

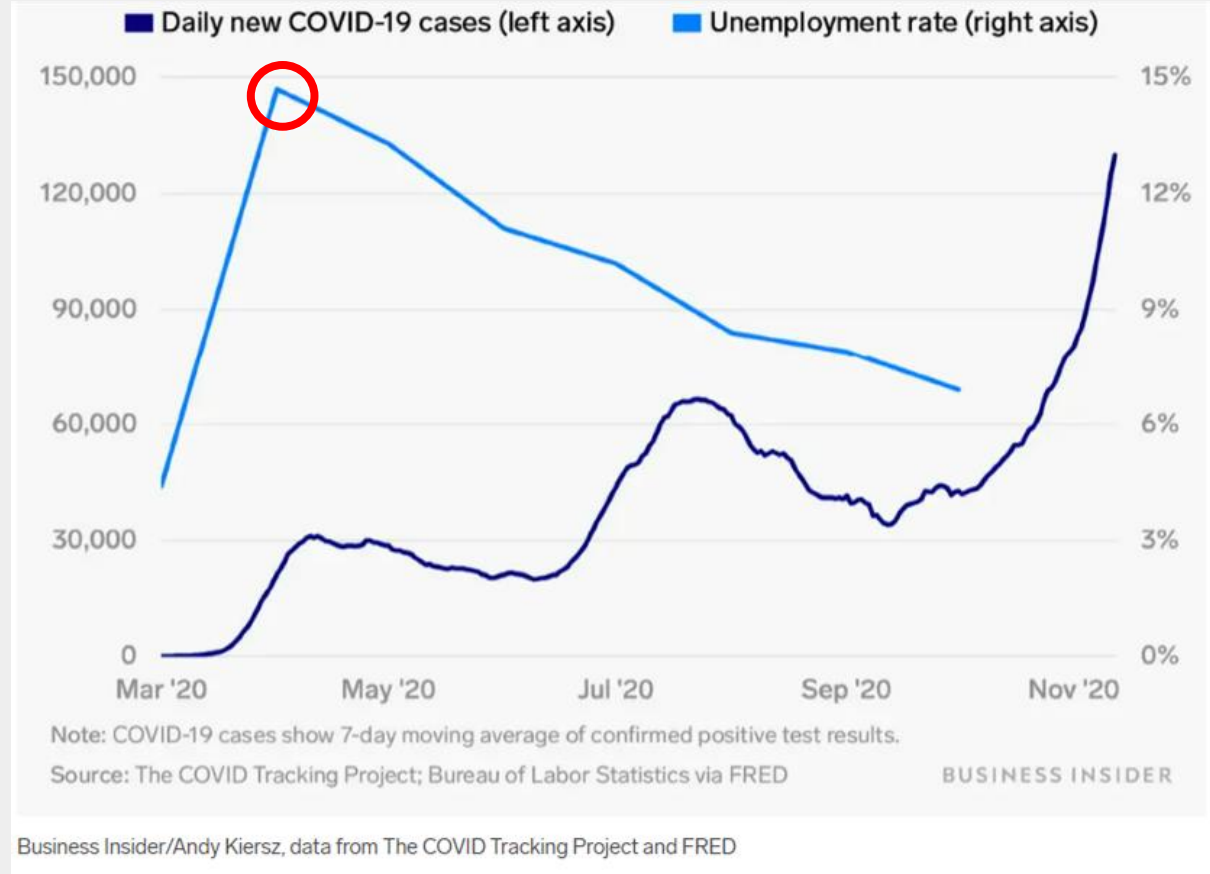
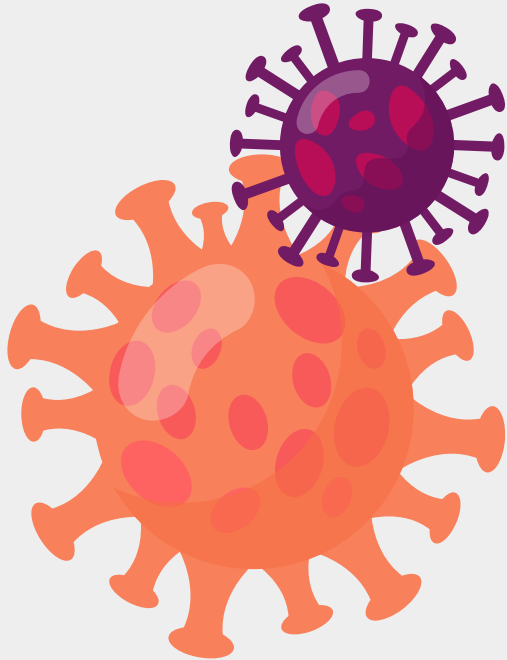
Group 2

Lynne, Jonna, Yen,
Sandy, Lilian

Does Vaccine Help the Economy?

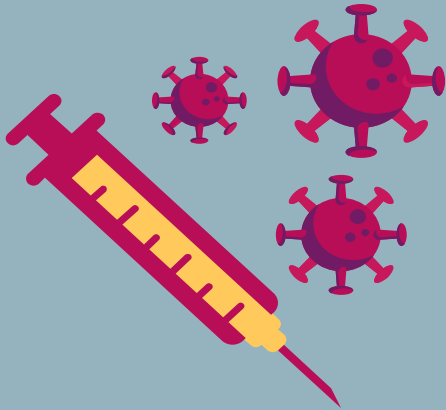


U.S. COVID-19 cases and unemployment

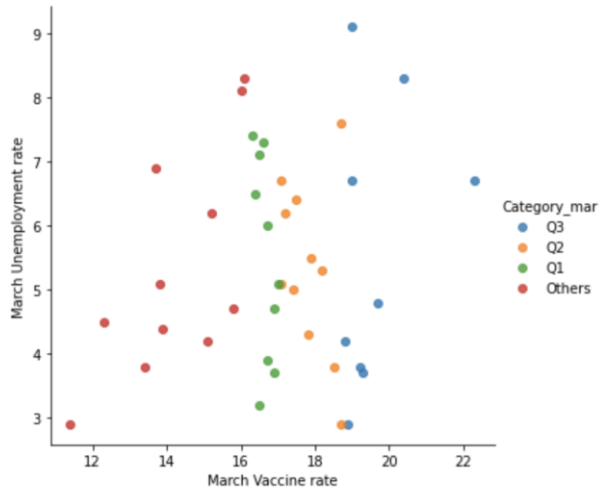


“Vaccines are absolutely, 100 percent the answer to economic challenges imposed by the pandemic.”

