Analyse der Coronastatistiken

Hans-Gert Gräbe, Leipzig

Version vom 9. April 2020

Dieser Text bezieht sich auf die im Verzeichnis `http://leipzig-data.de/demo/Corona-20/Code` verfügbaren Materialien.

1 Datenbasis

Als Datenbasis werden die von der John Hopkins Universität (JHU) als Excel-Datei veröffentlichten Daten[1] zur Entwicklung der weltweit registrierten Covid-19-Fälle verwendet. Die Statistik listet pro Land und Tag die kumulierte Zahl der Infizierten, der Genesenen und der Todesfälle auf.

Die Statistiken sind natürlich zu hinterfragen, da sich jedem Physiker die Haare sträuben würden, wenn die Debatte auf die messmethodische Grundlage dieser Datenerhebungen, also die Einheitlichkeit des Modells, zu sprechen kommt. Dennoch gibt es wohl derzeit nichts Besseres. Auch die Rolle der privaten (!) JHU und deren enge Verflechtung mit dem „Datenadel“ aus dem Silicon Valley kann hinterfragt werden, siehe dazu (Rügemer 2020). Spannend auf der anderen Seite, dass es genau dieser „Datenadel“ ist, der solche Daten zusammenträgt und der Allgemeinheit – im Gegensatz etwa zu der im deutschen Sprachraum viel zitierten `https://statista.com` – in einem maschinenlesbaren Format ohne Bezahlschranken zur Verfügung stellt.

Eine solche kritische Würdigung muss diesem Text vorangestellt werden, um die folgenden Rechnungen ins rechte Licht zu rücken, denn sie sind nicht als Beitrag zur Corona-Debatte selbst gedacht, wie dies in den ersten Tagen mehrfach falsch verstanden wurde, sondern soll die Möglichkeiten aufzeigen, die fortgeschrittene Oberstufenschülerinnen und -schüler heute zu einer eigenen Analyse von Daten haben. Der komplexe Gegenstand „CAS in der Schule“ ist in meinem Buch (Gräbe 2018) genauer dargestellt.

2 Installation

Zunächst muss das git Repo der JHU lokal geklont und der Pfad im Skript `extractData.pl` eingetragen werden. Die Daten werden für ausgewählte Länder mit diesem Perl-Skript für die weitere Verarbeitung aufbereitet und in einer Datei `BasicData.txt` gespeichert, um dann mit dem freien CAS *Maxima* weiterverarbeitet zu werden.

[1] Siehe deren github-Projekt `https://github.com/CSSEGISandData/COVID-19`.

3 Datentransformation

In der Datei `BasicData.txt` sind die Daten für jedes der ausgewählten Länder in einem Array mit drei Einträgen (infected, recovered, dead) gespeichert, die im Maxima-Skript `skript.m`[2] in einer Funktion `getland(Land)` zunächst einmal zu Paaren (t, y_t) ergänzt werden, wobei t für den Tag des Jahres 2020 ($1 = 01.01.2020$, **Tag 100 ist also Donnerstag der 9. April**) und y_t für die Zahl der Fälle aus dem jeweiligen Record stehen.

4 Fitting

Alle Grafiken zu Prognosen der Daten, die ich bisher gesehen habe, gehen von einer „Glockenkurve“ aus. Das kann natürlich nur die Entwicklung der Zuwächse pro Tag abbilden, die aus den kumulierten Daten zunächst als $d_t = y_t - y_{t-1}$ extrahiert werden müssen. Dies geschieht mit der im Skript definierten Maxima-Funktion `Delta`[3].

