Covid-19 Vaccination Campaign in Germany

The data used here were provided by Robert Koch Institute and the German federal ministry of Health.

These institutions publish the datasets and some analysis on the page impfdashboard.de.

Setup

Imports

In [70]:

```
# standard library
import datetime
import math
```

In [71]:

```
# third party
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import requests
import seaborn
```

Date this Notebook was run

In [72]:

```
today = datetime.datetime.today().strftime('%Y-%m-%d')
today
```

Out[72]:

```
'2021-11-01'
```

Set Defaults

In [73]:

```
# style like ggplot in R
plt.style.use('ggplot')
```

In [74]:

```
# Avoid cutting off part of the axis labels, see:
# https://stackoverflow.com/questions/6774086/why-is-my-xlabel-cut-off-in-my-matplotlib-plot
plt.rcParams.update({'figure.autolayout': True})
```

In [75]:

```
population_germany = 83_200_000
```

Get and Transform Data

In [76]:

```
vaccination_data_permalink = 'https://impfdashboard.de/static/data/germany_vaccinations_timeseries_v2.tsv'
vaccinations = pd.read_csv(
    vaccination_data_permalink,
    sep="\t")
```

Drop unnecessary / misleading columns

List all columns:

In [77]:

```
vaccinations.columns
```

Out[77]:

```
Index(['date', 'dosen_kumulativ', 'dosen_biontech_kumulativ',
       'dosen_biontech_erst_kumulativ', 'dosen_biontech_zweit_kumulativ',
       'dosen_biontech_dritt_kumulativ', 'dosen_moderna_kumulativ',
       'dosen_moderna_erst_kumulativ', 'dosen_moderna_zweit_kumulativ',
       'dosen_moderna_dritt_kumulativ', 'dosen_astra_kumulativ',
       'dosen_astra_erst_kumulativ', 'dosen_astra_zweit_kumulativ',
       'dosen_astra_dritt_kumulativ', 'dosen_johnson_kumulativ',
       'dosen_johnson_erst_kumulativ', 'dosen_johnson_zweit_kumulativ',
       'dosen_johnson_dritt_kumulativ', 'dosen_erst_kumulativ',
       'dosen_zweit_kumulativ', 'dosen_dritt_kumulativ',
       'dosen_differenz_zum_vortag', 'dosen_erst_differenz_zum_vortag',
       'dosen_zweit_differenz_zum_vortag', 'dosen_dritt_differenz_zum_vortag',
       'dosen_vollstaendig_differenz_zum_vortag', 'personen_erst_kumulativ',
       'personen_voll_kumulativ', 'personen_auffrisch_kumulativ',
       'impf_quote_erst', 'impf_quote_voll', 'dosen_dim_kumulativ',
       'dosen_kbv_kumulativ', 'indikation_alter_dosen',
       'indikation_beruf_dosen', 'indikation_medizinisch_dosen',
       'indikation_pflegeheim_dosen', 'indikation_alter_erst',
       'indikation_beruf_erst', 'indikation_medizinisch_erst',
       'indikation_pflegeheim_erst', 'indikation_alter_voll',
       'indikation_beruf_voll', 'indikation_medizinisch_voll',
       'indikation_pflegeheim_voll'],
      dtype='object')
```

Columns with names starting with 'indikation_' will not be analyzed as the data providers stopped updating them.

In [78]:

```
cols_to_drop = vaccinations.columns[vaccinations.columns.str.contains('indikation_')]
vaccinations.drop(columns=cols_to_drop, inplace=True)
```

Some more columns can be dropped, as there is no interest in analyzing differences on a vaccine level - especially since in some cases vaccines were mixed.

In [79]:

```
more_cols_to_drop = ['dosen_biontech_erst_kumulativ', 'dosen_biontech_zweit_kumulativ',
                     'dosen_moderna_erst_kumulativ', 'dosen_moderna_zweit_kumulativ',
                     'dosen_astra_erst_kumulativ', 'dosen_astra_zweit_kumulativ']
vaccinations.drop(columns=more_cols_to_drop, inplace=True)
```

Some columns are labeled misleadingly. As stated by the data provider the columns `personen_erst_kumulativ` and `impf_quote_erst` contain people vaccinated with the Johnson & Johnson vaccine. As this requires only one shot. the same persons are included in `personen_voll_kumulativ` . Therefore more columns are dropped and recalculated later.

