Project Report

-- Correlation between the Covid-19 vaccination rate in various school types and regions in Schleswig-Holstein, Germany --

Jan Baumgart

22541155

Master's Degree Program IIS

Professorship for Open-Source Software

23rd of December 2023

Table of Contents

Introduction 1

Motivation 1

Goal of the Project 1

Methods for Analysis 2

Data sources of the Project 2

The Data Pipeline preparation 4

The Data Pipeline loading and processing 6

Results 10

Question 1 10

Question 2 13

Conclusion 17

Discussion of the results and Context 17

Limitations and potential improvements in the future 17

Introduction

Motivation

The ongoing COVID-19 situation really got us thinking about how I can make sense of it all. It is important to know if and how vaccination rates are connected with the number of cases, especially in crucial places like schools. I want to take a closer look at how the vaccination rates in the population and the number of COVID-19 cases in schools might be linked, for that I are using Open Data from Schleswig-Holstein (SH) from different sources, but more to that later. I will break down the schools into several types and looking at how things play out in different School types as well as in different regions of Schleswig-Holstein. The idea is to figure out how different School types react to the pandemic and if there are significant differences between them, so that in the future, I can make better decisions when it comes to vaccinations or restrictions.
It is not just about numbers and diagraphs; it is about finding real-world knowledge to manage COVID-19 better.

Goal of the Project

My main goal here is as mentioned before: I want to untangle the connection between the COVID-19 vaccination rate and the number of cases in schools, respectively different regions of Schleswig-Holstein if there is one.

By breaking down schools into types and SH into its regions / counties, I hope to gain insights or even find patterns. I will also combine the numbers and graphs with context in my analysis. The ultimate goal is to produce recommendations that can actually help in planning future vaccination or political projects to deal with pandemics in SH. So, this project is all about me making sense of things and finding useful suggestions for handling COVID-19 (or other diseases) in the future.

Methods for Analysis

Data sources of the Project

Following I will introduce and quickly summarize my data sources for this Project.
If you are looking for more insights into the sources themselves, you can take a look into the 'Exploring" part of the *data_exploration.ipynb* Notebook.

- **Data Source 1: COVID-19 Vaccination Rates…**
        - Metadata: *https://www.govdata.de/web/guest/suchen/-/details/covid-19-impfungen-in-deutschland0aef2*
        - License: *Public Domain Mark 1.0 (PDM)*
    - **Data Source 1.1: … by federal states**
        - Type: *CSV*
        - Data: *https://github.com/robert-koch-institut/COVID-19-Impfungen_in_Deutschland/blob/main/Deutschland_Bundeslaender_COVID-19-Impfungen.csv*
        - Summary: *A collection of data showing the vaccination rates broken down by German federal states.*
    - **Data Source 1.2: … by counties**
        - Type: *CSV*
        - Data: *https://github.com/robert-koch-institut/COVID-19-Impfungen_in_Deutschland/blob/main/Deutschland_Landkreise_COVID-19-Impfungen.csv*
        - Summary: *A collection of data showing the vaccination rates broken down by counties (in SH).*

- **Data Source 2: Number of COVID-19 cases at schools by school type**
  - Metadata URL: *https://www.govdata.de/web/guest/suchen/-/details/anzahl-der-covid-19-falle-an-schulen-nach-schularten*
  - Data URL: *https://opendata.schleswig-holstein.de/dataset/08263e93-6e2c-4034-888a-31ac33c91bfe/resource/eddd721d-1789-4d89-85a1-a43ac6e1fd7f/download/data.csv*
  - License: *Public Domain Mark 1.0 (PDM)*
  - Data Type: *CSV*
  - Summary: *The data source provides data on COVID-19 cases at schools, separated by type of school in Germany.*

- **Data source 3: Pupils in Germany by School type in 2021/2022**
  - Metadata: *https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Bildung-Forschung-Kultur/Schulen/_inhalt.html#_ccbho9pou*
  - Data: *https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Bildung-Forschung-Kultur/Schulen/Publikationen/Downloads-Schulen/statistischer-bericht-allgemeinbildende-schulen-2110100227005.xlsx?__blob=publicationFile*
  - License: *Data license Germany – attribution – version 2.0* (https://www.govdata.de/dl-de/by-2-0)
  - Data Type: *XLSX*
  - Summary: *The data from the Federal Statistical Office shows the number of pupils, broken down by federal state, school year, gender, school type and year group*

- **Data source 4: Population of Germany by federal states and counties.**
  - Metadata URL: *https://www.destatis.de/DE/Themen/Laender-Regionen/Regionales/Gemeindeverzeichnis/Administrativ/04-kreise.html*
  - Data URL: *https://www.destatis.de/DE/Themen/Laender-Regionen/Regionales/Gemeindeverzeichnis/Administrativ/04-kreise.xlsx?__blob=publicationFile*
  - Data Type: *XLSX*
  - License: *Data license Germany – attribution – version 2.0*
  - Summary: *This data (also from the Federal Statistical Office) provides an overview of all independent cities and districts by area, population, and population density*

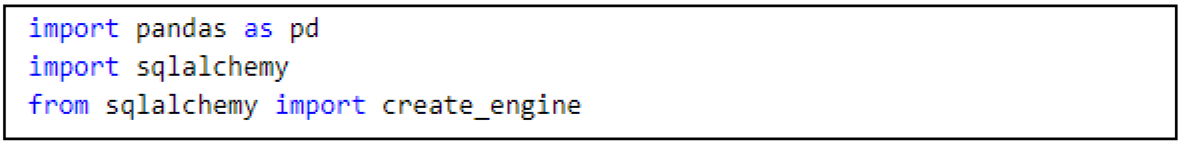
- **Data source 5: Number of COVID-19 cases at schools by counties**
    - Metadata: *https://www.govdata.de/web/guest/suchen/-/details/anzahl-der-covid-19-falle-an-schulen-nach-kreisen*
    - Data: *https://opendata.schleswig-holstein.de/dataset/acd37a65-b4ad-4abd-b318-a39fc37838f7/resource/5c7d0685-5ec2-441c-ab9e-dc5a691bef29/download/data.csv*
    - License: *Public Domain Mark 1.0 (PDM)*
    - Data Type: *CSV*
    - Summary: *The data shows the number of Covid Cases in SH, divided by the counties in SH over the time.*

The Data Pipeline preparation

For loading the data from the URLs and automatically store it in an easy manageable, scalable, and analyzable SQLite Database, I use an automated Data pipeline that I programmed in *Python*.

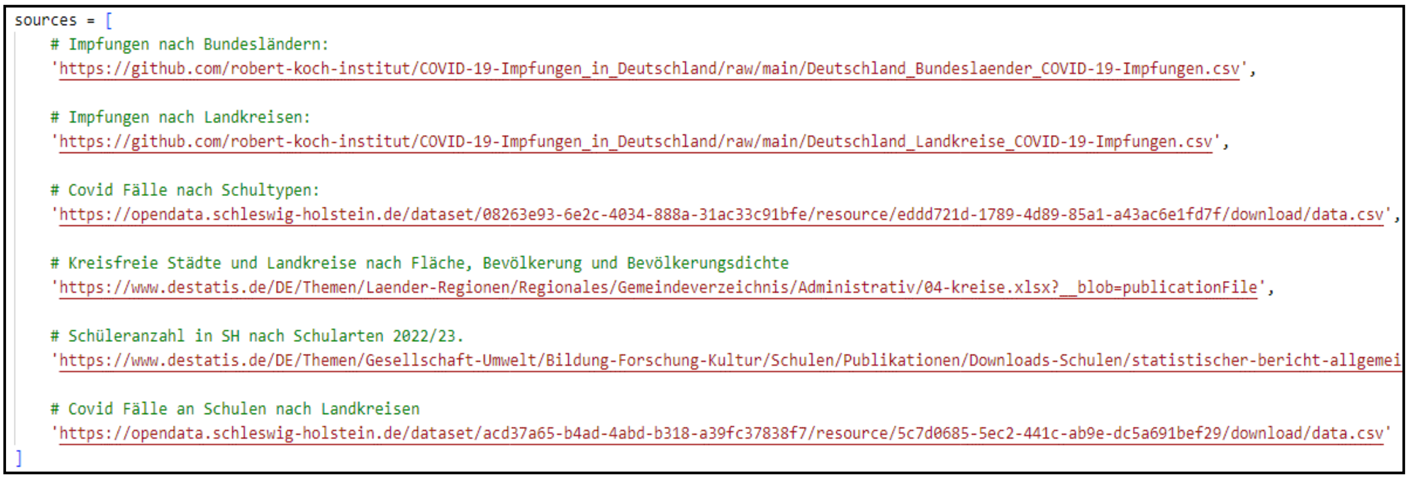
The Python script sequentially accesses the sources to retrieve the raw data. For each data source, I have implemented specific procedures to clean and structure the information appropriately (more to that in the next section).

I First imported a few packages in python, that are necessary to run the pipeline.



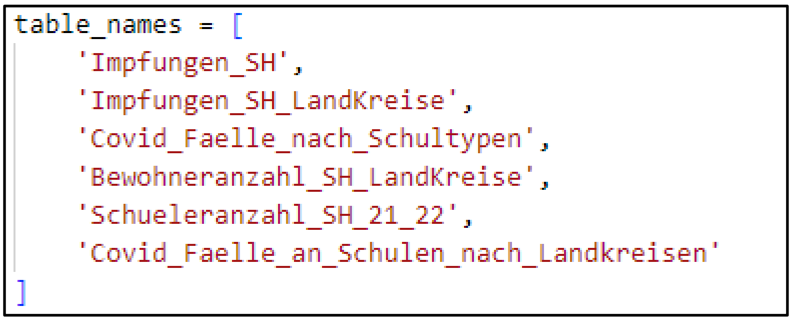
```
import pandas as pd
import sqlalchemy
from sqlalchemy import create_engine
```

Next, I defined a List '*sources*' that contains the URLs to the sources I'm using for the Project.



