

Análise automática do Painel Coronavírus

Alberto Saa

UNICAMP

(26/05/2020)

Resumo

A ideia deste pequeno projeto é a elaboração de um sistema em código Python para uma análise automática, no contexto de um modelo SIR simples, dos dados da epidemia de COVID-19 publicados diariamente pelo Ministério da Saúde. As análises diárias serão publicadas no endereço [1]. O objetivo do projeto é puramente educacional, com foco na análise de dados e programação em Python, e não em epidemiologia. Não obstante, todos os dados tratados aqui são reais e, portanto, os resultados talvez possam ter alguma relevância para se entender a dinâmica real da epidemia de COVID-19, a qual está muito bem analisada, por exemplo, em [2]. Se for citar este sistema ou documento, por favor faça-o como:

- A. Saa, “*Análise automática do Painel Coronavírus*”, 2020. Texto integralmente disponível em <https://vigo.ime.unicamp.br/COVID/covid.pdf>

Todos os códigos e arquivos de dados pertinentes estão disponíveis no repositório [3]. Infelizmente, o autor não pode dar nenhum tipo de suporte para a utilização do sistema, mas incentiva todos interessados a utilizar como quiserem todo o material disponível no repositório.

1 Introdução

Diariamente, o Ministério da Saúde (MS) publica no site [4] alguns dados agregados sobre a epidemia de COVID-19 no País. Infelizmente, contrariando-se as práticas mais elementares para tratamento público de dados, a divulgação é feita diariamente em formato proprietário, especificamente numa

planilha Microsoft Excel. No entanto, este ponto pode ser facilmente sanado, pois há várias ferramentas públicas, como o pacote LibreOffice, que permitem converter a planilha em Excel, por exemplo, em um arquivo de texto CSV (*comma-separated values*), possibilitando sua ampla utilização. Utilizaremos aqui o formato CSV.

De todos os dados publicados pelo MS, utilizaremos apenas o número de casos detectados, tanto em sua versão diária como total acumulado até uma certa data. Estes dados são interpretados no contexto do modelo epidemiológico mais simples: o modelo SIR. O número total de óbitos também é utilizado para a construção de um gráfico de mortes por milhão de habitantes, mas não na análise no contexto do modelo epidemiológico. É interessante notar que, mesmo no contexto bastante simples do modelo SIR, as tendências identificadas para o número básico reprodutivo r_0 são compatíveis com as análises mais detalhadas apresentadas, por exemplo, em [2].

2 Modelo SIR

Há uma vasta literatura sobre o modelo SIR e suas variantes, ver [5], por exemplo. Em sua versão mais simples, que é a empregada aqui, uma população de N indivíduos é dividida em três classe: os suscetíveis a infecção (S), os infectados (I) e os recuperados, ou removidos, (R). A ideia é simples. Os elementos suscetíveis S são aqueles que podem vir a ser infectados a partir do contato com os infectados I . Já os elementos da classe R são aqueles que não mais se infectam e nem vão infectar outros indivíduos, seja por cura com imunidade, seja por que foram afastados. Eventuais óbitos, neste tipo de modelo, são contados na classe R . A dinâmica do modelo é dada pelo sistema de EDO

$$\frac{dS}{dt} = -\frac{\beta IS}{N}, \quad (1)$$

$$\frac{dI}{dt} = \frac{\beta IS}{N} - \gamma I, \quad (2)$$

$$\frac{dR}{dt} = \gamma I, \quad (3)$$

sendo β e γ parâmetros positivos associados, normalmente, às taxas de infecção e de remoção, respectivamente, no modelo. Como admite-se que a população total $N = S + I + R$ é constante, basta escolhermos duas classes

para termos a descrição completa do modelo. Em nosso caso, escolheremos R e I .

