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.

Get and Transform Data

In [1]:
```
# standard library
import datetime
import math
```

In [2]:
```
# third party
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import requests
```

In [3]:
```
today = datetime.datetime.today().strftime('%Y-%m-%d')
today
```

Out[3]: '2021-04-13'

In [4]:
```
yesterday = (datetime.datetime.today() + datetime.timedelta(days=-1)).strftime('%Y-%m-%d')
yesterday
```

Out[4]: '2021-04-12'

In [5]:
```
vaccination_data_permalink = 'https://impfdashboard.de/static/data/germany_vaccinations_timeseries_v2.tsv'
vaccinations = pd.read_csv(
    vaccination_data_permalink,
    sep="\t")
```

In [6]:
```
vaccinations.head(3)
```

Out[6]:

	date	dosen_kumulativ	dosen_differenz_zum_vortag	dosen_erst_differenz_zum_vortag	dosen_zweit_differenz_zum_vortag	dosen_biontech_kumulativ

	date	dosen_kumulativ	dosen_differenz_zum_vortag	dosen_erst_differenz_zum_vortag	dosen_zweit_differenz_zum_vortag	dosen_biontech_kumulativ
0	2020-12-27	24296	24296	24159	137	24296
1	2020-12-28	42679	18383	18383	0	42679
2	2020-12-29	91574	48895	48306	589	91574

3 rows × 24 columns

In [7]:
```
# Drop unnecessary columns
# No analysis of indication planned:
cols_to_drop = vaccinations.columns[vaccinations.columns.str.contains('indikation_')]
vaccinations.drop(columns=cols_to_drop, inplace=True)
```

In [8]:
```
# Convert datatype of date column
vaccinations.iloc[ : , [0]] = vaccinations.iloc[ : , [0]].apply(pd.to_datetime)
```

In [9]:
```
vaccinations.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 107 entries, 0 to 106
Data columns (total 12 columns):
 #   Column                          Non-Null Count  Dtype         
---  ------                          --------------  -----         
 0   date                            107 non-null    datetime64[ns]
 1   dosen_kumulativ                 107 non-null    int64         
 2   dosen_differenz_zum_vortag      107 non-null    int64         
 3   dosen_erst_differenz_zum_vortag 107 non-null    int64         
 4   dosen_zweit_differenz_zum_vortag 107 non-null   int64         
 5   dosen_biontech_kumulativ        107 non-null    int64         
 6   dosen_moderna_kumulativ         107 non-null    int64         
 7   dosen_astrazeneca_kumulativ     107 non-null    int64         
 8   personen_erst_kumulativ         107 non-null    int64         
 9   personen_voll_kumulativ         107 non-null    int64         
 10  impf_quote_erst                 107 non-null    float64       
 11  impf_quote_voll                 107 non-null    float64       
dtypes: datetime64[ns](1), float64(2), int64(9)
memory usage: 10.2 KB
```

Doses Used

In [10]:
```
doses = vaccinations.loc[ : , ['date', 'dosen_differenz_zum_vortag']]
doses.columns = ['date', 'doses used']
```

Doses Daily

In [11]:
```
doses_daily = doses.set_index('date', inplace=False)
doses_daily.tail(1)
```

Out[11]:

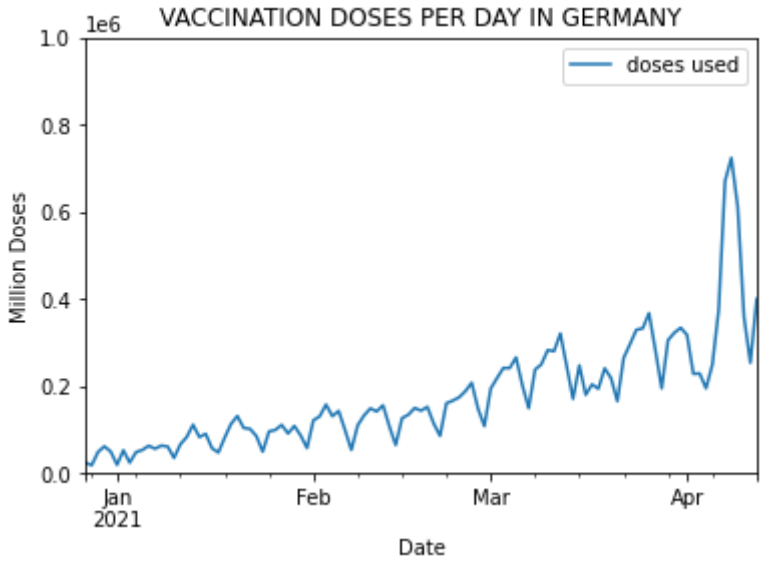
	doses used
date	
2021-04-12	399891

In [12]:
```
# What is the highest number of doses used in a day?
max_doses_daily = max(doses_daily['doses used'])
max_doses_daily
```

