Unemployment Data Set with FRED & Pandas

Viewing 2020 unemployment rate per state in the US (during Covid-19)

- Grabing the unemployment data from FRED
- Renaming the states names to a more understandable values \ acutal state name
- Visuallizing the data per state

installing the fred api

[ ]:
```
!pip install fredapi >/dev/null
```

Pandas Options

[2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
import seaborn as sns
from fredapi import Fred
import time

# In order for the dataset to work for others,
# they need a fred api key from their site
from kaggle_secrets import UserSecretsClient
user_secrets = UserSecretsClient()
fred_key = user_secrets.get_secret("fred_key")
```

Fred Key

[3]:
```
fred = Fred(api_key=fred_key)
```

Unemployment Data

- Searching for an Unemployment Data using the fred api

- Searching the UNRATE id and title
- Cleaning the data
- See unemployment rate in every state

[4]:
```
unemp_results = fred.search('unemployment')
unemp_results
```

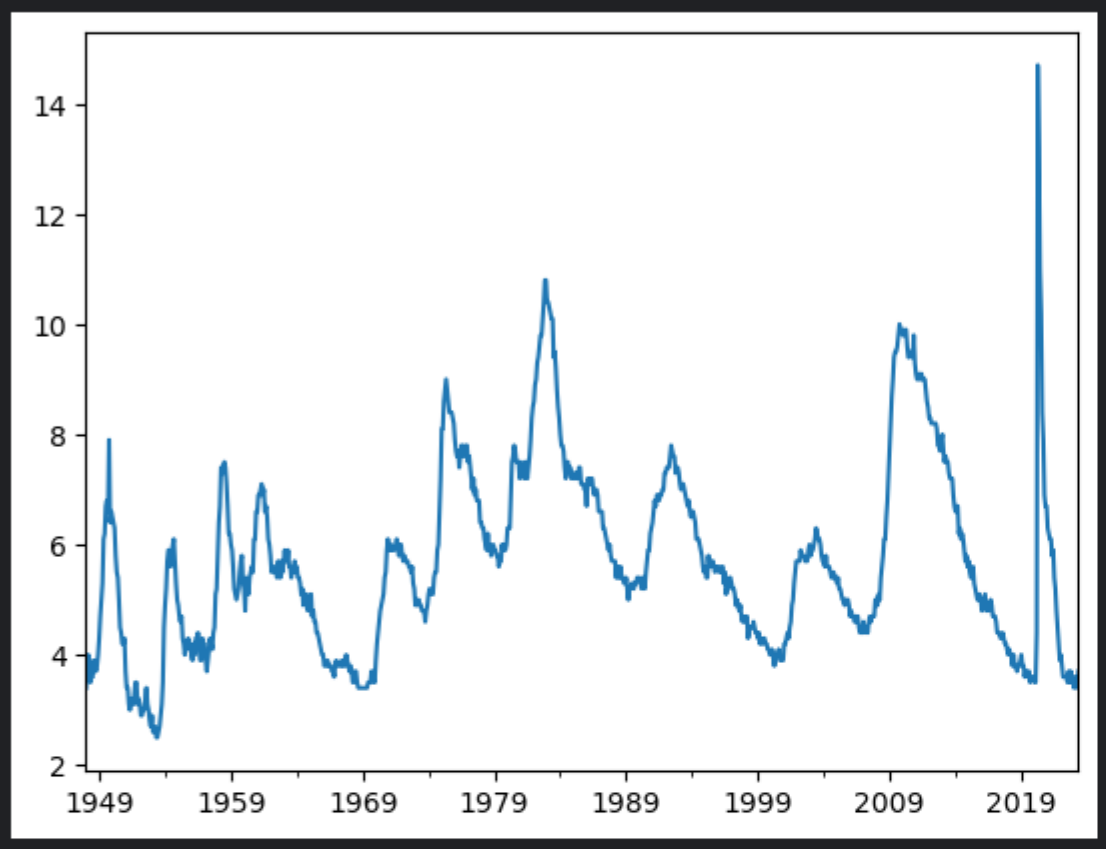
[4]:

series id	id	realtime_start	realtime_end	title	observation_start	observation_end	frequency	frequency_short	units	units_short	seasonal_adjus