Zur Abschätzung des Verlaufs längs einer Glockenkurve wird üblicherweise die Statistikfunktion $C \cdot \exp\left(-\left(\frac{t-m}{s}\right)^2\right)$ verwendet, die ich als Kurvenschar $f(t) = \exp\left(c - \left(\frac{t-m}{s}\right)^2\right)$ zur Parameterschätzung auf die Daten (t, d_t) ansetze. Die (kumulierten) Originaldaten (t, y_t) sollten dann auf die Funktion

$$\begin{aligned} h(x) &= \int_0^x \exp\left(c - \left(\frac{t-m}{s}\right)^2\right) dt \\ &= \exp(c) \cdot s \cdot \int_{-\frac{m}{s}}^{\frac{x-m}{s}} \exp\left(-u^2\right) du \\ &= \frac{1}{2}\sqrt{\pi} \cdot \exp(c) \cdot s \cdot \left(\operatorname{erf}\left(\frac{x-m}{s}\right) + \operatorname{erf}\left(\frac{m}{s}\right)\right) \end{aligned}$$

matchen, wobei $\operatorname{erf}(x)$ für die Fehlerfunktion steht und im zweiten Schritt die Variablensubstitution $u = \frac{x-m}{s}$ mit $du = s \cdot dt$ erfolgte.

Nun sind die Parameter (c, s, m) dieser Kurvenschar so zu fitten, dass die ermittelte Kurve besonders gut auf die Daten passt. In *Maxima* kann dazu das Paket *lsquares* verwendet werden.

Wir hätten natürlich auch versuchen können, die Originaldaten auf die Schar $h(t)$ zu fitten, aber Fitting auf nicht polynomialen Kurvenscharen ist eine schwierige und numerisch wenig stabile Angelegenheit, bei der Maxima schnell an seine Grenzen kommt (und die Ergebnisse anderer CAS sehr genau zu analysieren sind, da die Fitting-Ergebnisse stark von Startwerten der dabei eingesetzten Verfahren abhängen).

Maxima kommt auch beim Fitting der Schar $f(t)$ zu keinem Ergebnis. Einen einfacheren, nämlich quadratischen Zusammenhang $g(t) = c - \left(\frac{t-m}{s}\right)^2$ erhält man, wenn man zu Paaren

[2] Dies ist eine reine Textdatei mit Code-Schnipseln, die nicht für den Batchbetrieb konzipiert ist.

[3] `reverse(rest(reverse(l)))` entfernt das letzte Element der Liste, `append([0],...)` fügt vorn eine 0 an, womit eine Liste l_1 entsteht, in der alle Einträge um eine Position nach rechts verschoben sind. $l - l_1$ berechnet die Differenz der Listen, was in Maxima (und anderen CAS) als Subtraktion der entsprechenden Vektoren implementiert ist.

$(t, \log(d_t))$ übergeht. Damit lassen sich dann die Fittingparameter weitgehend stabil berechnen. Dafür müssen aber vorher Datenpunkte aussortiert werden, wo $d_t = 0$ ist.

Generell kann es sinnvoll sein, für ein gutes Fitting Datenpunkte unterhalb einer Schwelle auszusortieren. Eine solche Schwelle S ist als weiterer Parameter im Skript in der Funktion `FittingDelta(G,S)` vorgesehen. Für die meisten Datensätze ist die Schwelle 50 eine gute Wahl. G ist die Liste (t, y_t) der zu fittenden Datenpunkte.

Details sind im Skript `skript.m` zu finden.

Weitere Versuche, etwa mit einer Schar von Logistik-Funktionen, lassen sich mit dem CAS Maxima nicht erfolgreich zu Ende bringen.

5 Ergebnisse

Die Rechnungen werden für jedes der Länder nun wie folgt ausgeführt:

1. Fasse mit `l:getData(Land)` die drei Datensätze für das Land als Tripel von Listen (t, y_t) zusammen.
2. Berechne für jeden der drei Datensätze mit `getFittingFunctions(l,S)` das Fitting auf $(t, \log(d_t))$ gegen die Funktion $g(t)$ und verwende die berechneten drei Fittings, um Funktionen $h_1(t)$ (für infected), $h_2(t)$ (für recovered) und $h_3(t)$ (für dead) zu schätzen.
3. Erzeuge daraus einen Plot, welcher die Datenpunkte und die drei Kurven in verschiedenen Farben (rot für infected, grün für recovered, blau für dead) ausgibt.

