A Covid Analysis

Titouan Dupleich

Content

I. Data

- Data loaded from GitHub repository: https://github.com/titouanjd/ProgrammingForQA
- .csv and .xlsx files imported as pandas DataFrames
  - vaccinations
  - confirmed
  - deaths
  - recovered
  - population

II. Data Cleaning

a. Preliminary data cleaning – vaccinations dataset

Country	Date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations
Afghanistan	2021-02-22	NaN	NaN	NaN	NaN
Afghanistan	2021-02-23	NaN	NaN	NaN	1367.0
Afghanistan	2021-02-24	NaN	NaN	NaN	1367.0
Afghanistan	2021-02-25	NaN	NaN	NaN	1367.0
Afghanistan	2021-02-26	NaN	NaN	NaN	1367.0

II. Data Cleaning

a. Preliminary data cleaning – confirmed, deaths & recovered datasets

Country	Afghanistan	Albania	Algeria	Andorra	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria
Date										
2021-06-03	75119.0	132360.0	129976.0	13752.0	35140.0	1262.0	3884447.0	222978.0	30141.0	645834.0
2021-06-04	76628.0	132372.0	130361.0	13758.0	35307.0	1263.0	3915397.0	223050.0	30157.0	646167.0
2021-06-05	77963.0	132374.0	130681.0	13758.0	35594.0	1263.0	3939024.0	223143.0	30173.0	646438.0
2021-06-06	79224.0	132379.0	130958.0	13758.0	35772.0	1263.0	3955439.0	223180.0	30191.0	646800.0
2021-06-07	80841.0	132384.0	131283.0	13777.0	35854.0	1263.0	3977634.0	223212.0	30205.0	647079.0

II. Data Cleaning

a. Preliminary data cleaning – population dataset

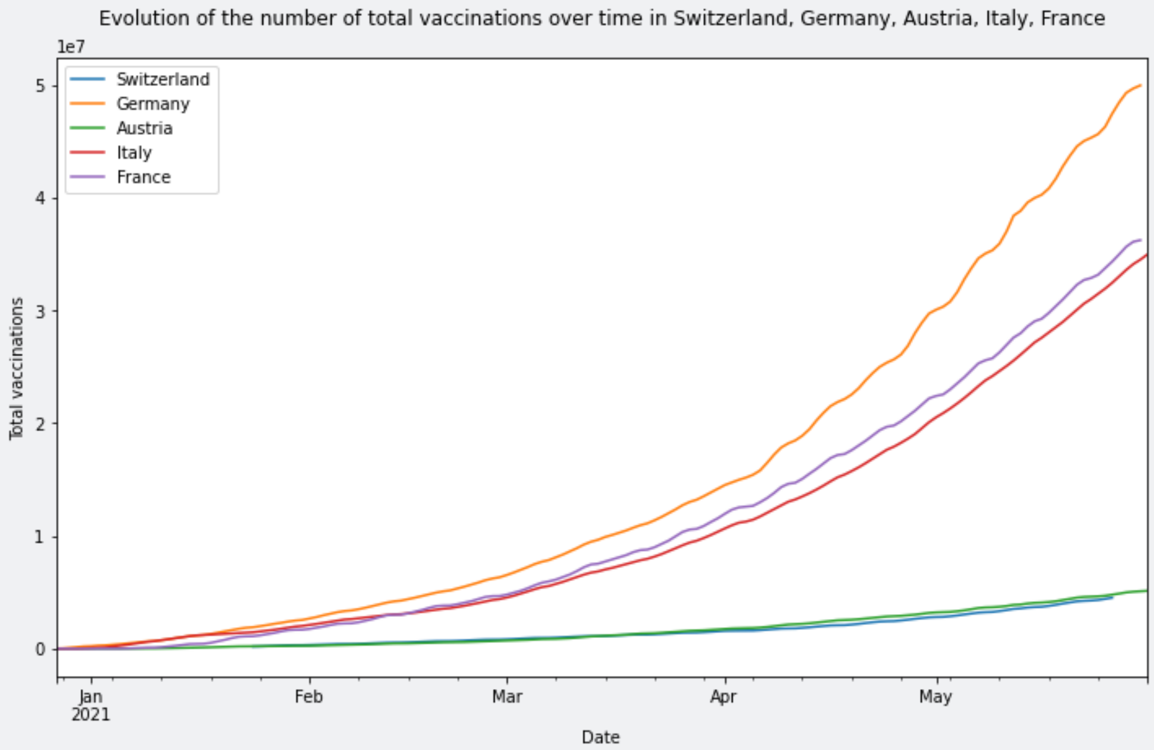
	Population
Country	
Afghanistan	39835428
Albania	2872934
Algeria	44616626
Andorra	77354
Angola	33933611

