Final Project: Reproducible Research

Xinyi Zhou, and Wuzhen Han

2024-11-28

```
import pandas as pd
import altair as alt
import geopandas as gpd
import json
import os
from vega_datasets import data
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from PyPDF2 import PdfReader
import spacy
from spacytextblob.spacytextblob import SpacyTextBlob

import warnings
warnings.filterwarnings('ignore')
alt.renderers.enable("png")
```

```
RendererRegistry.enable('png')
```

What is CARES Act

data was from "https://home.treasury.gov/system/files/136/Two-Year-ARP-Anniversary-Report.pdf"

```
pdf_path = '/Users/cynthia/Desktop/final-project-xy-wz/data/Extra credit.pdf'
output_path =
 ↪  '/Users/cynthia/Desktop/final-project-xy-wz/data/nlp_analysis.txt'
```

```
reader = PdfReader(pdf_path)
text = ""
for page in reader.pages:
    text += page.extract_text()

try:
    nlp = spacy.load("en_core_web_sm")
except OSError:
    import os
    os.system("python -m spacy download en_core_web_sm")
    nlp = spacy.load("en_core_web_sm")

nlp.add_pipe('spacytextblob')

doc = nlp(text)

with open(output_path, 'w', encoding='utf-8') as f:
    f.write("\n=== Sentiment Analysis ===\n")
    f.write(f"Overall Polarity: {doc._.blob.polarity}\n")
    f.write(f"Overall Subjectivity: {doc._.blob.subjectivity}\n")

    f.write("\nSentence-level Sentiment Analysis:\n")
    for sent in doc.sents:
        f.write(f"Sentence: {sent.text}\n")
        f.write(f"Polarity: {sent._.blob.polarity}\n")
        f.write(f"Subjectivity: {sent._.blob.subjectivity}\n")
        f.write("\n")

    f.write("\n=== Named Entity Recognition (NER) ===\n")
    for ent in doc.ents:
        f.write(f"Entity: {ent.text}, Label: {ent.label_}\n")

    f.write("\n=== Tokenization and POS tagging ===\n")
    for token in doc[:20]:
        f.write(f"Token: {token.text}, POS: {token.pos_}\n")

    f.write("\n=== Sentences ===\n")
    sentences = list(doc.sents)
    for sent in sentences[:5]:
        f.write(f"{sent}\n")
```

Merge Data

```
data_path = '/Users/cynthia/Desktop/final-project-xy-wz/data'
real_gdp_path = os.path.join(data_path, 'Real_GDP.csv')
unemployment_rate_path = os.path.join(data_path, 'Unemployment_rate.csv')

real_gdp = pd.read_csv(real_gdp_path)
unemployment_rate = pd.read_csv(unemployment_rate_path)

real_gdp['DATE'] = pd.to_datetime(real_gdp['DATE'])
unemployment_rate['DATE'] = pd.to_datetime(unemployment_rate['DATE'])

# Merge the datasets on the DATE column, keeping all rows from real_gdp left
 ↪  join
merged_data = pd.merge(real_gdp, unemployment_rate, on='DATE', how='left')

save_path = os.path.join(data_path, 'merged_data.csv')
merged_data.to_csv(save_path, index=False)
```

Data Preprocessing and Static graph

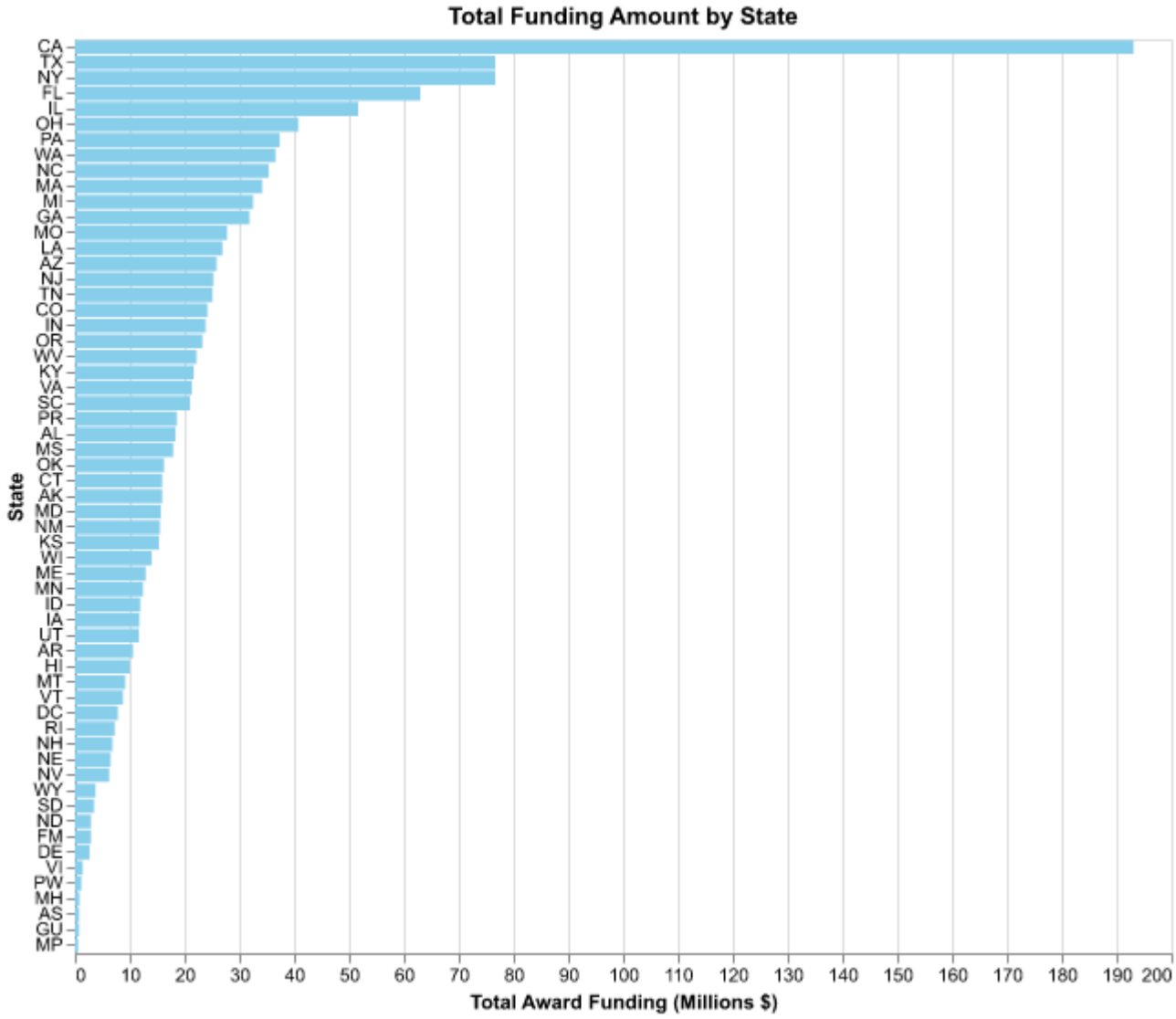
1.Bar chart of state-level CARES Act funding distribution

```
covid_report_path = os.path.join(data_path, 'COVID19_Grant_Report.csv')
data_raw = pd.read_csv(covid_report_path, skiprows=5)

original_number_of_grants = len(data_raw)
print(f"Original Number of Grants: {original_number_of_grants}")

# Remove dollar signs and commas from Award Funding and convert to float
data_raw['Award Funding'] = data_raw['Award Funding'].replace(
    r'[$,]', '', regex=True).astype(float)

converted_number_of_grants = len(data_raw)
print(
    f"Number of Grants after 'Award Funding' conversion:
     ↪  {converted_number_of_grants}")
```

```python
data_cleaned = data_raw.dropna(subset=['State', 'Award Funding'])
print(f"Number of Grants after dropping NaNs: {len(data_cleaned)}")

total_funding_dollars = data_cleaned['Award Funding'].sum()
print(
    f"Total Award Funding for all grants (Dollars):
     ↪  {total_funding_dollars:,.0f}")

# Group by State and sum the Award Funding, converting to millions
state_funding = data_cleaned.groupby(
    'State')['Award Funding'].sum().reset_index()

state_funding['Award Funding'] = state_funding['Award Funding'] / 1e6

# Sort states by funding amount in descending order
state_funding = state_funding.sort_values(
    by='Award Funding', ascending=False).reset_index(drop=True)

chart = alt.Chart(state_funding).mark_bar(color='skyblue').encode(
    y=alt.Y('State:N', sort='-x', title='State'),
    x=alt.X('Award Funding:Q', title='Total Award Funding (Millions $)'),
    tooltip=['State', 'Award Funding']
).properties(
    title='Total Funding Amount by State',
    width=600,
    height=500
).configure_axis(
    labelAngle=0
)

print("Top 3 states with the highest funding (in millions):")
print(state_funding.head(3))

chart
```

```
Original Number of Grants: 1389
Number of Grants after 'Award Funding' conversion: 1389
Number of Grants after dropping NaNs: 1387
Total Award Funding for all grants (Dollars): 1,316,374,135
Top 3 states with the highest funding (in millions):
  State  Award Funding
0    CA     193.072106
```

```
1     TX       76.701360
2     NY       76.696605
```