Bei erfolgreichem Fitting ist eine gute Übereinstimmung der jeweiligen Kurve[4]

$$h(t) = A\left(\operatorname{erf}\left(B(t-m)\right) + 1\right)$$

mit den Datenpunkten zu verzeichnen. Die berechneten Parameter haben folgende Bedeutung:

- m – Tag, an dem die Spitze in den Inkrementdaten erreicht ist.
- $B = \frac{1}{s}$ – $\sigma = \frac{s}{\sqrt{2}}$ ist die Standardabweichung.
- $2A$ – Zahl der am Ende insgesamt betroffenen Personen.

Auf der Basis der Daten vom 07.04.2020 lassen sich folgende Szenarien (!) für die Länder Deutschland, Italien, Spanien, Österreich und Schweden erstellen, siehe auch die Abbildungen.

Alles begann in der chinesischen Provinz Hubei (mit 58.5 Mio. Einwohnern hat sie eine mit Italien vergleichbare Einwohnerzahl). Auch die dortige Entwicklung ist dargestellt[5].

[4] Für $x = \frac{m}{s} \approx 10$ kann $\operatorname{erf}(x) = 1$ gesetzt werden. Ist dieser Wert im Fitting deutlich anders, ist das Fitting unbrauchbar.

[5] Maxima hängt sich bei der Berechnung dieses Fittings allerdings schnell auf.

	m	s	A
Deutschland			
infected	91.42	13.28	65811
recovered	91.97	6.11	15154
dead	97.54	10.99	1820
Italien			
infected	86.24	15.17	77746
recovered	92.91	17.70	18404
dead	88.91	14.85	10548
Spanien			
infected	89.71	11.61	81387
recovered	113.59	20.14	158580
dead	92.00	11.02	8598
Österreich			
infected	87.22	11.39	6074
recovered	98.59	10.16	4336
dead	93.17	12.34	180
Schweden			
infected	97.62	19.25	6923
recovered	103.71	16.41	2106
dead	93.69	8.98	347
China, Provinz Hubei			
infected	41.46	13.23	26029
recovered	bad fitting		
dead	46.19	15.37	1669

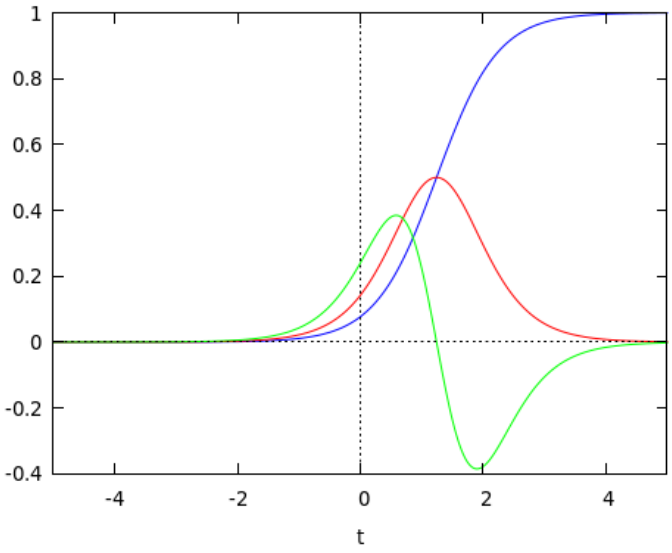
Natürlich müsste am Ende die Zahl der Infizierten mit der Summe der Zahlen der Genesenen und der Verstorbenen übereinstimmen. Die Schätzungen sind von einer solchen Invarianzforderung weit entfernt, was ein Licht auf die prognostische Qualität der Schätzungen für die weitere Zukunft (drei bis vier Wochen, also 30 Tage) wirft. Weitere Ergebnisse zeigen, dass selbst einem ohne Fehlermeldung zurückgegebenen Fitting nicht zu trauen ist und die Güte des Fittings genauer analysiert werden muss. Die Ergebnisse für die Provinz Hubei legen nahe, dass das verwendete Datenmodell für die Schätzung generell wenig geeignet ist, da die Zahl der Infektionen selbst im Nachgang zu klein geschätzt wird und für die Zahl der Genesenen kein brauchbares Fitting gefunden wird, obwohl sich die Zahlen in Bereichen bewegen, wie sie auch für europäische Länder charakteristisch sind.