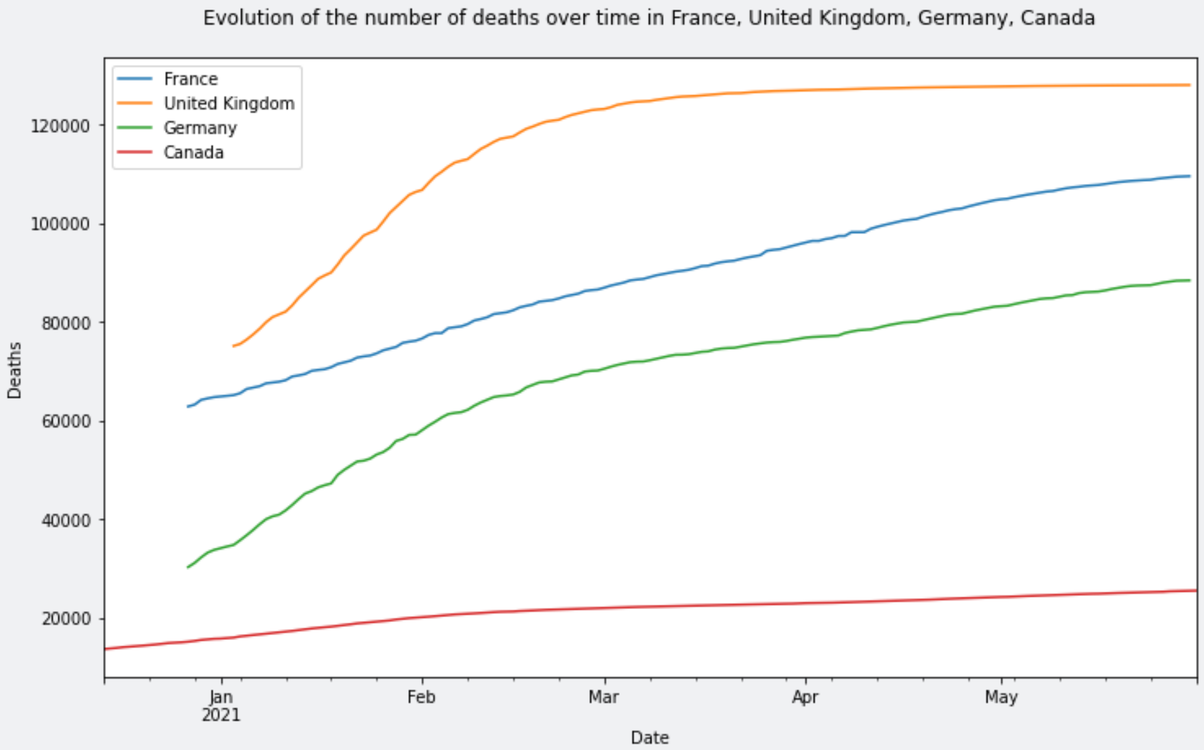
II. Data Cleaning

b. Combining datasets

- One **data** dictionary merging the **vaccinations**, **confirmed**, **deaths** and **recovered** datasets on date
- Keys are the country names
- Values contain measurements

