Contents

Abstract

Covid-19 analysis on Germany is being done for a period of 11-months. We discussed the trend of New Cases and New Deaths. And discussed if the rules of Wearing masks have made any significant effect on the new cases and deaths. We also discussed the effects of relaxation in lockdown, opening labor markets on the cases. Finally, we compared the cases of all the states and checked if we found any relation(linear) between new cases and deaths.

About Data

The dataset contains two comma separated value(csv) files.

1. **cases-rki-by-state.csv:** It consists of the count of the daily covid-19 infected person across all the 16-states and total cases in Germany. The dataset covers data for 11 months i.e. from 2nd March 2020 to 8th Feb 2021. The data is cumulative and does not give the count for the particular day but gives the total count of cases till the given day. We refer this dataset as "*Infected Dataset".*

```
[3]: df_cases.head(2)
```

[3]:

	time_iso8601	DE-BB	DE-BE	DE-BW	DE-BY	DE-HB	DE-HE	DE-HH	DE-MV	DE-NI	DE-NW	DE-RP	DE-SH	DE-SL	DE-SN	DE-ST	DE-TH	sum_cases
0	2020-03-02T17:00:00+0000	2	0	30	34	2	9	2	0	4	116	2	4	3	1	0	0	209
1	2020-03-03T17:00:00+0000	2	6	54	40	4	13	5	2	10	145	3	5	3	1	0	1	294

2. **deaths-rki-by-state.csv:** It consists of the daily death counts similar to the *Infected Dataset.* Both the datasets are compatible to each other with same timeframe. We refer this dataset as "*Death Dataset".*

Initial Data Processing

In the first step, all the libraries to be used were imported and both the csv files were read into pandas dataframe. The date format is first changed to YYYY-MM-DD format. After that the date column is parsed from object to Datetime datatype. Then the info of the dataframe was checked.

```
df_cases.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 344 entries, 0 to 343
Data columns (total 18 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   time_iso8601  344 non-null    datetime64[ns]
 1   DE-BB         344 non-null    int64
 2   DE-BE         344 non-null    int64
 3   DE-BW         344 non-null    int64
```

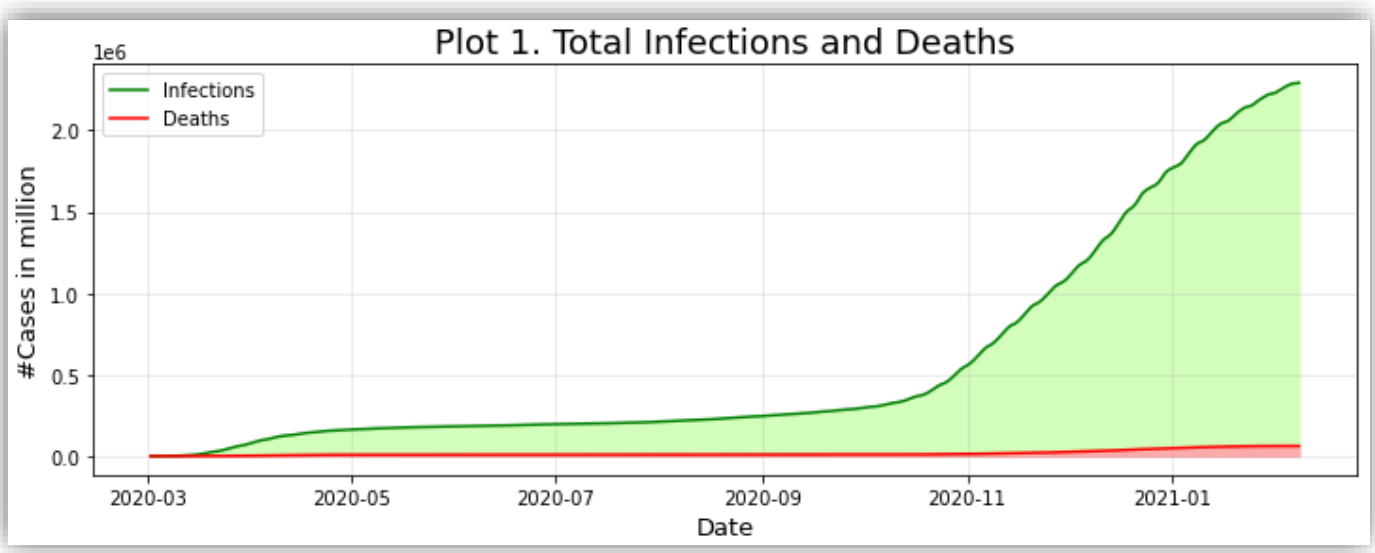
Code Snippet for import and data processing:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
df_cases = pd.read_csv('cases-rki-by-state.csv')
df_cases['time_iso8601'] = pd.to_datetime(df_cases['time_iso8601']).dt.date
df_cases['time_iso8601'] = pd.to_datetime(df_cases['time_iso8601'])
```

Similar pre-processing steps were performed for the Death Dataset.

Descriptive Analysis

After the data is pre-processed for any inconsistencies, the trend of covid-19 infections and deaths are checked.



Code Snippet for Plot 1:

