021077  1.026575

# generate plot
mkEpiPlot(tbCovid,fitCovid,plot.title="US",hide.max.active=TRUE,plot.recovered=FALSE)
```

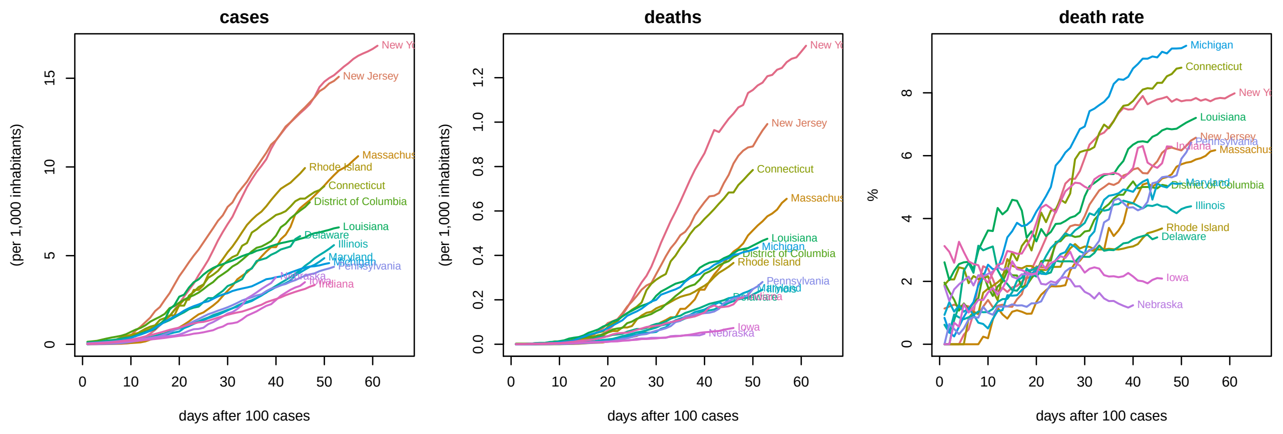


It is possible to compare the scenario (without projections) in different countries, or in different states in the US.

```
location <- c("Uruguay","US","Belgium","Spain","Italy","France","United Kingdom",
              "Canada","Brazil","Germany")
data <- "global"
par(mfrow=c(1,3))
tmp <- mkEpiComparePlot(location,data,tmp.folder,tmp.filename)
```



```
# sort information on US states by those with most cases/1,000 inhabitants
aux <- data.frame(state=character(0),total=numeric(0))
for(k in loc$US_states[!(loc$US_states %in% c("American Samoa","Guam",
     "Northern Mariana Islands","Virgin Islands","Puerto Rico",
     "Diamond Princess","Grand Princess"))])
{
  tmp <- mkEpiTable(extract_covid19_data(location=k,data="US",tmp.folder,tmp.filename))
  if(nrow(tmp) > 0)
  {
    aux <- rbind(aux,data.frame(state=k,total=1000*tmp$confirmed[nrow(tmp)]/
            USstatesData$pop2019[USstatesData$state == k]))
    rm(tmp)
  }
}
aux <- aux[sort(aux$total,decreasing=TRUE,index.return=TRUE)$ix,]
rownames(aux) <- 1:nrow(aux)
head(aux)

##                   state     total
## 1              New York 16.833371
## 2            New Jersey 15.085356
## 3         Massachusetts 10.608097
## 4          Rhode Island  9.939954
## 5           Connecticut  8.914850
## 6  District of Columbia  8.011347

location <- aux$state[1:15]
par(mfrow=c(1,3))
tmp <- mkEpiComparePlot(location,data="US",tmp.folder,tmp.filename)
```


cases
deaths
death rate
(per 1,000 inhabitants)
days after 100 cases
New York
New Jersey
Massachusetts
Rhode Island
Connecticut
District of Columbia
Louisiana
Delaware
Illinois
Maryland
Michigan
Pennsylvania
Nebraska
Indiana
Iowa