In [80]:

```
vaccinations.drop(columns=['impf_quote_erst', 'impf_quote_voll'], inplace=True)
```

Convert datatype of date column

In [81]:

```
vaccinations.iloc[ : , [0]] = vaccinations.iloc[ : , [0]].apply(pd.to_datetime)
```

Show Data

In [82]:

```
vaccinations.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 309 entries, 0 to 308
Data columns (total 25 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   date                            309 non-null    datetime64[ns]
 1   dosen_kumulativ                 309 non-null    int64
 2   dosen_biontech_kumulativ        309 non-null    int64
 3   dosen_biontech_dritt_kumulativ  309 non-null    int64
 4   dosen_moderna_kumulativ         309 non-null    int64
 5   dosen_moderna_dritt_kumulativ   309 non-null    int64
 6   dosen_astra_kumulativ           309 non-null    int64
 7   dosen_astra_dritt_kumulativ     309 non-null    int64
 8   dosen_johnson_kumulativ         309 non-null    int64
 9   dosen_johnson_erst_kumulativ    309 non-null    int64
 10  dosen_johnson_zweit_kumulativ   309 non-null    int64
 11  dosen_johnson_dritt_kumulativ   309 non-null    int64
 12  dosen_erst_kumulativ            309 non-null    int64
 13  dosen_zweit_kumulativ           309 non-null    int64
 14  dosen_dritt_kumulativ           309 non-null    int64
 15  dosen_differenz_zum_vortag      309 non-null    int64
 16  dosen_erst_differenz_zum_vortag         309 non-null    int64
 17  dosen_zweit_differenz_zum_vortag        309 non-null    int64
 18  dosen_dritt_differenz_zum_vortag        309 non-null    int64
 19  dosen_vollstaendig_differenz_zum_vortag 309 non-null    int64
 20  personen_erst_kumulativ                 309 non-null    int64
 21  personen_voll_kumulativ                 309 non-null    int64
 22  personen_auffrisch_kumulativ            309 non-null    int64
 23  dosen_dim_kumulativ                     309 non-null    int64
 24  dosen_kbv_kumulativ                     309 non-null    int64
dtypes: datetime64[ns](1), int64(24)
memory usage: 60.5 KB
```

In [83]:

```
vaccinations.tail(3)
```

Out[83]:

	date	dosen_kumulativ	dosen_biontech_kumulativ	dosen_biontech_dritt_kumulativ	dosen_moderna_kumulativ	dosen_moderna_dritt_kumulativ	dos
306	2021-10-29	111893730	86099599	1953254	9784373	74008	
307	2021-10-30	111935530	86136572	1967626	9787450	75594	
308	2021-10-31	111949073	86148792	1972605	9788280	76082	

3 rows × 25 columns

Check Validity

In [84]:

```
# get the last row / the newest available data
last_row = vaccinations.tail(1)
```

In [85]:

```
doses_used = last_row['dosen_kumulativ']
doses_used
```

Out[85]:

```
308    111949073
Name: dosen_kumulativ, dtype: int64
```

In [86]:

```
# The number of person having been vaccinated at least once, includes those fully vaccinated
at_least_once = last_row['personen_erst_kumulativ']
fully_vaccinated_people = last_row['personen_voll_kumulativ']
partially_vaccinated_people = at_least_once - fully_vaccinated_people
```

```
# The johnson & Johnson vaccine is the only one used in Germany that only needs a single shot:
johnson_doses = last_row['dosen_johnson_kumulativ']
```

In [87]:

```
# Must be exactly 0
result_substraction = doses_used - partially_vaccinated_people - (fully_vaccinated_people - johnson_doses) * 2 - johnso
result_substraction
```

Out[87]:

```
308    2049762
dtype: int64
```

In [88]:

```
result_substraction ==  0
```

Out[88]:

```
308    False
dtype: bool
```

Calculate columns

In [89]:

```
vaccinations['partly vaccinated'] = round(
    (vaccinations['personen_erst_kumulativ'] - vaccinations['personen_voll_kumulativ']) * 100 / population_germany,
    2)
```