UNRATE	UNRATE	2023-06-11	2023-06-11	Unemployment Rate	1948-01-01	2023-05-01	Monthly	M	Percent	%	Seasonally Ac
UNRATENSA	UNRATENSA	2023-06-11	2023-06-11	Unemployment Rate	1948-01-01	2023-05-01	Monthly	M	Percent	%	Not Sea: Ac
CCSA	CCSA	2023-06-11	2023-06-11	Continued Claims (Insured Unemployment)	1967-01-07	2023-05-27	Weekly, Ending Saturday	W	Number	Number	Seasonally Ac
CCNSA	CCNSA	2023-06-11	2023-06-11	Continued Claims (Insured Unemployment)	1967-01-07	2023-05-27	Weekly, Ending Saturday	W	Number	Number	Not Sea: Ac
NROU	NROU	2023-06-11	2023-06-11	Noncyclical Rate of Unemployment	1949-01-01	2033-10-01	Quarterly	Q	Percent	%	Not Sea: Ac
...	...	...	...	...	...	...	...	...	...	...	
TXTRAV3URN	TXTRAV3URN	2023-06-11	2023-06-11	Unemployment Rate in Travis County, TX	1990-01-01	2023-04-01	Monthly	M	Percent	%	Not Sea: Ac
FLBAYC5URN	FLBAYC5URN	2023-06-11	2023-06-11	Unemployment Rate in Bay County, FL	1990-01-01	2023-04-01	Monthly	M	Percent	%	Not Sea: Ac
MIGENE9URN	MIGENE9URN	2023-06-11	2023-06-11	Unemployment Rate in Genesee County, MI	1990-01-01	2023-04-01	Monthly	M	Percent	%	Not Sea: Ac