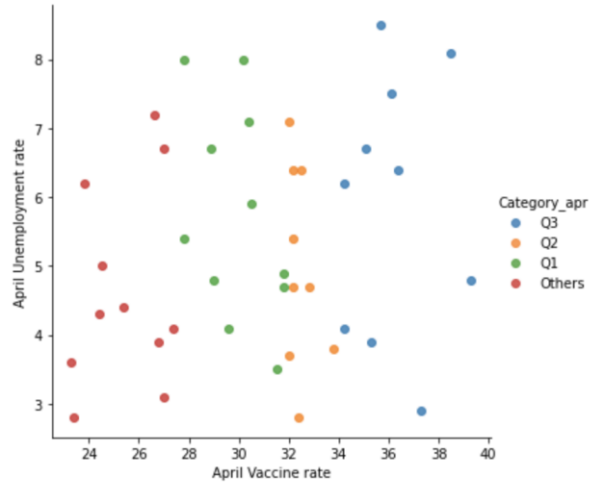
- Economics Professor Jonathan Me



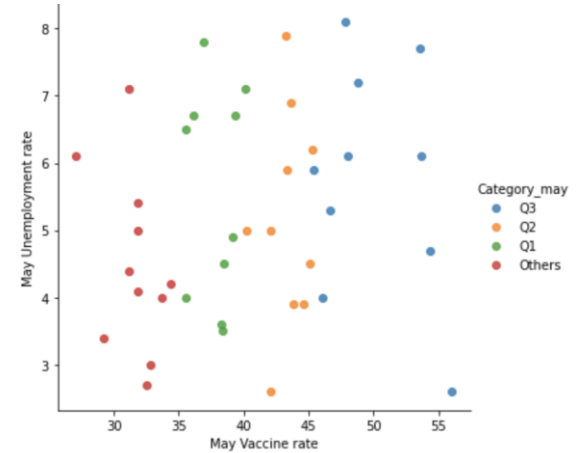
Scatter Plot by states



March



April



May

X-Axis : Vaccination Rate
Y-Axis : Unemployment Rate

Categorizing States Using quartile

```
#We first categorize states by their vaccine rate using quartile
def v_categories(x):
    if x['03_vac'] > unem_vacc_all['03_vac'].quantile(0.75):
        return "Q3"
    elif x['03_vac'] > unem_vacc_all['03_vac'].quantile(0.5):
        return "Q2"
    elif x['03_vac'] > unem_vacc_all['03_vac'].quantile(0.25):
        return "Q1"
    else:
        return "Others"

def v_categories1(x):
    if x['04_vac'] > unem_vacc_all['04_vac'].quantile(0.75):
        return "Q3"
    elif x['04_vac'] > unem_vacc_all['04_vac'].quantile(0.5):
        return "Q2"
    elif x['04_vac'] > unem_vacc_all['04_vac'].quantile(0.25):
        return "Q1"
    else:
        return "Others"

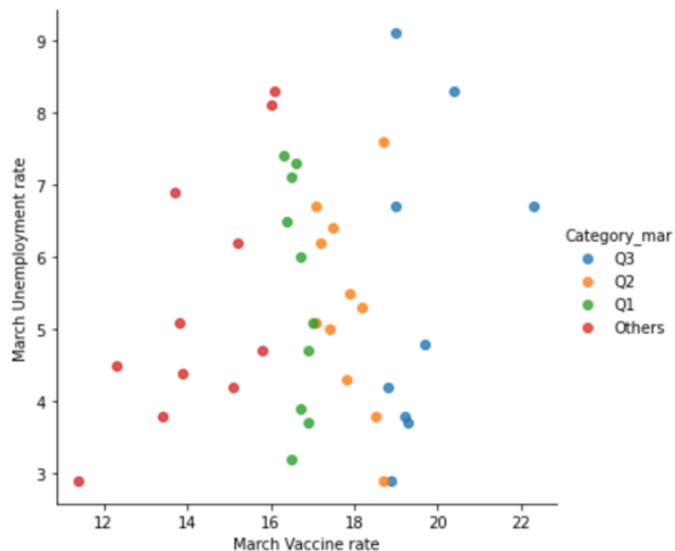
def v_categories2(x):
    if x['05_vac'] > unem_vacc_all['05_vac'].quantile(0.75):
        return "Q3"
    elif x['05_vac'] > unem_vacc_all['05_vac'].quantile(0.5):
        return "Q2"
    elif x['05_vac'] > unem_vacc_all['05_vac'].quantile(0.25):
        return "Q1"
    else:
        return "Others"
```

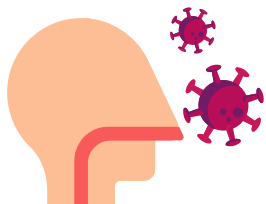
```
#Then we draw the scatterplot to see if there is an obvious relationship between the two variables
```

```
#March
```

```
mar = sns.lmplot(x='03_vac',y='03_unem',hue = 'Category_mar',hue_order=['Q3','Q2','Q1','Others'],data=unem_vacc_all,fit_reg = False)  
mar.set_axis_labels('March Vaccine rate', 'March Unemployment rate')
```

```
<seaborn.axisgrid.FacetGrid at 0x7f5416a0b510>
```