Um dos parâmetros mais importantes neste tipo de modelo é o chamado número básico reprodutivo

$$r_0 = \frac{\beta}{\gamma}. \quad (4)$$

Trata-se de um adimensional cuja interpretação mais simples é o número médio de novos casos gerados por um infectado em uma população não imune. A Eq. (2) pode ser escrita como

$$\frac{dI}{dt} = \left(r_0 \frac{S}{N} - 1 \right) \gamma I, \quad (5)$$

de onde podemos inferir alguns comportamentos qualitativos interessantes. Notem, primeiro, que $\frac{S}{N}$ é a fração dos suscetíveis na população e, portanto, está no intervalo $[0, 1]$. É evidente de (5) que, para $r_0 < 1$, o número de infectados irá decrescer monotonicamente. Trata-se da extinção da epidemia. Para $r_0 > 1$, temos um comportamento qualitativo diferente. Para $r_0 \frac{S}{N} > 1$, o número de infectados cresce, implicando a expansão da epidemia. Porém, com a expansão da epidemia, a fração $\frac{S}{N}$ tende a diminuir, desacelerando o ritmo de crescimento do número de infectados. Ao chegar ao valor dado por

$$\frac{S}{N} = \frac{1}{r_0}, \quad (6)$$

o número de infectados I deixará de crescer, e a dinâmica do sistema (1)-(3) implicará na extinção da epidemia. Esta é a situação conhecida como “imunidade de rebanho” (*herd immunity*). Iremos considerá-la do ponto de vista dos indivíduos removidos R . Como $I = 0$ no estágio dominado pela imunidade de rebanho, temos que ele corresponde à situação com

$$n_R = \frac{R}{N} = 1 - \frac{1}{r_0} = \text{constante}. \quad (7)$$

A situação limite $r_0 = 1$ normalmente está associada a fases endêmicas da infecção. Nestas fases, o número de infectados permanece constante no tempo. Para nossos propósitos, basta notarmos que $r_0 > 1$ implica na expansão da epidemia e, quanto maior for o valor de r_0 , mais rápida será a expansão e maior será o coeficiente n_R dado por (7) para a imunidade de rebanho. Obviamente, quanto maior for r_0 , maior será o impacto da epidemia. Por outro

lado, para $r_0 < 1$ a epidemia sempre se extingue, e quanto menor r_0 , mas rápida será esta extinção.

Necessitamos agora interpretar os dados divulgados pelo MS no contexto do modelo SIR. Tomaremos para nossa análise o número acumulado de casos confirmados. Dada a situação específica da epidemia de COVID-19, sabe-se que estes casos correspondem a indivíduos que, ou estão hospitalizados, ou acudiram a um hospital com sintomas moderados. É razoável supor que, a partir deste diagnóstico, eles terminem internados ou liberados com recomendações estritas de isolamento e quarentena. Nessa situação, parece razoável considerá-los como elementos removidos R de nosso sistema, pois não se espera que continuem a propagar a doença, independente do resultado final, seja ela cura e imunização, ou óbito.

A determinação do número de infectados I é um problema muito mais complicado e delicado. Iremos estimá-lo usando a Eq. (3) e supondo γ constante. Os elementos de I são aqueles que estão infectados e transmitem a doença. No caso específico da epidemia de COVID-19, é razoável supor que estes são os infectados assintomáticos e também aqueles com sintomas leves que não são atendidos clinicamente e, portanto, desconhecem suas situações de infectados. Na ausência de testes em massa, estes números devem ser estimados a partir de outras observações. Esta é uma tarefa extremamente complexa e que está completamente fora do escopo deste pequeno projeto. Baseado nas análises apresentadas em [6], usaremos como hipótese que a relação entre os casos detectados e os reais no País está na faixa entre 1 : 10 e 1 : 20, correspondendo a fixar $\frac{1}{20} < \gamma < \frac{1}{10}$ na Eq. (3). Há consistentes indícios de relações entre casos detectados e reais desta mesma ordem de magnitude também em outros países [7]. Supondo-se γ constante, podemos obter a partir de (2) e (3).

$$r_0(t) = \frac{\ddot{R} + \gamma \dot{R}}{\dot{R} \left(\gamma - \frac{\gamma \dot{R} + \ddot{R}}{N} \right)}. \quad (8)$$

Esta expressão para r_0 “instantâneo” será a base de nossa análise. É interessante notar que, enquanto estivermos longe da imunidade de rebanho, é natural admitir que $\frac{\gamma \dot{R} + \ddot{R}}{N} \ll 1$ e, portanto, teremos a aproximação

$$r_0(t) = 1 + \frac{\ddot{R}}{\gamma \dot{R}}. \quad (9)$$

Esta última expressão ajudará a entender porque a tendência de expansão ou extinção da epidemia não deve depender do valor de γ , pelo menos nos estágios longe da imunidade de rebanho.

