

Modelo SIR aplicado ao COVID-19

Bianca Natal

Eleonora Avello

Gustavo Martins

Gustavo Rodrigues

Sumário

1	Introdução	3
2	Justificativa	4
3	Objetivo	5
4	Modelagem matemática na pandemia de COVID-19	6
5	Desenvolvimento	7
6	A implementação do Método de Runge-Kutta de ordem 4	9
7	Coleta de dados	11
8	Aproximação de Função	15
8.1	Estudo do mês de Julho	15
8.2	Estudo do mês de Novembro	16
9	Taxas de crescimento e recuperação	18
9.1	Dinâmica do Modelo SIR	18
9.2	O número reprodutivo basal R_0	19
10	Código	21
11	Conclusões	25
12	Referências	29

1

Introdução

O mundo tem sofrido muito com as epidemias ao longo da história. E a facilidade em termos de viagens internacionais facilita que epidemias tornem-se pandemias, como foi o caso da gripe espanhola em 1918. Coincidentemente, foi justamente esta facilidade de ir e vir que fez com que a COVID-19, surgida na China no final de 2019, se tornasse, em março de 2020, uma pandemia. Apesar de ser uma temática muito recente, já há um vasto material bibliográfico sobre a modelagem matemática sobre a dinâmica da contaminação do coronavírus ou SARS-CoV-2.

A maioria das modelagens nesta temática parte do modelo SIR (Suscetíveis, Infectados, Removidos), onde indivíduos que contraem a doença deixam o status de suscetíveis e passam a integrar o status de infectados e deste, passam a ser removidos. Observa-se que, no status de removidos estão os indivíduos que vieram a se recuperar da doença e os vieram a óbito. Este modelo foi proposto originalmente por Kermarck e McKendrick em 1927.

Justificativa

Vivemos desde o mês de março a Pandemia da Covid-19 no Brasil, sendo que a Secretaria Municipal da Saúde (SMS) de Curitiba registrou o primeiro caso na cidade no dia 11 de março. Com o avanço da Pandemia e o aumento do número de casos foi estipulado o Protocolo de Responsabilidade Sanitária e Social, onde existem três categorias, ou bandeiras. A primeira bandeira é a Amarela que é considerada um sinal de alerta constantes e orientações a serem seguidas. A segunda bandeira é a Laranja onde são impostas restrições no funcionamento de serviços e do comércio e também de áreas onde podem ser propiciadas aglomerações de pessoas. A última bandeira é a Vermelha, que é o nível de alerta total onde há restrição da circulação de pessoas e apenas os serviços essenciais podem funcionar.

Abaixo, temos as informações presentes no site da Prefeitura de Curitiba e para maiores detalhes dos indicadores é possível encontrar através link < [https : //coronavirus.curitiba.pr.gov.br/](https://coronavirus.curitiba.pr.gov.br/) >



Figura 2.1: Protocolo de Responsabilidade Sanitária e Social

3

Objetivo

Fazer a análise da interferência da bandeira em vigência em relação ao número de casos ativos diários. O modelo a ser utilizado é o Modelo SIR (Suscetível-Infectado-Recuperado) que tem sido amplamente utilizado para modelar a COVID-19. Neste modelo, S é o número de habitantes suscetíveis que podem entrar em contato com o coronavírus. I é o número de infectados e R o número de removidos/recuperados. N_h corresponde ao total de habitantes, e considera-se que $N_h = S + I + R$, de modo que não são considerados nascimentos ou mortes de causas naturais que modificam a quantidade de suscetíveis S . Esta estrutura requer dois parâmetros: β (taxa de crescimento) e γ (taxa de recuperação/removidos). Abaixo estão as seguintes equações diferenciais que descreve este modelo SIR:

$$\left\{ \begin{array}{l} \frac{dS}{dt} = -\frac{\beta \cdot I \cdot S}{N} \\ \frac{dI}{dt} = \frac{\beta \cdot I \cdot S}{N} - \gamma \cdot I \\ \frac{dR}{dt} = \gamma \cdot I \end{array} \right.$$

O parâmetro γ é equivalente ao inverso do tempo de infecção, ou seja, $\gamma = \frac{1}{T_{inf}}$. É fato que T_{inf} é visto como período no qual o infectado está infeccioso, ou seja, pode transmitir a doença para outras pessoas. Um parâmetro muito importante também em qualquer que seja a epidemia é o que se convencionou chamar de índice de reprodução basal, que é definido como a taxa de crescimento vezes o tempo de infecção: $R_0 = \beta T_{inf} = \frac{\beta}{\gamma}$. Para $S = S_0 = N_h$ e $I = I_0$ no tempo inicial t_0 é demonstrado que $I = e^{|\beta - \gamma|t} I_0$ na fase inicial da pandemia. Logo, se $\beta > \gamma \rightarrow R_0 > 1$, então haverá um crescimento exponencial na epidemia no início do processo de contágio. Se $R_0 < 1$, haverá então um decrescimento exponencial, até acabar a epidemia. Com isso, podemos concluir que o parâmetro R_0 é fundamental para o controle e entendimento da epidemia.

Modelagem matemática na pandemia de COVID-19

Os modelos matemáticos, segundo a publicação "Modelos matemáticos serão estratégicos no combate ao coronavírus", da Sociedade Brasileira de Medicina Tropical (SBMT), permitem saber como será a disseminação de determinada doença e estimar o número de infectados, mortes e hospitalizações. Além disso, essas ferramentas permitem a simulação de inúmeros cenários, o que pode projetar a eficiência - ou a ineficiência - de determinada intervenção dos órgãos públicos responsáveis pela saúde.

Apesar de serem ferramentas úteis, os modelos matemáticos têm suas limitações. Particularmente, as modelagens envolvidas na COVID-19 contam com dados que nem sempre são exatos. Um exemplo disso é o número de infectados assintomáticos, que também são transmissores da doença, mas boa parte desse número não é contabilizado e, conseqüentemente, não é levado em conta em alguns modelos - como é o caso do modelo SIR, com o qual trabalharemos.

Visto isso, é importante ressaltar que, apesar de ser uma ferramenta muito importante, os modelos matemáticos podem estimar um cenário futuro que, posteriormente, não será concretizado por conta de inúmeros fatores. Surge disso a ideia desta modelagem: identificar um mês cuja bandeira em Curitiba é igual à atual, usar o modelo SIR nesse mês anterior e, em seguida, usar esse modelo para estimar um cenário futuro. Dessa forma, podemos testar a eficácia do modelo a ser usado, fazer uma estimativa futura e apresentar, mais a seguir, o motivo pelo qual Curitiba não deveria estar na bandeira amarela no mês de novembro, mas na bandeira laranja.

5

Desenvolvimento

Como vimos anteriormente, o Modelo SIR é dado pelo seguinte sistema de Equações Diferenciais:

$$\begin{cases} \frac{dS}{dt} = -\frac{\beta \cdot I \cdot S}{N} \\ \frac{dI}{dt} = \frac{\beta \cdot I \cdot S}{N} - \gamma \cdot I \\ \frac{dR}{dt} = \gamma \cdot I \end{cases}$$

Para a implementação do código em Julia, primeiramente foi optado por utilizar um Pacote para resolver as Equações Diferenciais. Devido a problemas para a instalação do pacote, o grupo decidiu então por utilizar outra forma para implementar o código. Após algumas pesquisas em busca de Métodos Numéricos para Equações Diferenciais, encontramos o Método de Euler.

O Método de Euler é um método de primeira ordem para solucionar equações diferenciais ordinárias com um valor inicial dado considerando o Problema de Valor Inicial

$$\begin{cases} y' = f(x, y(x)) \\ y(x_0) = y_0 \end{cases}$$

e é baseado na expansão da função $y(x)$ em séries de Taylor e com isso, é possível expandir a função $y(x)$ na vizinhança do ponto x_n até a ordem 1. Escolhendo para o tamanho de cada passo um valor h e atribuindo a cada passo um ponto dentro do intervalo, temos então que $x_n = x_0 + n \cdot h$ e ainda, $x_{n+1} = x_n + h$. Logo, $y_{n+1} = y_n + h \cdot f(x_n, y_n)$. Portanto, o Método de Euler é dado por

$$y_{n+1} = y_n + h \cdot f(x_n, y_n).$$

Ainda, com o auxílio da aula de "Equações diferenciais - Problemas de Valor Inicial 2 - Sistemas" do professor Abel Siqueira para a disciplina de Cálculo Numérico em Julia, vimos o Método de Runge-Kutta de quarta ordem que também é utilizado na solução numérica de problemas com equações diferenciais ordinárias.

Novamente, considerando o problema de valor inicial abaixo

$$\begin{cases} y' = f(x, y(x)) \\ y(x_0) = y_0 \end{cases}$$

temos que o Método de Runge-Kutta de quarta ordem é dado por

$$\begin{cases} y_{n+1} = y_n + \frac{h}{6} \cdot (k_1 + 2k_2 + 2k_3 + k_4) \\ x_{n+1} = x_n + h \end{cases}$$

onde y_{n+1} é a aproximação por Runge-Kutta de quarta ordem de $y(x_{n+1})$ e

$$\begin{aligned} k_1 &= f(x_n, y_n) \\ k_2 &= f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2} \cdot k_1\right) \\ k_3 &= f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2} \cdot k_2\right) \\ k_4 &= f(x_n + h, y_n + h \cdot k_3) \end{aligned}$$

E como temos um sistema, utilizamos o método com notação vetorial. Além disso, tem-se que o erro dado pela aproximação pelo Método de Runge-Kutta é menor que o erro dado pelo Método de Euler. É devido ao fato de o Método de Runge-Kutta, método ao qual focaremos para a implementação do Modelo SIR, apresentar esse menor erro em relação ao Método de Euler, que a seguir tem-se uma seção voltada apenas para a implementação do Método de Runge-Kutta.