Overall Analysis Framework

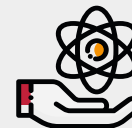
1

DATA CLEANING BEFORE SAMPLING



2

DATA CLEANING BEFORE VISUALIZATION

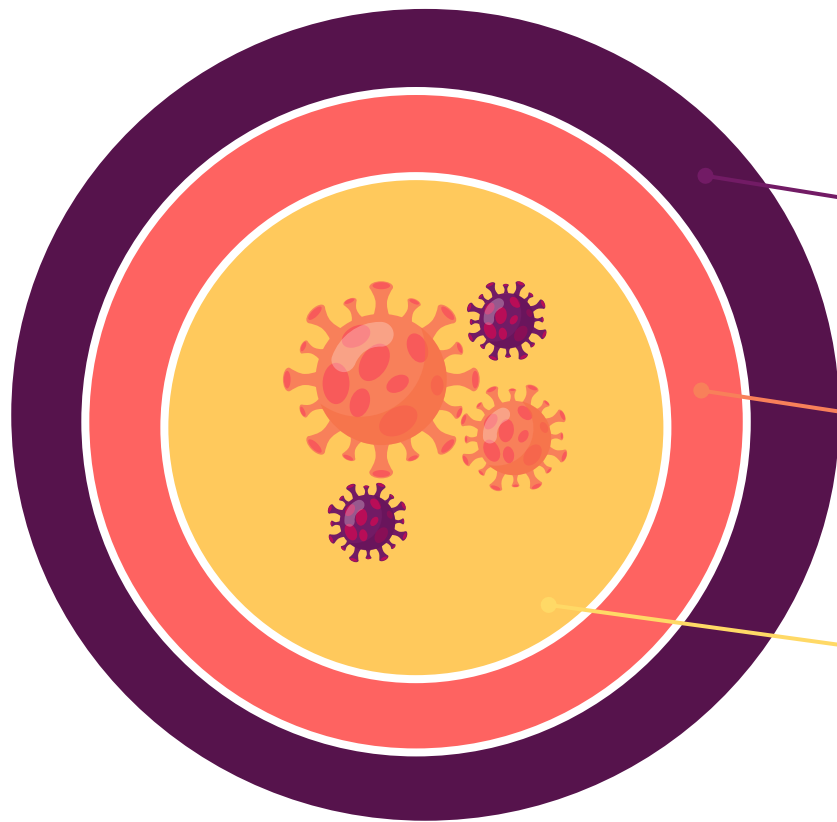


3

DATA VISUALIZATION USING MATPLOTLIB



Sampling Criteria



1

Population

U.S. States

2

Similar Economic Development

Top 25 in State GDP

3

Similar Reported Covid cases

Top 5 in Case Rate per 100,000

States Filtered by GDP Data



- ★ Ensure the economic impact can be directly observed.
- ★ Choose similar GDP state to avoid other factors that may cause economic index bias.

```
[4] gdp_file_dir = 'state_gdp_raw_in_millions.csv'
     state_gdp = pd.read_csv(gdp_file_dir, index_col = 'Geography')

[5] Q1_gdp = pd.DataFrame(state_gdp, columns = ['2020 Q1'])
     gdp_25_withUS = Q1_gdp.nlargest(26, '2020 Q1')

[50] gdp_25_withoutUS = gdp_25_withUS[1:] #without the first row:USA
      gdp_25_withoutUS.head(26).astype(int)
```

	2020 Q1
Geography	
California	3171221
Texas	1805780
New York	1762518
Florida	1116161
Illinois	877747
Pennsylvania	802664
Ohio	690048
New Jersey	638617
Georgia	625826
Washington	623211
Massachusetts	595752

Covid Cases Rate Merged with Candidate

```
covin_case = pd.read_csv('Cases number.csv', index_col = 'State/Territory')

gdp_case_25 = gdp_25_withoutUS.merge(covin_case, how= 'left', left_index= True,
gdp_case_25 = gdp_case_25.fillna(0)
```

	2020 Q1	Total Cases	Confirmed Cases	Probable Cases	Cases in Last 7 Days	Case Rate per 100000	Total Deaths	Confirmed Deaths	Probable Deaths	Deaths in Last 7 Days	Death Rate per 100000
Geography											
California	3.171222e+06	3708861.0	0.0	0.0	7143.0	9387.0	62890.0	0.0	0.0	284.0	159.0
Texas	1.805781e+06	2982177.0	0.0	0.0	6631.0	10285.0	51163.0	0.0	0.0	177.0	176.0
New York	1.762518e+06	1156796.0	0.0	0.0	915.0	10464.0	19919.0	0.0	0.0	30.0	180.0

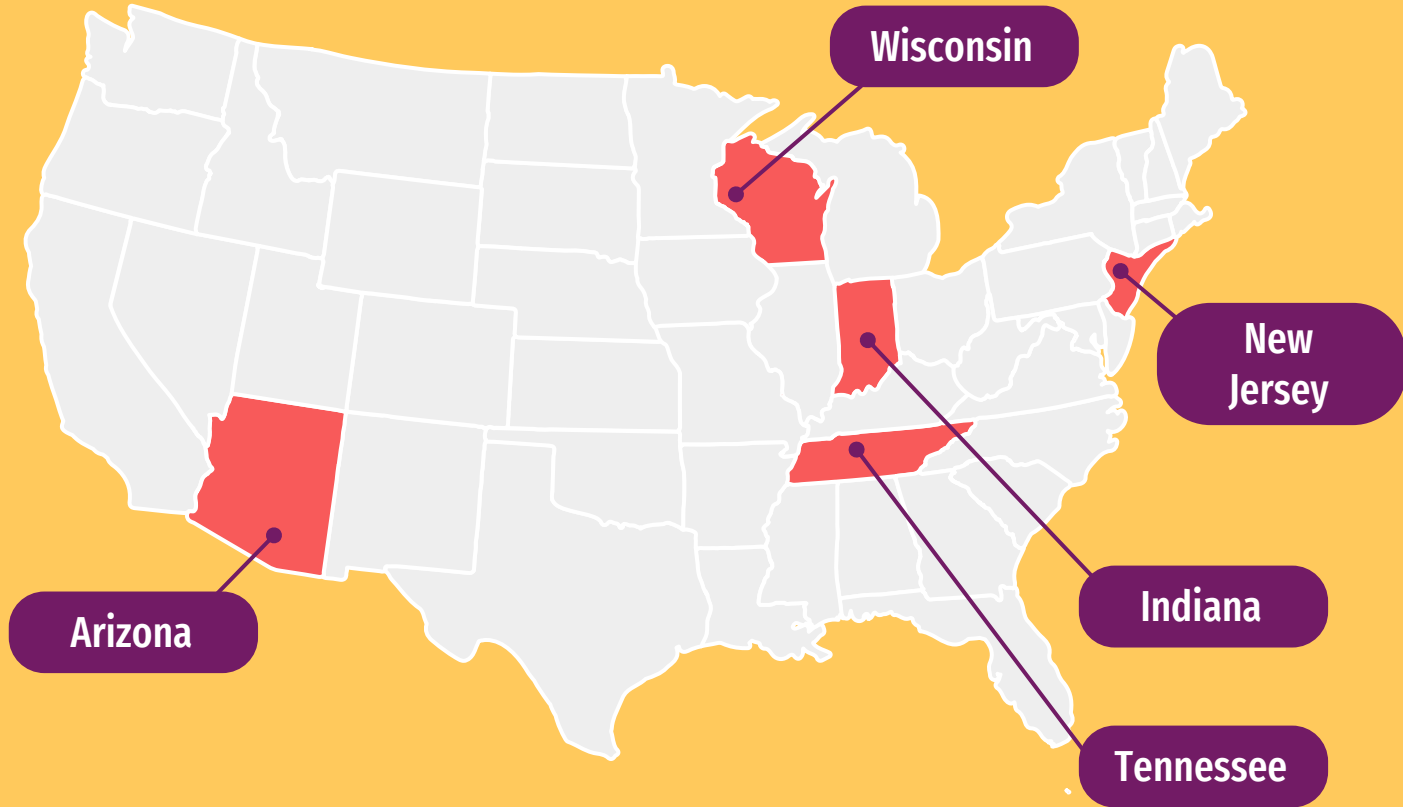
Top 5 States Extracted

```
top5cases = gdp_case_25.nlargest(5, 'Case Rate per 100000') #top
top5cases.index.name = "State"
df_t5 = pd.DataFrame(top5cases, columns = ['Case Rate per 100000
confirmed_portion = df_t5["Case Rate per 100000"]/1000
df_t5.insert(0, 'confirmed_portion(%)', confirmed_portion)
```



confirmed_portion(%) Case Rate per 100000		
State		
Tennessee	12.692	12692.0
Arizona	12.258	12258.0
Wisconsin	11.632	11632.0
New Jersey	11.511	11511.0
Indiana	11.180	11180.0