In [90]:

```
vaccinations['fully vaccinated'] = round(
    vaccinations['personen_voll_kumulativ'] * 100 / population_germany,
    2)
```

In [91]:

```
vaccinations.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 309 entries, 0 to 308
Data columns (total 27 columns):
 #   Column                                 Non-Null Count  Dtype
---  ------                                 --------------  -----
 0   date                                   309 non-null    datetime64[ns]
 1   dosen_kumulativ                        309 non-null    int64
 2   dosen_biontech_kumulativ               309 non-null    int64
 3   dosen_biontech_dritt_kumulativ         309 non-null    int64
 4   dosen_moderna_kumulativ                309 non-null    int64
 5   dosen_moderna_dritt_kumulativ          309 non-null    int64
 6   dosen_astra_kumulativ                  309 non-null    int64
 7   dosen_astra_dritt_kumulativ            309 non-null    int64
 8   dosen_johnson_kumulativ                309 non-null    int64
 9   dosen_johnson_erst_kumulativ           309 non-null    int64
 10  dosen_johnson_zweit_kumulativ          309 non-null    int64
 11  dosen_johnson_dritt_kumulativ          309 non-null    int64
 12  dosen_erst_kumulativ                     309 non-null    int64
 13  dosen_zweit_kumulativ                    309 non-null    int64
 14  dosen_dritt_kumulativ                    309 non-null    int64
 15  dosen_differenz_zum_vortag               309 non-null    int64
 16  dosen_erst_differenz_zum_vortag          309 non-null    int64
 17  dosen_zweit_differenz_zum_vortag         309 non-null    int64
 18  dosen_dritt_differenz_zum_vortag         309 non-null    int64
 19  dosen_vollstaendig_differenz_zum_vortag  309 non-null    int64
 20  personen_erst_kumulativ                  309 non-null    int64
 21  personen_voll_kumulativ                  309 non-null    int64
 22  personen_auffrisch_kumulativ             309 non-null    int64
 23  dosen_dim_kumulativ                      309 non-null    int64
 24  dosen_kbv_kumulativ                      309 non-null    int64
 25  partly vaccinated                        309 non-null    float64
 26  fully vaccinated                         309 non-null    float64
dtypes: datetime64[ns](1), float64(2), int64(24)
memory usage: 65.3 KB
```

In [92]:

```
vaccinations.tail(3)
```

Out[92]:

	date	dosen_kumulativ	dosen_biontech_kumulativ	dosen_biontech_dritt_kumulativ	dosen_moderna_kumulativ	dosen_moderna_dritt_kumulativ	dos
306	2021-10-29	111893730	86099599	1953254	9784373	74008	
307	2021-10-30	111935530	86136572	1967626	9787450	75594	
308	2021-10-31	111949073	86148792	1972605	9788280	76082	

3 rows × 27 columns

Last Update

Often the data is not updated on weekends, so get the highest date in the dataset.

In [93]:

```
last_update = vaccinations.loc[vaccinations.index[-1], "date"].strftime('%Y-%m-%d')
last_update
```

Out[93]:

```
'2021-10-31'
```

Doses Used

In [94]:

```
doses = vaccinations.loc[ : , ['date', 'dosen_differenz_zum_vortag']]
# Rename columns
doses.columns = ['date', 'doses used']
```

In [95]:

```
# Scale number of doses as millions
doses['doses used'] = doses['doses used'] / 1_000_000
```

Doses Daily

In [96]:

```
doses_daily = doses.set_index('date', inplace=False)
doses_daily.tail(1)
```

Out[96]:

	doses used
date	
2021-10-31	0.013543

In [97]:

```
# What is the highest number of doses used in a day?
max_doses_daily = max(doses_daily['doses used'])
max_doses_daily
```

Out[97]:

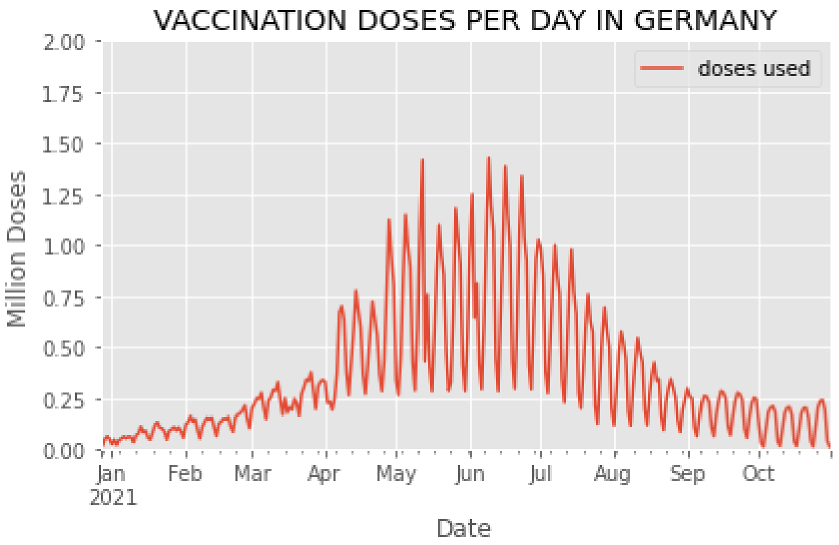
```
1.428418
```

In [98]:

```
doses_daily.plot(
    ylim=(0,math.ceil(max_doses_daily)),
    xlabel='Date',
    ylabel='Million Doses',
    title='VACCINATION DOSES PER DAY IN GERMANY')
```

Out[98]:

```
<AxesSubplot:title={'center':'VACCINATION DOSES PER DAY IN GERMANY'}, xlabel='Date', ylabel='Million Doses'>
```



Doses per Weekday (in the last 6 weeks)

In [99]:

```
last_6_weeks = doses.tail(42)
```

In [100…

```
# Yields a warning, but exactly like the docs prescribe and it works
# https://pandas.pydata.org/docs/getting_started/intro_tutorials/05_add_columns.html
last_6_weeks['weekday'] = last_6_weeks['date'].dt.day_name()
```

```
<ipython-input-100-45013977109e>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a
-view-versus-a-copy
  last_6_weeks['weekday'] = last_6_weeks['date'].dt.day_name()
```

In [101…

```
# check:
last_6_weeks.tail(3)
```

Out[101…

	date	doses used	weekday
306	2021-10-29	0.192689	Friday
307	2021-10-30	0.041800	Saturday
308	2021-10-31	0.013543	Sunday

In [102…

```
# drop the date column
last_6_weeks = last_6_weeks.drop(labels=['date'], axis=1)
```

In [103…

```
#last_6_weeks.set_index('weekday', inplace=True)
last_6_weeks.tail(3)
```

Out[103…

```
     doses used   weekday
306    0.192689    Friday
307    0.041800  Saturday
308    0.013543    Sunday
```

In [104…

```
pivot_table =last_6_weeks.pivot(columns='weekday', values='doses used')
pivot_table.tail()
```

Out[104…

```
weekday    Friday  Monday  Saturday    Sunday  Thursday  Tuesday  Wednesday
    304       NaN     NaN       NaN       NaN       NaN      NaN   0.235502
    305       NaN     NaN       NaN       NaN  0.240031      NaN        NaN
    306  0.192689     NaN       NaN       NaN       NaN      NaN        NaN
    307       NaN     NaN    0.0418       NaN       NaN      NaN        NaN
    308       NaN     NaN       NaN  0.013543       NaN      NaN        NaN
```

In [105…

```
# Reorder the columns
pivot_table = pivot_table[['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']]
# Rename the columns
pivot_table.columns=['Mo', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun']
pivot_table.tail()
```

