



EPIDEMIOLOGY EDUCATIONAL CODE

www.EpidemicCode.org

How to cite this material?

B. Pavlack, J. Basilio, E. Dantas, M. Grave, L. de la Roca, A. Cunha Jr,
et al., EPIDEMIC - Epidemiology Educational Code, 2020
www.EpidemicCode.org

Development team

Professors/ Researchers:	Americo Cunha Lisandro Lovisolo Malú Grave Rodrigo Burgos	(UERJ) (UERJ) (UFRJ) (UERJ)
-------------------------------------	--	--------------------------------------

Students:	Bruna Pavlack Diego Matos Eber Dantas João P. Norenberg Julio Basilio	(IFMS) (UERJ) (UFRJ) (UNESP) (UERJ)	Leonardo de la Roca Lucas Chaves Marcos Issa Michel Tosin Roberto Luo	(UERJ) (UFU) (UERJ) (UERJ) (UERJ)
------------------	---	---	---	---

Graphic design: Amanda Cunha Guyt
Luthiana Souza Soares

Contact: Americo Cunha americocunha@uerj.br

What is EPIDEMIC?

EPIDEMIC is an educational tool for epidemiological analysis (simulations, monitoring, etc.), developed by researchers and professors of several research institutions during the COVID-19 pandemic.

Mission:

- analyze indicators of an epidemic evolution;
- Inform about simulations, analyzes, epidemiological predictions;
- provide a didactic and intuitive tool for interested audiences.



EPIDEMIC is available at: www.EpidemicCode.org

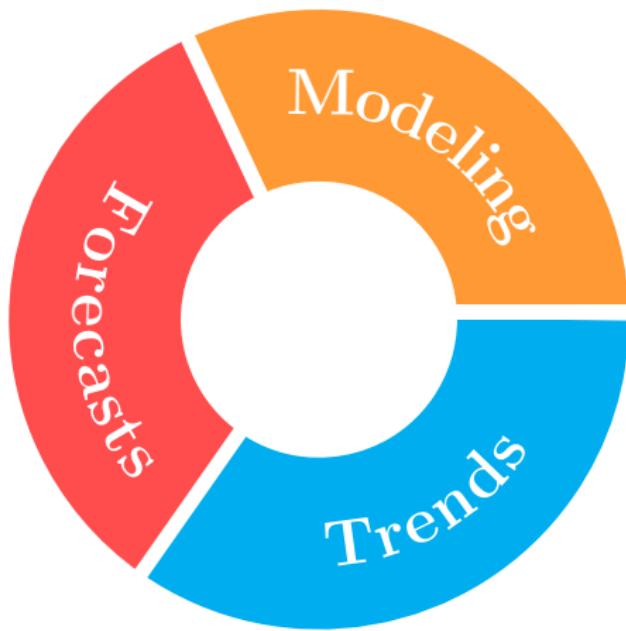
EPIDEMIC philosophy

- Contribute, in a didactic way, to the understanding of some of the main epidemiological forecasting tools;
- Make computational codes available in an accessible way for epidemiological modeling, following the principles of transparency, reproducibility and validity;
- Provide a customizable environment in which the user can adjust the codes according to their context of interest;
- Reinforce the importance of science in combating an epidemic.



Towards data science. Transparency, Reproducibility, and Validity of COVID-19 Projection Models, <shorturl.at/IuK49>. Access: June 30, 2020.

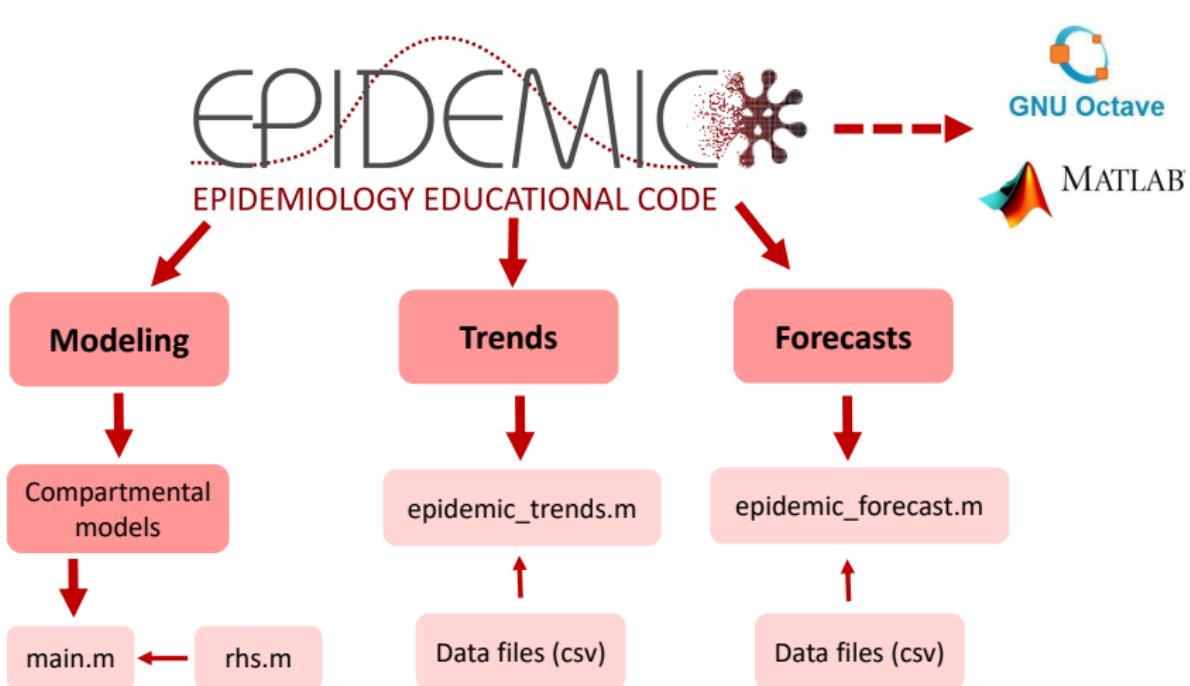
EPIDEMIC modules



EPIDEMIC modules

- **MODELING:** compartmental models described by differentials equation to simulate population dynamics during an epidemic;
- **TRENDS:** graphical analysis of trends to explore the dynamic behavior of an epidemic and its progress;
- **FORECASTS:** statistical regressor that uses data to perform short-term forecasts quantities of interest.

EPIDEMIC Structure



Code files in EPIDEMIC

Modeling		Trends	Forecasts
SIR	main_SIR.m rhs_SIR.m		
SEIR	main_SEIR.m rhs_SEIR.m		
SEIRD	main_SEIRD.m rhs_SEIRD.m	epidemic_trends.m	epidemic_forecasts.m
SEIAHRD	main_SEIAHRD.m rhs_SEIAHRD.m		
SIRG	main_SIRG.m rhs_SIRG.m		

Software

The EPIDEMIC code was developed in free software GNU Octave[©].



EPIDEMIC is also compatible with the proprietary software MATLAB[©].

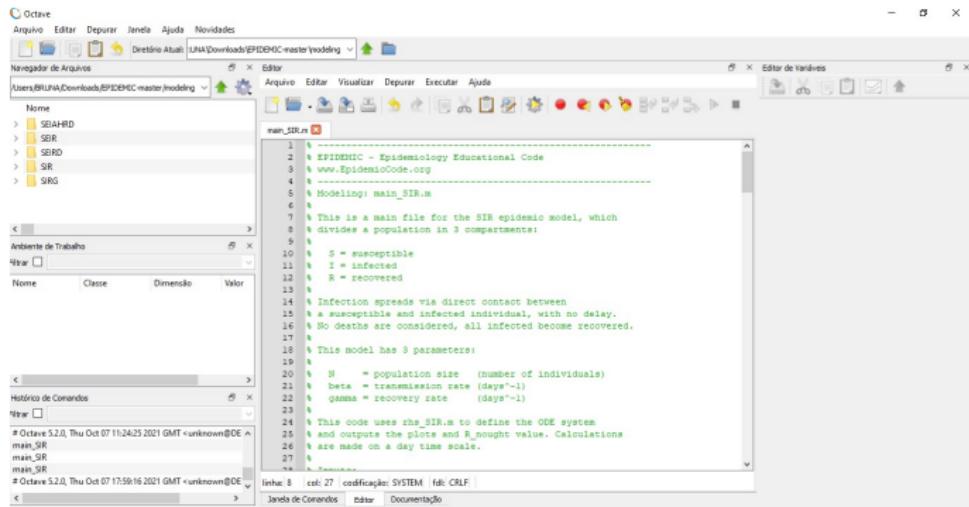
- EPIDEMIC codes run from GNU Octave 5.2.0 and MATLAB R2016b versions;
- Information on the language MATLAB can be found in the book by D. J. Higham and N. J. Higham (2017).

Installation and use

To run the EPIDEMIC code, follow these steps:

- ① Access the EPIDEMIC repository, available at www.github.com/americo/EPIDEMIC and download the folder with the code;
- ② The working directory of the compatible software (Octave or MATLAB) is placed in the place where the folder with the EPIDEMIC codes is located and it is executed via the terminal;
- ③ If it is in the module Modeling, the executable file that generates the simulations is the files *main*. In the files *rhs* is possible to modify the differential equations, if that is the interest. In the module Trends, the executable file is *epidemic_trends.m*. And in the module Forecasts the executable file is *epidemic_forecasts.m*.

Code execution example in Octave GNU



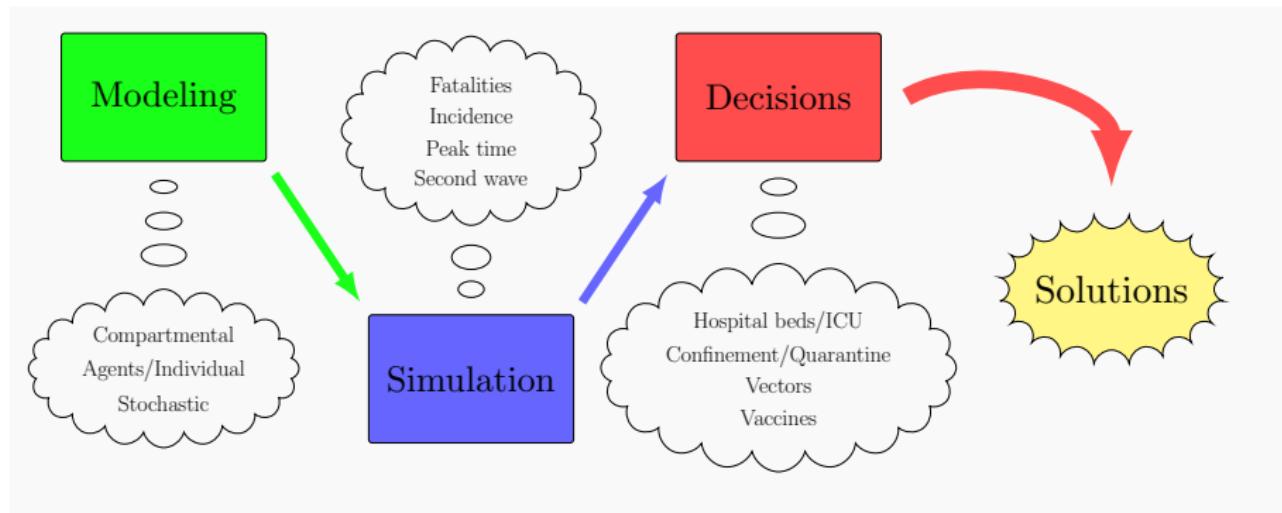
In the example on the screen above, the Octave GNU working directory is in place [EPIDEMIC-master\modeling](#), which allows accessing and executing the codes of the module Modeling of EPIDEMIC, being able to choose among the models: SEIAHRD, SEIR, SEIRD, SIR and SIRG. To run the script in Octave, just click on the icon .

Summary

- 1** Modeling
- 2** Trends
- 3** Forecasts
- 4** Acknowledgment
- 5** References

Modeling

Epidemiological modeling



Compartmental models

In a compartmental model, the population of interest is divided into **compartments** according to their relationship with the disease over time t : susceptible, infected, hospitalized, quarantined, confined, etc.

The simplest compartmental model is the **SIR**.

- **Susceptible**: can become infected;
- **Infected**: carry the pathogen; can recover;
- **Recovered**: got rid of the pathogen.



The arrows dictate the flow of individuals from one compartment to another.

Compartmental models

Deterministic compartmental models: with a previous state you can predict all the progress of the dynamic.

Considering $N = S + I + R$ the total number of the population, the number of individuals in each compartment over time t is determined by the solution of system of differential equations.

$$\frac{dS}{dt} = -\beta S \frac{I}{N}, \quad \frac{dI}{dt} = \beta S \frac{I}{N} - \gamma I, \quad \frac{dR}{dt} = \gamma I$$

- The flows are proportional to the **compartment size**;
- Transmission occurs from I to S at a rate β ;
- Individuals in I recover at a rate γ .

MODELING → SIMULATING EPIDEMICS → ANALYZE CHARACTERISTICS

Compartmental models: advantages

- **Simple:** summarize the dynamics as the action of groups (low dimension).
- **Flexible:**
 - more compartments;
 - more than one population;
 - classes by socioeconomic factors (e.g.: age, income);
 - climatic effects/seasonality;
 - spatial effects;
 - stochastic variables.
- **Intuitive:** parameters and basic number of reproduction very clear.

What is the \mathcal{R}_0 and what is its importance?

\mathcal{R}_0 : basic reproduction number.

Classic definition: “the average number of secondary infections produced when a single infected individual is introduced into a population of hosts where everyone is susceptible”.

In mathematical modeling: dimensionless number that determines whether or not an epidemic will occur. It is a model parameter that delimits the point of *bifurcation* of the epidemic system.

The \mathcal{R}_0 of a model does not necessarily replicate the classic definition, because it depends on the **hypotheses** of model.



What is the \mathcal{R}_0 and what is its importance?

When there is one infected, there is only $S \approx N$, so that

$$\frac{dI}{dt} = \beta S \frac{I}{N} - \gamma I \quad \rightarrow \quad \frac{dI}{dt} = (\beta - \gamma) I$$

In this scenario, the infected curve presents an exponential behavior

$$I(t) = \exp(\gamma(\mathcal{R}_0 - 1)t), \quad \mathcal{R}_0 = \frac{\beta}{\gamma}$$

- If $\mathcal{R}_0 < 1 \rightarrow \text{There is no epidemic}$
- If $\mathcal{R}_0 > 1 \rightarrow \text{There is an epidemic}$

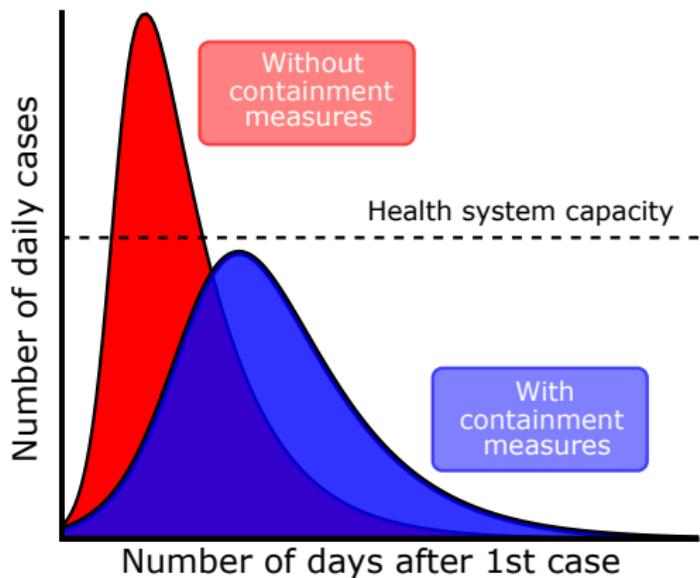
In cases of epidemics: smaller $\mathcal{R}_0 \rightarrow I(t)$ grows more slowly

The curve is “flattened”!

Why “flatten” the curve of infected?

“flatten” the curve means delaying the spread of the pathogen.

Very sharp curve → collapse of the health system → **more deaths**.



Modeling limitations

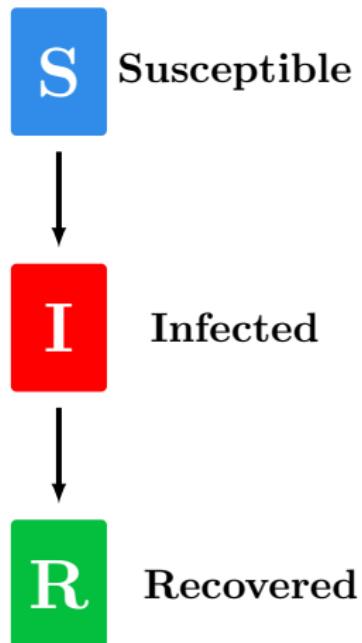
- Any model is a caricature of reality;
- Compartmental models do not consider individual effects, they assume that the population is homogeneous;
- Models need data to be validated;
- Very simple models have difficulty reproducing reality;
- Very complex models are difficult to use.

SIR model

- A population of constant size N is divided into **3 compartments**:

$$N = S(t) + I(t) + R(t)$$

- **Susceptible**: can become infected;
- **Infected**: carry the pathogen; infect susceptible; can recover;
- **Recovered**: got rid of the pathogen; do not transmit; have become immune;
- Transmission: direct contact between **I** and **S**.



Dynamics of the SIR model

- Initial Value Problem, which ODEs are:

$$\frac{dS}{dt} = -\beta S \frac{I}{N} \quad (\text{susceptible rate}),$$

$$\frac{dI}{dt} = \beta S \frac{I}{N} - \gamma I \quad (\text{infected rate}),$$

$$\frac{dR}{dt} = \gamma I \quad (\text{recovered rate}).$$

- Initial conditions $(S(0), I(0), R(0)) \xrightarrow[\text{of ODEs}]{\text{integration}} (S(t), I(t), R(t));$
- Time series $(S(t), I(t), R(t))$: describe how many individuals are in each compartment in time t .

Parameters of the SIR model



- N : total population (individuals);
- β : transmission rate (time^{-1});
- γ : recovery rate (time^{-1}).

Basic reproduction number — SIR

$$\mathcal{R}_0 = \frac{\beta}{\gamma}$$

Force of infection λ

- In general, $\frac{dS}{dt} = -\lambda(t) S$.
- Frequency dependent: $\lambda(t) = -\beta \frac{I(t)}{N}$.
 - β : constant transmission rate;
 - $I(t)/N$: probability of contact with an infected individual.
- Other models exist:
 - β dependent on population size N ;
 - β is time dependent t .



M. Begon, M. Bennett, R.G. Bowers, et al., A clarification of transmission terms in host-microparasite models: numbers, densities and areas, *Epidemiol. Infect.*, 129, 2002. DOI: <https://doi.org/10.1017/S0950268802007148>

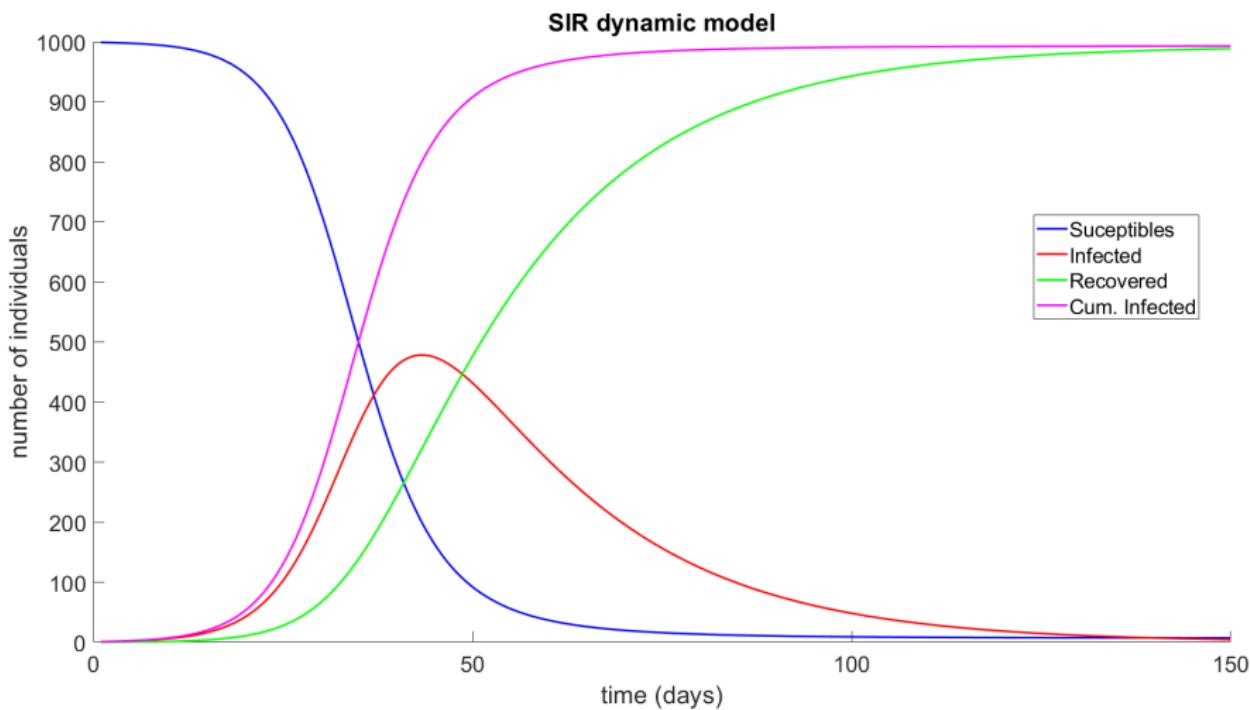
Cumulative of Infected

- The compartment $I(t)$ informs how many infected there are in time t ;
- The **cumulative of infected** $C(t)$ informs how many infected there are until the time t ;
- $C(t)$ can be calculated during integration, using the entry into I . For the SIR model:

$$\frac{dC}{dt} = \beta S \frac{I}{N}$$

- The **cumulative of infected** makes it possible to calculate a common measure of the epidemiology (**prevalence**) → comparison with data.