The Selection of the 5 States



Comparing Factor Import



Unemployment Rate

- Reflect short-term economic situation
- Related to person



Vaccination Rate



Unemployment Rate Data Import

```
unemployment = pd.read_csv("unemployment_rate_202005-202105.csv")
unemployment.set_index("Geography", inplace=True)
unemployment.index.name = "State"
```

	2020/1/1	2020/2/1	2020/3/1	2020/4/1	2020/5/1	2020/6/1	2020/7/1	2020/8/1	2020/9/1
State									
United States	3.5	3.5	4.4	14.8	13.3	11.1	10.2	8.4	7.9
Alabama	2.7	2.6	2.6	13.2	7.9	7.7	7.4	7.1	6.8
Alaska	5.0	5.1	5.1	11.8	11.6	11.3	11.0	7.2	6.9
Arizona	4.8	4.9	5.0	14.2	10.6	10.7	10.4	7.0	6.7
Arkansas	3.7	3.8	3.9	10	8.5	7.8	7.1	6.5	6.2
California	4.2	4.3	4.5	16	15.6	14.1	13.2	12.3	11.8

Vaccination Data

★ Extract Specific Date Information

-> Pivot

★ Data Merge & Cleaning

```
vaccine_data = pd.read_csv("vaccine row data.csv", index_col="Location")
ME_vaccine_data = pd.DataFrame(vaccine_data, columns=["Date", "Series_Complete_Pop_Pct"])
PV_ME_vaccine = ME_vaccine_data.pivot(index=None,
                                       columns="Date",
                                       values="Series_Complete_Pop_Pct")

PV_ME_vaccine = pd.merge(PV_ME_vaccine, stateabbr, on=['Location'], how='right')
PV_ME_vaccine.set_index("State", inplace=True)
#rename column name
PV_ME_vaccine = PV_ME_vaccine.rename(columns={'03/31/2021': '2021/03 Vaccine Rate', '04
PV_ME_vaccine.dropna(how="all", inplace=True)
```



	Location	2021/03 Vaccine Rate	2021/04 Vaccine Rate	2021/05 Vaccine Rate
State				
Alabama	AL	13.4	23.3	29.2
Alaska	AK	22.3	35.1	39.3
Arizona	AZ	17.1	28.9	36.1
Arkansas	AR	13.9	25.4	31.2
California	CA	16.1	30.2	43.2

Data Cleaning Before Visualization

Merging Datasets

Unemployment Rate Dataset
Vaccination Rate Dataset

1

Retrieving Specific Data

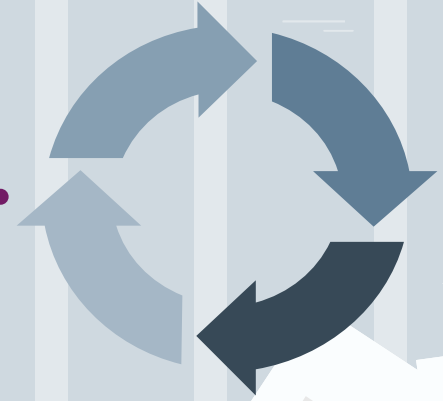
Filter the Top 5 states
Rename and Extract Specific Columns

2

Transpose

Switch Row to Column making
vaccination and unemployment rate in
the same column

3



Merging Datasets

1

Retrieving Specific Data

2

Transpose

3

```
#combine unemployment + vaccine report
```

```
unem_vacc = pd.merge(PV_ME_vaccine, unemployment, on = ["State"])  
unem_vacc
```

```
#top 5 state to analyze
```

```
unem_vacc_T5 = pd.merge(unem_vacc, top5cases, on = ['State'], how = 'right')  
unem_vacc_T5
```

Merging Datasets

1

Retrieving Specific Data

2

Transpose

3

```
#Rename the columns we want
```

```
unem_vacc_T5 = unem_vacc_T5.rename(columns={'2021/03 Vaccine Rate':'03_vac', '2021/04 Vaccine Rate':'04_vac',  
                                             '2021/05 Vaccine Rate':'05_vac', '2021/3/1':'03_unem', '2021/4/1':'04_unem', '2021/5/1':'05_unem'})
```

```
unem_vacc_T5
```

```
#Extract the columns we want (unemployment rate and vaccine rate from March to May)
```

```
va_un03_05 = unem_vacc_T5[['Location', '03_unem', '03_vac', '04_unem', '04_vac', '05_unem', '05_vac']]
```

```
va_un03_05
```

Merging Datasets

1

Retrieving Specific Data

2

Transpose

3

	Location	03_unem	03_vac	04_unem	04_vac	05_unem	05_vac
State							
Tennessee	TN	5.1	13.8	5.0	24.5	5.0	31.8
Arizona	AZ	6.7	17.1	6.7	28.9	6.7	36.1
Wisconsin	WI	3.8	18.5	3.9	35.3	3.9	44.6
New Jersey	NJ	7.6	18.7	7.5	36.1	7.2	48.8
Indiana	IN	3.9	16.7	3.9	26.8	4.0	35.5

Merging Datasets

1

Retrieving Specific Data

2

Transpose

3

```
va_un03_05.loc[len(va_un03_05.index)]=['Location', '03_unem', '03_vac', '04_unem', '04_vac', '05_unem', '05_vac']  
va_un03_05
```

	Location	03_unem	03_vac	04_unem	04_vac	05_unem	05_vac
State							
Tennessee	TN	5.1	13.8	5	24.5	5	31.8
Arizona	AZ	6.7	17.1	6.7	28.9	6.7	36.1
Wisconsin	WI	3.8	18.5	3.9	35.3	3.9	44.6
New Jersey	NJ	7.6	18.7	7.5	36.1	7.2	48.8
Indiana	IN	3.9	16.7	3.9	26.8	4	35.5
5	Location	03_unem	03_vac	04_unem	04_vac	05_unem	05_vac