3 Tratamento dos dados

Nossos dados são séries temporais diárias. Assim, iremos discretizar nossa variável independente $t \in \mathbb{Z}$. Todas as derivadas serão substituídas por diferenças finitas. Optou-se por diferenças atrasadas, *i.e.*

$$\dot{R}_t = R_t - R_{t-1} \quad \text{e} \quad \ddot{R}_t = \dot{R}_t - \dot{R}_{t-1}, \quad (10)$$

mas este ponto parece ter pouca influência no problema. A Fig. 1 mostra os

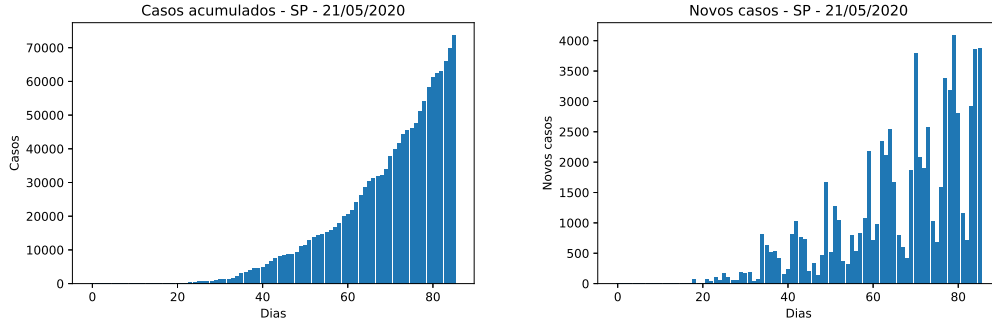


Figura 1: Casos acumulados e novos casos para o estado de São Paulo, obtidos a partir dos dados publicados pelo Ministério da Saúde em [4]. É evidente a presença de “ruído” que impossibilita qualquer análise da variações diretamente destes dados.

casos acumulados (R_t) e os casos novos (\dot{R}_t) para o estado de São Paulo. É evidente destes dados que não conseguiremos realizar nenhuma análise que envolva diferenças finitas, pois há claramente demasiado “ruído” nos dados. Não obstante, é claro que há um evidente regime de crescimento para as duas quantidades. Para podermos estimar r_0 a partir de (8), precisaremos “suavizar” os dados a fim de conseguir, pelo menos, a segunda derivada \ddot{R}_t com um comportamento minimamente suave.

Há diversas técnicas para suavização de dados. Utilizaremos a que talvez seja a mais simples de todas, a dos filtros de média móvel. A ideia subjacente

destes filtros é substituir o valor de um elemento da série temporal por uma média calculada numa vizinhança simétrica com $2n + 1$ elementos, $n \in \mathbb{N}$,

$$\bar{R}_t = \frac{1}{2n + 1} \sum_{k=t-n}^{t+n} R_k. \quad (11)$$

A vizinhança sobre a qual se calcula a média é normalmente chamada de “janela” do filtro. Este tipo de filtro é extremamente atraente para o nosso problema específico, já que um dos ruídos mais comuns em nossos dados é a defasagem de alguns dias na incorporação de novos casos. Por certo, esta é a fonte principal do ruído de período semanal presente na Fig. 1. Vários casos do fim de semana acabam notificados apenas na segunda ou terça-feira. Este tipo de problema pode ser conveniente sanado com filtros do tipo média móvel, com $n = 3$, ou uma semana de janela. Este será nosso filtro padrão.