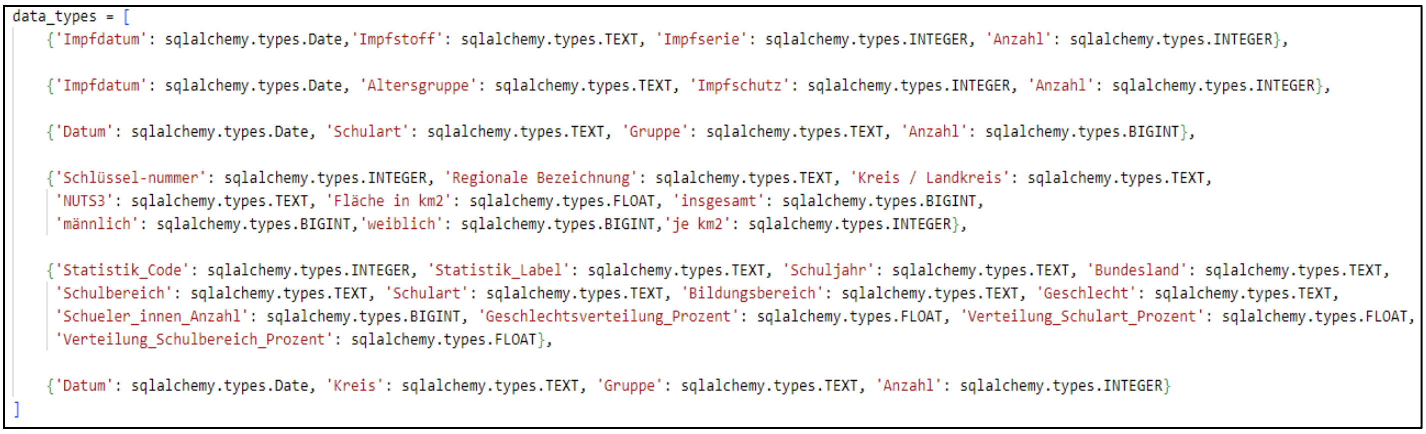
```
sources = [
    # Impfungen nach Bundesländern:
    'https://github.com/robert-koch-institut/COVID-19-Impfungen_in_Deutschland/raw/main/Deutschland_Bundeslaender_COVID-19-Impfungen.csv',

    # Impfungen nach Landkreisen:
    'https://github.com/robert-koch-institut/COVID-19-Impfungen_in_Deutschland/raw/main/Deutschland_Landkreise_COVID-19-Impfungen.csv',

    # Covid Fälle nach Schultypen:
    'https://opendata.schleswig-holstein.de/dataset/08263e93-6e2c-4034-888a-31ac33c91bfe/resource/eddd721d-1789-4d89-85a1-a43ac6e1fd7f/download/data.csv',

    # Kreisfreie Städte und Landkreise nach Fläche, Bevölkerung und Bevölkerungsdichte
    'https://www.destatis.de/DE/Themen/Laender-Regionen/Regionales/Gemeindeverzeichnis/Administrativ/04-kreise.xlsx?__blob=publicationFile',

    # Schüleranzahl in SH nach Schularten 2022/23.
    'https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Bildung-Forschung-Kultur/Schulen/Publikationen/Downloads-Schulen/statistischer-bericht-allgemei

    # Covid Fälle an Schulen nach Landkreisen
    'https://opendata.schleswig-holstein.de/dataset/acd37a65-b4ad-4abd-b318-a39fc37838f7/resource/5c7d0685-5ec2-441c-ab9e-dc5a691bef29/download/data.csv'
]
```

After that, I created a List ‘*table_names*’ that stores the names in the same order as the sources.



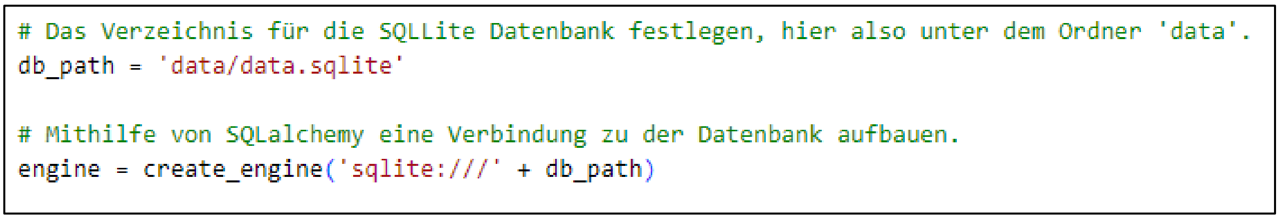
```
table_names = [
    'Impfungen_SH',
    'Impfungen_SH_LandKreise',
    'Covid_Faelle_nach_Schultypen',
    'Bewohneranzahl_SH_LandKreise',
    'Schueleranzahl_SH_21_22',
    'Covid_Faelle_an_Schulen_nach_Landkreisen'
]
```

Then I also add a list with libraries that define the datatypes of the columns in the database. Again, with the same order as the lists before.



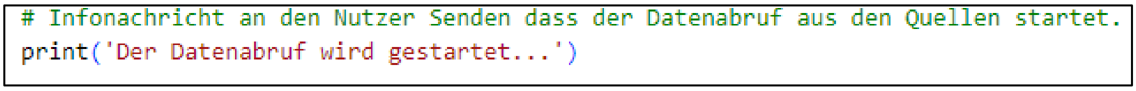
```
data_types = [
    {'Impfdatum': sqlalchemy.types.Date,'Impfstoff': sqlalchemy.types.TEXT, 'Impfserie': sqlalchemy.types.INTEGER, 'Anzahl': sqlalchemy.types.INTEGER},

    {'Impfdatum': sqlalchemy.types.Date, 'Altersgruppe': sqlalchemy.types.TEXT, 'Impfschutz': sqlalchemy.types.INTEGER, 'Anzahl': sqlalchemy.types.INTEGER},

    {'Datum': sqlalchemy.types.Date, 'Schulart': sqlalchemy.types.TEXT, 'Gruppe': sqlalchemy.types.TEXT, 'Anzahl': sqlalchemy.types.BIGINT},

    {'Schlüssel-nummer': sqlalchemy.types.INTEGER, 'Regionale Bezeichnung': sqlalchemy.types.TEXT, 'Kreis / Landkreis': sqlalchemy.types.TEXT,
     'NUTS3': sqlalchemy.types.TEXT, 'Fläche in km2': sqlalchemy.types.FLOAT, 'insgesamt': sqlalchemy.types.BIGINT,
     'männlich': sqlalchemy.types.BIGINT,'weiblich': sqlalchemy.types.BIGINT,'je km2': sqlalchemy.types.INTEGER},

    {'Statistik_Code': sqlalchemy.types.INTEGER, 'Statistik_Label': sqlalchemy.types.TEXT, 'Schuljahr': sqlalchemy.types.TEXT, 'Bundesland': sqlalchemy.types.TEXT,
     'Schulbereich': sqlalchemy.types.TEXT, 'Schulart': sqlalchemy.types.TEXT, 'Bildungsbereich': sqlalchemy.types.TEXT, 'Geschlecht': sqlalchemy.types.TEXT,
     'Schueler_innen_Anzahl': sqlalchemy.types.BIGINT, 'Geschlechtsverteilung_Prozent': sqlalchemy.types.FLOAT, 'Verteilung_Schulart_Prozent': sqlalchemy.types.FLOAT,
     'Verteilung_Schulbereich_Prozent': sqlalchemy.types.FLOAT},

    {'Datum': sqlalchemy.types.Date, 'Kreis': sqlalchemy.types.TEXT, 'Gruppe': sqlalchemy.types.TEXT, 'Anzahl': sqlalchemy.types.INTEGER}
]
```

Then finally I define the path for the database and use SQLalchemy to build and connect it.



```
# Das Verzeichnis für die SQLLite Datenbank festlegen, hier also unter dem Ordner 'data'.
db_path = 'data/data.sqlite'

# Mithilfe von SQLalchemy eine Verbindung zu der Datenbank aufbauen.
engine = create_engine('sqlite:///' + db_path)
```

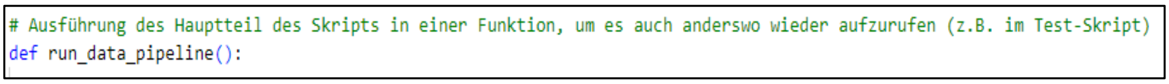
Following I print a message to the user that the Data loading is starting and the now the actual loading of the data and its processing starts in the next section.



```
# Infonachricht an den Nutzer Senden dass der Datenabruf aus den Quellen startet.
print('Der Datenabruf wird gestartet...')
```

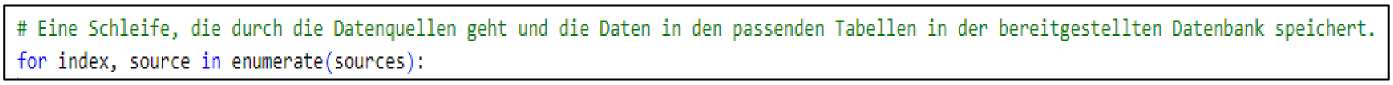
The Data Pipeline loading and processing

The heart of the Data pipeline is now the actual loading, cleaning, and transformation of the data. I define a Function that holds the name '*run_data_pipeline'* that takes no input and is simply used to execute the pipeline, that is useful when running tests with *pytest* for example and check the output.



```python
# Ausführung des Hauptteil des Skripts in einer Funktion, um es auch anderswo wieder aufzurufen (z.B. im Test-Skript)
def run_data_pipeline():
```

The first step is a loop that iterates through the list of sources, to ensure a sequential order of storing the data in the database.