Time series - typical case



Time series - typical case

```
-----  
+++++ SIR model +++++  
-----  
* population      = 1000  
  (individuals)  
* transmission rate = 0.25  
  (days^-1)  
* recovery rate    = 0.05  
  (days^-1)  
* R_nought        = 5  
  (dimensionless)  
-----
```

- S : decay. The outbreak ends when S stabilizes (“very low”);
- I : exponential growth “to the beginning”;
- R : accumulation compartment. It does not interfere with the dynamics;
- **Cumulative of infected**: reports the cumulative number of infected to the time t . It does not interfere with the dynamics.

New cases

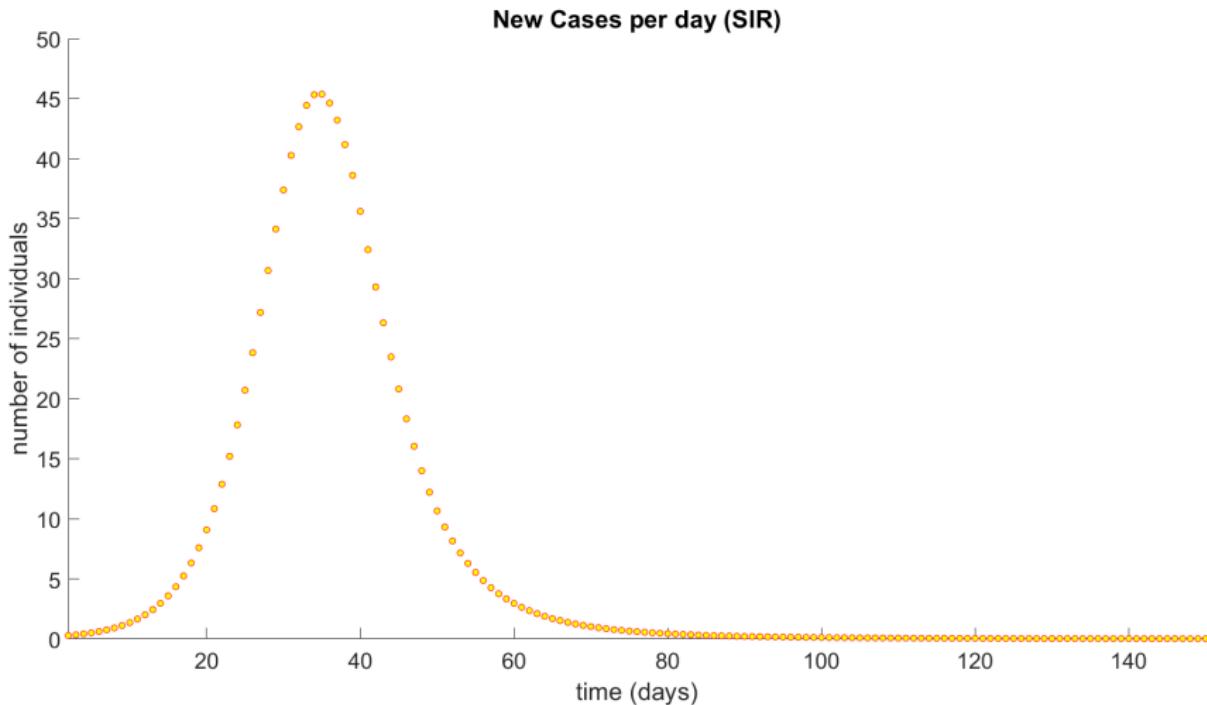
- The number of new cases per day can be obtained from the cumulative of infected:

$$\text{NewCases}(t_i) = C(t_i) - C(t_{i-1}),$$

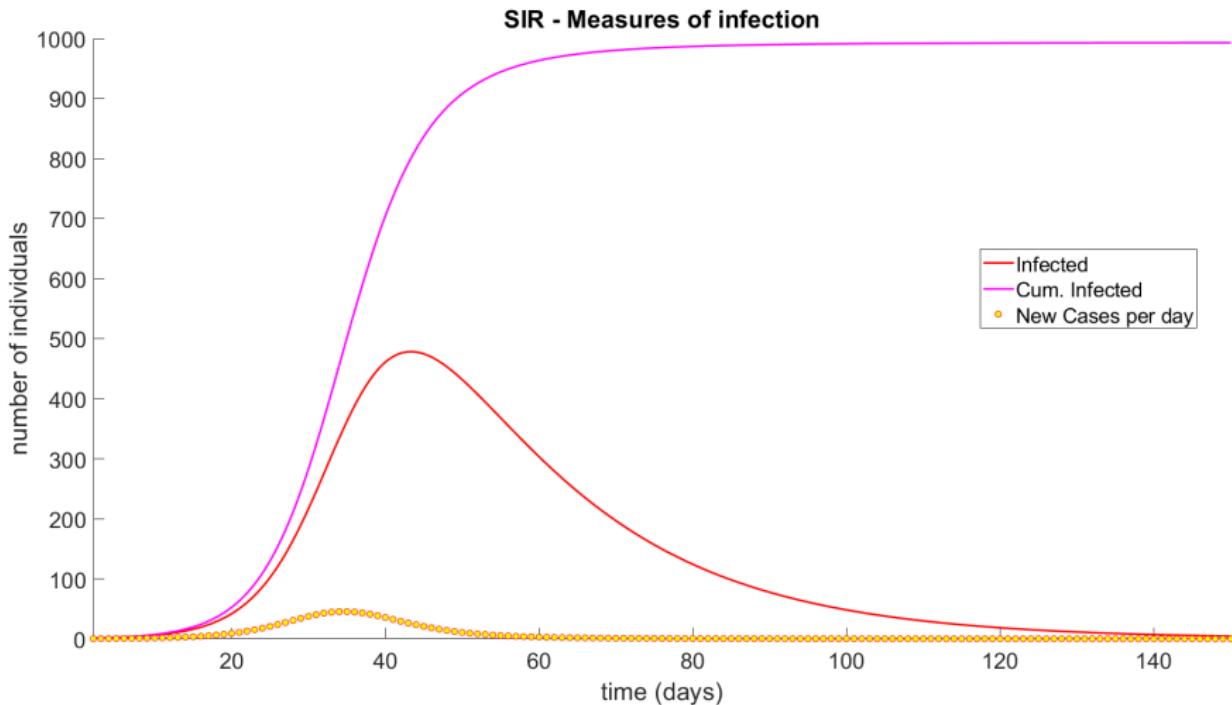
where t_i is the instant selected to represent a certain day.

- NewCases depends on the interval: per day, per week, per month, etc.
- $\text{NewCases} \rightarrow$ epidemiology (**incidence**) \rightarrow comparison with data.
- Infection measures (SIR):
 - $I(t)$, **current** number of infected in time t ;
 - $C(t)$, **total** of infected **until** the time t ;
 - $\text{NewCases}(t_i)$, **new** infected in t_i **in relation** to t_{i-1} .

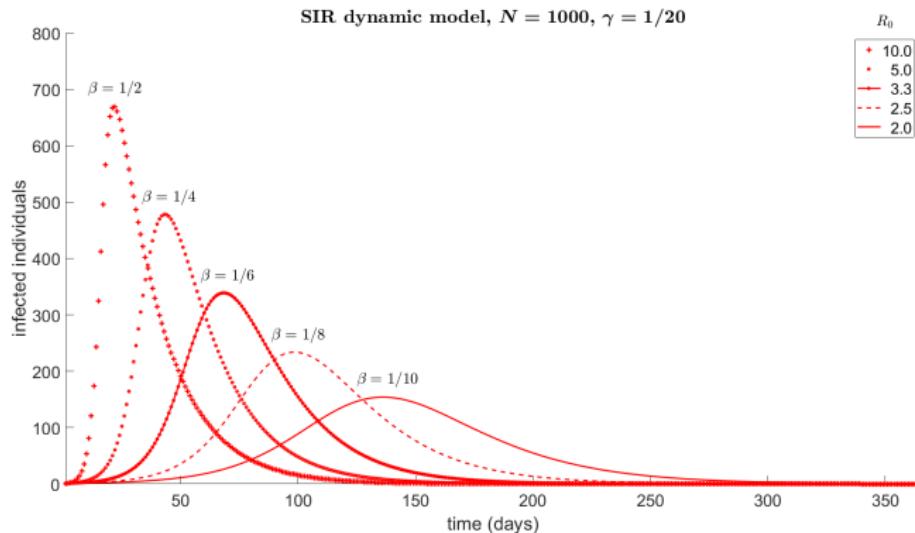
New cases



$I \times C \times NewCases$

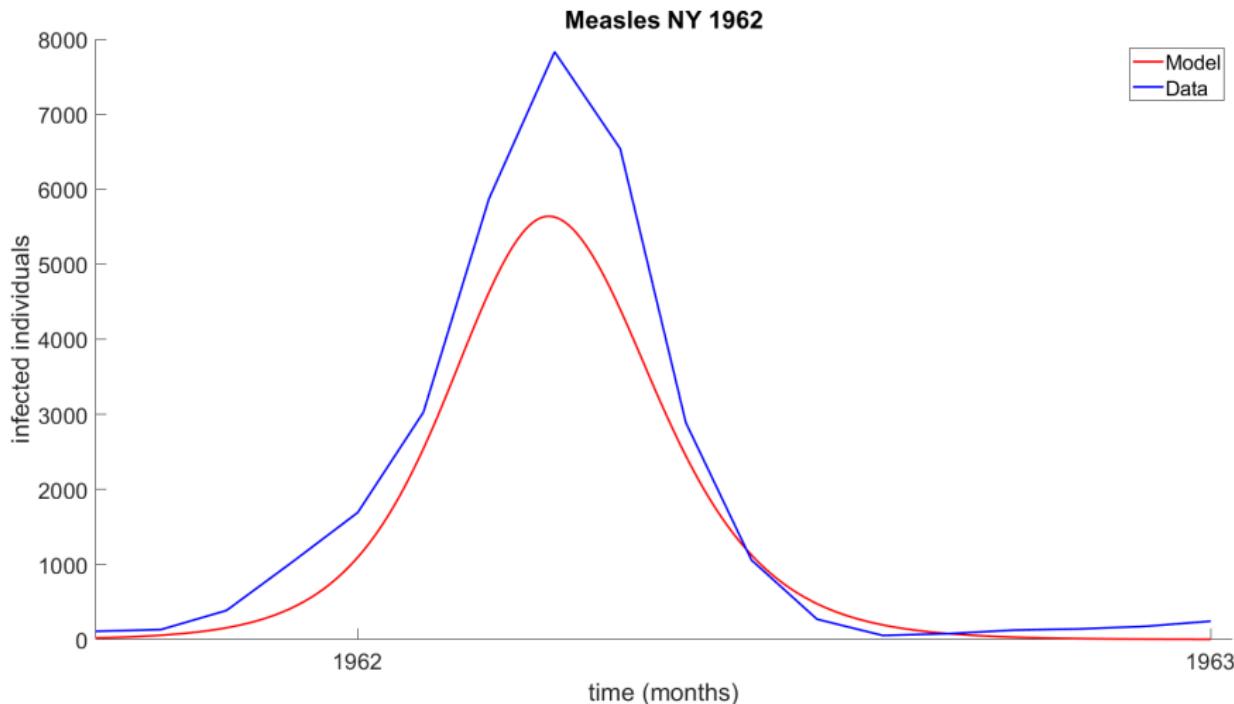


\mathcal{R}_0 and control



- “At the beginning”, $I(t) = \exp(\gamma(\mathcal{R}_0 - 1)t)$;
- Reduce $\mathcal{R}_0 \rightarrow$ “flatten the curve”;
- Control: reduce transmission \rightarrow reduce β .

Practical example of SIR: Measles, New York, 1962



F.Brauer, P. van den Driessche and J. Wu, **Mathematical Epidemiology**, Springer-Verlag, Berlin, 2008, p.5-11.
DOI: <https://doi.org/10.1007/978-3-540-78911-6>

Practical example of SIR: Measles, New York, 1962

```
-----  
+++++++ SIR model ++++++  
-----  
* population      = 7781984  
  (individuals)  
* transmission rate = 3.6  
  (days^-1)  
* recovery rate    = 0.2  
  (days^-1)  
* R_nought        = 18  
  (dimensionless)  
* S0              = 0.065*N  
  (individuals)  
* I0              = 123*5/30  
  (individuals)  
-----
```

- Simple model → does not reproduce the data perfectly;
- Qualitative reproduction of the outbreak: peak time, growth rate.



F.Brauer, P. van den Driessche and J. Wu, **Mathematical Epidemiology**, Springer-Verlag, Berlin, 2008, p.5-11.
DOI: <https://doi.org/10.1007/978-3-540-78911-6>

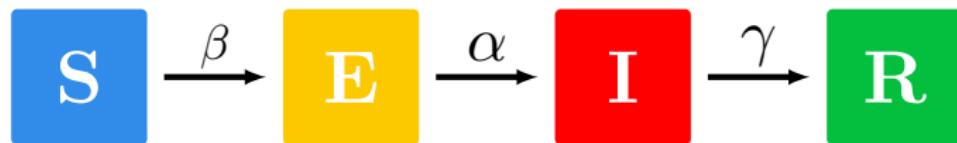
Limitations of the SIR model

- Does not consider **incubation/latency**;
- Does not consider host **mortality**;
- Does not consider **control measures**: vaccines, hospitalization, quarantine, etc;
- Does not consider **spatial** factors: assumes a homogeneously distributed population;
- Does not consider **age** differences or **biological sex** of hosts;
- Does not consider **migrations** nor **climatic effects**;
- Assumes only **one transmission via**;
- Assumes that all those infected acquire **full immunity**.

SEIR model

- **4 compartments:** susceptible (**S**), exposed (**E**), infectious (**I**), and recovered (**R**).
- A population of constant size N :

$$N = S(t) + E(t) + I(t) + R(t)$$



SEIR model

- **Susceptible**: can be infected, becoming exposed;
- **Exposed**: carry the pathogen but cannot transmit it yet; eventually they become infectious;
- **Infectious**: carry the pathogen and infect susceptible people; can recover;
- **Recovered**: got rid of the pathogen and do not transmit; have become immune;
- Transmission: direct contact of **I** with **S**.

Dynamics of the SEIR model

- Initial Value Problem, which ODEs are:

$$\frac{dS}{dt} = -\beta S \frac{I}{N} \quad (\text{susceptible rate}),$$

$$\frac{dE}{dt} = \beta S \frac{I}{N} - \alpha E \quad (\text{exposed rate}),$$

$$\frac{dI}{dt} = \alpha E - \gamma I \quad (\text{infectious rate}),$$

$$\frac{dR}{dt} = \gamma I \quad (\text{recovered rate}).$$

- Initial conditions $\xrightarrow[\text{of ODEs}]{\text{integration}}$ time series.

Parameters of the SEIR model

- N : total population (individuals);
- β : transmission rate (time^{-1});
- α : latency rate (time^{-1});
- γ : recovery rate (time^{-1}).

Basic Reproduction Number — SEIR

$$\mathcal{R}_0 = \frac{\beta}{\gamma}$$

Exposed

Exposure Moment: when the pathogen finds a susceptible.

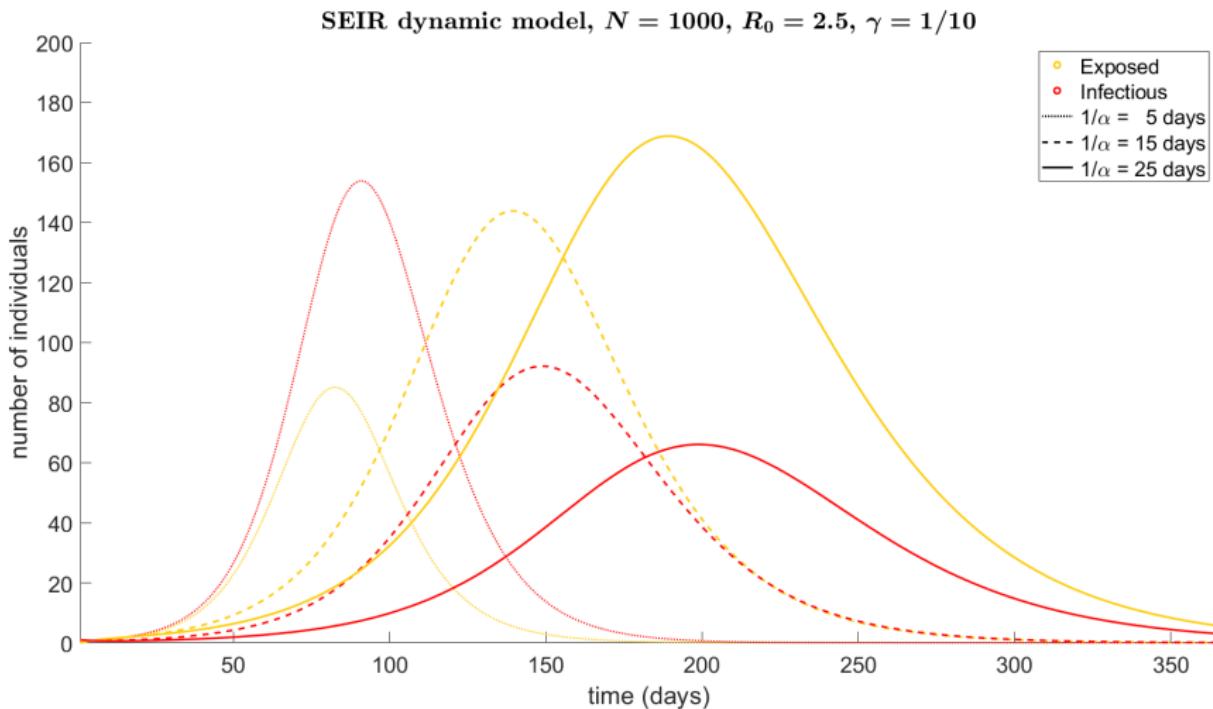
Delays (*time delays*) referring to the moment of exposure.

- Incubation period: delay until symptoms develop;
- Latency period: delay until you can transmit (**infectious**).

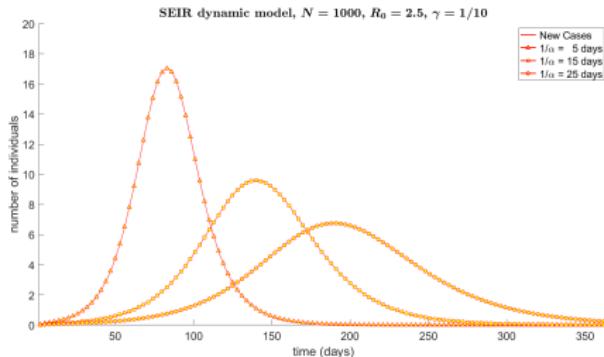
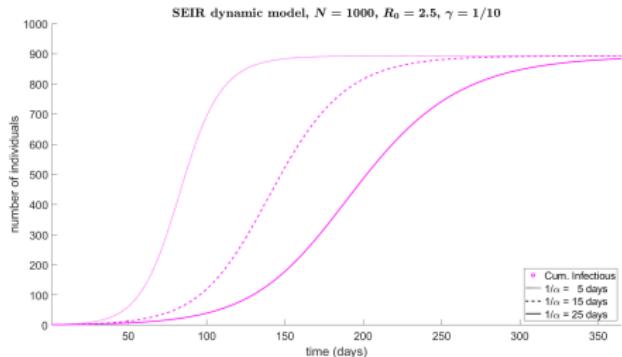
The compartment **exposed** models latency with the parameter α .

