Covid-19 Analysis

Project for Computer Programming and Data Management

Federico Stragliotto 893690, Regin Kaci 873353, Elena Martarello 872265

Schedule

1

Covid-19 vaccination Analysis

- Studying the Covid-19 vaccination campaign in EU in 2021
- Focusing on the situation in Italy

2

Impact of Covid-19 on the economy

- Performance of Pfizer, Astrazeneca, Moderna and Johnson & Johnson on the stock exchange over the last three years
- Performance of SP500, Nasdaq, Dow Jones and FTSE MIB over the last 3 years

3

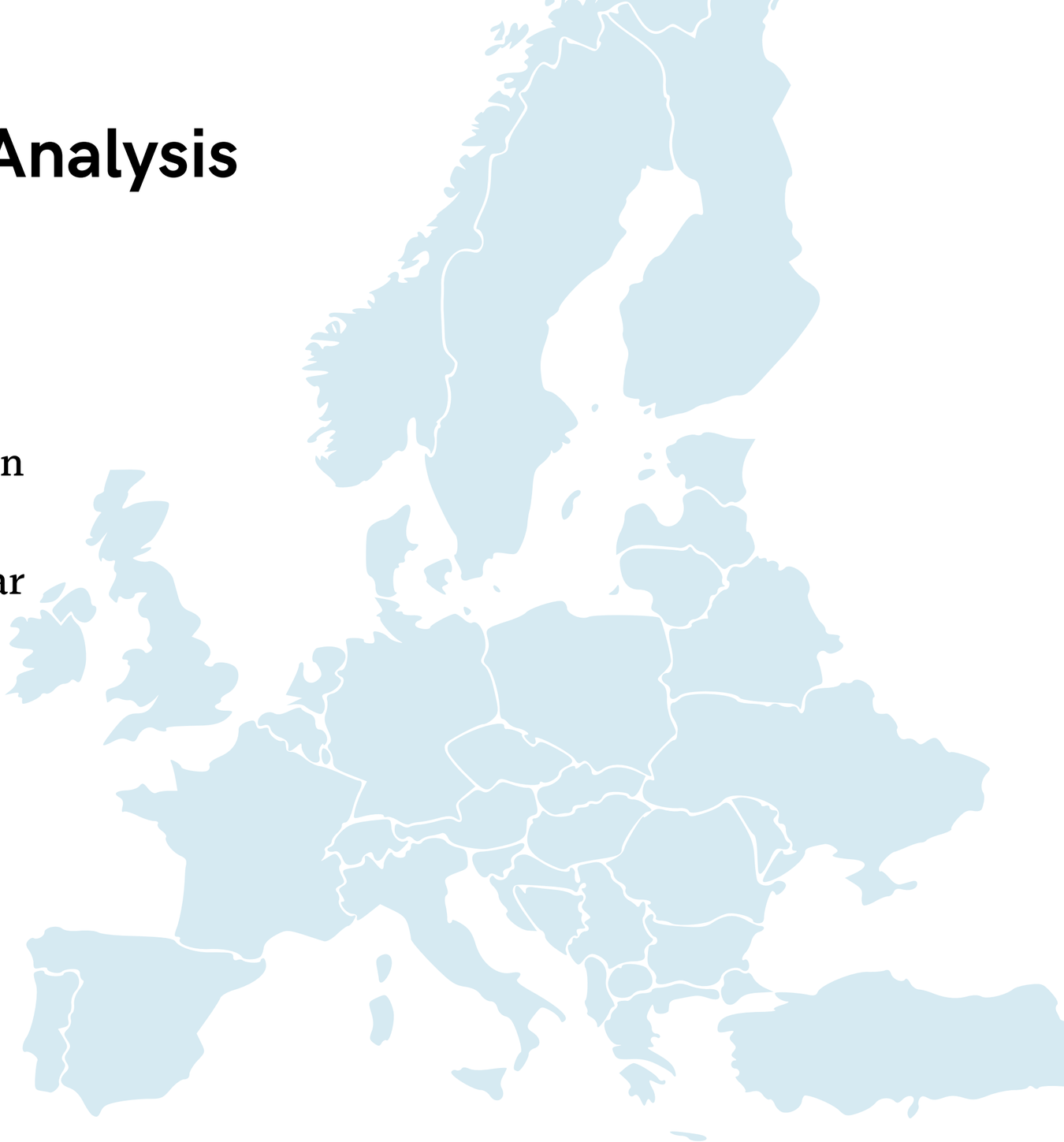
Covid-19's impact on the environment

- Overall analysis of the main air quality indices among 2019,2020,2021
- Greater focus on Italy, Norway,Romania

1 Covid-19 vaccination Analysis

Focus on European Union

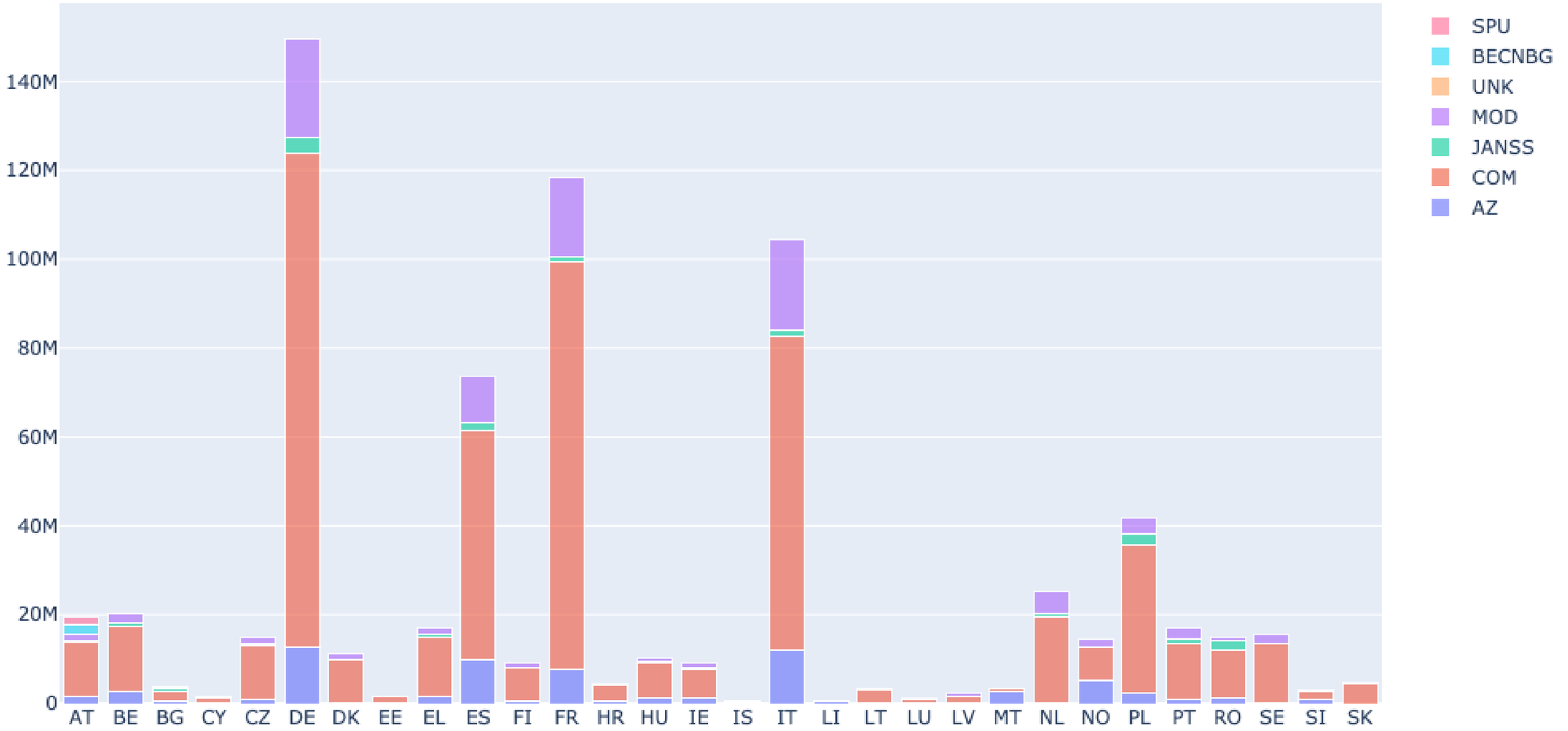
First of all we analysed the situation in the European Union.
Using data, we created a table and a bar chart based on frequency.



What is the most commonly used vaccine in EU?