Merging Datasets

1

Retrieving Specific Data

2

Transpose

3

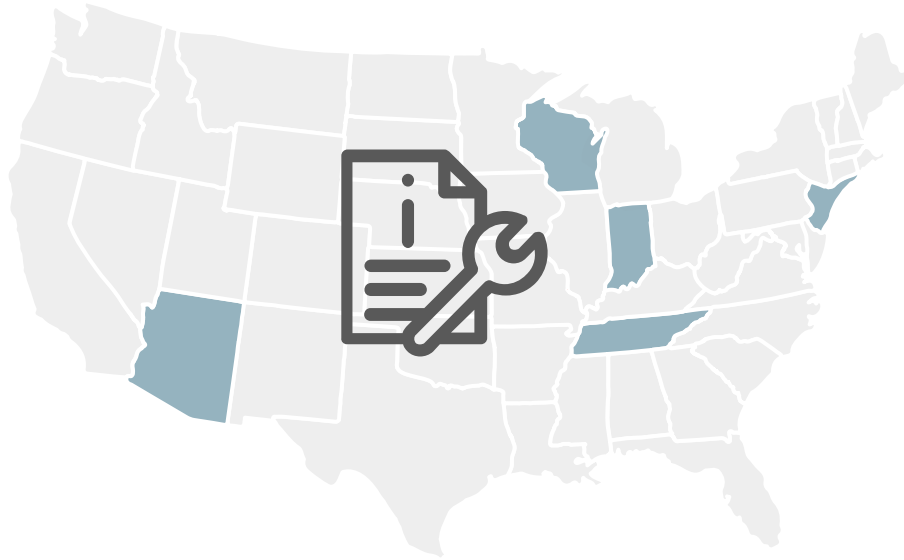
```
va_un03_05_1=va_un03_05.T #Transpose the data  
va_un03_05_1
```

	Location	03_unem	03_vac	04_unem	04_vac	05_unem	05_vac
State							
Tennessee	TN	5.1	13.8	5.0	24.5	5.0	31.8
Arizona	AZ	6.7	17.1	6.7	28.9	6.7	36.1
Wisconsin	WI	3.8	18.5	3.9	35.3	3.9	44.6
New Jersey	NJ	7.6	18.7	7.5	36.1	7.2	48.8
Indiana	IN	3.9	16.7	3.9	26.8	4.0	35.5



State	Tennessee	Arizona	Wisconsin	New Jersey	Indiana	5
Location	TN	AZ	WI	NJ	IN	Location
03_unem	5.1	6.7	3.8	7.6	3.9	03_unem
03_vac	13.8	17.1	18.5	18.7	16.7	03_vac
04_unem	5	6.7	3.9	7.5	3.9	04_unem
04_vac	24.5	28.9	35.3	36.1	26.8	04_vac
05_unem	5	6.7	3.9	7.2	4	05_unem
05_vac	31.8	36.1	44.6	48.8	35.5	05_vac

Data Visualization Using Matplotlib



1

Extract States one at a time

2

Split Columns (Month/Type)

3

Exclude Irrelevant Data

4

**Rename & Convert Data
Type of the Column**

5

Separate Vac and Unem

1

Extract States one at a time

2

Split columns (Month/Type)

3

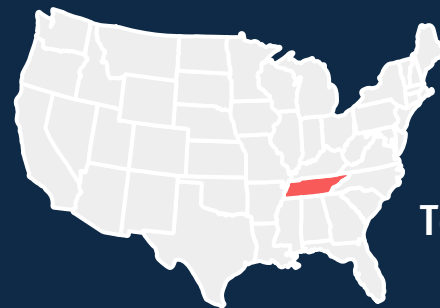
Exclude Irrelevant Data

4

Rename & Convert Data Type
of the Column

5

Separate vac and unem



Tennessee

#Tennessee

```
va_un03_05_TN=va_un03_05_1[['Tennessee',5]] #Extract columns "Tennessee" and 5
```

```
va_un03_05_TN=pd.merge(va_un03_05_TN, (va_un03_05_TN[5].str.split('_',expand=True)),how='left',left_index=True, right_index=True)
```

```
va_un03_05_Tennessee=va_un03_05_TN[1:7] #Exclude the first row, because it is the state abbreviation name
```

```
va_un03_05_Tennessee.rename(columns={"Tennessee":"value",0:"month",1:"type"},inplace=True) #Rename columns
```

```
va_un03_05_Tennessee['value']=va_un03_05_Tennessee['value'].astype('float') #Manage datatype. Column "value" should be float
```

```
va_un03_05_Tennessee
```

State	Tennessee	Arizona	Wisconsin	New Jersey	Indiana	5
Location	TN	AZ	WI	NJ	IN	Location
03_unem	5.1	6.7	3.8	7.6	3.9	03_unem
03_vac	13.8	17.1	18.5	18.7	16.7	03_vac
04_unem	5	6.7	3.9	7.5	3.9	04_unem
04_vac	24.5	28.9	35.3	36.1	26.8	04_vac
05_unem	5	6.7	3.9	7.2	4	05_unem
05_vac	31.8	36.1	44.6	48.8	35.5	05_vac

1

Extract States one at a time

2

Split columns (Month/Type)

3

Rename & Convert Data
Type of the Column

4

Exclude Irrelevant Data

5

Separate vac and unem

```
#Tennessee
va_un03_05_TN=va_un03_05_1[['Tennessee',5]] #Extract columns "Tennessee" and 5
va_un03_05_TN=pd.merge(va_un03_05_TN, (va_un03_05_TN[5].str.split('_', expand=True)),how='left',left_index=True, right_index=True) #Split
va_un03_05_Tennessee=va_un03_05_TN[1:7] #Exclude the first row, because it is the state abbreviation name
va_un03_05_Tennessee.rename(columns={"Tennessee":"value",0:"month",1:"type"},inplace=True) #Rename columns
va_un03_05_Tennessee['value']=va_un03_05_Tennessee['value'].astype('float') #Manage datatype. Column "value" should be float
va_un03_05_Tennessee
```

	value	5	month	type
03_unem	5.1	03_unem	03	unem
03_vac	13.8	03_vac	03	vac

1

Extract States one at a time

2

Split columns (Month/Type)