Uma observação mais atenta de (11) revela que este filtro, como posto, está definido para uma sequência com k elementos apenas para $n < t < k - n$. Nas regiões próximas às “bordas” da sequência, não conseguimos mais definir a vizinhança simétrica e não podemos calcular (11). Para resolver este problema, devemos impor algumas condições sobre as médias móveis próximas as bordas da sequência. Esta é uma questão delicada em qualquer análise de séries temporais, pois estas condições devem ser escolhidas a fim de não comprometer as tendências que queremos identificar. Na prática, devemos estender a sequência R_t acrescentando $2n$ novos elementos, correspondentes a $1 - n < t \leq 0$ e $k < t \leq k + n$. Para o problema em questão, optamos pelas seguintes extensões

$$R_{1-j} = R_1 \text{ e } R_{k+j} = R_k + R_{k-n+j} - R_{k-n}, \quad (12)$$

para $1 \leq j \leq n$. A primeira condição é evidente, admitimos a sequência constante na borda esquerda. A segunda condição parece menos clara, mas é também muito simples. Estamos admitindo que os n elementos R_t da sequência com $k - n < t \leq k$, devidamente deslocados verticalmente, sucedem o último elemento da esquerda da sequência original. Ambas extensões equivalem a supor que as tendências de crescimento, no contexto de um modelo epidemiológico, são preservadas nas bordas da sequência. Assim, podemos definir o filtro (11) para uma sequência R_t para qualquer $1 \leq t \leq n$.

O filtro (11) pode ser visto também como a convolução da função R_t com uma função retangular de altura 1 e largura $2n+1$. É comum iterarmos a ação destes filtros. Por exemplo, duas aplicações sucessivas de um filtro do tipo

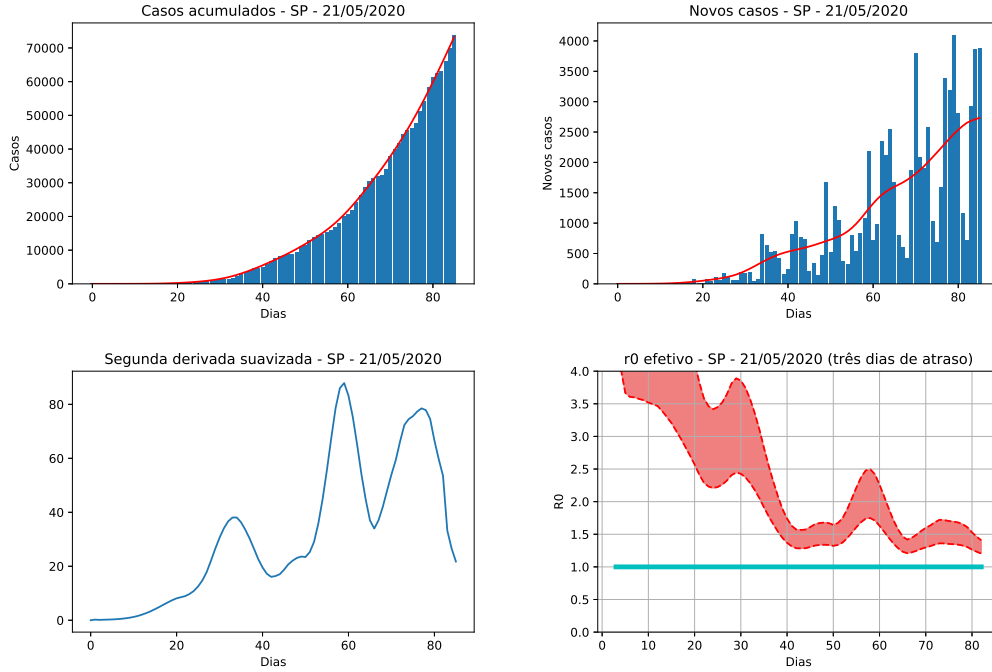


Figura 2: Acima: Casos acumulados e novos casos para o estado de São Paulo, obtidos dos dados publicados pelo Ministério da Saúde em [4], e suavizados com 4 iterações de (11) com $n = 3$, correspondendo a um filtro cúbico com janela de um mês. Abaixo: Segunda derivada \ddot{R}_t , calculada a partir da série suavizada, e r_0 , calculado a partir de (8).

média móvel com janela de $2n + 1$ elementos, corresponderá a convolução da sequência original com uma função triangular com largura $2n + 4$, por isso o nome de filtro triangular para dois usos sucessivos do filtro de média móvel. Uma terceira iteração corresponderia a um filtro de convolução com uma função quadrática e janela $6n + 3$, e assim sucessivamente. Escolhemos para o tratamento de nossos dados o filtro correspondente a 4 iterações de (11) com $n = 3$. Isto corresponde a uma convolução com uma função cúbica sobre uma janela de 4 semanas.