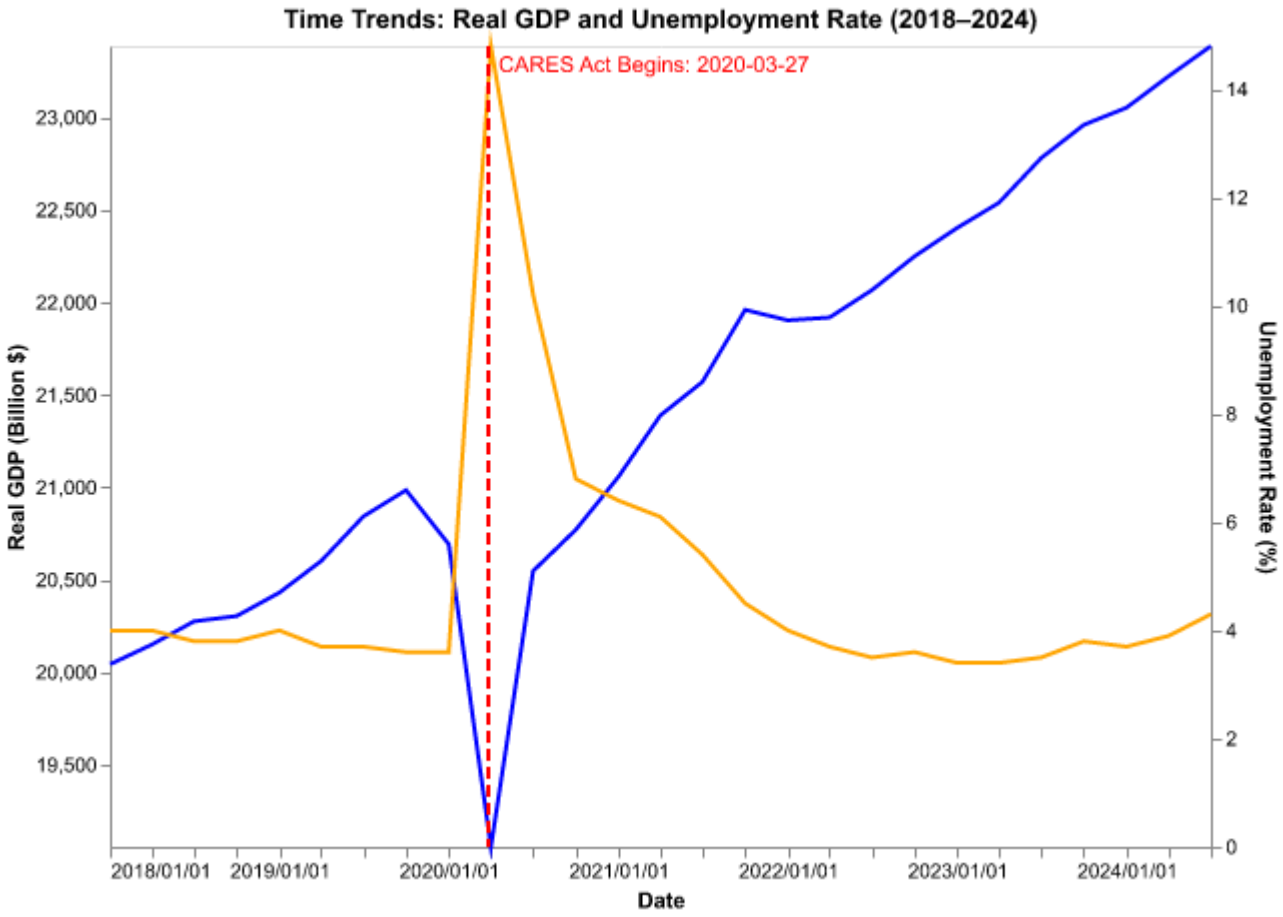


2. Economic Trends During COVID-19: Real GDP and Unemployment Rate (2018–2024)

```
import pandas as pd
import altair as alt
```

```
# Create a line chart for Real GDP
gdp_chart = alt.Chart(merged_data).mark_line(color='blue').encode(
    alt.X('DATE:T', title='Date', axis=alt.Axis(format='%Y/%m/%d',
 ↪  grid=False)),
    alt.Y('GDPC1:Q', title='Real GDP (Billion $)', axis=alt.Axis(grid=False),
          scale=alt.Scale(domain=[merged_data['GDPC1'].min(),
 ↪  merged_data['GDPC1'].max()]))
)

# Create a line chart for Unemployment Rate
unrate_chart = alt.Chart(merged_data).mark_line(color='orange').encode(
    alt.X('DATE:T', title='Date', axis=alt.Axis(format='%Y/%m/%d',
 ↪  grid=False)),
    alt.Y('UNRATE:Q', title='Unemployment Rate (%)',
 ↪  axis=alt.Axis(grid=False),
          scale=alt.Scale(domain=[0, merged_data['UNRATE'].max()]))
).properties(
    title="Time Trends: Real GDP and Unemployment Rate (2018-2024)"
)

# Add a dashed vertical line on March 27, 2020 (CARES Act implementation
 ↪  date)
vertical_line = alt.Chart(pd.DataFrame({'DATE': ['2020-03-27']})).mark_rule(
    color='red', strokeDash=[6, 3], strokeWidth=2
).encode(
    x=alt.X('DATE:T')
)

# Add text label for CARES Act
text_label = alt.Chart(pd.DataFrame({'DATE': ['2020-03-27'], 'label': ['CARES
 ↪  Act Begins: 2020-03-27']})).mark_text(
    align='left',
    baseline='middle',
    dx=5,
    dy=-190,
    color='red'
).encode(
    x=alt.X('DATE:T'),
    text='label'
)
```

```
chart = alt.layer(
    gdp_chart, unrate_chart, vertical_line, text_label
).resolve_scale(
    y='independent'
).properties(
    width=550,
    height=400
)

chart.show()
```



