Economic Data Analysis Project via Federal Reserve Economic Data (FRED)

Esse projeto foi inspirado no vídeo do canal: https://www.youtube.com/watch?v=R67XuYc9NQ4&t=14s, com dados da plataforma americana FRED (Federal Reserve Economic Data - https://fred.stlouisfed.org/) e tem como objetivo praticar coletar, limpar e explorar dados econômicos de forma eficiente. Através de técnicas de raspagem de dados, limpeza e exploração, o projeto busca fornecer insights valiosos sobre indicadores econômicos. Os dados serão coletados via API do próprio site.

1. Importando as bibliotecas que iremos usar:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
from fredapi import Fred
import time


# Estilo dos gráficos:

plt.style.use('fivethirtyeight')
pd.set_option("display.max_columns", 500)
color_pal = plt.rcParams["axes.prop_cycle"].by_key()["color"]
```

2. Baixando os dados iniciais:

```
!pip install fredapi >/dev/null

# >/dev/null para não aparecer todo o texto de instalação!


# Inserindo a api coletada do site:

fred_key = '48c84796306d7592190e31e2b816f182' # Solcite a sua api pelo site no informativo do projeto


# Criando um objeto FRED:
fred = Fred(api_key=fred_key)


# Fazendo a busca:

fred.search('S&P')
```

	id	realtime_start	realtime_end	title	observation_start	observation_end	frequency
series id							
CSUSHPINSA	CSUSHPINSA	2024-12-27	2024-12-27	S&P CoreLogic Case-Shiller U.S. National Home ...	1987-01-01	2024-09-01	Monthly
CSUSHPISA	CSUSHPISA	2024-12-27	2024-12-27	S&P CoreLogic Case-Shiller U.S. National Home ...	1987-01-01	2024-09-01	Monthly
SP500	SP500	2024-12-27	2024-12-27	S&P 500	2014-12-29	2024-12-26	Daily, Close
SPCS20RSA	SPCS20RSA	2024-12-27	2024-12-27	S&P CoreLogic Case-Shiller 20-City Composite H...	2000-01-01	2024-09-01	Monthly
SPCS20RNSA	SPCS20RNSA	2024-12-27	2024-12-27	S&P CoreLogic Case-Shiller 20-City Composite H...	2000-01-01	2024-09-01	Monthly
...	...	...	...	...	...	...	...
CRDQARAPABIS	CRDQARAPABIS	2024-12-27	2024-12-27	Total Credit to Private Non-Financial Sector, ...	1940-04-01	2024-04-01	Quarterly, End of Quarter
QARPAMUSDA	QARPAMUSDA	2024-12-27	2024-12-27	Total Credit to Private Non-Financial Sector, ...	1990-01-01	2024-04-01	Quarterly, End of Quarter
QCOPAMXDCA	QCOPAMXDCA	2024-12-27	2024-12-27	Total Credit to Private Non-Financial Sector, ...	1996-10-01	2024-04-01	Quarterly, End of Quarter
QCOPAMUSDA	QCOPAMUSDA	2024-12-27	2024-12-27	Total Credit to Private Non-Financial Sector, ...	1996-10-01	2024-04-01	Quarterly, End of Quarter
M0854AUSM498NNBR	M0854AUSM498NNBR	2024-12-27	2024-12-27	Labor Turnover, Total Separation Rate, Manufac...	1919-01-01	1930-12-01	Monthly

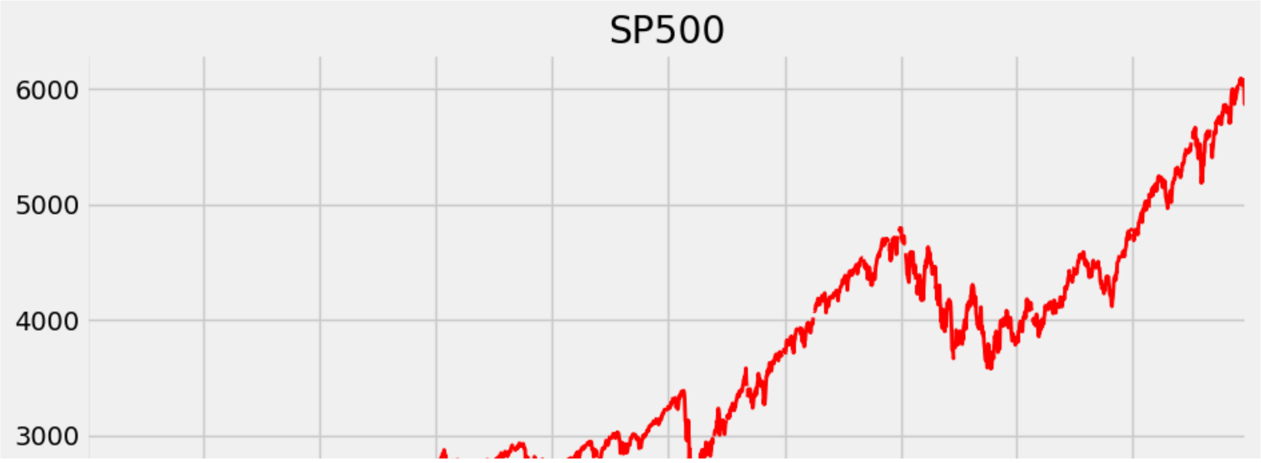
1000 rows × 15 columns