3

Rename & Convert Data
Type of the Column

4

Exclude Irrelevant Data

5

Separate Vac and Unem

```
#Tennessee
va_un03_05_TN=va_un03_05_1[['Tennessee',5]] #Extract columns "Tennessee" and 5
va_un03_05_TN=pd.merge(va_un03_05_TN, (va_un03_05_TN[5].str.split('_',expand=True)),how='left',left_index=True, right_index=True) #Split
va_un03_05_Tennessee=va_un03_05_TN[1:7] #Exclude the first row, because it is the state abbreviation name
va_un03_05_Tennessee.rename(columns={"Tennessee":"value",0:"month",1:"type"},inplace=True) #Rename columns
va_un03_05_Tennessee['value']=va_un03_05_Tennessee['value'].astype('float') #Manage datatype. Column "value" should be float
va_un03_05_Tennessee
```

State	Tennessee	Arizona	Wisconsin	New Jersey	Indiana
-------	-----------	---------	-----------	------------	---------

Location	TN	AZ	WI	NJ	IN
----------	----	----	----	----	----

1

Extract States one at a time

2

Split columns (Month/Type)

3

Exclude Irrelevant Data

4

Rename & Convert Data
Type of the Column

5

Separate Vac and Unem



#Tennessee

`va_un03_05_TN=va_un03_05_1[['Tennessee',5]] #Extract columns "Tennessee" and 5``va_un03_05_TN=pd.merge(va_un03_05_TN, (va_un03_05_TN[5].str.split('_',expand=True)),how='left',left_index=True, right_index=True) #Split``va_un03_05_Tennessee=va_un03_05_TN[1:7] #Exclude the first row, because it is the state abbreviation name``va_un03_05_Tennessee.rename(columns={"Tennessee":"value",0:"month",1:"type"},inplace=True) #Rename columns``va_un03_05_Tennessee['value']=va_un03_05_Tennessee['value'].astype('float') #Manage datatype. Column "value" should be float``va_un03_05_Tennessee`

1

Extract States one at a time

2

Split columns (Month/Type)

3

Rename & Convert Data
Type of the Column

4

Exclude Irrelevant Data

5

Separate Vac and Unem



```

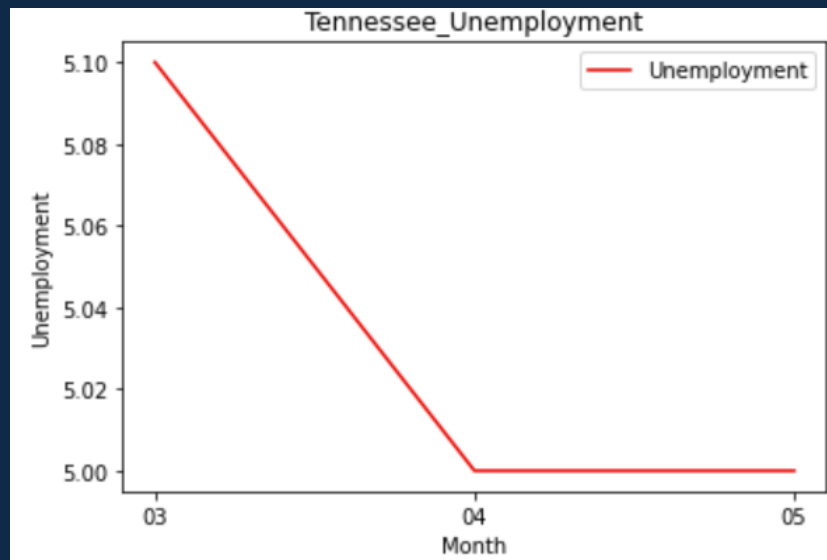
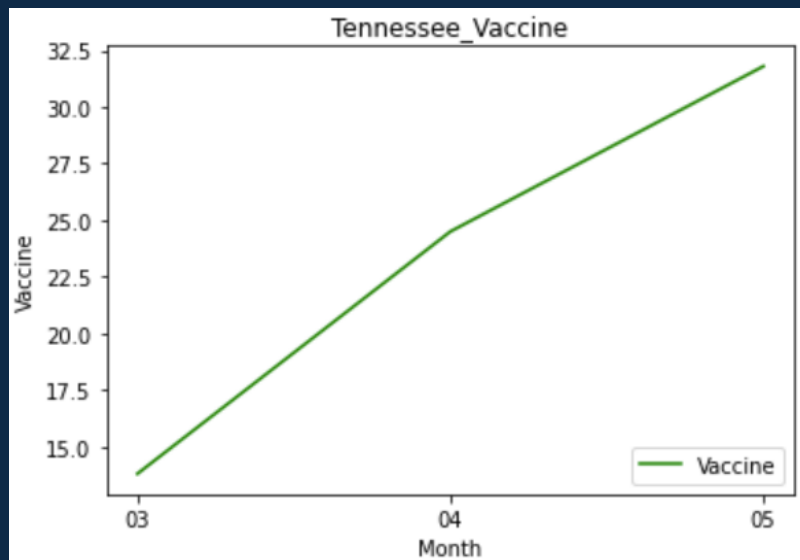
un03_05_Tennessee=va_un03_05_Tennessee.loc[va_un03_05_Tennessee['type']=='unem'] #filter
va03_05_Tennessee=va_un03_05_Tennessee.loc[va_un03_05_Tennessee['type']=='vac']
un03_05_Tennessee
va03_05_Tennessee

```

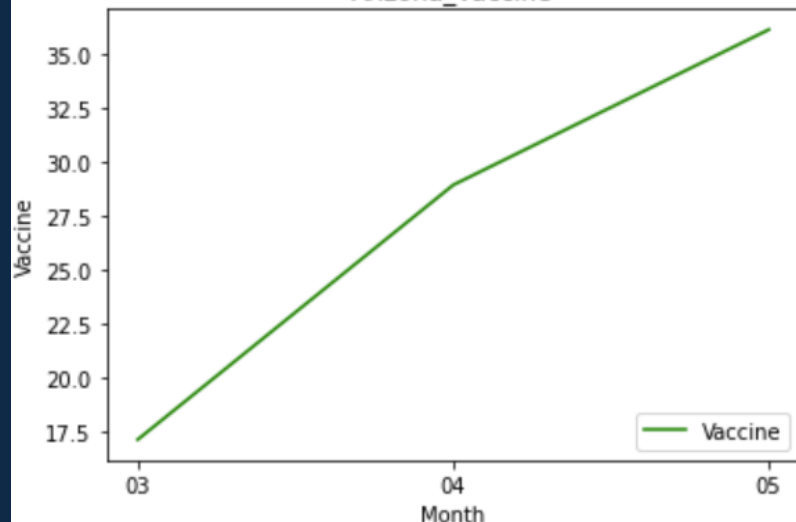
	value	5	month	type
03_vac	13.8	03_vac	03	vac
04_vac	24.5	04_vac	04	vac
05_vac	31.8	05_vac	05	vac

	value	5	month	type
03_unem	5.1	03_unem	03	unem
04_unem	5.0	04_unem	04	unem
05_unem	5.0	05_unem	05	unem