6

A implementação do Método de Runge-Kutta de ordem 4

Esta seção apresentará, de forma detalhada, o modo de funcionamento do Método de Runge-Kutta de ordem 4, bem como os procedimentos de cada iteração do método.

Suponha que, como problema de valor inicial, temos a equação diferencial de primeira ordem e o ponto inicial (x_0, y_0) :

$$\begin{cases} y' = g(x, y) \\ y(x_0) = y_0 \end{cases}$$

Descreveremos, a partir disso, a primeira iteração do método para encontrar o ponto (x_1, y_1) do gráfico. As demais iterações serão idênticas a essa primeira.

A partir do ponto inicial (x_0, y_0) , a abscissa x_1 do próximo ponto do gráfico é obtida fazendo $x_1 = x_0 + h$, onde h é o passo, definido pelo usuário do método.

O problema principal de cada iteração é determinar a ordenada do próximo ponto. Conhecendo y_0 , determinamos a próxima ordenada y_1 fazendo $y_1 = y_0 + kh$, onde k é a inclinação de uma reta que passa pelo ponto anterior (x_0, y_0) .

Sabendo disso, devemos encontrar essa inclinação k . Nesse método, a inclinação k é a média ponderada de quatro outras inclinações k_1 , k_2 , k_3 e k_4 . A seguir, temos uma breve explicação do que cada inclinação representa e a forma como calculamos cada uma.

- **1ª inclinação k_1 :** k_1 é a inclinação da reta tangente à curva solução no ponto (x_0, y_0) . Ela é dada por:

$$k_1 = g(x, y)$$

A partir de k_1 , determinamos o ponto $p_1 = \left(x_0 + \frac{h}{2}, y_0 + k_1 \cdot \frac{h}{2}\right)$

- **2ª inclinação k_2 :** para calcular k_2 , tomamos a metade do passo h , que é $\frac{h}{2}$, e fazemos:

$$k_2 = g(p_1)$$

Com o valor de k_2 , determinamos o ponto $p_2 = \left(x_0 + \frac{h}{2}, y_0 + k_2 \cdot \frac{h}{2}\right)$.

- **3ª inclinação k_3 :** agora, calculamos k_3 fazendo:

$$k_3 = g(p_2)$$

Com a inclinação k_3 , determinamos o ponto $p_3 = (x_0 + h, y_0 + k_3h)$.

- **4ª inclinação k_4 :** por último, determinamos k_4 :

$$k_4 = g(p_3)$$

Agora, determinamos k que, como foi dito anteriormente, é a média ponderada entre as quatro primeiras inclinações, onde k_1 e k_4 têm peso 1 e k_2 e k_3 têm peso 2, ou seja:

$$k = \frac{1}{6} \cdot (k_1 + 2k_2 + 2k_3 + k_4)$$

Conhecendo a inclinação k , podemos - finalmente - determinar $y_1 = y_0 + kh$ e, sabendo que $x_1 = x_0 + h$, temos o próximo ponto:

$$(x_1, y_1) = (x_0 + h, y_0 + kh)$$

Assim, finalizamos a primeira iteração do método. As demais iterações serão idênticas. Com isso, obtemos as equações presentes no fim da seção anterior.

7

Coleta de dados

No início das pesquisas, em busca dos dados a serem utilizados, buscávamos por gráficos separados para cada mês. Porém, são divulgados gráficos com os dados do início da pandemia até hoje, dividido por semanas epidemiológicas. E o que queríamos eram os dados de cada dia dos meses escolhidos, julho e novembro, foi necessário então buscar os dados dia a dia.

A seguir temos duas tabelas feitas a partir da coleta de dados. A primeira tabela possui os dados de novos casos e novos óbitos diários, o total de confirmados, casos em investigação, recuperados, total de óbitos e a bandeira vigente no mês de julho. A segunda tabela possui esses dados referentes ao mês de novembro. Todos os dados foram retirados do site da Prefeitura de Curitiba e serão utilizados posteriormente.

Mais abaixo, estão os gráficos desses dados para uma melhor visualização.

Julho							
Dia	Novos casos	Novos óbitos	Conf.	Invest.	Recup.	Óbitos	Bandeira
1	345	4	5.523	-	-	152	
2	434	15	5.957	522	2.909	167	
3	411	5	6.368	-	-	172	
4	119	10	6.487	-	-	182	
5	348	7	6.835	-	-	189	
6	695	6	7.530	-	-	195	
7	256	11	7.786	657	4.145	206	
8	640	10	8.425	-	-	216	
9	418	10	8.844	-	-	226	
10	454	7	9.298	-	-	233	
11	367	9	9.665	522	5.883	242	
12	184	11	9.849	530	5.735	253	
13	591	14	10.440	583	5.784	267	
14	233	20	10.673	-	-	287	
15	717	12	11.390	-	-	299	Laranja
16	768	11	12.158	526	6.717	310	
17	465	13	12.623	-	-	323	Laranja
18	468	13	13.091	542	7.134	336	
19	307	14	13.398	533	7.343	350	
20	537	19	13.935	554	7.710	369	
21	467	18	14.402	520	7.915	387	
22	662	17	15.064	537	8.275	404	Laranja
23	640	18	15.704	482	8.341	422	
24	720	22	16.424	-	8.407	444	
25	364	16	16.788	549	8.533	460	
26	-	-	-	-	-	-	
27	295	15	17.328	529	10.334	480	Laranja
28	392	11	17.720	533	10.944	491	
29	411	13	18.131	570	11.533	504	
30	496	20	18.627	595	12.413	524	
31	699	24	19.326	562	12.991	548	

Tabela 7.1: Dados da Covid-19 no mês de Julho na cidade de Curitiba

Novembro							
Dia	Novos casos	Novos óbitos	Conf.	Ativos	Recup.	Óbitos	Bandeira
1	-	-	-	-	-	-	Amarela
2	-	-	-	-	-	-	Amarela
3	896	10	53.785	3.965	48.330	1.490	Amarela
4	331	6	54.116	3.935	48.685	1.496	Amarela
5	407	8	54.523	4.012	49.007	1.504	Amarela
6	436	8	54.959	4.131	49.316	1.512	Amarela
7	473	9	55.432	4.236	49.675	1.521	Amarela
8	-	-	-	-	-	-	Amarela
9	536	7	55.968	4.395	-	1.528	Amarela
10	477	5	56.445	4.431	50.481	1.533	Amarela
11	736	5	57.181	4.763	50.880	1.538	Amarela
12	703	10	57.884	5.004	51.331	1.548	Amarela
13	749	5	58.633	5.306	51.744	1.553	Amarela
14	715	6	59.348	5.705	52.084	1.559	Amarela
15	-	-	-	-	-	-	Amarela
16	1.508	10	60.856	6.849	52.438	1.569	Amarela
17	879	13	61.735	7.449	52.704	1.582	Amarela
18	914	11	62.649	7.714	53.342	1.593	Amarela
19	1.381	9	64.030	8.415	54.013	1.602	Amarela
20	-	-	-	-	-	-	Amarela
21	1.380	8	66.819	9.637	55.561	1.621	Amarela
22	984	7	67.803	10.224	55.951	1.628	Amarela
23	1.339	10	69.142	11.323	56.272	1.638	Amarela
24	1.101	11	70.243	11.500	57.094	1.649	Amarela
25	1.597	11	71.840	12.139	58.041	1.660	Amarela
26	1.604	18	73.444	12.784	58.982	1.678	Amarela
27	1.571	16	75.015	12.973	60.348	1.694	Laranja
28	1.521	17	76.536	13.320	61.505	1.711	Laranja
29	970	18	77.506	13.582	62.195	1.729	Laranja
30	1.254	16	78.760	13.829	63.186	1.745	Laranja