A Fig. 2 mostra os dados da Fig. 1 devidamente suavizados com nosso filtro, assim como o resultado do cálculo de \ddot{R}_t e de r_0 de acordo com (8). O gráfico para r_0 merece mais explicações. A linha horizontal corresponde a $r_0 = 1$. Acima dessa linha, temos expansão da epidemia. Para debelá-la

antes de atingirmos a imunidade de rebanho, devemos sempre ter r_0 abaixo desse limiar. É clara a correlação que existe entre picos de r_0 e regiões de aceleração de casos, basta comparar o gráfico de r_0 com o de novos casos. O gráfico de r_0 apresenta uma região delimitada pelo valor de r_0 calculado a partir de (8) para os limites que consideramos para o parâmetro γ : $\frac{1}{20}$ e $\frac{1}{10}$, já que é razoável supor que o valor de r_0 esteja contido nessa região. Note que, pela natureza dos filtros utilizados, é prudente descartarmos as regiões das bordas. Por isso, o valor de r_0 efetivo sempre é calculado e mostrado com o alerta de estar 3 dias (o valor de n) atrasado.

4 Resultados

Os resultados publicados diariamente em [1] consistem na análise dos dados publicados em [4] para os seguintes casos: Brasil, Estado de São Paulo, Cidade de São Paulo, Cidade de Campinas, e em seguida todos os outros estados do País e o Distrito Federal, ordenados por população. Para cada caso, são calculados e apresentados graficamente os casos novos e acumulados, com suas respectivas suavizações, os valores de r_0 determinados a partir de (8) com $\frac{1}{20} \leq \gamma \leq \frac{1}{10}$, e a previsão para os próximos 5 dias para o número de casos acumulados. Esta previsão é feita a partir de uma regressão linear simples dos últimos 10 dias da série R_t . Também são calculados o r_0 efetivo das últimas 2 semanas como a média simples dos valores calculados a partir de (8) nesse período, e os respectivos limiares de imunidade de rebanho associados, a partir de (7). O código Python das análises, assim como todos os dados, podem ser encontrados em [3]. Na próxima seção, alguns detalhes do código são apresentados.

As tendências identificadas nestas análises são muito mais importantes que os valores numéricos em si, já que em última instância estes valores devem ser entendidos dentro do limitado escopo dos modelos SIR com as grandes incertezas no parâmetro γ . As tendências, por outro lado, podem ser interpretadas como “sinalizadores” para o comportamento da epidemia, e não dependem do parâmetro γ para estágios distantes da imunidade de rebanho, como já pode ser adiantado de (9). Como ilustração destas análises, as Figs. 3 e 4 em anexo apresentam os resultados, com a mesma faixa de valores para γ , para quatro países que estão em estágio mais avançado da epidemia: Espanha, Itália, Estados Unidos e Reino Unido.

É evidente que a contenção da epidemia (decréscimo do numero de novos

casos) ocorre nos períodos com $r_0 < 1$, independentemente do valor preciso de γ . Na ausência de intervenções farmacológicas, a única estratégia para diminuir efetivamente o valor de r_0 é dificultar a propagação do vírus, objetivo cuja maneira mais eficiente de ser alcançado para os casos de vírus respiratórios é diminuindo-se o contato social entre os indivíduos.

5 Alguns detalhes do código Python

O código deste projeto está escrito em Python 3. Para aplicações científicas, recomenda-se fortemente a distribuição Anaconda (versão 3, Python 3.7), disponível para várias plataformas em [8]. O código utiliza apenas recursos padrão e pacotes bem conhecidos. De particular importância para nossos propósitos, são os pacotes `numpy`, `matplotlib.pyplot` e `scipy.stats`, respectivamente, para manipulação numérica de matrizes, análise gráfica e estatística. O código é facilmente legível. Alguns detalhes, porém, merecem alguma atenção.