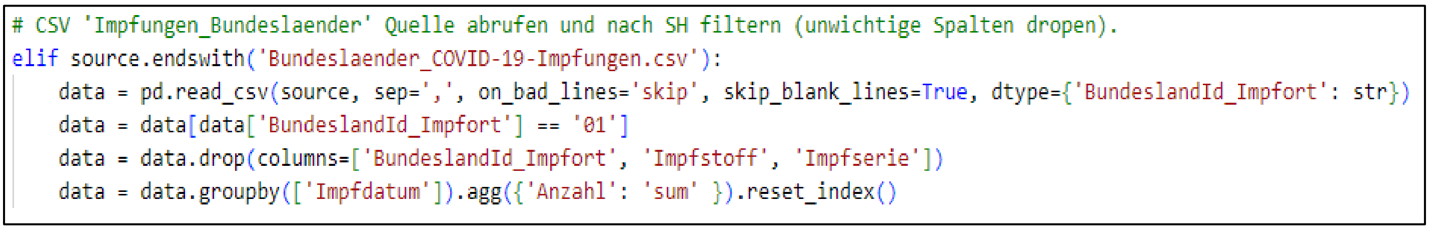


```python
# Eine Schleife, die durch die Datenquellen geht und die Daten in den passenden Tabellen in der bereitgestellten Datenbank speichert.
for index, source in enumerate(sources):
```

*(**Attention**, the following order is not the same as in the code but as in the list of sources before)*

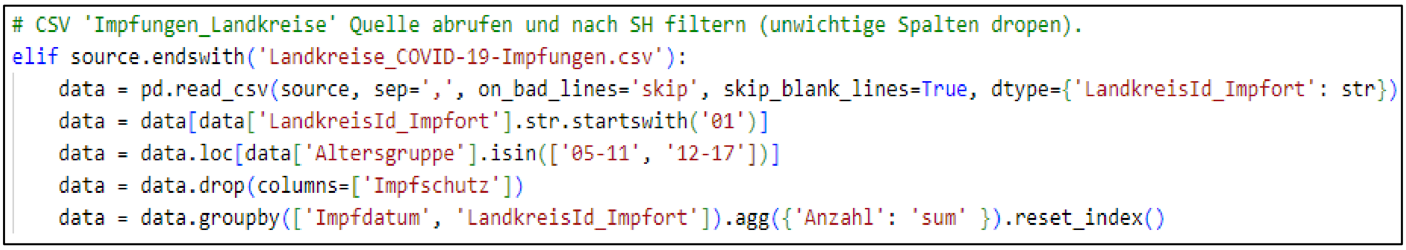
(I will also not explain same procedure twice, to keep it efficient)

1. I first load the data from the Source that stores the data about the number vaccinations by federal states in Germany.
    a. I first check with a 'if' statement if the source is the one, I want.
    b. I read the CSV document and define the datatype of 'BundeslandId_Impfort' separately, because it needs to be assigned earlier for the next step.
    c. I filter out only the data that has the ID 01 (Schleswig-Holstein)
    d. I then drop all the columns that are no longer needed for this project.
    e. I then define that only the 'Impfdatum' should remain with its absolute number of vaccinations on that day.



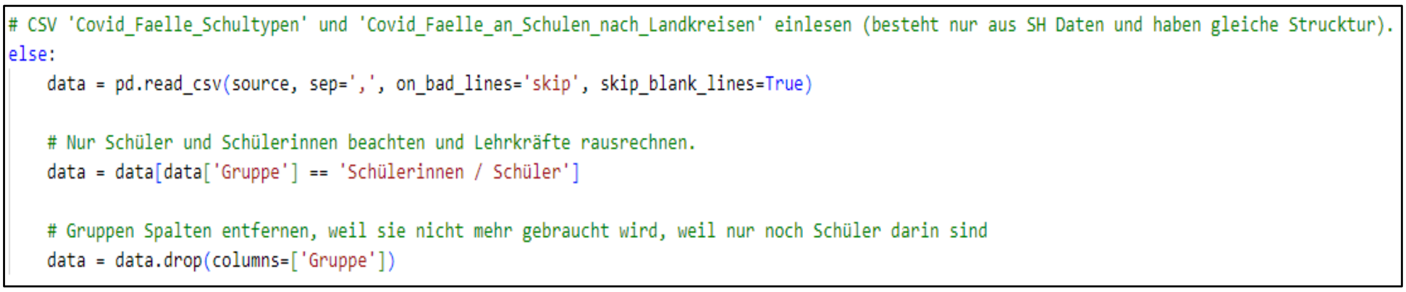
```python
# CSV 'Impfungen_Bundeslaender' Quelle abrufen und nach SH filtern (unwichtige Spalten dropen).
elif source.endswith('Bundeslaender_COVID-19-Impfungen.csv'):
    data = pd.read_csv(source, sep=',', on_bad_lines='skip', skip_blank_lines=True, dtype={'BundeslandId_Impfort': str})
    data = data[data['BundeslandId_Impfort'] == '01']
    data = data.drop(columns=['BundeslandId_Impfort', 'Impfstoff', 'Impfserie'])
    data = data.groupby(['Impfdatum']).agg({'Anzahl': 'sum' }).reset_index()
```

2. I continue with the second source ‘*Impfungen_SH_LandKreise*’.
    a. Here I make also sure that the ‘*Altersgruppe*’ only consists of people between 5-17 (which I take as pupil) before summing them up by *Impfdatum* und *LandkreisId_Impfort*.



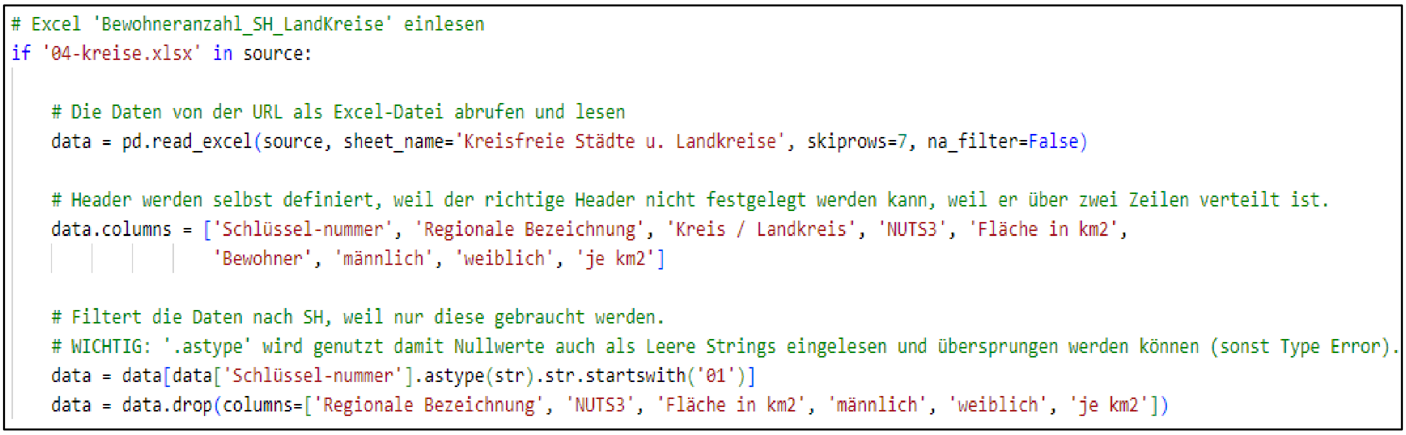
```
# CSV 'Impfungen_Landkreise' Quelle abrufen und nach SH filtern (unwichtige Spalten dropen).
elif source.endswith('Landkreise_COVID-19-Impfungen.csv'):
    data = pd.read_csv(source, sep=',', on_bad_lines='skip', skip_blank_lines=True, dtype={'LandkreisId_Impfort': str})
    data = data[data['LandkreisId_Impfort'].str.startswith('01')]
    data = data.loc[data['Altersgruppe'].isin(['05-11', '12-17'])]
    data = data.drop(columns=['Impfschutz'])
    data = data.groupby(['Impfdatum', 'LandkreisId_Impfort']).agg({'Anzahl': 'sum' }).reset_index()
```

3. I continue with ‘*Covid_Faelle_nach_Schultyp*’.
    a. I only use data from pupil and then drop the ‘*Gruppe*’ column because it’s no longer needed.



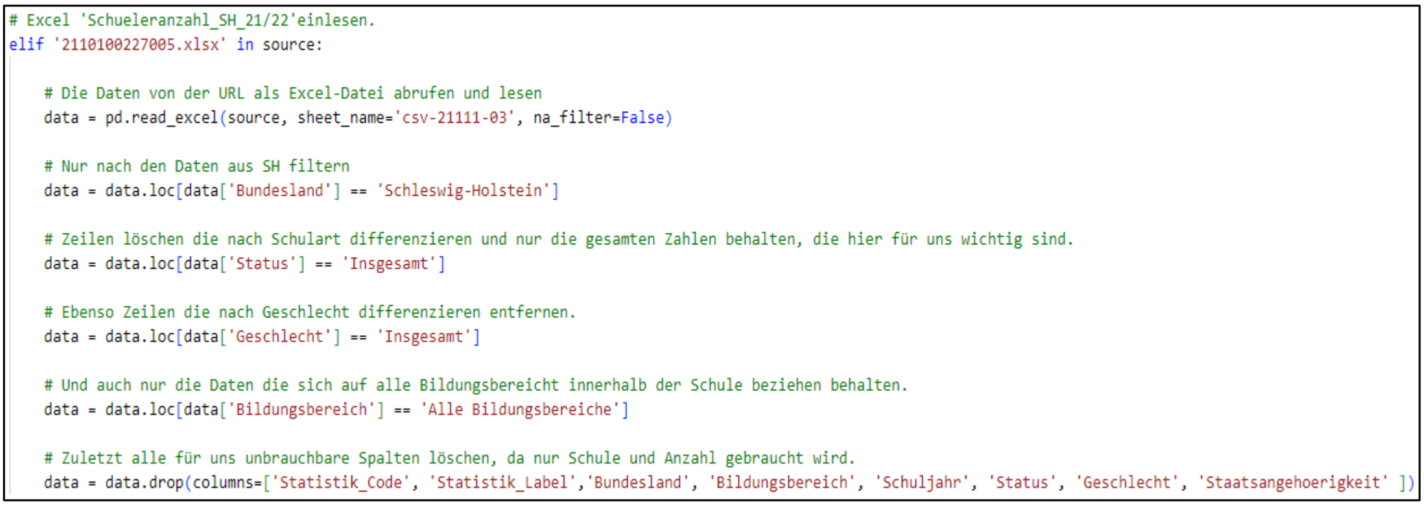
```
# CSV 'Covid_Faelle_Schultypen' und 'Covid_Faelle_an_Schulen_nach_Landkreisen' einlesen (besteht nur aus SH Daten und haben gleiche Strucktur).
else:
    data = pd.read_csv(source, sep=',', on_bad_lines='skip', skip_blank_lines=True)

    # Nur Schüler und Schülerinnen beachten und Lehrkräfte rausrechnen.
    data = data[data['Gruppe'] == 'Schülerinnen / Schüler']

    # Gruppen Spalten entfernen, weil sie nicht mehr gebraucht wird, weil nur noch Schüler darin sind
    data = data.drop(columns=['Gruppe'])
```

4. Now we go to *'Bewohneranzahl_SH_LandKreise'*
    a. *Here I need to give the columns names manually because the actual head in the excel is split in more than one row.*
    b. The ‘*Schlüsslnummer*’ needs to start with 01 which indicates SH.