```
fig = plt.figure(figsize=(12, 4))      # Defines the size of plot
x_values = df_cases['time_iso8601']
y_values = df_cases['sum_cases']
y_values_deaths = df_deaths['sum_deaths']
plt.plot(x_values, y_values, color = 'green', label='Infections')
plt.plot(x_values, y_values_deaths, color = 'red', label='Deaths')
plt.xlabel(xlabel= Date, fontsize=13)
plt.ylabel(ylabel='#Cases in million', fontsize=13)
plt.legend(loc='upper left')           # Places Legend on the top left corner
plt.fill_between(x_values, y_values, color='#D2FDBB') #Shaded area under plot
plt.fill_between(x_values, y_values_deaths, color='#fdaaaa')
plt.title('Plot 1. Total Infections and Deaths', fontsize=18)
plt.grid(alpha=0.3)                                  # Draws grids on the plot
plt.savefig("Reports/Infection_Death_Trend.jpeg")#Save plot in local drive
```

Data Manipulations

The Plot.1 was not capturing the daily trends. To analyse the distribution, more granular data was calculated(Daily data). To get number of daily new cases, we calculated a running difference across dataframe.
For example. *#new cases on 2nd September = #cases till 3rd September - #cases till 2nd September*

Code Snippet to calculate running difference:

```
df_cases_daily = df_cases.copy()
df_cases_daily.index = df_cases_daily.time_iso8601
df_cases_daily = df_cases_daily.diff() # calculate running difference
```

Similar operation is done on the Death Dataset.

```
[10]: df_cases_daily.tail(2)
```

[10]:

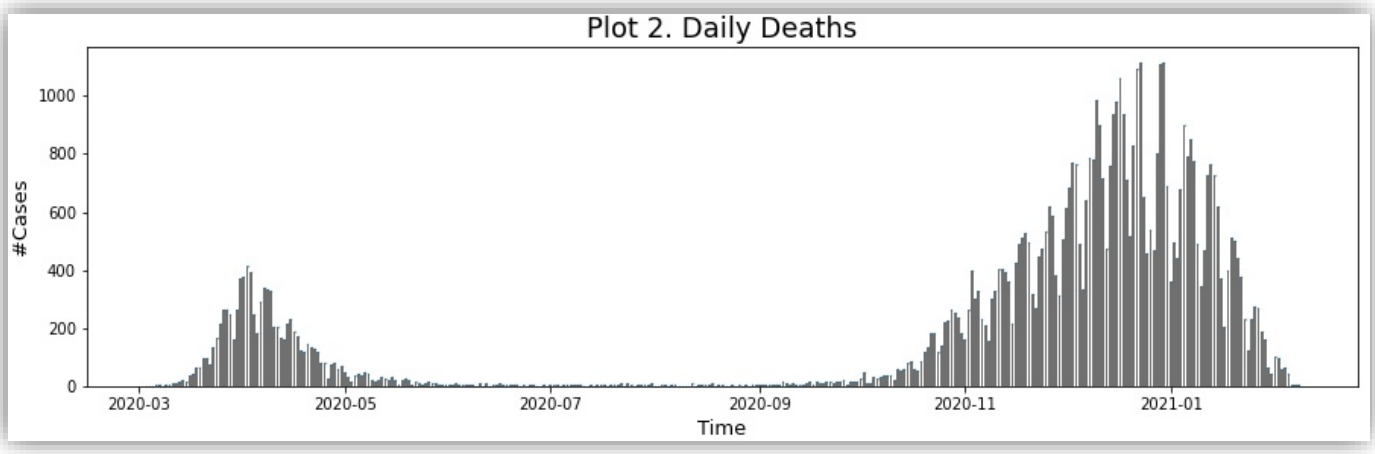
time_iso8601	time_iso8601	DE-BB	DE-BE	DE-BW	DE-BY	DE-HB	DE-HE	DE-HH	DE-MV	DE-NI	DE-NW	DE-RP	DE-SH	DE-SL	DE-SN	DE-ST	DE-TH	sum_cases
2021-02-07	1 days	105.0	26.0	389.0	579.0	45.0	180.0	121.0	81.0	244.0	573.0	170.0	106.0	39.0	136.0	136.0	91.0	3021.0
2021-02-08	1 days	119.0	233.0	294.0	299.0	11.0	180.0	35.0	61.0	105.0	186.0	161.0	135.0	18.0	25.0	56.0	156.0	2074.0

Daily cases and Daily deaths are plotted individually to analyze the trends.

Code Snippet for Plot 2:

```
fig = plt.figure(figsize=(12, 4))
```

```
x_values = df_deaths_daily.index
y_values = df_deaths_daily['sum_deaths']
plt.bar(x_values, y_values, alpha=0.6, color='#111111')
plt.xlabel(xlabel='Time', fontsize=13)
plt.ylabel(ylabel='#Cases', fontsize=13)
plt.title('Plot 2. Daily Deaths', fontsize=18)
```



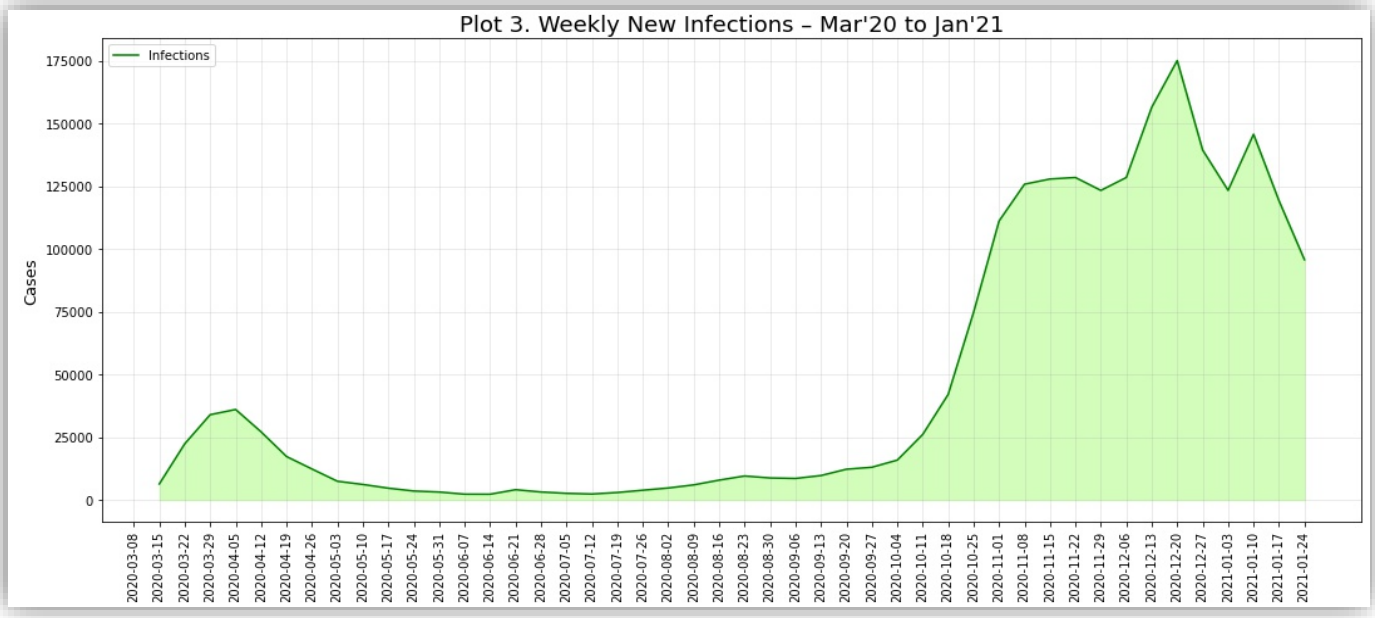
Similar trend is seen in daily infected cases. Code snippet used is similar as well. The trend seen on the daily data plot is very spikey. A better(smooth) trend could be seen if data is plotted by grouping data on weekly basis.

Code Snippet for grouping data Weekly:

```
df_deaths_daily['time_iso8601'] = df_deaths_daily.index
df_deaths_weekly=df_deaths_daily.groupby(df_deaths_daily['time_iso8601'].dt.s
trftime('%W'), sort=False).sum() # Grouping the data on weekly basis
```

Pattern of number of Confirmed Cases in Germany from the beginning of 2020 to January 2021

From plot 3, we can see the initial rising trend of cases. But from 5th May, daily cases were reduced significantly. Again after 12th July, the cases start to rise again slowly. And a cases suddenly shoot up after starting of October 2020 and reached the peak(175,000 cases/week) around the week of 12th Dec 2020.



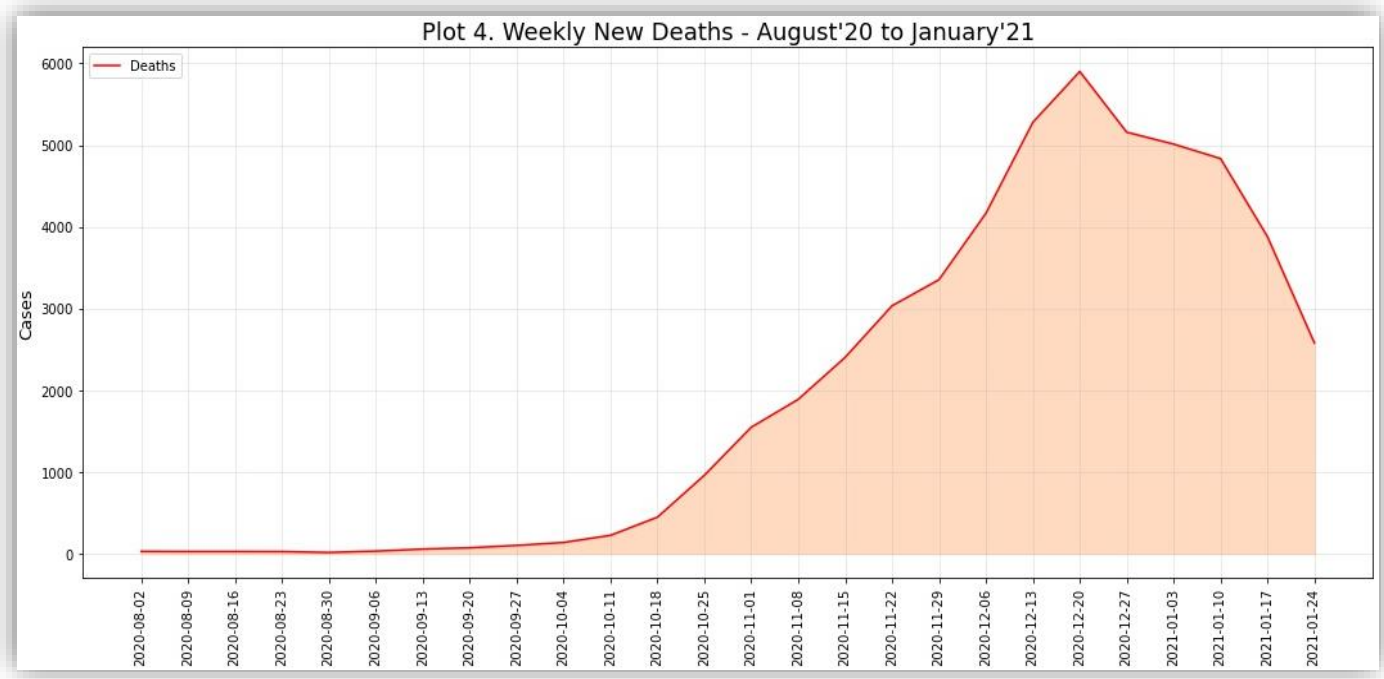
Code Snippet to for Plot 3:

```
df_cases_weekly_2 = df_cases_weekly.loc[ : '2021-01-31'] # truncated the
dataset to fetch data until January'21
x_values = df_cases_weekly_2['time_iso8601']
y_values = df_cases_weekly_2['sum_cases']
```

```
plt.plot(x_values, y_values, color = 'green', label='Infections')
plt.fill_between(x_values, y_values, color='#D2FDBB')
plt.xticks(df_cases_weekly_2['time_iso8601'], rotation = 90)
plt.ylabel(ylabel='Cases', fontsize=13)
plt.legend(loc='upper left')
plt.title("Plot 3. Weekly New Infections - Mar'20 to Jan'21", fontsize=18)
```

Rise of Number of Deaths from Aug'20 to Jan'21

A sharp rise in the number of daily deaths has been registered in the first week of October'20. Rising number of daily deaths continued till the mid of December'20 and after that it started to fall.

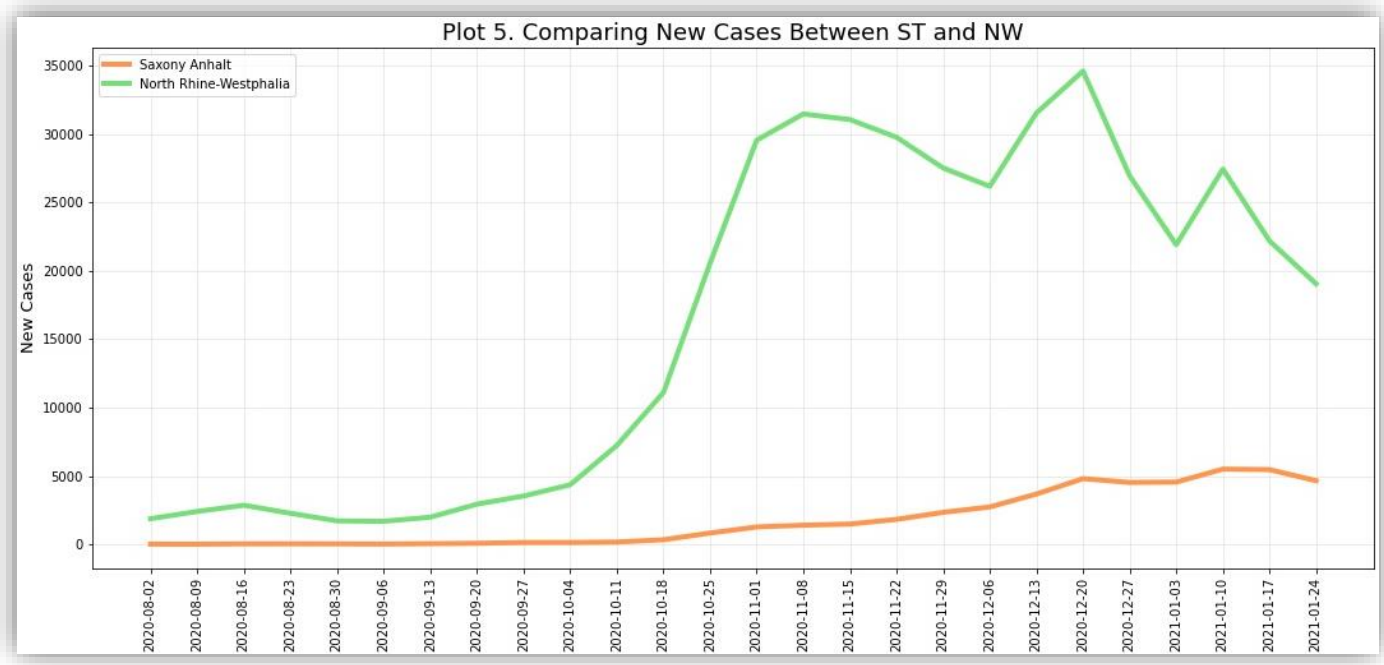


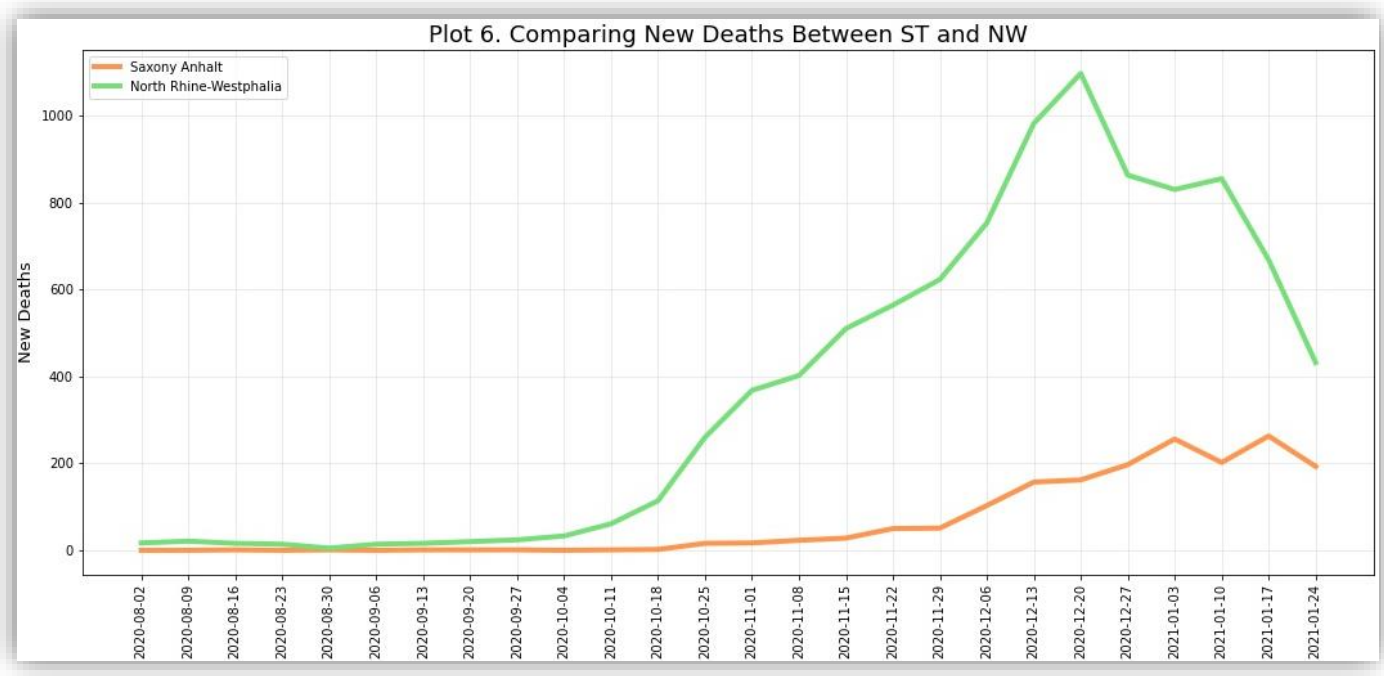
Code Snippet for Plot 4:

```
fig = plt.figure(figsize=(18, 6))
df_deaths_weekly_reduced = df_deaths_weekly.loc['2020-08-01':'2021-01-31']
x_values = df_deaths_weekly_reduced['time_iso8601']
y_values = df_deaths_weekly_reduced['sum_deaths']
plt.plot(x_values, y_values, color = 'red', label='Deaths')
plt.xticks(df_deaths_weekly_reduced['time_iso8601'], rotation = 90)
plt.fill_between(x_values, y_values, color='#ffdac1')
plt.ylabel(ylabel='Cases', fontsize=13)
plt.legend(loc='upper left')
plt.title("Plot 4. Weekly New Deaths - August'20 to January'21", fontsize=18)
plt.grid(alpha=0.3)
```

Sachsen-Anhalt vs North Rhine-Westphalia

One of the major hotspots in Germany was North Rhine-Westphalia. When compared with Saxony-Anhalt for new cases(Plot 5), it can be seen that the trend is quite similar for both NW and ST. But there is a huge difference in the number of daily cases in the given time. In the starting days(Feb'20 to Sept'20), the difference was not very large. But after October'20, the rate of new cases has been increased by a substantial margin in NW.
The exact similar trend can be seen in new Death cases between both the states(Plot 6).



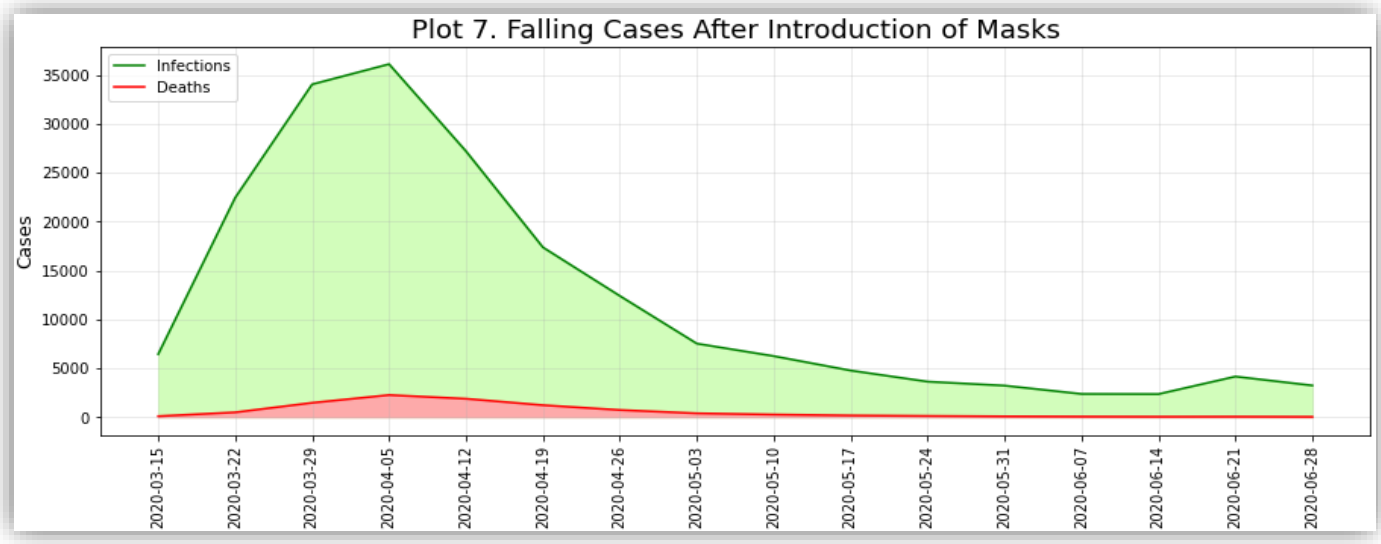


Code Snippet for Plot 6:

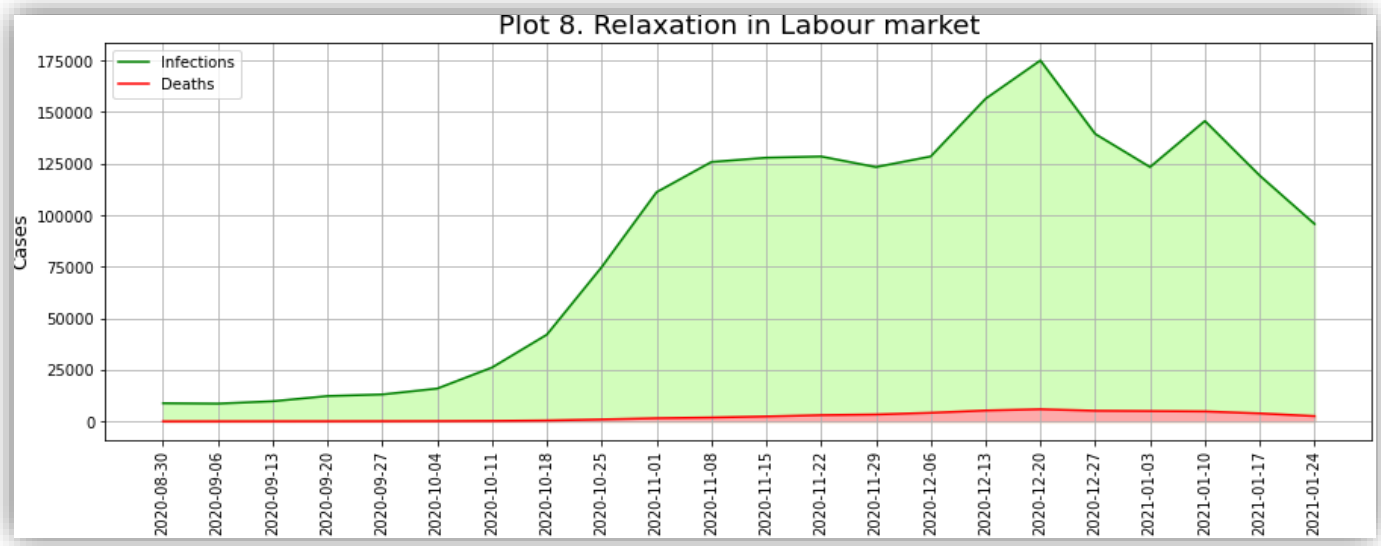
```
fig = plt.figure(figsize=(18, 7.5))
x_values = df_deaths_weekly_reduced['time_iso8601']