```
# Ordenando por popularidade:

fred.search('S&P', order_by='popularity')
```

	id	realtime_start	realtime_end	title	observation_start	observation_end	freque
series id							
BAMLH0A0HYM2	BAMLH0A0HYM2	2024-12-27	2024-12-27	ICE BofA US High Yield Index Option-Adjusted S...	1996-12-31	2024-12-26	D Cl
CSUSHPINSA	CSUSHPINSA	2024-12-27	2024-12-27	S&P CoreLogic Case-Shiller U.S. National Home ...	1987-01-01	2024-09-01	Mon
SP500	SP500	2024-12-27	2024-12-27	S&P 500	2014-12-29	2024-12-26	D Cl
BAMLH0A0HYM2EY	BAMLH0A0HYM2EY	2024-12-27	2024-12-27	ICE BofA US High Yield Index Effective Yield	1996-12-31	2024-12-26	D Cl
BAMLC0A0CM	BAMLC0A0CM	2024-12-27	2024-12-27	ICE BofA US Corporate Index Option-Adjusted Sp...	1996-12-31	2024-12-26	D Cl
...	...	...	...	...	...	...	
DDDI12SMA156NWDB	DDDI12SMA156NWDB	2024-12-27	2024-12-27	Private Credit by Deposit Money Banks and Othe...	2001-01-01	2020-01-01	Anr
Q03069USQ605NNBR	Q03069USQ605NNBR	2024-12-27	2024-12-27	Revenue Freight Tons Originated, Less Than Car...	1920-01-01	1943-07-01	Quart
CSHICPCZA156NRUG	CSHICPCZA156NRUG	2024-12-27	2024-12-27	Share of Gross Capital Formation at Current Pu...	1990-01-01	2019-01-01	Anr
CSHXCPVEA156NRUG	CSHXCPVEA156NRUG	2024-12-27	2024-12-27	Share of Merchandise Exports at Current Purcha...	1950-01-01	2019-01-01	Anr
CSHMCPNPA156NRUG	CSHMCPNPA156NRUG	2024-12-27	2024-12-27	Share of Merchandise Imports at Current Purcha...	1960-01-01	2019-01-01	Anr

1000 rows × 15 columns

```
# Ordenando a busca por popularidade e atribuindo a uma variável:

sp_search = fred.search('S&P', order_by='popularity')