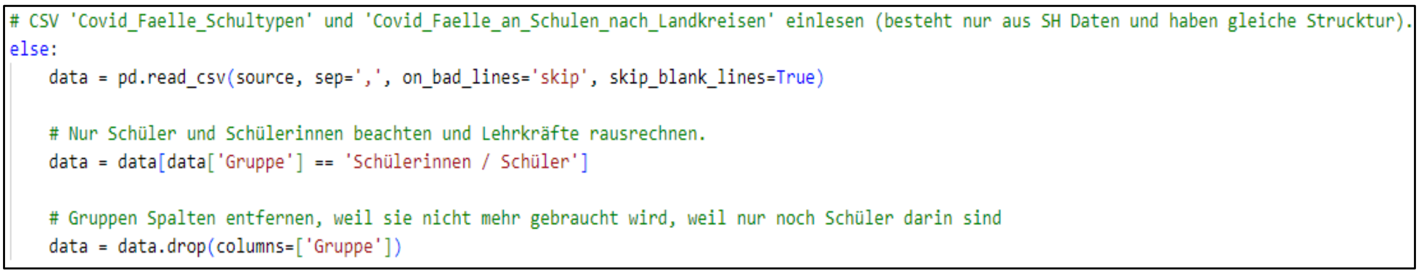
```
# Excel 'Bewohneranzahl_SH_LandKreise' einlesen
if '04-kreise.xlsx' in source:

    # Die Daten von der URL als Excel-Datei abrufen und lesen
    data = pd.read_excel(source, sheet_name='Kreisfreie Städte u. Landkreise', skiprows=7, na_filter=False)

    # Header werden selbst definiert, weil der richtige Header nicht festgelegt werden kann, weil er über zwei Zeilen verteilt ist.
    data.columns = ['Schlüssel-nummer', 'Regionale Bezeichnung', 'Kreis / Landkreis', 'NUTS3', 'Fläche in km2',
                    'Bewohner', 'männlich', 'weiblich', 'je km2']

    # Filtert die Daten nach SH, weil nur diese gebraucht werden.
    # WICHTIG: '.astype' wird genutzt damit Nullwerte auch als Leere Strings eingelesen und übersprungen werden können (sonst Type Error).
    data = data[data['Schlüssel-nummer'].astype(str).str.startswith('01')]
    data = data.drop(columns=['Regionale Bezeichnung', 'NUTS3', 'Fläche in km2', 'männlich', 'weiblich', 'je km2'])
```

5. Next is the source *'Schueleranzahl_SH_21_22'*
    a. I filter out the total number of pupils of the analyzable school types in SH. (e.g. not professional schools)



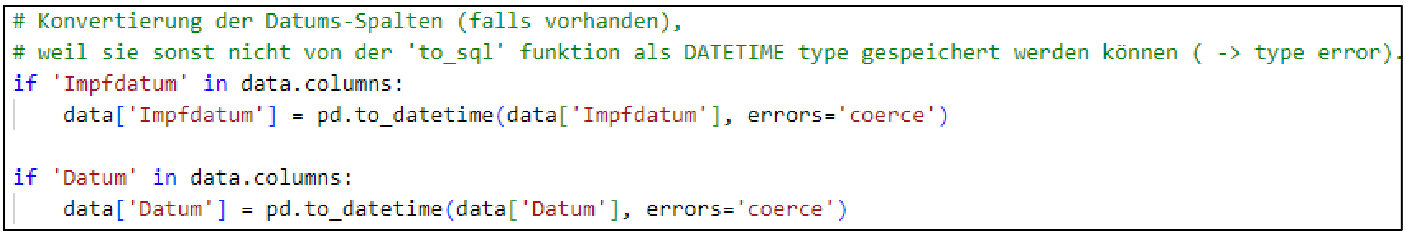
```
# Excel 'Schueleranzahl_SH_21/22'einlesen.
elif '2110100227005.xlsx' in source:

    # Die Daten von der URL als Excel-Datei abrufen und lesen
    data = pd.read_excel(source, sheet_name='csv-21111-03', na_filter=False)

    # Nur nach den Daten aus SH filtern
    data = data.loc[data['Bundesland'] == 'Schleswig-Holstein']

    # Zeilen löschen die nach Schulart differenzieren und nur die gesamten Zahlen behalten, die hier für uns wichtig sind.
    data = data.loc[data['Status'] == 'Insgesamt']

    # Ebenso Zeilen die nach Geschlecht differenzieren entfernen.
    data = data.loc[data['Geschlecht'] == 'Insgesamt']

    # Und auch nur die Daten die sich auf alle Bildungsbereicht innerhalb der Schule beziehen behalten.
    data = data.loc[data['Bildungsbereich'] == 'Alle Bildungsbereiche']

    # Zuletzt alle für uns unbrauchbare Spalten löschen, da nur Schule und Anzahl gebraucht wird.
    data = data.drop(columns=['Statistik_Code', 'Statistik_Label','Bundesland', 'Bildungsbereich', 'Schuljahr', 'Status', 'Geschlecht', 'Staatsangehoerigkeit' ])
```

6. Last 'Covid_Faelle_an_Schulen_nach_Landkreisen'



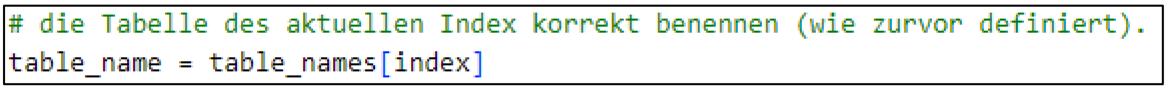
```
# CSV 'Covid_Faelle_Schultypen' und 'Covid_Faelle_an_Schulen_nach_Landkreisen' einlesen (besteht nur aus SH Daten und haben gleiche Strucktur).
else:
    data = pd.read_csv(source, sep=',', on_bad_lines='skip', skip_blank_lines=True)

    # Nur Schüler und Schülerinnen beachten und Lehrkräfte rausrechnen.
    data = data[data['Gruppe'] == 'Schülerinnen / Schüler']

    # Gruppen Spalten entfernen, weil sie nicht mehr gebraucht wird, weil nur noch Schüler darin sind
    data = data.drop(columns=['Gruppe'])
```

At the end of each of these 6 source processes I first check if there are dates with DATETIME types and convert them to prevent errors with the datatype.



```
# Konvertierung der Datums-Spalten (falls vorhanden),
# weil sie sonst nicht von der 'to_sql' funktion als DATETIME type gespeichert werden können ( -> type error).
if 'Impfdatum' in data.columns:
    data['Impfdatum'] = pd.to_datetime(data['Impfdatum'], errors='coerce')

if 'Datum' in data.columns:
    data['Datum'] = pd.to_datetime(data['Datum'], errors='coerce')
```

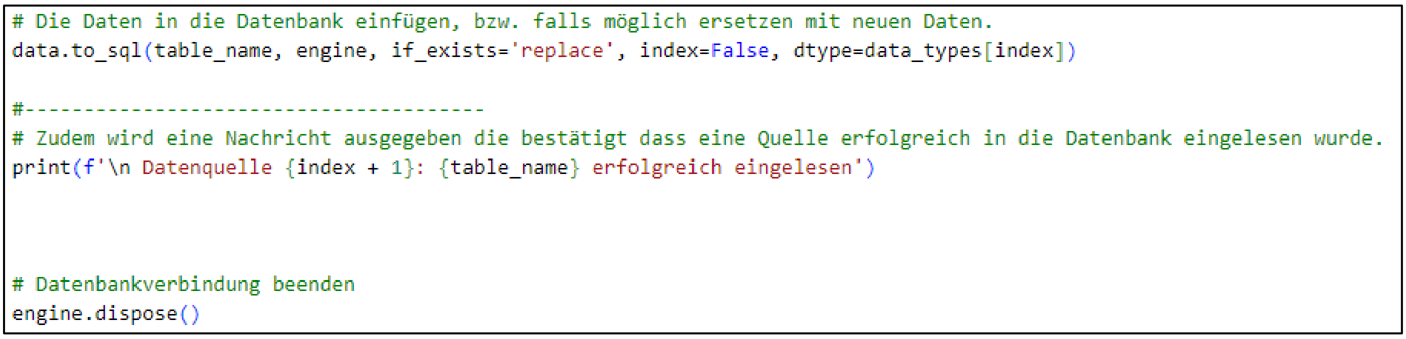
Then I assign the table names for each table in the sequence.



```
# die Tabelle des aktuellen Index korrekt benennen (wie zurvor definiert).
table_name = table_names[index]
```

And in the end, I load the tables in the SQLite Database we created earlier and print out a message that the table is loaded in the Database successfully.