Tabela 7.2: Dados da Covid-19 no mês de Novembro na cidade de Curitiba

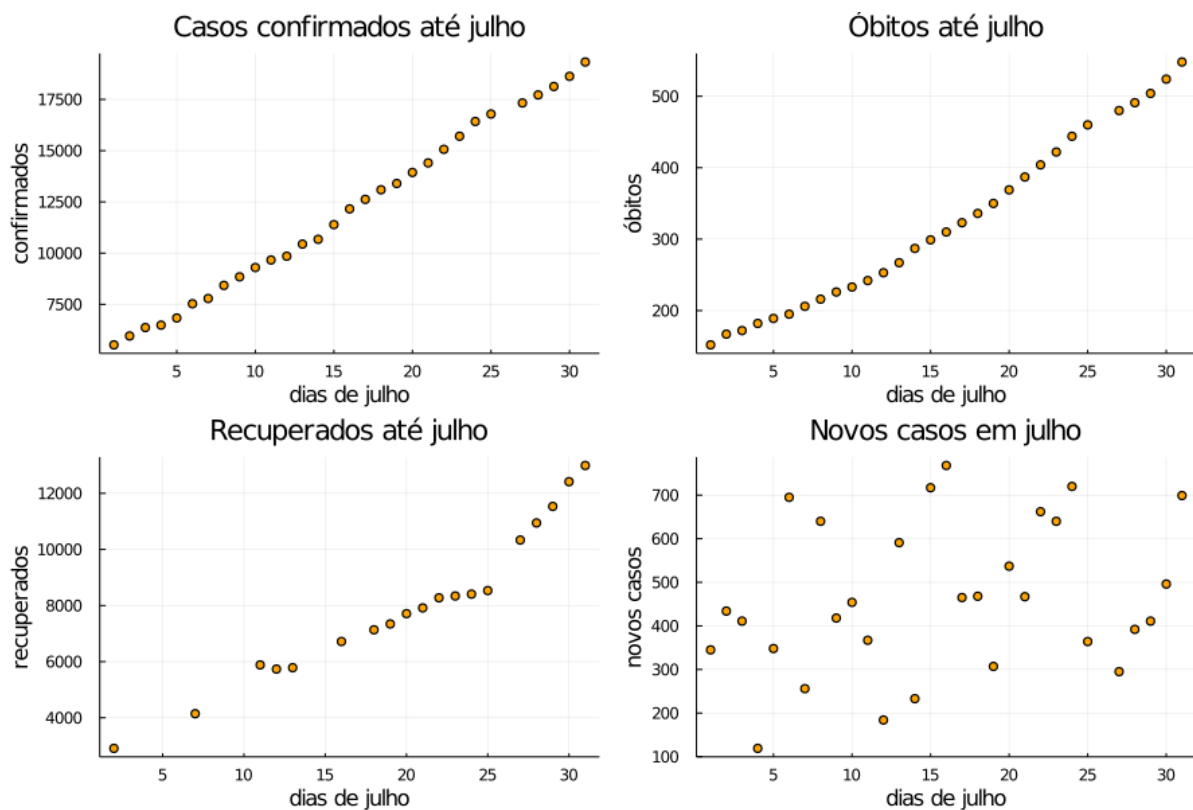


Figura 7.1: Dados do mês de julho da COVID-19 em Curitiba

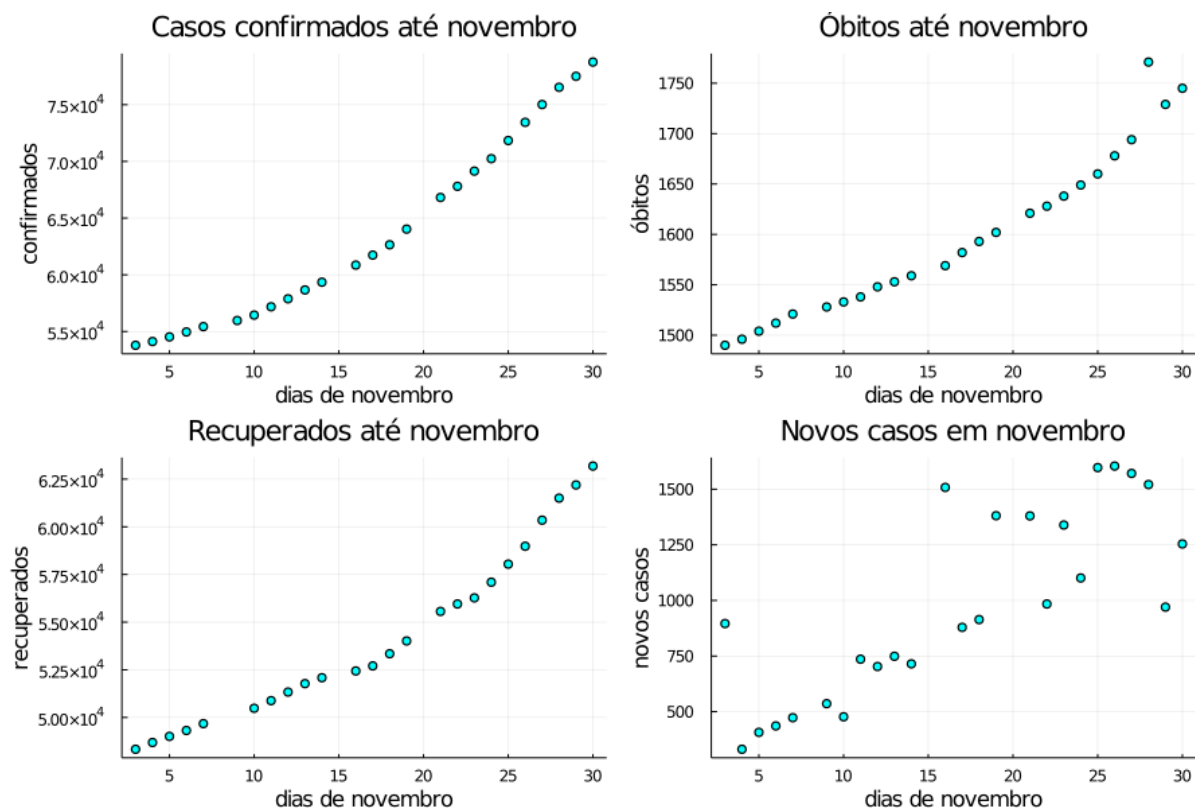


Figura 7.2: Dados do mês de Novembro da COVID-19 em Curitiba

Aproximação de Função

Visto que o Modelo SIR faz a análise da disseminação da doença na população e uma previsão do avanço da pandemia, fizemos a aproximação da função dos dados obtidos dos meses de julho e novembro. Para uma melhor análise dos dados obtidos, foi utilizando uma calculadora que faz a Aproximação de Função com Análise de Regressão e para cada grupo de dados, foi selecionada a função com menor erro. Para a obtenção da aproximação de função, foram usados como valores para X os dias e os valores para Y os respectivos dados.

8.1 Estudo do mês de Julho

No mês de Julho tínhamos dados relacionados aos novos casos, novos óbitos, total de confirmados, casos em investigação, total de recuperados e o total de óbitos. Sendo que o foco para estudo foram os casos confirmados e recuperados até o mês de julho, o total de óbitos até esse mês e os novos casos registrados. Ainda, foi levado em consideração a falta de dados em alguns dias do mês.

Com o auxílio da calculadora, obtemos a seguinte função para os casos confirmados até o mês de Julho:

$$f(x) = -0,2321 \cdot x^3 + 12,6913 \cdot x^2 + 275,1272 \cdot x + 5329,4509,$$

com coeficiente de correlação 0,9992, coeficiente de determinação 0,9984 e erro relativo médio 1,3179%.

A função obtida para o total de óbitos registrados até o mês de julho foi:

$$f(x) = -0,0067 \cdot x^3 + 0,4968 \cdot x^2 + 3,4556 \cdot x + 156,5620,$$

com coeficiente de correlação 0,9992, coeficiente de determinação 0,9984 e erro relativo médio 1,3083%.

Função para o total de recuperados até o mês de julho:

$$f(x) = 0,7293 \cdot x^3 - 30,2455 \cdot x^2 + 607,0170 \cdot x + 1616,3719,$$

com coeficiente de correlação 0,9949, coeficiente de determinação 0,9899 e erro relativo médio 3,1326%.

E por fim, a função para os dados de novos casos confirmados em julho:

$$f(x) = e^{5,3849+0,0143 \cdot x},$$

com coeficiente de correlação 0,1697, coeficiente de determinação 0,0288 e erro relativo médio 35,7916%.

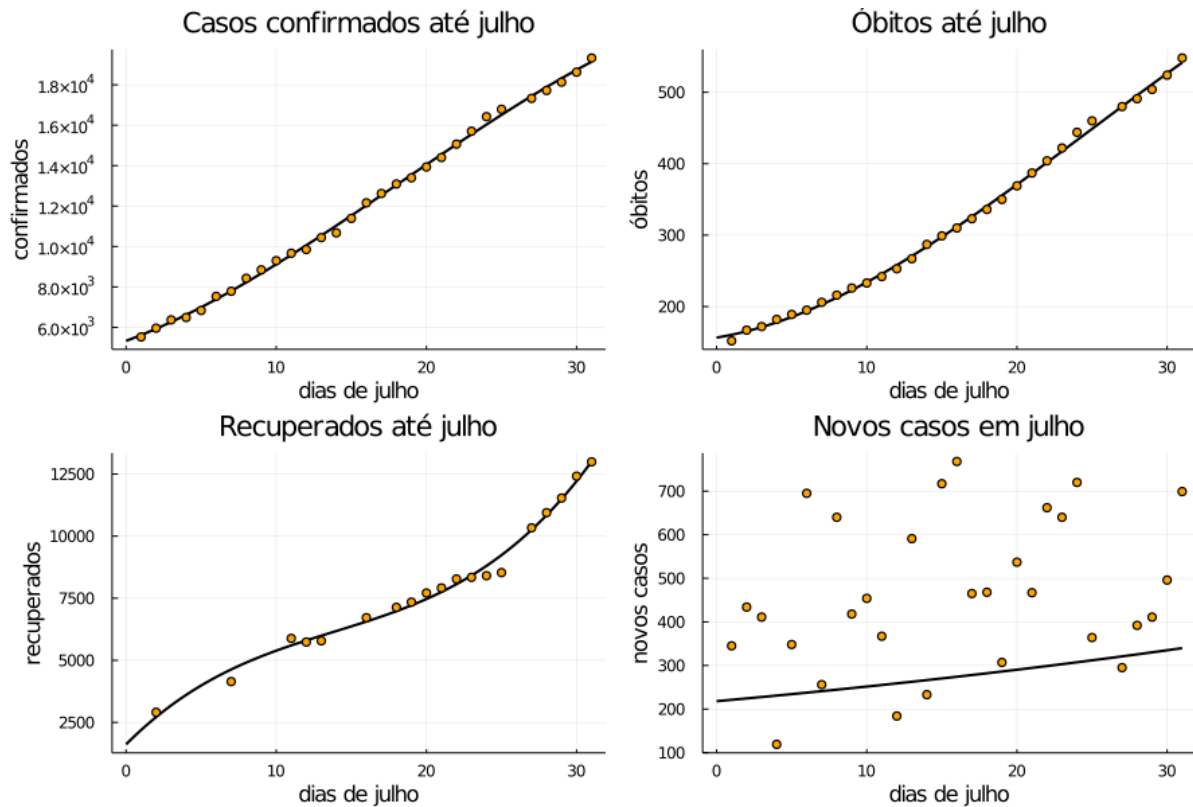


Figura 8.1: Aproximação de função dos dados do mês de julho da COVID-19 em Curitiba

8.2 Estudo do mês de Novembro

No mês de Novembro tínhamos dados relacionados aos novos casos, novos óbitos, total de confirmados, casos ativos, total de recuperados e o total de óbitos. Sendo que o foco para estudo foram os casos confirmados e recuperados até o mês de novembro, o total de óbitos até esse mês e os novos casos registrados. Também foi levado em consideração a falta de dados em alguns dias desse mês.