- **Other models exist!** Example: ODEs with delay, $I(t - \tau)$;
- **Limitation:** data depend on the onset of symptoms (incubation).

Latency rate α and delay



Latency rate α and delay



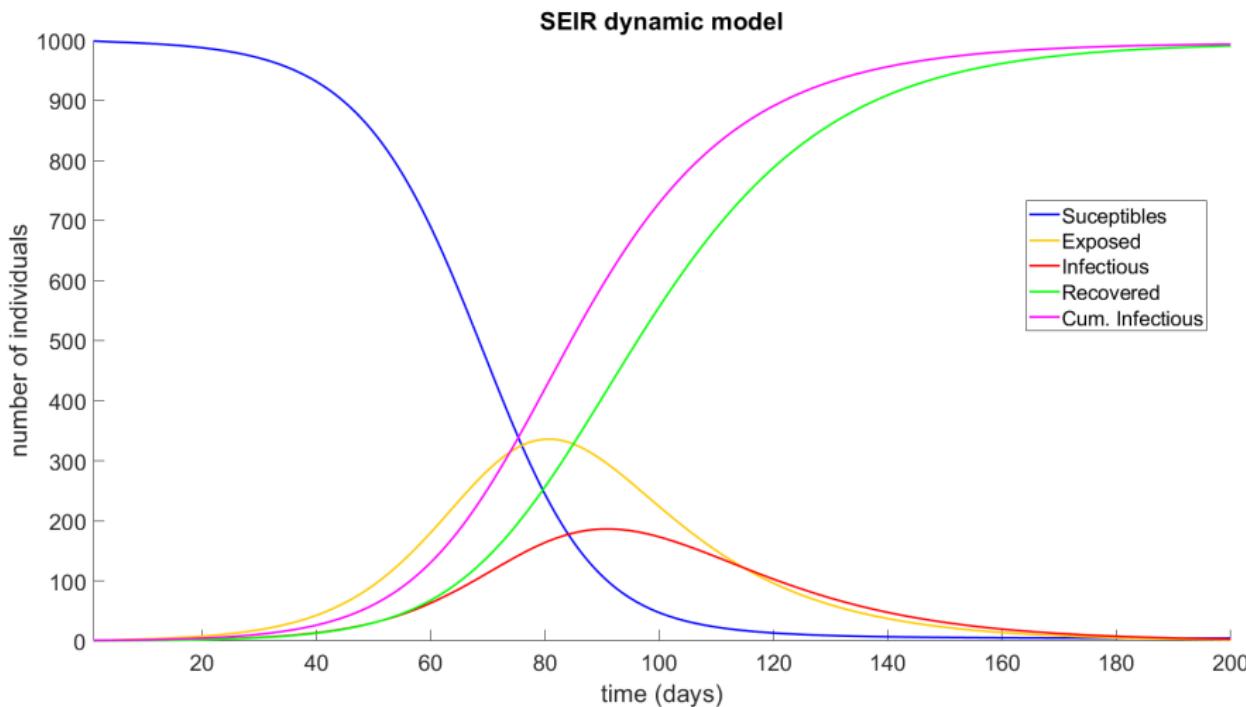
- Latency delay affects the number of infected **simultaneous**:
 - at end, even total infected (C);
 - does not affect transmission \rightarrow same R_0 ;
 - low latency \rightarrow many **infectious in the same t** (I , $NewCases$).
- **Limitation:** E very full \rightarrow difficult to detect in **real time**.

Cumulative Infectious

- Infected: had contact with the pathogen. In the SEIR: $E + I$;
- Infectious: infected able to transmit. In the SEIR: I ;
- Comparison with data → cumulative of infectious:

$$\frac{dC}{dt} = \alpha E$$

Time series



Parameters

```
-----
+-----+ SEIR model +-----+
-----
* population      = 1000
  (individuals)
* transmission rate = 0.458
  (days^-1)
* latent rate      = 0.052632
  (days^-1)
* recovery rate     = 0.083333
  (days^-1)
* R_nought         = 5.496
  (dimensionless)
-----
```

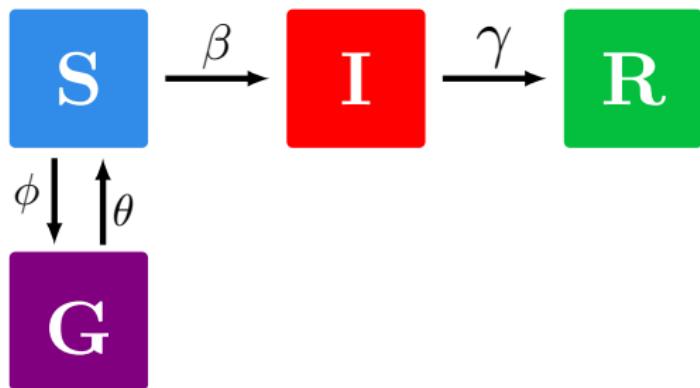
Limitations of the SEIR model

- Does not consider **incubation/latency**;
- Does not consider host **mortality**;
- Does not consider **control measures**: vaccines, hospitalization, quarantine, etc;
- Does not consider **spatial** factors: assumes a homogeneously distributed population;
- Does not consider **age** differences or **biological sex** of hosts;
- Does not consider **migrations** nor **climatic effects**;
- Assumes only **one transmission via**;
- Assumes that all those infected acquire **full immunity**.

SIRG model

- **4 compartments:** susceptible (**S**), infectious (**I**), recovered (**R**), and guarded (**G**);
- A population of constant size N :

$$N = S(t) + I(t) + R(t) + G(t)$$



SIRG model

- **Susceptible**: can be infected or quarantine;
- **Infectious**: carry the pathogen and infect susceptible people; can recover;
- **Recovered**: got rid of the pathogen and do not transmit; have become immune;
- **Guarded**: in the sense of confined. They do not become infected with the pathogen, but they can be susceptible when leaving quarantine;
- Transmission: direct contact of **I** with **S**.

Dynamics of the SIRG model

- Initial Value Problem, which ODEs are:

$$\frac{dS}{dt} = -\beta S \frac{I}{N} - \phi S + \theta G \quad (\text{susceptible rate}),$$

$$\frac{dI}{dt} = \beta S \frac{I}{N} - \gamma I \quad (\text{infectious rate}),$$

$$\frac{dR}{dt} = \gamma I \quad (\text{recovered rate}),$$

$$\frac{dG}{dt} = \phi S - \theta G \quad (\text{guarded rate}).$$

- Initial conditions $\xrightarrow[\text{das ODEs}]{\text{integration}}$ time series.

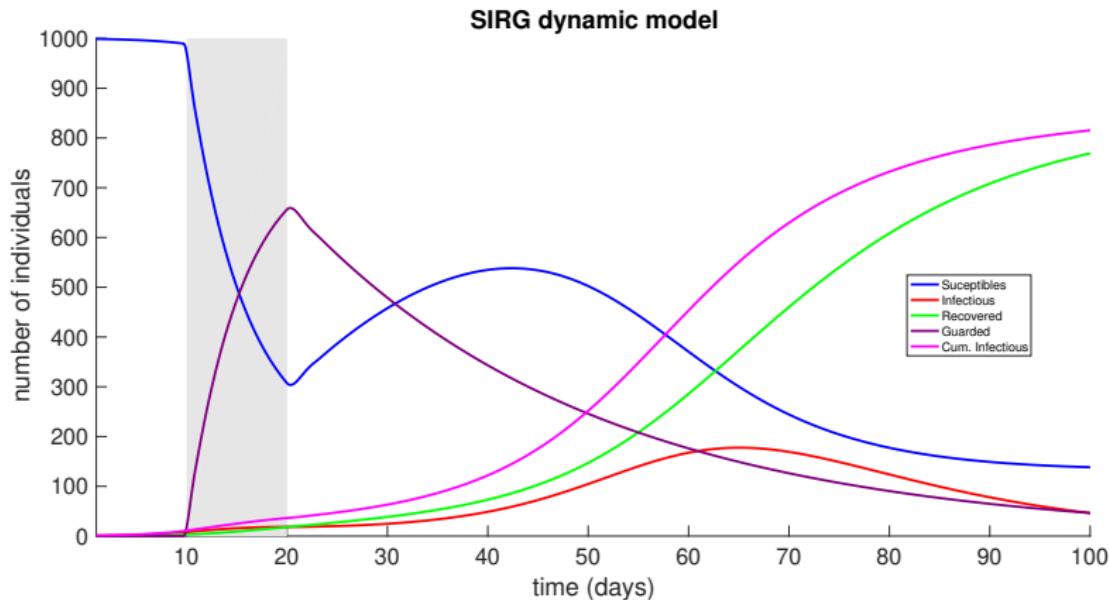
Parameters of the SIRG model

- N : total population (individuals);
- β : transmission rate (time^{-1});
- γ : recovery rate (time^{-1});
- ϕ : quarantine-in rate (time^{-1});
- θ : quarantine-out rate (time^{-1});
- t_{q0} : start of quarantine (days);
- t_{q1} : end of quarantine (days).

Basic Reproduction Number — SIRG

$$\mathcal{R}_0 = \frac{\beta}{\gamma}$$

Time series: example 1

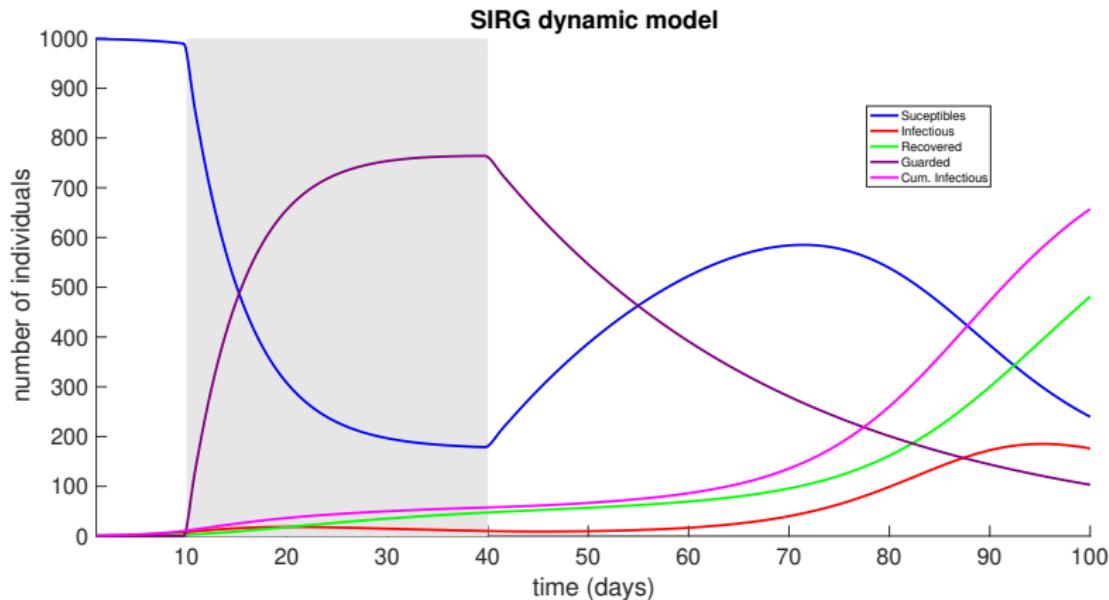


NOTE: the gray band in the graph represents the confinement period (quarantine). In this example, the quarantine starts on the 10th day and ends on the 20th day

Parameters: example 1

```
-----
++++++ SIRG model ++++++
-----
* total population      = 1000
  (individuals)
* transmission rate = 0.33333
  (days^-1)
* recovery rate       = 0.1
  (days^-1)
* quarantine-in rate = 0.14286
  (days^-1)
* quarantine-out rate = 0.033333
  (days^-1)
* quarantine start    = 10
  (days)
* quarantine end      = 20
  (days)
* R_nought            = 3.3333
  (adimensional)
-----
```

Time series: example 2

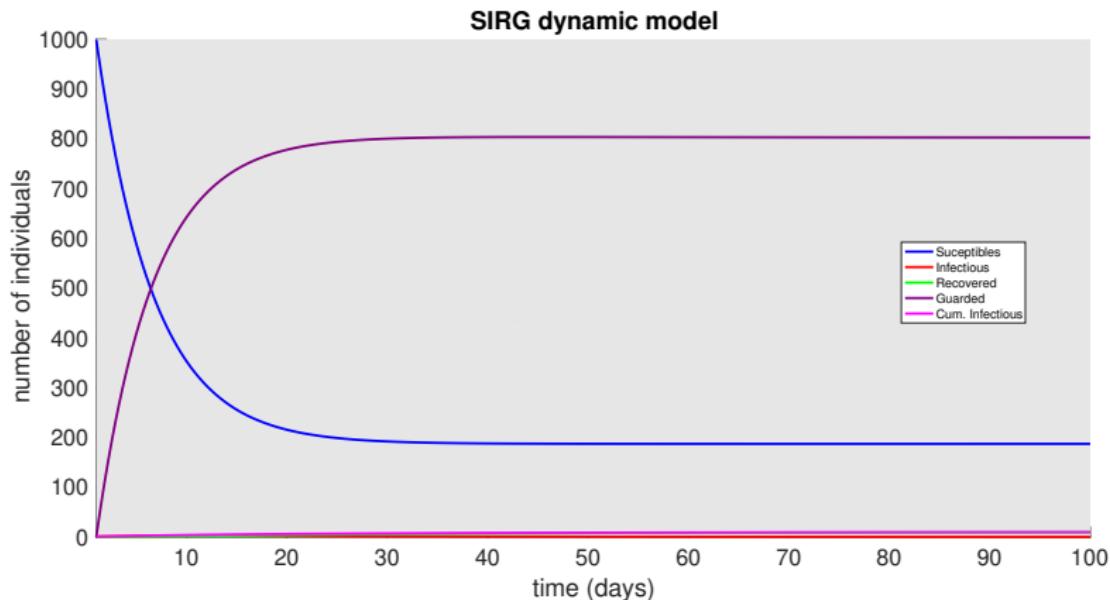


In this example, the quarantine starts on the 10th day and ends on the 40th day. Compared to the previous example, it can be seen that the infected curve has a delay in its growth.

Parameters: example 2

```
-----  
++++++ SIRG model ++++++  
-----  
* total population      = 1000  
  (individuals)  
* transmission rate = 0.33333  
  (days^-1)  
* recovery rate       = 0.1  
  (days^-1)  
* quarantine-in rate = 0.14286  
  (days^-1)  
* quarantine-out rate = 0.033333  
  (days^-1)  
* quarantine start    = 10  
  (days)  
* quarantine end      = 40  
  (days)  
* R_nought            = 3.3333  
  (adimensional)  
-----
```

Time series: example 3



In this case, quarantine occurs throughout the time interval.

Parameters: example 3

```
-----  
++++++ SIRG model ++++++  
-----
```

```
* total population      = 1000  
  (individuals)  
* transmission rate = 0.33333  
  (days^-1)  
* recovery rate       = 0.1  
  (days^-1)  
* quarantine-in rate = 0.14286  
  (days^-1)  
* quarantine-out rate = 0.033333  
  (days^-1)  
* quarantine start    = 0  
  (days)  
* quarantine end      = 101  
  (days)  
* R_nought            = 3.3333  
  (adimensional)
```

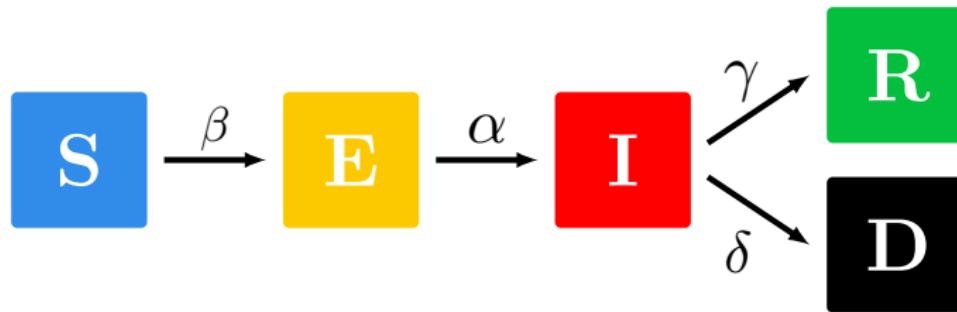
Limitations of the SIRG model

- Does not consider **incubation/latency**;
- Does not consider host **mortality**;
- Does not consider **control measures**: vaccines, hospitalization, **quarantine**, etc;
- Does not consider **spatial** factors: assumes a homogeneously distributed population;
- Does not consider **age** differences or **biological sex** of hosts;
- Does not consider **migrations** nor **climatic effects**;
- Assumes only **one transmission via**;
- Assumes that all those infected acquire **full immunity**.

SEIRD model

- **5 compartments:** susceptible (**S**), exposed (**E**), infectious (**I**), recovered (**R**), and deceased (**D**).
- A population of initial size N_0 :

$$N_0 = S(t) + E(t) + I(t) + R(t) + D(t)$$



SEIRD model

- **Susceptible**: can be infected, becoming exposed;
- **Exposed**: carry the pathogen but cannot transmit it yet; eventually they become infectious;
- **Infectious**: carry the pathogen and infect susceptible people; may recover or die due to the pathogen;
- **Recovered**: got rid of the pathogen and do not transmit; have become immune;
- **Deceased**: died because of the pathogen;
- Transmission: direct contact between **I** and **S**.

Dynamics of the SEIRD model

- **Initial Value Problem**, which ODEs are:

$$\frac{dS}{dt} = -\beta S \frac{I}{N}, \quad \frac{dE}{dt} = \beta S \frac{I}{N} - \alpha E,$$

$$\frac{dI}{dt} = \alpha E - (\gamma + \delta) I, \quad \frac{dR}{dt} = \gamma I,$$

$$\frac{dD}{dt} = \delta I,$$

where $N(t) = N_0 - D(t)$;

- Initial conditions $\xrightarrow[\text{of the ODEs}]{\text{integration}}$ time series.

Parameters of the SEIRD model

- N_0 : initial population (individuals);
- β : transmission rate (time^{-1});
- α : latency rate (time^{-1});
- γ : recovery rate (time^{-1});
- δ : mortality rate (time^{-1}).

Basic Reproduction Number — SEIRD

$$\mathcal{R}_0 = \frac{\beta}{\gamma + \delta}$$

Deceased

The compartment **deceased** accumulate how many died until the time t due to the pathogen.

- Initial population N_0 : constant; distributed between compartments;

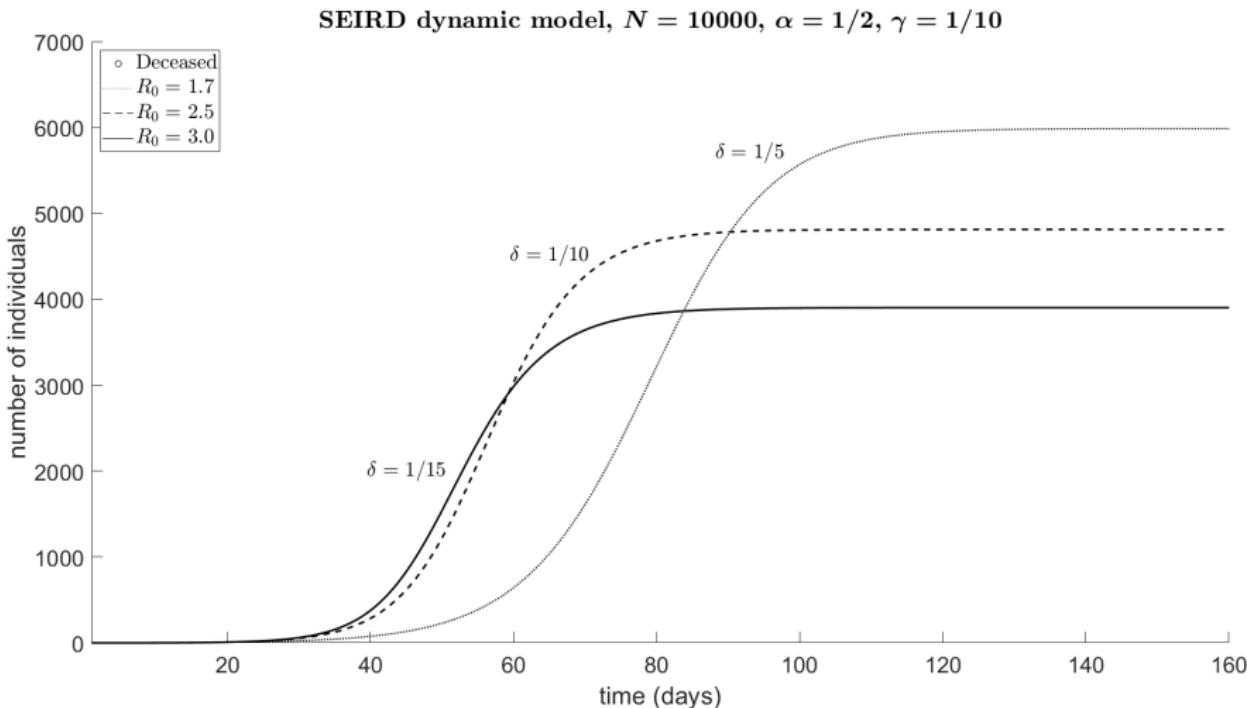
$$N_0 = S(t) + E(t) + I(t) + R(t) + D(t)$$

- Living population N : relevant to the term of the force of infection;

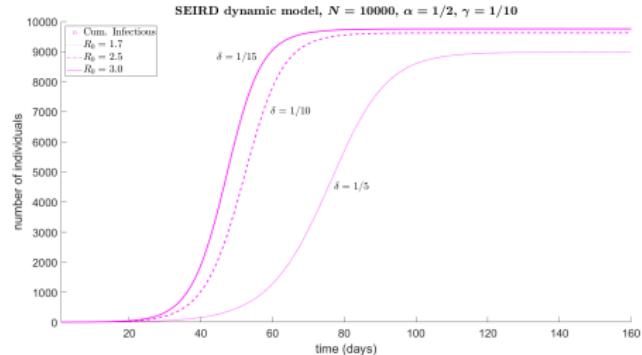
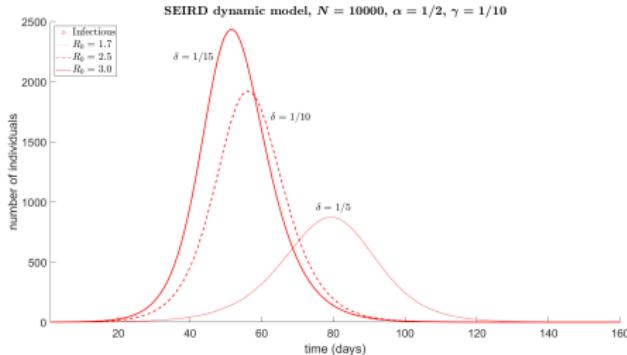
$$N(t) = S(t) + E(t) + I(t) + R(t) \rightarrow N(t) = N_0 - D(t)$$

- In general, epidemiological measures regarding the **deceased** are more abundant (**lethality, mortality**) \rightarrow comparison with data.