3. State-Level Unemployment Rate: Dynamic Heatmap by Time

getting data

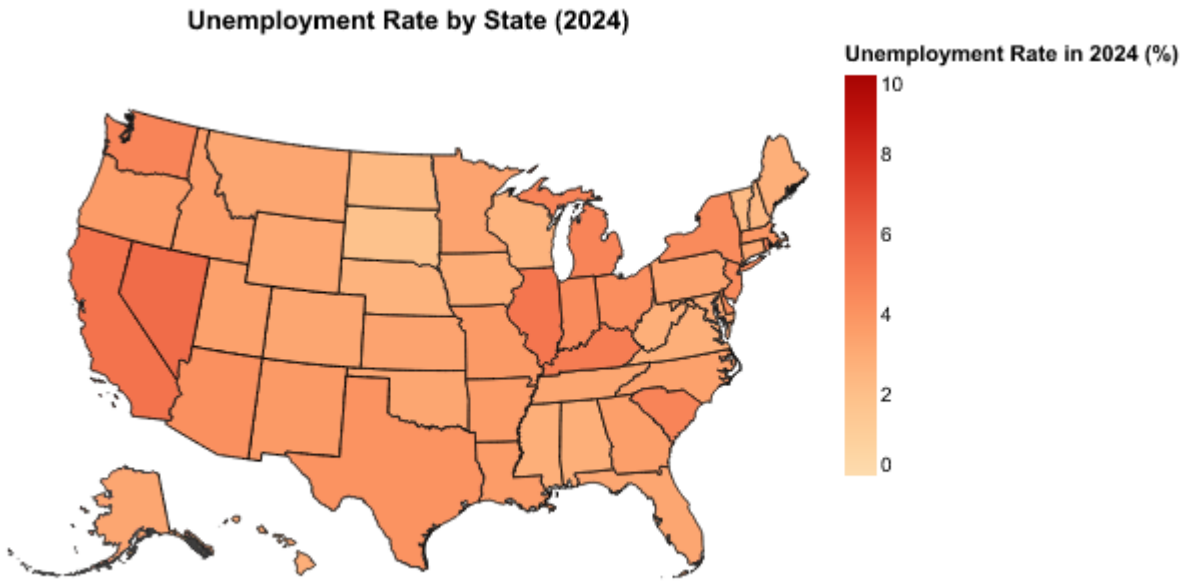
Download the required unemployment rate documentation https://dlt.ri.gov/media/15101/download?language=en .It is then processed manually to extract key information from the file, such as the year, state name, etc

a. Create static maps of unemployment rates by state

2024

```
unemp_state = os.path.join(data_path, 'anunemp.csv')
unemployment_data = pd.read_csv(unemp_state)

unemployment_data_filtered = unemployment_data.loc[unemployment_data['State']
 ↪  != "United States"].copy(
)
unemployment_data_filtered['State'] =
 ↪  unemployment_data_filtered['State'].str.strip(
).str.title()

data_path_shp =
 ↪  '/Users/cynthia/Desktop/final-project-xy-wz/cb_2018_us_state_500k'
shapefile_path = os.path.join(data_path_shp, 'cb_2018_us_state_500k.shp')
gdf_states = gpd.read_file(shapefile_path)

gdf_states['NAME'] = gdf_states['NAME'].str.strip().str.title()

gdf_merged = gdf_states.merge(
    unemployment_data_filtered, how='left', left_on='NAME', right_on='State'
)
gdf_merged['Rate_2024'] = gdf_merged['Rate_2024'].fillna(0)

geojson = gdf_merged.to_json()
geojson_data = json.loads(geojson)
```

```
states = alt.Data(values=geojson_data['features'])

unemployment_chart_2024 = alt.Chart(states).mark_geoshape(
    stroke='black',
    strokeWidth=1
).encode(
    color=alt.Color('properties.Rate_2024:Q',
                    scale=alt.Scale(domain=[0, 10], scheme='orangered'),
                    title='Unemployment Rate in 2024 (%)'),
    tooltip=[
        alt.Tooltip('properties.NAME:N', title='State'),
        alt.Tooltip('properties.Rate_2024:Q', title='Unemployment Rate (%)')
    ]
).project(
    type='albersUsa'
).properties(
    width=400,
    height=300,
    title='Unemployment Rate by State (2024)'
)

unemployment_chart_2024.display()

# Top 3 states with highest unemployment rate
top_3_states = gdf_merged.nlargest(3, 'Rate_2024')[['NAME', 'Rate_2024']]
print(top_3_states)
```

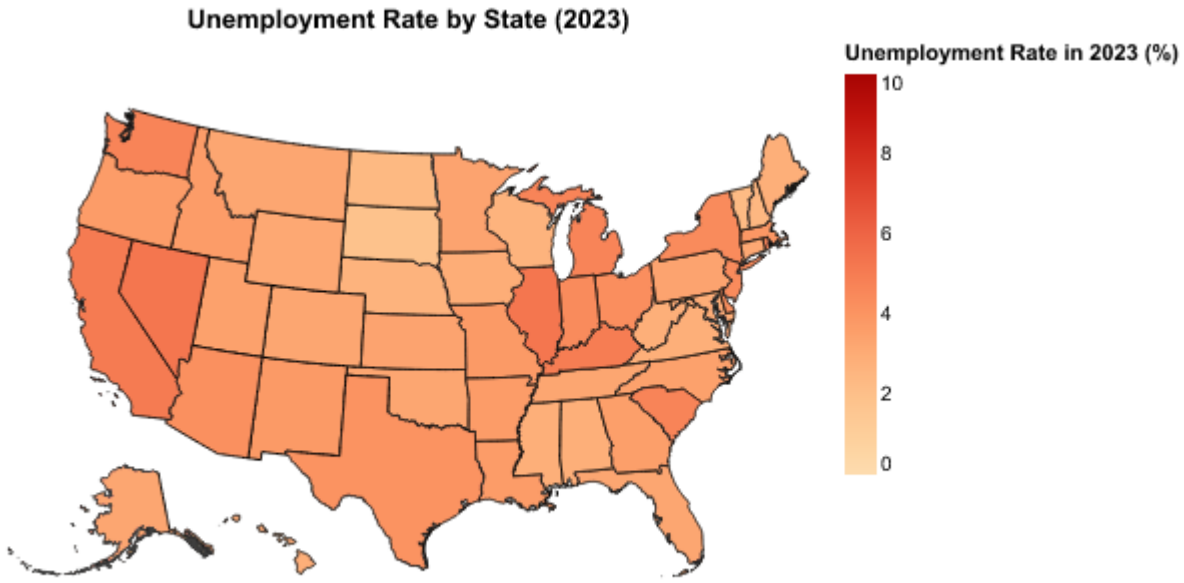


	NAME	Rate_2024
28	Nevada	5.7
36	District Of Columbia	5.7
16	California	5.4

2023

```
gdf_merged_2023 = gdf_states.merge(
    unemployment_data_filtered, how='left', left_on='NAME', right_on='State'
)
gdf_merged_2023['Rate_2023'] = gdf_merged_2023['Rate_2023'].fillna(0)

geojson = gdf_merged_2023.to_json()
geojson_data = json.loads(geojson)

states = alt.Data(values=geojson_data['features'])

unemployment_chart_2023 = alt.Chart(states).mark_geoshape(
    stroke='black',
    strokeWidth=1
).encode(
```

```
    color=alt.Color('properties.Rate_2023:Q',
                    scale=alt.Scale(domain=[0, 10], scheme='orangered'),
                    title='Unemployment Rate in 2023 (%)'),
    tooltip=[
        alt.Tooltip('properties.NAME:N', title='State'),
        alt.Tooltip('properties.Rate_2023:Q', title='Unemployment Rate (%)')
    ]
).project(
    type='albersUsa'
).properties(
    width=400,
    height=300,
    title='Unemployment Rate by State (2023)'
)

unemployment_chart_2023.display()

top_3_states_2023 = gdf_merged_2023.nlargest(
    3, 'Rate_2023')[['NAME', 'Rate_2023']]
print(top_3_states_2023)
```