Then, after all sources are through the loop and loaded, I end the connection with the Database and the pipeline is done



```
# Die Daten in die Datenbank einfügen, bzw. falls möglich ersetzen mit neuen Daten.
data.to_sql(table_name, engine, if_exists='replace', index=False, dtype=data_types[index])

#----------------------------------------
# Zudem wird eine Nachricht ausgegeben die bestätigt dass eine Quelle erfolgreich in die Datenbank eingelesen wurde.
print(f'\n Datenquelle {index + 1}: {table_name} erfolgreich eingelesen')



# Datenbankverbindung beenden
engine.dispose()
```

Results

Question 1

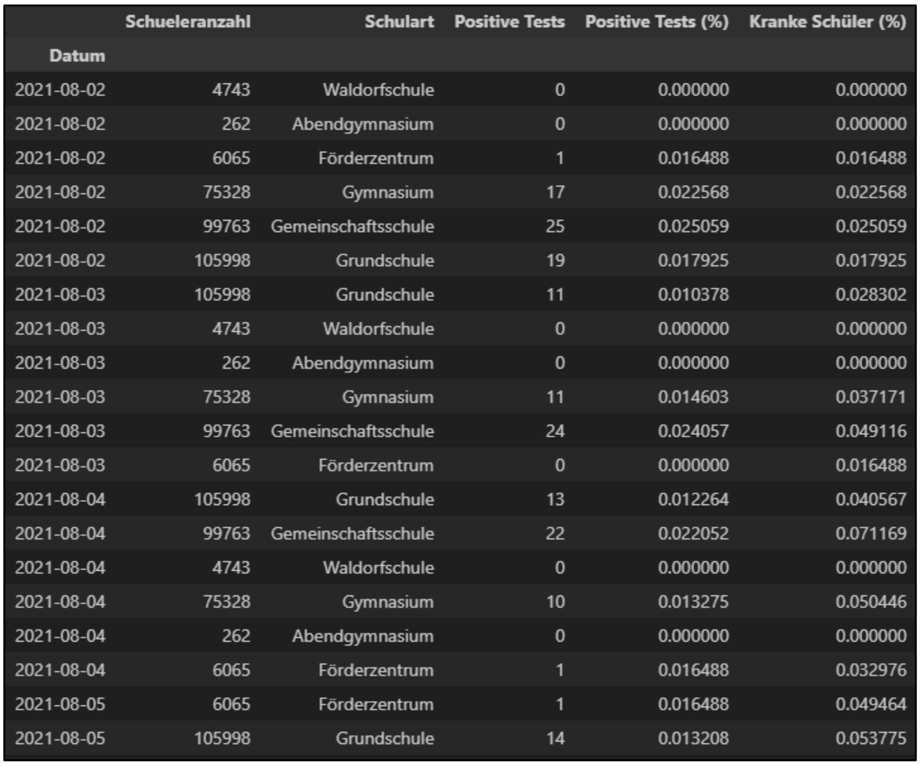
First, I will address the first question and try to find out whether the vaccination rate in SH has a different influence on different types of schools and if these types behave differently.

For that I separated the Pupil in the 6 biggest school types in SH:

- *Grundschule*
- *Waldorfschule*
- *Förderzentrum* (**possibly** another name for *Regionalschule*)
- *Gemeinschaftsschule*
- *Gymnasium*
- *Abendgymnasium*

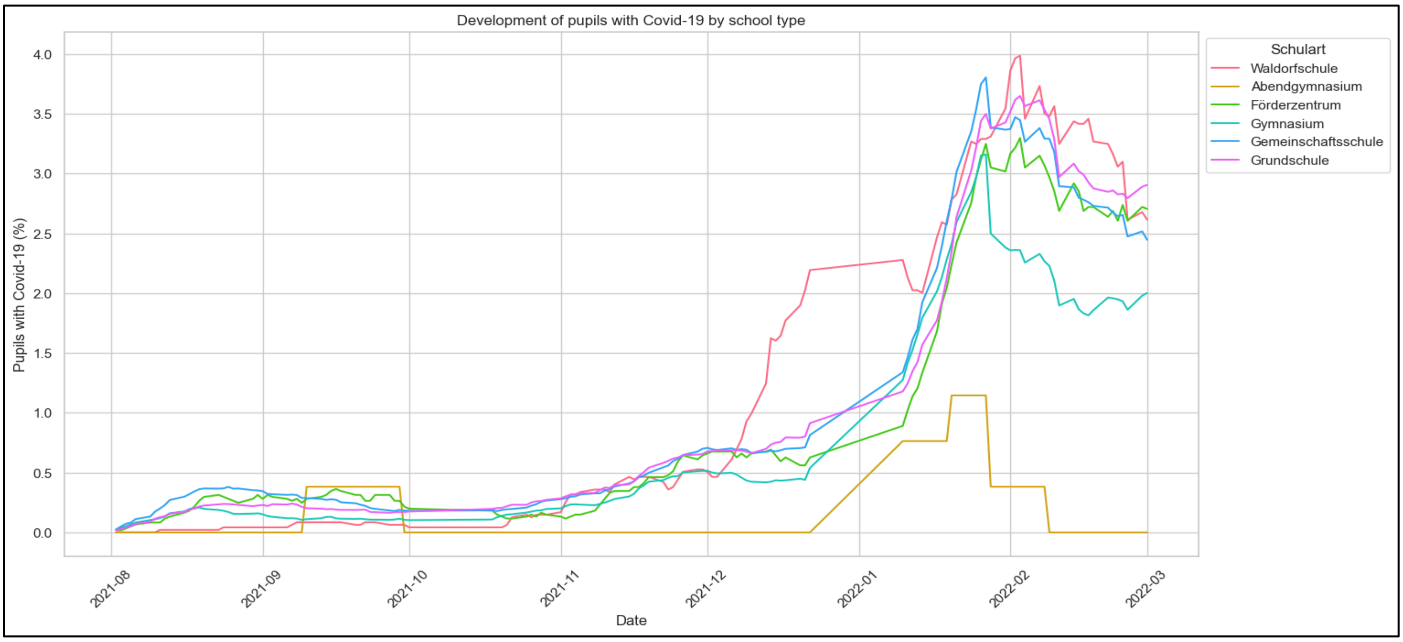
First, I merged the Sources that contain the total number of pupils in SH by School type (df2) and the one with the positive covid tests / cases per school type in SH (df3) to a new Data frame (merged_df2_3). Then I added a new column that shows the rate of positive tests in the school type at that day and then another one that shows the currently sick pupil from that school type (I use the pandas rolling function with 14 days of incubation period).

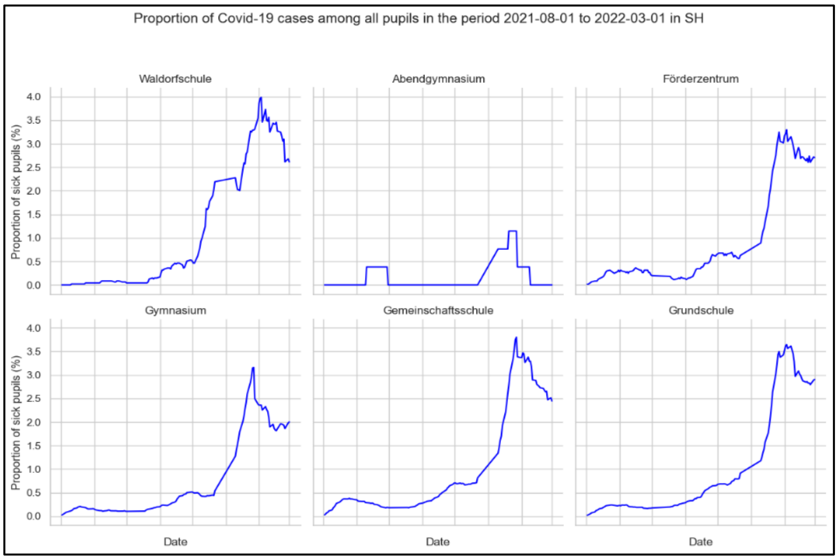
Following we see the first 20 rows of the Data frame:



	Schueleranzahl	Schulart	Positive Tests	Positive Tests (%)	Kranke Schüler (%)
Datum					
2021-08-02	4743	Waldorfschule	0	0.000000	0.000000
2021-08-02	262	Abendgymnasium	0	0.000000	0.000000
2021-08-02	6065	Förderzentrum	1	0.016488	0.016488
2021-08-02	75328	Gymnasium	17	0.022568	0.022568
2021-08-02	99763	Gemeinschaftsschule	25	0.025059	0.025059
2021-08-02	105998	Grundschule	19	0.017925	0.017925
2021-08-03	105998	Grundschule	11	0.010378	0.028302
2021-08-03	4743	Waldorfschule	0	0.000000	0.000000
2021-08-03	262	Abendgymnasium	0	0.000000	0.000000
2021-08-03	75328	Gymnasium	11	0.014603	0.037171
2021-08-03	99763	Gemeinschaftsschule	24	0.024057	0.049116
2021-08-03	6065	Förderzentrum	0	0.000000	0.016488
2021-08-04	105998	Grundschule	13	0.012264	0.040567
2021-08-04	99763	Gemeinschaftsschule	22	0.022052	0.071169
2021-08-04	4743	Waldorfschule	0	0.000000	0.000000
2021-08-04	75328	Gymnasium	10	0.013275	0.050446
2021-08-04	262	Abendgymnasium	0	0.000000	0.000000
2021-08-04	6065	Förderzentrum	1	0.016488	0.032976
2021-08-05	6065	Förderzentrum	1	0.016488	0.049464
2021-08-05	105998	Grundschule	14	0.013208	0.053775

I then first plotted, by using the *seaborn* and *matplotlib.pyplot* package, the development of the proportion of Covid-19 sick pupils per school type. We take a look at the Winter period of the school year 21/22. There were sadly no data from march on available in my source.