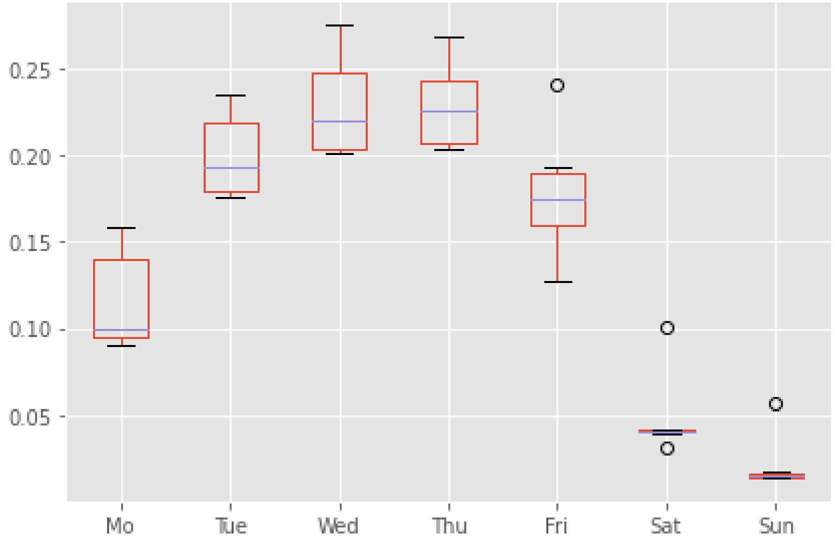
Out[105…

```
      Mo  Tue       Wed       Thu       Fri     Sat  Sun
304  NaN  NaN  0.235502       NaN       NaN     NaN  NaN
305  NaN  NaN       NaN  0.240031       NaN     NaN  NaN
306  NaN  NaN       NaN       NaN  0.192689     NaN  NaN
307  NaN  NaN       NaN       NaN       NaN  0.0418  NaN
```

	Mo	Tue	Wed	Thu	Fri	Sat	Sun
308	NaN	NaN	NaN	NaN	NaN	NaN	0.013543

In [106…

```
weekday_boxplot = pivot_table.boxplot()
```



In [107…

```
fig = weekday_boxplot.get_figure()
fig.savefig('img/weekday_boxplot.png')
```

Doses per Week

In [108…

```
# W-Mon in order to start the week on a Monday, see:
# https://pandas.pydata.org/pandas-docs/stable/user_guide/timeseries.html#anchored-offsets
doses_weekly = doses.groupby(pd.Grouper(key='date',freq='W-Mon')).sum()
doses_weekly.columns = ['million doses used']
doses_weekly.tail()
```

Out[108…

	million doses used
date	
2021-10-04	0.987565
2021-10-11	0.930570

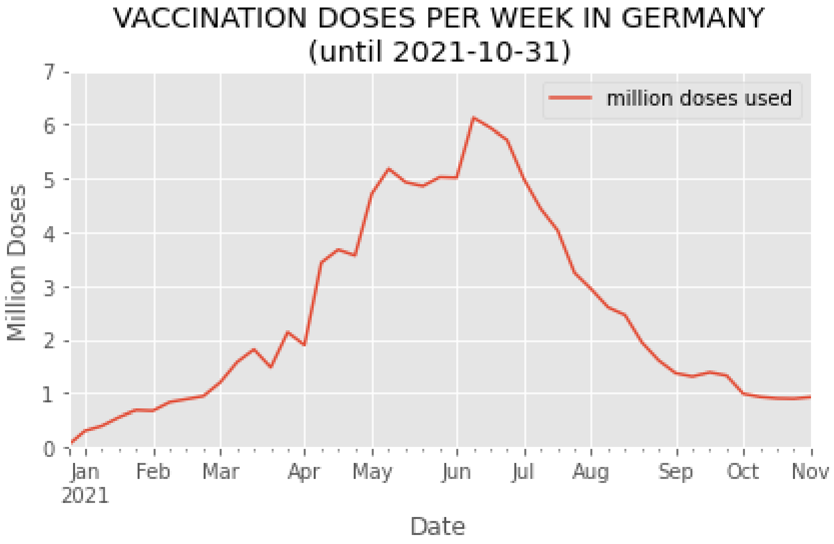
	million doses used
date	
2021-10-18	0.903601
2021-10-25	0.897888
2021-11-01	0.927439

```
In [109…
# What is the highest number of doses used in a week?
max_million_doses_weekly = max(doses_weekly['million doses used'])
max_million_doses_weekly
```

```
Out[109… 6.125344999999999
```

```
In [110…
doses_weekly.plot(
    ylim=(0, math.ceil(max_million_doses_weekly)),
    xlabel='Date',
    ylabel='Million Doses',
    title=f"VACCINATION DOSES PER WEEK IN GERMANY\n(until {last_update})")
```

```
Out[110… <AxesSubplot:title={'center':'VACCINATION DOSES PER WEEK IN GERMANY\n(until 2021-10-31)'}, xlabel='Date', ylabel='Milli
on Doses'>
```



