In [50]:
```python
# Import necessary libraries for data analysis, visualization, and install the fredapi package
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
```

In [51]:
```python
# Import Fred from fredapi
from fredapi import Fred
```

Import Fred API Key

In [52]:
```python
fred_key = ('1c1f54581ea0001cf7fb832af9e3dd6e')
```

Create Fred Object

In [53]:
```python
fred = Fred(api_key=fred_key)
```

Search For S&P 500 Data

In [54]:
```python
# Search the FRED database for data series related to the S&P index, sorted by popularity
sp_search = fred.search("S&P", order_by='popularity')
```

In [67]:
```python
# Inspect the results
sp_search.head(5)
```

Out[67]:

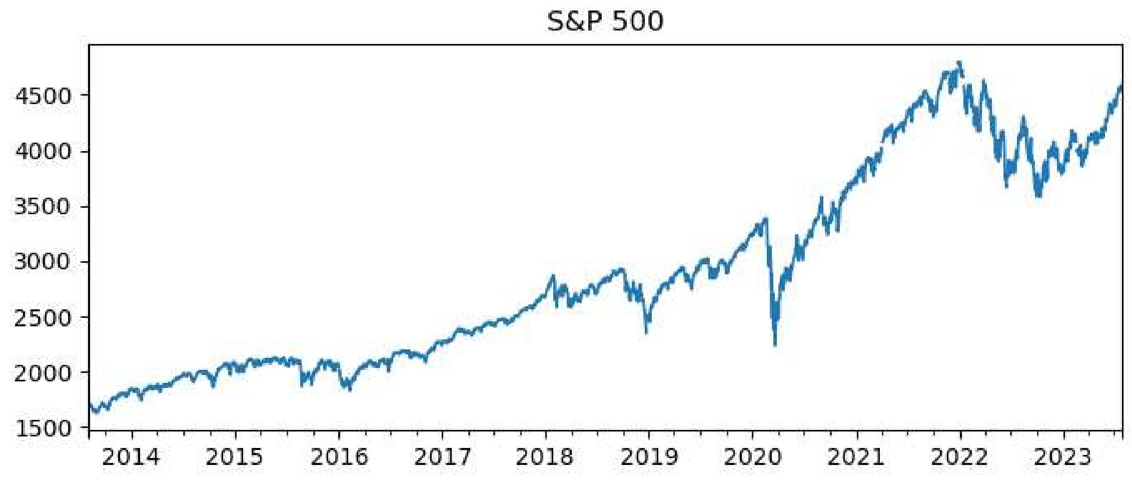
	id	realtime_start	realtime_end	title	observation_start	observation_end	freq
series id							
BAMLH0A0HYM2	BAMLH0A0HYM2	2023-07-31	2023-07-31	ICE BofA US High Yield Index Option-Adjusted S...	1996-12-31	2023-07-28	
CSUSHPINSA	CSUSHPINSA	2023-07-31	2023-07-31	S&P/Case-Shiller U.S. National Home Price Index	1987-01-01	2023-05-01	M
				ICE BofA US High			

Pull Raw Data

In [56]:
```python
# Retrieve data for the S&P 500 index from the FRED database
sp500 = fred.get_series(series_id='SP500')
```

In [57]:
```
# Create a line plot of the S&P 500 data with specified size, title, and line width
sp500.plot(figsize=(8,3), title='S&P 500', lw=1.25)
```

Out[57]:
```
<Axes: title={'center': 'S&P 500'}>
```



Pull And Join Multiple Data Series

In [58]:
```
# Search the FRED database for data series related to unemployment
uemp_results = fred.search('unemployment rate state')
uemp_results.head(5)
```