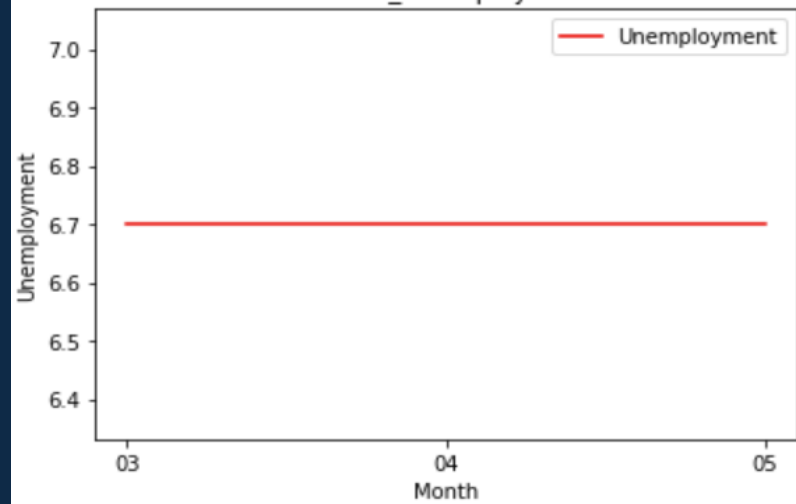
```
import matplotlib.pyplot as plt
plt.plot(va03_05_Tennessee.month, va03_05_Tennessee.value, 'g', label='Vaccine')
plt.xlabel('Month')
plt.ylabel('Vaccine')
plt.xticks(va03_05_Tennessee.month)
plt.title('Tennessee Vaccine')
plt.legend(loc='lower right')
plt.show()
```