Doses per Month

In [111…

```python
# M = month end frequency
doses_monthly = doses.groupby(pd.Grouper(key='date',freq='M')).sum()
doses_monthly.tail()
```

Out[111…

	doses used
date	
2021-06-30	24.761235
2021-07-31	17.266566
2021-08-31	9.270937
2021-09-30	5.876751
2021-10-31	3.927444

In [112…

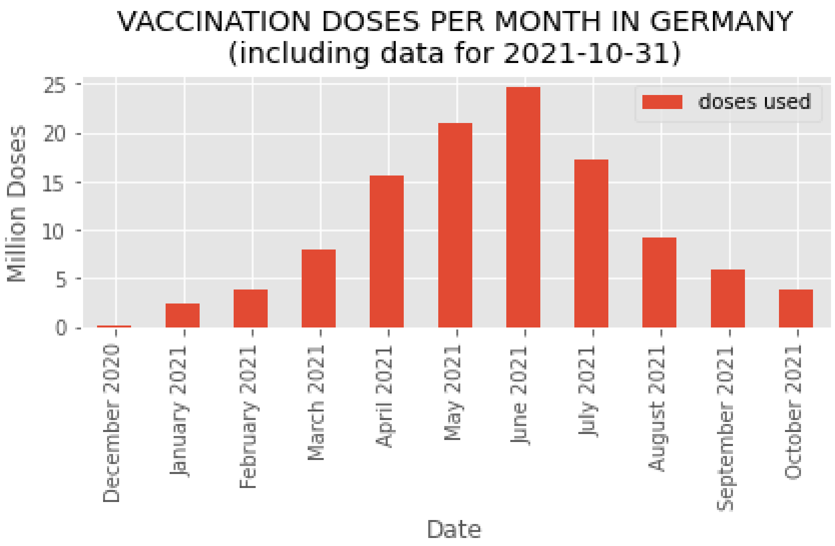
```python
max_doses_monthly = max(doses_monthly['doses used'])
max_doses_monthly
doses_monthly['month'] = doses_monthly.index.strftime('%B')
doses_monthly['year'] =  doses_monthly.index.strftime('%Y')
doses_monthly['label'] = doses_monthly['month'] + ' ' + doses_monthly['year']
doses_monthly.drop(columns=['month', 'year'], inplace=True)
doses_monthly.set_index('label', inplace=True)
doses_monthly.tail(6)
```

Out[112…

	doses used
label	
May 2021	21.058458
June 2021	24.761235
July 2021	17.266566
August 2021	9.270937
September 2021	5.876751
October 2021	3.927444

In [113…

```python
monthly_plot = doses_monthly.plot.bar(
    ylim=(0,math.ceil(max_doses_monthly) + 1),
    xlabel='Date',
    ylabel='Million Doses',
    title=f"VACCINATION DOSES PER MONTH IN GERMANY\n(including data for {last_update})")
```



In [114…

```
fig = monthly_plot.get_figure()
fig.savefig('img/monthly_doses_germany.png')
```

Vaccination Campaign Progress

In [115…

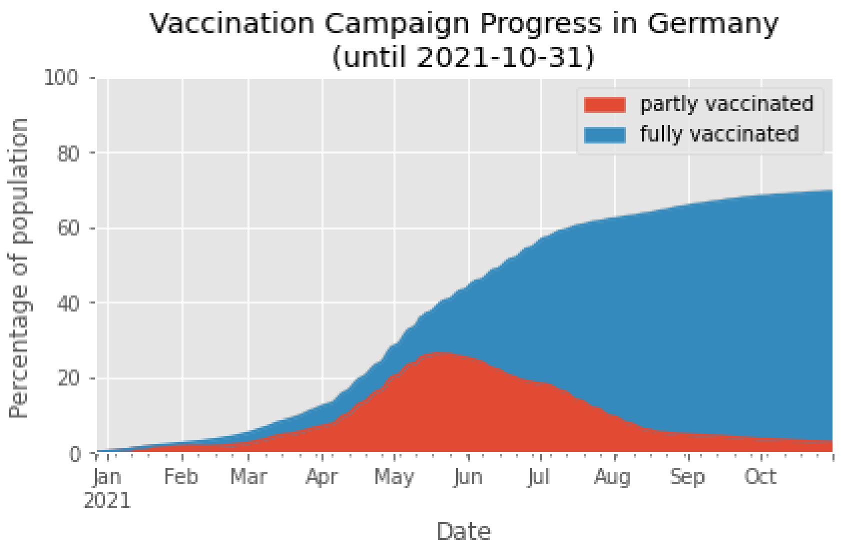
```
doses_cumulative = vaccinations.loc[ : , ['date', 'partly vaccinated', 'fully vaccinated']]
doses_cumulative.set_index('date', inplace=True)
doses_cumulative.tail(3)
```