Type of vaccines	Total doses
AZ	68959937
BECNBG	2239144
COM	525532100
JANSS	18732124
MOD	101712995
SPU	1845080
UNK	25204

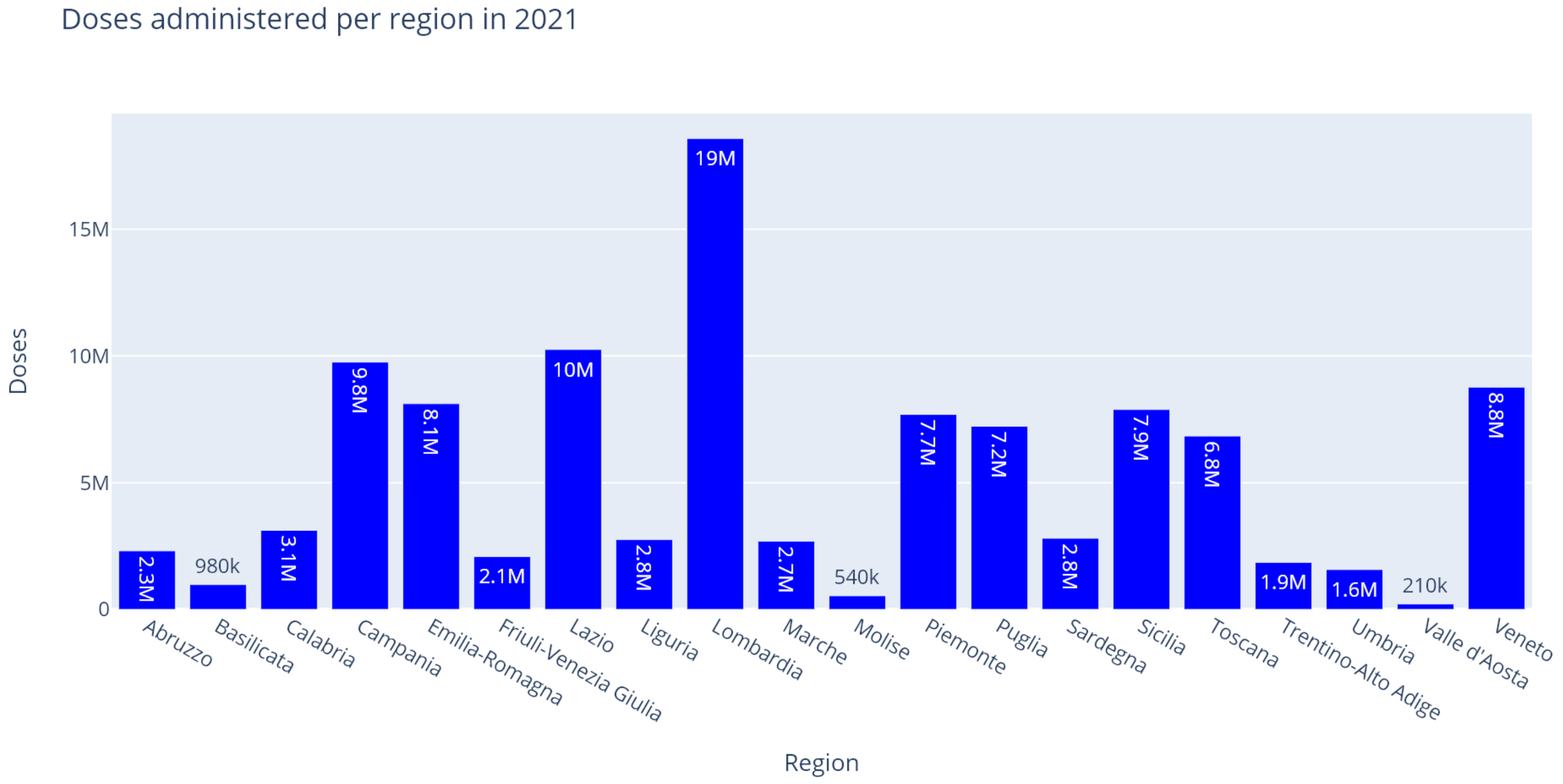
How the vaccination campaign is going in EU?



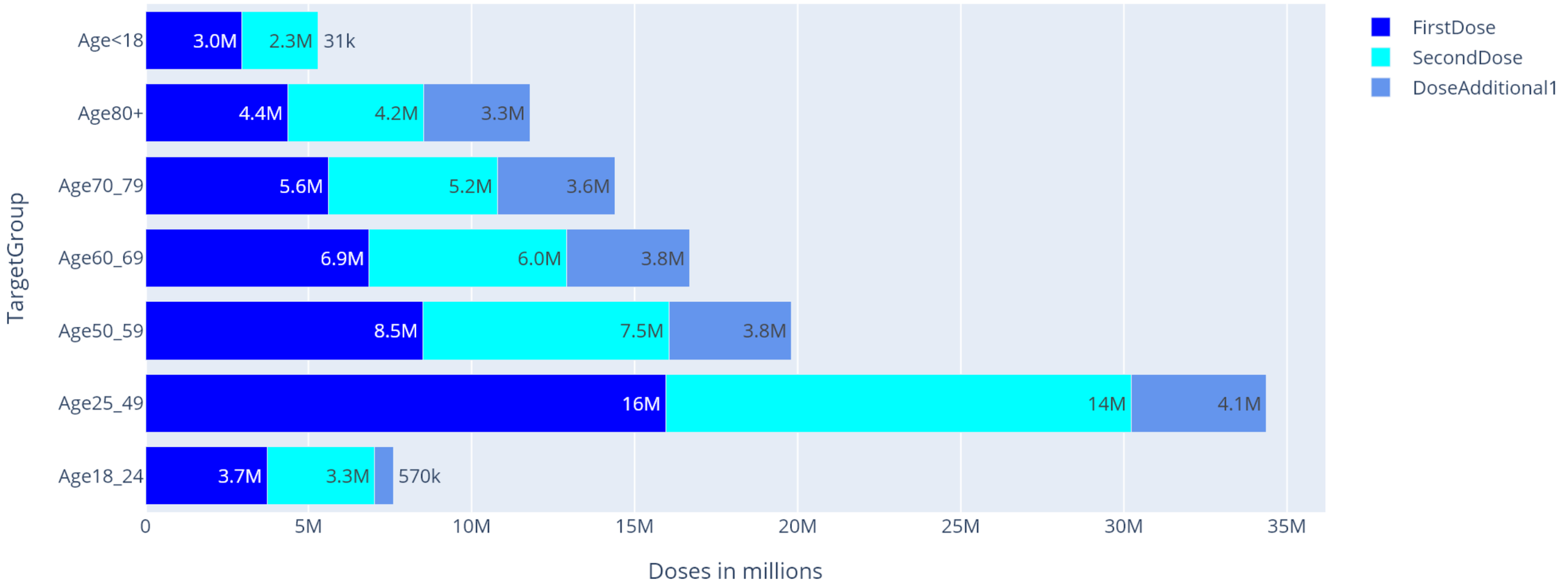
1 Covid-19 vaccination Analysis

Focus on Italy

Then we focused our research on Italy and we discovered that in 2020, have been administered 124'543 doses, while in 2021 the doses were 109'877'005.
From this results we made some graphs showing the progress of the vaccination campaign in this country.