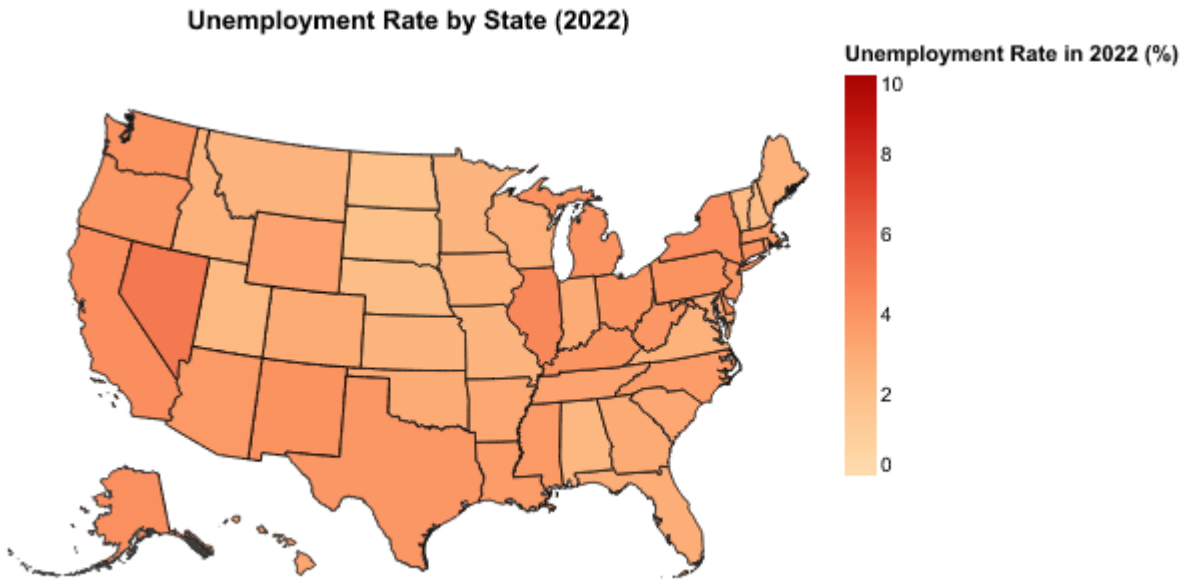



```
          NAME  Rate_2023
28      Nevada        5.3
```

```
29    Illinois        5.3
16  California        5.1
```

2022

```
gdf_merged_2022 = gdf_states.merge(
    unemployment_data_filtered, how='left', left_on='NAME', right_on='State'
)
gdf_merged_2022['Rate_2022'] = gdf_merged_2022['Rate_2022'].fillna(0)

geojson_2022 = gdf_merged_2022.to_json()
geojson_data_2022 = json.loads(geojson_2022)

states_2022 = alt.Data(values=geojson_data_2022['features'])

unemployment_chart_2022 = alt.Chart(states_2022).mark_geoshape(
    stroke='black',
    strokeWidth=1
).encode(
    color=alt.Color('properties.Rate_2022:Q',
                    scale=alt.Scale(domain=[0, 10], scheme='orangered'),
                    title='Unemployment Rate in 2022 (%)'),
    tooltip=[
        alt.Tooltip('properties.NAME:N', title='State'),
        alt.Tooltip('properties.Rate_2022:Q', title='Unemployment Rate (%)')
    ]
).project(
    type='albersUsa'
).properties(
    width=400,
    height=300,
    title='Unemployment Rate by State (2022)'
)

unemployment_chart_2022.display()

top_3_states_2022 = gdf_merged_2022.nlargest(3, 'Rate_2022')[['NAME',
 ↪  'Rate_2022']]
print(top_3_states_2022)
```

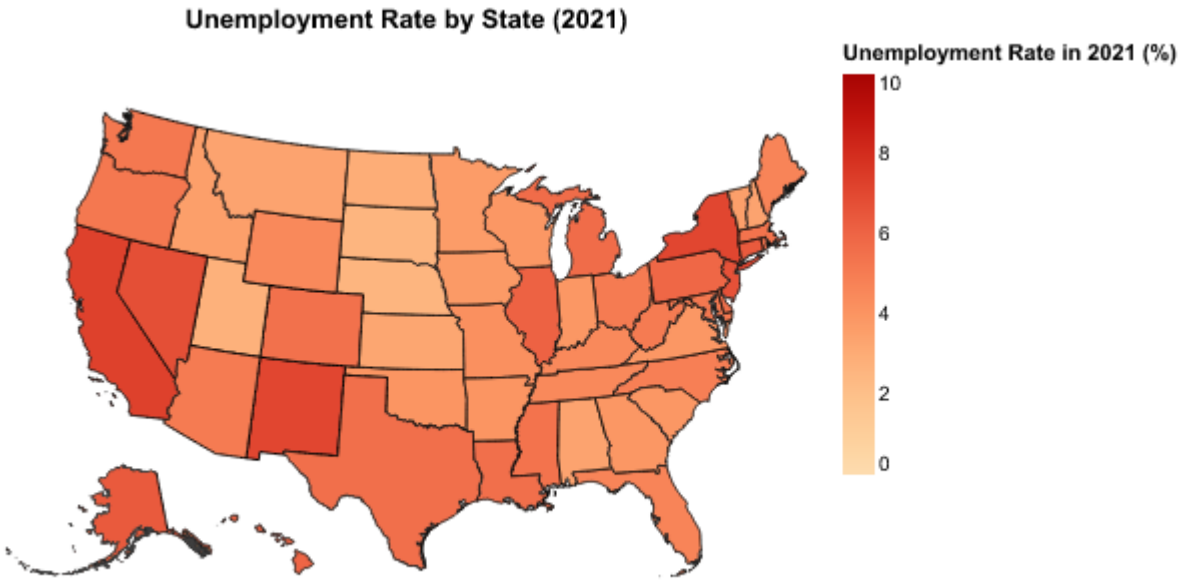


	NAME	Rate_2022
28	Nevada	5.2
36	District Of Columbia	4.7
29	Illinois	4.6

2021

```
gdf_merged_2021 = gdf_states.merge(
    unemployment_data_filtered, how='left', left_on='NAME', right_on='State')
gdf_merged_2021['Rate_2021'] = gdf_merged_2021['Rate_2021'].fillna(0)

geojson_2021 = gdf_merged_2021.to_json()
geojson_data_2021 = json.loads(geojson_2021)

states_2021 = alt.Data(values=geojson_data_2021['features'])

unemployment_chart_2021 = alt.Chart(states_2021).mark_geoshape(
    stroke='black',
    strokeWidth=1
).encode(
    color=alt.Color('properties.Rate_2021:Q',
```

```
                    scale=alt.Scale(domain=[0, 10], scheme='orangered'),
                    title='Unemployment Rate in 2021 (%)'),
    tooltip=[
        alt.Tooltip('properties.NAME:N', title='State'),
        alt.Tooltip('properties.Rate_2021:Q', title='Unemployment Rate (%)')
    ]
).project(
    type='albersUsa'
).properties(
    width=400,
    height=300,
    title='Unemployment Rate by State (2021)'
)

unemployment_chart_2021.display()

top_3_states_2021 = gdf_merged_2021.nlargest(
    3, 'Rate_2021')[['NAME', 'Rate_2021']]
print(top_3_states_2021)
```




```
          NAME  Rate_2021
16  California        7.3
```