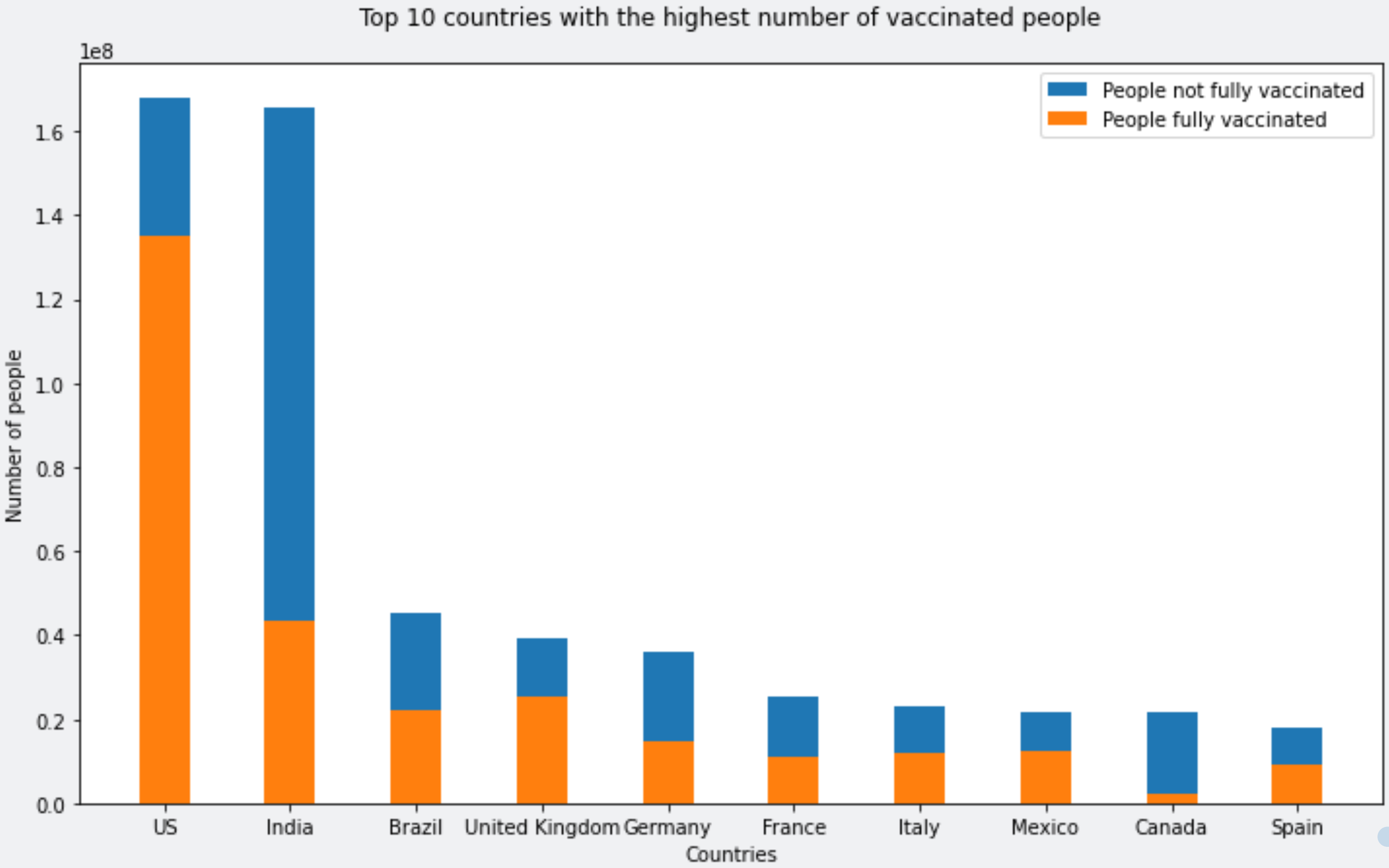
III. Plot data





- Can plot any variable for any combination of countries
- Curves, title, legend update automatically

III. Plot data



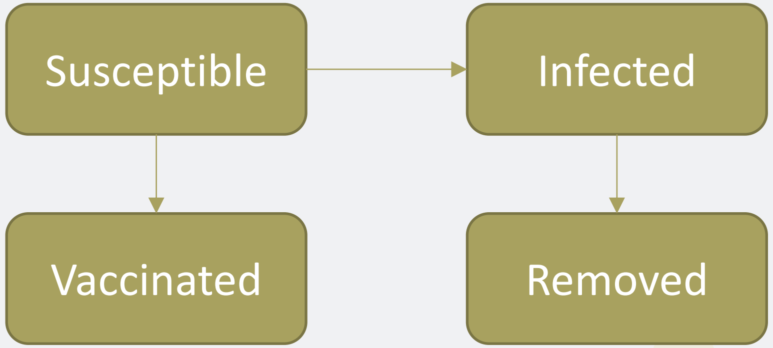
IV. SIRV Model

"Epidemiological model that computes the theoretical number of people infected with a contagious illness in a closed population over time."

