Forward
School

Program Code: J620-002-4:2020

Program Name: FRONT-END SOFTWARE DEVELOPMENT

Title : Covid 19 Project

Name: Ooi Caaron

IC Number: 990701-07-5837

Date : 28/6/23

Introduction :

Conclusion : Still need to practice more

Covid 19 Python Project (use all your knowledge thus far to solve this)

From Wikipedia,

Coronavirus disease 2019 (COVID-19) is an infectious disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). The disease was first identified in 2019 in Wuhan, China, and has since spread globally, resulting in the 2019–20 coronavirus pandemic. Common symptoms include fever, cough and shortness of breath. Muscle pain, sputum production and sore throat are less common. The rate of deaths per number of diagnosed cases is on average 3.4%, ranging from 0.2% in those less than 20 to approximately 15% in those over 80 years old.

Data Source (Date wise) : 2019 Novel Coronavirus COVID-19 (2019-nCoV) Data Repository by Johns Hopkins CSSE

Data Source: https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_daily_reports (https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_daily_reports)

File naming convention

MM-DD-YYYY.csv in UTC.

Field description

Province/State: China - province name; US/Canada/Australia/ - city name, state/province name; Others - name of the event (e.g., "Diamond Princess" cruise ship); other countries - blank. Country/Region: country/region name conforming to WHO (will be updated). Last Update: MM/DD/YYYY HH:mm (24 hour format, in UTC). Confirmed: the number of confirmed cases. For Hubei Province: from Feb 13 (GMT +8), we report both clinically diagnosed and lab-confirmed cases. For lab-confirmed cases only (Before Feb 17), please refer to who_covid_19_situation_reports. For Italy, diagnosis standard might be changed since Feb 27 to "slow the growth of new case numbers." (Source) Deaths: the number of deaths. Recovered: the number of recovered cases. Update frequency Files after Feb 1 (UTC): once a day around 23:59 (UTC). Files on and before Feb 1 (UTC): the last updated files before 23:59 (UTC).

In [2]:

```
import numpy as np
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
```

In [22]:

```
path = ('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_d
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
df = pd.read_csv(path)
df
```

Out[22]:

	FIPS	Admin2	Province_State	Country_Region	Last_Update	Lat	Long_	Conf
0	NaN	NaN	NaN	Afghanistan	2020-10-11 04:23:46	33.939110	67.709953	
1	NaN	NaN	NaN	Albania	2020-10-11 04:23:46	41.153300	20.168300	
2	NaN	NaN	NaN	Algeria	2020-10-11 04:23:46	28.033900	1.659600	
3	NaN	NaN	NaN	Andorra	2020-10-11 04:23:46	42.506300	1.521800	
4	NaN	NaN	NaN	Angola	2020-10-11 04:23:46	-11.202700	17.873900	
5	NaN	NaN	NaN	Antigua and Barbuda	2020-10-11 04:23:46	17.060800	-61.796400	

Q1. Write Python code to display first 5 rows from COVID-19 dataset. Also print the dataset information and check the missing values.

In [23]:

```
from IPython.display import display
print(df.isnull().sum())
display(df.head(5))
```

```
FIPS                   730
Admin2                 725
Province_State         176
Country_Region           0
Last_Update              0
Lat                     85
Long_                   85
Confirmed                0
Deaths                   0
Recovered                0
Active                   0
Combined_Key             0
Incidence_Rate          85
Case-Fatality_Ratio     55
dtype: int64
```

	FIPS	Admin2	Province_State	Country_Region	Last_Update	Lat	Long_	Confirr
0	NaN	NaN	NaN	Afghanistan	2020-10-11 04:23:46	33.93911	67.709953	39
1	NaN	NaN	NaN	Albania	2020-10-11 04:23:46	41.15330	20.168300	15
2	NaN	NaN	NaN	Algeria	2020-10-11 04:23:46	28.03390	1.659600	52
3	NaN	NaN	NaN	Andorra	2020-10-11 04:23:46	42.50630	1.521800	2
4	NaN	NaN	NaN	Angola	2020-10-11 04:23:46	-11.20270	17.873900	6