Iniciemos com a manipulação de arquivos CSV (*comma-separated values*). As instruções no Quadro 1 ilustram a manipulação de arquivos CSV em

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Tue May 26 09:47:38 2020

@author: asaa
"""

import csv
csvfile = open("20200519.csv")
linecsv = csv.reader(csvfile)

for row in linecsv:
    if (row[1] == "SP") and (row[2] == "Campinas"):
        print(row[7], row[10], 1e6 * int(row[10]) / int(row[9]))

csvfile.close()
```

Quadro 1: Exemplo de manipulação de arquivos CSV em Python. Arquivo `csv-example.py` disponível em [3].

Python, da maneira que usamos em nosso código. O objeto `linecsv` é varrido no loop e retorna a lista `row` para cada linha do arquivo `20200519.csv`. Os elementos desta lista são as strings correspondentes a cada uma das colunas, que estão sempre separadas por vírgulas no formato CSV padrão. Nesse caso em particular, lê-se o arquivo completamente, seleciona-se as linhas correspondentes ao estado de SP (segunda coluna do arquivo) e cidade de Campinas (terceira coluna), e imprime-se a data (coluna 8), o número de casos (coluna 11) e a razão casos por milhão de habitantes. A coluna 10 contém o número de habitantes da cidade. Como todas estas colunas tem formato de string, é necessário convertê-las para valores numéricos, inteiros neste caso, para poder fazer manipulações matemáticas.

No Quadro 2, temos a implementação do nosso filtro de média móvel. A

```
import numpy as np

def smooth(Y,n):
    k = Y.size
    Y_smooth = np.zeros(k)

    Y_edge = np.concatenate((Y[0]*np.ones(n),Y,Y[k-n:k]+Y[k-1]-Y[k-n-1]))

    for i in range (0,k):
        Y_smooth[i] = np.sum(Y_edge[i:i+2*n+1])/(2*n+1)

    return Y_smooth
```

Quadro 2: Filtro de média móvel com a extensão dos dados iniciais, de acordo com o discutido na Seção 3.

suavização utilizada corresponde a 4 iterações desse filtro. Conforme descrito na Seção 3, a implementação deste tipo de filtro requer a extensão dos dados iniciais em n elementos, tanto à “esquerda” como à “direita”. No Quadro 2, mostra-se a implementação da extensão discutida na Seção 3. Note que este filtro requer $k \geq n$. Testou-se também o filtro de Savitzky-Golay, disponível como `savgol_filter` no pacote `scipy.signal`, mas os resultados foram inferiores.

O último aspecto que merece ser discutido sobre o código são os gráficos interativos. Eles são implementados em JavaScript, utilizando bibliotecas públicas Google. Os arquivos html contendo o scrip são gerados na execução

do código em Python. O Quadro 3 tem um exemplo simples do JavaScript necessário para se criar um gráfico interativo como os do projeto.

```
<!DOCTYPE HTML>
<html> <head>
<script type="text/javascript"
    src="https://www.gstatic.com/charts/loader.js"></script>
<script>
google.charts.load('current', {packages: ['corechart', 'line']} );
google.charts.setOnLoadCallback(drawChart);

function drawChart() {
    var data = new google.visualization.DataTable();
    data.addColumn('number', 'x');
    data.addColumn('number', 'Curva A');
    data.addColumn('number', 'Curva B');
    data.addRows([
        [0.0,0.0,0.0],[1.0,1.0,2.0],[2.0,2.0,3.0],[3.0,2.0,2.0],]);

    var options = {title: 'Duas curvas', width: 900, height: 700,
        series: {0: { lineWidth: 1 }, 1: { lineWidth: 1 }},
        hAxis: {title: 'x'}, vAxis: {title: 'f(x)'};

    var chart = new
        google.visualization.LineChart(document.getElementById('chart_div'));
    google.visualization.events.addListener(chart, 'select', function()
        { highlightLine(chart,data, options); });
    chart.draw(data, options);
}

function highlightLine(chart,data,options) {
    var selectedLineWidth = 5;
    var selectedItem = chart.getSelection()[0];
    for(var i in options.series) {options.series[i].lineWidth = 1;}
    options.series[selectedItem.column-1].lineWidth = selectedLineWidth;
    chart.draw(data, options); }