# Baixando esses dados para a memória do nosso notebook:

sp500 = fred.get_series(series_id = 'SP500')
sp500.plot(figsize = (10, 5), title = 'SP500', lw = 2, color = 'r')
plt.plot()
```



O gráfico apresenta a evolução do índice S&P 500 (O S&P 500 é um índice que mede o desempenho de 500 grandes empresas listadas nas bolsas dos EUA, sendo usado como referência da economia americana e dos mercados de ações globais) entre 2015 e 2024, mostrando um crescimento constante com destaque para a queda abrupta em 2020, causada pela pandemia de COVID-19 possivelmente, seguida de uma rápida recuperação e contínua valorização até alcançar novos picos em 2024. Esse comportamento reflete a resiliência do mercado financeiro americano, impulsionado por políticas de estímulo econômico e avanços tecnológicos.

3. Juntando a outros data-sets:

```
# Levantando os data frames das taxas de desemprego nos EUA:

unemp_df = fred.search('unemployment rate state', filter=('frequency','Monthly'))
unemp_df
```

	id	realtime_start	realtime_end	title	observation_start	observation_end	frequency	frequency_
series id								
UNRATE	UNRATE	2024-12-27	2024-12-27	Unemployment Rate	1948-01-01	2024-11-01	Monthly	
UNRATENSA	UNRATENSA	2024-12-27	2024-12-27	Unemployment Rate	1948-01-01	2024-11-01	Monthly	
LNS14000006	LNS14000006	2024-12-27	2024-12-27	Unemployment Rate - Black or African American	1972-01-01	2024-11-01	Monthly	
UNEMPLOY	UNEMPLOY	2024-12-27	2024-12-27	Unemployment Level	1948-01-01	2024-11-01	Monthly	
LNU03000000	LNU03000000	2024-12-27	2024-12-27	Unemployment Level	1948-01-01	2024-11-01	Monthly	
...	...	...	...	...	...	...	...	
INBART5URN	INBART5URN	2024-12-27	2024-12-27	Unemployment Rate in Bartholomew County, IN	1990-01-01	2024-10-01	Monthly	
MOPRURN	MOPRURN	2024-12-27	2024-12-27	Unemployment Rate in Perry County, MO	1990-01-01	2024-10-01	Monthly	
PENS812UR	PENS812UR	2024-12-27	2024-12-27	Unemployment Rate in Pensacola-Ferry Pass-Bren...	1990-01-01	2024-10-01	Monthly	
AKRO439UR	AKRO439UR	2024-12-27	2024-12-27	Unemployment Rate in Akron, OH (MSA)	1990-01-01	2024-10-01	Monthly	
PORT912UR	PORT912UR	2024-12-27	2024-12-27	Unemployment Rate in Port St. Lucie, FL (MSA)	1990-01-01	2024-10-01	Monthly	

1000 rows × 15 columns

```
# Filtrando os que são mais interessantes para o nosso caso

unemp_df = fred.search('unemployment rate state', filter=('frequency','Monthly'))
unemp_df = unemp_df.query('seasonal_adjustment == "Seasonally Adjusted" and units == "Percent"')
unemp_df
```

	id	realtime_start	realtime_end	title	observation_start	observation_end	fre
series id							
UNRATE	UNRATE	2024-12-27	2024-12-27	Unemployment Rate	1948-01-01	2024-11-01	N
LNS14000006	LNS14000006	2024-12-27	2024-12-27	Unemployment Rate - Black or African American	1972-01-01	2024-11-01	N
U6RATE	U6RATE	2024-12-27	2024-12-27	Total Unemployed, Plus All Persons Marginally ...	1994-01-01	2024-11-01	N
CAUR	CAUR	2024-12-27	2024-12-27	Unemployment Rate in California	1976-01-01	2024-11-01	N
TXUR	TXUR	2024-12-27	2024-12-27	Unemployment Rate in Texas	1976-01-01	2024-11-01	N
...	...	...	...	...	...	...	
M08311USM156SNBR	M08311USM156SNBR	2024-12-27	2024-12-27	Unemployment Rate, Married Males, Spouse Prese...	1954-11-01	1968-01-01	N
LNS13008397	LNS13008397	2024-12-27	2024-12-27	Of Total Unemployed, Percent Unemployed Less T...	1948-01-01	2024-11-01	N
LBSSA20	LBSSA20	2024-12-27	2024-12-27	Labor Force Participation Rate for Kansas	1976-01-01	2024-11-01	N
LNS14000150	LNS14000150	2024-12-27	2024-12-27	Unemployment Rate - Married Men	1955-01-01	2024-11-01	N
NEIPTERM156SFRBRIC	NEIPTERM156SFRBRIC	2024-12-27	2024-12-27	Hornstein-Kudlyak-Lange Non-Employment Index i...	1994-01-01	2024-11-01	N

158 rows × 15 columns

