

Übung 03: Pandemieausbruch

Tobias Blesgen und Leonardo Thome

09.06.2021

Im Folgenden wollen wir ein Simulierte Pandemieausbreitung von SARS-CoV-2 für verschiedene Parameter im SIR Modell betrachten. Das SIR Modell beschreibt den zeitlichen Verlauf mit folgendem Differentialgleichungssystem.

$$S'(t) = -\frac{\beta}{N}S(t)I(t) - \Gamma(t) + \delta V(t) \quad (1)$$

$$I'(t) = \frac{\beta}{N}S(t)I(t) - \alpha I(t) \quad (2)$$

$$R'(t) = \alpha I(t) \quad (3)$$

$$V'(t) = \Gamma(t) - \delta V(t) \quad (4)$$

Hierbei entspricht S der Zahl der noch nicht Infektierten, I der aktuell kranken, R der Genesenen und V der Geimpften. Wir werden im Verlauf der Simulationen die Geimpfenzahl mit der Anzahl der Personen mit Zweitimpfung gleichsetzen. Desweiteren ist N die Gesamtbevölkerung, R_0 entspricht der Basisreproduktionszahl, α der inversen ANSTECKBARZEIT (?), $\beta = R_0 * \alpha$, Γ die Impfungsrate und δ die Rate mit der Geimpfte erkranken.

Lunge Jutta Verfahren

Um die differenziellen Gleichungssysteme auszuwerten, verwenden wir das Runge-Kutta Verfahren nach

$$x_{i+1} = x_i + \frac{h}{2}[f(t_i, x_i) + f(t_i + h, x_i + hf(t_i, x_i))] \quad (5)$$

Wobei sich in unserem f aus den vier Anteilen von S, I, R und V zusammensetzt. Wir werden die Auswirkungen der Schrittweite am Ende der Auswertung genauer betrachten.

Implementation

```
#include <Rcpp.h>
#include <stdlib.h>
#include <stdio.h>
#include <vector>
#include <algorithm>
```

```

using namespace Rcpp;

// Wir verwenden Strukturen um Funktionsargumente uebersichtlicher zu halten
typedef struct
{
    double S, I, R, V;
} Status;

typedef struct
{
    double alpha, beta, delta, N, gamma;
} Parameter;

// Template zum zerschneiden der verwendeten Vektoren
template<typename T>
std::vector<T> slice(std::vector<T> const &v, int m, int n)
{
    auto first = v.cbegin() + m;
    auto last = v.cbegin() + n + 1;

    std::vector<T> vec(first, last);
    return vec;
}

// Berechnung der Ableitungen nach den DGL
void f(Status alterStatus, Parameter parameter, Status& neuerStatus){
    double Gamma = std::min(parameter.gamma, alterStatus.I);
    neuerStatus.V = Gamma - parameter.delta * alterStatus.V;
    neuerStatus.R = parameter.alpha * alterStatus.I;
    neuerStatus.I = parameter.beta * alterStatus.S * alterStatus.I/parameter.N
        - parameter.alpha * alterStatus.I;
    neuerStatus.S = -parameter.beta * alterStatus.S * alterStatus.I/parameter.N
        - Gamma + parameter.delta * alterStatus.V;
}

// Ein Intergrationsschritt nach Runge-Kutta
void rkSchritt(Status& status, Parameter parameter, double h){
    Status fStatus;           // Standart Ableitung
    f(status, parameter, fStatus);
    Status f2Status;          // Mischterm Ableitung
    Status gemischt = {.S = status.S + h*fStatus.S, .I = status.I
        + h*fStatus.I, .R = status.R + h*fStatus.R, .V = status.V + h*fStatus.V};
    f(gemischt, parameter, f2Status);
    status.V = status.V + h/2*(fStatus.V + f2Status.V);
    status.R = status.R + h/2*(fStatus.R + f2Status.R);
    status.S = status.S + h/2*(fStatus.S + f2Status.S);
    status.I = status.I + h/2*(fStatus.I + f2Status.I);
}

//[[Rcpp::export]]
Rcpp::List durchlauf(const int maxSchritte, const double h,
                    const double S, const double I, const double R,
                    const double V, const double alpha,