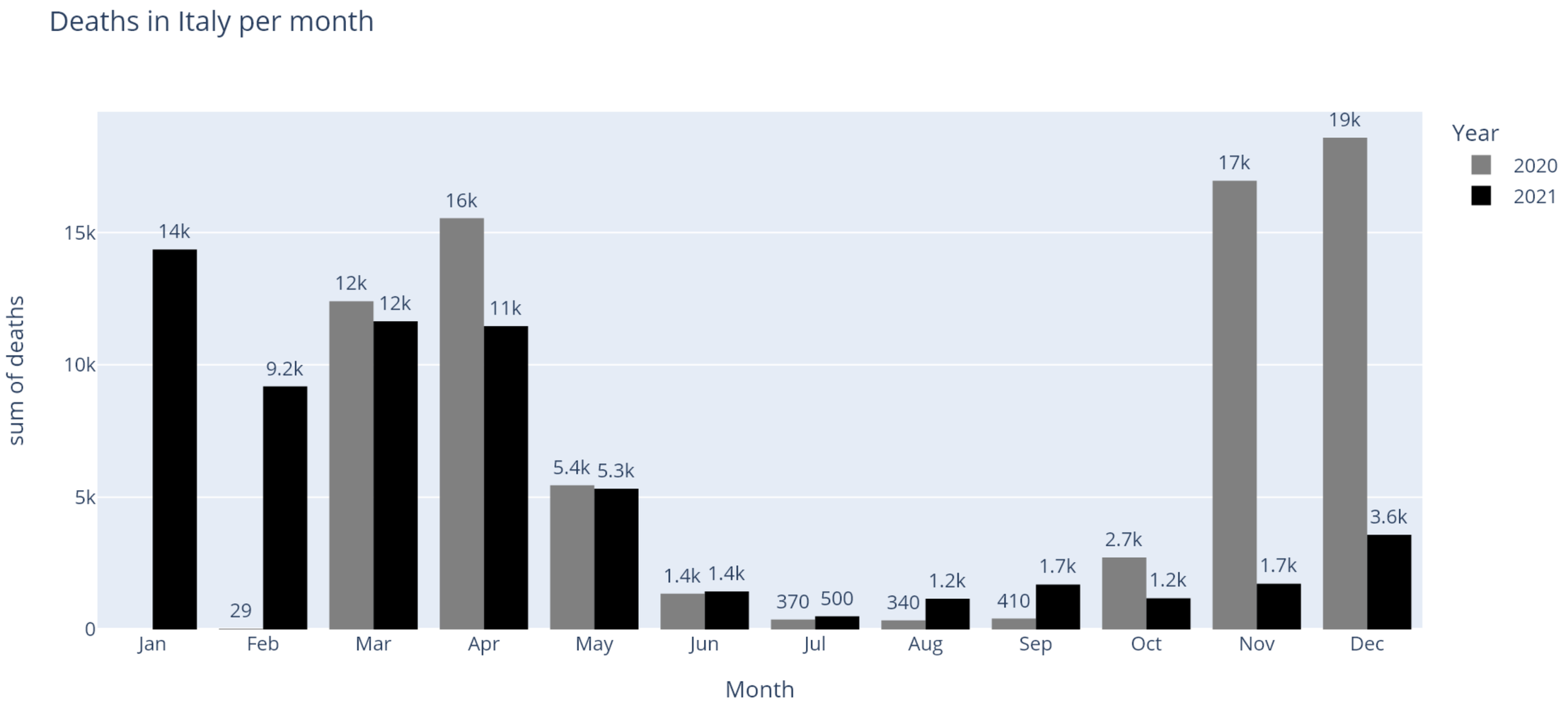

Doses administered per region in 2021
Doses
15M
10M
5M
0
2.3M
980k
3.1M
9.8M
8.1M
2.1M
10M
2.8M
19M
2.7M
540k
7.7M
7.2M
2.8M
7.9M
6.8M
1.9M
1.6M
210k
8.8M
Abruzzo
Basilicata
Calabria
Campania
Emilia-Romagna
Friuli-Venezia Giulia
Lazio
Liguria
Lombardia
Marche
Molise
Piemonte
Puglia
Sardegna
Sicilia
Toscana
Trentino-Alto Adige
Umbria
Valle d'Aosta
Veneto
Region

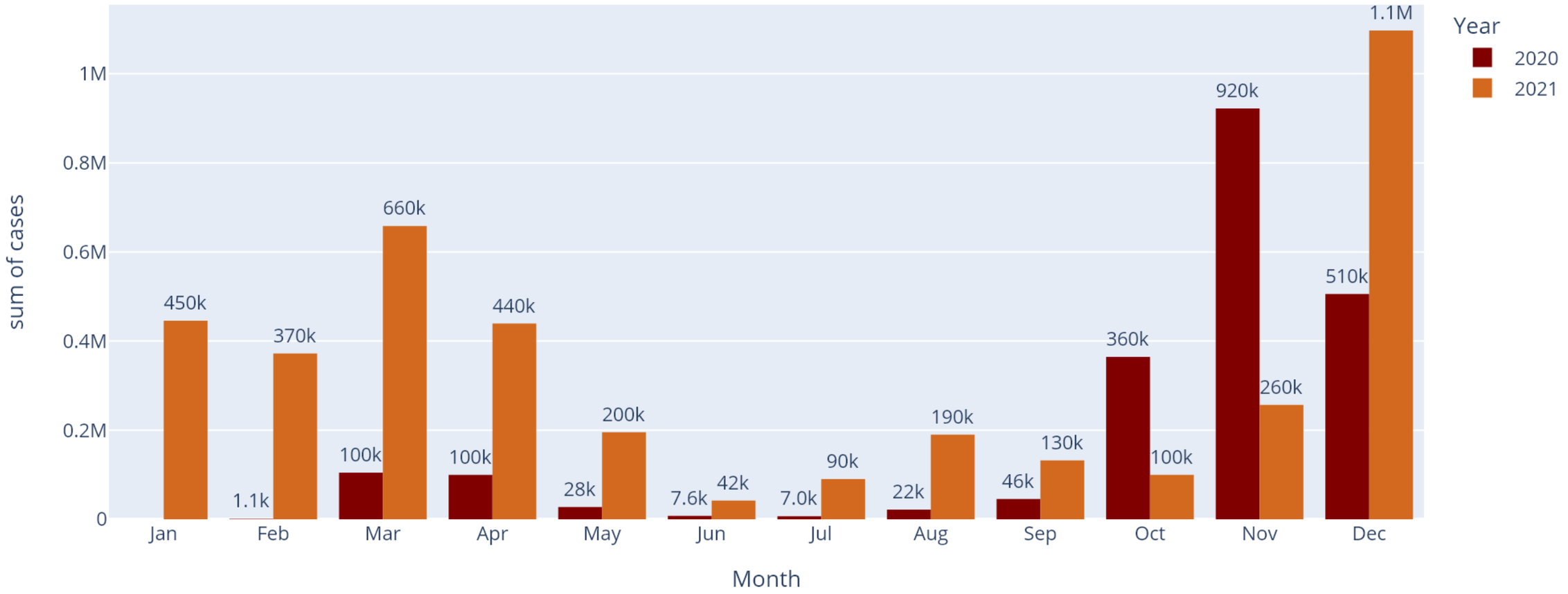
Doses administered per age in 2020-21



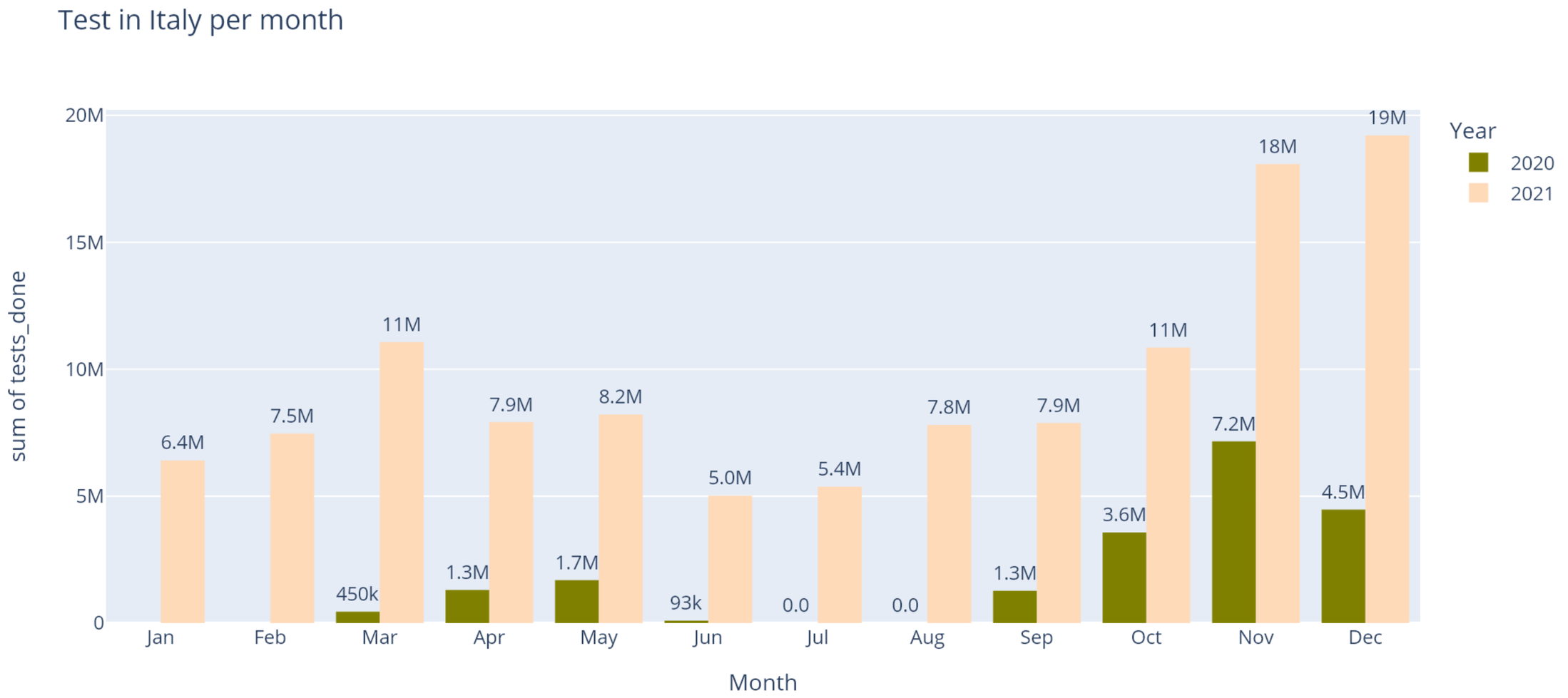

Vaccine administered in 2021
sum of Doses
15M
10M
5M
0
Month
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
1.0
2.0M
330k
1.9M
1.9M
4.5M
2.3M
6.4M
3.0M
2.0M
12M
1.6M
1.6M
11M
2.4M
1.9M
9.7M
450k
2.0M
5.9M
19k
3.7M
6.9k
3.7M
190
2.5M
6.1M
14
7.3M
4.0M
Vaccine
COM
JANSS
MOD
AZ