Country population divided into 4 states

- S: Fraction of the population which is susceptible to have COVID-19 in the future
- I: Fraction of the population which is infected
- R: Fraction of the population which is removed (individuals who have either recovered or deceased from COVID-19)
- V: Fraction of the population which is vaccinated

$$S + I + R + V = 1$$



IV. SIRV Model

Other parameters

- β : infection rate (number of contacts per infected person per day)
- p : Fraction of the population subject to a successful vaccination per day
- $p_{g,0} = \sqrt[29]{\frac{v_0}{v_{-30}}} - 1$: Growth rate of p
- $\gamma = 1/n$: rate of recovery (number of recoveries per person per day) where n is the number of days required to recover from COVID-19

$R_0 = \frac{\beta}{\gamma}$: basic reproduction number (average number of secondary infections caused by one infected individual during his/her entire infectious period)

IV. SIRV Model

SIRV rates

- Rate of change of S: $\frac{dS}{dt} = -\beta SI - pS$
- Rate of change of I: $\frac{dI}{dt} = \beta SI - \gamma I$
- Rate of change of R: $\frac{dR}{dt} = \gamma I$
- Rate of change of V: $\frac{dV}{dt} = pS$

Model assumptions

- No births
- No immigration
- Recovery gives total immunity
- There's a fixed daily infection rate β
- There's a fixed recovery time
- The vaccine is 100% effective instantly from the first dose
- p will continue to grow at the same rate as it did in the past 30 days
- It takes exactly 14 days to recover from COVID-19

IV. SIRV Model

a. The SIRV function

```
def SIRV(S_today, I_today, R_today, V_today, p, p_growth, beta=0.4, ndays=180, dt=0.1, ndays_recovery=14):

  npts = int(ndays/dt) # number of data points to plot
  gamma = 1.0 / ndays_recovery # recovery rate

  # create arrays for storing S, I, R and t
  S = np.zeros(npts) # susceptible
  I = np.zeros(npts) # infective
  R = np.zeros(npts) # removed
  V = np.zeros(npts) # vaccinated
  t = np.arange(npts) * dt # number of days for which we want to predict the model

  # values at day 0
  S[0] = S_today
  I[0] = I_today
  R[0] = R_today
  V[0] = V_today

  # compute the expected values for the upcoming periods
  for i in range(npts - 1):
    # update the value of p with p_growth for each day
    if i % (1 / dt) == 0:
      p *= (1 + p_growth)

    # update S, I, R and V values
    S[i + 1] = S[i] - (beta * S[i] * I[i] + p * S[i]) * dt
    I[i + 1] = I[i] + (beta * S[i] * I[i] - gamma * I[i]) * dt
    R[i + 1] = R[i] + (gamma * I[i]) * dt
    V[i + 1] = V[i] + (p * S[i]) * dt # could also be written as: V[i + 1] = 1 - S[i+1] - I[i+1] - R[i+1]

  # calculate the basic reproduction number
  R_0 = beta / gamma

  return t, S, I, R, V, ndays, dt, R_0
```

IV. SIRV Model

a. The SIRV function

```python
def SIRV(S_today, I_today, R_today, V_today, p, p_growth, beta=0.4, ndays=180, dt=0.1, ndays_recovery=14):

    npts = int(ndays/dt) # number of data points to plot
    gamma = 1.0 / ndays_recovery # recovery rate

    # create arrays for storing S, I, R and t
    S = np.zeros(npts) # susceptible
    I = np.zeros(npts) # infective
```