A seguinte função é a aproximação para os dados de casos confirmados até o mês de Novembro:

$$f(x) = -0,2482 \cdot x^3 + 40,2731 \cdot x^2 - 149,7415 \cdot x + 54212,1593,$$

com coeficiente de correlação 0,9995, coeficiente de determinação 0,9990 e erro relativo médio 0,3201%.

A função obtida para o total de óbitos registrados até o mês de novembro foi:

$$f(x) = 0,0123 \cdot x^3 - 0,3454 \cdot x^2 + 9,0592 \cdot x + 1466,1374,$$

com coeficiente de correlação 0,9895, coeficiente de determinação 0,9791 e erro relativo médio 0,3344%.

Função para o total de recuperados até o mês de novembro:

$$f(x) = 0,7414 \cdot x^3 - 19,8918 \cdot x^2 + 477,4610 \cdot x + 47048,1630,$$

com coeficiente de correlação 0,9990, coeficiente de determinação 0,9979 e erro relativo médio 0,02787%.

E por fim, a função para os dados de novos casos confirmados em novembro:

$$f(x) = -0,2966 \cdot x^3 + 14,1247 \cdot x^2 - 148,0600 \cdot x + 934,1163,$$

com coeficiente de correlação 0,8687, coeficiente de determinação 0,7547 e erro relativo médio 17,6437%.

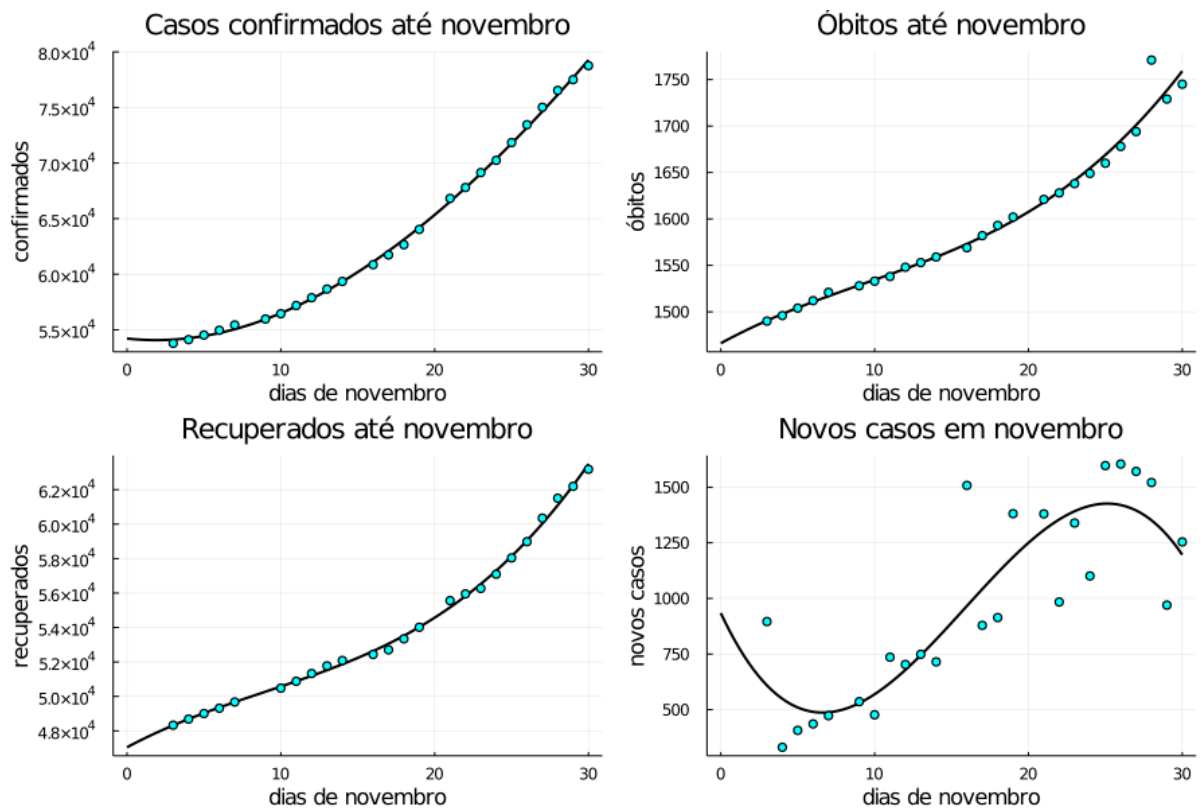


Figura 8.2: Aproximação de função dos dados do mês de Novembro da COVID-19 em Curitiba

9

Taxas de crescimento e recuperação

9.1 Dinâmica do Modelo SIR

A dinâmica do Modelo SIR é ditada, principalmente, pela número reprodutivo basal R_0 e o tempo de infecção, que representa quanto tempo uma pessoa fica, em média, infectada antes de passar da classe dos infectados para a classe dos removidos.

Na prática, porém, usamos duas outras taxas:

- Taxa de transmissão (β): é obtida à partir dos valores R_0 e γ ;
- Taxa de recuperação (γ): essa taxa é igual à razão entre 1 e o tempo de infecção, que denota quanto tempo, em média, uma pessoa fica infectada pela doença. Ou seja:

$$\gamma = \frac{1}{\text{tempo de infecção}}$$

A relação entre essas três taxas é dada por:

$$R_0 = \frac{\beta}{\gamma}$$

Ou seja, na prática, podemos determinar a taxa de transmissão β fazendo:

$$\beta = R_0 \cdot \gamma$$

Para aplicar o Modelo SIR com os dados coletados nos meses de julho e novembro, foi necessário - primeiramente - determinar a taxa de recuperação γ e encontrar o número reprodutivo basal R_0 , para obter a taxa de transmissão β .

De acordo com a Organização Mundial da Saúde, a maior parte dos casos de pessoas infectadas pelo coronavírus apresentam sintomas leves. Nesse caso, a recuperação ocorre, em média, em 14 dias. Em vista disso, para determinar γ , foi adotado que o tempo médio de infecção pelo coronavírus é de 14 dias. Ou seja: $\gamma = \frac{1}{14}$.

Feito isso, o próximo passo foi encontrar o valor de R_0 para determinar a taxa de transmissão β . Essa foi uma das partes mais trabalhosas deste projeto. De acordo com o artigo "A simple statistical physics model for the epidemic with incubation period", publicado em abril de 2020, estima-se que - no caso da Covid-19 - o valor de R_0 é tal que $2 < R_0 < 4$, mas esse valor pode ser até maior do que 4 nas grandes metrópoles com grandes concentrações de pessoas.

Tendo em vista a dificuldade de encontrar um valor exato de R_0 para a cidade de Curitiba, esse assunto será abordado separadamente na seção seguinte.

9.2 O número reprodutivo basal R_0

Segundo Monich (2018, p. 9),

A reprodutividade basal é utilizada para entender a dinâmica da infecção, pois se $R_0 > 1$, então a infecção se espalhará e, se $R_0 < 1$, o número de infectados vai reduzir e a doença se disseminará. Isso ocorre pois sua fórmula é dada pela divisão da taxa de transmissão da doença pela taxa de recuperação.

Ou seja, se a taxa de transmissão β é maior do que a taxa de recuperação γ , temos um número reprodutivo basal $R_0 > 1$, o que indica que a doença está se transmitindo mais rápido do que o tempo necessário para que os infectados se recuperem, causando a propagação da doença. Por outro lado, com $\beta < \gamma$, temos $R_0 < 1$, um indicador de que os infectados estão se recuperando mais rapidamente em relação ao número de novas infecções e, portanto, a propagação está diminuindo.

No Modelo SIR, o valor de R_0 interfere diretamente na previsão de um cenário futuro, de modo que uma pequena variação nesse número pode acarretar uma enorme diferença na estimativa do número de contaminados futuramente. Por essa razão, foi tomado um cuidado especial na procura do R_0 para a modelagem.

Para trabalhar com o Modelo SIR com os dados de julho e novembro, foi necessário encontrar o valor de R_0 no fim do mês de julho e seu valor atualmente.

Depois de procurar essas taxas em matérias e jornais, foi decidido que a fonte mais confiável para a retirada desses dados é o "Observatório COVID-19 BR". O observatório apresenta uma divisão de dados por município. Acessando a página da cidade de Curitiba, foi possível encontrar um gráfico com o valor estimado para o número de reprodução efetivo R_e desde o mês de abril, além de um intervalo de confiança, que funciona como uma margem de erro. Um novo valor para a referida taxa é colocado no gráfico a cada 7 dias. Segundo o site, a chance de o valor exato da taxa estar dentro do intervalo de confiança é de 95%.