We studied how deaths and positive cases changed after this vaccination campaign

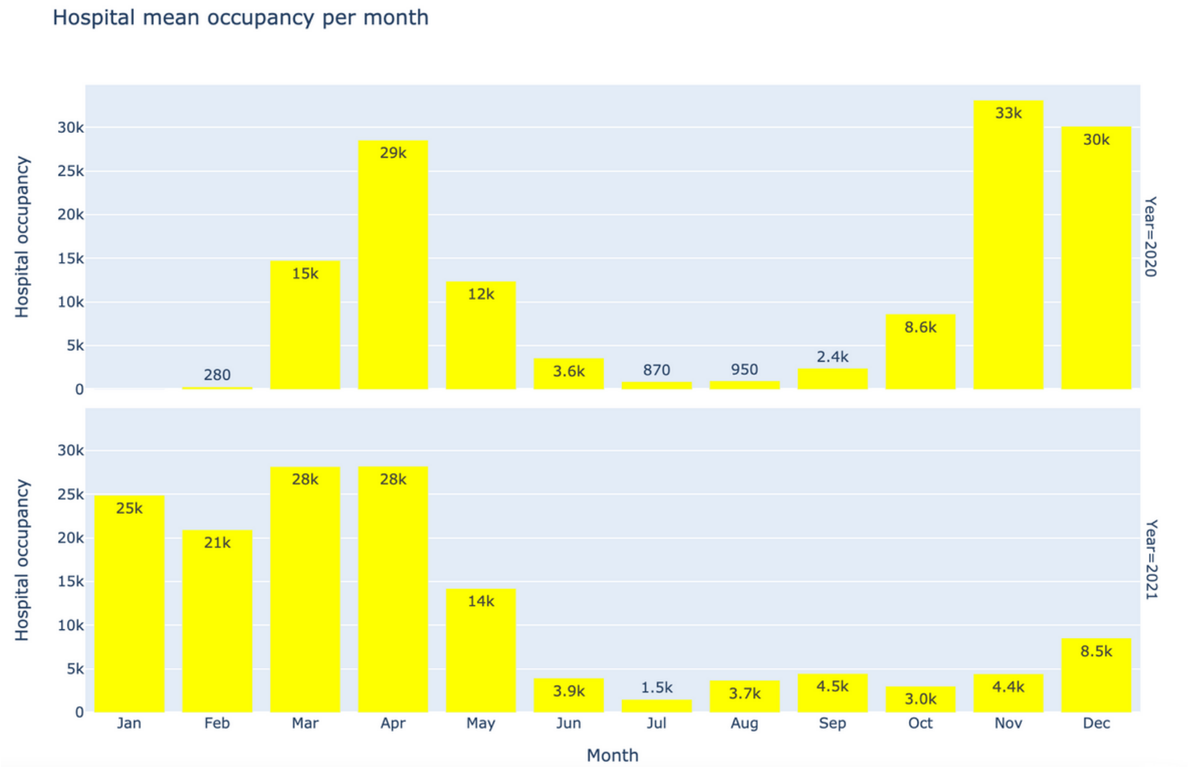


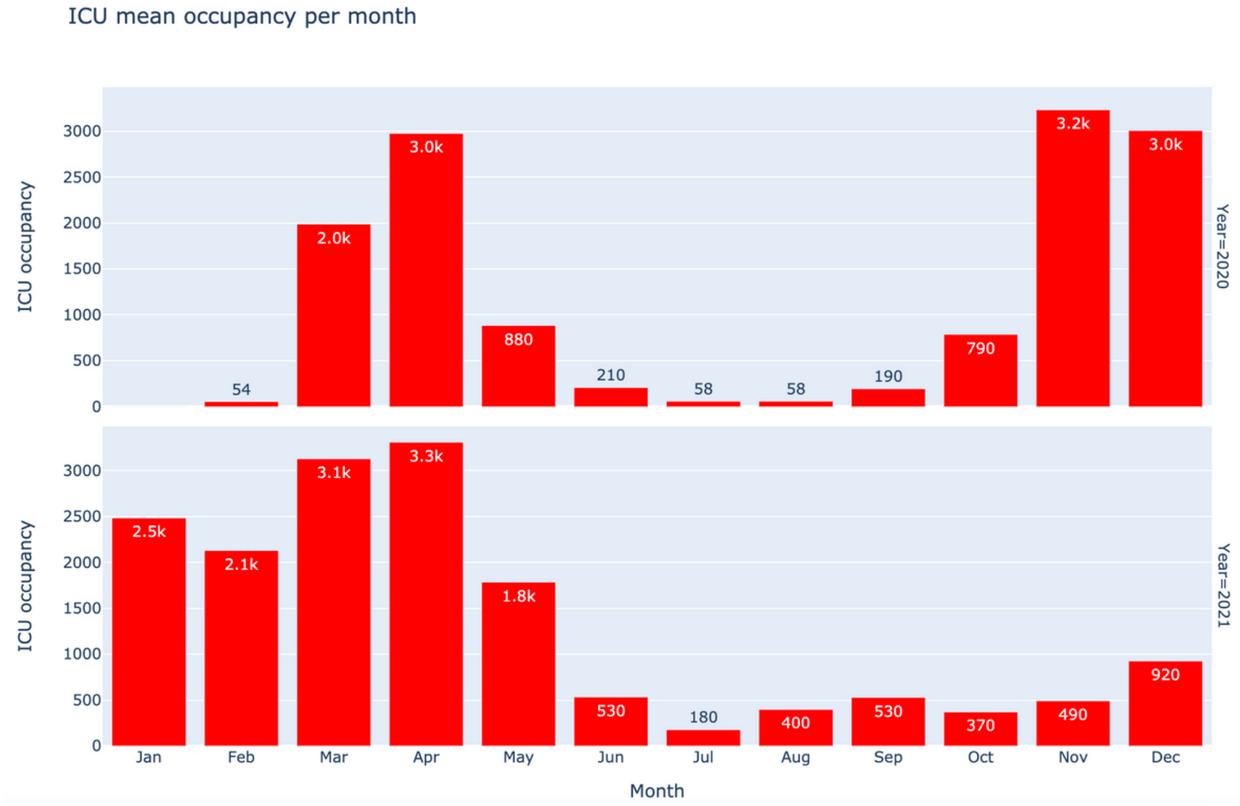

Cases in Italy per month
sum of cases
1M
0.8M
0.6M
0.4M
0.2M
0
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month
450k
1.1k
370k
100k
660k
100k
440k
28k
200k
7.6k
42k
7.0k
90k
22k
190k
46k
130k
360k
100k
920k
260k
510k
1.1M
Year
2020
2021

The number of deaths seems to be decreased while the number of positive cases is higher than the previous year, this can be explained by the higher number of test done in the 2021.



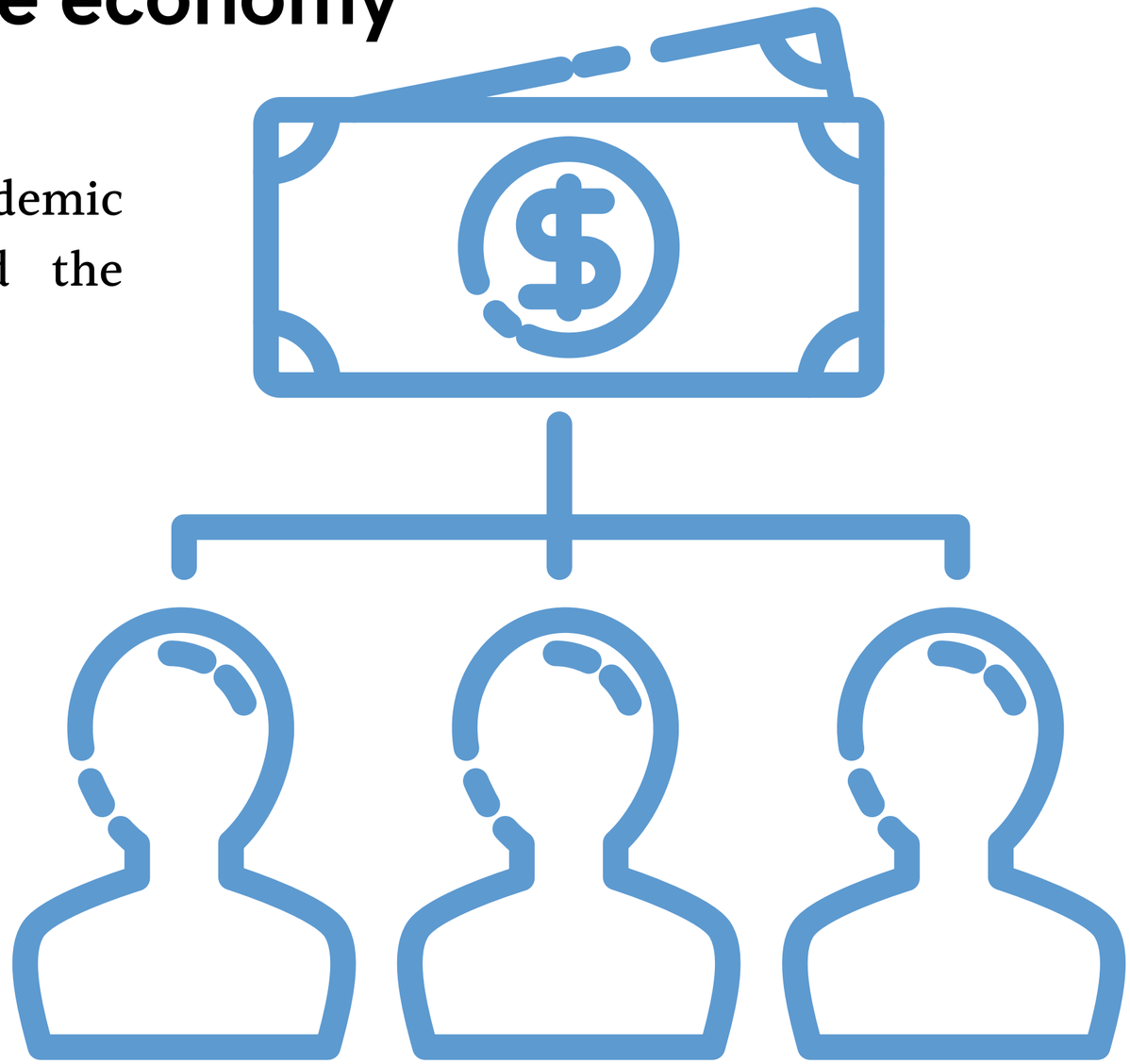
We checked even if the number of mean patient per month in ICU and in Hospital changed from the 2020