```
# Pegando apenas aqueles que falam de 'Unemployment Rate' na coluna title e colocando na lista com todos os resultados:

all_results = []
for myid in unemp_df.index:
    results = fred.get_series(myid)
    results = results.to_frame(name=myid)
    all_results.append(results)
    time.sleep(0.1) # Não requisitar muito rápido e ser bloqueado! (limite de 500 estabelecido na primeira linha)
uemp_results = pd.concat(all_results, axis=1) # Concatena todos juntos
```

Clique duas vezes (ou pressione "Enter") para editar

4. Verrificando a existencia de nulos:

```
# Fazendo o check dos valores nulos

uemp_results.isna()
```

	UNRATE	LNS14000006	U6RATE	CAUR	TXUR	M0892AUSM156SNBR	LNS14000024	LNS14000031	M0892BUSM156SNBR	NYUR	U2RATE	FLUR	LI
1929-04-01	True	True	True	True	True	False	True	True	True	True	True	True	
1929-05-01	True	True	True	True	True	False	True	True	True	True	True	True	
1929-06-01	True	True	True	True	True	False	True	True	True	True	True	True	
1929-07-01	True	True	True	True	True	False	True	True	True	True	True	True	
1929-08-01	True	True	True	True	True	False	True	True	True	True	True	True	
...	...	...	...	...	...	...	...	...	...	...	...	...	
2024-07-01	False	False	False	False	False	True	False	False	True	False	False	False	
2024-08-01	False	False	False	False	False	True	False	False	True	False	False	False	
2024-09-01	False	False	False	False	False	True	False	False	True	False	False	False	
2024-10-01	False	False	False	False	False	True	False	False	True	False	False	False	
2024-11-01	False	False	False	False	False	True	False	False	True	False	False	False	

1136 rows × 158 columns

```
uemp_results.isna().sum()
# Vemos que temos muitos valores nulos
```

	0
UNRATE	213
LNS14000006	501
U6RATE	765
CAUR	549
TXUR	549
...	...
M08311USM156SNBR	977
LNS13008397	213
LBSSA20	549
LNS14000150	297
NEIPTERM156SFRBRIC	765

158 rows × 1 columns

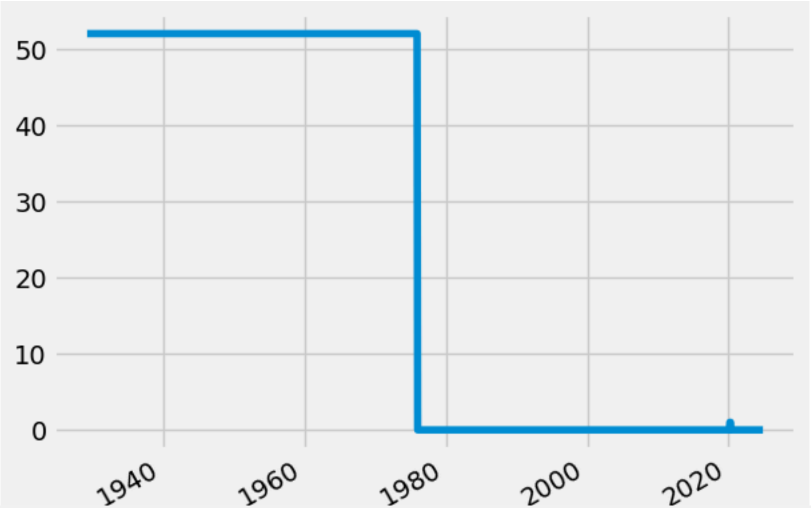
```
cols_to_drop = []
for i in uemp_results:
    if len(i) > 4:
        cols_to_drop.append(i)
uemp_results = uemp_results.drop(columns = cols_to_drop, axis=1)