Segue, abaixo, o gráfico do número reprodutivo basal em Curitiba

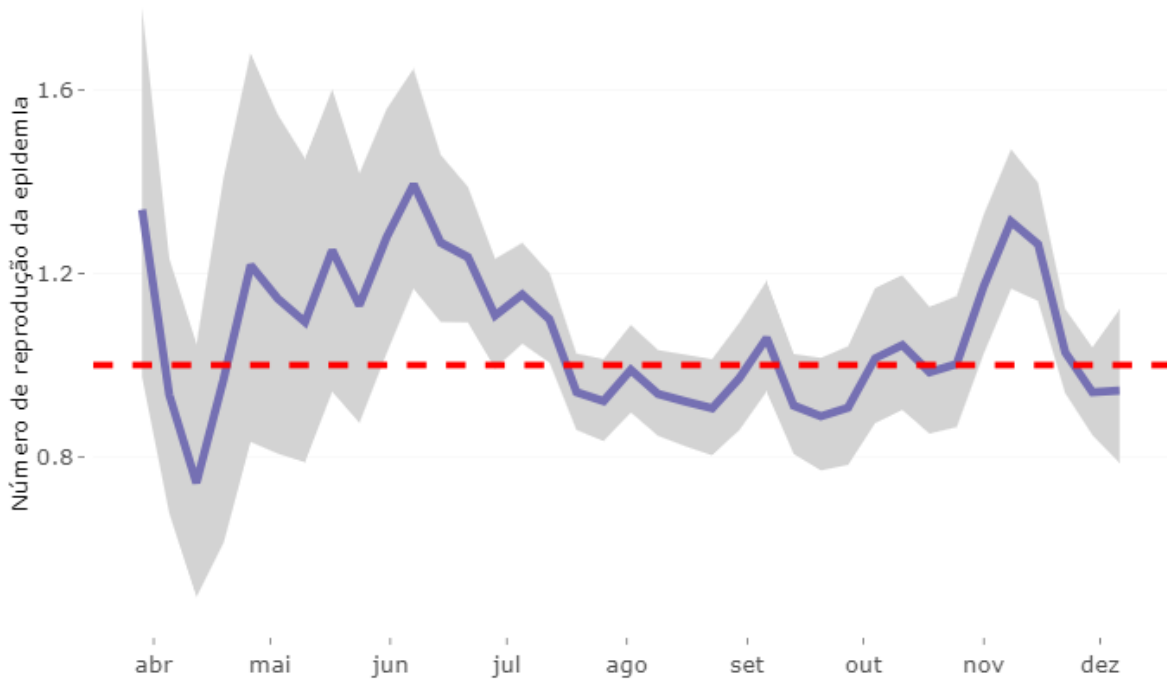


Figura 9.1: Gráfico da taxa de transmissão em Curitiba

A linha azul representa a estimativa para a taxa R_e , a área cinza é o intervalo de confiança e a linha pontilhada vermelha é constante e igual a 1.

Para o mês de julho, foi escolhida a taxa R_e do dia 26/07/2020, estimada em 0.921, com intervalo de confiança de 0.834 até 1.012. Essa foi a última estimativa para o R_e feita pelo observatório em julho.

Para a previsão com dados de novembro, foi escolhida a estimativa para o R_e feita em 06/12/2020, cujo valor é de 1.944, com intervalo de confiança que varia de 0.847 até 1.038.

Conhecendo a estimativa para a taxa de transmissão e os valores inicial e final do intervalo de confiança, é possível fazer uma previsão ampla de um cenário futuro. Com o valor estimado, faz-se uma previsão; com o menor valor do intervalo de confiança faz-se uma previsão otimista e, por último, com o maior valor do intervalo, faz-se uma previsão pessimista de um cenário futuro, apontando a pior situação levando em conta o valor de R_e . Esse era um possível rumo que o trabalho poderia tomar.

Esse número R_e é diferente do valor procurado R_0 . Porém, os dois valores possuem uma relação. Segundo o artigo "NÚMERO REPRODUTIVO DA COVID-19 E O RELAXAMENTO DO DISTANCIAMENTO SOCIAL NO BRASIL", de André Amin Martin Akhavan, existe a relação:

$$R_e = R_0 \cdot s,$$

onde s é a proporção de suscetíveis dentro da população estudada. Porém, o cálculo do R_0 por meio dessa relação não foi feito por conta do pouco tempo restante até a entrega deste trabalho. O foco foi dado a encontrar diretamente um valor para a taxa de transmissão β , para ser usada diretamente no modelo.

A partir disso, foram encontradas as taxas de crescimento $\beta = 1,65$ para o mês de julho e $\beta = 1,2$ para o mês de novembro.

10

Código

A primeira parte do código faz quatro gráficos com os dados coletados a respeito do avanço da Covid-19 nos meses de Julho e Novembro. Os gráficos estão dispostos da seguinte maneira:

1. Casos confirmados até o respectivo mês
2. Total de óbitos até o respectivo mês
3. Total de recuperados até o respectivo mês
4. Novos casos registrados até o respectivo mês

Para plotar os gráficos acima, foi criado um vetor com os dias os quais tínhamos encontrado os dados e um vetor para cada outra categoria de informações selecionadas. Sendo que junto a cada gráfico foi plotado ainda a aproximação de função com Análise de Regressão como já mencionado anteriormente. A imagem com os gráficos de cada mês foi salvo então em formato PNG, como pode-se ver nas figuras abaixo.

```
8 function graficos_julho()
9     #preenchendo os vetores com as informações de julho
10     dias_julho = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 27, 28, 29, 30, 31]
11     novos_casos_julho = [345, 434, 411, 119, 348, 695, 256, 640, 418, 454, 367, 184, 591, 233, 717, 768, 465, 468, 307, 537, 467,
12         662, 640, 720, 364, 295, 392, 411, 496, 699]
13     novos_obitos_julho = [4, 15, 5, 10, 7, 6, 11, 10, 10, 7, 9, 11, 14, 20, 12, 11, 13, 13, 14, 19, 18, 17, 18, 22, 16, 15, 11, 13, 20, 24]
14     confirmados_julho = [5523, 5957, 6368, 6487, 6835, 7530, 7786, 8425, 8844, 9298, 9665, 9849, 10440, 10673, 11390, 12158, 12623,
15         13091, 13398, 13935, 14402, 15064, 15704, 16424, 16788, 17328, 17728, 18131, 18627, 19326]
16     obitos_julho = [152, 167, 172, 182, 189, 195, 206, 216, 226, 233, 242, 253, 267, 287, 299, 310, 323, 336, 350, 369, 387, 404, 422,
17         444, 460, 480, 491, 504, 524, 548]
18
19     dias_julho2 = [2, 7, 11, 12, 13, 16, 18, 19, 20, 21, 22, 23, 24, 25, 27, 28, 29, 30, 31]
20     recuperados_julho = [2909, 4145, 5883, 5735, 5784, 6717, 7134, 7343, 7710, 7915, 8275, 8341, 8407, 8533, 10334, 10944, 11533,
21         12413, 12991]
22
23     f(x) = -0.2321 * x^3 + 12.6913 * x^2 + 275.1272 * x + 5329.4509 #Aproximação de função - Casos confirmados até julho
24     F(x) = -0.0067 * x^3 + 0.4968 * x^2 + 3.4556 * x + 156.5620 #Aproximação de função - Óbitos até julho
25     g(x) = 0.7263 * x^3 - 30.2455 * x^2 + 607.0170 * x + 1616.3719 #Aproximação de função - Recuperados até julho
26     h(x) = .^(5.3849 + 0.0143 * x) #Aproximação de função - Novos casos em julho
27     w = range(0, 31, length=100)
28
29     #plotando os gráficos numa grade 2x2
30     layout = grid(2, 2)
31     p = plot(layout=layout, leg=false)
32
33     #Plotando as funções
34     plot!(p[1], f, w, c=:black, lw=2)
35     plot!(p[2], F, w, c=:black, lw=2)
36     plot!(p[3], g, w, c=:black, lw=2)
37     plot!(p[4], h, w, c=:black, lw=2)
38
39     scatter!(p[1], dias_julho, confirmados_julho, title = "Casos confirmados até julho", xlabel = "dias de julho", ylabel = "confirmados", c=:orange)
40     scatter!(p[2], dias_julho, obitos_julho, title = "Óbitos até julho", xlabel = "dias de julho", ylabel = "óbitos", c=:orange)
41     scatter!(p[3], dias_julho2, recuperados_julho, title = "Recuperados até julho", xlabel = "dias de julho", ylabel = "recuperados", c=:orange)
42     scatter!(p[4], dias_julho, novos_casos_julho, title = "Novos casos em julho", xlabel = "dias de julho", ylabel = "novos casos", c=:orange)
43
44     #salvando a imagem com os gráficos
45     png(p, "graficos_julho")
46 end
47 graficos_julho()
```

Figura 10.1: Função para plotar os gráficos de Julho

```

53 function graficos_novembro()
54     dias_novembro = [3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30]
55     novos_casos_novembro = [896, 331, 407, 436, 473, 536, 477, 736, 703, 749, 715, 1508, 879, 914, 1381, 1380, 984, 1339, 1101,
56                             1597, 1604, 1571, 1521, 970, 1254]
57     novos_obitos_novembro = [10, 6, 8, 8, 9, 7, 5, 5, 10, 5, 6, 10, 13, 11, 9, 8, 7, 10, 11, 11, 18, 16, 17, 18, 16]
58     confirmados_novembro = [53785, 54116, 54523, 54959, 55432, 55968, 56445, 57181, 57884, 58663, 59348, 60856, 61735, 62649, 64030, 66819, 67803,
59                             69142, 70243, 71840, 73444, 75015, 76536, 77586, 78760]
60     ativos_novembro = [3965, 3935, 4012, 4131, 4236, 4395, 4431, 4763, 5004, 5306, 5705, 6849, 7449, 7714, 8415, 9637, 10224, 11323, 11500, 12139,
61                         12784, 12973, 13320, 13582, 13829]
62     obitos_novembro = [1490, 1496, 1504, 1512, 1521, 1528, 1533, 1538, 1548, 1553, 1559, 1569, 1582, 1593, 1602, 1621, 1628, 1638, 1649, 1660, 1678, 1694, 1771, 1729, 1745]
63
64     dias_novembro2 = [3, 4, 5, 6, 7, 10, 11, 12, 13, 14, 15, 17, 18, 19, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30]
65     recuperados_novembro = [48330, 48685, 49007, 49316, 49675, 50481, 50880, 51331, 51774, 52084, 52438, 52704, 53342, 54013, 55561, 55951, 56272,
66                             57094, 58041, 58982, 60348, 61505, 62195, 63186]
67
68     f(x) = -0.2482 * x^3 + 40.2731 * x^2 - 149.7415 * x + 54212.1593 #Aproximação de função - Casos confirmados até novembro
69     F(x) = 0.0123 * x^3 - 0.3454 * x^2 + 9.0592 * x + 1466.1374 #Aproximação de função - Óbitos até novembro
70     g(x) = 0.7414 * x^3 - 19.8918 * x^2 + 477.4610 * x + 47048.1630 #Aproximação de função - Recuperados até novembro
71     h(x) = -0.2966 * x^3 + 14.1247 * x^2 - 148.0600 * x + 934.1163 #Aproximação de função - Novos casos em novembro
72     x = range(0, 30, length=100)
73
74     #plotando os gráficos numa grade 2x2
75     layout = grid(2, 2)
76     p = plot(layout=layout, leg=false)
77
78     #plotando as funções
79     plot!(p[1], f, x, c=:black, lw=2)
80     plot!(p[2], F, x, c=:black, lw=2)
81     plot!(p[3], g, x, c=:black, lw=2)
82     plot!(p[4], h, x, c=:black, lw=2)
83
84     scatter!(p[1], dias_novembro, confirmados_novembro, title = "Casos confirmados até novembro", xlabel = "dias de novembro", ylabel = "confirmados", c=:cyan)
85     scatter!(p[2], dias_novembro, obitos_novembro, title = "Óbitos até novembro", xlabel = "dias de novembro", ylabel = "óbitos", c=:cyan)
86     scatter!(p[3], dias_novembro2, recuperados_novembro, title = "Recuperados até novembro", xlabel = "dias de novembro", ylabel = "recuperados", c=:cyan)
87     scatter!(p[4], dias_novembro, novos_casos_novembro, title = "Novos casos em novembro", xlabel = "dias de novembro", ylabel = "novos casos", c=:cyan)
88
89
90     #salvando a imagem com os gráficos
91     png(p, "graficos_novembro")
92 end
93 graficos_novembro()