y_values_ST = df_deaths_weekly_reduced['DE-ST']
y_values_NRW = df_deaths_weekly_reduced['DE-NW']
plt.plot(x_values, y_values_ST, color = '#ff964f', label='Saxony Anhalt',
linewidth = 4)
plt.plot(x_values, y_values_NRW, color = '#77df79', label='North Rhine-
Westphalia', linewidth = 4)
plt.xticks(df_deaths_weekly_reduced['time_iso8601'], rotation = 90)
plt.ylabel(ylabel='Cases', fontsize=13)
plt.legend(loc='upper left')
plt.title('Plot 6. Comparing New Deaths Between ST and NW', fontsize=18)
plt.grid(alpha=0.3)
```

Effect of Introduction of Mask vs Effect of Relaxation in Labour Market

States in Germany has started to make masks mandatory in public places from 1st April 2020. Jana was the first city to implement it[1][2]. From 1st April to 10th April most of the states had made wearing masks mandatory in public places. Because of which a sharp dip can be seen from April till June(Plot 7).



After August, it was noticed that the unemployment became the challenge for citizens and economy started to fall. States have given permission to relax the lockdown for labour market on September[3]. This relaxation has increased human interaction thus making a sharp rise in the number of infections and deaths from Oct'21(Plot 8).



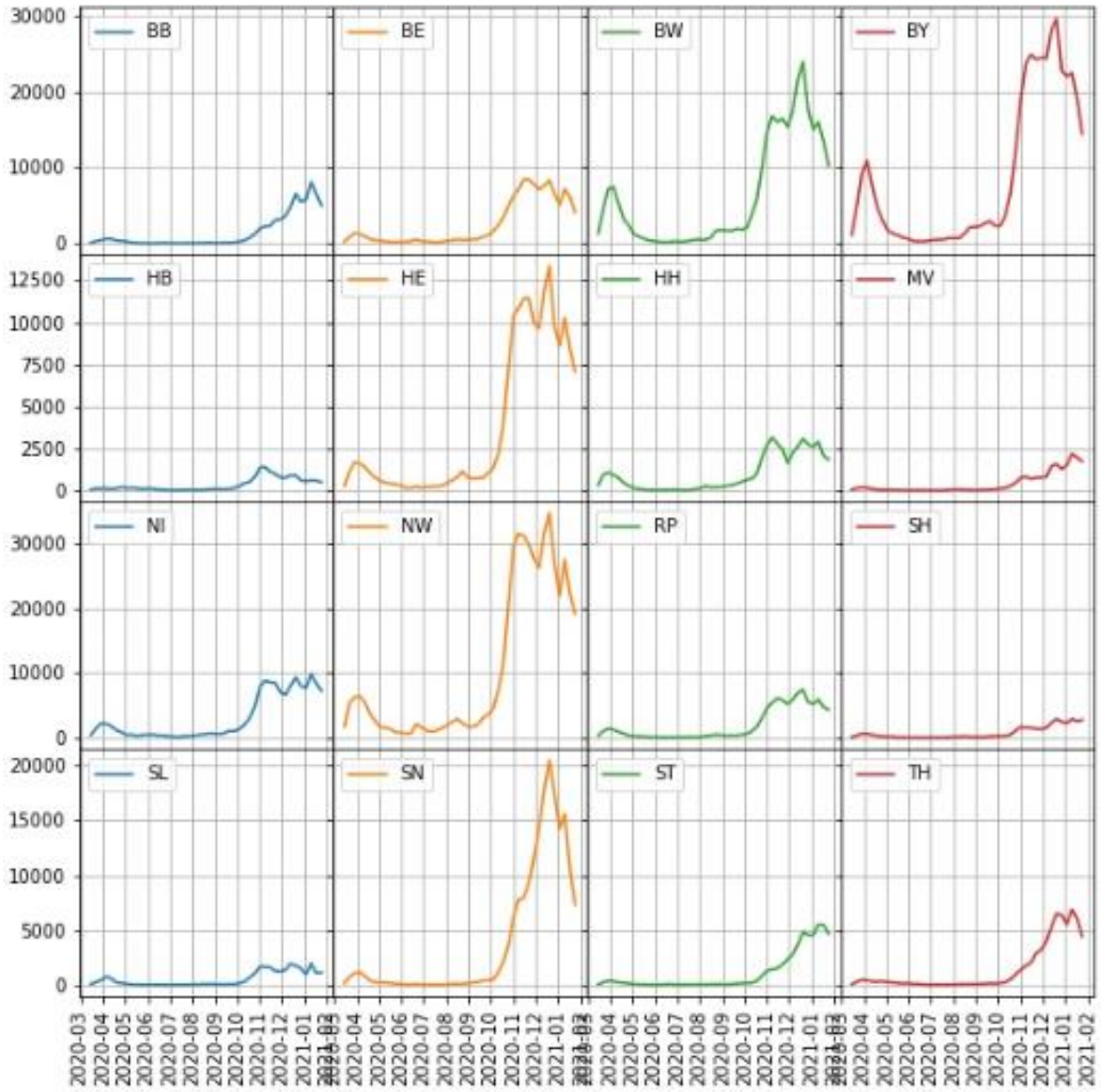
Code Snippet for Plot 7:

```
df_cases_weekly_Q4_2 = df_cases_weekly_2.loc[:'2020-06-30']
df_deaths_weekly_Q4_2 = df_deaths_weekly.loc[:'2020-06-30']
x_values = df_cases_weekly_Q4_2['time_iso8601']
y_values = df_cases_weekly_Q4_2['sum_cases']
y_values_deaths = df_deaths_weekly_Q4_2['sum_deaths']
plt.plot(x_values, y_values, color = 'green', label='Infections')
plt.plot(x_values, y_values_deaths, color = 'red', label='Deaths')
plt.xticks(df_cases_weekly_Q4_2['time_iso8601'], rotation = 90)
plt.ylabel(ylabel='Cases', fontsize=13)
plt.legend(loc='upper left')
plt.fill_between(x_values, y_values, color='#D2FDBB')
plt.fill_between(x_values, y_values_deaths, color='#fdaaaa')
plt.title('Plot 7. Falling Cases After Introduction of Masks', fontsize=18)
plt.grid(alpha=0.3)
```

Q5. Perhaps you can discuss any other interesting observation from these data with a plot or a table.

From plot 9, it can be seen that the most infected states are Baden-Wuttemberg, Bayern, North Rhine-Westphalia, and Saxony with peak cases more than 20,000/week. And the least effected states are Saarland, Schleswig-Holstein, Bremen, and Mecklenburg-Western Pomerania with peak cases less than 2,500/week. The rising and falling period of all the states are same.

Plot 9. Comparison of New Cases in all the States
New Cases v/s Time(Weekly)



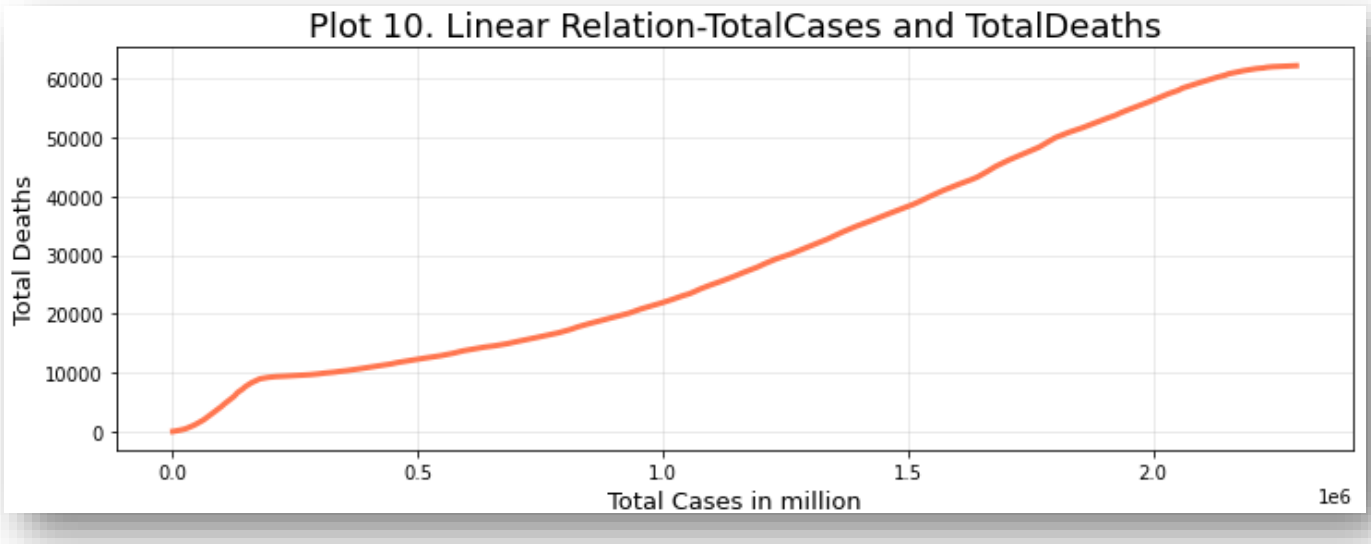
Code Snippet for Plot 9:

```
x_values = df_cases_weekly_2['time_iso8601']      # Setting X values
y_values_BB = df_cases_weekly_2['DE-BB']          # Setting Y values
y_values_BE = df_cases_weekly_2['DE-BE']
y_values_BW = df_cases_weekly_2['DE-BW']
y_values_BY = df_cases_weekly_2['DE-BY']
y_values_HB = df_cases_weekly_2['DE-HB']
y_values_HE = df_cases_weekly_2['DE-HE']
y_values_HH = df_cases_weekly_2['DE-HH']
y_values_MV = df_cases_weekly_2['DE-MV']
y_values_NI = df_cases_weekly_2['DE-NI']
```

```
y_values_NW = df_cases_weekly_2['DE-NW']
y_values_RP = df_cases_weekly_2['DE-RP']
y_values_SH = df_cases_weekly_2['DE-SH']
y_values_SL = df_cases_weekly_2['DE-SL']
y_values_SN = df_cases_weekly_2['DE-SN']
y_values_ST = df_cases_weekly_2['DE-ST']
y_values_TH = df_cases_weekly_2['DE-TH']

fig, axs = plt.subplots(4, 4, sharex='col', sharey='row',
gridspec_kw={'hspace': 0, 'wspace': 0}, figsize=(10,10))
(ax1, ax2, ax3, ax4), (ax5, ax6, ax7, ax8), (ax9, ax10, ax11, ax12 ), (ax13,
ax14, ax15, ax16) = axs
fig.suptitle('Comparison of New Cases in all the States\n New Cases v/s
Time(Weekly)')
ax1.plot(x_values, y_values_BB,  label='BB')
ax2.plot(x_values, y_values_BE, 'tab:orange', label='BE')
ax3.plot(x_values, y_values_BW, 'tab:green', label='BW')
ax4.plot(x_values, y_values_BY, 'tab:red', label='BY')
ax5.plot(x_values, y_values_HB,  label='HB')
ax6.plot(x_values, y_values_HE, 'tab:orange', label='HE')
ax7.plot(x_values, y_values_HH, 'tab:green', label='HH')
ax8.plot(x_values, y_values_MV, 'tab:red', label='MV')
ax9.plot(x_values, y_values_NI,  label='NI')
ax10.plot(x_values, y_values_NW, 'tab:orange', label='NW')
ax11.plot(x_values, y_values_RP, 'tab:green', label='RP')
ax12.plot(x_values, y_values_SH, 'tab:red', label='SH')
ax13.plot(x_values, y_values_SL,  label='SL')
ax14.plot(x_values, y_values_SN, 'tab:orange', label='SN')
ax15.plot(x_values, y_values_ST, 'tab:green', label='ST')
ax16.plot(x_values, y_values_TH, 'tab:red', label='TH')
for ax in axs.flat:          # Loop to set label, legend grid and axis rotation
    ax.label_outer()         # to all the subplots
    ax.legend(loc='upper left')
    ax.grid()
    plt.setp(ax.xaxis.get_majorticklabels(), rotation=90)
```

Another interesting finding has been seen when a line graph is plotted between the Total Deaths vs Total New Cases. Number of deaths has a linear relationship with the number of cases(Plot 10).



Code Snippet for Plot 10:

```
x_values_cases = df_cases['sum_cases']
y_values_deaths = df_deaths['sum_deaths']
plt.plot(x_values_cases, y_values_deaths, color = '#ff764f', linewidth=3)
plt.title('Plot 10. Linear Relation-TotalCases and TotalDeaths', fontsize=18)
plt.xlabel(xlabel='Total Cases in million', fontsize=13)
plt.ylabel(ylabel='Total Deaths', fontsize=13)
plt.grid(alpha=0.3)
```

Sources

[1] https://www.theguardian.com/world/2020/mar/31/calls-grow-for-germany-wide-use-of-face-masks-covid-19
[2] https://www.pnas.org/content/117/51/32293
[3] https://www.deutschland.de/en/news/coronavirus-in-germany-informations
[4] Pandas developer doc
[5] https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.groupby.html
[6] https://matplotlib.org/stable/
[7] https://matplotlib.org/stable/gallery/subplots_axes_and_figures/subplots_demo.html
[8] https://stackoverflow.com/questions