</script></head>
<div id="chart_div"></div>
```

Quadro 3: Exemplo de JavaScript utilizado para os gráficos interativos. Arquivo graf.html disponível em [3].

Referências

- [1] <http://vigo.ime.unicamp.br/COVID>
- [2] <https://covid19br.github.io/>
- [3] <https://github.com/albertosaa/COVID>
- [4] <https://covid.saude.gov.br/>
- [5] https://en.wikipedia.org/wiki/Compartmental_models_in_epidemiology
- [6] <https://ciis.fmrp.usp.br/covid19/>
- [7] <https://elpais.com/sociedad/2020-04-07/mas-del-90-de-contagios-estan-ocultos.html>
- [8] <https://anaconda.org/>

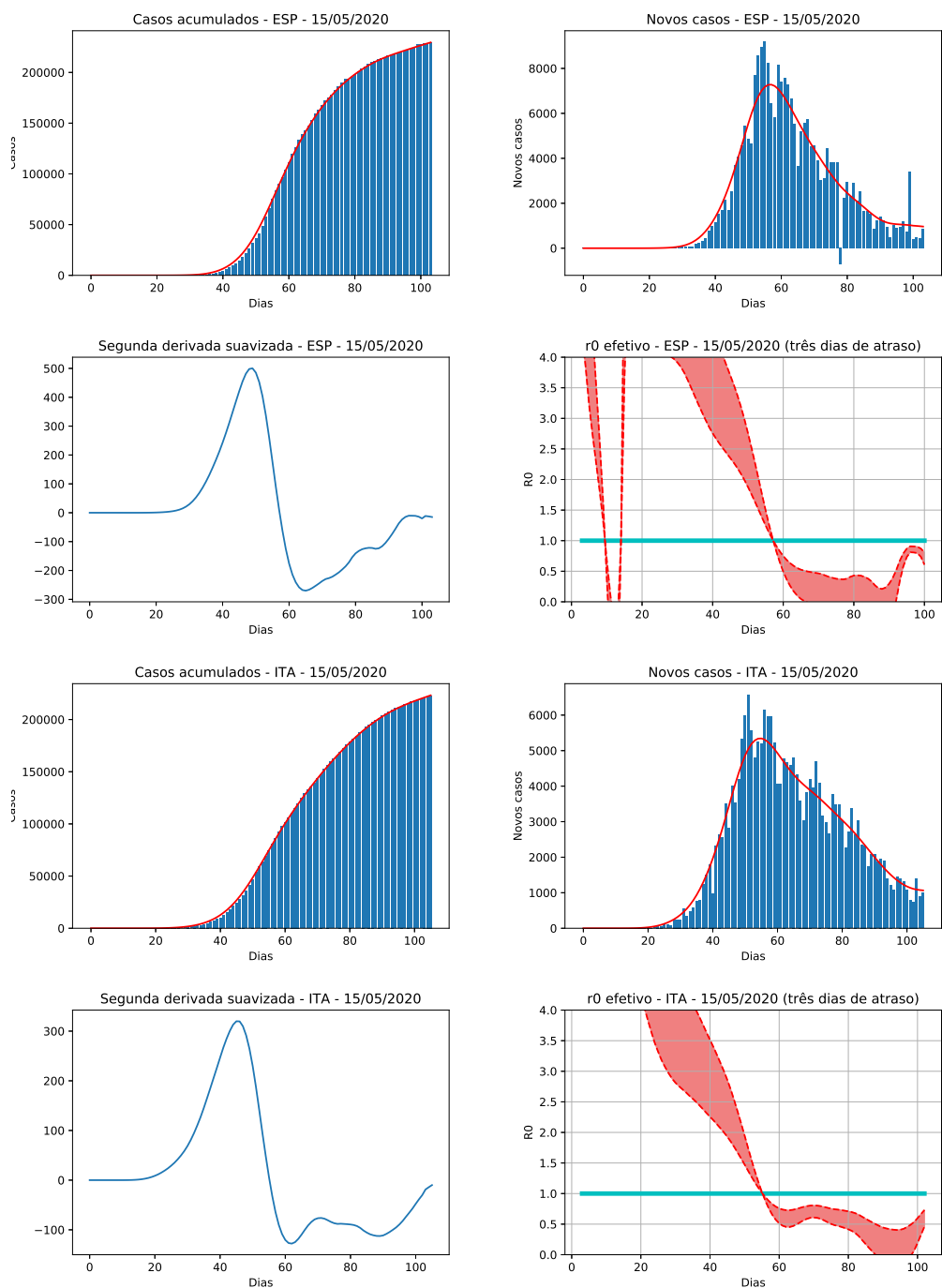


Figura 3: Análises para Espanha e Itália.

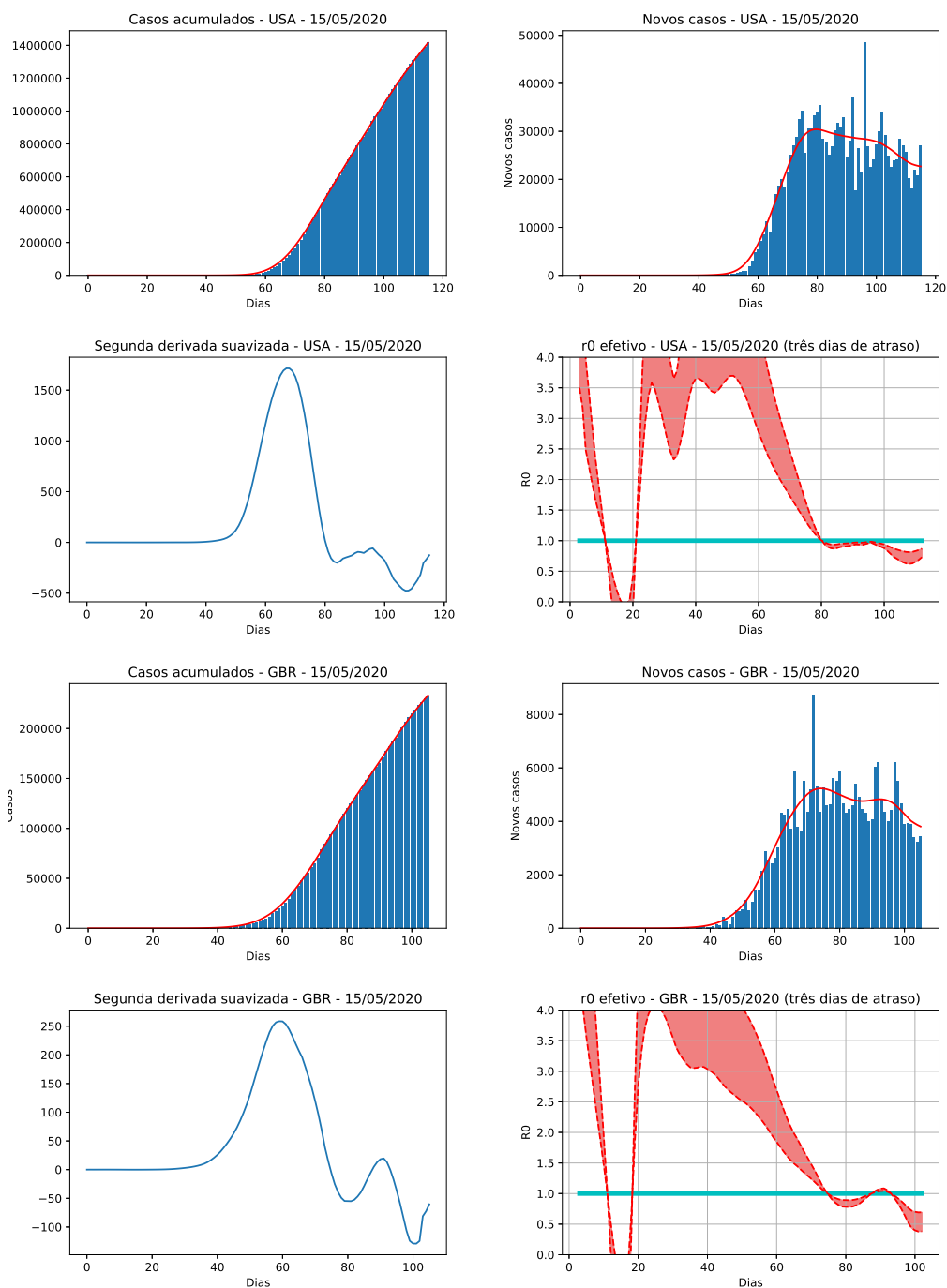


Figura 4: Análises para Estados Unidos e Reino Unido.