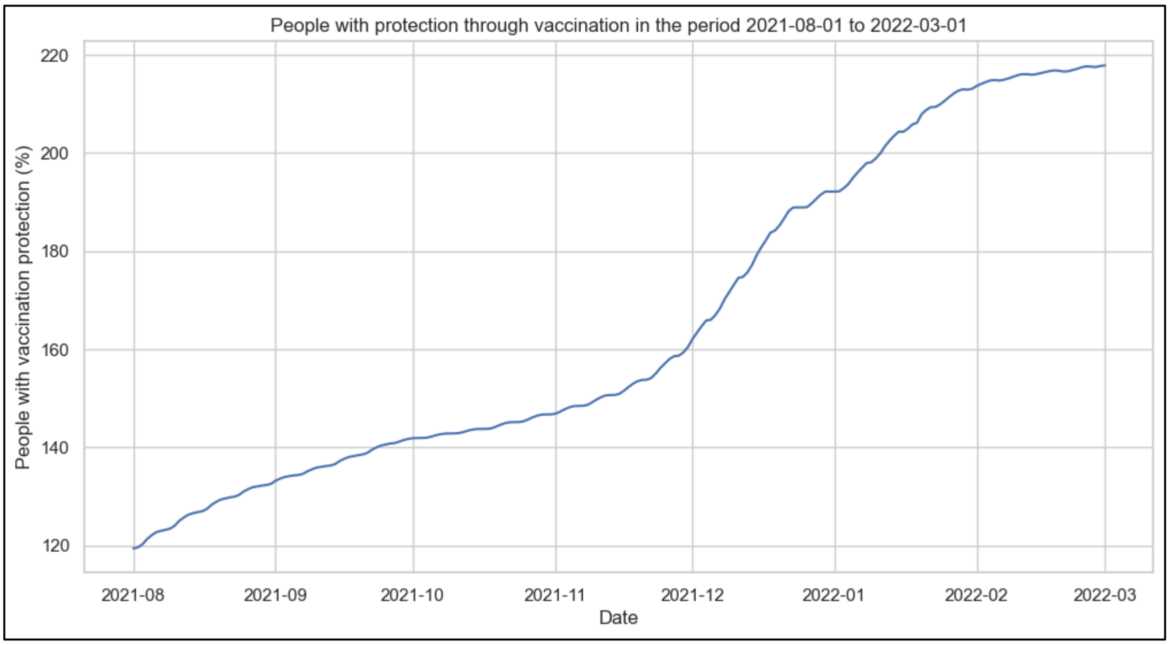
Here we can see at first that the data from 'Abendgymnasium' is not sufficient to build a meaningful analysis on.

Secondly, we see an overall at least to some extend a similar progression of all school types, except *Waldorfschule*, which recorded higher numbers of sick pupils early, from December 2021 on. However, it has adapted to the other school types in SH by mid-January. Nevertheless, it can also be seen that the *Waldorfschule* also had the highest proportion of sick pupils this school year.

Another thing to note is that *Gymnasium* schools have not had to fight with such high levels of illness, and the proportion they have had has also decreased a bit earlier by a larger proportion compared to other types of school.

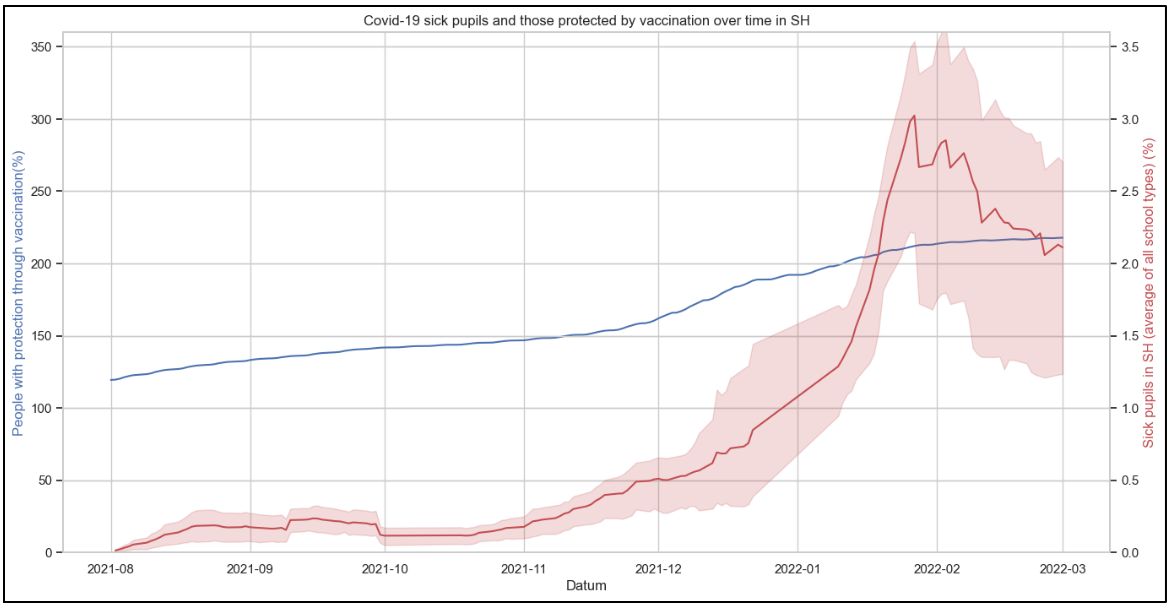
Now that we have gained an insight into the different trajectories of Covid-19 sick student proportions at the school types, let's look at the vaccination rate in Schleswig Holstein.

In order to plot the proportion of vaccination guns, I first had to use the .rolling function to commute the new vaccinations. According to the Rober Koch Institute, the time you can expect to be protected by vaccination is about a year and the population of SH is at around 2.897.000 at the time of this project.



We are seeing a relatively stable growth over time during the winter period of the sample school year in the proportion of the population with vaccination protection. So that around January 10, the 200% mark of the total population is reached *(more to that in the limitations part)*.

If we now combine the plots, we can see whether we can recognize any anomalies.



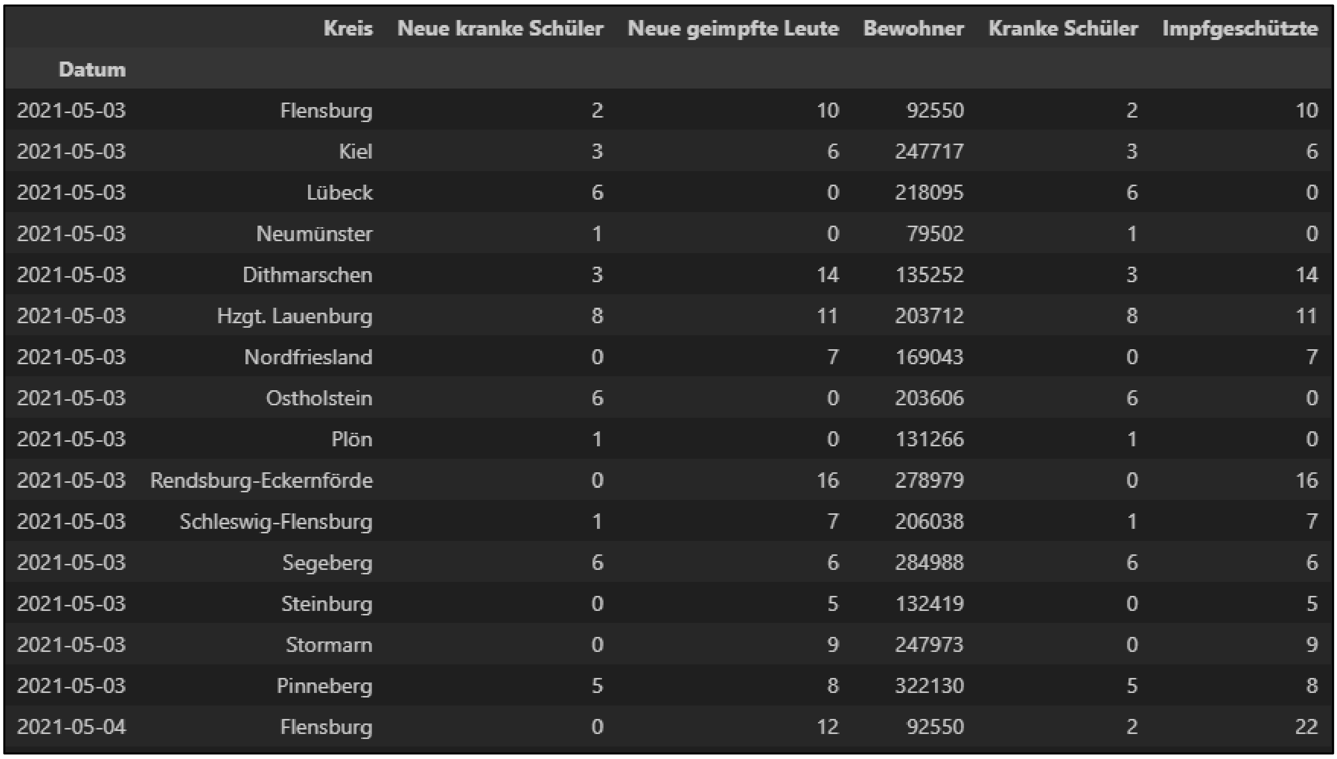
We can see when looking at both lines, that the vaccination one has a rather stable development, whereas the curve with the sick pupil comes exponential as a wave that reaches its peek at the end of January. We will see if we can interpret that in the last chapter.

Question 2

The second investigation I want to make is whether there are significant differences between the districts in SH.

To do this, we merge the data frames of the number of inhabitants of the counties in SH (df1), the vaccinations by counties (df5) and the number of pupils with Covid-19 by counties (df6). After that we add two more columns that keep track of the current total count of sick pupil and people with an active vaccination.