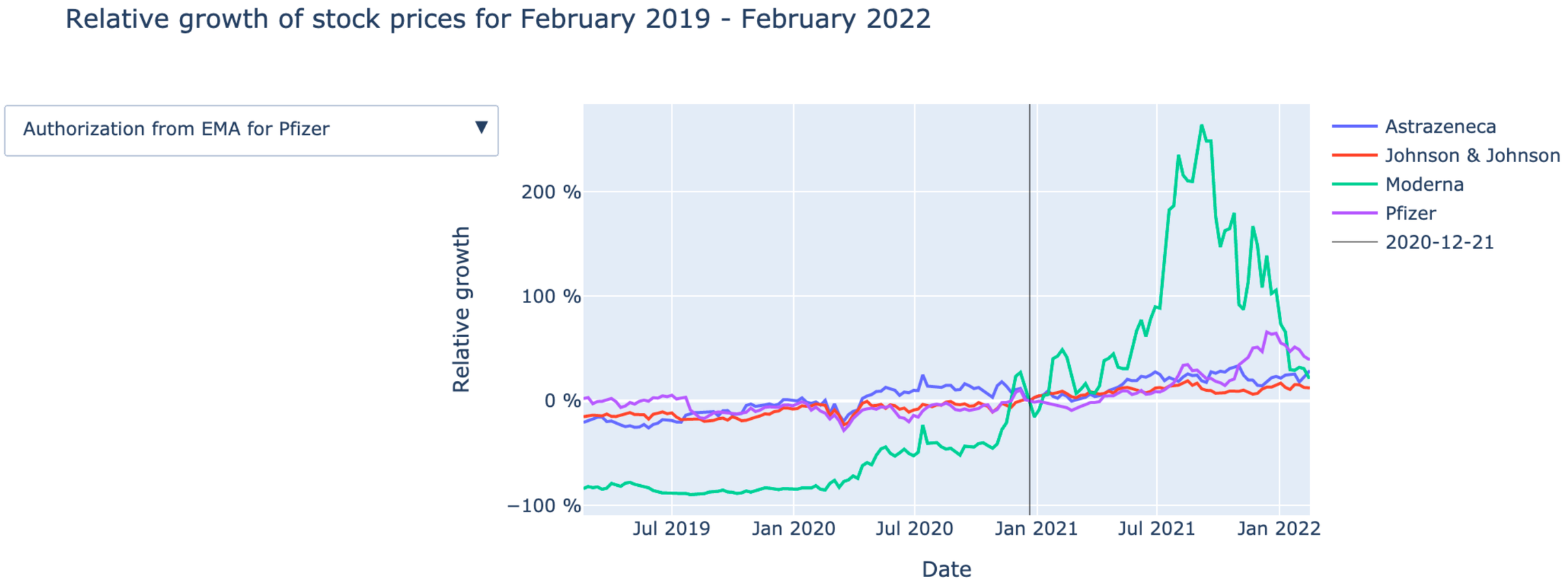


2 Impact of Covid-19 on the economy

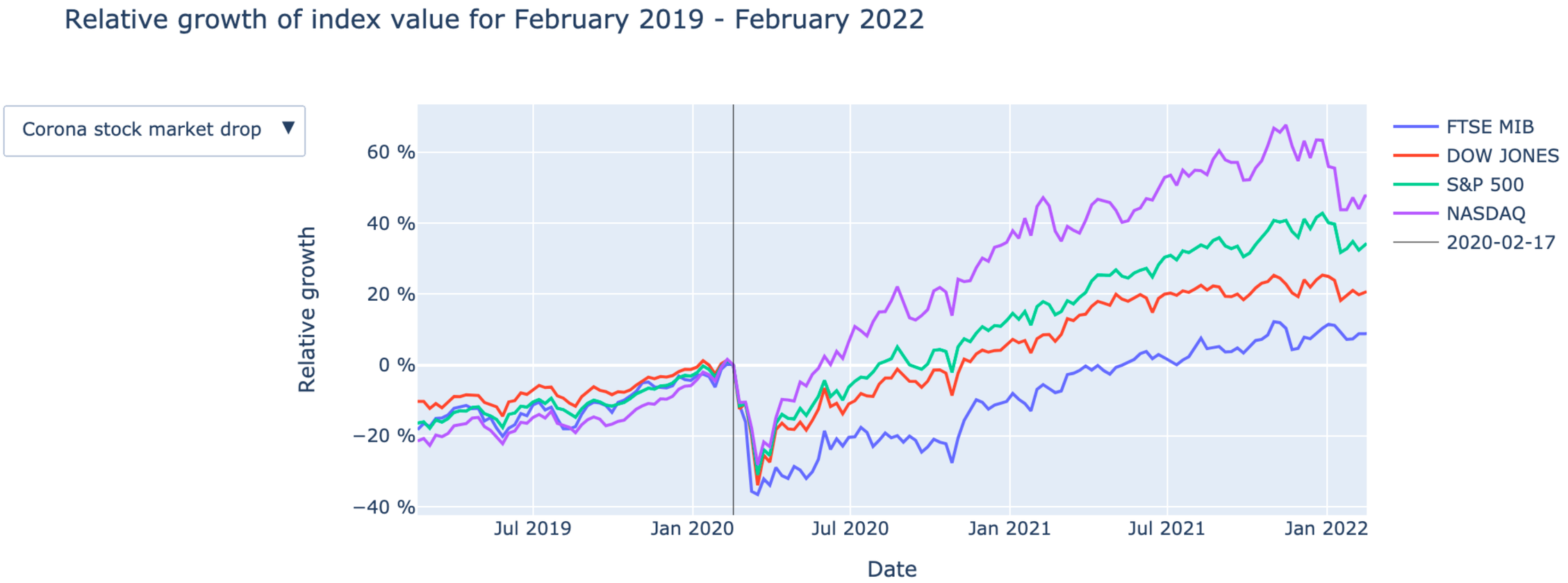
In this section we analysed how the pandemic and the vaccination campaign affected the economy by answering two questions.



We asked ourself how the 4 vaccine producers performed on the stock exchange during the whole pandemic.



We also wanted to see how the stock market has reacted over the last 3 years, so we studied the performance of 4 indices.



Chunks of code mainly used to produce histograms and bar charts

```
icu_italy[['Year', 'Month', 'Day']]=icu_italy['date'].str.split('-', expand=True)
```

```
df_icu=icu_italy.pivot_table(index=['Year', 'Month'], columns='indicator', values='value', aggfunc='mean').reset_index()
```

```
from datetime import datetime
from datetime import date
#create a function to convert the YearWeek format
def convert_from_ISO_to_date(d):
    return datetime.strptime(d + '-1', "%Y-W%W-%w")

#if the character at position 5 of the column YearWeekISO at index 0 is equal to 'W' then
if df_vaccine['YearWeekISO'][0][5]=='W':
    result=[]
    for i in df_vaccine['YearWeekISO']:
        i=convert_from_ISO_to_date(i)
        string=str(i)[0:7]
        result.append(string)
    df_vaccine['YearWeekISO']=result #put the result list into the column YearWeekISO
```

A bit of code used to create the line charts

```
for ref_key, ref_value in ref_dict.items():
    ref_date_fig.add_trace(go.Scatter(
        create_reference_line(
            x=ref_value['closest_date'],
            name=ref_value['closest_date'].strftime("%Y-%m-%d"),
            meta=dict(reference_date=ref_key)
            )
        ))


ref_date_fig.update_layout(
    updatemenus=[dict(buttons=create_buttons(), x=-0.12)])

# hide traces on initialization, by default all traces are shown
ref_date_fig.update_traces(visible=False)
ref_date_fig.update_traces(
    visible=True,
    selector=dict(meta={'reference_date': list(ref_dict.keys())[0]})
    )

ref_date_fig.show()
```

```
def create_visibility_array(ref_date):
    return [True if trace['meta']['reference_date'] == ref_date
            else False for trace in ref_date_fig['data']]
```

```
def create_indexed_columns(date, df, top_level_name=""):
    """Returns indexed columns for given dataframe"""

    # find index of the date that is closest to our reference date
    closest_date_index = df.index.get_loc(date, method="nearest")

    # get the values in the initial columns for the reference date
    reference_values = df.iloc[closest_date_index]['Adj Close']

    # divide initial columns by values at ref. date and store in intermediate df
    inter_df = df['Adj Close'].div(reference_values)*100 - 100

    # create a multindex for the intermediate df using the date as top-level index
    closest_date = df.index[closest_date_index]
    inter_df.columns = pd.MultiIndex.from_product(
                [[top_level_name if top_level_name else str(closest_date)], inter_df.columns])

    return inter_df, closest_date
```

```
def create_buttons():
    """Returns list of button objects."""

    button_list = []
    for ref_date in ref_dict.keys():
        ref_button = dict(
            label=ref_date,
            method="restyle",
            args=[dict(
                visible=create_visibility_array(ref_date))]
        )
        button_list.append(ref_button)
    return button_list
```

3 Covid-19's impact on the environment

Focus on Italy, Romania and Norway

We choose these three nations due to their geographic position (Nord, South an East Europe).
First we plotted a table of the mean value air indicators over the three year period.
Then we investigated the deviation of the averages of the main air pollution indices for 2019, 2020 and 2021.