Out[58]:

	id	realtime_start	realtime_end	title	observation_start	observation_end	frequency	freq
series id								
UNRATE	UNRATE	2023-07-31	2023-07-31	Unemployment Rate	1948-01-01	2023-06-01	Monthly	
UNRATENSA	UNRATENSA	2023-07-31	2023-07-31	Unemployment Rate	1948-01-01	2023-06-01	Monthly	
CCSA	CCSA	2023-07-31	2023-07-31	Continued Claims (Insured Unemployment)	1967-01-07	2023-07-15	Weekly, Ending Saturday	
LNS14000006	LNS14000006	2023-07-31	2023-07-31	Unemployment Rate - Black or African American	1972-01-01	2023-06-01	Monthly	
UNEMPLOY	UNEMPLOY	2023-07-31	2023-07-31	Unemployment Level	1948-01-01	2023-06-01	Monthly	

In [59]:
```
# Retrieve data for the unemployment rate using get_series function
unrate = fred.get_series('UNRATE')
```

In [60]:
```
# Filter the search down to monthly frequency
uemp_df= fred.search('unemployment rate by state',filter=('frequency','Monthly'))

# Filter again to include only seasonally adjusted data in % units
uemp_df= uemp_df.query('seasonal_adjustment == "Seasonally Adjusted" and units=="Percent"')

# Further filter the search results to only include data series with titles containing 'Unemployment
# To get data for all the states.
uemp_df=uemp_df.loc[uemp_df['title'].str.contains('Unemployment Rate in')]
```

In [61]:
```
# Data with all the states
uemp_df.head(5)
```

Out[61]:

	id	realtime_start	realtime_end	title	observation_start	observation_end	frequency	frequency_short
series id								
CAUR	CAUR	2023-07-31	2023-07-31	Unemployment Rate in California	1976-01-01	2023-06-01	Monthly	M
TXUR	TXUR	2023-07-31	2023-07-31	Unemployment Rate in Texas	1976-01-01	2023-06-01	Monthly	M
FLUR	FLUR	2023-07-31	2023-07-31	Unemployment Rate in Florida	1976-01-01	2023-06-01	Monthly	M
NYUR	NYUR	2023-07-31	2023-07-31	Unemployment Rate in New York	1976-01-01	2023-06-01	Monthly	M
MAUR	MAUR	2023-07-31	2023-07-31	Unemployment Rate in Massachusetts	1976-01-01	2023-06-01	Monthly	M

In [62]:
```
# Retrieve data for each series in uemp_df and store the results in a list of DataFrames
all_results=[]
for myid in uemp_df.index:
    results=fred.get_series(myid)
    results= results.to_frame(name=myid)
    all_results.append(results)

# Drop Unnecessary Columns
uemp_results = pd.concat(all_results,axis=1).drop(columns=['LASMT391746000000003','LASMT261982000000
uemp_results.head(5)
```

Out[62]:

	CAUR	TXUR	FLUR	NYUR	MAUR	OHUR	ALUR	NJUR	MIUR	AKUR	...	MTUR	NEUR	PRUR	MSUR	MEUR
1976-01-01	9.2	5.8	9.7	10.3	10.5	8.1	6.6	10.3	9.9	7.1	...	5.8	3.3	19.6	6.7	8.7
1976-02-01	9.2	5.8	9.7	10.3	10.5	8.1	6.6	10.3	9.9	7.1	...	5.8	3.3	19.5	6.7	8.7
1976-03-01	9.1	5.9	9.6	10.2	10.5	8.1	6.6	10.3	9.9	7.0	...	5.8	3.3	19.3	6.6	8.6
1976-04-01	9.1	5.9	9.5	10.2	10.3	8.0	6.5	10.3	9.8	6.9	...	5.8	3.2	19.0	6.4	8.6
1976-05-01	9.0	5.9	9.3	10.1	10.1	7.8	6.4	10.3	9.6	6.9	...	5.8	3.1	18.9	6.3	8.5