Mortality rate δ



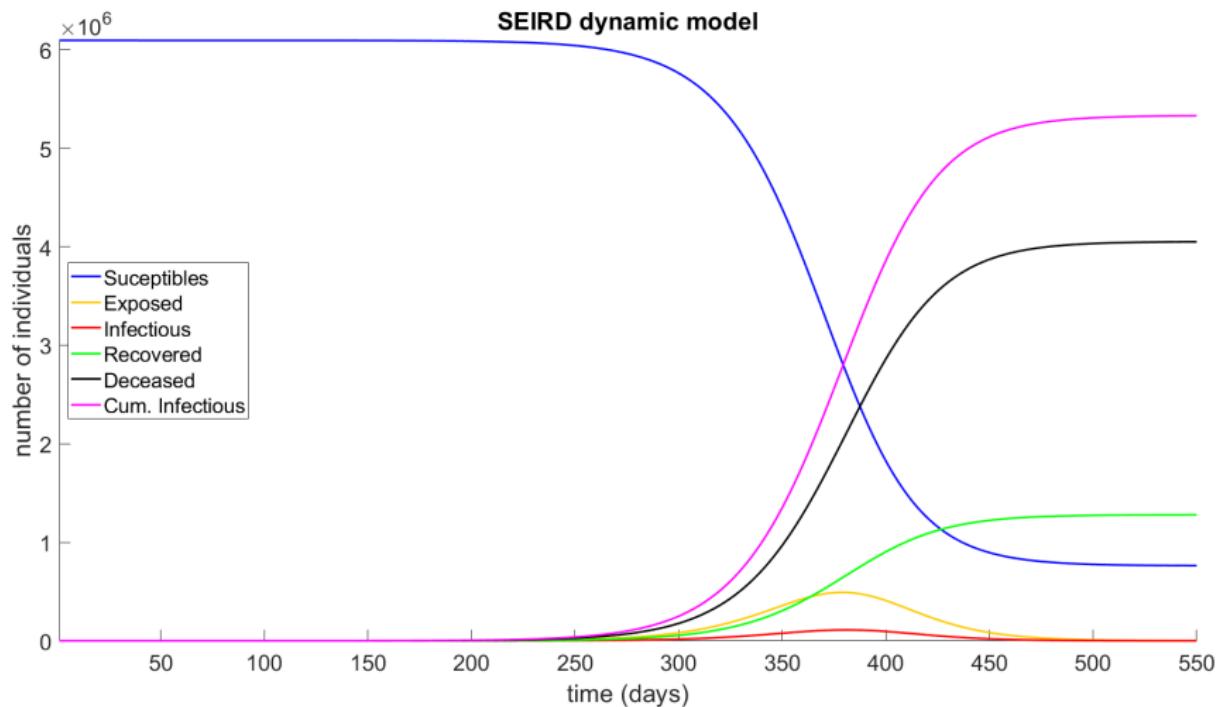
Mortality rate δ



- Mortality matters for transmission:

- at end, impacts the total number of infected C ;
- very lethal \rightarrow reduces live infectious **in the same t** (I , $NewCases$);
- high mortality rate \rightarrow lowest mean time in $I \rightarrow$ smaller \mathcal{R}_0 .

Time series



Parameters

```
-----  
+++++ SEIRD model +++++  
-----  
* initial population = 6092000  
  (individuals)  
* transmission rate   = 0.7124  
  (days^-1)  
* latent rate         = 0.11111  
  (days^-1)  
* recovery rate       = 0.11832  
  (days^-1)  
* death rate          = 0.37468  
  (days^-1)  
* R_nought            = 1.445  
  (dimensionless)  
-----
```

Some parameters from the 2014 Ebola epidemic in Sierra Leone.



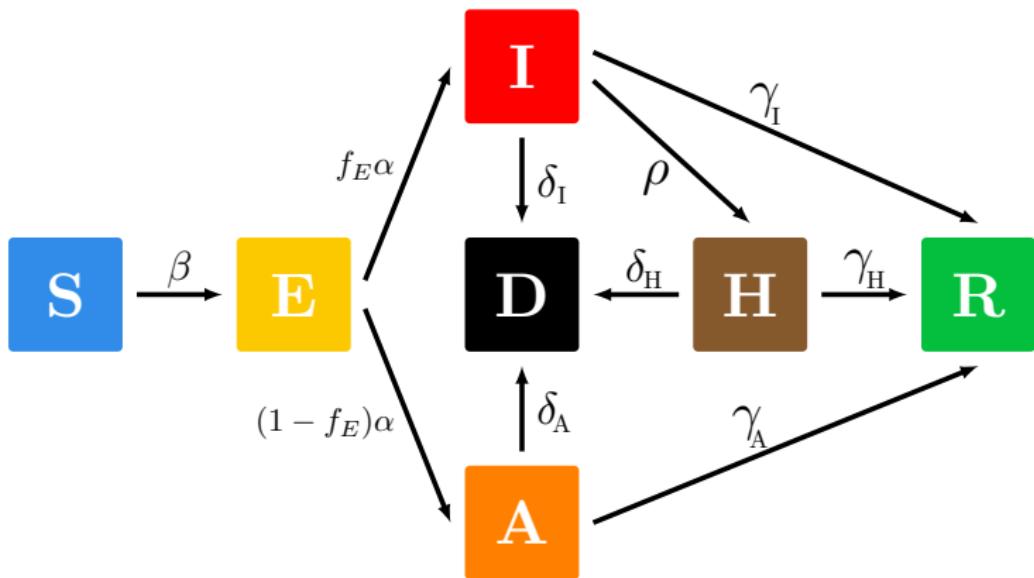
P. Diaz, P. Constantine, K. Kalmbach, E. Jones and S. Pankavichm. **A modified SEIR model for the spread of Ebola in Western Africa and metrics for resource allocation**, Applied Mathematics and Computation, 324(1): 141-155, 2018.
DOI: <https://doi.org/10.1016/j.amc.2017.11.039>

Limitations of the SEIRD model

- Does not consider **incubation/latency**;
- Does not consider host **mortality**;
- Does not consider **control measures**: vaccines, hospitalization, quarantine, etc;
- Does not consider **spatial** factors: assumes a homogeneously distributed population;
- Does not consider **age** differences or **biological sex** of hosts;
- Does not consider **migrations** nor **climatic effects**;
- Assumes only **one transmission via**;
- Assumes that all those infected acquire **full immunity**.

SEIR(+AHD) model

- 7 compartments: susceptible (S), exposed (E), symptomatic infectious (I), recovered (R), asymptomatic infectious (A), hospitalized (H), and deceased (D).



SEIR(+AHD) model

- A population of initial size N_0 :

$$N_0 = S(t) + E(t) + I(t) + A(t) + H(t) + R(t) + D(t)$$

- **Susceptible**: they can be infected, becoming exposed;
- **Exposed**: they carry the pathogen but cannot transmit it yet; eventually they become infectious;
- **Symptomatic infectious**: they carry the pathogen and infect susceptible people; they have symptoms; they may recover, be hospitalized or die due to the pathogen;
- **Asymptomatic infectious**: they carry the pathogen and infect susceptible people; they have no symptoms; they may recover or die due to the pathogen.

SEIR(+AHD) model

- **Hospitalized:** they carry the pathogen and infect susceptible people; they may recover or die due to the pathogen;
- **Recovered:** they got rid of the pathogen and do not transmit; they have become immune;
- **Deceased:** they died because of the pathogen;
- Transmission: direct contact from *S* with *I* or *A* or *H*.

Dynamics of the SEIR(+AHD) model

- Initial Value Problem, which ODEs are:

$$\frac{dS}{dt} = -\beta S (I + A + H) / N,$$

$$\frac{dE}{dt} = \beta S (I + A + H) / N - \alpha E,$$

$$\frac{dI}{dt} = f_E \alpha E - (\gamma + \rho + \delta) I,$$

$$\frac{dA}{dt} = (1 - f_E) \alpha E - (\gamma + \delta) A,$$

$$\frac{dH}{dt} = \rho I - (\gamma + \kappa_H \delta) H,$$

$$\frac{dR}{dt} = \gamma (I + A + H),$$

$$\frac{dD}{dt} = \delta (I + A + \kappa_H H), \text{ where } N(t) = N_0 - D(t).$$

- Initial conditions $\xrightarrow[\text{of the ODEs}]{\text{integration}}$ time series.

Parameters of the SEIR(+AHD) model

- N_0 : initial population (individuals);
- β : transmission rate (time^{-1});
- α : latency rate (time^{-1});
- f_E : symptomatic fraction (dimensionless);
- γ : recovery rate (time^{-1});
- ρ : hospitalization rate (time^{-1});
- δ : mortality rate (time^{-1});
- κ_H : mortality factor during hospitalization (dimensionless).

Symptoms

The compartment **asymptomatic infectious** models individuals who can transmit the pathogen even **without symptoms**.

- Parameter f_E ($0 < f_E < 1$): fraction of infectious **with symptoms**;
- **Limitation:** many asymptomatic → difficult accurate detection;
- **Other models exist!**
 - More compartments: according to type and level of symptoms;
 - More parameters: different transmission rates for **symptomatic** e **asymptomatic**.

Hospitalized

The compartment hospitalized models **symptomatic infectious** individuals that are in hospital treatment.

- Parameter ρ : models how much the **symptomatic** are **hospitalized**;
- Parameter κ_H ($0 < \kappa_H < 1$): models the reduction in mortality of the **infectious** due to being **hospitalized**;
- Other models exist!
 - More compartments: quarantine, ICU, etc;
 - More parameters: diagnostic rate, positive test fraction, etc.
- Hospitalized → **hospital beds and tests** → comparison with data.

Basic reproduction number and control

Hospitalization of the **infectious** is a strategy to control the epidemic.

A dimensionless number analogous to \mathcal{R}_0 can be defined by assuming that the control strategy has been active since $S \approx N$.

Control Reproduction Number — SEIR(+AHD)

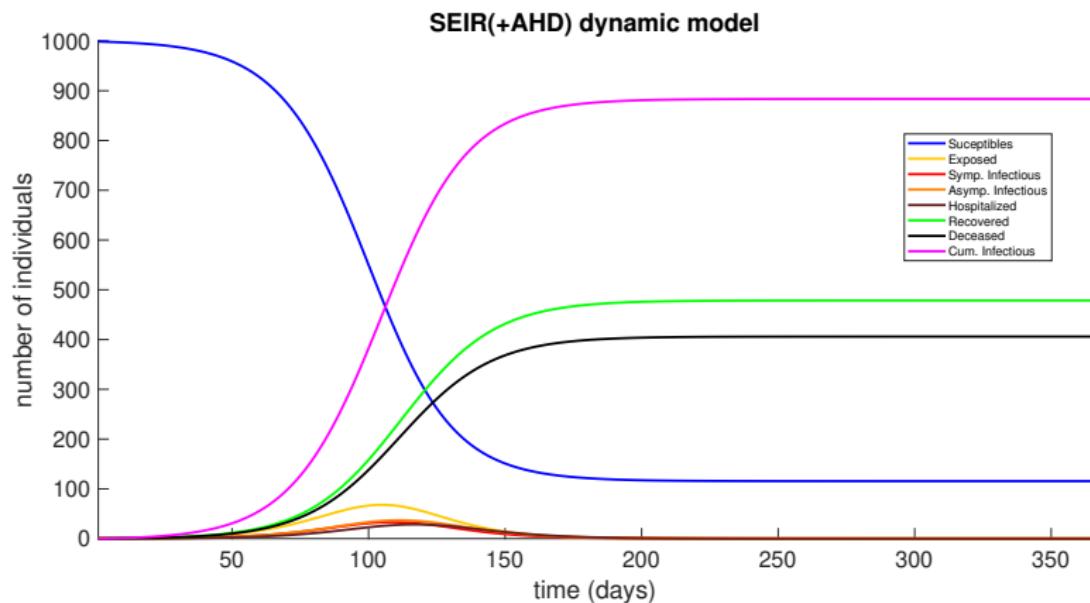
$$\mathcal{R}_C = \frac{f_E \beta}{\gamma + \rho + \delta} + \frac{(1 - f_E) \beta}{\gamma + \delta} + \frac{\rho f_E \beta}{(\gamma + \rho + \delta)(\gamma + \kappa_H \delta)}$$

In the absence of delay or control strategies ($\rho = \kappa_H = 0$), the SEIR(+AHD) model has a \mathcal{R}_0 conventional.

Basic reproduction number — SEIR(+AHD)

$$\mathcal{R}_0 = \frac{\beta}{\gamma + \delta}$$

Time series



Parameters

```
-----  
+++++ SEIAHRD model +++++  
-----  
* initial population      = 1000  
  (individuals)  
* transmission rate       = 0.25  
  (days^-1)  
* latent rate             = 0.2  
  (days^-1)  
* symptomatic fraction    = 0.6  
  (dimensionless)  
* recovery rate           = 0.071429  
  (days^-1)  
* hospitalization rate    = 0.1  
  (days^-1)  
* death rate              = 0.07  
  (days^-1)  
* hospitalization recovery = 0.55  
  (dimensionless)  
* R_nought                = 1.7677  
  (dimensionless)  
* R_control               = 1.8936  
  (dimensionless)
```

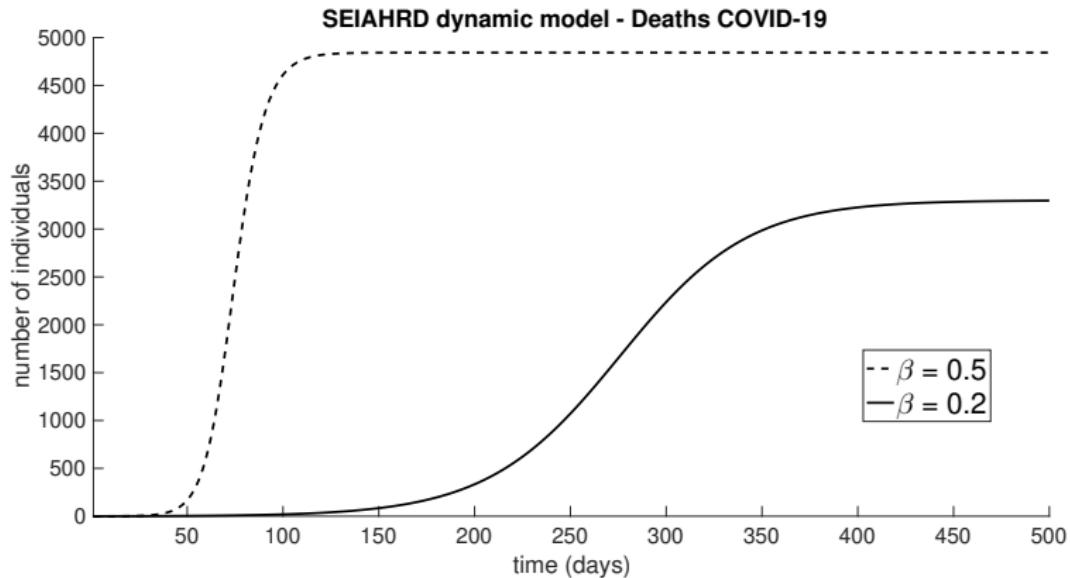
Some parameters used in the Sars-CoV-2 epidemic (2020) in Brazil.



W. Lyra, J. Nascimento, J. Belkhiria, L. de Almeida, P. P. Chrispim and I. Andrade. **COVID-19 pandemics modeling with SEIR(+CAQH), social distancing, and age stratification. The effect of vertical confinement and release in Brazil.**, medRxiv, 2020. DOI: <https://doi.org/10.1101/2020.04.09.20060053>.

Comparison of deaths for COVID-19 compared to β

- $N = 10.000$ individuals;
- For $\beta = 0,5 \Rightarrow \mathcal{R}_0 = 7$ and for $\beta = 0,2 \Rightarrow \mathcal{R}_0 = 2,8$.



Limitations of the SEIR(+AHD) model

- Does not consider **incubation/latency**;
- Does not consider host **mortality**;
- Does not consider **control measures**: vaccines, **hospitalization**, quarantine, etc;
- Does not consider **spatial** factors: assumes a homogeneously distributed population;
- Does not consider **age** differences or **biological sex** of hosts;
- Does not consider **migrations** nor **climatic effects**;
- Assumes only **one transmission via**;
- Assumes that all those infected acquire **full immunity**.

Computational code files

- The codes of the models of module Modeling are in the directory [EPIDEMIC-master\modeling](#);
- For each model, the EPIDEMIC uses two files. Example, SIR:
 - **[main_SIR.m]** Main file. Defines the parameters, initial conditions, and time interval for the integration of the ODEs system. It also calculates reproduction numbers and plots the results of the time series.
 - **[rhs_SIR.m]** Defines the ODEs system used by the main file.

File main: definition of parameters

The user defines the parameters:

```
% parameters and initial conditions [USER INPUT]
%
% population size (number of individuals)
N = 1000;

% transmission rate (days^-1)
beta = 1/4;

% recovery period (days)
Tgamma = 10;

% recovery rate (days^-1)
gamma = 1/Tgamma;
```

For convenience, some parameters are defined through their inverses, for example, gamma:

$$\gamma : \text{recovery rate} = (\text{recovery period})^{-1}$$

File main: initial conditions

The user defines the initial conditions (ICs).

- The ICs are the compartment values at the time t_0 ;
- Comment: suggestion for ICs at the initiation (ideal) of the epidemic;
- To compute the cumulative of infected, C_0 is also required.

```
% initial conditions
%
% — Set the initial number of infected.
% — The number of susceptible will be the remaining population.
% — For an invasion scenario, set initial infected to 1.

R0 = 0;           % initial recovered   (number of individuals)
I0 = 1;           % initial infected    (number of individuals)
S0 = N-I0-R0;    % initial susceptible (number of individuals)

% initial cumulative infected (number of individuals)
C0 = I0;
%
```

File main: \mathcal{R}_0 and display

The file automatically calculates the basic reproduction number. The \mathcal{R}_0 and the defined parameters are displayed on the command screen after a header of the EPIDEMIC.

```
% computing the basic reproduction number R_nought
R_nought = beta/gamma;

disp(' ')
disp('_____')
disp(' EPIDEMIC – Epidemiology Educational Code ')
disp(' by A. Cunha, E. Dantas, et al. ')
disp(' ')
disp(' This is an easy to run educational toolkit ')
disp(' for epidemiological analysis. ')
disp(' ')
disp(' www.EpidemicCode.org ')
disp('_____')
```

In each model, different quantities that depend on the parameters are computed and exposed in that part.