So that we get this table:

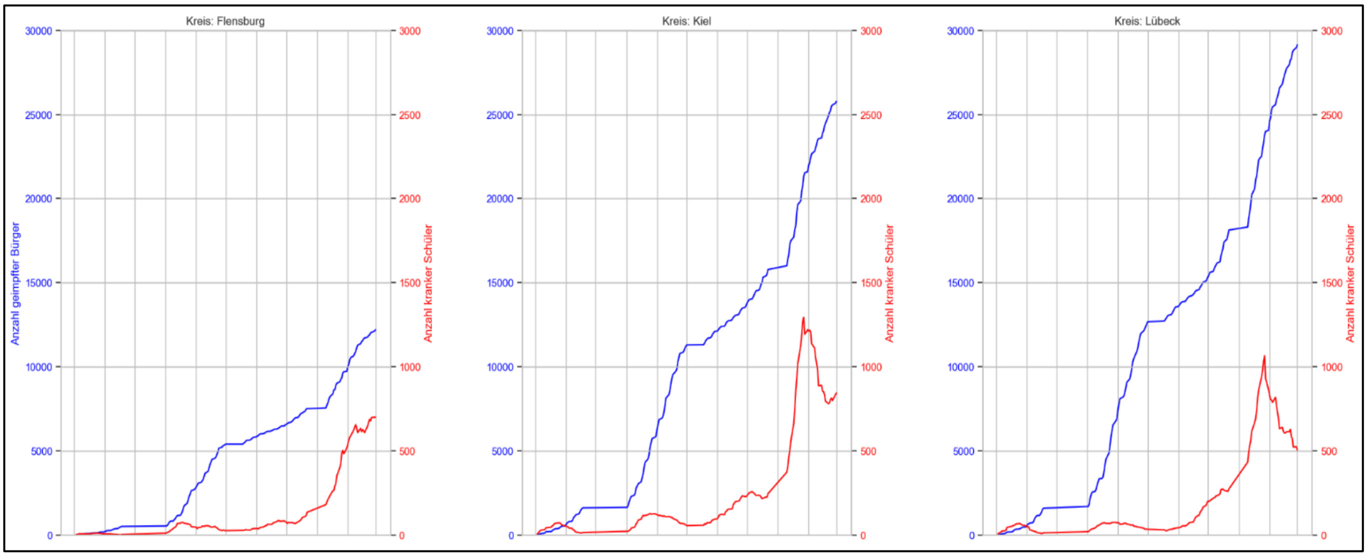


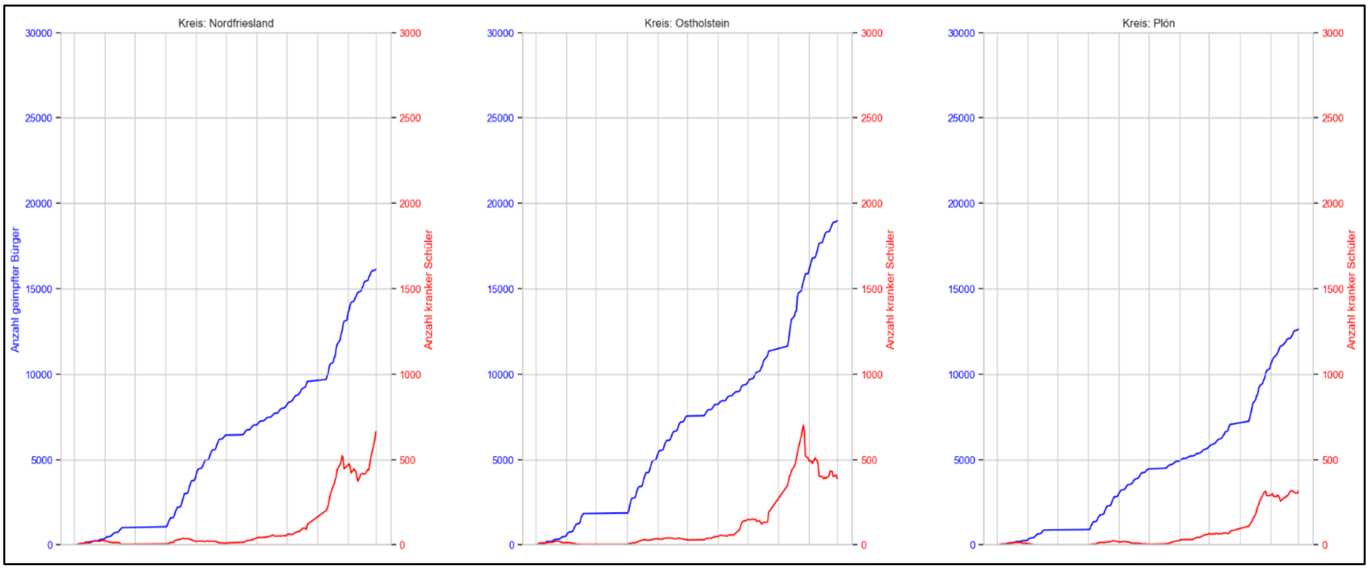
	Kreis	Neue kranke Schüler	Neue geimpfte Leute	Bewohner	Kranke Schüler	Impfgeschützte
Datum						
2021-05-03	Flensburg	2	10	92550	2	10
2021-05-03	Kiel	3	6	247717	3	6
2021-05-03	Lübeck	6	0	218095	6	0
2021-05-03	Neumünster	1	0	79502	1	0
2021-05-03	Dithmarschen	3	14	135252	3	14
2021-05-03	Hzgt. Lauenburg	8	11	203712	8	11
2021-05-03	Nordfriesland	0	7	169043	0	7
2021-05-03	Ostholstein	6	0	203606	6	0
2021-05-03	Plön	1	0	131266	1	0
2021-05-03	Rendsburg-Eckernförde	0	16	278979	0	16
2021-05-03	Schleswig-Flensburg	1	7	206038	1	7
2021-05-03	Segeberg	6	6	284988	6	6
2021-05-03	Steinburg	0	5	132419	0	5
2021-05-03	Stormarn	0	9	247973	0	9
2021-05-03	Pinneberg	5	8	322130	5	8
2021-05-04	Flensburg	0	12	92550	2	22

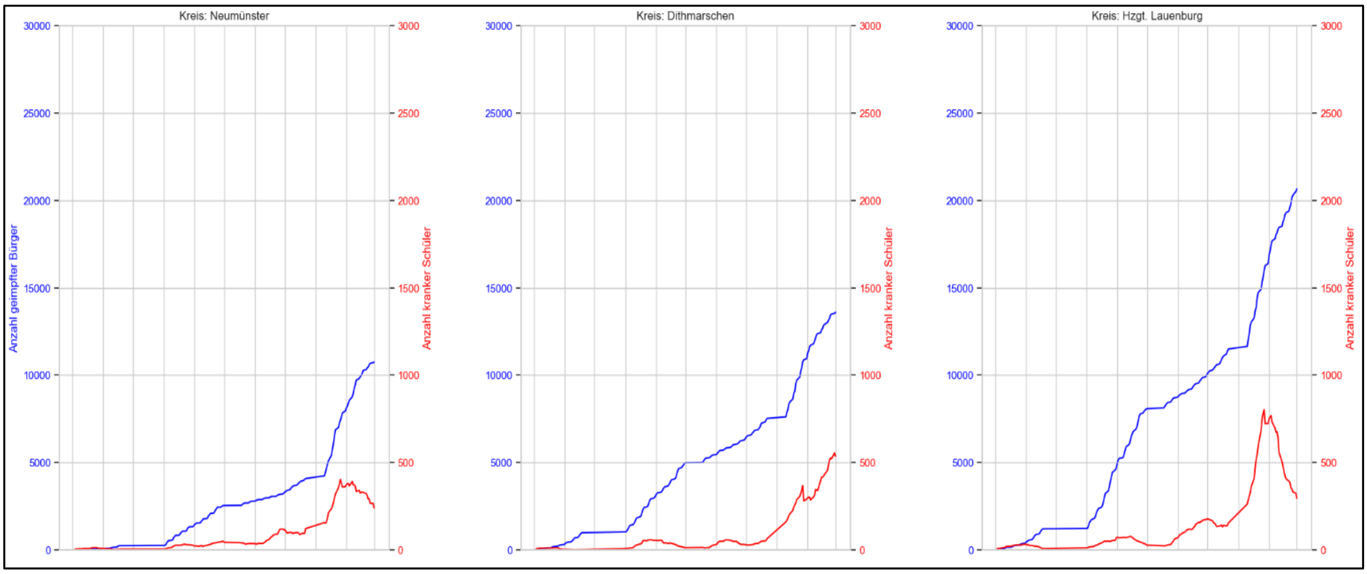
we can already see at first glance that the vaccination campaigns in SH probably did not start at the same time, we will see how things develop over time. The Time period we have data for are from the **2021-05-03** to the **2022-02-28**.

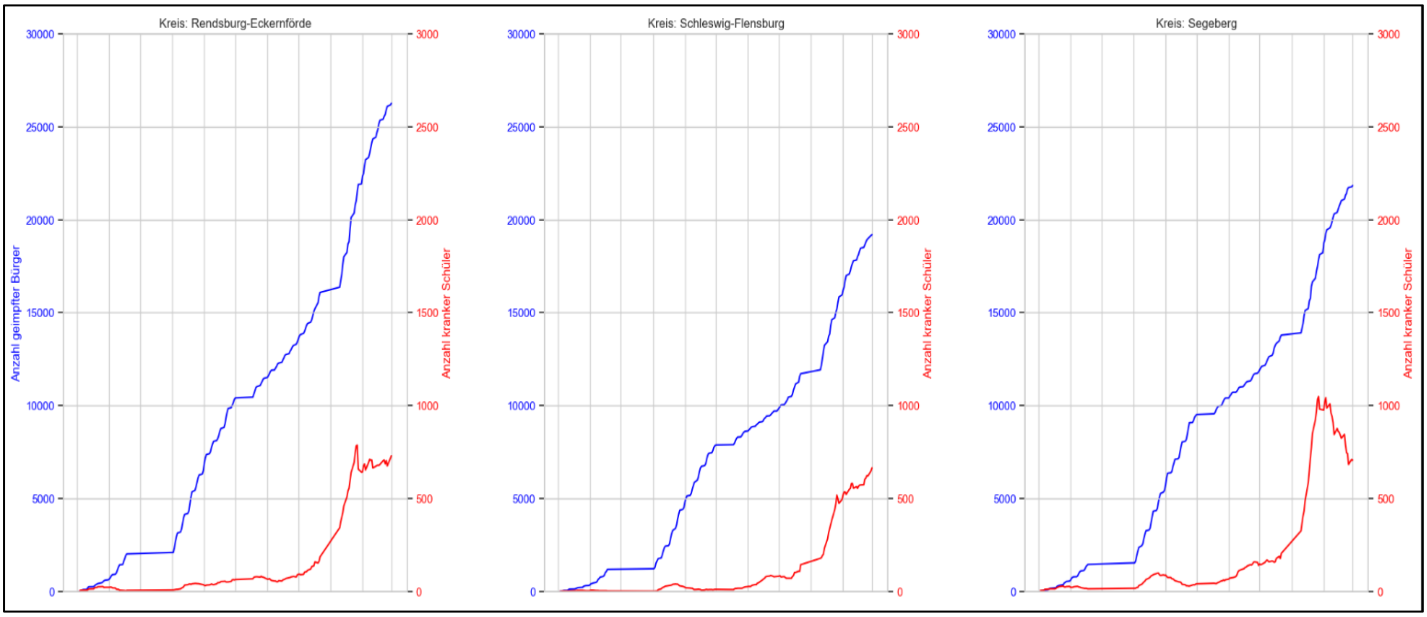
On the next pages we can see a large visualization of all counties and their trends in the increase in the number of cases among pupil and vaccination recipients.