5 rows × 52 columns

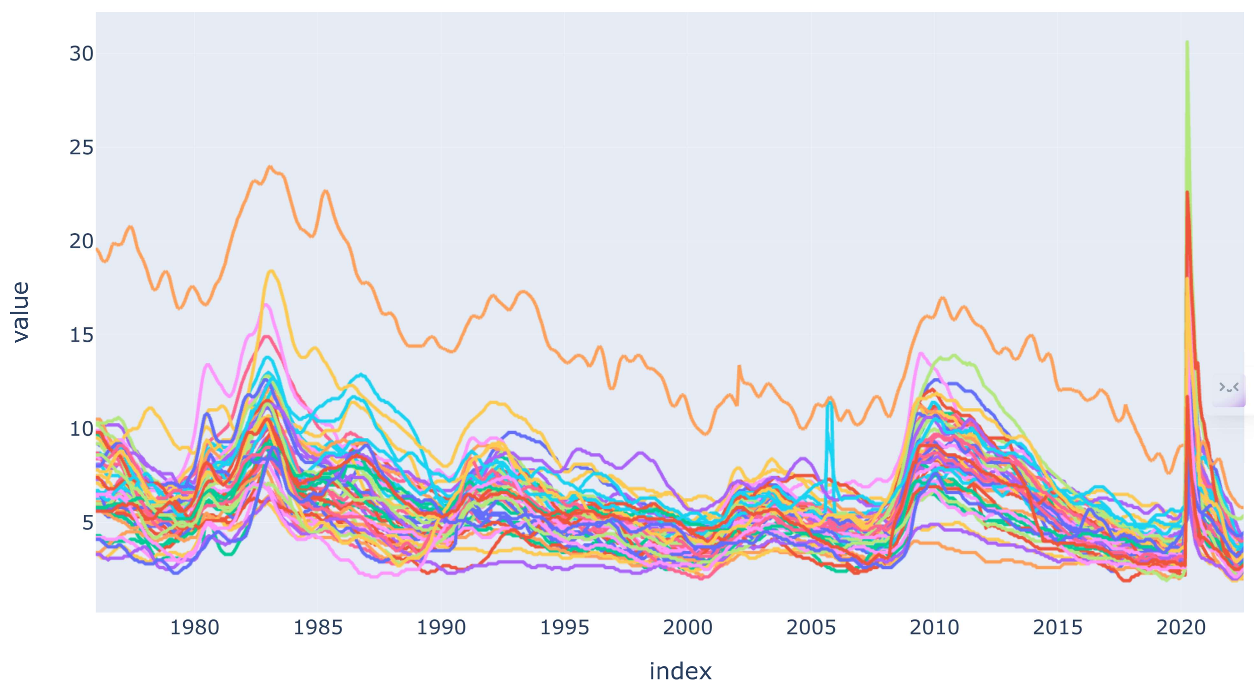
In [63]:
```
# Create a dictionary mapping series IDs to state names by removing 'Unemployment Rate in' from the
id_to_state = uemp_df['title'].str.replace('Unemployment Rate in','').to_dict()
```

In [64]:
```
# Rename the columns of the uemp_results DataFrame using the id_to_state dictionary
uemp_results.columns = [id_to_state[c] for c in uemp_results.columns]
```