File main: integration

The user can change the details of the integration process.

- The vectors `param` and `IC` gather the parameters and initial conditions for the file **rhs**. Pay attention to the order in these vectors;
- The vector `tspan` sets the analysis interval according to the initial time `t0`, final time `t1` and the step `dt`.

```
% parameters vector
param = [N beta gamma];

% initial conditions vector
IC = [S0 I0 R0 C0];

% time interval of analysis
t0 = 1; % initial time (days)
t1 = 365; % final time (days)
dt = 0.1; % time steps (days)
tspan = t0:dt:t1; % interval of analysis
Ndt = length(tspan); % number of time steps
```

File main: integration

The solver `ode45` uses the file `rhs` as a function to evaluate derivatives at different times.

- Parameters and initial conditions are informed by `param` and `IC`;
- The Runge-Kutta 4(5) method is employed by the solver for numerical integration in the interval `tspan`;
- Output: the matrix `y` provides the time series.

```
% ODE solver Runge-Kutta45
[time, y] = ode45(@(t,y)rhs_SIR(t,y,param),tspan,IC);

% time series
S = y(:,1); % susceptible (number of individuals)
I = y(:,2); % infected (number of individuals)
R = y(:,3); % recovered (number of individuals)
C = y(:,4); % cumulative infected (number of individuals)
```

For convenience, time series are separated into vectors.

File main: post-processing

The amount *NewCases* is computed for day intervals.

```
% NewCases ( per day) computation
tu = 1; % time unit (days)
%
% --- tu/dt must be an integer
% --- tu = 1 defines a 'per day' computation
% --- For a 'per week' computation, use tu = 7 and t0 = 7
NewCases = zeros(floor((Ndt-1)/(tu/dt)),1);
for n = 1:length(NewCases)
    NewCases(n) = C((n)*tu/dt +1) - C((n-1)*tu/dt +1);
end
```

Graphics are generated with time series and *NewCases* (discreet).

```
% plot all compartments of SIR model
figure(1)
hold on
fig1(1) = plot(time,S);
fig1(2) = plot(time,I);
fig1(3) = plot(time,R);
fig1(4) = plot(time,C);
hold off

% plot NewCases (per day) of SIR model
figure(2)
fig2 = scatter([1:length(NewCases)]+1,NewCases);
```

The others lines contain details for personalizing the figures.

File **rhs**: input and output

Defines a *function* for the GNU Octave\ MATLAB with the inputs:

- t: time vector;
- y: states vector;
- param: set of parameters.

```
function dydt = rhs_SIR(t,y,param)
```

The output of *function* is the vector dydt that informs the derivatives each time.

File rhs: checks

The file also checks the parameters entered by the user:

```
if length(param) < 3
    error('Warning: To few model parameters')
elseif length(param) > 3
    error('Warning: To many model parameters')
end

if mod(param(1),1) ~= 0
    error('Warning: Use a integer population size')
end

if sum(param<0)~=0
    error('Warning: Use positive parameters values')
end
```

File rhs: input arguments

The arguments t and y are automatically organized by ode45 at the main file, according to:

- vector $tspan \rightarrow t$;
- vector $IC \rightarrow y$.

The argument `param` is manually adjusted in the main file to be opened in the file `rhs`. It is important to pay attention to the order in the vector.

```
% model parameters: param = [N beta gamma]
N      = param(1); % population size    (number of individuals)
beta   = param(2); % transmission rate (days^-1)
gamma  = param(3); % recovery rate     (days^-1)
```

File rhs: differential equations

The main part of the file defines the differential equations.

```
% SIR dynamic model:  
%  
%     y = [S I R C]           is the state vector  
%     dydt = [dSdt dIdt dRdt dCdt] is the evolution law  
%  
% dSdt - rate of susceptible      (number of individuals/days)  
% dIdt - rate of infected        (number of individuals/days)  
% dRdt - rate of recovered       (number of individuals/days)  
% dCdt - rate of cumulative infected (number of individuals/days)  
  
[S I R C] = deal(y(1),y(2),y(3),y(4));  
  
dSdt = - beta*S.* (I/N);  
dIdt = beta*S.* (I/N) - gamma*I;  
dRdt = gamma*I;  
dCdt = beta*S.* (I/N);  
dydt = [dSdt; dIdt; dRdt; dCdt];
```

The vector order IC in the main file is important for the state vector y and to the output $dydt$ of the *function*.

Customization (SIR → SEIR)

Compartmental models can be easily modified.

Example: SIR → SEIR

- **Change the ODEs on the rhs.** Add the compartment E . Adjust the input (positive) and output (negative) terms in the compartments.

```
[S I R C] = deal(y(1),y(2),y(3),y(4));
```

```
dSdt = - beta*S.* (1/N);  
dIdt = beta*S.* (1/N) - gamma*I;  
dRdt = gamma*I;  
dCdt = beta*S.* (1/N);  
dydt = [dSdt; dIdt; dRdt; dCdt];
```

```
[S E I R C] = deal(y(1),y(2),y(3),y(4),y(5));
```

```
dSdt = - beta*S.* (1/N);  
dEdt = beta*S.* (1/N) - alpha*E;  
dIdt = alpha*E - gamma*I;  
dRdt = gamma*I;  
dCdt = alpha*E;  
dydt = [dSdt; dEdt; dIdt; dRdt; dCdt];
```

Note that C is changed to continue checking the input of the I .

Make sure the order in the vector y and $dydt$ is coinciding.

Customization (SIR → SEIR)

- **Change the parameters and ICs in the main and rhs.** Modify the vectors param and IC in the main. Make sure that the order in the vectors matches between the files.

```
% parameters vector  
param = [N beta gamma];  
  
% initial conditions vector  
IC = [S0 I0 R0 C0];
```

```
% parameters vector  
param = [N beta alpha gamma];  
  
% initial conditions vector  
IC = [S0 E0 I0 R0 C0];
```

- **Separate the time series.** Adjust the time series separation according to the order of the vector IC.

```
% time series  
S = y(:,1);  
I = y(:,2);  
R = y(:,3);  
C = y(:,4);
```

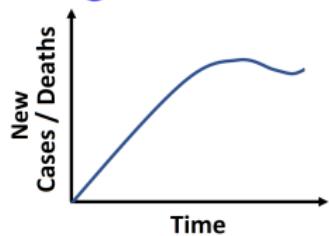
```
% time series  
S = y(:,1);  
E = y(:,2);  
I = y(:,3);  
R = y(:,4);  
C = y(:,5);
```

Trends

Monitoring

The analysis of trends in epidemiology aims to monitor the behavior of an epidemic in a certain geographic region (country, state, city, etc.). In EPIDEMIC, these analyzes are performed using two basic visualization strategies:

Contagion or Mortality



Epidemic progress



- Evaluates the temporal evolution of the numbers of infected or dead. Evaluates the incidence as a function of time.
- Evaluates the rate of change of the epidemic in relation to the total number of cases or deaths.

Graphs specificities

Each monitoring graph generated by the code provides different types of analysis. For this reason, understanding the context in which the code is applied, as well as the graphic representation generated by it, is of great importance to obtain the desired information.

Time analysis

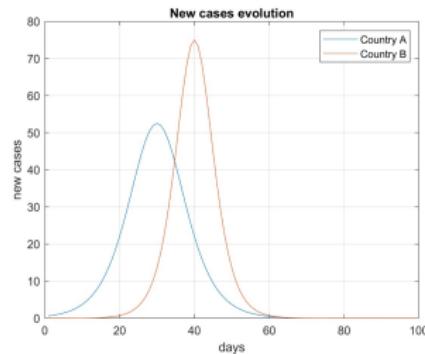
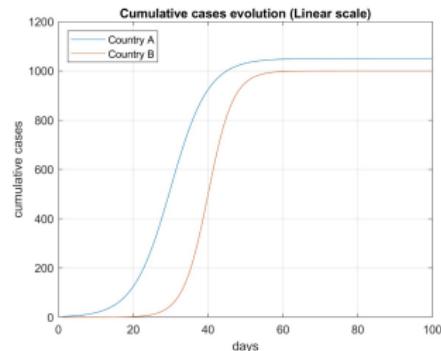
Graphical analyzes based on time are made in relation to the number of days since a "X" number of cases or deaths was recorded. Thus, it is possible to perform a temporal comparison of the incidence of the disease between different locations, even if it occurred in different periods.

Relative incidence

The assessment of the proportion of infected people in a place of interest is done through the relative incidence (RI) of the disease. The same curves of the temporal analysis are used, but with the number of cases proportional to the population, per million (or one hundred thousand) inhabitants. This makes it possible to compare the impact of the epidemic in each region in relative terms.

The graphs and their analysis

Observing two countries, hypothetical, after a certain epidemic; the curves (to the right) of these countries show that country "B" had a more pronounced growth of cases in a short period of time when compared to country "A". Additionally, the curves give the impression that country "B" presented more cases than country "A".



However, by analyzing the graph on the left, country "B" had a slightly lower number of cases than country "A". Therefore, **the desired information is closely related to the type of graph analyzed**, in this case, if you want to know which country has more cases, the graph with the accumulated value is more useful.



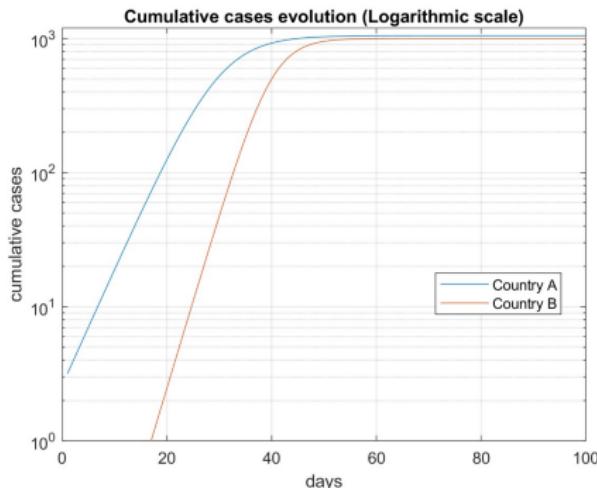
A. Cunha Jr et al. Relatório 01 Progresso da COVID-19 no Brasil e no Estado do Rio de Janeiro 21^a Semana Epidemiológica do Calendário 2020 (17/05/2020) até (23/05/2020). COVID-19: Observatório Fluminense, (24/05/2020). DOI: <https://doi.org/10.12957/eduerj.covid19rj.relatorio1>

Use of the logarithmic scale

During the epidemic, while the curves are still being constructed, how do you know if the peak is near?

One way to obtain evidence about the peak of the epidemic is to generate graphs using **logarithmic scale**.

On the logarithmic scale, exponential growth appears as a growing “line”. Thus, it is easier to observe when the growth rate has fallen, as can be seen in the graph on the side.



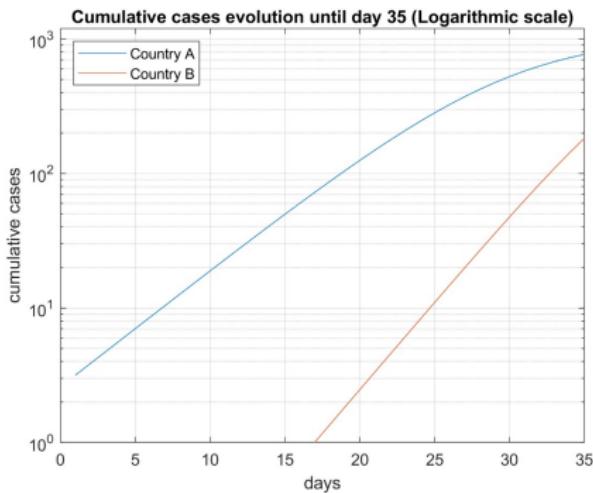
A. Cunha Jr et al. Relatório 01 Progresso da COVID-19 no Brasil e no Estado do Rio de Janeiro 21ª Semana Epidemiológica do Calendário 2020 (17/05/2020) até (23/05/2020). COVID-19: Observatório Fluminense, (24/05/2020). DOI: <https://doi.org/10.12957/eduerj.covid19rj.relatorio1>

Use of the logarithmic scale

During the epidemic, while the curves are still being constructed, how do you know if the peak is near?

Observing the graph on the side, of the scenarios of countries "A" and "B" until the 35° day of analysis.

It is possible to verify the exponential behavior of the curves most of the time, however the curve of the country "A", after 25° day, has a certain slope/curvature. This fact points to the stabilization of cases, in addition to indicating that the peak of the epidemic may be near.



*It should be noted that these are hypothetical curves, well-behaved and generated without noise.



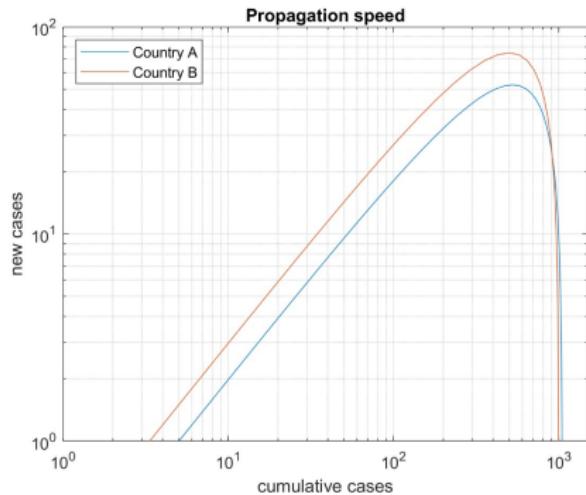
A. Cunha Jr et al. Relatório 01 Progresso da COVID-19 no Brasil e no Estado do Rio de Janeiro 21^a Semana Epidemiológica do Calendário 2020 (17/05/2020) até (23/05/2020). COVID-19: Observatório Fluminense, (24/05/2020). DOI: <https://doi.org/10.12957/eduerj.covid19rj.relatorio1>

Change in the progress of the epidemic

A change in the progress of the epidemic can best be assessed in the graph of the number of new cases as a function of the total number of cases.

In this graph, the horizontal axis corresponds to the total number of cases, so time is implicit in the assessment.*

It is possible to see the change in behavior in both curves, that is, the “fall”.



*Not explaining the time allows to compare the progress of the epidemic in different places at different times.

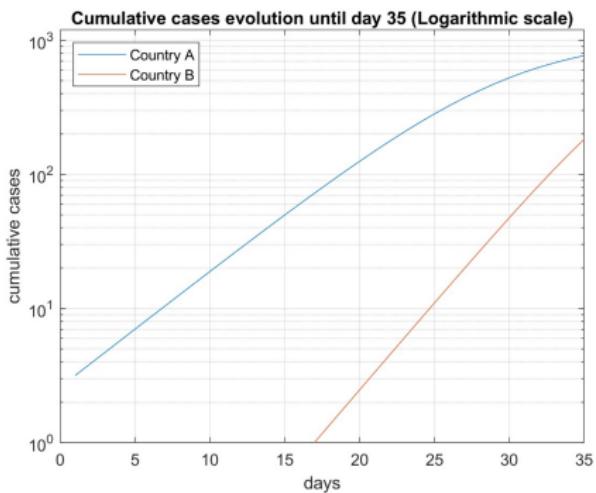


A. Cunha Jr et al. Relatório 01 Progresso da COVID-19 no Brasil e no Estado do Rio de Janeiro 21^a Semana Epidemiológica do Calendário 2020 (17/05/2020) até (23/05/2020). COVID-19: Observatório Fluminense, (24/05/2020). DOI: <https://doi.org/10.12957/eduerj.covid19rj.relatorio1>

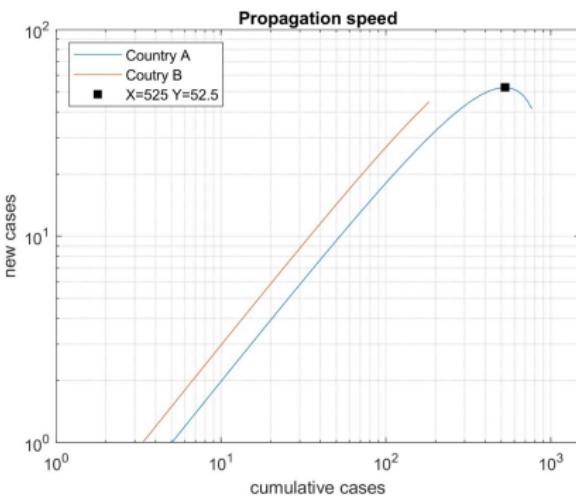
Change in the progress of the epidemic

To represent the progress of the epidemic when complete data is not available, the graph with the **propagation speed** is an alternative for understanding the evolution/trend of the outbreak through new cases. What is not possible to observe in the **time graph**.

Temporal Graph



"Speed" Graph

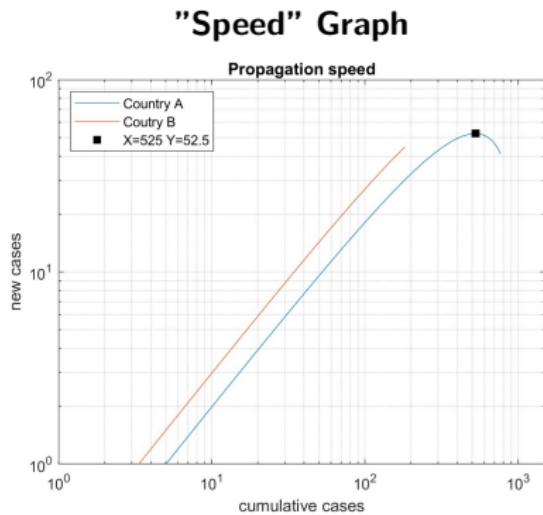


A. Cunha Jr et al. Relatório 01 Progresso da COVID-19 no Brasil e no Estado do Rio de Janeiro 21^a Semana Epidemiológica do Calendário 2020 (17/05/2020) até (23/05/2020). COVID-19: Observatório Fluminense, (24/05/2020). DOI: <https://doi.org/10.12957/eduerj.covid19rj.relatorio1>

Change in the progress of the epidemic

To represent the progress of the epidemic when complete data is not available, the graph with the **propagation speed** is an alternative for understanding the evolution/trend of the outbreak through new cases.

On the graph **propagation speed** is possible to observe the drop in the curve of country "A", which peak occurs on the 30º day and corresponds to a total of 525 cases.



A. Cunha Jr et al. Relatório 01 Progresso da COVID-19 no Brasil e no Estado do Rio de Janeiro 21ª Semana Epidemiológica do Calendário 2020 (17/05/2020) até (23/05/2020). COVID-19: Observatório Fluminense, (24/05/2020). DOI: <https://doi.org/10.12957/eduerj.covid19rj.relatorio1>

Monitoring graphs

The types of graphics generated with the file `epidemic_trends.m` for monitoring are as follows:

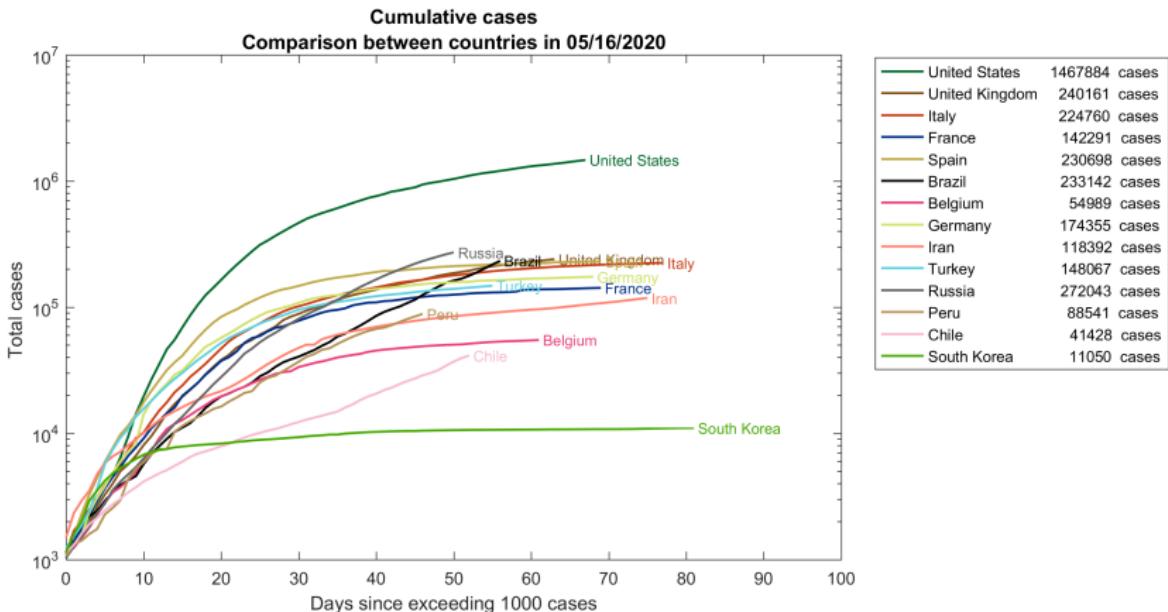
- Total `cases/deaths` in relation to time;
- Total `cases/deaths` per 1M inhabitants in relation to time;
- New `cases/deaths` per week in relation to time;
- New `cases/deaths` per week per 1M inhabitants in relation to time;
- New `cases/deaths` per week in relation to the total number of cases/deaths.