```

```

        const double beta, const double delta,
        const double N, const double gamma,
        const double x0, const int I_min = 1){
// Arrays der Werte zur späteren Ausgabe
    std::vector<double> xValue(maxSchritte);
    std::vector<double> SValue(maxSchritte);
    std::vector<double> IValue(maxSchritte);
    std::vector<double> RValue(maxSchritte);
    std::vector<double> VValue(maxSchritte);
// Quelltext
    Status status = {.S = S, .I = I, .R = R, .V = V};
    Parameter parameter = {.alpha = alpha, .beta = beta, .delta = delta, .N = N, .gamma = gamma};
// Schleife bis zur Abbruchbedingung
    for (int i = 0; i < maxSchritte; i++){
        xValue[i] = x0 + i*h;
        SValue[i] = status.S;
        IValue[i] = status.I;
        RValue[i] = status.R;
        VValue[i] = status.V;
        rkSchritt(status, parameter, h);
// Vorzeitige Abbruchbedingung
        if(status.I < I_min && i>10){
            // Rückgabe für eine grafische Wiedergabe
            return List::create(Named("x") = slice(xValue,0,i),
                                Named("S") = slice(SValue,0,i),
                                Named("I") = slice(IValue,0,i),
                                Named("R") = slice(RValue,0,i),
                                Named("V") = slice(VValue,0,i)
                                );
        }
    }
// Rückgabe für eine grafische Wiedergabe
    return List::create(Named("x") = xValue, Named("S") = SValue,
                        Named("I") = IValue, Named("R") = RValue,
                        Named("V") = VValue
                        );
}

```

Auswertung

Ohne Gegenmaßnahmen

Abb. 1 bestätigt bereits, dass unsere Funktion die erwarteten Infektionskurven ausgibt. Betrachten wir nun die Kurven für $R_0 \in [1, 20]$, so können wir in Abb. 2 sehen, wie sich die Krankheit für große Basisreproduktionszahlen auf wenige Tage fokussiert und die Infektionszahlen in die Höhe steigen. Im Vergleich dazu sind die Kurven für kleine Basisreproduktionszahlen deutlich flacher, strecken sich jedoch auch über ein Vielfaches der Zeit. Für einen echten Krankheitsverlauf würde Ersteres die Kapazitäten von Krankenhäusern und anderen Institutionen extrem belasten, während zweiteres lang andauernde Beschränkungen für soziale Interaktionen mit sich bringen könnte.

Jo Abb. 3 nimmt uns ziemlich Hops!