MAWORC7URN	MAWORC7URN	2023-06-11	2023-06-11	Unemployment Rate in Worcester County, MA	1990-01-01	2023-04-01	Monthly	M	Percent	%	Not Sea: Ac
TXTARR9URN	TXTARR9URN	2023-06-11	2023-06-11	Unemployment Rate in Tarrant County, TX	1990-01-01	2023-04-01	Monthly	M	Percent	%	Not Sea: Ac

[ ]:
```
unrate = fred.get_series ('UNRATE')
# unrate
```

[6]:
```
unrate.plot()
```

[6]: <Axes: >



Per State

[7]:
```
unrate = fred.search('unemployment state', filter=('frequency', 'Monthly'))
unrate = unrate.query('seasonal_adjustment == "Seasonally Adjusted" and units == "Percent"')
unrate = unrate.loc[unrate['title'].str.contains('Unemployment Rate in')]
```

Sorting the unemployment rate into a list by dates

[8]:
```
all_results = []
```

[9]:
```
for myid in unrate.index:
    results = fred.get_series(myid)
    results = results.to_frame(name=myid)
    all_results.append(results)
    #time.sleep(0.2) # Don't request to fast and get blocked
unrate_results = pd.concat(all_results, axis=1)
```

[10]:
```
cols_to_drop = []
```

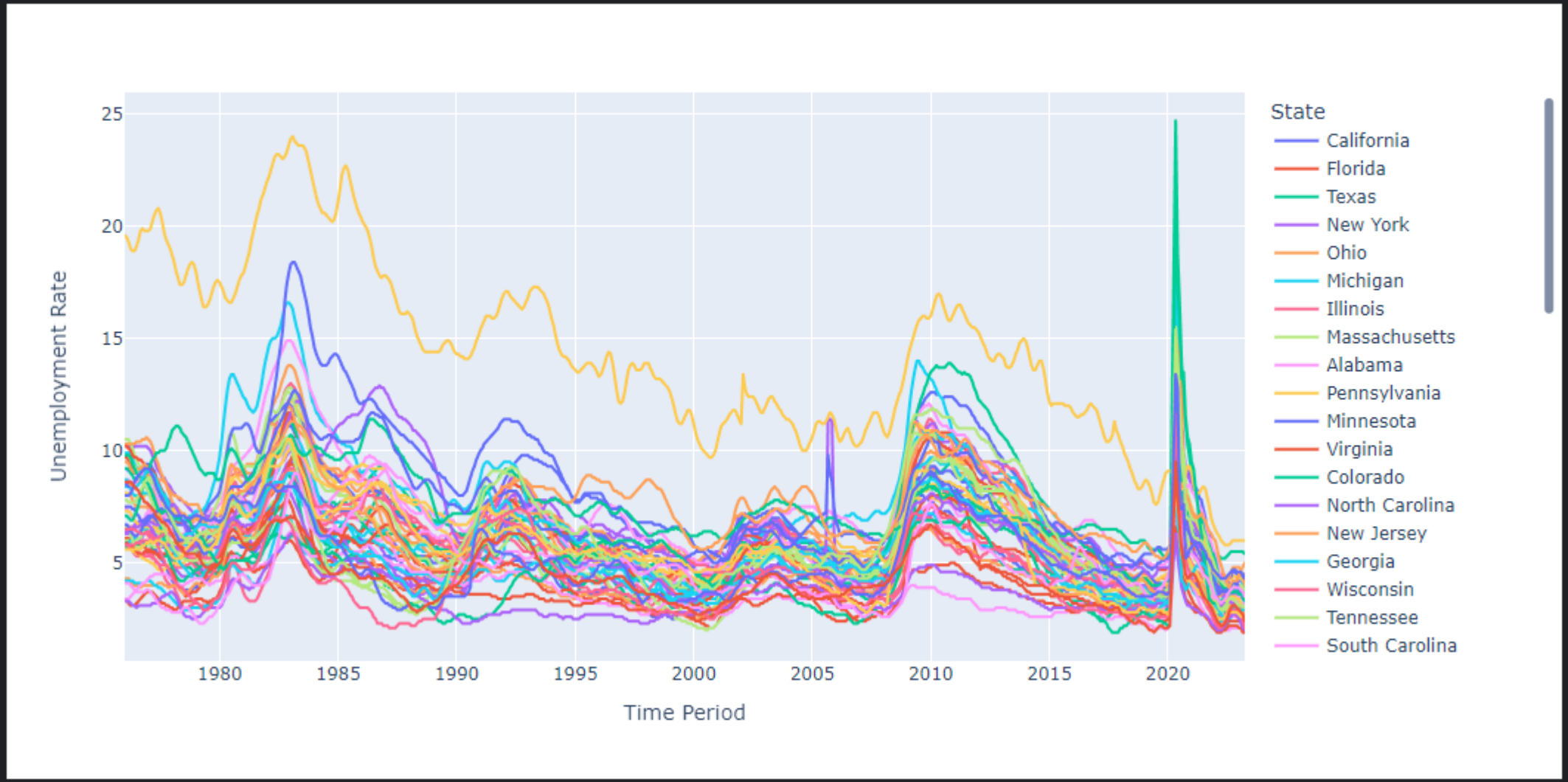
[11]:
```
for i in unrate_results:
    if len(i) > 4:
        cols_to_drop.append(i)