Contagion Mortality Progress

Example: monitoring of COVID-19

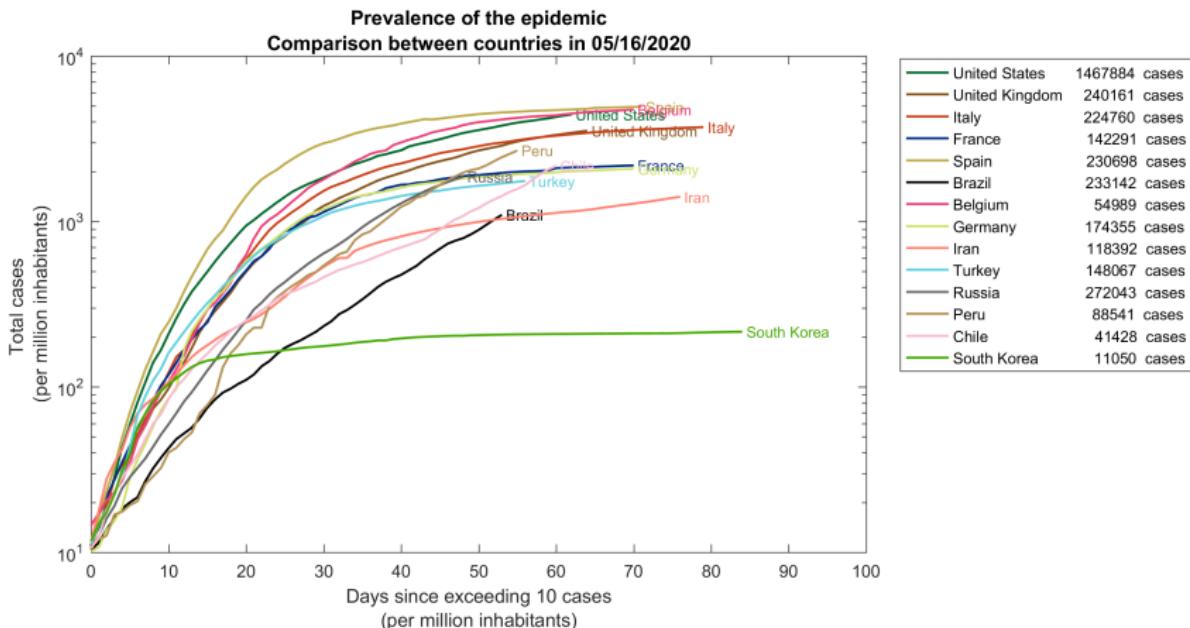
- Six types of monitoring graphs are presented. Using the EPIDEMIC code, it is possible to generate for each type presented here a version explaining the number of cases and the number of deaths;
- The following examples were generated using the EPIDEMIC educational code trend module for the COVID-19 pandemic situation in some countries. Therefore, the graphical results when executing the code should be similar to those presented here.

Total number of cases in relation to time



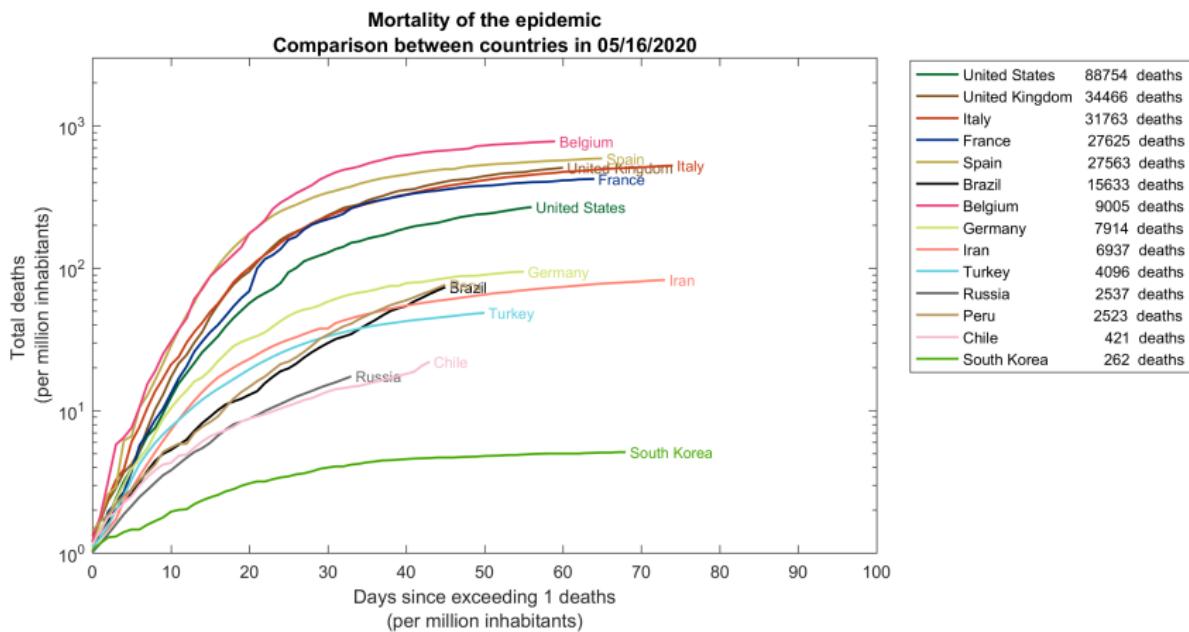
The graph analyzes the contagion evolution of the epidemic, showing the total number of cases as a function of time. The same analysis can be applied to the number of deaths (evolution of mortality).

Total number of cases per 1M inhabitants evolution



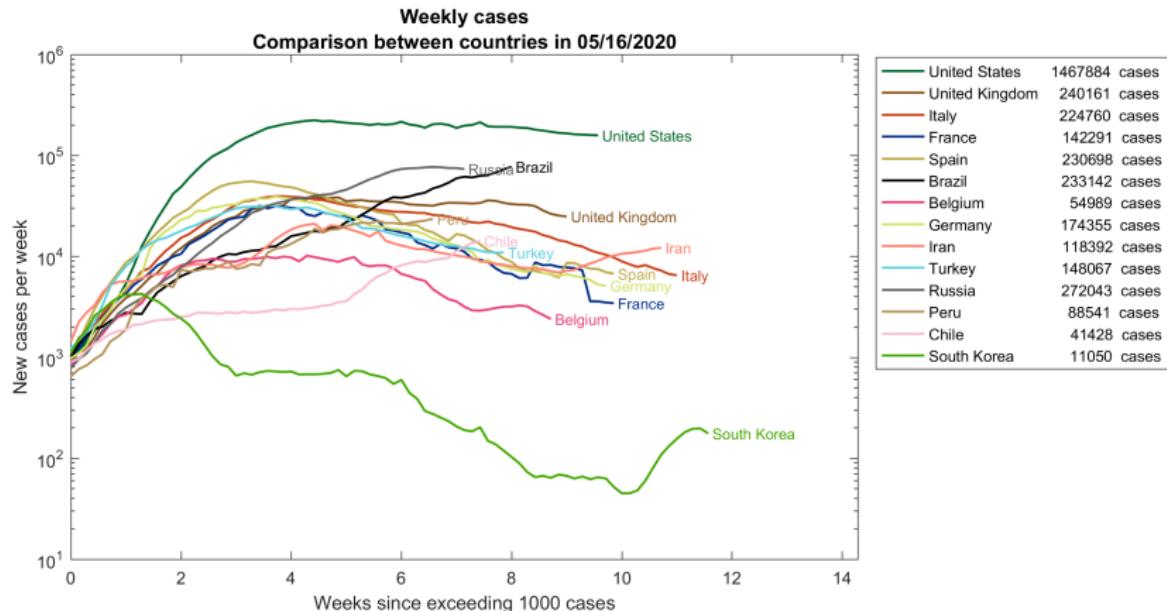
The graph makes it possible to analyze the prevalence of the epidemic through the proportion of cases in the countries listed in the legend, for which the total number of cases per million inhabitants in the countries evaluated is used.

Total number of deaths per 1M inhabitants evolution



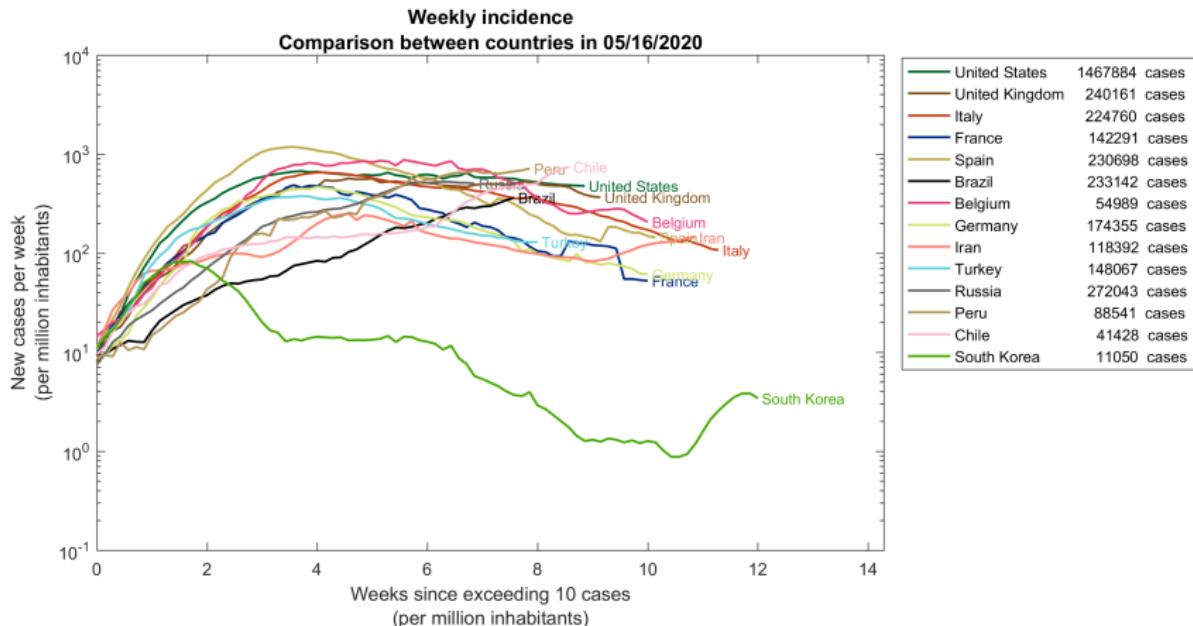
As in the previous graph, the comparison of mortality between countries employs the total number of deaths per million inhabitants in the country.

New cases per week evolution



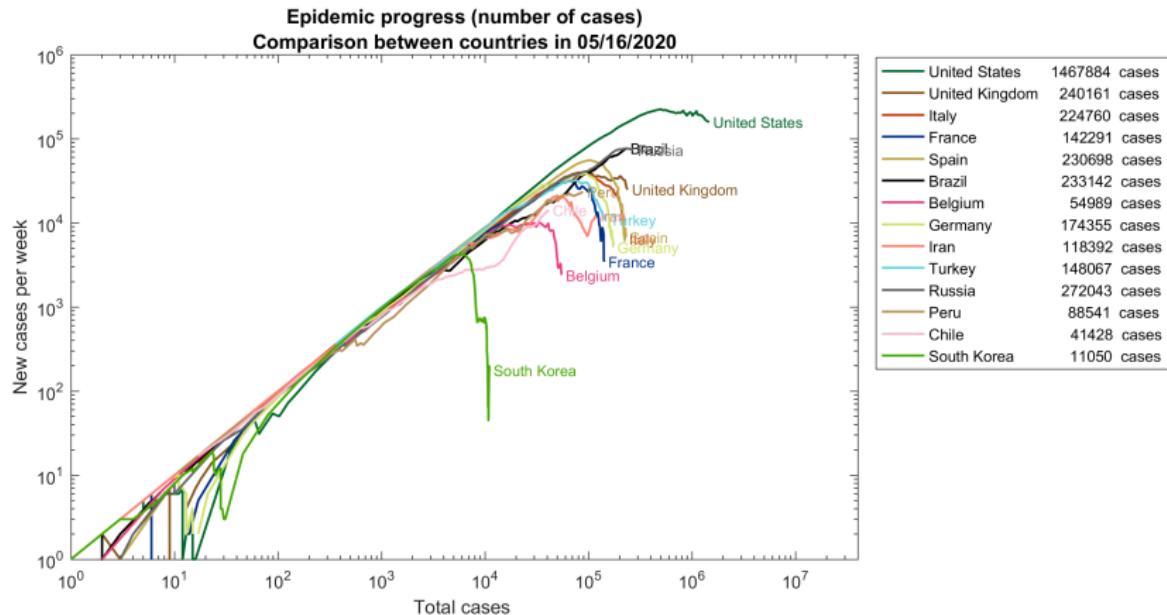
The choice of the week as a parameter to measure the evolution of the epidemic is due to the fact that daily behavior is very oscillatory. The sum of all week values filters the fluctuations present in the daily update.

New cases per week per 1M inhabitants evolution



This graph is similar to the previous, but it uses the number of new cases per million inhabitants, making it possible to compare the growth in the number of deaths in relation to the population (incidence).

New cases per week in relation to the total



The graph assesses the progress of the epidemic by comparing the number of new weekly cases with the total number of cases. If the epidemic is in an exponential phase, the number of new cases increases proportionally to the total, which is reflected in a straight line with a positive slope. When the epidemic goes into exhaustion there is a fall.

The code and data

Monitoring the numbers of cases and deaths from an epidemic requires realistic and reliable sources of data. In this sense, to understand how the code works, the data from the COVID-19 pandemic was used in the simulations.

The codes of module Trends are in the directory [EPIDEMIC-master\trends;](#)

To execute the code [epidemic_trends.m](#), you need a file with epidemic data with the same layout as the file information “owid-covid-data.csv”, which provides information about the COVID-19 pandemic in some countries, and which is available for download at

<https://ourworldindata.org/coronavirus-source-data>.

In addition, this file must be saved in the same directory as the code.

Display: EPIDEMIC and graphic types

When executing the code, an EPIDEMIC header is printed in the command window along with the types of graphics generated by the code.

```
disp('')
disp('=====')
disp(' EPIDEMIC — Epidemiology Educational Code ')
disp(' by E. Dantas, M. Grave, L. Roca, et al. ')
disp('')
disp(' This is an easy to run educational toolkit ')
disp(' for epidemiological analysis. ')
disp('')
disp(' www.EpidemicCode.org ')
disp('=====')
disp('')
disp('=====')
disp(' ++++++ EPIDEMIC TRENDS GRAPHICS ++++++ ')
disp('')
disp(' 1) Cumulative deaths')
disp(' 2) Cumulative cases')
disp(' 3) Epidemic progress (number of deaths)')
disp(' 4) Epidemic progress (number of cases)')
disp(' 5) Weekly deaths')
disp(' 6) Weekly cases')
disp(' 7) Mortality of the epidemic')
disp(' 8) Prevalence of the epidemic')
disp(' 9) Weekly deaths per million')
disp('10) Weekly incidence')
disp('')
```

Importing the data

The code starts with reading the data file and transforming it into the matrix format.

```
% Reading the data file
% (After downloading, save this file in the same directory as the code)
%
A = fileread('owid-covid-data.csv');
B = strsplit(A, '\n');

for i = 1:length(B(1,:))
    all_data(i,:) = strsplit(B{1,i}, ',', 'CollapseDelimiters', false);
end
```

Verification of imported data

Then, the imported data is checked for the disposition and positioning of the information necessary for the correct execution of the code. For this, the command “**if**” is used. If the data is not distributed correctly, a message will indicate the possible error.

```
if ~strcmp(all_data(1,2), 'location')
    error('Warning: location names must be in the 2nd column of the file')
elseif ~strcmp(all_data(1,3), 'date')
    error('Warning: dates must be in the 3rd column of the file')
elseif ~strcmp(all_data(1,4), 'total_cases')
    error('Warning: total cases must be in the 4th column of the file')
elseif ~strcmp(all_data(1,5), 'new_cases')
    error('Warning: new cases must be in the 5th column of the file')
elseif ~strcmp(all_data(1,6), 'total_deaths')
    error('Warning: total deaths must be in the 6th column of the file')
elseif ~strcmp(all_data(1,7), 'new_deaths')
    error('Warning: new deaths must be in the 7th column of the file')
elseif ~strcmp(all_data(1,8), 'total_cases_per_million')
    error('Warning: total cases per million must be in the 8th column of the file')
elseif ~strcmp(all_data(1,9), 'new_cases_per_million')
    error('Warning: new cases per million must be in the 9th column of the file')
elseif ~strcmp(all_data(1,10), 'total_deaths_per_million')
    error('Warning: total deaths per million must be in the 10th column of the file')
elseif ~strcmp(all_data(1,11), 'new_deaths_per_million')
    error('Warning: new deaths per million must be in the 11th column of the file')
end
```

Selecting the relevant columns

After completing the data import, the columns with the relevant data for plotting the graphs are defined.

```
data = [str2double(all_data(:, 4)) ,...
        str2double(all_data(:, 5)) ,...
        str2double(all_data(:, 6)) ,...
        str2double(all_data(:, 7)) ,...
        str2double(all_data(:, 8)) ,...
        str2double(all_data(:, 9)) ,...
        str2double(all_data(:,10)) ,...
        str2double(all_data(:,11))];
```

%

Configuring the analyzed locations

A loop is used to produce the curves of the monitored countries. They are plotted in the order defined by the value of the variable “**init**”. This command defines which countries will be part of the generated graphic, as well as the colors. The country must be indicated as described in the data file.

```
% Running the loop command in the 14 countries studied
%
for init = 1:1:14

    % Cleaning up reused variables inside the loop
    clearvars -except plot_type init all_data data name paises tot_deaths tot_cases

    % Order by countries that have more death
    if (init == 6); country='Brazil' ; color=[ 0, 0, 0]/255; end
    if (init == 14); country='South Korea' ; color=[ 69,169, 0]/255; end
    if (init == 10); country='Turkey' ; color=[ 96,209,224]/255; end
    if (init == 12); country='Peru' ; color=[181,147, 87]/255; end
    if (init == 9); country='Iran' ; color=[255,130,113]/255; end
    if (init == 8); country='Germany' ; color=[209,227,105]/255; end
    if (init == 13); country='Chile' ; color=[248,187,208]/255; end
    if (init == 1); country='United States' ; color=[ 0,104, 44]/255; end
    if (init == 4); country='France' ; color=[ 0, 45,135]/255; end
    if (init == 2); country='United Kingdom' ; color=[135, 85, 30]/255; end
    if (init == 3); country='Italy' ; color=[203, 63, 23]/255; end
    if (init == 5); country='Spain' ; color=[191,171, 72]/255; end
    if (init == 7); country='Belgium' ; color=[236, 64,122]/255; end
    if (init == 11); country='Russia' ; color=[102,102,102]/255; end
```

Defining the data to be collected

The desired data, contained in the data file, is then located using the command “**find**”.

```
% Collecting country data
location = data(find(strcmp([all_data(:,2)], country)),1:8);

% Defining the matrix with dates and the final day
all_data_date = all_data(:,3);
dates = all_data_date(find(strcmp([all_data(:,2)], country)),:);
end_time = max(datenum(dates))-1;
```

Vectorizing the data of interest

The data are separated into vectors containing total cases, total deaths, new daily cases and new daily deaths. So that these vectors have a length equal to the number of days evaluated and correspond to the data of the chosen country.

```
% Separating the data
tot_cases = location(:,1);
new_cases = location(:,2);
tot_deaths = location(:,3);
new_deaths = location(:,4);
```

The analysis per million inhabitants is also presented, because more populous countries will have absolute numbers higher than less populous countries (probably). Thus, the ratio per million “**pm**” minimizes the influence of population value on graphical comparisons.

```
% Data per million
tot_cases_pm = location(:,5);
new_cases_pm = location(:,6);
tot_deaths_pm = location(:,7);
new_deaths_pm = location(:,8);
```

New cases and deaths per week

The numbers of new cases and new deaths can vary widely from one day to the next. In this context, the following code is used to define that new cases or new deaths are evaluated per week.

```
% Consolidating new cases and deaths per week
for i = 7:1:max(max(size(dates)))
    new_cases7(i,1) = sum(new_cases(i-6:i,1));
    new_deaths7(i,1) = sum(new_deaths(i-6:i,1));
    new_cases7_pm(i,1) = sum(new_cases_pm(i-6:i,1));
    new_deaths7_pm(i,1) = sum(new_deaths_pm(i-6:i,1));
end
```