Function parameters

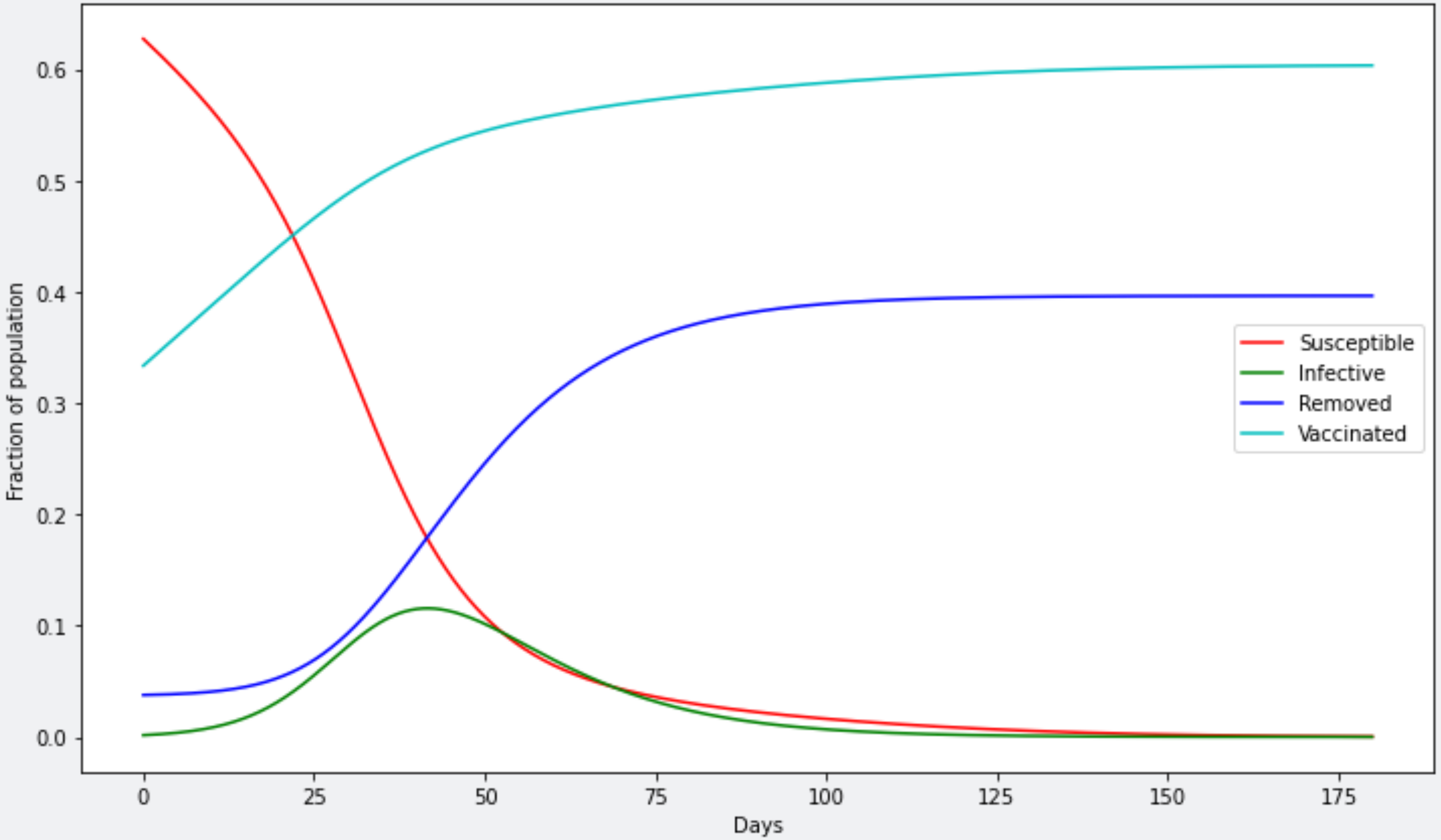
- `S_today`: fraction of the population which is susceptible at time 0
- `I_today`: fraction of the population which is infected at time 0
- `R_today`: fraction of the population which is removed at time 0
- `V_today`: fraction of the population which is vaccinated at time 0
- `p`: fraction of the population subject to a successful vaccination per day
- `p_growth`: daily growth rate of `p` based on past historical growth rate
- `beta`: infection rate
- `ndays`: number of days for which we want to predict the model
- `dt`: time step in days
- `ndays_recovery`: no of days it takes to recover from COVID

```python
to predict the model

as: V[i + 1] = 1 - S[i+1] - I[i+1] - R[i+1]

    # calculate the basic reproduction number
    R_0 = beta / gamma

    return t, S, I, R, V, ndays, dt, R_0
```

IV. SIRV Model

a. The SIRV function

Predicted susceptible, infective, removed and vaccinated proportions over the next 180 days (updated every 0.1 day(s)) in Switzerland (computed on 2021-05-26)