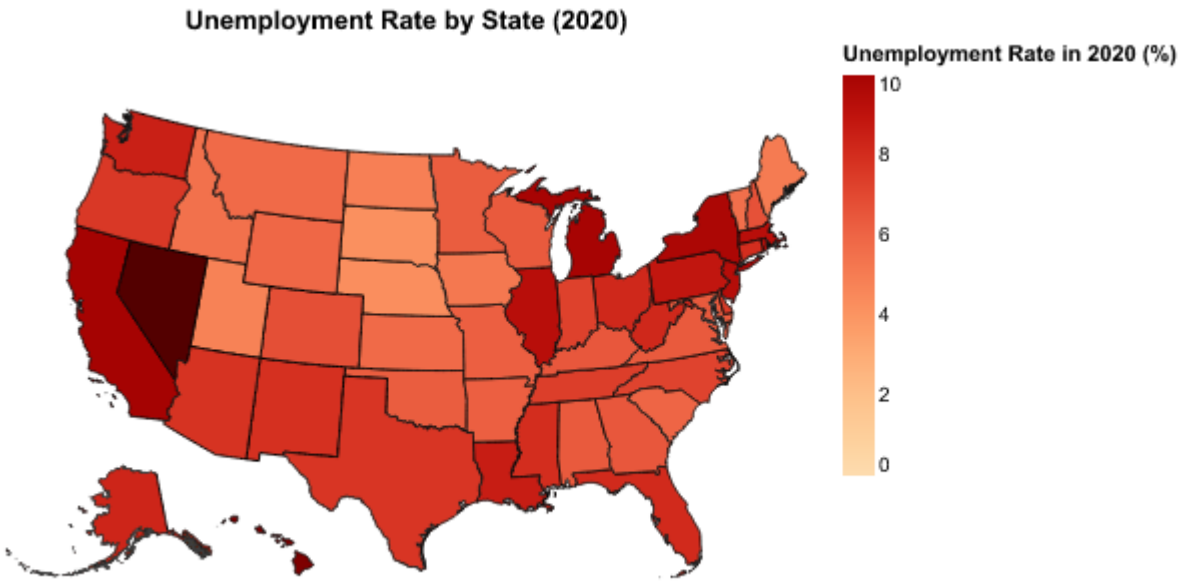
```
12  New Mexico        7.1
25    New York        7.1
```

2020 2019 2018

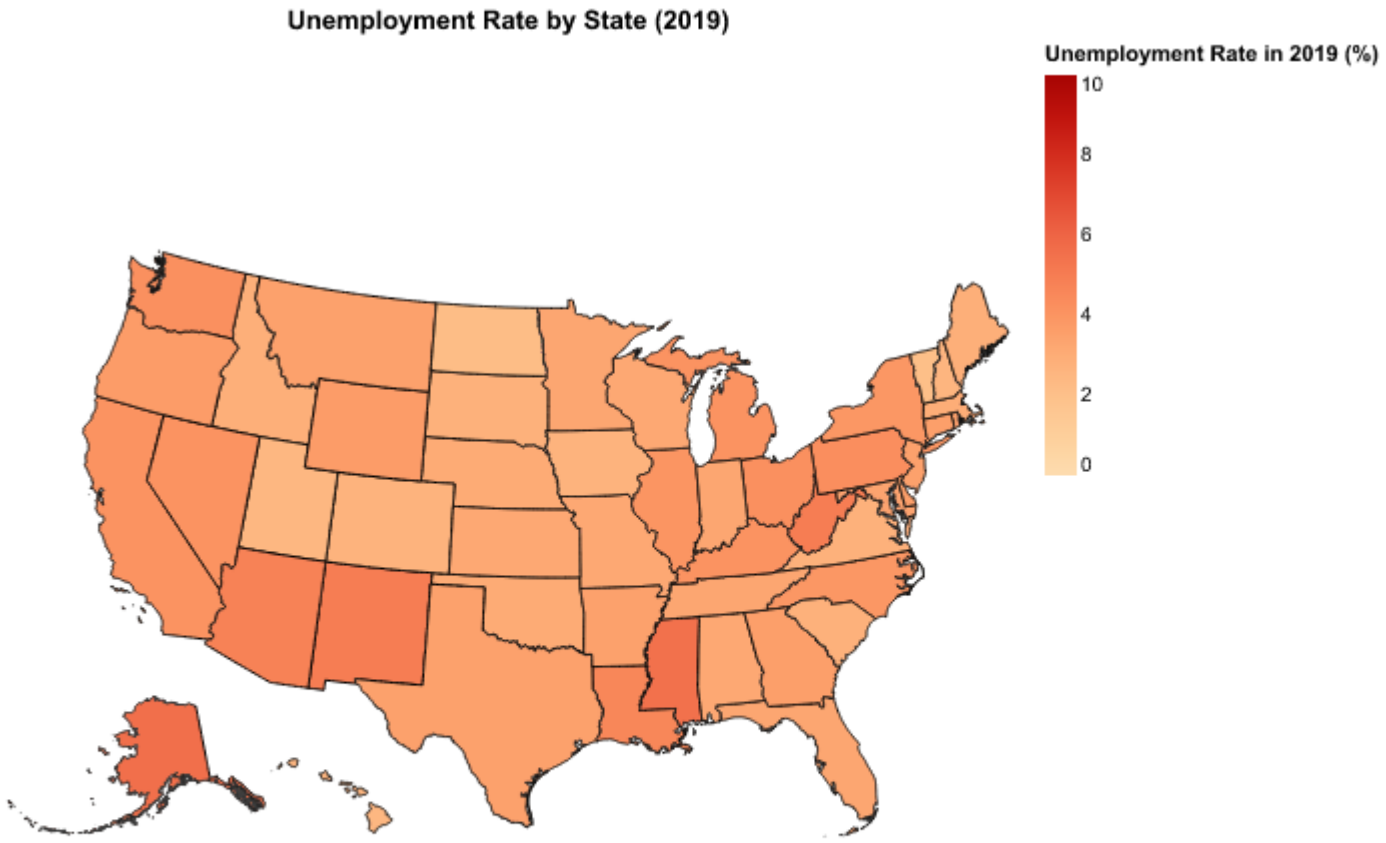
```
gdf_merged_2020 = gdf_states.merge(
    unemployment_data_filtered, how='left', left_on='NAME',
 ↪  right_on='State').copy()
gdf_merged_2020['Rate_2020'] = gdf_merged_2020['Rate_2020'].fillna(0)

geojson_2020 = gdf_merged_2020.to_json()
geojson_data_2020 = json.loads(geojson_2020)

states_2020 = alt.Data(values=geojson_data_2020['features'])

unemployment_chart_2020 = alt.Chart(states_2020).mark_geoshape(
    stroke='black',
    strokeWidth=1
).encode(
    color=alt.Color('properties.Rate_2020:Q',
                    scale=alt.Scale(domain=[0, 10], scheme='orangered'),
                    title='Unemployment Rate in 2020 (%)'),
    tooltip=[
        alt.Tooltip('properties.NAME:N', title='State'),
        alt.Tooltip('properties.Rate_2020:Q', title='Unemployment Rate (%)')
    ]
).project(
    type='albersUsa'
).properties(
    width=400,
    height=300,
    title='Unemployment Rate by State (2020)'
)

unemployment_chart_2020.display()

top_3_states_2020 = gdf_merged_2020.nlargest(
    3, 'Rate_2020')[['NAME', 'Rate_2020']]
print(top_3_states_2020)
```

```
gdf_merged_2019 = gdf_states.merge(
    unemployment_data_filtered, how='left', left_on='NAME',
 ↪  right_on='State').copy()
gdf_merged_2019['Rate_2019'] = gdf_merged_2019['Rate_2019'].fillna(0)

geojson_2019 = gdf_merged_2019.to_json()
geojson_data_2019 = json.loads(geojson_2019)

states_2019 = alt.Data(values=geojson_data_2019['features'])

unemployment_chart_2019 = alt.Chart(states_2019).mark_geoshape(
    stroke='black',
    strokeWidth=1
).encode(
    color=alt.Color('properties.Rate_2019:Q',
                    scale=alt.Scale(domain=[0, 10], scheme='orangered'),
                    title='Unemployment Rate in 2019 (%)'),
    tooltip=[
        alt.Tooltip('properties.NAME:N', title='State'),
        alt.Tooltip('properties.Rate_2019:Q', title='Unemployment Rate (%)')
    ]
).project(
    type='albersUsa'
).properties(
    width=500,
    height=500,
    title='Unemployment Rate by State (2019)'
)

unemployment_chart_2019.display()

top_3_states_2019 = gdf_merged_2019.nlargest(
    3, 'Rate_2019')[['NAME', 'Rate_2019']]
print(top_3_states_2019)

gdf_merged_2018 = gdf_states.merge(
    unemployment_data_filtered, how='left', left_on='NAME',
 ↪  right_on='State').copy()
gdf_merged_2018['Rate_2018'] = gdf_merged_2018['Rate_2018'].fillna(0)

geojson_2018 = gdf_merged_2018.to_json()
geojson_data_2018 = json.loads(geojson_2018)
```