Defining the time axis (days)

Finally, data is generated since “X_deaths” deaths “X_cases” and cases are exceeded. The user can define these initial values, but it is indicated that they are values already reached in all the countries analyzed.

```
% Creating vectors from day zero from X deaths ('_deaths')
% or from X cases ('_cases').
% User sets zero day for cases or deaths
X_deaths_pm = 1;
X_cases_pm = 10;
X_deaths    = 100;
X_cases     = 1000;
```

Defining the vertical axis (data)

The commands for viewing each information are shown below. These being: number of deaths, number of cases, number of cases per million and number of deaths per million.

```
% Relative to total deaths
n = 0;
for i = 1:1:max(max(size(dates)))
    if (tot_deaths(i,1)    >= X_deaths)
        n = n + 1;
        tot_deaths_X (n,1) = tot_deaths (i,1);
        new_deaths_X (n,1) = new_deaths (i,1);
        new_deaths7_X(n,1) = new_deaths7(i,1);
    end
end

% Relative to total cases
n = 0;
for i = 1:1:max(max(size(dates)))
    if (tot_cases(i,1)    >= X_cases)
        n = n + 1;
        tot_cases_X (n,1) = tot_cases (i,1);
        new_cases_X (n,1) = new_cases (i,1);
        new_cases7_X(n,1) = new_cases7(i,1);
    end
end
```

Defining the vertical axis (data)

```
% Related deaths per million
n = 0;
for i = 1:1:max(max(size(dates)))
    if (tot_cases_pm(i,1)  >= X_cases_pm)
        n = n + 1;
        tot_cases_pmX (n,1) = tot_cases_pm (i,1);
        new_cases_pmX (n,1) = new_cases_pm (i,1);
        new_cases7_pmX(n,1) = new_cases7_pm(i,1);
    end
end

% Related cases per million
n = 0;
for i = 1:1:max(max(size(dates)))
    if (tot_deaths_pm(i,1)  >= X_deaths_pm)
        n = n + 1;
        tot_deaths_pmX (n,1) = tot_deaths_pm (i,1);
        new_deaths_pmX (n,1) = new_deaths_pm (i,1);
        new_deaths7_pmX(n,1) = new_deaths7_pm(i,1);
    end
end
```

Basic graphics settings

For the plot, some initial information must be included, such as:

- Definition of the fonts used in the title, in the axes and other texts contained in the graphics;
- Definition of the number of days that will be expressed in the axis x in the time graphs.

```
% Basics graphics settings
```

```
% Fonts
```

```
font_title = 10.5;  
font_labels = 10;  
font_default = 9;  
font_location = 8;
```

```
% X-axis size
```

```
day_axis = 100;
```

Caption alignment

To improve the presentation of the list of countries and their respective numbers of cases or deaths in the legend, spacing is used after the name of the countries using the following commands:

```
% Aligning the legend
if strcmp(country,'Brazil')      ); country_leg='Brazil'      ' ; end
if strcmp(country,'South Korea') ); country_leg='South Korea' ' ; end
if strcmp(country,'Turkey')       ); country_leg='Turkey'       ' ; end
if strcmp(country,'Peru')        ); country_leg='Peru'        ' ; end
if strcmp(country,'Iran')        ); country_leg='Iran'        ' ; end
if strcmp(country,'Germany')     ); country_leg='Germany'    ' ; end
if strcmp(country,'Chile')       ); country_leg='Chile'       ' ; end
if strcmp(country,'United States'); country_leg='United States' ' ; end
if strcmp(country,'France')      ); country_leg='France'      ' ; end
if strcmp(country,'United Kingdom'); country_leg='United Kingdom' ' ; end
if strcmp(country,'Italy')        ); country_leg='Italy'        ' ; end
if strcmp(country,'Spain')       ); country_leg='Spain'       ' ; end
if strcmp(country,'Belgium')     ); country_leg='Belgium'     ' ; end
if strcmp(country,'Russia')      ); country_leg='Russia'      ' ; end
```

Caption alignment

Then, the numbers of cases or deaths are aligned to the right, visually improving the data presentation.

```
deaths_leg = [ ' ', num2str(max(tot_deaths))];  
if size(num2str(max(tot_deaths)))<5; deaths_leg=[ ' ', ' ', ' ', ' ', ' ', num2str(max(tot_deaths))]; end  
if size(num2str(max(tot_deaths)))<4; deaths_leg=[ ' ', ' ', ' ', ' ', ' ', num2str(max(tot_deaths))]; end  
if size(num2str(max(tot_deaths)))<3; deaths_leg=[ ' ', ' ', ' ', ' ', ' ', num2str(max(tot_deaths))]; end  
if size(num2str(max(tot_deaths)))<2; deaths_leg=[ ' ', ' ', ' ', ' ', ' ', num2str(max(tot_deaths))]; end  
  
cases_leg = [ ' ', num2str(max(tot_cases))];  
if size(num2str(max(tot_cases)))<7; cases_leg=[ ' ', ' ', ' ', ' ', ' ', ' ', ' ', num2str(max(tot_cases))]; end  
if size(num2str(max(tot_cases)))<6; cases_leg=[ ' ', ' ', ' ', ' ', ' ', ' ', ' ', num2str(max(tot_cases))]; end  
if size(num2str(max(tot_cases)))<5; cases_leg=[ ' ', ' ', ' ', ' ', ' ', ' ', ' ', num2str(max(tot_cases))]; end  
if size(num2str(max(tot_cases)))<4; cases_leg=[ ' ', ' ', ' ', ' ', ' ', ' ', ' ', num2str(max(tot_cases))]; end
```

Plotting the graphs

There are two formats for plotting graphs that can be used. The first example is related to **temporal analysis**. The following is presented an example using the total deaths over time.

```
% Plot total deaths by time (zero day defined by deaths)
figure (1);
n = max(max(size(tot_deaths_X)));
days = 0:1:n-1;
fig = semilogy(days,tot_deaths_X, ...
    'DisplayName',[country_leg,' ',deaths_leg,' ', 'deaths'], ...
    'color',color, ...
    'LineWidth',1.25);
hold on;
text (n-1,tot_deaths_X(n,1),[' ',country_leg], ...
    'FontSize',font_location, ...
    'color',color, ...
    'Clipping','on');
```

The other temporal plots are obtained by changing the term **tot_deaths** for **tot_cases** for cases and the **tot_deaths_X** for:

- **new_deaths7_X** (new deaths) or **new_deaths7_pmX** (per million);
- **tot_cases_X** (total cases) or **tot_cases_pmX** (per million);
- **new_cases7_X** (new cases) or **new_cases7_pmX** (per million).

Plotting the graphs

The second example is related to **temporal analyzes**, which relate the new data to the total data. The following is an example of new deaths by total.

```
% Plot new deaths by time (zero day defined by deaths)
figure (5);
n    = max(max(size(new_deaths_X)));
days = 0:1:n-1;
fig = semilogy(days/7,new_deaths7_X, ...
    'DisplayName',[country_leg , ' ', deaths_leg , ' ', 'deaths'] , ...
    'color',color , ...
    'LineWidth',1.25);
hold on;
text ((n-1)/7,new_deaths7_X(n,1),[' ',country_leg] , ...
    'FontSize',fontSize , ...
    'color',color , ...
    'Clipping','on');
```

The other timeless plots are obtained by changing the term **deaths** for **cases**.

Aesthetic graphics settings

Finally, configurations are made in relation to the size of the figure, title of the graph and axes and the position of the legend. The following are the settings for two plot examples shown previously.

```
% Aesthetic graphics settings
%
figure (1);
set(gca,'FontSize',font_default)
set(gcf,'units','normalized','OuterPosition',[0 0 0.7 0.7])
title ({'Cumulative deaths',...
    ['Comparison between countries in ',datestr(end_time,23)]},...
    'FontSize',font_title);
xlabel (['Days since exceeding ',num2str(X_deaths),' deaths'],...
    'FontSize',font_labels);
ylabel ('Total deaths ','FontSize',font_labels);
legend ('location','northeastoutside');
y_init = 100;
max_y = 200000;
axis([0 day_axis y_init max_y]);
```

Aesthetic graphics settings

The command “**axis**” is used to define the intervals of the axes. This setting can be changed according to the user's interest.

```
figure (5);
set(gca,'FontSize',font_default)
set(gcf,'units','normalized','OuterPosition',[0 0 0.7 0.7])
title ({'Weekly deaths',...
    ['Comparison between countries in ',datestr(end_time,23)]},...
    'FontSize',font_title);
xlabel ({['Weeks since exceeding ',num2str(X_deaths),' deaths']},...
    'FontSize',font_labels);
ylabel ({'New deaths per week'},'FontSize',font_labels);
legend ('location','northeastoutside');
y_init = 1;
max_y = 100000;
axis([0 day_axis/7 y_init max_y]);
```

Saving the graphics

There are ten types of graphs plotted by this code. Being them:

Graph type	Figure name
1) Cumulative deaths - absolute value (total deaths per time)	deaths-total-abs_AAAA-MM-DD
2) Cumulative cases - absolute value (total cases per time)	cases-total-abs_AAAA-MM-DD
3) Progress of the deaths - absolute value (new deaths by total deaths)	deaths-progress-abs_AAAA-MM-DD
4) Progress of the cases - absolute value (new cases per total cases)	cases-progress-abs_AAAA-MM-DD
5) Weekly deaths - absolute value (new deaths by time)	deaths-weekly-abs_AAAA-MM-DD
6) Weekly cases - absolute value (new cases by time)	cases-weekly-abs_AAAA-MM-DD
7) Mortality - per million inhabitants (total deaths per million)	mortality-pm_AAAA-MM-DD
8) Prevalence - per million inhabitants (total cases per million over time)	prevalence-pm_AAAA-MM-DD
9) Weekly deaths - per million inhabitants (new weekly deaths per million)	deaths-weekly-pm_AAAA-MM-DD
10) Incidence - per million inhabitants (new weekly cases per million)	incidence-weekly-pm_AAAA-MM-DD

Saving the graphics

If you need to save the plots, just use the command “*print*”, defining the file name and the format in which it will be saved. The following are the commands used to save the previous examples.

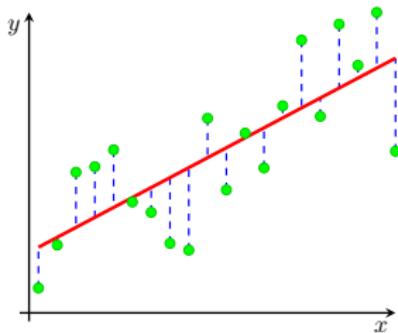
```
% Saving the charts
%
print(figure( 1 ),[ 'deaths-total-abs_' , datestr(end_time,29) , '.png' ], '-dpng', '-r300');
print(figure( 5 ),[ 'deaths-weekly-abs_' , datestr(end_time,29) , '.png' ], '-dpng', '-r300');
```

The name of the graphics is presented according to the original code and in the standard defined earlier in this document.

Forecasts

Regressor

- The forecasts are calculated with the aid of statistical regressors;
- The estimation process uses the **Least Squares Method**.



Goals:

Get a curve fitting (—) for a certain data set (•), in order to minimize the sum of the squares of the differences (---) between the estimated values and the observed data.

Methodology

- The last 5 days of the data samples were considered, as they exhibit the most recent trend;
- The last 5 days have been placed on the logarithmic scale (base 10), because if the growth is exponential, it will be reflected in a straight line;
- A linear regression is performed, using the Least Squares Method, and building a 95% confidence band;
- The estimated values have been plotted with the estimated confidence band.

Least Squares Method

The linear regression using Least Squares Method has been done by obtaining the estimators β_1 and β_2 , following the equation: $Y_i = \beta_1 + \beta_2 X_i$, where

$$\beta_2 = \frac{\sum_{i=1}^n (X_i - \bar{X}_n)(Y_i - \bar{Y}_n)}{\sum_{i=1}^n (X_i - \bar{X}_n)^2}$$

$$\beta_1 = \bar{Y}_n - \beta_2 \bar{X}_n$$

such that \bar{X}_n and \bar{Y}_n are the sample means.



L. Wasserman. All of Statistics: A Concise Course in Statistical Inference. Springer, New York, NY, 2004. DOI:
<https://doi.org/10.1007/978-0-387-21736-9>

95% Confidence interval

The 95% confidence interval is obtained through the regression and prediction variances, given by the equation below:

$$Y_* \pm 1.96 \xi_n$$

where

- $\xi_n^2 = \hat{\sigma}^2 \left(\frac{\sum_{i=1}^n (X_i - X_*)^2}{n \sum_i (X_i - \bar{X})^2} + 1 \right)$
- $\hat{\sigma}^2 = (\frac{1}{n-2}) \sum_{i=1}^n \epsilon_i^2$
- $\epsilon_i = Y_i - (\beta_1 + \beta_2 X_i)$

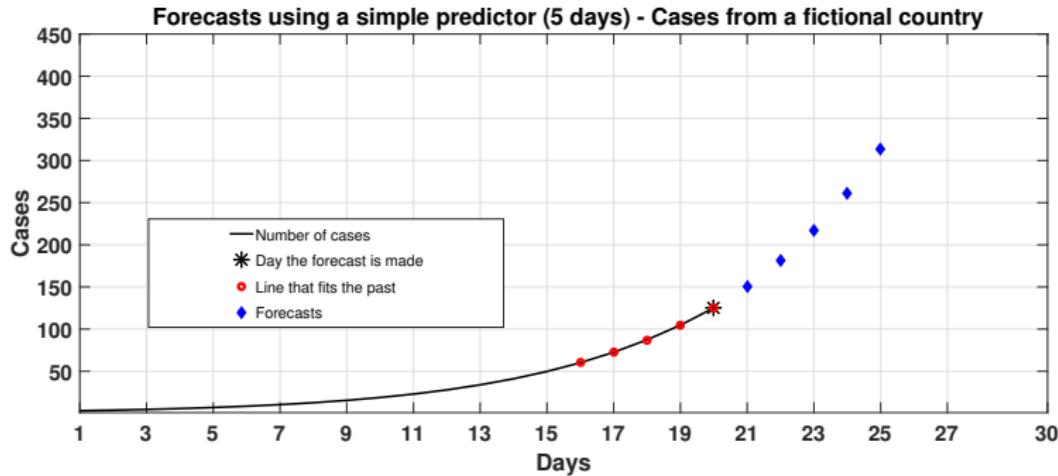


L. Wasserman. All of Statistics: A Concise Course in Statistical Inference. Springer, New York, NY, 2004. DOI: <https://doi.org/10.1007/978-0-387-21736-9>.

Comments

- The estimates epidemic progression can change over time, thus using very early data may not generate a reliable sample for the forecast;
- Performing forecasts many days ahead may also not adequately represent reality, so we choose to make predictions with only 5 days ahead.

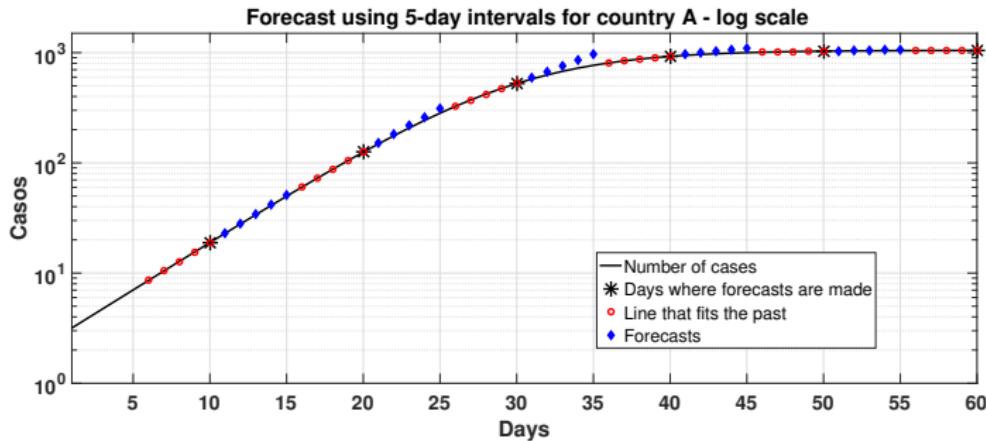
Short-term data forecasts



- ① Find the curve that best approaches to the logarithm of the data sequence until the 5th day before the date on which the forecast is made (20 day). The red circles “○” are points on the curve;
- ② Extrapolate the curve from the date of the forecast attempting to predict the next 5 days, marked on the graph with “◇”.

Example: short-term forecasts

- The logarithmic scale graph shows the forecasts of cases in a fictional country A;



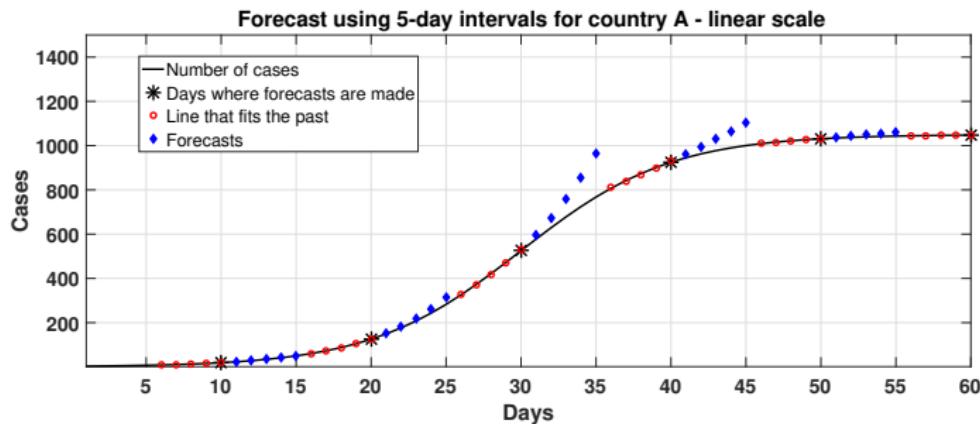
- We have in this graph the original case curve over a period of 60 days, with five forecasts made “ * ”, allowing a comparison between predicted and realized;
- As the curve moves away from exponential behavior, the difference between “ ♦ ” and the line “ - ”, e.i., the **forecast errors**, increase.



A. Cunha Jr et al. Relatório 01 Progresso da COVID-19 no Brasil e no Estado do Rio de Janeiro 21^a Semana Epidemiológica do Calendário 2020 (17/05/2020) até (23/05/2020). COVID-19: Observatório Fluminense, (24/05/2020). DOI: <https://doi.org/10.12957/eduerj.covid19rj.relatorio1>

Example: short-term forecasts

- The linear scale graph shows the forecast of cases in a fictional country A;



- Using the same forecasting tool, but now on the linear scale, the differences between forecasts and the original curve are visually greater in the range from the 30th to the 45th day;
- This difference occurs due to the forecast has been performed using a line on the logarithmic scale, and in this region, the original curve has no exponential behavior.



A. Cunha Jr et al. Relatório 01 Progresso da COVID-19 no Brasil e no Estado do Rio de Janeiro 21^a Semana Epidemiológica do Calendário 2020 (17/05/2020) até (23/05/2020). COVID-19: Observatório Fluminense, (24/05/2020). DOI: <https://doi.org/10.12957/eduerj.covid19rj.relatorio1>

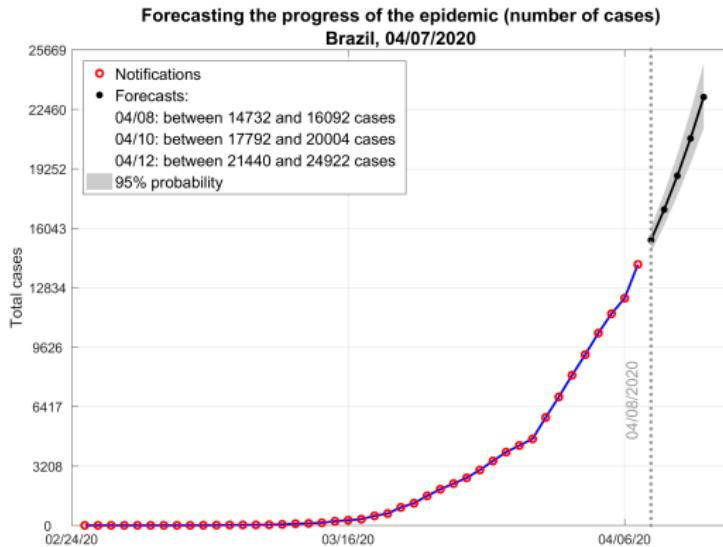
Example: COVID-19

- Use of the problem situation of COVID-19 in Brazil;
- Two types of graphics: **forecasting cases** and **forecasting deaths**;
- The analyzed local can be modified.