```

Figura 10.2: Função para plotar os gráficos de Novembro

Em seguida foi criada uma função para o Método de Euler e uma função para o Método Runge-Kutta, como pode-se ver nas Figuras 10.3 e 10.4, de quarta ordem que serão utilizados para fazer os gráficos do Modelo SIR. Nessas duas funções temos o x_0 e o y_0 que é onde se inicia a nossa função, sendo o x_0 dado pelo tempo em dias e o y_0 dado pelo número de casos. Temos ainda o x_f que é o tempo final e o N o número de passos que a função vai dar.

```

99 function metodo_de_euler(x0, y0 :: Vector, f, xn, N)
100     h = (xn - x0) / N
101     m = length(y0)
102     y = zeros(m, N + 1)
103     y[:,1] = y0
104     x = range(x0, xn, length = N + 1)
105     for k = 1:N
106         y[:,k+1] = y[:,k] + h * f(x[k], y[:,k])
107     end
108     return x, y
109 end

```

Figura 10.3: Função para o Método de Euler

```

115 function metodo_runge_kutta(x0, y0 :: Vector, f, xn, N)
116     h = (xn - x0) / N
117     m = length(y0)
118     y = zeros(m, N + 1)
119     y[:,1] = y0
120     x = range(x0, xn, length=N+1)
121     for k = 1:N
122         k1 = f(x[k], y[:,k])
123         k2 = f(x[k] + 0.5 * h, y[:,k] + h * k1 / 2)
124         k3 = f(x[k] + 0.5 * h, y[:,k] + h * k2 / 2)
125         k4 = f(x[k+1], y[:,k] + h * k3)
126         y[:,k+1] = y[:,k] + h * (k1 + 2*k2 + 2*k3 + k4) / 6
127     end
128     return x, y
129 end

```

Figura 10.4: Função para o Método Runge-Kutta

Criamos então duas funções para o Modelo SIR, uma para o mês de Julho e outra pra o mês de Novembro. A estrutura é a mesma, a diferença está nos dados presentes em casa uma.A

entrada das duas funções é dada por um i que é o tempo de infecção da doença (sendo que utilizamos a média de 14 dias) e o método a ser utilizado, Método de Euler ou Método de Runge-Kutta. O T é dado pelo número de habitantes na cidade de Curitiba, o γ é a taxa de recuperação e o β a taxa de crescimento nos respectivos meses. Ainda, temos o y_0 que muda de um mês para o outro por ser dado pelo número de casos ativos no início de cada mês, sendo que em Julho é 2.881 e Novembro 3.965. E por fim é plotado cada gráfico em formato PNG, como pode-se ver nas Figuras abaixo.

```

134 function sir_julho(i, metodo)
135     # i é o tempo de infecção. Ex.: i=14
136     if metodo != :euler && metodo != :runge
137         error("insira ':runge' ou ':euler'")
138     end
139     T = 1_752_000 #Total de habitantes na cidade de Curitiba
140     γ = 1/i #Taxa de recuperação
141     β = 1.65 #Taxa de crescimento
142
143     # Temos o vetor y = [S, I, R] onde S é o número de suscetíveis, I é o número de Infectados e R o número de Recuperados.
144     # Com isso, o y[1] abaixo se refere aos Suscetíveis e o y[2] se refere aos Infectados
145     # A função é composta pelas equações que temos através do Modelo SIR
146     f(x, y) = [-β / T * y[2] * y[1];
147               β / T * y[2] * y[1] - γ * y[2];
148               γ * y[2]]
149
150     x0 = 0.0 #Onde se inicia, dado pelo tempo
151     y0 = [T - 2881.0; 2881.0; 0.0] #Onde se inicia, número dos casos
152     xf = 30.0 #Onde termina, ou seja, o tempo final
153     N = 1000 #Quantos passos a função dará
154
155     if metodo == :euler
156         x, y = metodo_de_euler(x0, y0, f, xf, N)
157     else
158         x, y = metodo_runge_kutta(x0, y0, f, xf, N)
159     end
160
161     #Plotando os gráficos de Suscetíveis, Infectados e Recuperados
162     p = plot(x, y[1,:], c=:blue, lab="S", legend=:right, title= "Modelo SIR - Julho")
163     plot!(x, y[2,:], c=:green, lab="I")
164     plot!(x, y[3,:], c=:red, lab="R")
165
166     # Salvando os gráficos
167     png(p, "gráficos_modelo_sir_julho")
168 end

```

Figura 10.5: Função para o Modelo SIR - Julho

```

173 function sir_novembro(i, metodo)
174     # i é o tempo de infecção. Ex.: i=14
175     if metodo != :euler && metodo != :runge
176         error("insira ':runge' ou ':euler'")
177     end
178     T = 1_752_000 #Total de habitantes na cidade de Curitiba
179     γ = 1/i #Taxa de recuperação
180     β = 1.20 #Taxa de crescimento
181
182     # Temos o vetor y = [S, I, R] onde S é o número de suscetíveis, I é o número de Infectados e R o número de Recuperados.
183     # Com isso, o y[1] abaixo se refere aos Suscetíveis e o y[2] se refere aos Infectados
184     # A função é composta pelas equações que temos através do Modelo SIR
185     f(x, y) = [-β / T * y[2] * y[1];
186               β / T * y[2] * y[1] - γ * y[2];
187               γ * y[2]]
188
189     x0 = 0.0 #Onde se inicia, dado pelo tempo
190     y0 = [T - 3965.0; 3965.0; 0.0] #Onde se inicia, número dos casos
191     xf = 30.0 #Onde termina, ou seja, o tempo final
192     N = 1000 #Quantos passos a função dará
193
194     if metodo == :euler
195         x, y = metodo_de_euler(x0, y0, f, xf, N)
196     else
197         x, y = metodo_runge_kutta(x0, y0, f, xf, N)
198     end
199
200     #Plotando os gráficos de Suscetíveis, Infectados e Recuperados
201     p = plot(x, y[1,:], c=:blue, lab="S", legend=:right, title= "Modelo SIR Novembro")
202     plot!(x, y[2,:], c=:green, lab="I")
203     plot!(x, y[3,:], c=:red, lab="R")
204
205     # Salvando os gráficos
206     png(p, "gráficos_modelo_sir_novembro")
207 end

```

Figura 10.6: Função para o Modelo SIR - Novembro

Fizemos ainda o que chamamos de "Modelo SIR - Ideal" onde ainda utilizamos os dados dos casos ativos no início de cada mês mas uma taxa de crescimento que é considerada pequena,

para utilizarmos como justificativas nas conclusões obtidas ao longo da realização do projeto.

```

212 function sir_julho_ideal(i, metodo)
213     # i é o tempo de infecção. Ex.: i=14
214     if metodo != :euler && metodo != :runge
215         error("insira ':runge' ou ':euler'")
216     end
217     T = 1_752_000 #Total de habitantes na cidade de Curitiba
218     γ = 1/i #Taxa de recuperação
219     β = 0.5 #Taxa de crescimento
220
221     # Temos o vetor y = [S, I, R] onde S é o número de suscetíveis, I é o número de Infectados e R o número de Recuperados.
222     # Com isso, o y[1] abaixo se refere aos Suscetíveis e o y[2] se refere aos Infectados
223     # A função é composta pelas equações que temos através do Modelo SIR
224     f(x, y) = [-β / T * y[2] * y[1];
225               β / T * y[2] * y[1] - γ * y[2];
226               γ * y[2]]
227
228     x0 = 0.0 #Onde se inicia, dado pelo tempo
229     y0 = [T - 2881.0; 2881.0; 0.0] #Onde se inicia, número dos casos
230     xf = 30.0 #Onde termina, ou seja, o tempo final
231     N = 1000 #Quantos passos a função dará
232
233     if metodo == :euler
234         x, y = metodo_de_euler(x0, y0, f, xf, N)
235     else
236         x, y = metodo_runge_kutta(x0, y0, f, xf, N)
237     end
238
239     #Plotando os gráficos de Suscetíveis, Infectados e Recuperados
240     p = plot(x, y[1,:], c=:blue, lab="S", legend=:right, title= "Modelo SIR Julho - Ideal")
241     plot!(x, y[2,:], c=:green, lab="I")
242     plot!(x, y[3,:], c=:red, lab="R")
243
244     # Salvando os gráficos
245     png(p, "gráficos_sir_julho_ideal")
246 end