Out[12]:
```
723916
```

In [13]:
```
doses_daily.plot(
    ylim=(0,math.ceil(max_doses_daily / 10**6) * 10**6),
    xlabel='Date',
    ylabel='Million Doses',
    title='VACCINATION DOSES PER DAY IN GERMANY')
```

Out[13]:
```
<AxesSubplot:title={'center':'VACCINATION DOSES PER DAY IN GERMANY'}, xlabel='Date', ylabel='Million Doses'>
```



Doses per Week

In [14]:
```
# 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.tail()
```

Out[14]:

	doses used
date	
2021-03-15	1797487
2021-03-22	1467147
2021-03-29	2108962
2021-04-05	1875013
2021-04-12	3393485

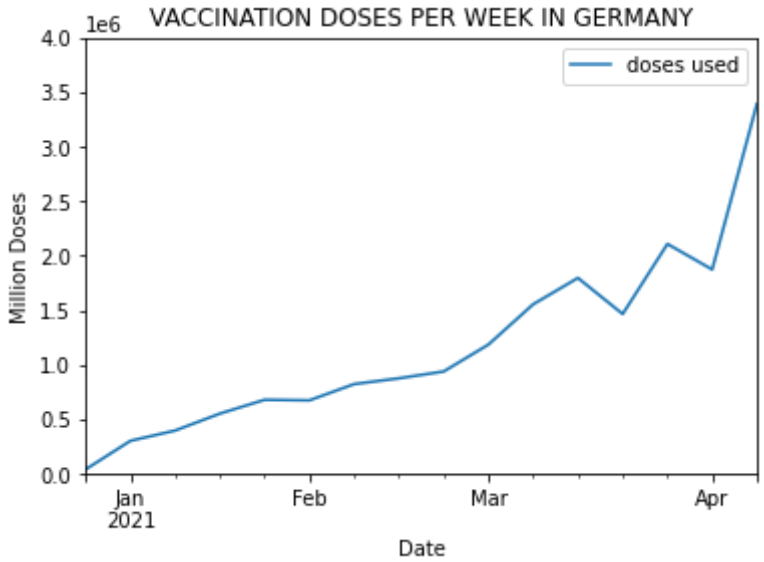
In [15]:
```
# What is the highest number of doses used in a week?
max_doses_weekly = max(doses_weekly['doses used'])
max_doses_weekly
```

3393485

Out[15]:

In [16]:
```
doses_weekly.plot(
    ylim=(0,math.ceil(max_doses_weekly / 10**6) * 10**6),
    xlabel='Date',
    ylabel='Million Doses',
    title='VACCINATION DOSES PER WEEK IN GERMANY')
```

Out[16]: <AxesSubplot:title={'center':'VACCINATION DOSES PER WEEK IN GERMANY'}, xlabel='Date', ylabel='Million Doses'>



Doses per Month

In [17]:
```
# M = month end frequency
doses_monthly = doses.groupby(pd.Grouper(key='date',freq='M')).sum()
doses_monthly.tail()
```

Out[17]:

	doses used
date	
2020-12-31	202796
2021-01-31	2332630
2021-02-28	3756932

date	doses used
2021-03-31	7780390
2021-04-30	4612125

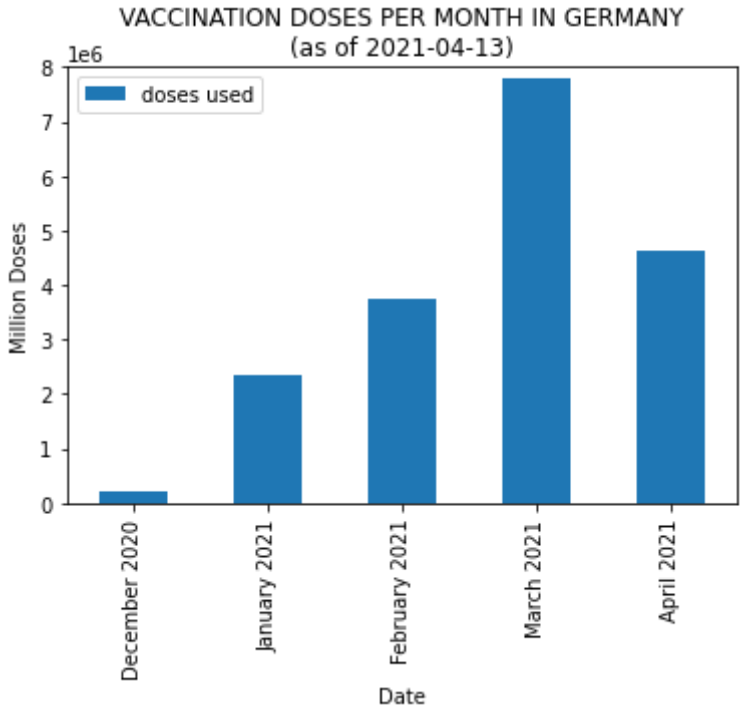
In [18]:
```
max_doses_monthly = max(doses_monthly['doses used'])
max_doses_monthly
doses_monthly['month'] = doses_monthly.index.strftime('%B')
doses_monthly['year'] =  doses_monthly.index.strftime('%Y')
doses_monthly['label'] = doses_monthly['month'] + ' ' + doses_monthly['year']
doses_monthly.drop(columns=['month', 'year'], inplace=True)
doses_monthly.set_index('label', inplace=True)
doses_monthly.tail(6)
```