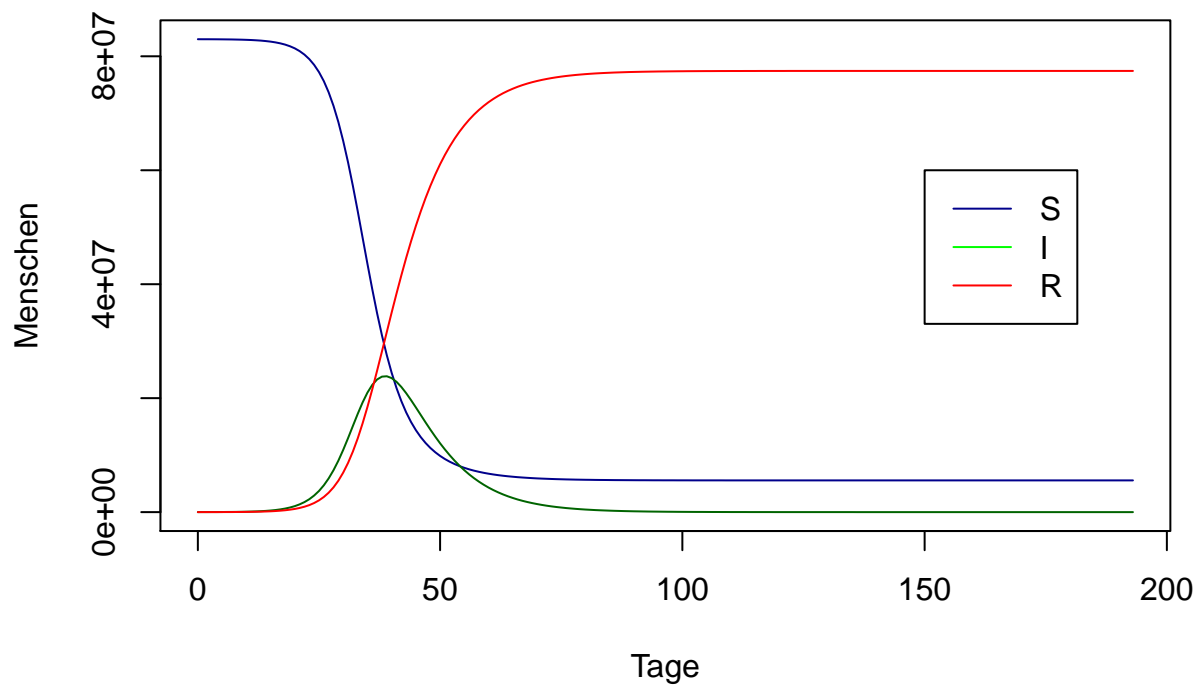


Abbildung 1: Entwicklungskurven

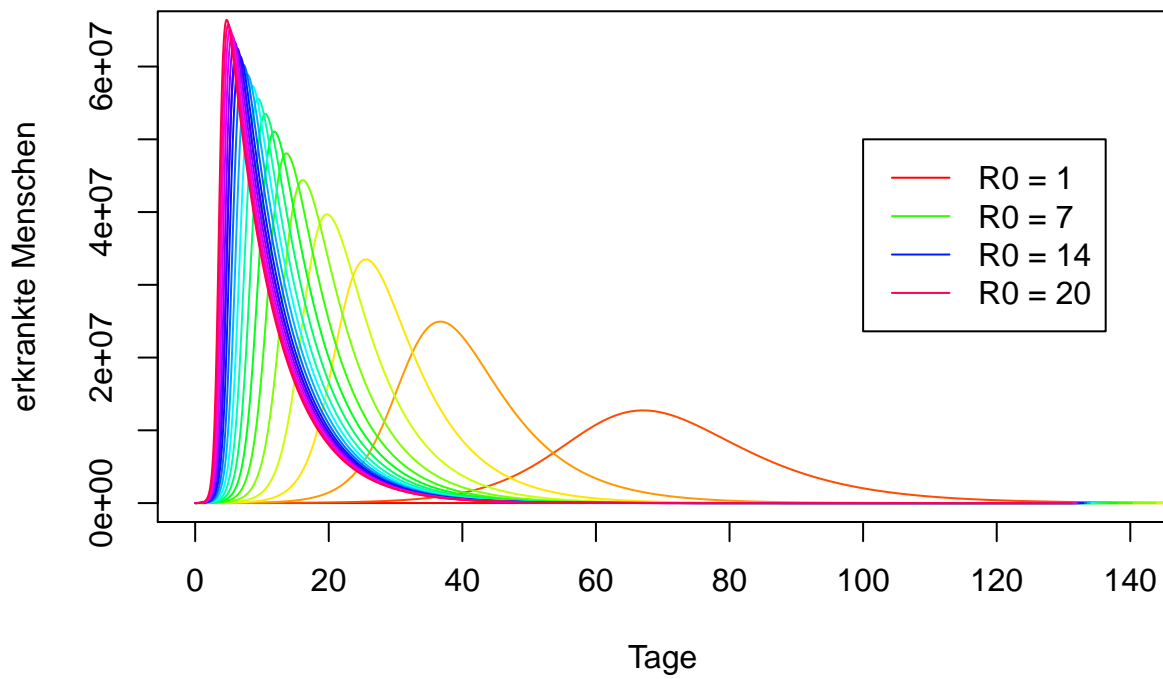


Abbildung 2: R_0 abhängige Infektionsverläufe

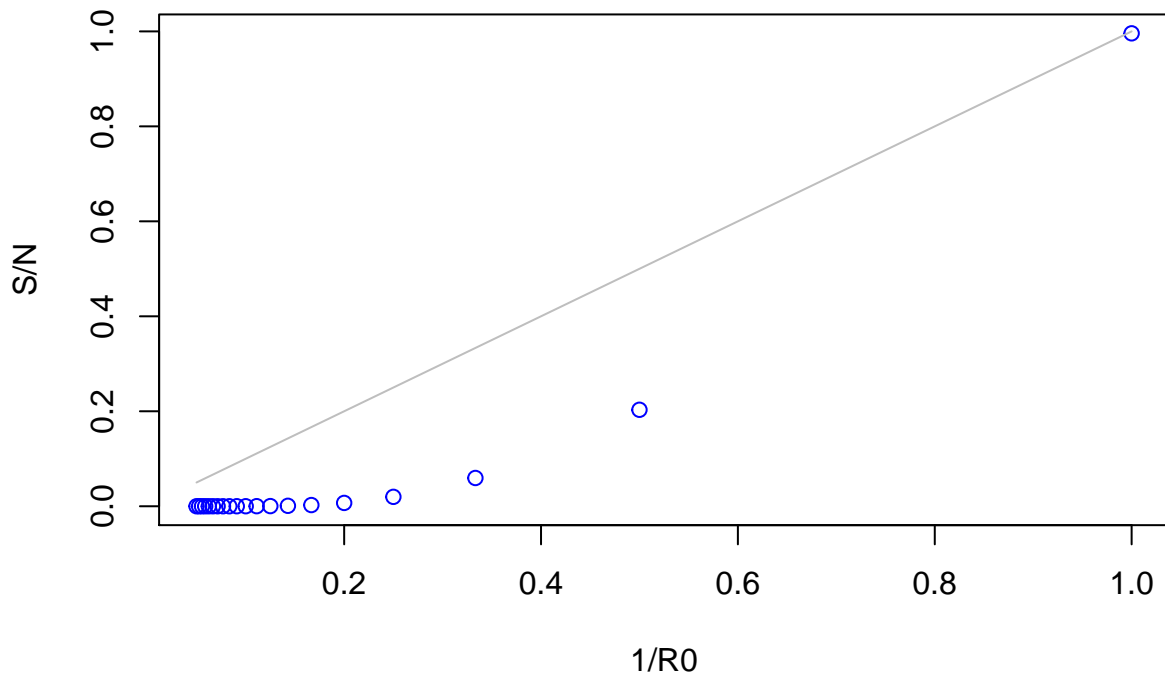


Abbildung 3: R_0 abhängige Herdenimmunität

Mit Gegenmaßnahmen

Unter der Verwendung der Daten des RKI versuchen wir den COVID-19-Verlauf zu modellieren. Als Startwert wurden die Daten des 03.01.2021 [1] mit $I = 349500$, $R = 1765666-I$, $V = 238809$ verwendet. Desweiteren wurde γ und N_0 als Treppenfunktion mit den Werten

WIE MACHT MAN TABELLEN? γ Tag aktive Fälle

Januar: 40665 3 349500 Februar: 61132 37 193200 März: 60776 71 127100 April: 57895 110 279400 Mai: 406455 142 167100 Juni: 337478 152 95900

verwendet. $\frac{1}{\delta}$ wird auf 150 Tage gesetzt.