Example: COVID-19

Forecast of **total cases** from COVID-19 in Brazil (time-depending)

- EPIDEMIC graphic result:

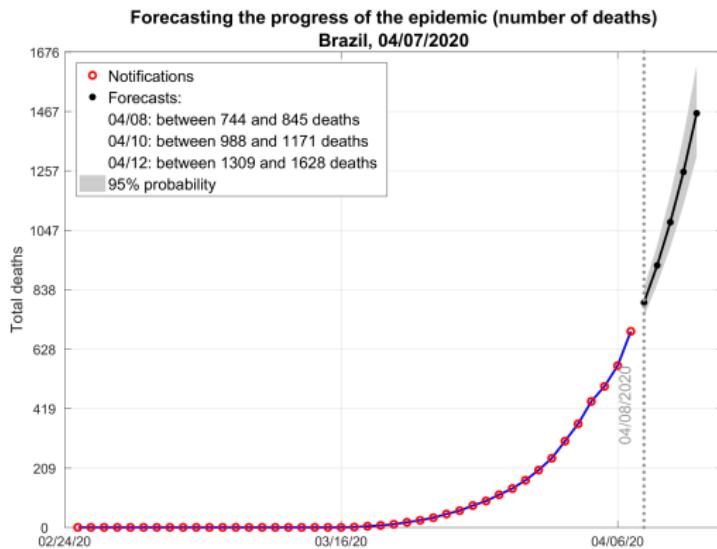


- In the graph, the gray color shows the 95 % confidence band.

Example COVID-19

Forecast of **total deaths** from COVID-19 in Brazil (time-depending)

- EPIDEMIC graphic result:



- In the graph, the gray color shows the 95 % confidence band.

The code and data

- As in the Trends module, reliable data sources are needed to forecast cases and deaths in an epidemic. To know how the code works, data from the COVID-19 pandemic is used.
- The codes for the module Forecasts are in the directory EPIDEMIC-master\forecasts;
- To run the `epidemic_forecasts.m` code, an epidemic data file with the same layout as the “cases-brazil-states.csv” file is needed, which provides information about the COVID-19 pandemic in the federal states of Brazil (available for download at <https://github.com/wcota/covid19br>). Also, this file must be saved in the same code directory.
- The algorithm is divided into two sections. The first with the commands for importing and organizing data related to the analyzed place, thus the second is for plotting the graphs of the case and death forecasts.

Display: EPIDEMIC, objective and forecast graphs

When executing the forecast code, the objective and the types of forecast graphs generated by the code are printed in the command window.

```
disp('_____')
disp(' EPIDEMIC — Epidemiology Educational Code ')
disp(' by E. Dantas, M. Grave, L. Roca, et al. ')
disp('_____')
disp(' This is an easy to run educational toolkit ')
disp(' for epidemiological analysis.')
disp('_____')
disp(' www.EpidemicCode.org')
disp('_____')
disp(')')
disp('_____')
disp(' ++++++ OBJECTIVE ++++++')
disp('_____')
disp(' To present the number of cases and deaths ')
disp(' over time, with a 5-day forecast ahead ')
disp(' determined by linear regression on the ')
disp(' logarithmic scale of the number of cases ')
disp(' and deaths, with a 95% confidence interval.')
disp('_____')
disp(')')
disp('_____')
disp(' ++++++ EPIDEMIC FORECASTS GRAPHICS ++++++')
disp('_____')
disp(' Forecasting the progress of the epidemic: ')
disp(' 1) Number accumulated of cases')
disp(' 2) Number accumulated of deaths')
```

Importing data

The code begins with the date of the last data update. Then the data file is read and transferred from numerical data to a matrix.

```
%% Organizing death and case data
%
% Insert the last data update date, in the format MM/DD/AAAA
%
up_date = '04/07/2020';
%
%
% Reading the data file
%(After downloading, save this file in the same directory as the code)
%
A = fileread('cases-brazil-states.csv');
B = strsplit(A, '\n');

for i = 1:length(B(:, :))
    all_data(i, :) = strsplit(B{1, i}, ',', 'CollapseDelimiters', false);
end
```

Verification of imported data

Then, using the command “**if**”, the imported data is checked for the disposition and positioning of the information necessary for the correct execution of the code.

```
if ~strcmp(all_data(1,1), 'date')
    error('Warning: dates must be in the 1st column of the file')
elseif ~strcmp(all_data(1,3), 'state')
    error('Warning: states names must be in the 3rd column of the file')
elseif ~strcmp(all_data(1,6), 'totalDeaths')
    error('Warning: total deaths must be in the 6th column of the file')
elseif ~strcmp(all_data(1,8), 'totalCases')
    error('Warning: total cases must be in the 8th column of the file')
end
```

If the data is not distributed correctly, a message will indicate the possible error.

Defining the analysis location

With the imported data in hand, it is necessary to define the location, in this example is Brazil, and is to be the sum of the data of its states defined by its acronyms. The “**find**” and “**for**” commands are used to find and set the data size based on the state with the largest number of data.

```
% Defining country as the sum of data for all its federal states
%
country = 'Brazil';
states = {'RO' 'AC' 'AM' 'RR' 'PA' 'AP' 'TO' 'MA' 'PI' ...
          'CE' 'RN' 'PB' 'PE' 'AL' 'SE' 'BA' 'MG' 'ES' ...
          'RJ' 'SP' 'PR' 'SC' 'RS' 'MS' 'MT' 'GO' 'DF'};
%
%
% Checking the state data size
%
for i = 1:length(states)
    location = all_data(find(strcmp(all_data(:,3),states(1,i))),:);
    maxcases(1,i) = length(cellfun(@str2double,(location(:,8))));
    maxdeaths(1,i) = length(cellfun(@str2double,(location(:,6))));
end
```

Collecting case and death data

To complete the first section of data organization the “**find**” and “**for**” commands are used again to gather cases and deaths data.

```
% Collecting data on total cases and deaths for each state
%
cases_states = zeros(length(states),max(maxcases) );
deaths_states = zeros(length(states),max(maxdeaths));

for i = 1:length(states)
    location = all_data(find(strcmp(all_data(:,3),states(1,i))),:);
    cases_states(i,:) = [zeros(1,max(maxcases)-maxcases(1,i)),...
        nonzeros(cellfun(@str2double,location(:,8)))'];
    deaths_states(i,:) = [zeros(1,max(maxdeaths)-maxdeaths(1,i)),...
        cellfun(@str2double,location(:,6))'];
end
%
```

Consolidating case and death data

The country data are consolidated as the sum of states using the same commands.

```
% Consolidating data on total cases and deaths for the country
%
cases_country = zeros(1,length(cases_states (1,:)));
deaths_country = zeros(1,length(deaths_states(1,:)));

for i = 1:length(states)
    cases_country = cases_country + cases_states (i,:);
    deaths_country = deaths_country + deaths_states(i,:);
end

cases = cases_country ;
deaths = deaths_country;
%
```

Loop for calculating regression and plotting

- In this section of the code, linear regression and settings for plotting data and forecasts are performed.
- The forecast plotting commands are the same for cases and deaths, so a loop of commands performs it in the order defined by the variable “init”.

```
%% Plotting the forecast of cases/deaths by time
%
for init = 1:1:2

    % Cleaning up reused variables inside the loop
    clearvars -except up_date init cases country deaths all_data ...
               cases_country deaths_country cases_states ...
               deaths_states location maxcases maxdeaths states

    % Data loop for plotting cases and then deaths
    if (init == 1); data = cases ; info = 'cases' ; end
    if (init == 2); data = deaths; info = 'deaths' ; end
```

Days setting

The first step is to set how the dates will be read. Next, the date is converted from a “**string**” to a numeric data using the “**str2num**” command. Thus, it is generated the vector of the days, one by one for the number of cases and deaths.

```
% Defaults: Adjust days and dates
%
date = [up_date(1,1:2), '—', up_date(1,4:5), '—', up_date(1,7:10)];
days = length (data);
day = str2num(up_date(1,1:2));
%
```

Setting dates in numbers

The setting of dates (forecast days) is performed in “**numbers**” using the “**datenum**” command, which are the next, third, and fifth days following the date of the last value.

```
% Setting string for days ahead (Transforming the date to strings)
% This make sure that, given the day, the program gives the days ahead
% without end of month problems like "30/05, 32/05, 34/05, ..."
%
Datestring = date;
formatIn  = 'mm-dd-yyyy';

ND = datenum(Datestring,formatIn)+1;
ND3 = ND+2;
ND5 = ND+4;
%
```

Setting dates in “string”

Next, the dates are set in “**string**” for inclusion in the legends and x-axis of the forecast plots.

```
% Setting to include in the legend
%
str1 = datestr(ND , 'mm-dd-yyyy ');
str3 = datestr(ND3, 'mm-dd-yyyy ');
str5 = datestr(ND5, 'mm-dd-yyyy ');
%
LINE_DATE = [str1(1,1:2), '/', str1(1,4:5), '/', str1(1,7:10)];
d1 = [str1(1,1:2), '/', str1(1,4:5)];
d3 = [str3(1,1:2), '/', str3(1,4:5)];
d5 = [str5(1,1:2), '/', str5(1,4:5)];
%
%
% Setting to include in the X-axis
%
Datestring = date;
formatIn = 'mm-dd-yyyy';
final = datenum(Datestring,formatIn);
first = final - days;
str = datestr(first:21:final , 'mm/dd/yy');
```

Vector of days

Then, it is generated a vector of days, one by one for the number of cases and deaths.

```
% Vector of days, one by one for the number of cases/deaths.  
%  
days = sparse(1,length(data));  
  
for i = 1:length(data)  
    days(1,i) = i;  
end  
%
```

Linear regression

To perform linear regression, take the last five days of the data as a sample and then apply the regression using the “**polyfit**” command, generating the coefficients of the line that will be used to calculate the 5-day forecast.

```
% Take the last 5 days of the data as a sample
% Note: "5s" refers to the last 5 days taken as a sample
%
data5s = sparse(1,5);
days5s = sparse(1,5);

for i = 0:4
    data5s(1,i+1) = data(1,length(data)-4+i);
    days5s(1,i+1) = length(data)-4+i;
end
%
%
% Linear regression of log points vs days of the sample (last 5 days)
%
[coeffs5s ,S_5s] = polyfit(days5s ,log10(data5s) ,1);
%
```

5-day forecast

After applying linear regression, the 5-day forecast values are generated using the “**polyval**” command with the regression coefficients and the forecast days.

```
% 5-day prediction
% Note: "5f" refers to the 5-day forecast
%
data5f = sparse(1,5);
days5f = sparse(1,5);

for i = 1:5
    days5f(1,i) = length(data) + i;
end

data5f = polyval(coeffs5s ,days5f);
%
```

Confidence band

Finally, the 95 % confidence band is obtained. For this, the prediction and correction variances are calculated.

```
% Confidence band calculation
%
% Method 1
%
% Fitted line
datafit = coeffs5s(1,2) + coeffs5s(1,1)*days5s;
% Variance estimation
N = length(days5s);
sigma2 = sum((log10(data5s) - datafit).^2)/(N-1);
% Denominator
den = N*sum((days5s-mean(days5s)).^2);

for i = 1:N;
    % Numerator
    num = sum((days5s-days5f(1,i)).^2);
    % Variance square correction
    ksi2(1,i) = sigma2*((num/den)+1);
end
% Variance correction
ksi = sqrt(ksi2);

% Confidence band
upperCB = data5f + 1.96*ksi;
lowerCB = data5f - 1.96*ksi;
```

Confidence band (alternative method)

Another method to calculate is by the “**polyconf**” command.

```
% Method 2 (This method uses a direct command)
%
% To run on MATLAB disable the commands below
pkg install -forge optim
pkg load optim
%
[deaths5f,dy] = polyconf(coeffs5s, days5f, S_5s);
upperCB_data5f = data5f + dy;
lowerCB_data5f = data5f - dy;
```

Graph settings

From now on, the graphics settings are displayed. Starting plotting reported cases/deaths and defining titles.

```
% Chart configuration
%
figure

% Plot command notifications points
set(gca , 'YScale','linear')
plot(days,data , 'b' , 'LineWidth' ,2);
hold on
c = scatter(days,data , 'r' , 'LineWidth' ,2);

% Titles configuration
set(gca , 'FontSize',12)
title({{['Forecasting the progress of the epidemic '...
    '(number of ',info , ')'] ,...
    [country , ' , ',up_date]}},...
    'FontSize',16);
ylabel({{['Total ',info]}}, 'FontSize',14)
hold on
```

Forecast and confidence band plotting

Next, the cases/deaths for the 5-day forecast and the confidence band are plotted.

```
% Confidence band
CB = fill([days5f fliplr(days5f)],...
[10.^lowerCB fliplr(10.^upperCB)],...
[0.8 0.8 0.8],...
'LineStyle','none');

% Forecasts points
plot(days5f,10.^(data5f),'k','LineWidth',2);
hold on
b = scatter(days5f,10.^(data5f),'k','filled');
hold on
```

Legend setting

The legend is set by adding the range of cases and deaths predicted for the following days.

```
% Legend configuration
aaa = scatter(1,1,1,[1 1 1]);
bbb = scatter(1,1,1,[1 1 1]);
ccc = scatter(1,1,1,[1 1 1]);
lgd = legend([c b aaa bbb ccc CB],...
    'Notifications',...
    'Forecasts:',...
    [d1,: between ' num2str(round(10.^lowerCB(1,1))) ...
        ' and ' ...
        num2str(round(10.^upperCB(1,1))) ...
        ' ',info]...
    [d3,: between ' num2str(round(10.^lowerCB(1,3))) ...
        ' and ' ...
        num2str(round(10.^upperCB(1,3))) ...
        ' ',info]...
    [d5,: between ' num2str(round(10.^lowerCB(1,5))) ...
        ' and ' ...
        num2str(round(10.^upperCB(1,5))) ...
        ' ',info]...
    '95% probability',...
    'Location','northwest','Orientation','vertical');
set(lgd,'FontSize',14);
```

Breaking notifications and forecasts

A dotted line breaking notifications and forecasts is used to show what the actual and predicted data are.

```
% Dotted line between notifications and forecasts
scat = scatter(ones(1,70)*days5f(1,1) ,...
    0:...
    (10^(upperCB(1,5))*1.03) / 69:...
    10^(upperCB(1,5))*1.03 ,...
    10,[0.6 0.6 0.6], 'filled');

txxt = text(days5f(1,1)-1.5 ,...
    0.2*10^(data5f(1,5)) ,...
    LINE_DATE ,...
    'FontSize' ,14 ,...
    'Color',[0.6 0.6 0.6]);
set(txxt , 'Rotation',90);
```

Scale setting

The scales are adjusted to finish the graph settings.

```
% Scale configuration
grid

ylim([0 10^(upperCB(1,5))*1.03])
set(gca,'ytick',...
    floor(0:(10^(upperCB(1,5))*1.03)/8:10^(upperCB(1,5))*1.03),...
    'yticklabel',...
    floor(0:(10^(upperCB(1,5))*1.03)/8:10^(upperCB(1,5))*1.03));

set(gca,'xtick',0:21:length(days),'xticklabel',str);

set(gcf,'Position',[100, 100, 1000, 700]);
```

Saving graphics

Finally, it is used the “**print**” command to save the graphic in the same directory in which the code run.

```
% Saving the chart
%
print(figure(1) ,['forecasts-cases' ,'.png'], '-dpng',' -r300');
%
print(figure(2) ,['forecasts-deaths' ,'.png'], '-dpng',' -r300');
```

Acknowledgment

Financial support



Institutional support



References

References: books

-  D. J. Higham and N. J. Higham **MATLAB Guide**, Third Edition, SIAM, 2017. DOI: <https://doi.org/10.1137/1.9781611974669>
-  F. Brauer, P. van den Driessche and J. Wu. **Mathematical Epidemiology**, Springer-Verlag, Berlin, 2008. DOI: <https://doi.org/10.1007/978-3-540-78911-6>
-  F. Brauer and C. Castillo-Chávez. **Mathematical Models in Population Biology and Epidemiology**, Springer, New York, 2012. DOI: <https://doi.org/10.1007/978-1-4614-1686-9>
-  L. Wasserman. **All of Statistics: A Concise Course in Statistical Inference**, Springer, New York, NY, 2004. DOI: <https://doi.org/10.1007/978-0-387-21736-9>
-  M. Martcheva. **An Introduction to Mathematical Epidemiology**, Springer US, New York, 2015. DOI: <https://doi.org/10.1007/978-1-4899-7612-3>
-  P. C. Hansen, V. Pereyra and G. Scherer. **Least Squares Data Fitting with Applications**. Baltimore: Johns Hopkins University Press., 2012. DOI: [10.1353/book.21076](https://doi.org/10.1353/book.21076)
-  R. M. Anderson and R. M. May. **Infectious Diseases of Humans: Dynamics and control**, Oxford University Press, Oxford, 1992. DOI: <https://doi.org/10.1002/hep.1840150131>

References: papers and reports

-  A. Cunha Jr et al. Relatório 01 Progresso da COVID-19 no Brasil e no Estado do Rio de Janeiro 21ª Semana Epidemiológica do Calendário 2020 (17/05/2020) até (23/05/2020). COVID-19: Observatório Fluminense, (24/05/2020). DOI: <https://doi.org/10.12957/eduerj.covid19rj.relatorio1>
-  D. Volpatto, A. C. Resende, L. Dos Anjos, J. Silva, C. Dias, R. Almeida and S. Malta. **Assessing social distancing release strategies in Brazil and Rio de Janeiro state**, Scientific Electronic Library Online, 2020. DOI: <https://doi.org/10.1590/SciELOPreprints.595>
-  D. Volpatto, A. C. Resende, L. Dos Anjos, J. Silva, C. Dias, R. Almeida and S. Malta. **Spreading of COVID-19 in Brazil: Impacts and uncertainties in social distancing strategies**, medrxiv, 2020. DOI: <https://www.medrxiv.org/content/10.1101/2020.05.30.20117283v1>
-  E. Dantas, M. Tosin and A. Cunha Jr. **Calibration of a SEIR–SEI epidemic model to describe the Zika virus outbreak in Brazil**, Applied Mathematics and Computation, vol.338, p.249-259, 2018. DOI: <https://doi.org/10.1016/j.amc.2018.06.024>
-  M. Begon, M. Bennett, R. G. Bowers, et al. **A clarification of transmission terms in host-microparasite models: numbers, densities and areas**, Epidemiol. Infect., 129, 2002. DOI: <https://doi.org/10.1017/S0950268802007148>

References: papers and reports

-  P. Diaz, P. Constantine, K. Kalmbach, E. Jones and S. Pankavichm. **A modified SEIR model for the spread of Ebola in Western Africa and metrics for resource allocation**, Applied Mathematics and Computation, 324(1): 141-155, 2018.
DOI: <https://doi.org/10.1016/j.amc.2017.11.039>
-  W. Cota. **Monitoring the number of COVID-19 cases and deaths in Brazil at municipal and federative units level**. Scientific Electronic Library Online, 2020.
DOI: <https://doi.org/10.1590/SciELOPreprints.362>
-  W. Lyra, J. Nascimento, J. Belkhiria, L. de Almeida, P. P. Chrissim and I. Andrade. **COVID-19 pandemics modeling with SEIR(+CAQH), social distancing, and age stratification. The effect of vertical confinement and release in Brazil.**, medRxiv, 2020.
DOI: <https://doi.org/10.1101/2020.04.09.20060053>
-  Y. Li, X. Liu, and L. Wang. **Modelling the transmission dynamics and control of Mumps in Mainland China**, Int J Environ Res Public Health, 15(1): 33., 2018.
DOI: <https://doi.org/10.3390/ijerph15010033>

References: sites

-  Centers for Disease Control and Prevention, **Smallpox**
<https://www.cdc.gov/smallpox>
-  COVID-19: Observatório Fluminense (COVID19RJ)
<https://www.covid19rj.org>
-  Covid Trends
<https://aatishb.com/covidtrends>
-  Grupo de Métodos Analíticos em Vigilância Epidemiológica
<https://covid-19.procc.fiocruz.br>
-  Ministério da Saúde, **Caxumba: o que é, causas, sintomas, tratamento, diagnóstico e prevenção**
<https://saude.gov.br/saude-de-a-z/caxumba>

References: sites

-  Nature, **Five ways to ensure that models serve society: a manifesto**
<https://www.nature.com/articles/d41586-020-01812-9>
-  Observatório Covid-19 BR
<https://covid19br.github.io>
-  Our World in Data, **Coronavirus Source Data**
<https://ourworldindata.org/coronavirus-source-data>
-  Towards data science. **Transparency, Reproducibility, and Validity of COVID-19 Projection Models**
shorturl.at/IuK49
-  Wikipédia, **Método dos Mínimos Quadrados**
https://pt.wikipedia.org/wiki/M%C3%A9todo_dos_m%C3%ADnimos_quadrados