2019

Country	Air quality index	Carbon Monoxide	Humidity	Nitrogen Dioxide	Ozone	Airborne particulates Matter	Fine particulate matter	Sulfur dioxide	Temperature
IT	0	87.47	121.4	122.48	107.74	100.07	101.46	77.29	122.47
RO	0	0	82.75	66.89	56.11	63.57	0	67.57	83.11
NO	0	140.01	101.85	95.94	68.78	89.9	78.31	0	102.81

2020

Country	Air quality index	Carbon Monoxide	Humidity	Nitrogen Dioxide	Ozone	Airborne particulates Matter	Fine particulate matter	Sulfur dioxide	Temperature
IT	0	76.24	119.91	129.09	115.95	100.91	102.47	80.24	119.98
RO	0	0	81.11	71.06	58.25	68.26	35.91	69.65	81.42
NO	0	128.77	143.26	135.84	23.21	132.57	126.81	0	143.81

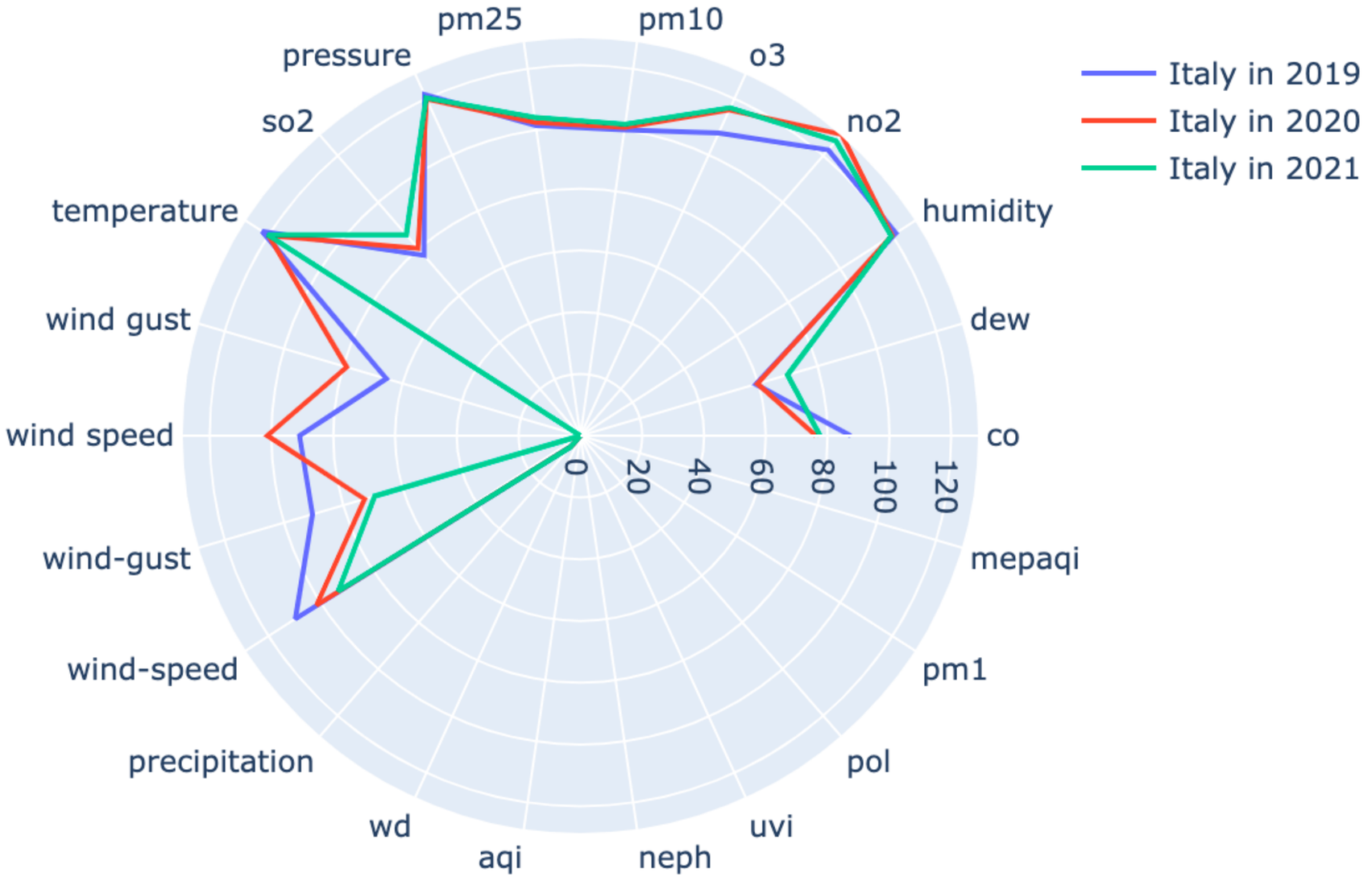
2021

Country	Air quality index	Carbon Monoxide	Humidity	Nitrogen Dioxide	Ozone	Airborne particulates Matter	Fine particulate matter	Sulfur dioxide	Temperature
IT	0	77.54	119.57	126.33	116.69	101.9	104.12	86.03	120.39
RO	0	0	82.92	77.86	46.26	78.76	45.99	57.39	82.99
NO	0	210.03	133.36	129.67	23.3	123.64	121.64	0	133.44