# Vamos dropar UNRANTE, pois não é um estado:

uemp_states = uemp_results.copy()#.drop('UNRATE', axis=1)


uemp_results
```

	CAUR	TXUR	NYUR	FLUR	PAUR	OHUR	MIUR	ALUR	MAUR	NCUR	ILUR	NJUR	GAUR	COUR	AZUR	WIUR	VAUR	KYUR	SCUR	NMUR	MNUR	TN
1929-04-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
1929-05-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
1929-06-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
1929-07-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
1929-08-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
2024-07-01	5.2	4.1	4.3	3.3	3.4	4.5	4.4	2.8	3.5	3.7	5.2	4.7	3.4	3.9	3.4	3.0	2.7	4.7	3.9	4.0	3.2	
2024-08-01	5.3	4.1	4.4	3.3	3.4	4.5	4.5	2.8	3.7	3.8	5.3	4.8	3.6	4.0	3.4	2.9	2.8	4.8	4.3	4.1	3.3	
2024-09-01	5.3	4.1	4.4	3.3	3.4	4.5	4.5	2.9	3.8	3.8	5.3	4.7	3.6	4.0	3.5	2.9	2.9	4.9	4.5	4.2	3.4	
2024-10-01	5.4	4.1	4.4	3.3	3.4	4.3	4.6	2.9	3.9	3.7	5.3	4.7	3.6	4.1	3.6	2.9	2.9	5.0	4.7	4.3	3.4	
2024-11-01	5.4	4.2	4.4	3.4	3.5	4.3	4.8	3.1	4.0	3.7	5.3	4.6	3.7	4.3	3.7	2.9	3.0	5.1	4.8	4.4	3.5	

1136 rows × 52 columns

```
# Verificando os NA values:

uemp_results.isna().sum(axis = 1).plot()
```

```
<Axes: >
```



Em meados de 1980 foi quando se iniciou o fornecimento de dados mensais, por isso podemos apagar sem problemas os dados vazios!

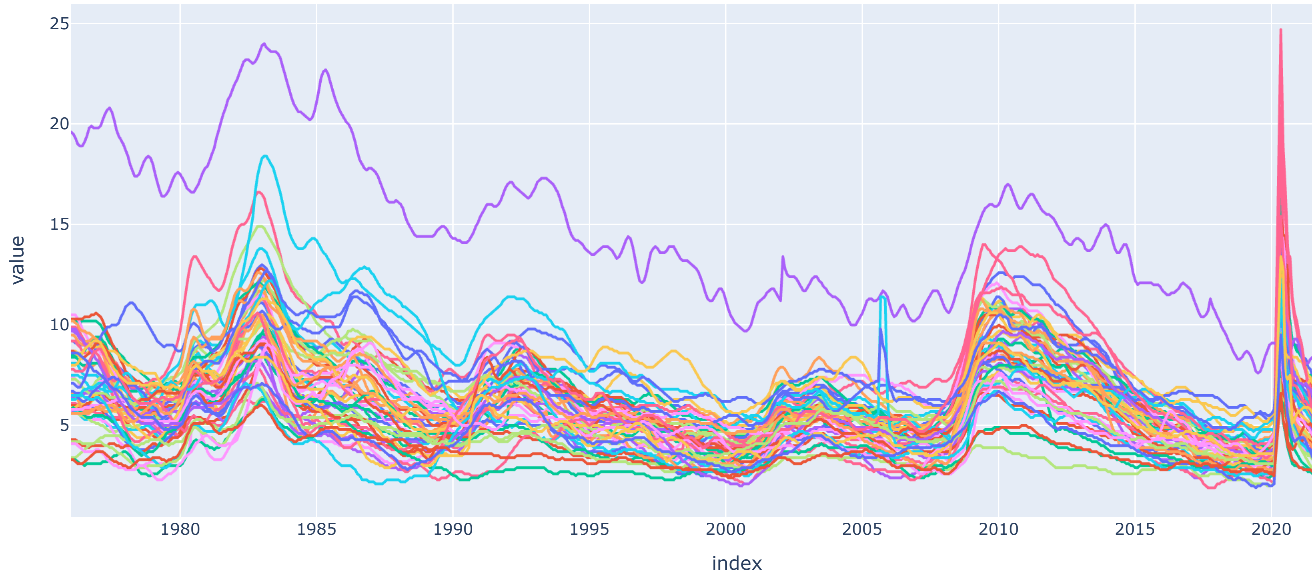
```
# Tirando os valores NA:
uemp_states = uemp_results.copy()
uemp_states = uemp_states.dropna()
id_to_state = unemp_df['title'].str.replace('Unemployment Rate in ','').to_dict()
uemp_states.columns = [id_to_state[c] for c in uemp_states.columns]
```