Q2. Write a Python program to get the latest number of confirmed, deaths, recovered and active cases of Novel Coronavirus (COVID-19) Country wise

In [31]:

```
df.groupby("Country_Region")[["Confirmed", "Deaths", "Recovered", "Active"]].sum()
```

Out[31]:

	Confirmed	Deaths	Recovered	Active
Country_Region				
Afghanistan	39789	1477	33064	5248
Albania	15231	416	9406	5409
Algeria	52940	1795	37170	13975
Andorra	2696	55	1814	827
Angola	6246	218	2716	3312
Antarctica	0	0	0	0
Antigua and Barbuda	111	3	97	11
Argentina	883882	23581	709464	150837
Armenia	55736	1016	45771	8949

Q3. Write a Python program to get the Chinese province wise cases of confirmed, deaths and recovered cases of Novel Coronavirus (COVID-19)

In [33]:

```
china = df[df['Country_Region'] == 'China']
china.groupby(["Province_State"])[["Confirmed", "Deaths", "Recovered", "Active"]].sum()
```

Out[33]:

	Confirmed	Deaths	Recovered	Active
Province_State				
Anhui	991	6	985	0
Beijing	936	9	927	0
Chongqing	585	6	578	1
Fujian	415	1	400	14
Gansu	170	2	168	0
Guangdong	1858	8	1823	27
Guangxi	260	2	256	2
Guizhou	147	2	145	0
Hainan	171	6	165	0
Hebei	365	6	358	1
Heilongjiang	948	13	935	0
Henan	1281	22	1255	4
Hong Kong	5175	105	4914	156
Hubei	68139	4512	63627	0
Hunan	1019	4	1015	0
Inner Mongolia	268	1	261	6
Jiangsu	667	0	664	3
Jiangxi	935	1	934	0
Jilin	157	2	155	0
Liaoning	276	2	269	5
Macau	46	0	46	0
Ningxia	75	0	75	0
Qinghai	18	0	18	0
Shaanxi	428	3	397	28
Shandong	832	7	824	1
Shanghai	1048	7	980	61
Shanxi	206	0	203	3
Sichuan	721	3	673	45
Tianjin	244	3	236	5
Tibet	1	0	1	0
Unknown	5201	0	0	0
Xinjiang	902	3	899	0
Yunnan	211	2	200	9
Zhejiang	1283	1	1272	10

Q4. Write a Python program to get the latest country wise deaths cases of Novel Coronavirus (COVID-19)

In [38]:

```
df.groupby(['Country_Region'])['Deaths'].sum()
```

Out[38]:

```
Country_Region
Afghanistan                      1477
Albania                           416
Algeria                          1795
Andorra                            55
Angola                            218
Antarctica                          0
Antigua and Barbuda                 3
Argentina                       23581
Armenia                          1016
Australia                         898
Austria                          1072
Azerbaijan                        608
Bahamas                           106
Bahrain                           273
Bangladesh                       5500
Barbados                            7
Belarus                           885
```

Q5. Write a Python program to list countries with no cases of Novel Coronavirus (COVID-19) recovered

In [59]:

```
recovered = df.groupby(['Country_Region'])[['Confirmed','Deaths','Recovered','Active']].s
recovered[recovered['Recovered']==0]
```

Out[59]:

	Confirmed	Deaths	Recovered	Active
Country_Region				
Antarctica	0	0	0	0
Kiribati	0	0	0	0
Korea, North	0	0	0	0
MS Zaandam	9	2	0	7
Nauru	0	0	0	0
Palau	0	0	0	0
Samoa	0	0	0	0
Serbia	34685	762	0	33923
Summer Olympics 2020	0	0	0	0
Sweden	98451	5894	0	92557
Tonga	0	0	0	0
Tuvalu	0	0	0	0
Winter Olympics 2022	0	0	0	0

Q6. Write a Python program to get the latest number of confirmed deaths and recovered people of Novel Coronavirus (COVID-19) cases Country/Region - Province/State wise.