Out[18]:

label	doses used
December 2020	202796
January 2021	2332630
February 2021	3756932
March 2021	7780390
April 2021	4612125

In [19]:
```
doses_monthly.plot.bar(
    ylim=(0,math.ceil(max_doses_monthly / 10**6) * 10**6),
    xlabel='Date',
    ylabel='Million Doses',
    title=f"VACCINATION DOSES PER MONTH IN GERMANY\n(as of {today})")
```

Out[19]:
```
<AxesSubplot:title={'center':'VACCINATION DOSES PER MONTH IN GERMANY\n(as of 2021-04-13)'}, xlabel='Date', ylabel='Mill
ion Doses'>
```



Vaccination Campaign Progress

In [20]:
```
doses_cumulative = vaccinations.loc[ : , ['date', 'personen_erst_kumulativ', 'personen_voll_kumulativ']]
doses_cumulative.set_index('date', inplace=True)
doses_cumulative.head(3)
```

Out[20]:

date	personen_erst_kumulativ	personen_voll_kumulativ
2020-12-27	24159	137
2020-12-28	42542	137
2020-12-29	90848	726

In [21]:
```
population_germany = 83_200_000
# Calculate new fields
doses_cumulative['first vaccination'] = round(
    doses_cumulative['personen_erst_kumulativ'] * 100 / population_germany,
```

```
    2)
doses_cumulative['fully vaccinated'] = round(
    doses_cumulative['personen_voll_kumulativ'] * 100 / population_germany,
    2)
doses_cumulative.drop(columns=['personen_erst_kumulativ','personen_voll_kumulativ'], inplace=True)
doses_cumulative.tail(3)
```

Out[21]:

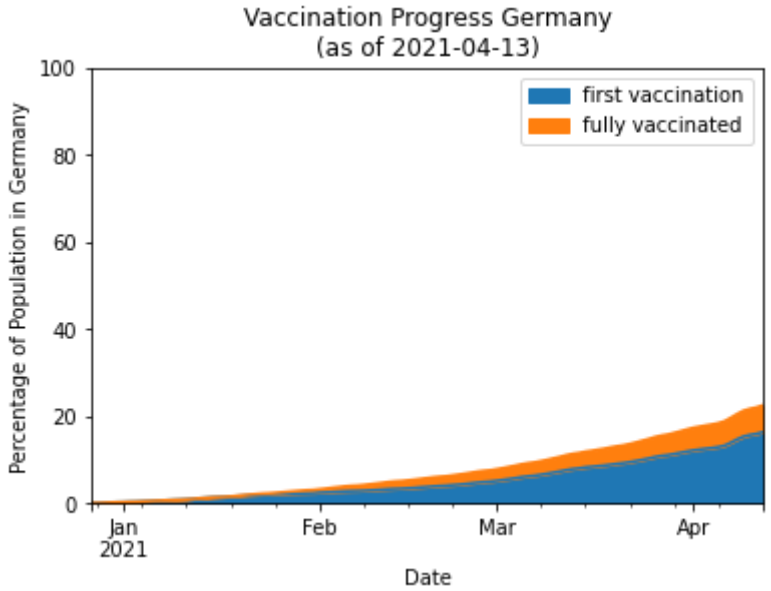
date	first vaccination	fully vaccinated
2021-04-10	15.67	6.00
2021-04-11	15.92	6.06
2021-04-12	16.31	6.15

In [22]:

```
doses_cumulative.plot.area(
    ylim=(0,100),
    xlabel='Date',
    ylabel='Percentage of Population in Germany',
    title=f"Vaccination Progress Germany\n(as of {today})")
```