IT

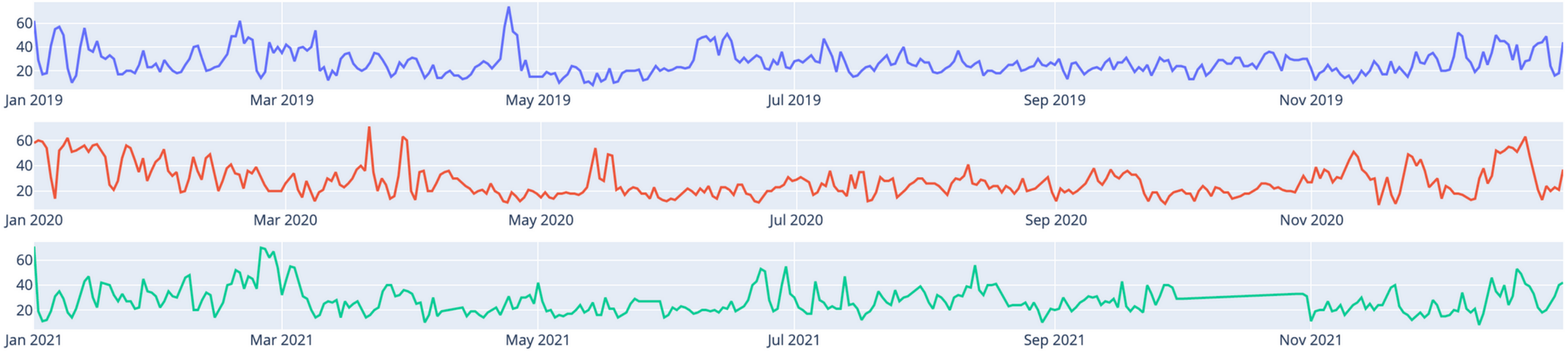
Italy

Trend of main pollution indicators in Italy



PM10

Rome

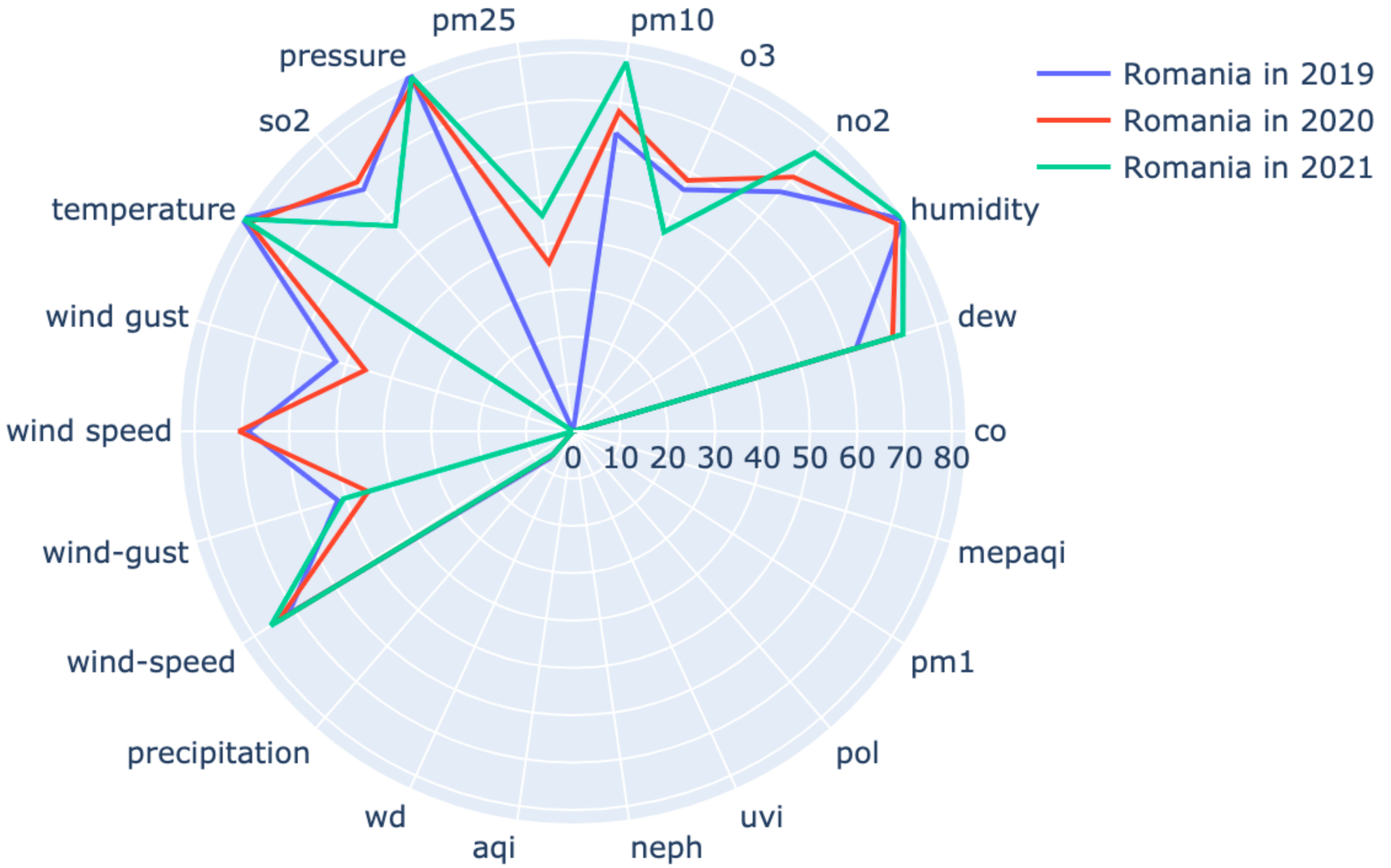


Year	Day count where Pm10>50µg/m³	rate of exceeding legal limits	Pm10 Annual Average	Variance of Pm10
Rome 2019	11	2.96	26.67	107.92
Rome 2020	31	8.03	27.48	142.5
Rome 2021	15	4.26	27.78	116.52

RO

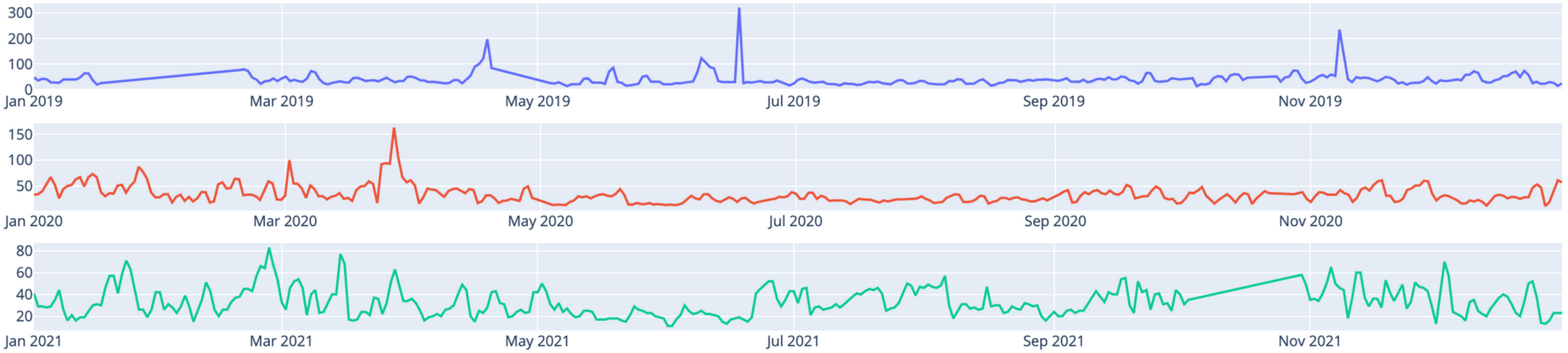
Romania

Trend of main pollution indicators in Romania



PM10

Bucharest

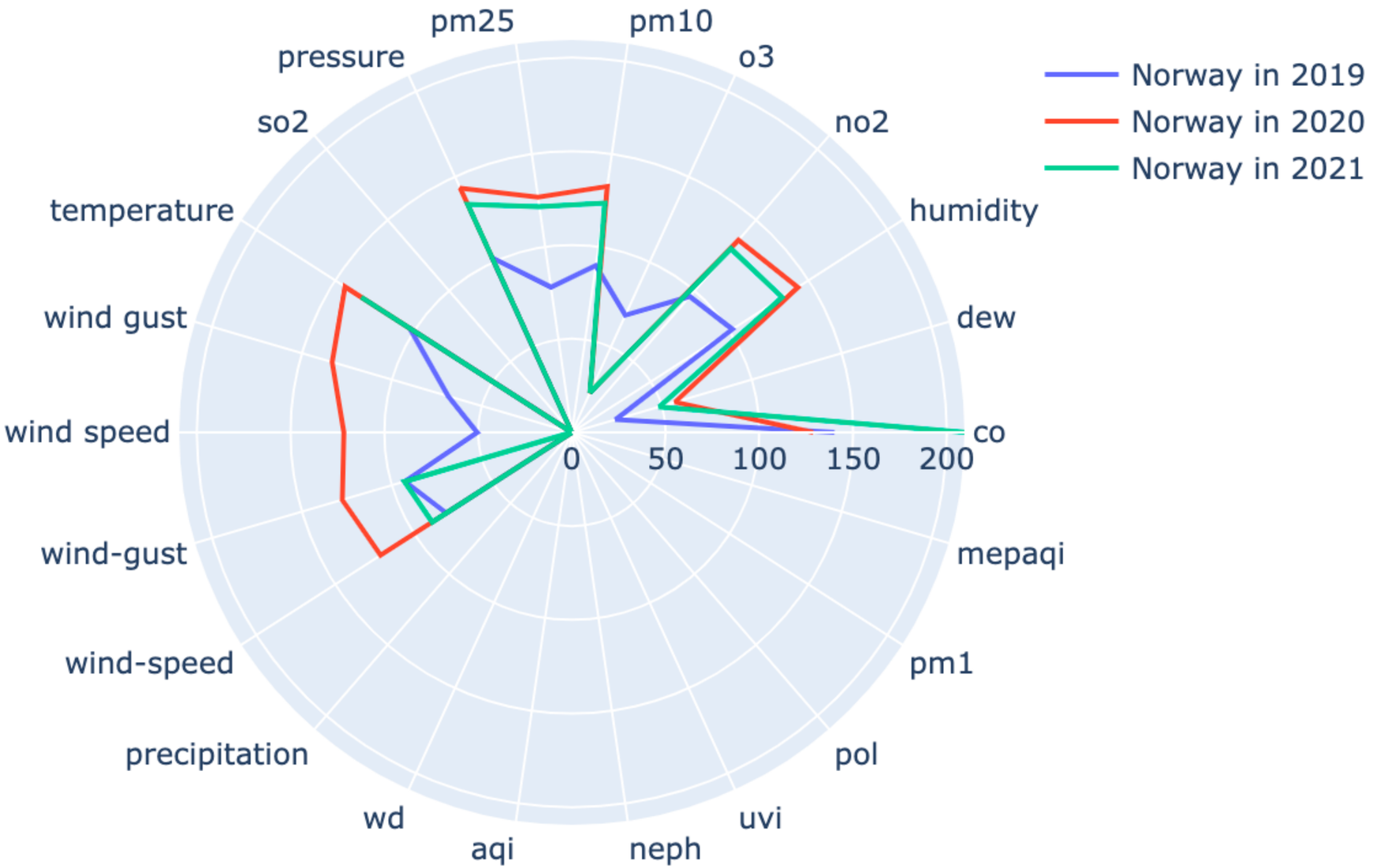


Year	Day count where Pm10>50µg/m³	rate of exceeding legal limits	Pm10 Annual Average	Variance of Pm10
Bucharest 2019	55	16.87	40.63	740.84
Bucharest 2020	50	13.09	33.26	272.66
Bucharest 2021	34	9.44	33.23	162.36