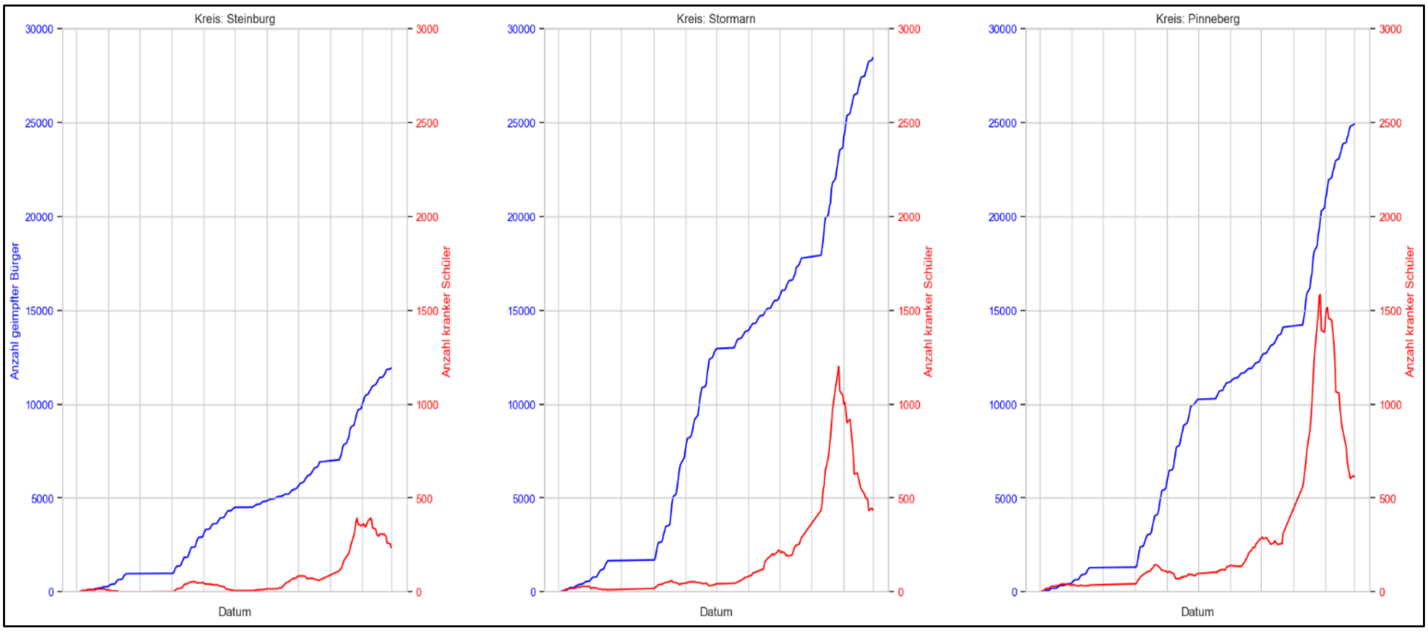
*(**Attention**: these first three are cities that are their own counties)*











What we can see is that is either a mistake in the Data or there were no vaccinations for nearly two months in June and July 2021.

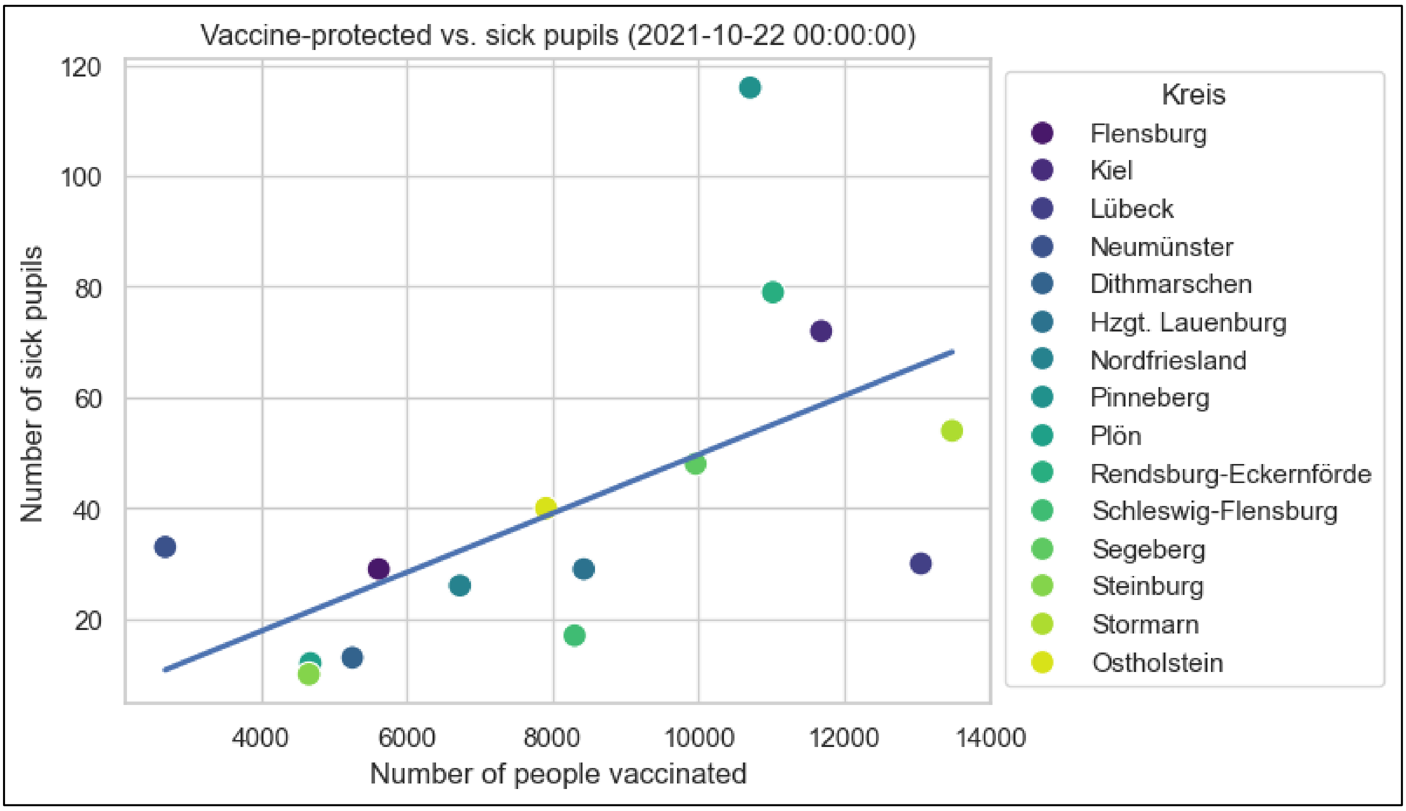
Secondly, we can see that the city Kiel and the countie Stormarn have the compared to the others very high numbers of sick pupil, the highest number by far has Pinneberg.

We can also see that the process is timewise pretty similar in all districts. Around the end of January 2022, the count of sick pupil is the highest.

What can also be quickly seen from the diagram is that the number of people with active vaccination is constantly increasing.

We of course also want do visualize if we can find a correlation or clusters in the data of sick pupil and vaccinated people in SH. For that we use both a seaborn Scatter and a Regression plot:

(The middle day of the dataset (22.10.2021) was used as the reference date)



Here we can clearly see that the number of sick pupil and the number of vaccinated people has a positive connection, I'm not saying correlation here on purpose because that will be thematized in the discussion part.

Conclusion

Discussion of the results and Context

We got some very interesting insights into the topic with the analysis of Vaccination and Covid-19 illness in Schleswig-Holstein. The first thing we obviously saw is that despite an increasing vaccination rate, pupils still fall ill during the winter period in a wave (in winter 21 we talk about the 3rd wave). This can therefore no more be prevented than, for example, a flu infection. Nevertheless, we can see differences in the progression of the third wave, with *Waldorfschule* and *Gymnasium* standing out in particular, even if there were no different rules for schools among each other.

We also have seen the regions of SH behaved differently in the 3rd covid-19 wave, some of them had a rapidly rising and falling rate, others only small numbers that also fell more slowly. In addition, there were also counties that experienced an increase but then remained stable at this number and did not fall again (or even continued to rise), despite rising vaccination numbers. This could indicate a different behavior of the population towards Covid.

In addition, a correlation was found between the number of sick pupils and the number of vaccinated pupils between the counties. However, it can also be assumed that this is due to the common positive connection to the number of inhabitants of the district. This should be investigated further with relative figures. I will discuss more about this in the next part 'Limitations'.

Limitations and potential improvements in the future

We first need to talk about the first question and the fact that the data about the important *Regionalschule* (the combined *Haupt-* und *Realschule* in SH), which probably makes up a large proportion of all pupils, is missing in the dataset. When looking in the data I can't imagine that *Förderzentrum* here is the same as *Regionalschule* because we work there with around 6000 pupils, where at the same time *Gymnasium* has over 75000 for example. So a possible improvement here would the inclusion of a dataset that also represents the *Regionalschulen*.

The next major thing that limited my work was the rather short period of time, both the data of the vaccinations and the number of sick pupils, are available in my sources. That can possibly be improved with the research for other possible sources.

Another thing is the problem, that people that are vaccinated twice or even more times during our timeframe and that I didn't had the time beside all the rest of the project to also improve the code in that way that we implement different timeframes or find a way to use the rolling method in a better way, but I know that it would definitely be possible by using the column '*Impfserie*' (that I deleted in the preprocessing part). in my opinion, however, it was sufficient this time, as I were more concerned with the shape of the curve anyway. But for the future that would be another point for improvement too.

When talking about the second part I first need to mention that the timeframe I was working with is again shorter than the pandemic itself and could therefore be expanded if the appropriate data is found.

Another thing is that I do not have a source for the total number of pupils in SH per countie. As a result, the analysis is based only on absolute figures, which means that there may be distortions in the analysis of the visualized data. That is definitely a point that should be improved when continuing this work.