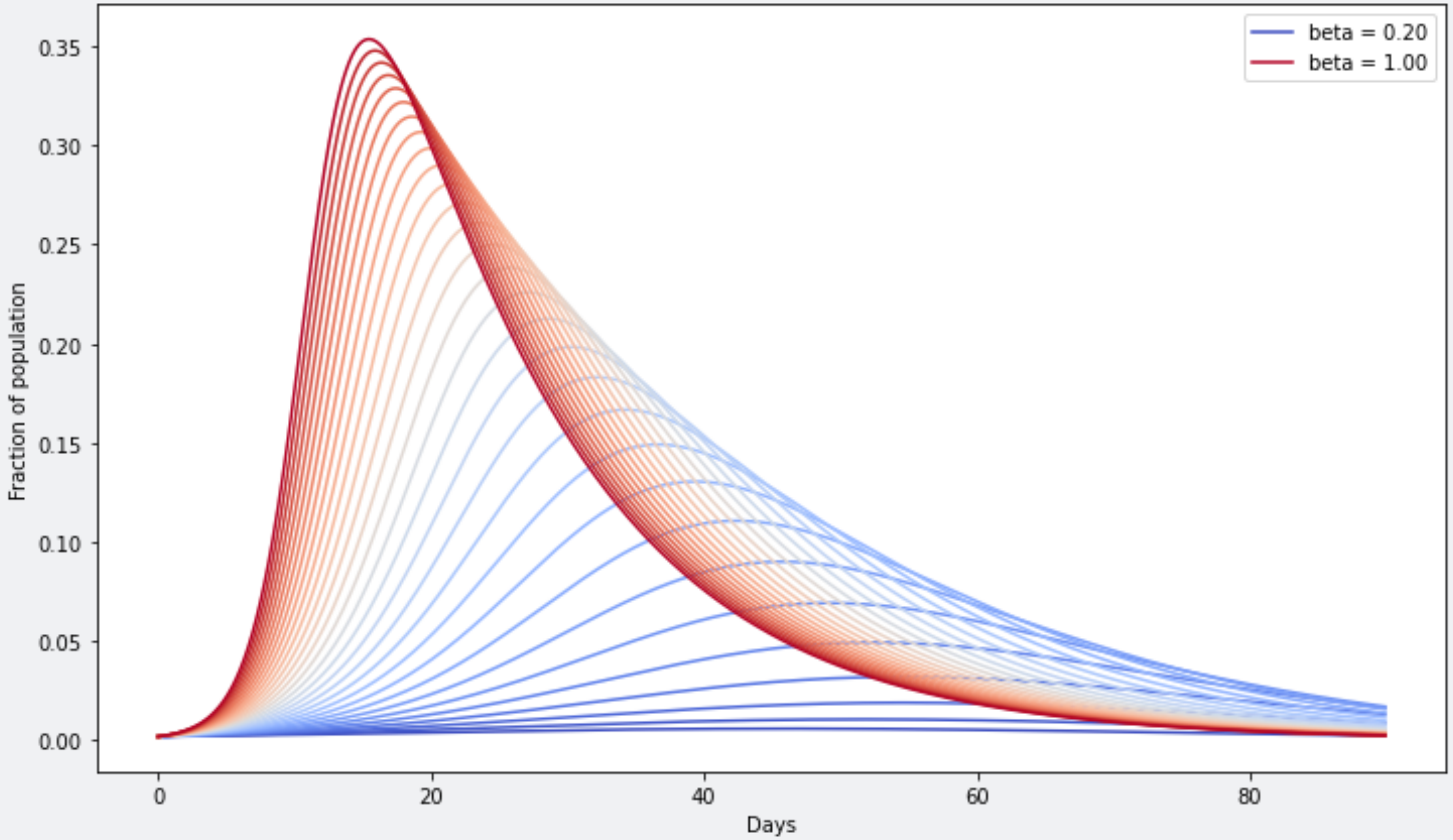


- $R_0 = 5.6 > 1$: the COVID infection will spread
- Max percentage of infected people at the same time: 11.56% in 42 days

IV. SIRV Model

b. SIRV model simulations for different values of β

Predicted infected proportion for 30 different values of beta in [0.20, 1.00] over the next 90 days (updated every 0.1 day(s)) in Switzerland (computed on 2021-05-26)



V. References

Datasets

- **vaccinations**: https://www.kaggle.com/gpreda/covid-world-vaccination-progress
- **confirmed**, **deaths** & **recovered**: https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_time_series from https://ourworldindata.org/coronavirus#coronavirus-country-profiles
- **population**: https://ourworldindata.org/search?q=population

SIRV Model

- Main SIR explanation and code: https://youtube.com/playlist?list=PLN0b-Zk854ab_KQ_pa8YkAYK1FhuombtY
- "Vaccinated V" component of the model: http://hplgit.github.io/prog4comp/doc/pub/._p4c-solarized-Python021.html
- R_0 explanation: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1804098/