

COVID-19 Auswertung

DIVI und WHO Daten

```
In [23]: # import libraries
import pandas as pd
from matplotlib import pyplot as plt

from datetime import datetime #performance
import os
now = datetime.now() #performance
```

```
In [24]: # Webabruf - CSV einlesen
data = pd.read_csv("https://covid19.who.int/WHO-COVID-19-global-dat
a.csv")
#gleitender Mittelwert der cases und deaths, 7 Tage
data['MA'] = data['New_cases'].rolling(window=7,min_periods=1).mean
()
data['MA_deaths'] = data['New_deaths'].rolling(window=7, min_periods
=1).mean()
```

```
In [35]: # Date_reported in eine Datumsspalte umwandeln
data['Date_reported'] = pd.to_datetime(data.Date_reported)

#Timestamp definieren YYYY-MM-DD
ts = pd.to_datetime('2020-9-01')

#dataframe reduzieren
data = data.loc[data.Date_reported >= ts, :]
data.head(2)
```

Out[35]:

	Date_reported	Country_code	Country	WHO_region	New_cases	Cumulative_cases
242	2020-09-01	AF	Afghanistan	EMRO	34	38282
243	2020-09-02	AF	Afghanistan	EMRO	47	38329

```
In [26]: # Länder definieren
ger = data[data.Country == 'Germany']
fr = data[data.Country == 'France']
at = data[data.Country == 'Austria']
pl = data[data.Country == 'Poland']
cz = data[data.Country == 'Czechia']
ch = data[data.Country == 'Switzerland']
it = data[data.Country == 'Italy'] # 22.11.2020
es = data[data.Country == 'Spain'] # 22.11.2020
gb = data[data.Country == 'The United Kingdom'] # 22.11.2020
se = data[data.Country == 'Sweden'] # 22.11.2020
```

In [27]: # Variablen Definition für Plots

```
# Linien Stärke
lws = 3
lwb = 7
# Bezugsschriftgröße
size = 28
```

In [28]: # Größe im 16:9 format und mit Umrechnungsfaktor 1.2 (durch Test ermittelt) für PowerPoint angepasst
plt.figure(figsize=(16/1.2, 9/1.2))

```
plt.style.use('seaborn')
plt.grid(True)
```