NO

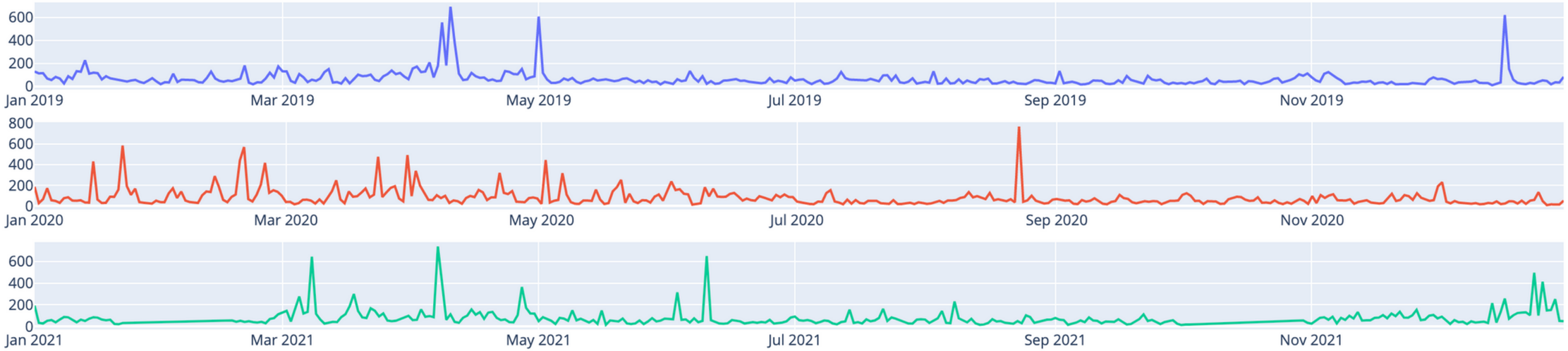
Norway

Trend of main pollution indicators in Norway



PM10

Oslo



Year	Day count where Pm10>50µg/m³	rate of exceeding legal limits	Pm10 Annual Average	Variance of Pm10
Oslo 2019	175	47.04	64.09	4925.81
Oslo 2020	220	56.85	82.57	8074.88
Oslo 2021	212	63.1	79.28	6761.17

A bit of code to make the tables

```
countries = ['IT', 'RO', 'NO' ]


fig = make_subplots(
    rows=3, cols=1,
    shared_xaxes=True,
    vertical_spacing=0.03,
    specs=[[{"type": "table"}],
           [{"type": "table"}],
           [{"type": "table"}]],
    subplot_titles=("2019", "2020", "2021")
    )

# 2019
fig.add_trace(go.Table( columnwidth = [10,20,20, 20,20,20,20,20,20,20,20,20,20,20,20,20,20,20],
                 header=dict(values=['<b>Country</b>', '<b>Air quality index</b>', '<b>Carbon Monoxide</b>',
                                     '<b>Humidity</b>', '<b>Nitrogen Dioxide</b>', '<b>Ozone</b>', '<b>Airborne particulates Matter</b>',
                                     '<b>Fine particulate matter</b>', '<b>Sulfur dioxide</b>', '<b>Temperature</b>']),
                cells=dict(values=[
                     [country for country in countries],
                     [round(value,2) for value in [res[(res['Specie'] == 'aqi') & (res['Country'] == country) & (res['Year'] == '2019')]['count'] for country in countries]],
                     [round(value,2) for value in [res[(res['Specie'] == 'co') & (res['Country'] == country) & (res['Year'] == '2019')]['count'] for country in countries]],
                     [round(value,2) for value in [res[(res['Specie'] == 'humidity') & (res['Country'] == country) & (res['Year'] == '2019')]['count'] for country in countries]],
                     [round(value,2) for value in [res[(res['Specie'] == 'no2') & (res['Country'] == country) & (res['Year'] == '2019')]['count'] for country in countries]],
                     [round(value,2) for value in [res[(res['Specie'] == 'o3') & (res['Country'] == country) & (res['Year'] == '2019')]['count'] for country in countries]],
                     [round(value,2) for value in [res[(res['Specie'] == 'pm10') & (res['Country'] == country) & (res['Year'] == '2019')]['count'] for country in countries]],
                     [round(value,2) for value in [res[(res['Specie'] == 'pm25') & (res['Country'] == country) & (res['Year'] == '2019')]['count'] for country in countries]],
                     [round(value,2) for value in [res[(res['Specie'] == 'so2') & (res['Country'] == country) & (res['Year'] == '2019')]['count'] for country in countries]],
                     [round(value,2) for value in [res[(res['Specie'] == 'temperature') & (res['Country'] == country) & (res['Year'] == '2019')]['count'] for country in countries]
                                ]
                           )
                      ),
                row = 1, col = 1
                )

# 2020
fig.add_trace(go.Table( columnwidth = [10,20,20, 20,20,20,20,20,20,20,20,20,20,20,20,20,20,20],
                 header=dict(values=['<b>Country</b>', '<b>Air quality index</b>', '<b>Carbon Monoxide</b>',
                                     '<b>Humidity</b>', '<b>Nitrogen Dioxide</b>', '<b>Ozone</b>', '<b>Airborne particulates Matter</b>',
```

A bit of code to make the radars

```
species = ['co', 'dew', 'humidity', 'no2', 'o3', 'pm10', 'pm25', 'pressure',
           'so2', 'temperature', 'wind gust', 'wind speed', 'wind-gust',
           'wind-speed', 'precipitation', 'wd', 'aqi', 'neph', 'uvi', 'pol',
           'pm1', 'mepaqi']

fig = go.Figure()

fig.add_trace(go.Scatterpolar(
      r=[value for value in res.loc[(res['Country'] == 'NO') & (res['Year'] == '2019')]['count']],
      theta=species,
      name='Norway in 2019'
))
fig.add_trace(go.Scatterpolar(
      r=[value for value in res.loc[(res['Country'] == 'NO') & (res['Year'] == '2020')]['count']],
      theta=species,
      name='Norway in 2020'
))
fig.add_trace(go.Scatterpolar(
      r=[value for value in res.loc[(res['Country'] == 'NO') & (res['Year'] == '2021')]['count']],
      theta=species,
      name='Norway in 2021'
))


fig.update_layout(
  width=600,
    height=500,
  polar=dict(
    radialaxis=dict(
```

Last but not least... Let's make a Dashboard

Put some of our graphs in a Dashboard

1 Open the project form the following github repository:

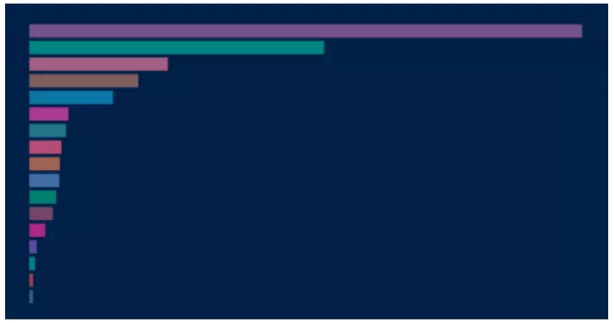
2 Open your terminal.

3 Run '**make run**'.

4 Open this link: https://github.com/elenaabcc/CovidAnalysis4CPDM

SITOGRAPHY

Source of data used for the analysis



Coronavirus (COVID-19) Vaccinations

Our vaccination dataset uses the most recent official numbers from governments and health ministries worldwide. Population estimates for per-capita metrics are based on the United...

Our World in Data

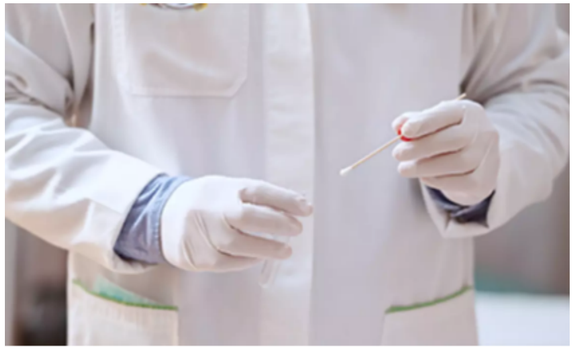
Data on COVID-19 vaccination in the EU/EEA

Data in various file formats with information on COVID-19 vaccine doses administered

Data on hospital and ICU admission rates and current occupancy for COVID-19

Data in multiple file formats with information about hospitalisation and Intensive Care Unit (ICU) admission rates and current occupancy for COVID-19, sorted by date and...

European Centre for Disease Prevention and Control / Feb 10



Data on testing for COVID-19 by week and country

Data in multiple file formats with information about testing volume for COVID-19, sorted by week and country, and

COVID-19 Worldwide Air Quality data

How polluted is the air today? Check out the real-time air pollution map, for more than 100 countries.

aqicn.org / The World Air Quality Index project