Out[115…

date	partly vaccinated	fully vaccinated
2021-10-29	2.71	66.66
2021-10-30	2.70	66.68
2021-10-31	2.70	66.68

In [116…

```
doses_area_plot = doses_cumulative.plot.area(
    ylim=(0,100),
    xlabel='Date',
    ylabel='Percentage of population',
    title=f"Vaccination Campaign Progress in Germany\n(until {last_update})")
```



In [117…

```
fig = doses_area_plot.get_figure()
fig.savefig('img/vaccinations_germany_area_plot.png')
```

As of Today

In [118…

```
# get the last line of the data
current_state = doses_cumulative.iloc[-1]
current_state
```

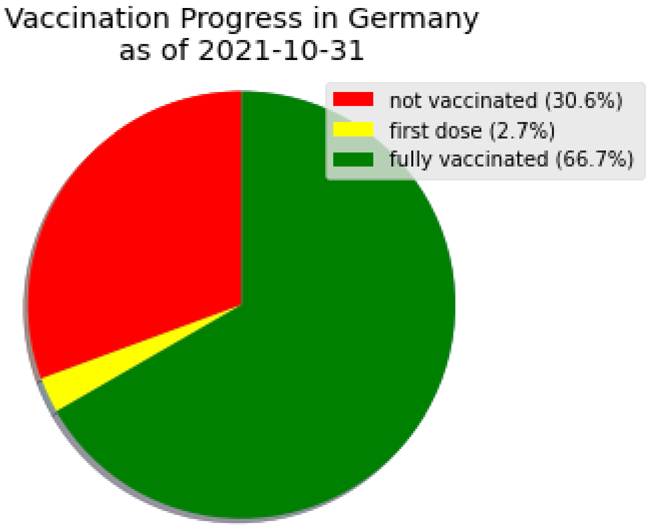
Out[118…

```
partly vaccinated     2.70
fully vaccinated     66.68
Name: 2021-10-31 00:00:00, dtype: float64
```

In [119…

```
percentage_not_vacc = 100 - current_state['partly vaccinated'] - current_state['fully vaccinated']
labels = [f"not vaccinated ({round(percentage_not_vacc, 1)}%)",
          f"first dose ({round(current_state['partly vaccinated'], 1)}%)",
          f"fully vaccinated ({round(current_state['fully vaccinated'], 1)}%)"]
colors = ['red', 'yellow', 'green']
sizes = [percentage_not_vacc,
         current_state['partly vaccinated'],
         current_state['fully vaccinated']]
fig1, ax1 = plt.subplots()
```

```
ax1.pie(sizes, shadow=True, startangle=90)
ax1.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
patches, texts = plt.pie(sizes, colors=colors, startangle=90)
plt.legend(patches, labels, loc="best")
plt.title(f"Vaccination Progress in Germany\nas of {last_update}")
# plt.savefig must be before show()
# BEWARE plt.savefig must be in the same Jupyter code cell that creates the graph!
# See comment by ijoseph here:
# https://stackoverflow.com/questions/9012487/matplotlib-pyplot-savefig-outputs-blank-image
plt.savefig('img/vaccination_in_germany_pie.png', bbox_inches='tight')
plt.show()
```