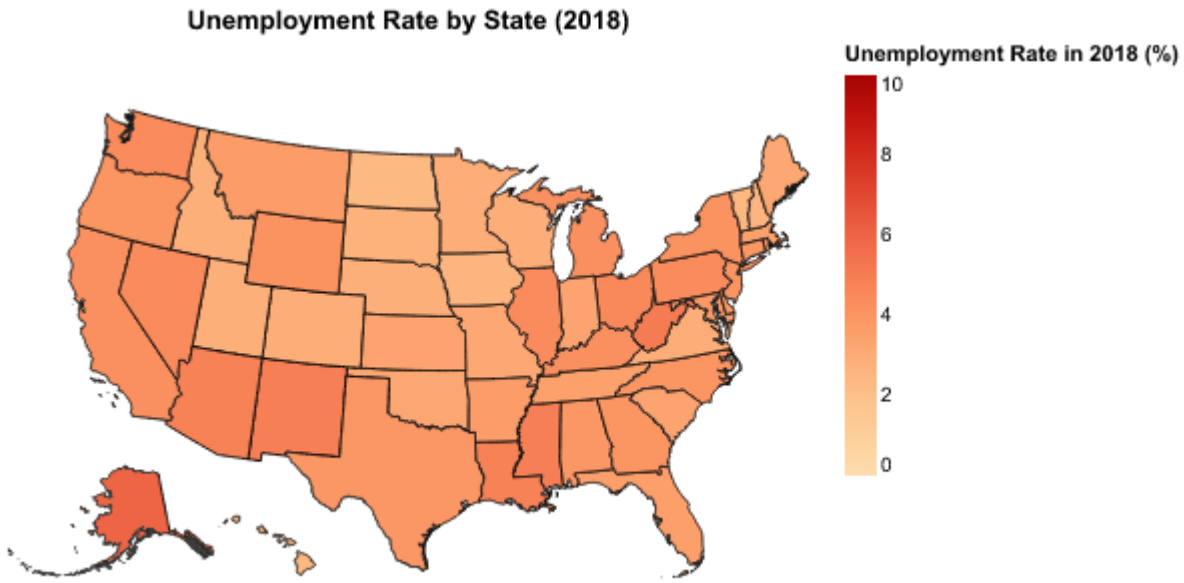
```
states_2018 = alt.Data(values=geojson_data_2018['features'])

unemployment_chart_2018 = alt.Chart(states_2018).mark_geoshape(
    stroke='black',
    strokeWidth=1
).encode(
    color=alt.Color('properties.Rate_2018:Q',
                    scale=alt.Scale(domain=[0, 10], scheme='orangered'),
                    title='Unemployment Rate in 2018 (%)'),
    tooltip=[
        alt.Tooltip('properties.NAME:N', title='State'),
        alt.Tooltip('properties.Rate_2018:Q', title='Unemployment Rate (%)')
    ]
).project(
    type='albersUsa'
).properties(
    width=400,
    height=300,
    title='Unemployment Rate by State (2018)'
)

unemployment_chart_2018.display()

top_3_states_2018 = gdf_merged_2018.nlargest(
    3, 'Rate_2018')[['NAME', 'Rate_2018']]
print(top_3_states_2018)
```



	NAME	Rate_2020
28	Nevada	13.5
42	Hawaii	11.7
16	California	10.1



	NAME	Rate_2019
27	Alaska	5.6
0	Mississippi	5.5
36	District Of Columbia	5.5



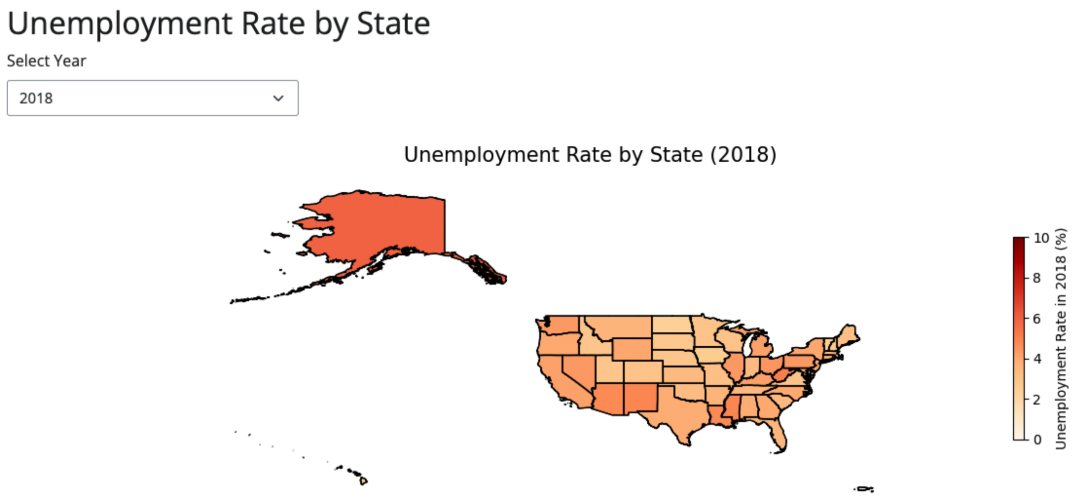
	NAME	Rate_2018
27	Alaska	6.0
36	District Of Columbia	5.7
4	West Virginia	5.1

```
output_path = '/Users/cynthia/Desktop/final-project-xy-wz/picture/dynamic_1'

unemployment_chart_2018.save(os.path.join(
    output_path, 'unemployment_rate_2018.png'))
unemployment_chart_2019.save(os.path.join(
    output_path, 'unemployment_rate_2019.png'))
unemployment_chart_2020.save(os.path.join(
    output_path, 'unemployment_rate_2020.png'))
unemployment_chart_2021.save(os.path.join(
    output_path, 'unemployment_rate_2021.png'))
unemployment_chart_2022.save(os.path.join(
    output_path, 'unemployment_rate_2022.png'))
unemployment_chart_2023.save(os.path.join(
    output_path, 'unemployment_rate_2023.png'))
unemployment_chart_2024.save(os.path.join(
    output_path, 'unemployment_rate_2024.png'))
```