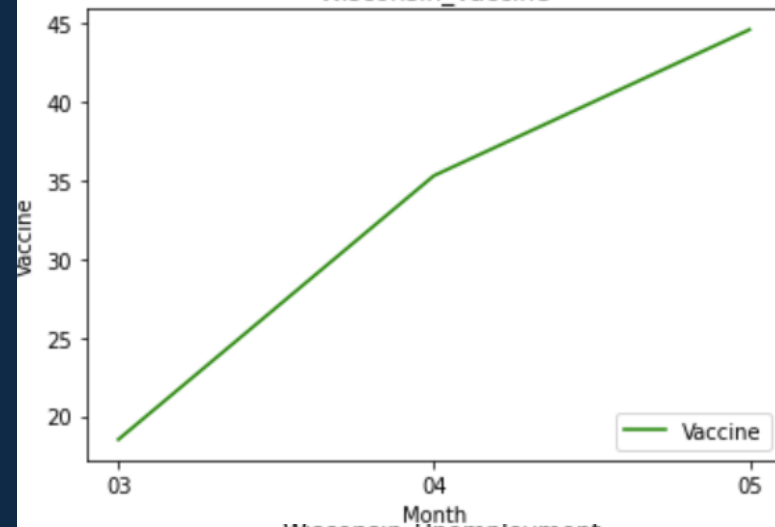
Arizona_Vaccine



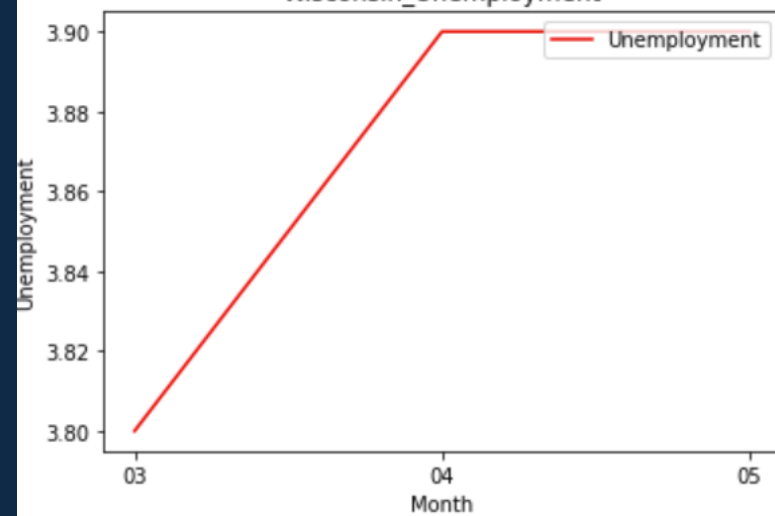
Arizona_Unemployment



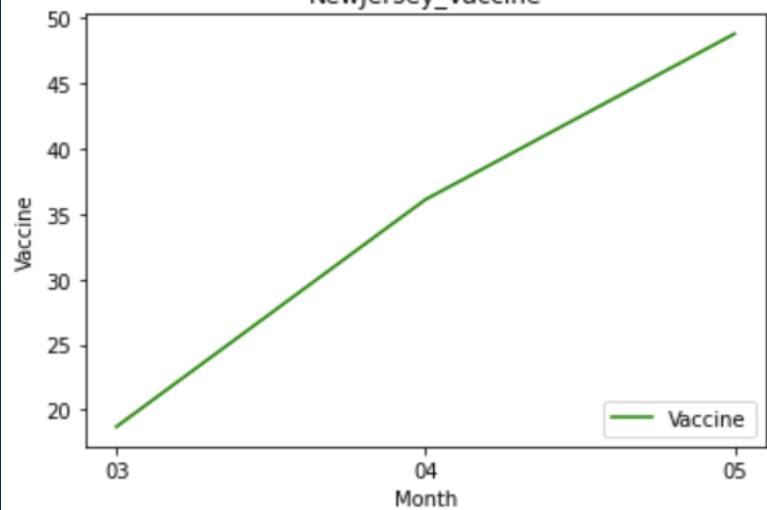
Wisconsin_Vaccine



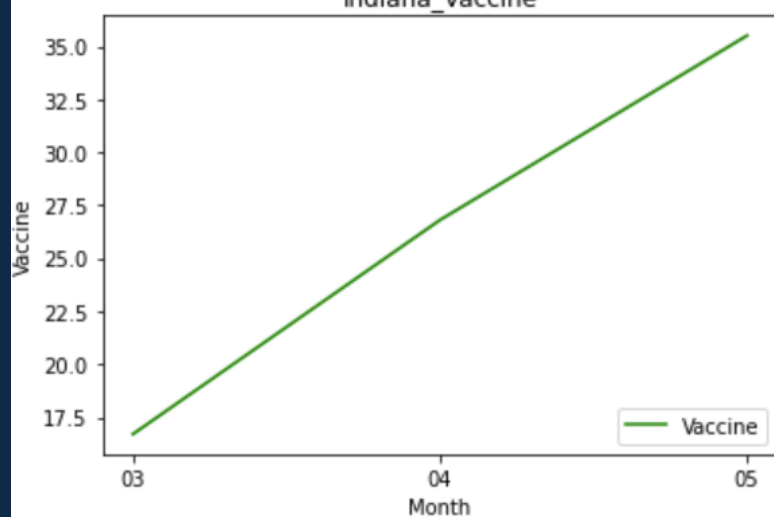
Wisconsin_Unemployment



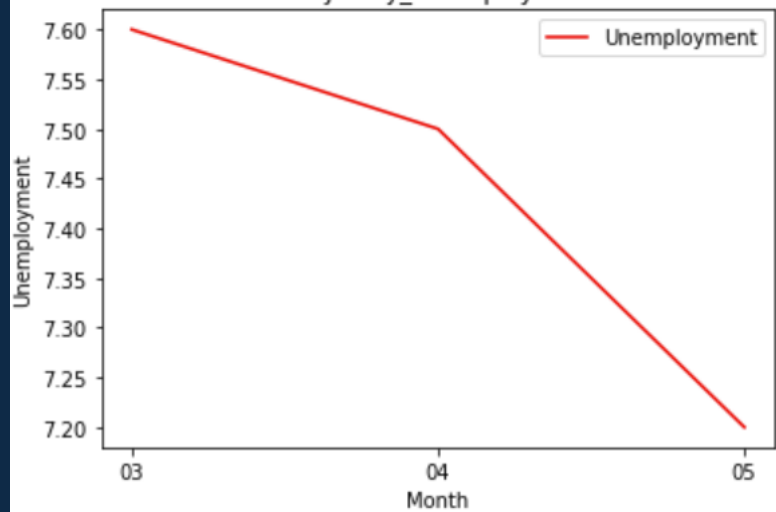
NewJersey_Vaccine



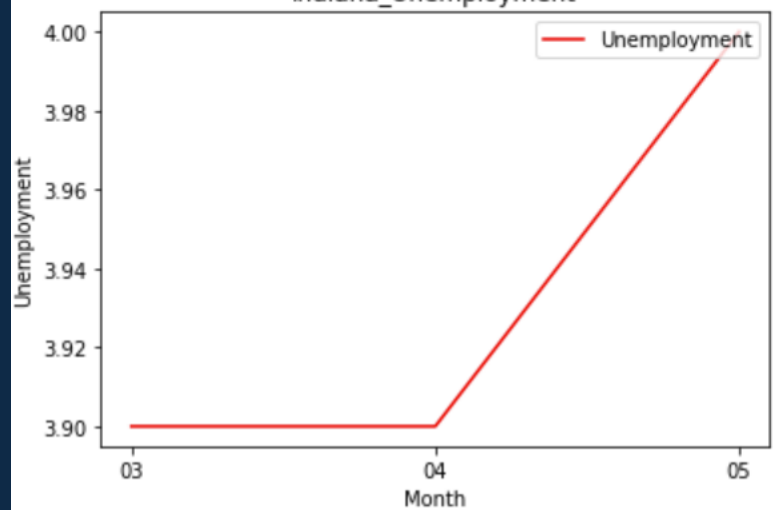
Indiana_Vaccine



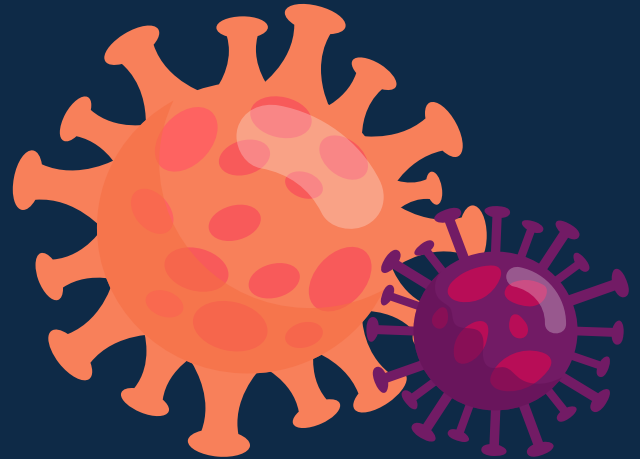
NewJersey_Unemployment



Indiana_Unemployment



The relationship between
Unemployment and Vaccination Rate
is not clear.





Conclusion



1

Current vaccination data available on CDC only covered 3 months, there is limited data to analyze and easily affected by the result of single month

Also, we doubt that the unemployment rate was not affected immediately.

A longer period of observation would help us gain more evidence to verify the relationship between vaccination and unemployment rate.

MONEY

Economy added 850K jobs in June, and unemployment rose to 5.9% as COVID cases declined, states lifted restrictions, vaccinations rose

Paul Davidson USA TODAY

Published 8:32 a.m. ET Jul. 2, 2021 | Updated 3:17 p.m. ET Jul. 2, 2021



Conclusion



2

**Correlation does not imply causation.
More parameters should be considered.**



New York and Other Northeast States See Large Drop in Unemployment

Rising vaccination rates and business reopenings are driving labor-market gains in a region hit hard by the pandemic

THE WALL STREET JOURNAL.

World U.S. Politics **Economy** Business Tech Markets Opinion Life & Arts

the next largest decreases, 0.4 percentage point declines. The unemployment rate in New York was 7.8%, and 7.7% in Connecticut.

High vaccination rates and the related loosening of restrictions are driving the labor market's improvement in the Northeast, economists say. The latest data show a resurgence for a region whose economy was among the most battered when the coronavirus pandemic first took hold in the U.S. in March 2020.

"The Northeastern region has benefited from a rapid rollout of vaccines," said Daniel Zhao, senior economist at job-search site Glassdoor. "The health situation is improving and economic activity is unlocked, allowing workers to return and consumers to go out and spend."

By [Bryan Mena](#)

June 26, 2021 5:30 am ET

Thank you
for your listening !



Sources

- State-economic-monitor:
<https://apps.urban.org/features/state-economic-monitor/>
- CDC: https://covid.cdc.gov/covid-data-tracker/#compare-trends_totalcasesper100k
- <https://data.cdc.gov/Vaccinations/COVID-19-Vaccinations-in-the-United-States-Jurisdi/unsk-b7fc/data>