Out[22]:

```
<AxesSubplot:title={'center':'Vaccination Progress Germany\n(as of 2021-04-13)'}, xlabel='Date', ylabel='Percentage of Population in Germany'>
```

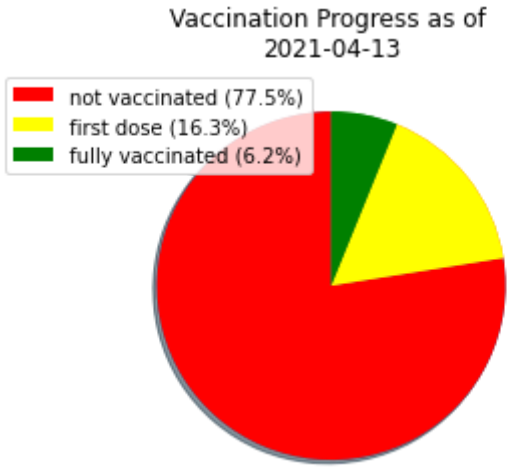


As of Today

In [23]:
```
# get the last line of the data
current_state = doses_cumulative.iloc[-1]
current_state
```

Out[23]:
```
first vaccination    16.31
fully vaccinated      6.15
Name: 2021-04-12 00:00:00, dtype: float64
```

In [24]:
```
percentage_not_vacc = 100 - current_state['first vaccination'] - current_state['fully vaccinated']
labels = [f"not vaccinated ({round(percentage_not_vacc, 1)}%)",
          f"first dose ({round(current_state['first vaccination'],1)}%)",
          f"fully vaccinated ({round(current_state['fully vaccinated'],1)}%)"]
colors = ['red', 'yellow', 'green']
sizes = [percentage_not_vacc,
         current_state['first vaccination'],
         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 as of \n{today}")
plt.show()
```



Vaccines in Use

In [25]:
```
vaccine_use = vaccinations.loc[ : , ['date', 'dosen_biontech_kumulativ', 'dosen_moderna_kumulativ', 'dosen_astrazeneca_
# Rename columns
vaccine_use.columns = ['date', 'Biontech', 'Moderna', 'AstraZeneca']
# make 'date' an index
vaccine_use.set_index('date', inplace=True)
vaccine_use.tail(3)
```

Out[25]:

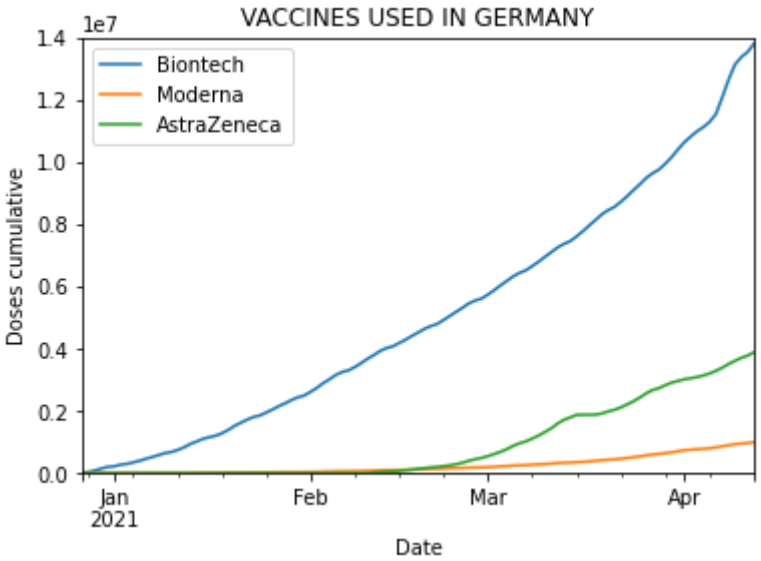
date	Biontech	Moderna	AstraZeneca
2021-04-10	13373064	951512	3706910
2021-04-11	13536869	969137	3778976
2021-04-12	13804290	995734	3884849

In [26]:
```
# To get the maximum for the y axis, round the highest
# number of doses up to the next million
max_doses = math.ceil(max(vaccine_use.iloc[-1]) / 10**6) * 10**6
max_doses
```

Out[26]:
```
14000000
```

In [27]:
```
vaccine_use.plot(
    ylim=(0,max_doses),
    xlabel='Date',
    ylabel='Doses cumulative',
    title='VACCINES USED IN GERMANY')
```

Out[27]:
```
<AxesSubplot:title={'center':'VACCINES USED IN GERMANY'}, xlabel='Date', ylabel='Doses cumulative'>
```


VACCINES USED IN GERMANY
1e7
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Biontech
Moderna
AstraZeneca
Doses cumulative
Jan
2021
Feb
Mar
Apr
Date