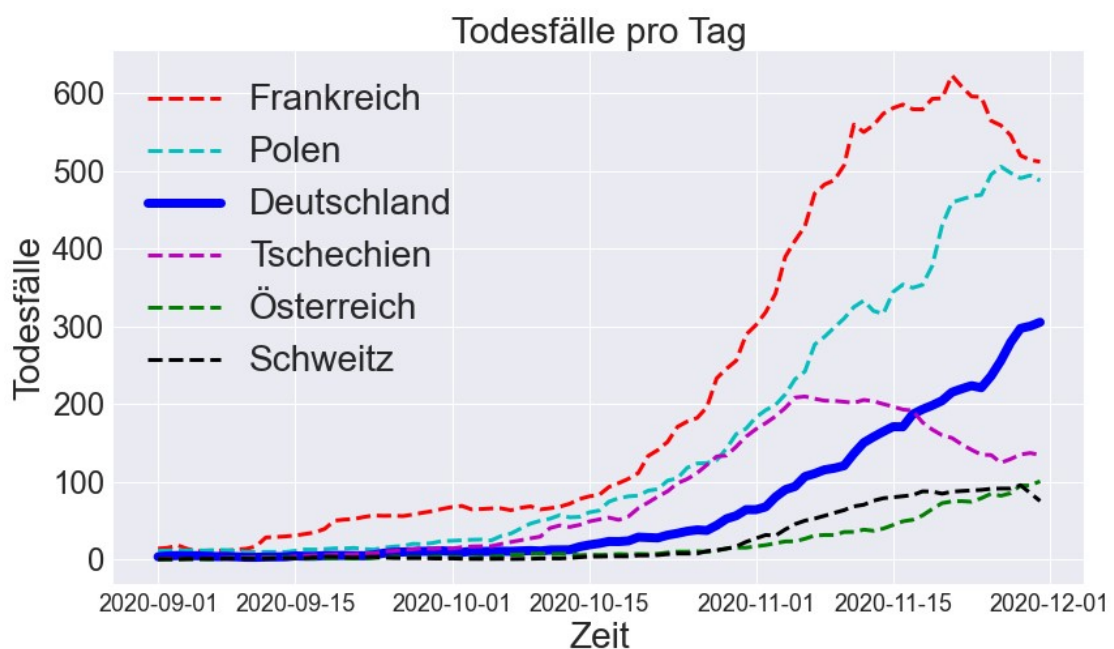
```
plt.plot(fr.Date_reported, fr['MA_deaths'], 'r--', linewidth = lws)
plt.plot(pl.Date_reported, pl['MA_deaths'], 'c--', linewidth = lws)
plt.plot(ger.Date_reported, ger['MA_deaths'], 'b-', linewidth = lwb)
plt.plot(cz.Date_reported, cz['MA_deaths'], 'm--', linewidth = lws)
plt.plot(at.Date_reported, at['MA_deaths'], 'g--', linewidth = lws)
plt.plot(ch.Date_reported, ch['MA_deaths'], 'k--', linewidth = lws)
```

```
plt.legend(['Frankreich', 'Polen', 'Deutschland', 'Tschechien', 'Österreich', 'Schweitz'], fontsize=size)
```

```
# Schriftgrößen x und y achsenwerte
plt.xticks(fontsize=size - 10)
plt.yticks(fontsize=size - 4)
```

```
# plt.xticks([])
plt.ylabel('Todesfälle', fontsize=size)
plt.title('Todesfälle pro Tag', fontsize=size)
plt.xlabel('Zeit', fontsize=size)
```

```
# Diagramm als Bild exportieren und Auflösung definieren
plt.savefig('plot_deaths.jpg', dpi = 150)
```



```
In [29]: # Größe im 16:9 format und mit Umrechnungsfaktor 1.2 (durch Test erm
         ittelt) für PowerPoint angepasst
         plt.figure(figsize=(16/1.2, 9/1.2))

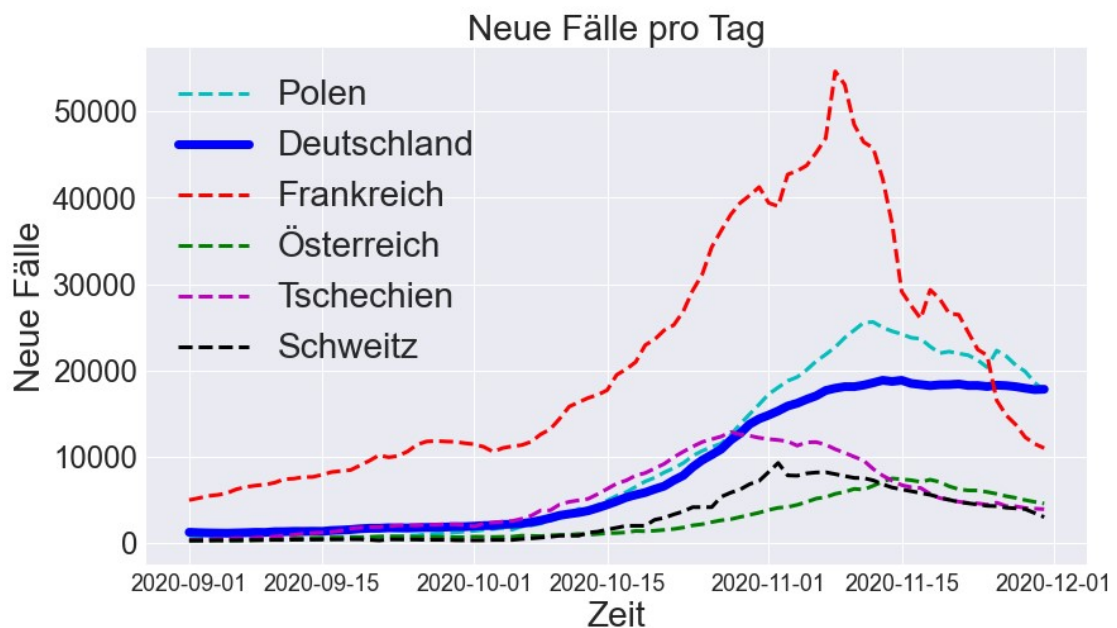
         plt.plot(pl.Date_reported, pl['MA'], 'c--', linewidth = lws)
         plt.plot(ger.Date_reported, ger['MA'], 'b-', linewidth = lwb)
         plt.plot(fr.Date_reported, fr['MA'], 'r--', linewidth = lws)
         plt.plot(at.Date_reported, at['MA'], 'g--', linewidth = lws)
         plt.plot(cz.Date_reported, cz['MA'], 'm--', linewidth = lws)
         plt.plot(ch.Date_reported, ch['MA'], 'k--', linewidth = lws)

         plt.legend(['Polen', 'Deutschland', 'Frankreich', 'Österreich', 'Tsc
         hechien', 'Schweitz'], fontsize=size)

         # Schriftgrößen x und y achsenwerte
         plt.xticks(fontsize=size - 10)
         plt.yticks(fontsize=size - 4)

         # plt.xticks([])
         plt.ylabel('Neue Fälle', fontsize=size)
         plt.title('Neue Fälle pro Tag', fontsize=size)
         plt.xlabel('Zeit', fontsize=size)

         # Diagramm als Bild exporieren und Auflösung definieren
         plt.savefig('plot_cases.jpg', dpi = 150)
```