Plot States Unemployment Rate

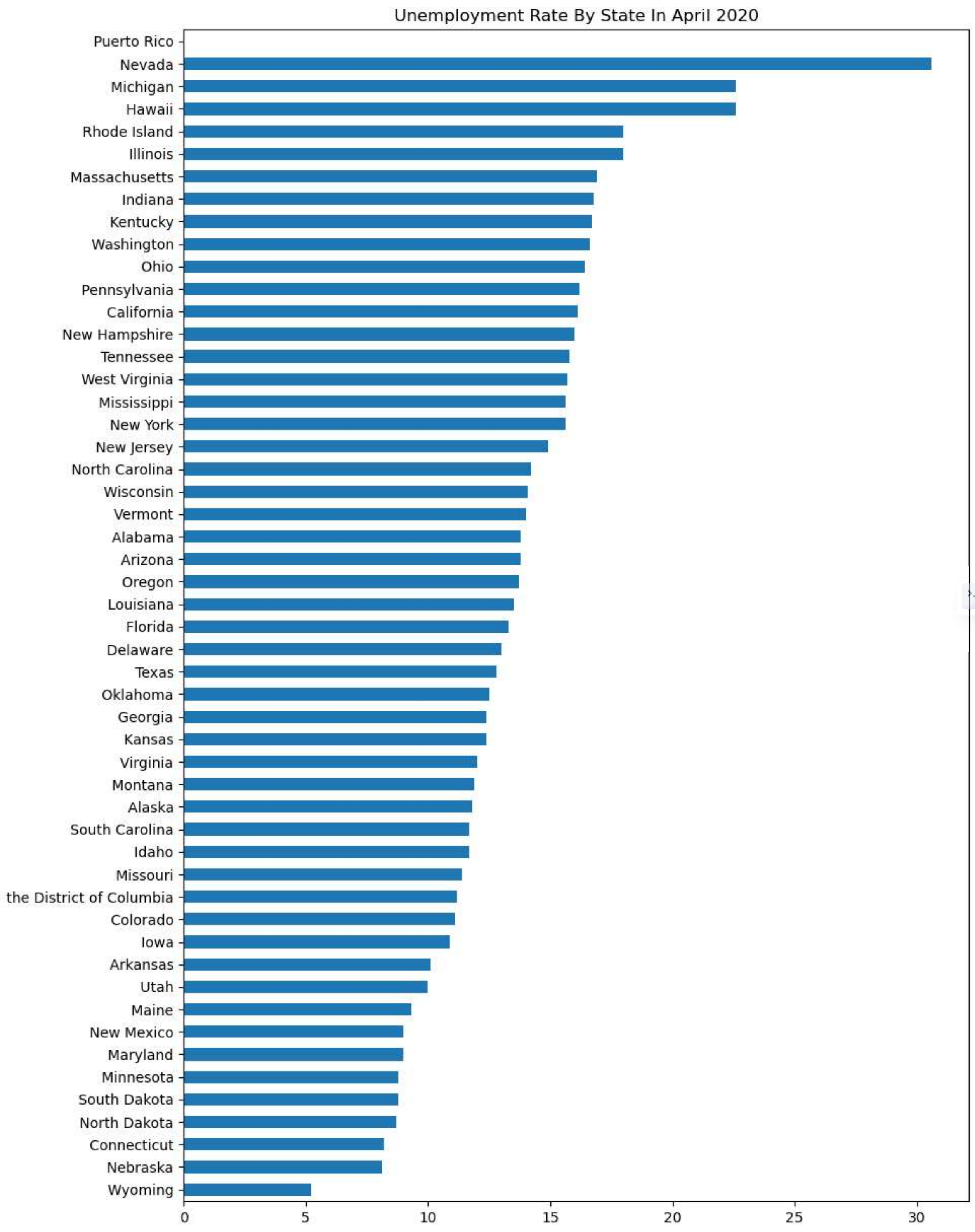
In [65]:
```
px.line(uemp_results, width=1000, height=500)

# It can be seen that there has been a massive spike in unemployment rate during April 2020 during t
# Now let's analyse the unemployment rate in April 2020 for all states.
```



Pull April 2020 Unemployment Rate By State

In [84]:
```
# Create a horizontal bar chart of unemployment rates by state for April 2020
graph = uemp_results.loc[uemp_results.index=='2020-04-01'].transpose().sort_values('2020-04-01').plo
figsize=(10,15),width=.55,title="Unemployment Rate By State In April 2020")
graph.legend().remove()
```