In [64]:

```
df.groupby(['Country_Region','Province_State'])[['Confirmed','Deaths','Recovered']].sum()
```

Out[64]:

Country_Region	Province_State	Confirmed	Deaths	Recovered
Australia	Australian Capital Territory	113	3	110
	New South Wales	4278	53	0
	Northern Territory	33	0	33
	Queensland	1161	6	1152
	South Australia	473	4	466
	Tasmania	230	13	217
	Victoria	20281	810	19207
	Western Australia	694	9	670
Brazil	Acre	29031	674	26979

Q7. Write a Python program to list countries with all cases of Novel Coronavirus (COVID-19) died

In [152]:

```
d_cases = df.groupby(['Country_Region'])[['Confirmed', 'Deaths', 'Recovered', 'Active']].
d_cases = d_cases[d_cases['Deaths'] == d_cases['Confirmed']]
d_cases = d_cases[d_cases['Confirmed'] > 0]
d_cases = d_cases[d_cases['Active']==0]
d_cases = d_cases[d_cases['Recovered']==0]
print(d_cases)
```

```
Empty DataFrame
Columns: [Country_Region, Confirmed, Deaths, Recovered, Active]
Index: []
```

Q8.Write a Python program to list countries with all cases of Novel Coronavirus (COVID-19) recovered.

In [151]:

```
# recovered = df.groupby(['Country_Region'])[['Confirmed','Deaths','Recovered','Active']]
# recovered[recovered['Recovered']>0]
d_cases = df.groupby(['Country_Region'])[['Confirmed', 'Deaths', 'Recovered',"Active"]].s
d_cases = d_cases[d_cases['Recovered'] == d_cases['Confirmed']]
d_cases = d_cases[d_cases['Confirmed'] > 0]
d_cases = d_cases[d_cases['Active']==0]
d_cases = d_cases[d_cases['Deaths']==0]
print(d_cases)
```

```
                      Country_Region  Confirmed  Deaths  Recovered  Activ
e
70                           Grenada         24       0         24  
0
76                          Holy See         12       0         12  
0
149  Saint Vincent and the Grenadines         64       0         64  
0
176                      Timor-Leste         28       0         28  
0
```

Q9. Write a Python program to get the top 10 countries data (Last Update, Country/Region, Confirmed, Deaths, Recovered) of Novel Coronavirus (COVID-19).

In [95]:

```
df.groupby('Country_Region')[['Confirmed','Deaths','Recovered']].sum().sort_values(by='Co
```