```

Figura 10.7: Função para o Modelo SIR Ideal - Julho

```

251 function sir_novembro_ideal(i, metodo)
252     # i é o tempo de infecção. Ex.: i=14
253     if metodo != :euler && metodo != :runge
254         error("insira ':runge' ou ':euler'")
255     end
256     T = 1_752_000 #Total de habitantes na cidade de Curitiba
257     γ = 1/i #Taxa de recuperação
258     β = 0.5 #Taxa de crescimento
259
260     # Temos o vetor y = [S, I, R] onde S é o número de suscetíveis, I é o número de Infectados e R o número de Recuperados.
261     # Com isso, o y[1] abaixo se refere aos Suscetíveis e o y[2] se refere aos Infectados
262     # A função é composta pelas equações que temos através do Modelo SIR
263     f(x, y) = [-β / T * y[2] * y[1];
264               β / T * y[2] * y[1] - γ * y[2];
265               γ * y[2]]
266
267     x0 = 0.0 #Onde se inicia, dado pelo tempo
268     y0 = [T - 3965.0; 3965.0; 0.0] #Onde se inicia, número dos casos
269     xf = 30.0 #Onde termina, ou seja, o tempo final
270     N = 1000 #Quantos passos a função dará
271
272     if metodo == :euler
273         x, y = metodo_de_euler(x0, y0, f, xf, N)
274     else
275         x, y = metodo_runge_kutta(x0, y0, f, xf, N)
276     end
277
278     #Plotando os gráficos de Suscetíveis, Infectados e Recuperados
279     p = plot(x, y[1,:], c=:blue, lab="S", legend=:right, title= "Modelo SIR Novembro - Ideal")
280     plot!(x, y[2,:], c=:green, lab="I")
281     plot!(x, y[3,:], c=:red, lab="R")
282
283     # Salvando os gráficos
284     png(p, "gráficos_sir_novembro_ideal")
285 end

```

Figura 10.8: Função para o Modelo SIR Ideal - Novembro

Conclusões

Tínhamos como objetivo fazer a análise da interferência da bandeira em vigência em relação ao número de casos ativos diários sendo que nos meses escolhidos de Julho e Novembro, grande parte do tempo a bandeira vigente era a amarela sendo a mais leve em relação às restrições da circulação de pessoas e funcionamento do comércio, sendo que no mês de Novembro apenas nos últimos 4 dias havia a vigência da bandeira Laranja. Com isso, podemos perceber há apenas uma pequena diferença na previsão do Modelo SIR para os dois meses, mesmo que no mês de Julho mais dias estavam sob a bandeira Laranja.

Com a análise dos gráficos do Modelo SIR para os meses de Julho e Novembro e os gráficos do Modelo SIR que chamamos de Ideal, visto que a taxa de crescimento usada foi menor do que as reais, podemos perceber que o cenário teria sido totalmente diferente, caso o crescimento da pandemia tivesse sido melhor controlada. Os gráficos feitos mostram claramente a diferença na quantidade de Infectados, e conseqüentemente teria um número menor de óbitos, caso tivesse sido implementada a bandeira Laranja ou Vermelha na cidade de Curitiba e a fiscalização da circulação de pessoas.

Vale lembrar que, apesar do enorme número de infectados, foi considerado como "Ideal" um cenário cuja taxa de crescimento β fosse menor do que as taxas encontradas, para ilustrar uma situação com a curva de infectados mais achatada.

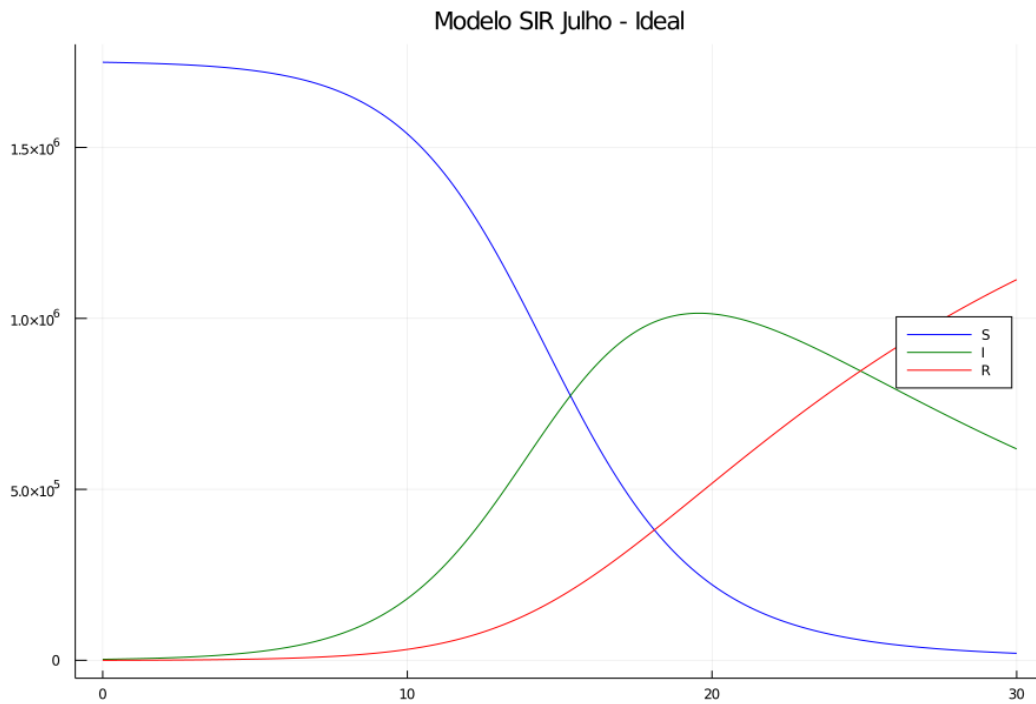


Figura 11.1: Modelo SIR com $\beta = 0,5$ para Julho

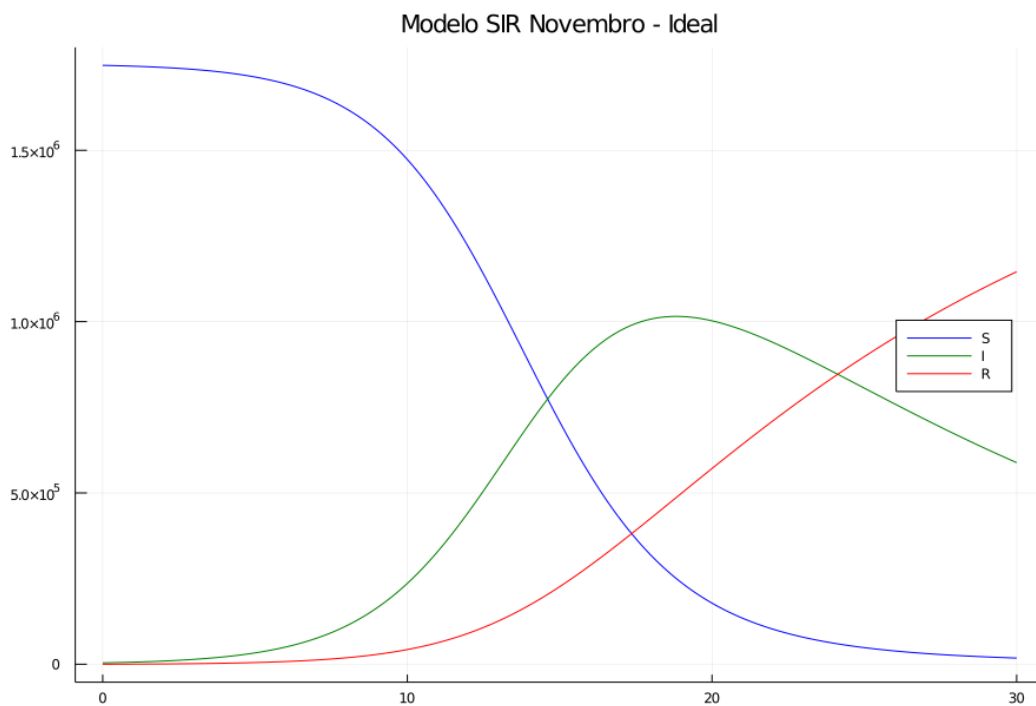


Figura 11.2: Modelo SIR com $\beta = 0,5$ para Novembro

Concluimos então, que os gráficos do Modelo SIR de cada mês e os gráficos do Modelo SIR Ideal para esses meses, mostram claramente que existia e ainda existe atualmente a necessidade de restrições rígidas para o controle da taxa de crescimento, visto que ainda não estamos em processo de vacinação.