In []:

```
In [34]: # CSV einlesen
         df = pd.read_csv("data-WvhXR.csv")
         df.head(2)
```

Out [34]:

	date	COVID-19-Fälle
0	2020-03-20T12:15:00+01:00	200
1	2020-03-21T12:15:00+01:00	308

```

In [31]: # Date_reported in eine Datumsspalte umwandeln
df['date'] = pd.to_datetime(df.date)

In [32]: # Größe im 16:9 format und mit Umrechnungsfaktor 1.2 (durch Test erm
         ittelt) für PowerPoint angepasst
plt.figure(figsize=(16/1.2, 9/1.2))

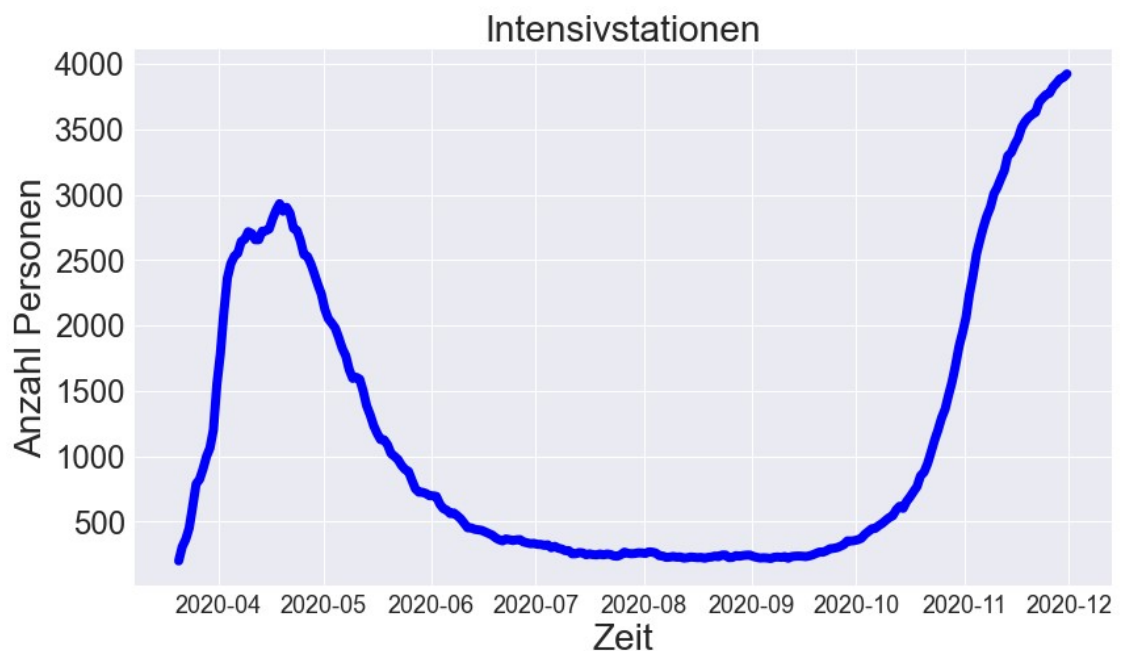
plt.style.use('seaborn')
plt.grid(True)
plt.plot(df.date, df['COVID-19-Fälle'], 'b-', linewidth = lwb)
# plt.legend(["Intensivstationen"], fontsize=size)

# Schriftgrößen x und y achsenwerte
plt.xticks(fontsize=size - 10)
plt.yticks(fontsize=size - 4)

# plt.xticks([])
plt.ylabel('Anzahl Personen', fontsize=size)
plt.title('Intensivstationen', fontsize=size)
plt.xlabel('Zeit', fontsize=size)

# Diagramm als Bild exportieren und Auflösung definieren
plt.savefig('plot_intensiv.jpg', dpi = 150)

```



```

In [33]: #performance
pc = os.environ['COMPUTERNAME']
now2 = datetime.now()
print(f'performance {pc} = {now2-now} seconds')

performance DOUG = 0:00:01.201076 seconds

```