b. Create the dynamic maps of unemployment rates by state in app.py file

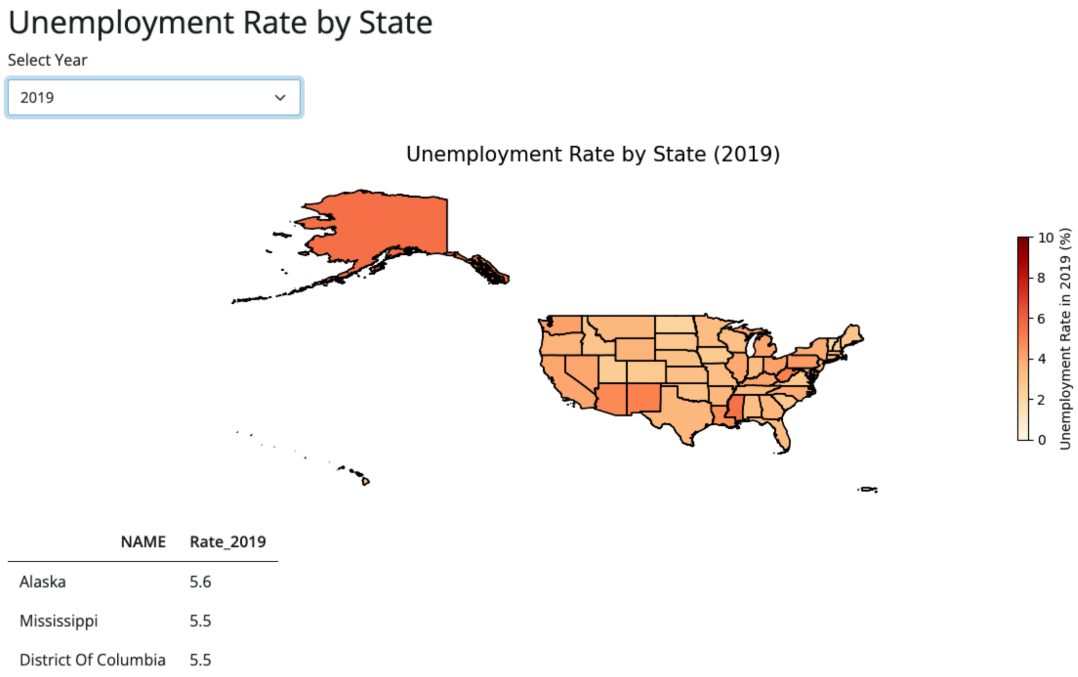
For a detailed view of how these figures were generated, please refer to the code in app.py. The dynamic trends presented here are the final output of our detailed data processing and visualization pipeline.

```
dynamic_1_path =
 ↪  '/Users/cynthia/Desktop/final-project-xy-wz/picture/dynamic_1'

image_path_1 = os.path.join(dynamic_1_path, 'p1.png')

img_1 = mpimg.imread(image_path_1)
plt.figure(figsize=(10, 8))
plt.imshow(img_1)
plt.axis('off')
plt.show()
```



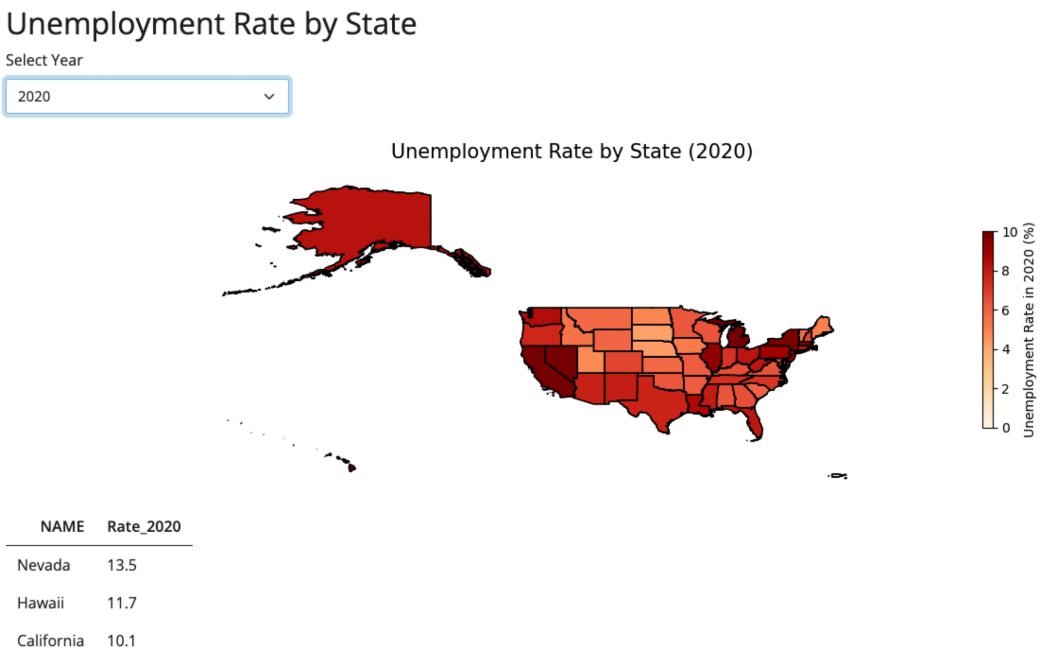
NAME	Rate_2018
Alaska	6.0
District Of Columbia	5.7
West Virginia	5.1

```
image_path_2 = os.path.join(dynamic_1_path, 'p2.png')

img_2 = mpimg.imread(image_path_2)
plt.figure(figsize=(10, 8))
plt.imshow(img_2)
plt.axis('off')
plt.show()
```



NAME	Rate_2019
Alaska	5.6
Mississippi	5.5
District Of Columbia	5.5

```
image_path_3 = os.path.join(dynamic_1_path, 'p3.png')

img_3 = mpimg.imread(image_path_3)
plt.figure(figsize=(10, 8))
plt.imshow(img_3)
plt.axis('off')
plt.show()
```



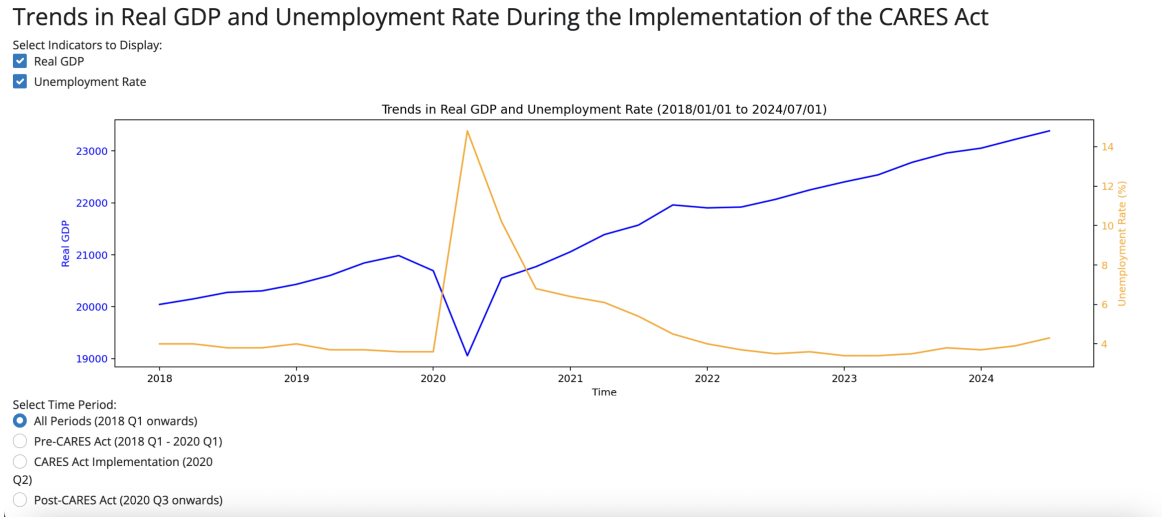
NAME	Rate_2020
Nevada	13.5
Hawaii	11.7
California	10.1