6 Logistische Funktion

Generell ist ein Modell auf der Basis einer *Logistischen Funktion*

$$u(t) = \frac{K}{1 + C \cdot \exp(-rt)} \tag{L.1}$$

die anerkanntere Form der Modellierung der Ausbreitung einer Infektion, siehe dazu den entsprechenden Wikipedia-Eintrag.



Logistische Kurve $u(t) = \frac{1}{1+12\exp(-2t)}$ (blau) sowie deren erste (rot) und zweite Ableitung (grün) Ableitung

K steht dabei für die Sättigungsgrenze $\lim_{t\to\infty} u(t)$ und C ist üblicherweise als $C = \frac{K}{u(0)} - 1$ angeschrieben, was sich unmittelbar aus der Umstellung der Formel für $u(0)$ nach C ergibt. Der Wendepunkt dieser Funktion und damit das Maximum der ersten Ableitung liegt als Nullstelle der zweiten Ableitung bei $t_0 = \frac{\log(C)}{r}$, was sich unmittelbar mit Maxima berechnen lässt. Dies ist gerade der Median der Kurve, sollte also mit dem oben berechneten m zusammenfallen.

Derartige Funktionen lassen sich aber deutlich schlechter schätzen als Funktionen, die sich wie oben auf einfache Weise auf einen polynomialen Zusammenhang reduzieren lassen, da sie inhärent transzendent sind. Siehe hierzu aber die Arbeit von (Engel 2010) und die Modellierung mit GeoGebra in (Elschenbroich 2020).

In (Engel 2010) wird insbesondere darauf hingewiesen, dass sich mit einer guten Schätzung von K die anderen beiden Parameter mit einem linearen Fitting bestimmen lassen. Wir setzen dazu $C = \exp(c)$, womit sich Formel $(L.1)$ zu

$$\log\left(\frac{K}{u(t)} - 1\right) = c - rt \tag{L.2}$$

umstellen lässt. Der Parameter K ist dabei manuell zu schätzen, so dass die gefittete Kurve möglichst gut auf die Daten passt.

Im Skript ist das Ganze in einer Funktion `lFit(G,K0)` implementiert, der eine Liste G zu übergeben ist, in der vorab alle Datenpunkte mit $y_t \leq 10$ ausgefiltert werden. Es ergeben sich folgende Schätzungen für die Zahl der infizierten Personen:

Land	K	c	r	c/r
Deutschland	125000	18.058	0.198	91.11
Italien	145000	17.445	0.207	84.47
Österreich	13000	23.053	0.268	85.90
Spanien	160000	23.815	0269	88.42
China (Hebei)	70000	4.664	0.103	45.12

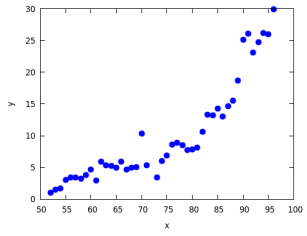
7 Die „Verdoppelungsdebatte“

In den letzten Tagen (Anfang April 2020) kommt eine Diskussion hoch, dass man die rigiden Beschränkungen erst aufheben könne, wenn „die Verdopplungszeit der Infektionen größer als 14 Tage“ sei.