Com as taxas de crescimento β encontradas, foram plotados os gráficos dos meses de julho e novembro, conforme segue:

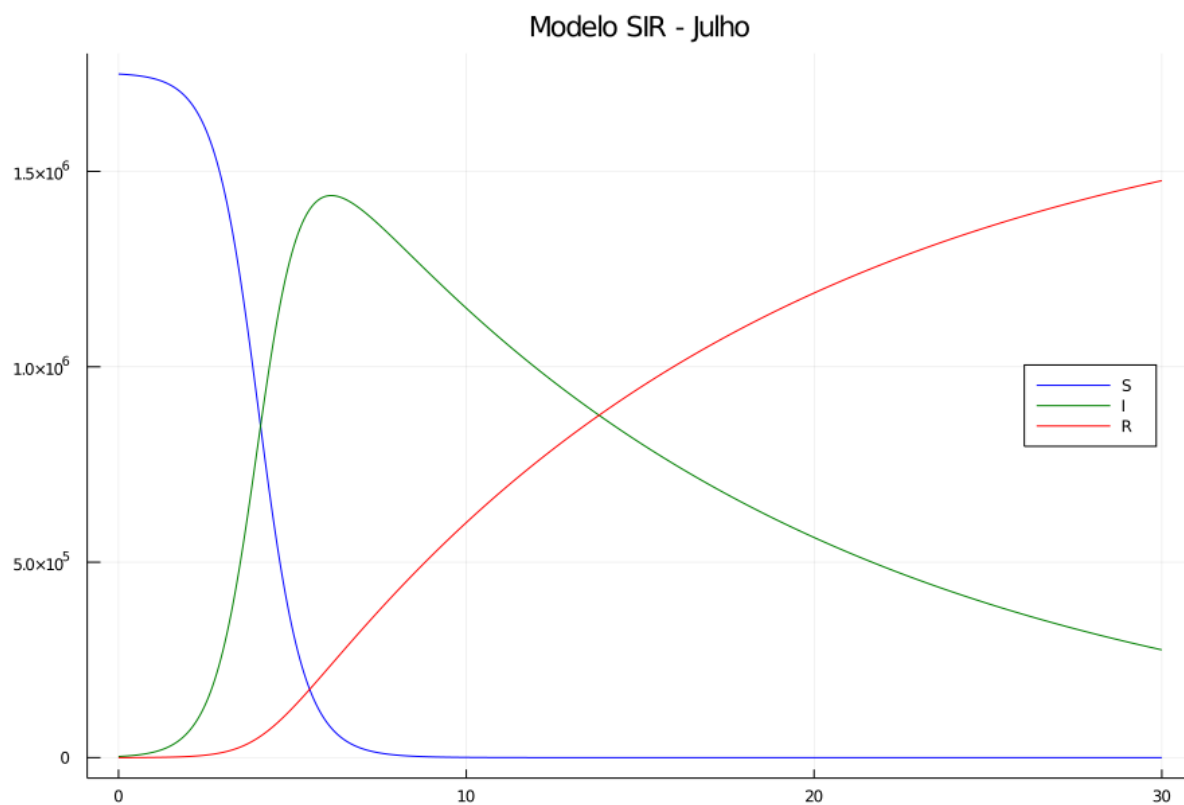


Figura 11.3: Modelo SIR com $\beta = 1,65$ para julho.

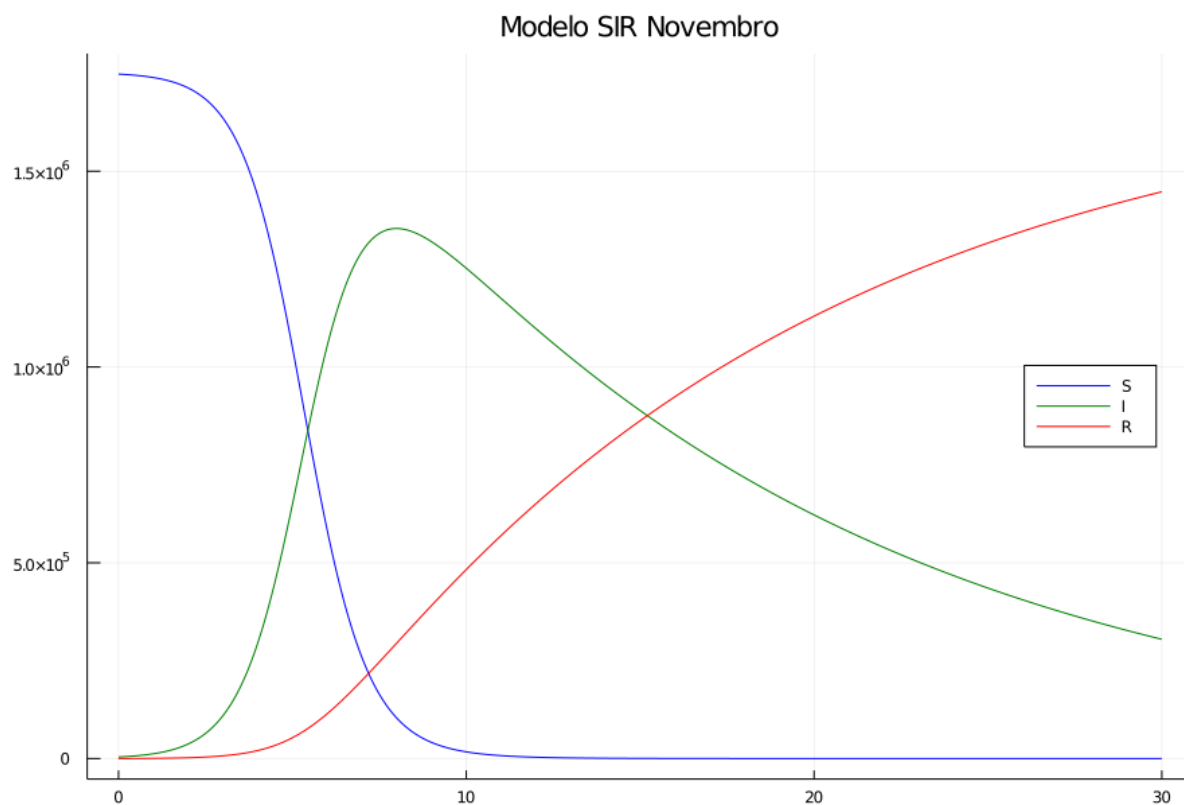


Figura 11.4: Modelo SIR com $\beta = 1,20$ para novembro.

Essas taxas de 1,65 e 1,20 foram encontradas por meio de uma pesquisa. Sabe-se que o valor de β pode ser calculado usando como base os dados reais do número de infectados e o número de dias. Contudo, em vista do tempo escasso para finalizar o projeto, isso não foi feito.

Percebe-se, claramente, que os resultados obtidos fogem da realidade, visto que nas aproximações feitas ocorreram mais do que 1 milhão de infecções pela doença em 20 dias ou menos, até mesmo nos modelos que consideramos ideais.

Essa divergência dos resultados obtidos com a realidade ocorreu, muito provavelmente, por conta das poucas informações encontradas acerca das taxas envolvidas no modelo. Principalmente sobre a taxa de crescimento β . Além disso, a falta de tempo também atrapalhou a obtenção de melhores resultados, pois impossibilitou o aprofundamento da pesquisa.

Referências

PROTOCOLO DE RESPONSABILIDADE SANITÁRIA E SOCIAL. Curitiba contra o Coronavírus, Prefeitura de Curitiba, Curitiba, 29/11/2020. Disponível em: <https://coronavirus.curitiba.pr.gov.br/numerosCovid> >. Acesso em: 29 de novembro de 2020 e datas anteriores.

MARTINS, Camila Marinelli et al. Modelo Preditivo da ocorrência de Covid-19 em município de médio porte no Brasil (Ponta Grossa - Paraná). Texto Contexto-Enfermagem, v. 29, 2020. Disponível em: https://www.scielo.br/scielo.php?pid=S0104-07072020000100204script=sci_arttexttln=pt >. Acesso em: 25 de novembro de 2020.

A Evolução epidêmica do COVID-19 - Modelo SIR. Disponível em: <https://wp.ufpel.edu.br/fentransporte/2020/04/09/a-evolucao-epidemica-do-covid-19-modelo-sir/> >. Acesso em: 25 de novembro de 2020.

Modelos matemáticos serão estratégicos no combate ao coronavírus. Publicado em 11 de Junho de 2020. Disponível em: <https://www.sbmt.org.br/portal/mathematical-models-will-be-strategic-in-combating-coronavirus/> >. Acesso em: 26 de novembro de 2020.

GOMES, Sebastião C. P. et al. MODELAGEM DINÂMICA DA COVID-19 COM APLICAÇÃO A ALGUMAS CIDADES BRASILEIRAS. Universidade Federal do Rio Grande, Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul. Rio Grande do Sul. Disponível em <https://imef.furg.br/images/stories/documentos/2020/artigo2.pdf> >. Acesso em: 26 de novembro de 2020.

R efetivo em Curitiba. Disponível em <https://covid19br.github.io/municipios.html?aba=aba3uf=PRmun=Curitibaq=dia> >. Acesso em: 10 de dezembro de 2020.

MONICH, J. W. Modelagem da Dinâmica Epidemiológica da Dengue. 2018. 50 f. Trabalho de Conclusão de Curso, Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2018. Disponível em: <https://maratona.ime.usp.br/map/tcc/2018/JeniferMonichV1.pdf> >.

<https://epoca.globo.com/sociedade/coronavirus-quanto-tempo-uma-pessoa-leva-para-se-recuperar-da-covid-19-1-24394950> >. Acesso em: 12/12/2020.

David B. Saakian. A simple statistical physics model for the epidemic with incubation period. A.I. Alikhanyan National Science Laboratory (Yerevan Physics Institute) Foundation, Armenia, 2020. Disponível em: <https://arxiv.org/pdf/2004.05778.pdf?fbclid=IwAR2s0CS1KB-xzsMuRdsI53iXKPnm3jX-Zlmf5RZLRTGRhExXlXWkyroJoQ> >.

< <https://www2.unifap.br/matematicaead/files/2016/03/Enviar-Ellison-TCC.pdf> >

< <https://www.ime.unicamp.br/valle/Teaching/2015/MS211/Aula14.pdf> >

< https://planetcalc.com/5992/?language_select=ptthanks=1 >

SIQUEIRA, Abel. Modelo de Equações Diferenciais SIR para o COVID-19. Partes 1 e 2. Disponíveis em: *< <https://www.twitch.tv/videos/577731792> >*, *< <https://www.twitch.tv/videos/577731791> >*. Acesso em: 04 de dezembro de 2020.