4. Dynamic Trends of Real GDP and Unemployment Rate Under CARES Act

For a detailed view of how these figures were generated, please refer to the code in app.py. The dynamic trends presented here are the final output of our detailed data processing and visualization pipeline.

```
dynamic_2_path =
 ↪  '/Users/cynthia/Desktop/final-project-xy-wz/picture/dynamic_2'

screenshot_path_1 = os.path.join(dynamic_2_path, 'All_period.png')
screenshot_path_2 = os.path.join(dynamic_2_path, 'Pre_Cares.png')
screenshot_path_3 = os.path.join(dynamic_2_path, 'Implementation_period.png')
screenshot_path_4 = os.path.join(dynamic_2_path, 'After_Implementation.png')


def show_image(image_path, figsize=(10, 8)):
```

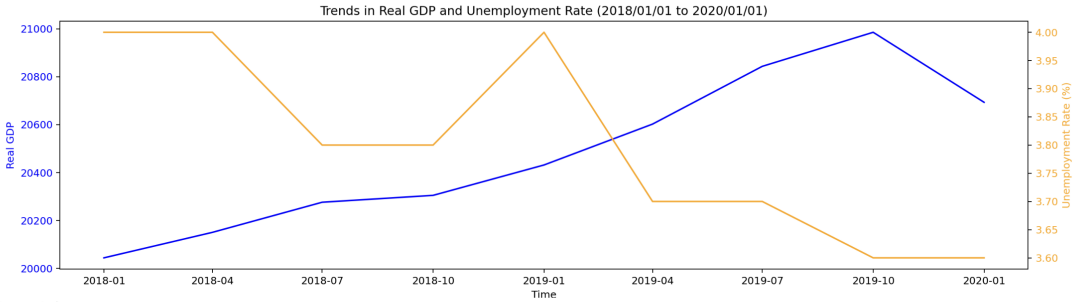
```
    img = mpimg.imread(image_path)
    plt.figure(figsize=figsize)
    plt.imshow(img)
    plt.axis('off')
    plt.show()

# Show all the images sequentially to highlight different phases under CARES
 ↪  Act
show_image(screenshot_path_1)
show_image(screenshot_path_2)
show_image(screenshot_path_3)
show_image(screenshot_path_4)
```



Trends in Real GDP and Unemployment Rate During the Implementation of the CARES Act

Select Indicators to Display:

- [x] Real GDP
- [x] Unemployment Rate



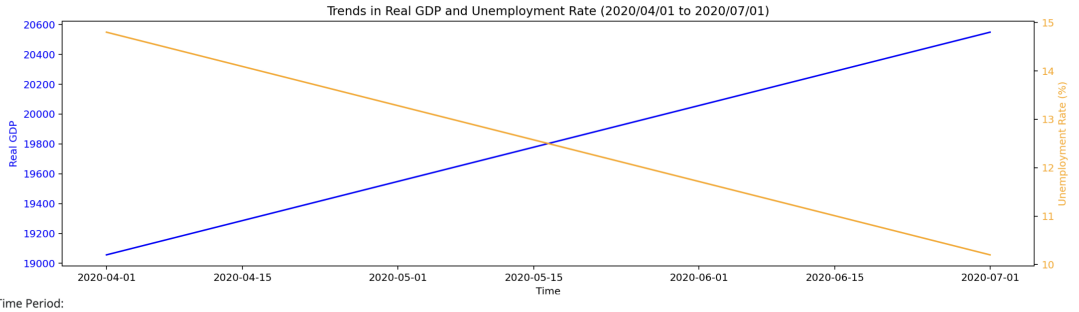
Select Time Period:

- ( ) All Periods (2018 Q1 onwards)
- (•) Pre-CARES Act (2018 Q1 - 2020 Q1)
- ( ) CARES Act Implementation (2020 Q2)
- ( ) Post-CARES Act (2020 Q3 onwards)

Trends in Real GDP and Unemployment Rate During the Implementation of the CARES Act

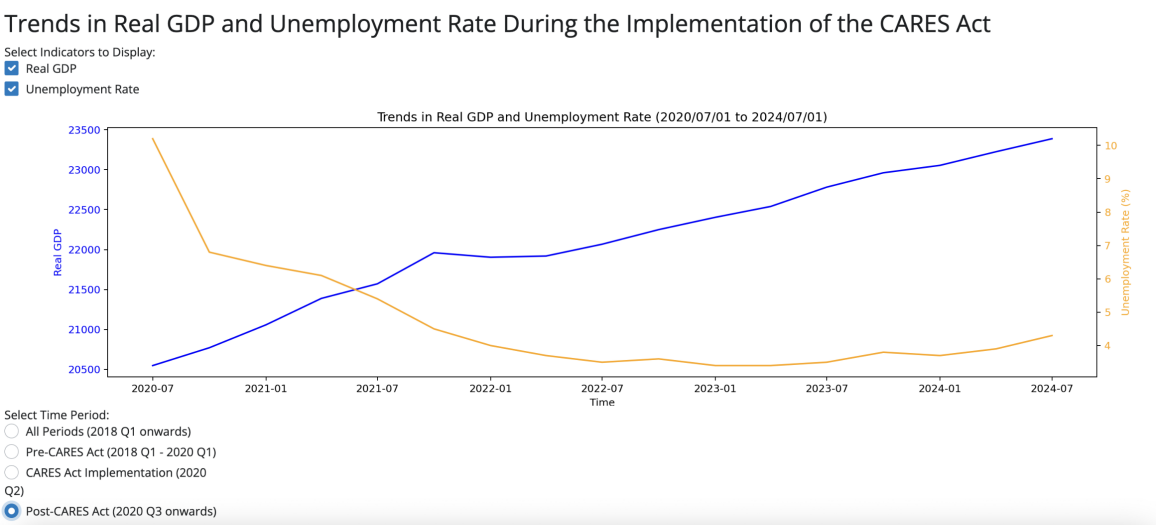
Select Indicators to Display:

- [x] Real GDP
- [x] Unemployment Rate



Select Time Period:

- ( ) All Periods (2018 Q1 onwards)
- ( ) Pre-CARES Act (2018 Q1 - 2020 Q1)
- (•) CARES Act Implementation (2020 Q2)
- ( ) Post-CARES Act (2020 Q3 onwards)


Trends in Real GDP and Unemployment Rate During the Implementation of the CARES Act
Select Indicators to Display:
Real GDP
Unemployment Rate
Trends in Real GDP and Unemployment Rate (2020/07/01 to 2024/07/01)
Real GDP
Unemployment Rate (%)
Time
Select Time Period:
All Periods (2018 Q1 onwards)
Pre-CARES Act (2018 Q1 - 2020 Q1)
CARES Act Implementation (2020 Q2)
Post-CARES Act (2020 Q3 onwards)