```
uemp_states
```

	California	Texas	New York	Florida	Pennsylvania	Ohio	Michigan	Alabama	Massachusetts	North Carolina	Illinois	New Jersey	Georgia
1976-01-01	9.2	5.8	10.3	9.7	8.0	8.1	9.9	6.6	10.5	6.4	6.6	10.3	8.4
1976-02-01	9.2	5.8	10.3	9.7	8.1	8.1	9.9	6.6	10.5	6.4	6.6	10.3	8.4
1976-03-01	9.1	5.9	10.2	9.6	8.1	8.1	9.9	6.6	10.5	6.4	6.6	10.3	8.4
1976-04-01	9.1	5.9	10.2	9.5	8.1	8.0	9.8	6.5	10.3	6.3	6.6	10.3	8.3
1976-05-01	9.0	5.9	10.1	9.3	8.1	7.8	9.6	6.4	10.1	6.1	6.6	10.3	8.1
...	...	...	...	...	...	...	...	...	...	...	...	...	...
2024-07-01	5.2	4.1	4.3	3.3	3.4	4.5	4.4	2.8	3.5	3.7	5.2	4.7	3.4
2024-08-01	5.3	4.1	4.4	3.3	3.4	4.5	4.5	2.8	3.7	3.8	5.3	4.8	3.6
2024-09-01	5.3	4.1	4.4	3.3	3.4	4.5	4.5	2.9	3.8	3.8	5.3	4.7	3.6
2024-10-01	5.4	4.1	4.4	3.3	3.4	4.3	4.6	2.9	3.9	3.7	5.3	4.7	3.6
2024-11-01	5.4	4.2	4.4	3.4	3.5	4.3	4.8	3.1	4.0	3.7	5.3	4.6	3.7

585 rows × 52 columns

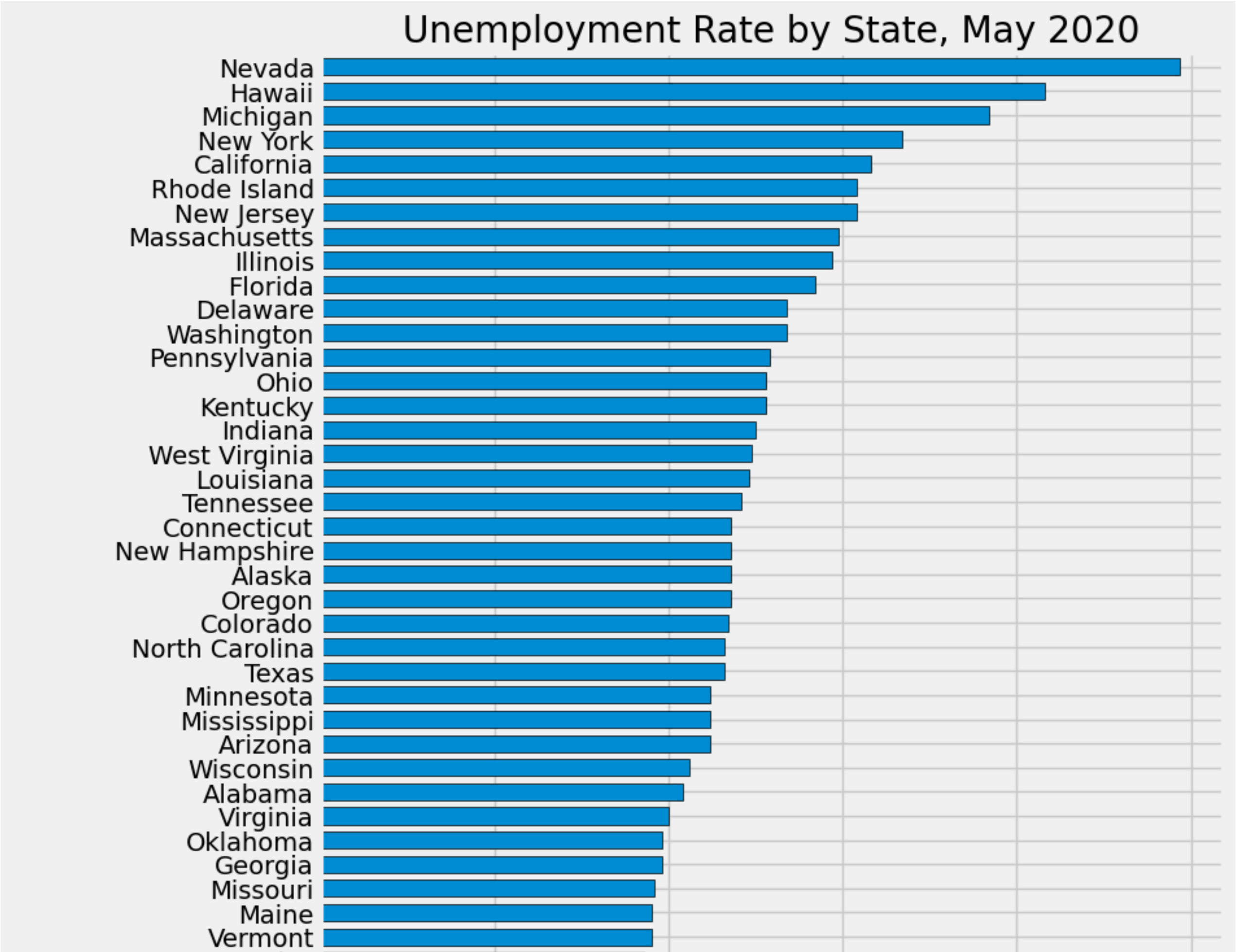
```
# Plotando o indice de desemprego por estado:

px.line(uemp_states)
```



5. Maio 2020

Vemos que em maio de 2020 há um pico gigantesco, vamos avaliar mais de perto!

```
ax = uemp_states.loc[uemp_states.index == '2020-05-01'].T \
    .sort_values('2020-05-01') \
    .plot(kind='barh', figsize=(8, 12), width=0.7, edgecolor='black',
          title='Unemployment Rate by State, May 2020')
ax.legend().remove()
ax.set_xlabel('% Unemployed')
plt.show()
```



O gráfico mostra a taxa de desemprego nos estados dos EUA em maio de 2020, evidenciando um forte impacto econômico da pandemia de COVID-19. Estados como Nevada e Hawaii apresentam as maiores taxas, próximas de 25%, refletindo a dependência de setores como turismo, enquanto Nebraska, South Dakota e North Dakota registram as menores, abaixo de 8%, devido a economias mais estáveis e menos afetadas. Há uma disparidade significativa entre estados, com áreas urbanas e industriais, como Michigan e New York, também sofrendo altas taxas. Essas variações regionais destacam a necessidade de políticas econômicas direcionadas para apoiar os estados mais impactados.

6. Trazendo a paritcipação por estado:

Para investigar mais de perto essa questão vamos trazer uma planilha de "participation rate state" seguindo o mesmo modelo!

```
part_df = fred.search('participation rate state', filter=('frequency','Monthly'))
part_df = part_df.query('seasonal_adjustment == "Seasonally Adjusted" and units == "Percent"')
```

```
# Repare que estamos basicamente repetindo o que foi feito anteriormente:

part_id_to_state = part_df['title'].str.replace('Labor Force Participation Rate for ','').to_dict()

all_results = []

for myid in part_df.index:
    results = fred.get_series(myid)
    results = results.to_frame(name=myid)
    all_results.append(results)
    time.sleep(0.1) # Don't request to fast and get blocked
part_states = pd.concat(all_results, axis=1)
part_states.columns = [part_id_to_state[c] for c in part_states.columns]
```

7. Plorando Desemprego x Taxa de participação por estado

```
uemp_states
```

	California	Texas	New York	Florida	Pennsylvania	Ohio	Michigan	Alabama	Massachusetts	North Carolina	Illinois	New Jersey	Georgia	C
1976-01-01	9.2	5.8	10.3	9.7	8.0	8.1	9.9	6.6	10.5	6.4	6.6	10.3	8.4	
1976-02-01	9.2	5.8	10.3	9.7	8.1	8.1	9.9	6.6	10.5	6.4	6.6	10.3	8.4	
1976-03-01	9.1	5.9	10.2	9.6	8.1	8.1	9.9	6.6	10.5	6.4	6.6	10.3	8.4	
1976-04-01	9.1	5.9	10.2	9.5	8.1	8.0	9.8	6.5	10.3	6.3	6.6	10.3	8.3	
1976-05-01	9.0	5.9	10.1	9.3	8.1	7.8	9.6	6.4	10.1	6.1	6.6	10.3	8.1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
2024-07-01	5.2	4.1	4.3	3.3	3.4	4.5	4.4	2.8	3.5	3.7	5.2	4.7	3.4	
2024-08-01	5.3	4.1	4.4	3.3	3.4	4.5	4.5	2.8	3.7	3.8	5.3	4.8	3.6	
2024-09-01	5.3	4.1	4.4	3.3	3.4	4.5	4.5	2.9	3.8	3.8	5.3	4.7	3.6	
2024-10-01	5.4	4.1	4.4	3.3	3.4	4.3	4.6	2.9	3.9	3.7	5.3	4.7	3.6	
2024-11-01	5.4	4.2	4.4	3.4	3.5	4.3	4.8	3.1	4.0	3.7	5.3	4.6	3.7	

585 rows × 52 columns

```
# Repare que na coluna do estado Columbia, temos 'the District of Columbia', vamos mudar isso:

uemp_states = uemp_states.rename(columns={'the District of Columbia':'District Of Columbia'})
```

```
# Plotando multiplos gráficos para cada estado:

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

i = 0
for state in uemp_states.columns:
    if state in ["District Of Columbia","Puerto Rico"]:
        continue
    ax2 = axs[i].twinx()
    uemp_states.query('index >= 2020 and index < 2022')[state] \
        .plot(ax=axs[i], label='Unemployment')
    part_states.query('index >= 2020 and index < 2022')[state] \
        .plot(ax=ax2, label='Participation', color=color_pal[1])
    ax2.grid(False)
    axs[i].set_title(state)
    i += 1
plt.tight_layout()
plt.show()
```