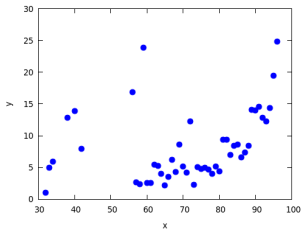
So meldet zum Beispiel der Deutschlandfunk am 04.04.2020[6]

> Die Verdopplungszeit der Ausbreitung von Coronavirus-Infektionen in Deutschland hat sich in den vergangenen Tagen verlangsamt.
>
> Für ganz Deutschland liegt sie nun bei 11,2 Tagen. Die Lage in den Bundesländern ist unterschiedlich. In den großen Flächenländern liegt die Verdopplungszeit in Nordrhein-Westfalen bei 13,1 Tagen, in Baden-Württemberg bei 12,5 Tagen und in Bayern bei 9,7 Tagen. In Berlin sind es inzwischen 12,8 Tage, in Hamburg 12,4. Im Saarland hingegen liegt die Verdoppelungszeit bei 5,5 Tagen, in Sachsen bei 11,0 Tagen.

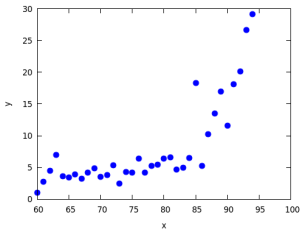
Auch wenn dies nicht immer deutlich wird, bezieht sich die Verdopplungszeit v_t auf die kumulierten Daten und steigt deshalb bereits durch die schiere Masse der Infizierten. Ist $y(t) = mt + n$ ein linearer Zusammenhang, so ergibt sich $v_t = \frac{y(t)}{m}$. Für einen annähernd linearen Zusammenhang kann man also $v_t = \frac{y(t)}{y'(t)}$ als Schätzung nehmen. Die Zahl lässt sich auch aus unseren Daten leicht berechnen: Ist y_t die kumulierte Zahl der Infizierten am Tag t und d_t die Zahl der Neuinfektionen, so ist nach $v_t = \frac{y_t}{d_t}$ Tagen eine Verdopplung der Infizierten erreicht, die Zuwachsrate d_t über diesen Zeitraum als konstant vorausgesetzt. Beide Datenreihen hatten wir schon oben extrahiert, so dass wir eine einfache Funktion `doublePlot(Land)` schreiben können, um die folgenden Plots zu erzeugen:



Italien



Deutschland



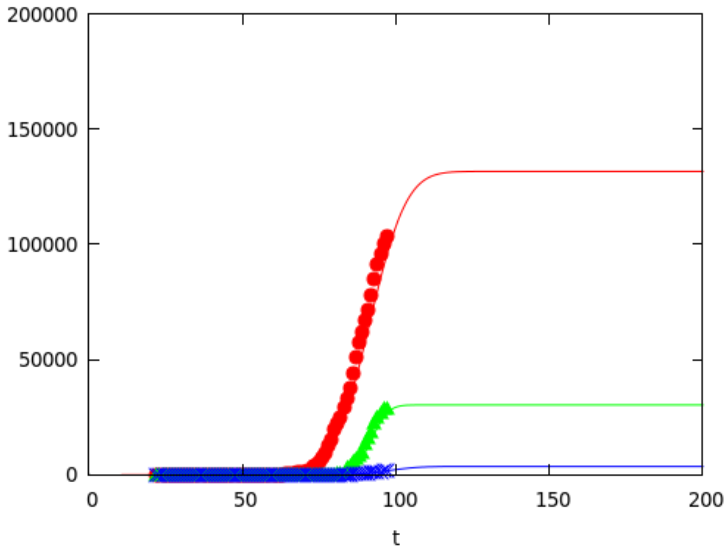
Österreich

[6] `https://www.deutschlandfunk.de/covid19-verdopplungszeit-der-coronavirus-infektionen-in.1939.de.html?drn:news_id=1117169`