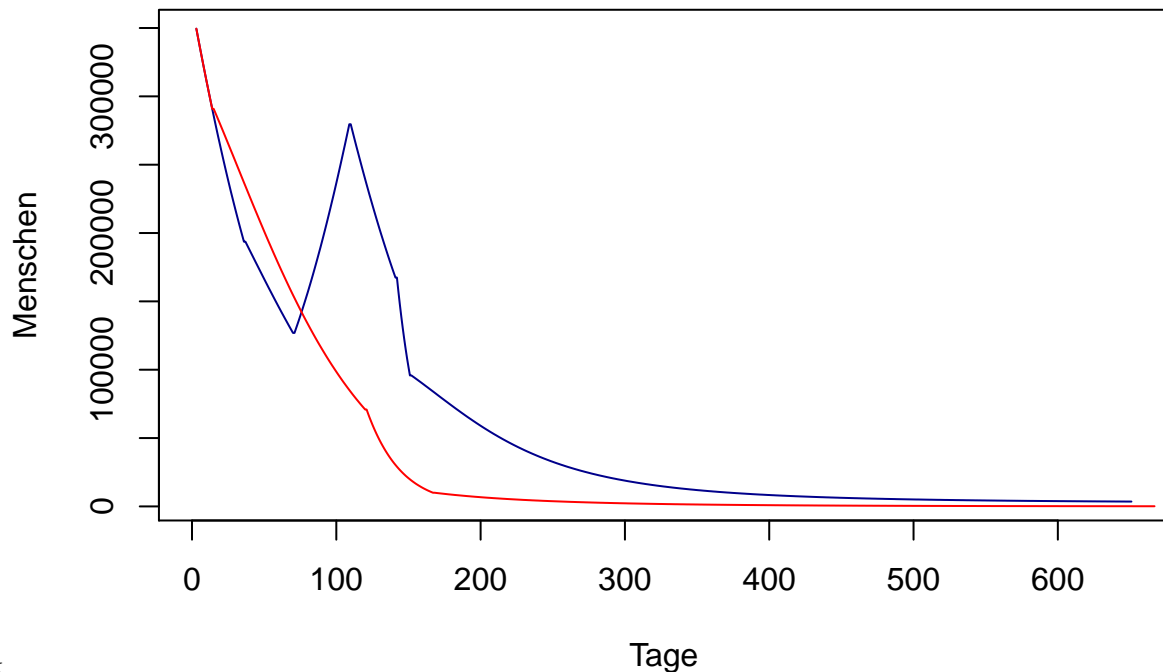
[1] 193693.4

[1] 126912.2

[1] 279589.8

[1] 167387.6

[1] 95831.58



ok

Literatur

- [1] *Täglicher Lagebericht des RKI zur Coronavirus-Krankheit-2019(COVID-19)*, Robert Koch Institut, https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Situationsberichte/Jan_2021/2021-01-03-de.pdf?__blob=publicationFile, Stand: 24.05.2021.
- [2] *Täglicher Lagebericht des RKI zur Coronavirus-Krankheit-2019(COVID-19)*, Robert Koch Institut, https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Situationsberichte/Feb_2021/2021-02-06-de.pdf?__blob=publicationFile, Stand: 24.05.2021.
- [3] *Täglicher Lagebericht des RKI zur Coronavirus-Krankheit-2019(COVID-19)*, Robert Koch Institut, https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Situationsberichte/Maerz_2021/2021-03-12-de.pdf?__blob=publicationFile, Stand: 24.05.2021.
- [4] *Täglicher Lagebericht des RKI zur Coronavirus-Krankheit-2019(COVID-19)*, Robert Koch Institut, https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Situationsberichte/Apr_2021/2021-04-20-de.pdf?__blob=publicationFile, Stand: 24.05.2021.
- [5] *Täglicher Lagebericht des RKI zur Coronavirus-Krankheit-2019(COVID-19)*, Robert Koch Institut, https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Situationsberichte/Mai_2021/2021-05-22-de.pdf?__blob=publicationFile, Stand: 24.05.2021.