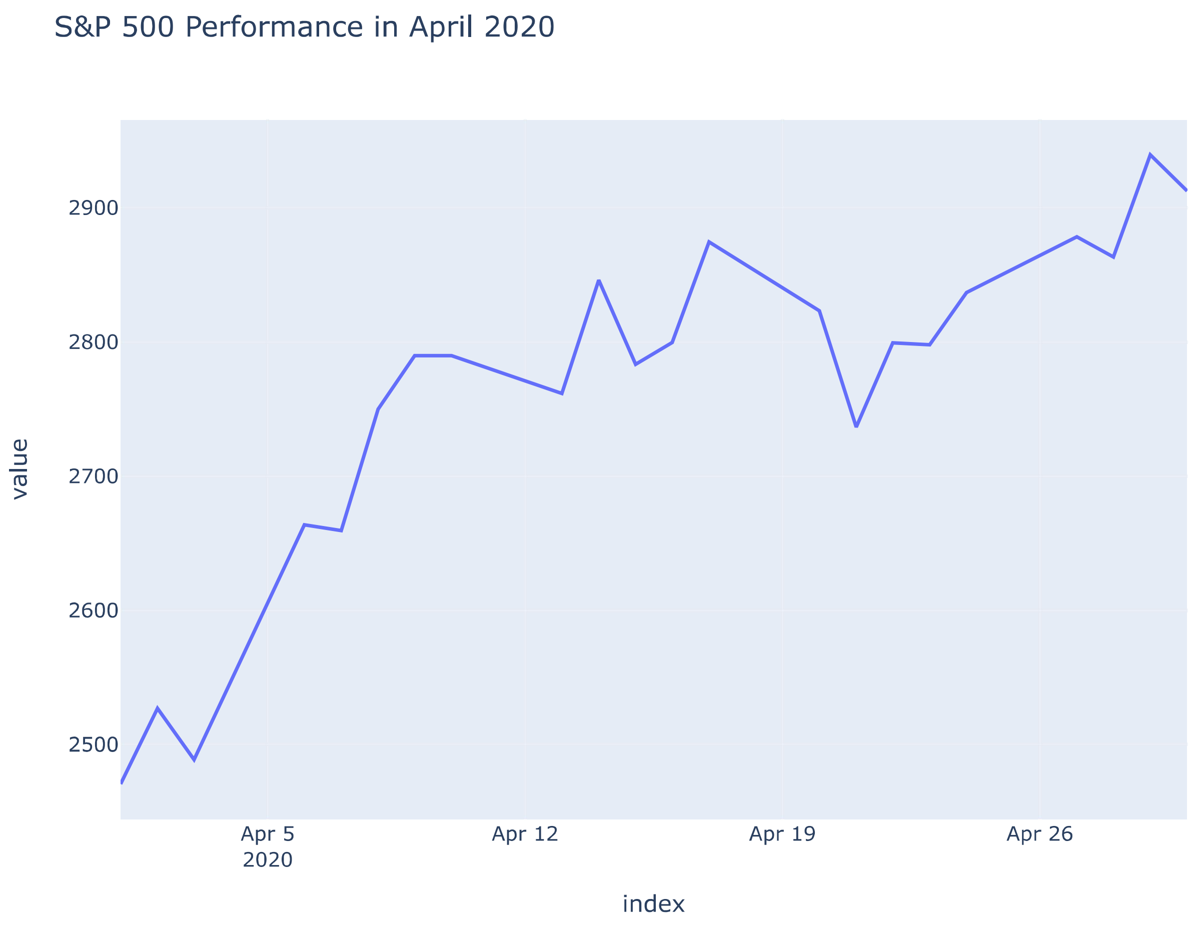


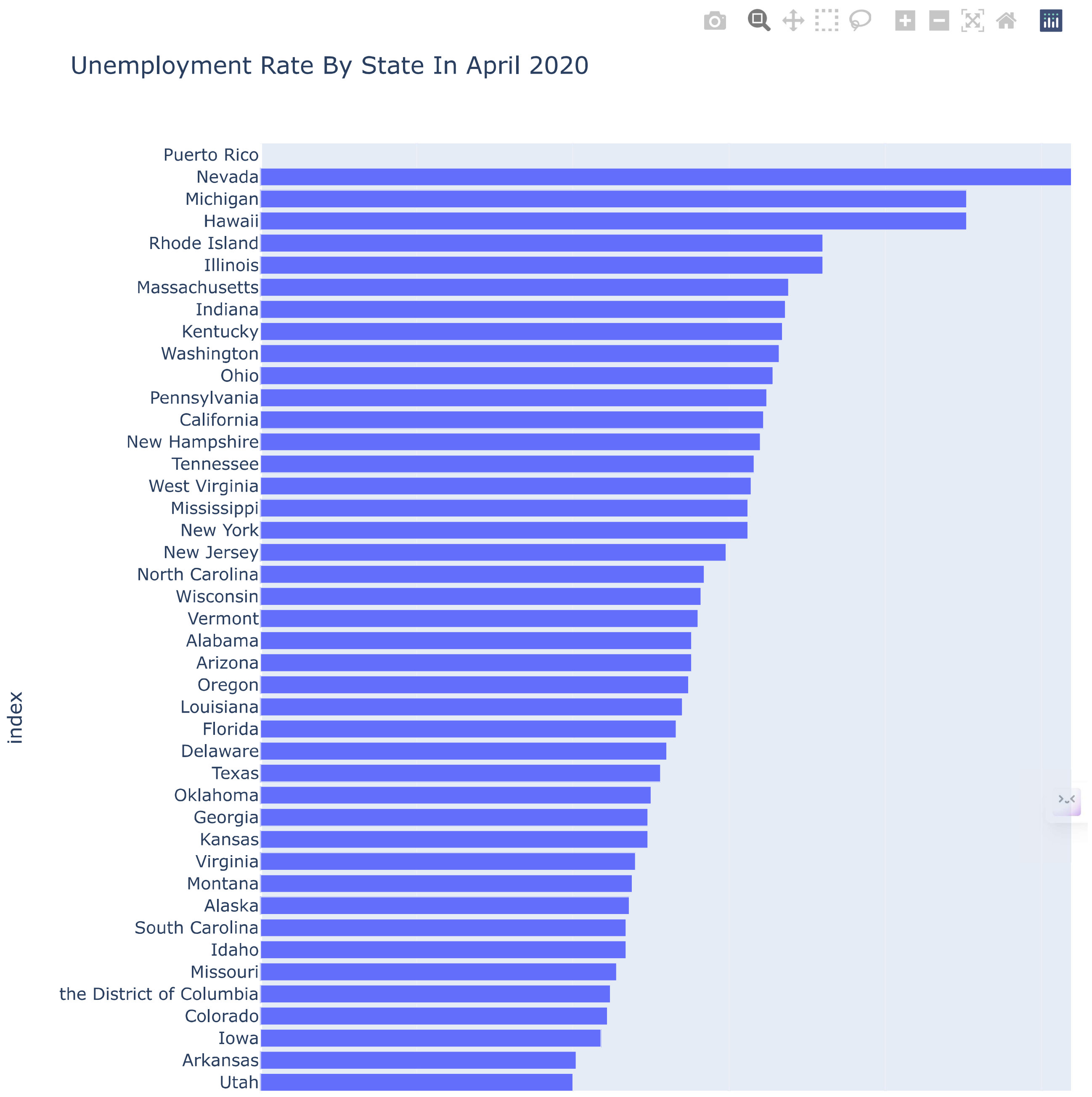
Create Subplots Using plotly.express

```
In [91]: # Fill missing values in the sp500 Series using forward fill

         # Create a line chart of the S&P 500 performance in April 2020
         sp500_filled = sp500.fillna(method='ffill')
         fig = px.line(sp500_filled.loc['2020-04'], title='S&P 500 Performance in April 2020')
         fig.update_layout(showlegend=False)
         fig.update_layout(width=800, height=600)
         fig.show()


         # Filter and sort the unemployment data for April 2020 and create a horizontal bar chart
         uemp_april = uemp_results.loc[uemp_results.index=='2020-04-01'].transpose().sort_values('2020-04-01'
         fig = px.bar(uemp_april, x='2020-04-01', y=uemp_april.index, title='Unemployment Rate By State In Ap
         fig.update_layout(showlegend=False, height=1000)
         fig.show()
```




Unemployment Rate By State In April 2020
index
Puerto Rico
Nevada
Michigan
Hawaii
Rhode Island
Illinois
Massachusetts
Indiana
Kentucky
Washington
Ohio
Pennsylvania
California
New Hampshire
Tennessee
West Virginia
Mississippi
New York
New Jersey
North Carolina
Wisconsin
Vermont
Alabama
Arizona
Oregon
Louisiana
Florida
Delaware
Texas
Oklahoma
Georgia
Kansas
Virginia
Montana
Alaska
South Carolina
Idaho
Missouri
the District of Columbia
Colorado
Iowa
Arkansas
Utah