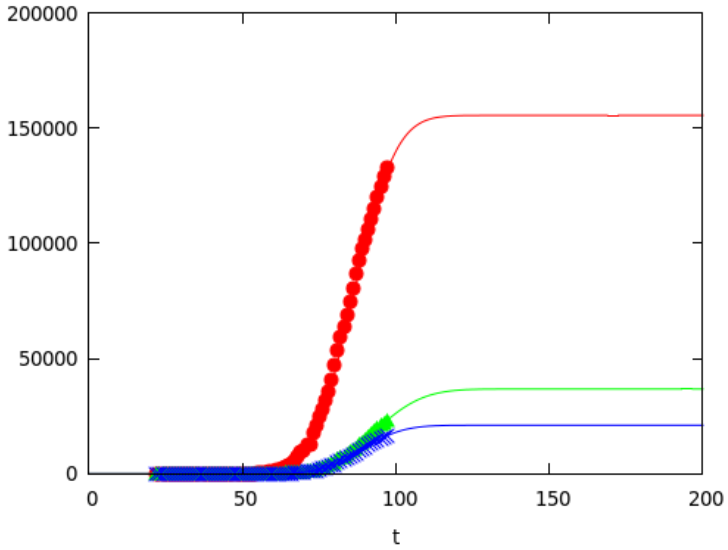
8 Literatur

- Hans-Jürgen Elschenbroich. Corona: Mathematik & Modellbildung. `https://www.geogebra.org/m/cfammtpe`. 2020.
- Joachim Engel. Parameterschätzen in logistischen Wachstumsmodellen. Stochastik in der Schule 30 (2010) 1, S. 13–18.
- Hans-Gert Gräbe. Computeralgebra im Abitur. Reihe „Eagle Starthilfe“. Eagle Verlag, Leipzig 2018. Siehe `https://hg-graebe.de/CAimAbitur/index.html`.
- Maxima. `http://maxima.sourceforge.net/de/`. Das CAS ist in Linux-Distributionen über den Paketmanager leicht zu installieren.
- Werner Rügemer. „Die USA haben das sicherste Gesundheitssystem der Welt“ – Die Johns Hopkins University und das globale Pandemien-Management. 01.04.2020. `https://www.nachdenkseiten.de/?p=59825`