Vaccines in Use

In [120…

```
vaccinations.columns
```

Out[120…

```
Index(['date', 'dosen_kumulativ', 'dosen_biontech_kumulativ',
       'dosen_biontech_dritt_kumulativ', 'dosen_moderna_kumulativ',
       'dosen_moderna_dritt_kumulativ', 'dosen_astra_kumulativ',
       'dosen_astra_dritt_kumulativ', 'dosen_johnson_kumulativ',
       'dosen_johnson_erst_kumulativ', 'dosen_johnson_zweit_kumulativ',
       'dosen_johnson_dritt_kumulativ', 'dosen_erst_kumulativ',
       'dosen_zweit_kumulativ', 'dosen_dritt_kumulativ',
       'dosen_differenz_zum_vortag', 'dosen_erst_differenz_zum_vortag',
       'dosen_zweit_differenz_zum_vortag', 'dosen_dritt_differenz_zum_vortag',
       'dosen_vollstaendig_differenz_zum_vortag', 'personen_erst_kumulativ',
       'personen_voll_kumulativ', 'personen_auffrisch_kumulativ',
       'dosen_dim_kumulativ', 'dosen_kbv_kumulativ', 'partly vaccinated',
       'fully vaccinated'],
      dtype='object')
```

In [121…

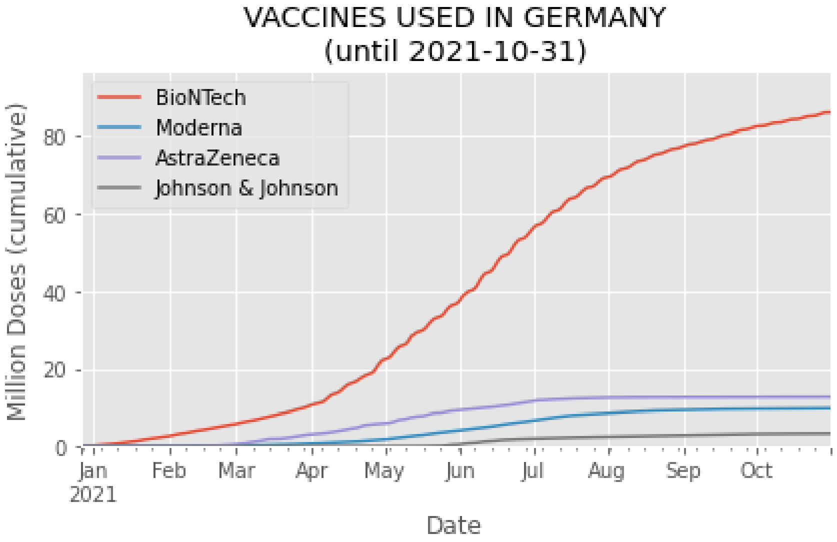
```
vaccine_use = vaccinations.loc[ : , ['date', 'dosen_biontech_kumulativ',
                                     'dosen_moderna_kumulativ',
                                     'dosen_astra_kumulativ',
                                     'dosen_johnson_kumulativ']]
# Rename columns
vaccine_use.columns = ['date', 'BioNTech', 'Moderna', 'AstraZeneca', 'Johnson & Johnson']
# make 'date' an index
vaccine_use.set_index('date', inplace=True)
# divide columns by 1 million
vaccine_use["BioNTech"] = vaccine_use["BioNTech"] / 1_000_000
vaccine_use["Moderna"] = vaccine_use["Moderna"] / 1_000_000
vaccine_use["AstraZeneca"] = vaccine_use["AstraZeneca"] / 1_000_000
vaccine_use["Johnson & Johnson"] = vaccine_use["Johnson & Johnson"] / 1_000_000
vaccine_use.tail(3)
```

Out[121…

	BioNTech	Moderna	AstraZeneca	Johnson & Johnson
date				
2021-10-29	86.099599	9.784373	12.704559	3.305199
2021-10-30	86.136572	9.787450	12.704721	3.306787
2021-10-31	86.148792	9.788280	12.704725	3.307276

In [122…

```
vaccines_used = vaccine_use.plot(
    # as it is cumulative, the last row must contain the single highest number
    ylim=(0,math.ceil(max(vaccine_use.iloc[-1]))+10),
    xlabel='Date',
    ylabel='Million Doses (cumulative)',
    title=f"VACCINES USED IN GERMANY\n(until {last_update})")
```



In [123…

```
fig = vaccines_used.get_figure()
fig.savefig('img/vaccines_used_in_germany.png')
```