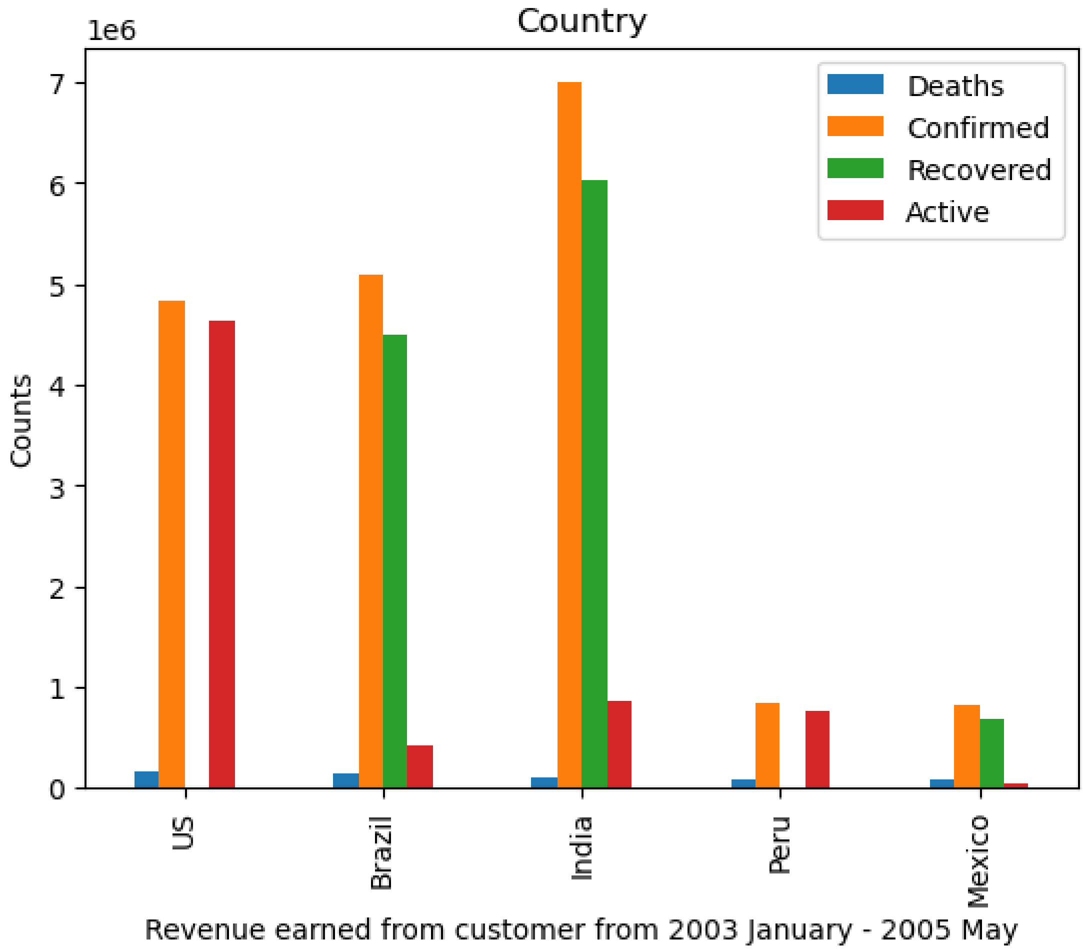
Out[95]:

	Confirmed	Deaths	Recovered
Country_Region			
US	7753210	213324	3062983
India	7053806	108334	6077976
Brazil	5084513	150302	4502854
Russia	1278245	22331	1011911
Colombia	902747	27660	783131
Argentina	883882	23581	709464
Spain	861112	32929	150376
Peru	846088	83825	733000
Mexico	814328	83642	689377
France	761384	32684	103232

Q10. Write a Python program to create a plot (lines) of total deaths, confirmed, recovered and active cases Country wise where deaths greater than 150.

In [114]:

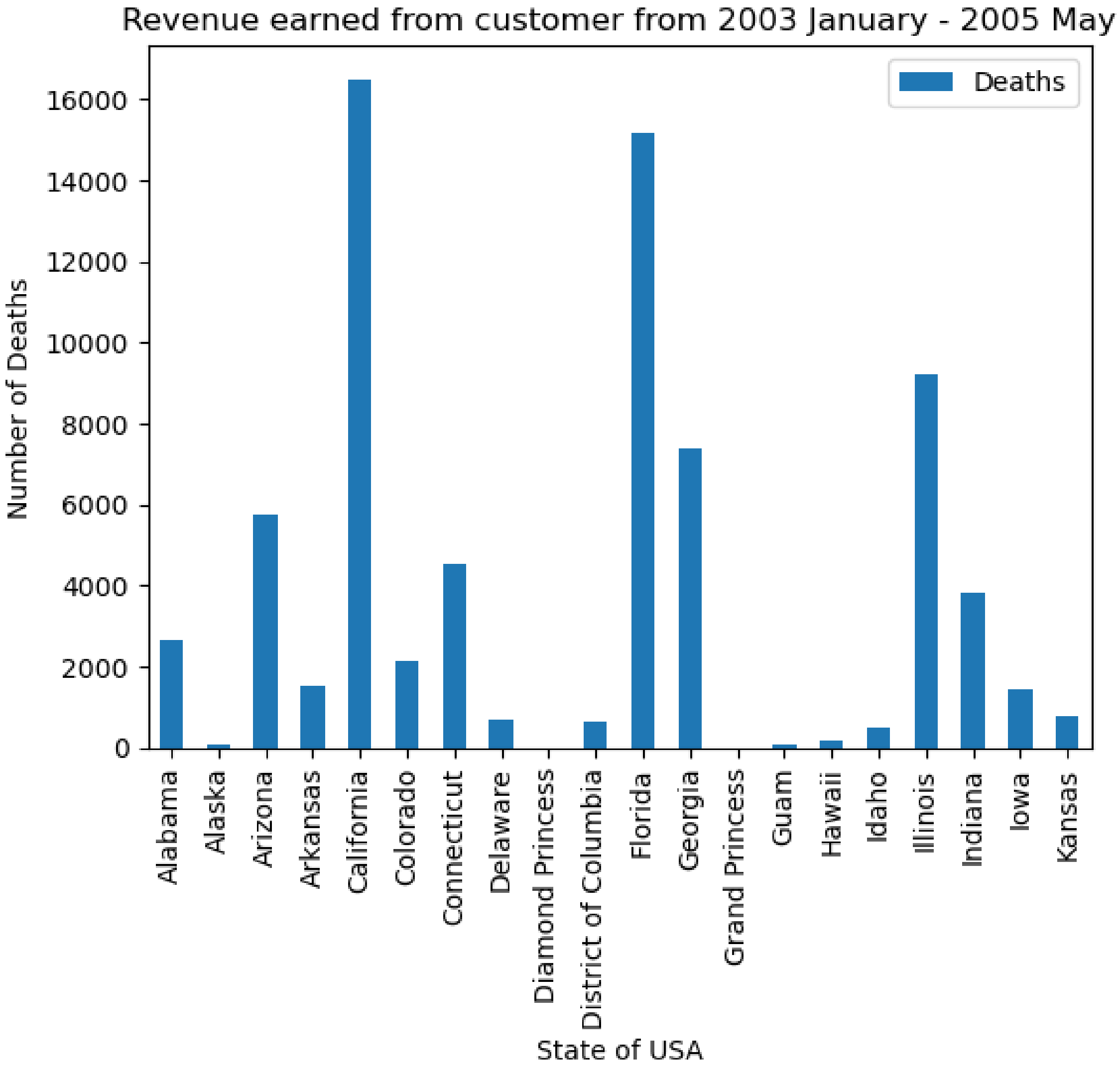
```
most = df[df['Deaths'] > 150]
country = most.groupby(['Country_Region'])[['Deaths','Confirmed', 'Recovered','Active']].

country.plot.bar()
plt.xlabel('Revenue earned from customer from 2003 January - 2005 May')
plt.ylabel('Counts')
plt.title('Country')
plt.show()
```



Q.11 Write a Python program to visualize the state/province wise death cases of Novel Coronavirus (COVID-19) in USA

In [125]:

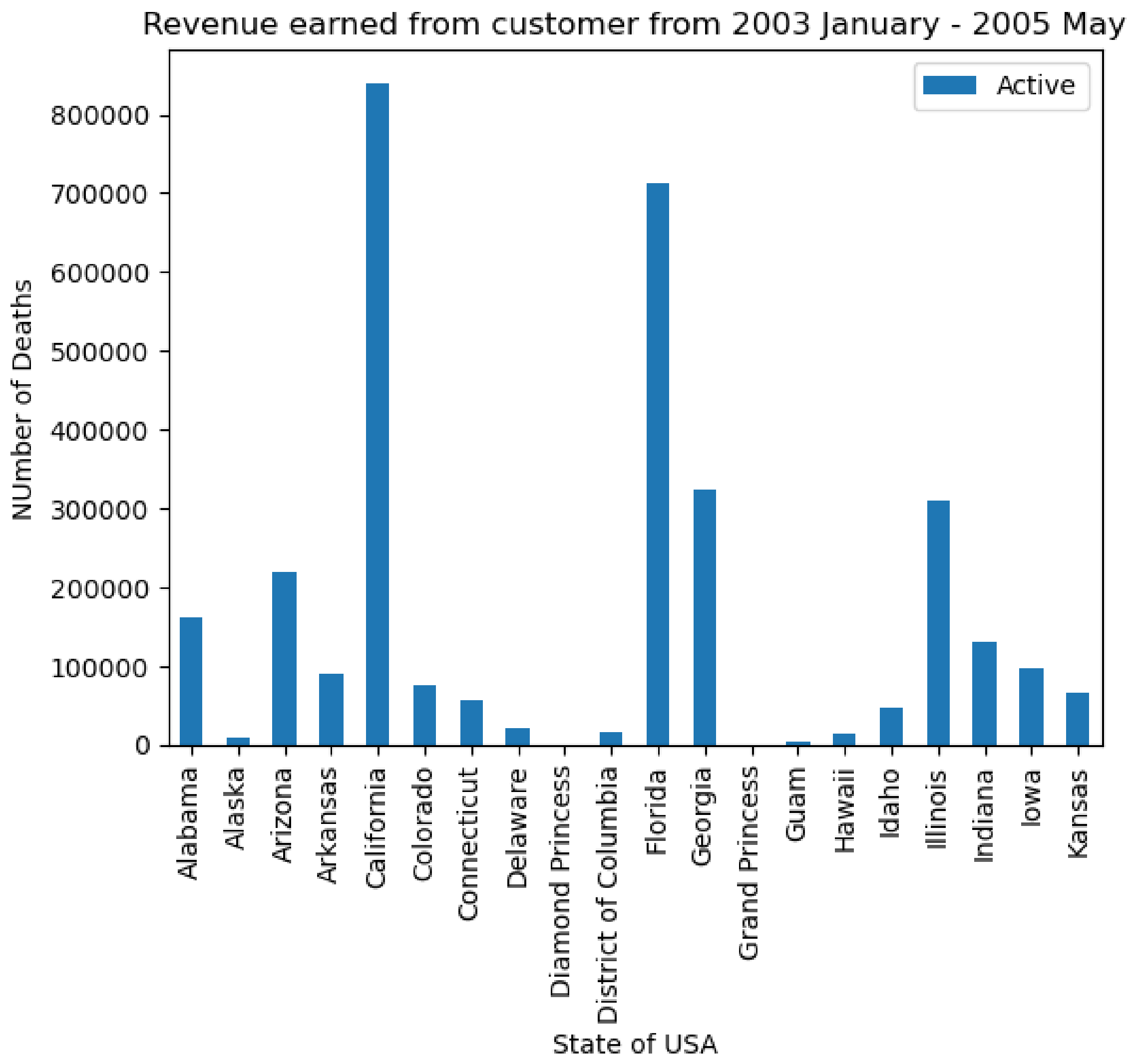
```
us = df[df['Country_Region'] == 'US']
died = us.groupby(["Province_State"])[["Deaths"]].sum().head(20)
died.plot.bar()
plt.xlabel('State of USA')
plt.ylabel('Number of Deaths')
plt.title('Revenue earned from customer from 2003 January - 2005 May')
plt.show()
```



Q.12 Write a Python program to visualize the state/province wise Active cases of Novel Coronavirus (COVID-19) in USA

In [124]:

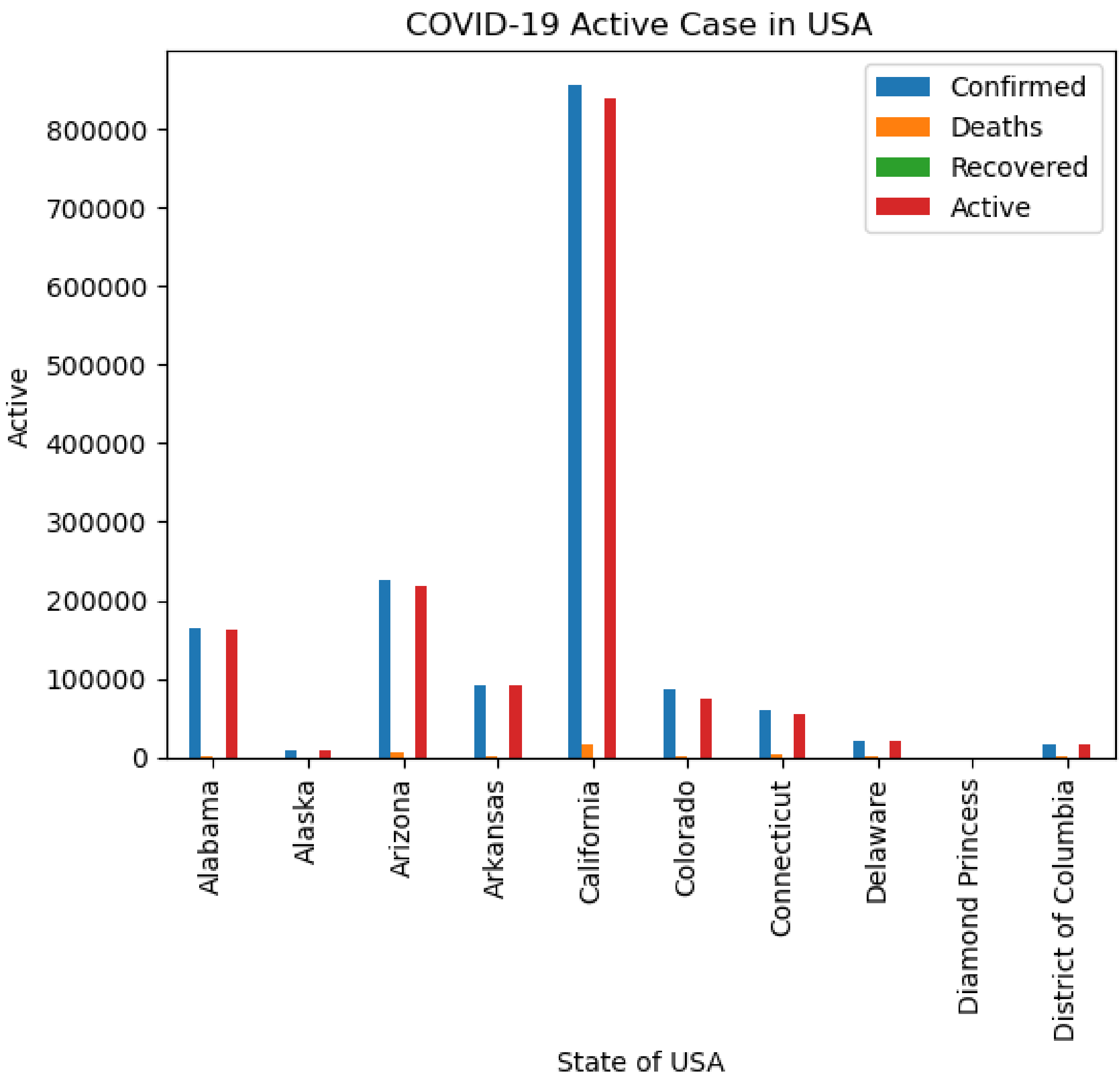
```
us = df[df['Country_Region'] == 'US']
died = us.groupby(["Province_State"])[["Active"]].sum().head(20)
died.plot.bar()
plt.xlabel('State of USA')
plt.ylabel('Number of Deaths')
plt.title('Revenue earned from customer from 2003 January - 2005 May')
plt.show()
```



Q.13 Write a Python program to visualize the state/province wise combine number of confirmed, deaths, recovered, active Novel Coronavirus (COVID-19) cases in USA.

In [128]:

```
us = df[df['Country_Region'] == 'US']
died = us.groupby(["Province_State"])[['Confirmed','Deaths','Recovered','Active']].sum().
died.plot.bar()
plt.xlabel('State of USA')
plt.ylabel('Active')
plt.title('COVID-19 Active Case in USA')
plt.show()
```



Q.14 Write a Python program to visualize Worldwide Confirmed Novel Coronavirus (COVID-19) cases over time

In [ ]:

```
#no need
```

In [ ]:

```
```