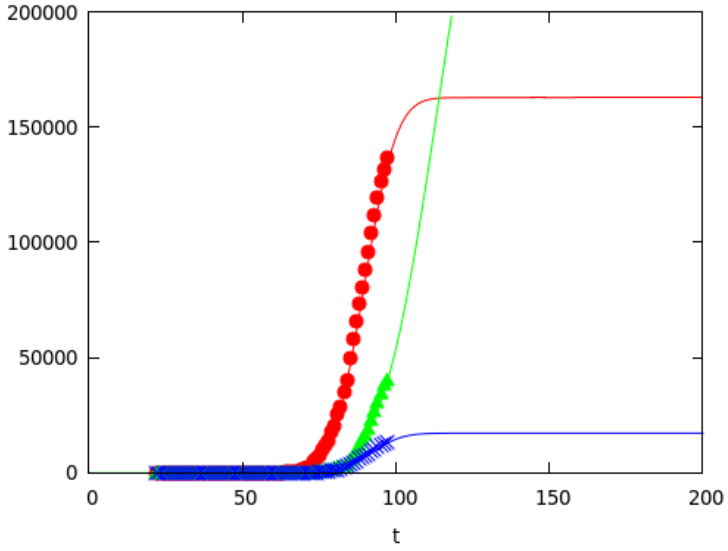
9 Grafiken



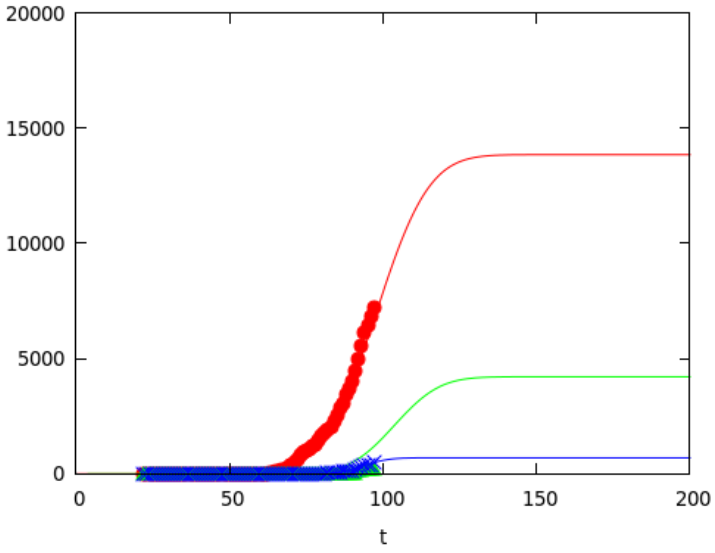
Szenario für Deutschland (82.9 Mio. Einwohner)



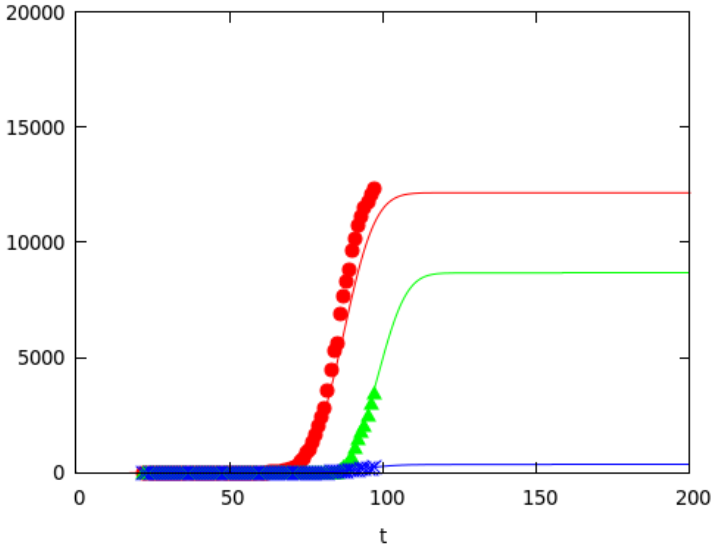
Szenario für Italien (60.4 Mio. Einwohner)



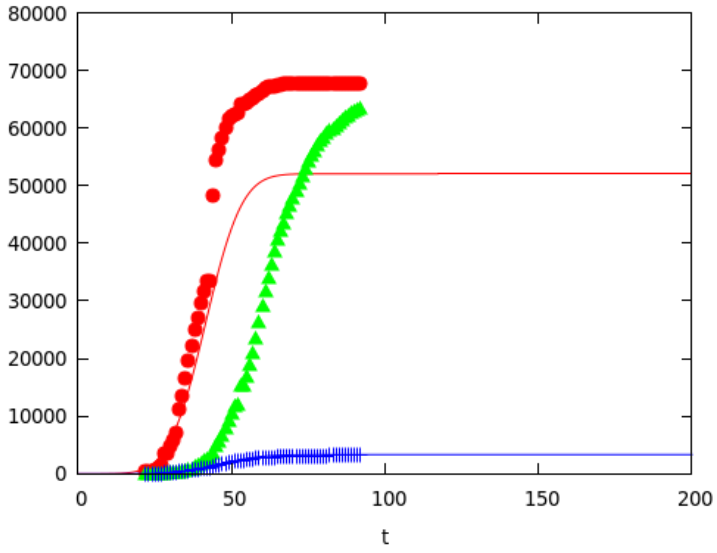
Szenario für Spanien (46.7 Mio. Einwohner)



Szenario für Schweden (66.4 Mio. Einwohner)



Szenario für Österreich (8.85 Mio. Einwohner)



Szenario für China, Provinz Hubei (58.5 Mio. Einwohner)