unrate_results = unrate_results.drop(columns = cols_to_drop, axis=1)
```

[12]:
```
unemp_states = unrate_results.copy() #.drop('UNRATE', axis=1)
unemp_states = unemp_states.dropna()
id_to_state = unrate['title'].str.replace('Unemployment Rate in ', '').to_dict()
unemp_states.columns = [id_to_state[c] for c in unemp_states.columns]
unemp_states.columns
```

[12]:
```
Index(['California', 'Florida', 'Texas', 'New York', 'Ohio', 'Michigan',
       'Illinois', 'Massachusetts', 'Alabama', 'Pennsylvania', 'Minnesota',
       'Virginia', 'Colorado', 'North Carolina', 'New Jersey', 'Georgia',
       'Wisconsin', 'Tennessee', 'South Carolina', 'Kentucky', 'Washington',
       'Arizona', 'Nevada', 'Louisiana', 'Oregon', 'Missouri', 'Oklahoma',
       'Indiana', 'New Mexico', 'Arkansas', 'West Virginia', 'Utah', 'Alaska',
       'Maryland', 'Iowa', 'Kansas', 'Montana', 'Connecticut', 'North Dakota',
       'Puerto Rico', 'Mississippi', 'South Dakota', 'Hawaii', 'Nebraska',
       'the District of Columbia', 'Maine', 'New Hampshire', 'Rhode Island',
       'Wyoming', 'Idaho', 'Delaware', 'Vermont'],
      dtype='object')
```

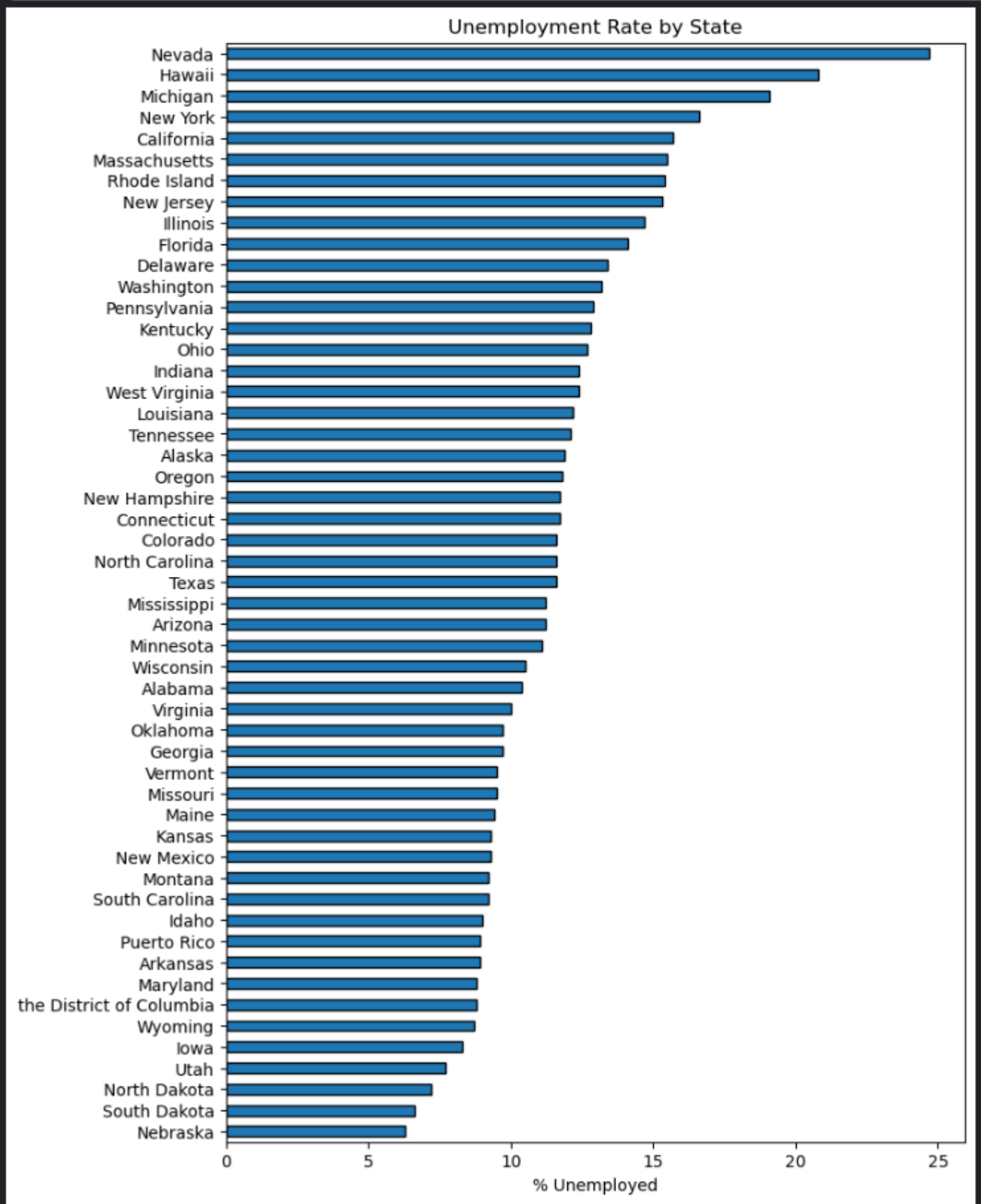
[13]:

```
px.line(unemp_states, labels={'variable':'State','index':'Time Period','value':'Unemployment Rate'})
```



April 2020 Unemployment Rate Per State

```
ax = unemp_states.loc[unemp_states.index == '2020-05-01'].T.sort_values('2020-05-01')/
.plot(kind='barh', figsize=(8, 12), width=0.5, edgecolor='black', title= 'Unemployment Rate by State')
ax.legend().remove()
ax.set_xlabel('% Unemployed')
plt.show()
```



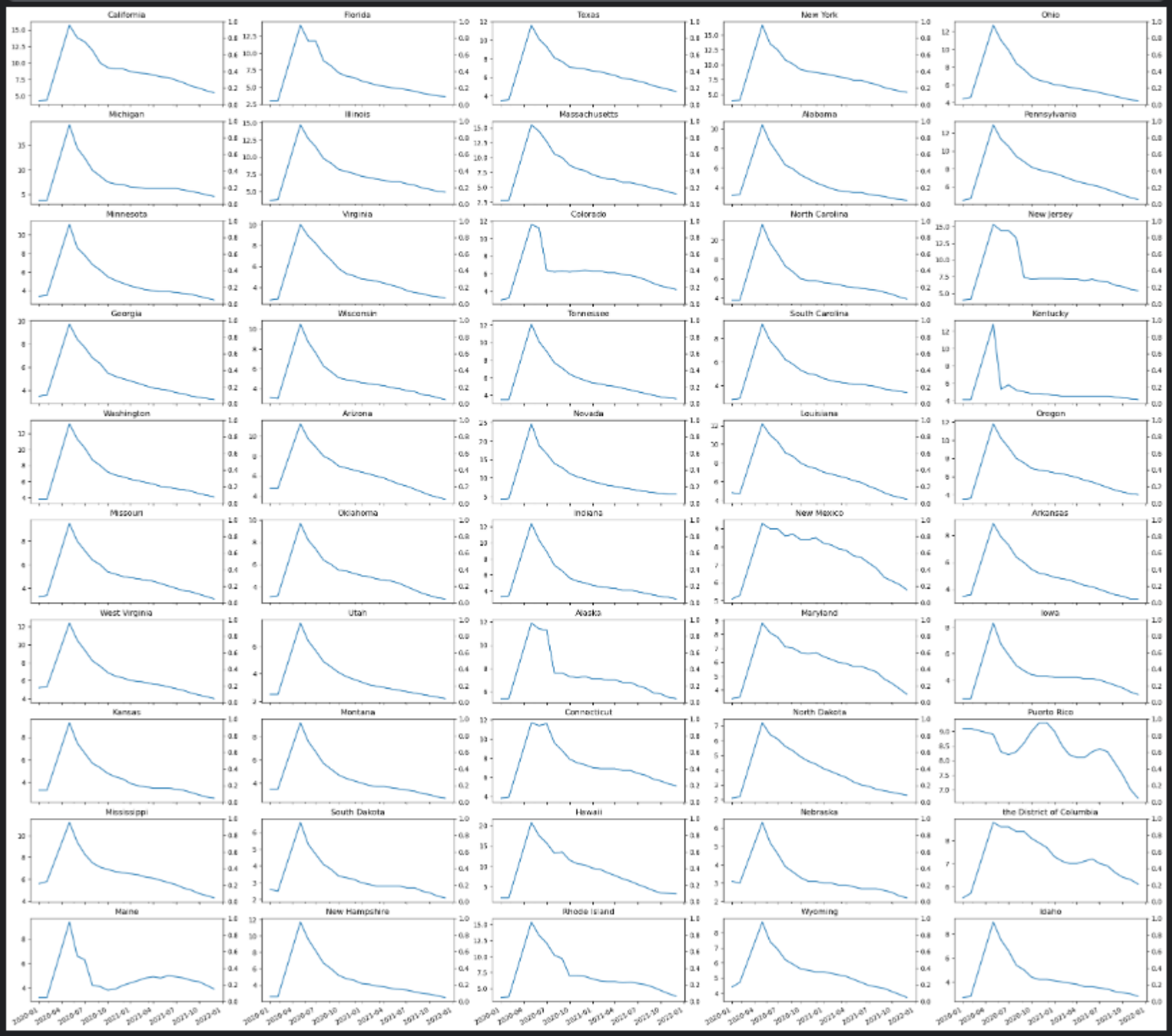
+ Code + Markdown

Unemployment Rate by State

- Going through the states and visuallizing them

```
fig, axs = plt.subplots(10, 5, figsize=(30, 30), sharex=True)
axs = axs.flatten()

i = 0
for state in unemp_states.columns:
    ax2 = axs[i].twinx()
    unemp_states.query('index >= 2020 and index < 2022')[state].plot(ax=axs[i], label='Unemployment')
    ax2.grid(False)
    axs[i].set_title(state)
    i += 1
plt.tight_layout()
plt.show()
```



+ Code + Markdown