Covid 19 Vaccine Analysis Milestone 4

October 2, 2024

0.1 Connect to API

[214]:
```
import requests
from datetime import datetime, timedelta
import pandas as pd
```

[215]:
```
country = 'USA'
```

[216]:
```
region = 'California'
```

[217]:
```
date = '2020-3-10'
```

[218]:
```
url = 'https://api.api-ninjas.com/v1/covid19?country={}&region={}&date={}'.
↪format(country, region, date)
```

[219]:
```
apikey = 'c7+4Fg6YFu+1z+x0BDQ7+A==syIxjQ7Riw1pnZD8'
```

[220]:
```
headers = {'X-Api-Key': apikey}
```

[221]:
```
response = requests.get(url, headers=headers)
```

[222]:
```
if response.status_code == 200:
    data = response.json()
    # print(type(data))
    # print(len(data))
    df = pd.DataFrame(data)
    # print(df)
else:
    print("Failed to fetch data. Status code:", response.status_code)
```

[223]:
```
# print(df.iloc[0]["county"])
```

[224]:
```
cases=df.iloc[0]["cases"]
```

[225]:
```
cases_df = pd.DataFrame(cases.items(), columns=["key", "value"])
```

[226]:
```
# cases_df["key"]
```

[227]:
```
# cases_df["value"][0]
```

[228]:
```
# type(cases_df["value"][0])
```

[229]:
```
cases_df[['sub_key', 'sub_value']] = cases_df['value'].apply(lambda x: pd.
 ↪Series(list(x.items())[1]))
cases_df['county']=df.iloc[0]["county"]
```

[230]:
```
print(cases_df)
```

```
            key                            value sub_key  sub_value   county
0     2020-01-22         {'total': 0, 'new': 0}     new          0  Alameda
1     2020-01-23         {'total': 0, 'new': 0}     new          0  Alameda
2     2020-01-24         {'total': 0, 'new': 0}     new          0  Alameda
3     2020-01-25         {'total': 0, 'new': 0}     new          0  Alameda
4     2020-01-26         {'total': 0, 'new': 0}     new          0  Alameda
…          …                                …     …         …       …
1138  2023-03-05    {'total': 401161, 'new': 0}     new          0  Alameda
1139  2023-03-06    {'total': 401161, 'new': 0}     new          0  Alameda
1140  2023-03-07  {'total': 402158, 'new': 997}     new        997  Alameda
1141  2023-03-08    {'total': 402158, 'new': 0}     new          0  Alameda
1142  2023-03-09    {'total': 402160, 'new': 2}     new          2  Alameda

[1143 rows x 5 columns]
```

[231]:
```
for i in range(1,len(data)):
    cases=df.iloc[i]["cases"]
    cases_df_tmp = pd.DataFrame(cases.items(), columns=["key", "value"])
    cases_df_tmp[['sub_key', 'sub_value']] = cases_df_tmp['value'].apply(lambda␣
 ↪x: pd.Series(list(x.items())[1]))
    cases_df[df.iloc[i]["county"]]=cases_df_tmp['sub_value']
```

[232]:
```
cases_df
```

[232]:
```
            key                            value sub_key  sub_value   county  \
0     2020-01-22         {'total': 0, 'new': 0}     new          0  Alameda
1     2020-01-23         {'total': 0, 'new': 0}     new          0  Alameda
2     2020-01-24         {'total': 0, 'new': 0}     new          0  Alameda
3     2020-01-25         {'total': 0, 'new': 0}     new          0  Alameda
4     2020-01-26         {'total': 0, 'new': 0}     new          0  Alameda
…          …                                …     …         …       …
1138  2023-03-05    {'total': 401161, 'new': 0}     new          0  Alameda
1139  2023-03-06    {'total': 401161, 'new': 0}     new          0  Alameda
1140  2023-03-07  {'total': 402158, 'new': 997}     new        997  Alameda
1141  2023-03-08    {'total': 402158, 'new': 0}     new          0  Alameda
1142  2023-03-09    {'total': 402160, 'new': 2}     new          2  Alameda

      Alpine  Amador  Butte  Calaveras  Colusa  …  Stanislaus  Sutter  \
```

```
0          0       0      0          0       0  …           0       0
1          0       0      0          0       0  …           0       0
2          0       0      0          0       0  …           0       0
3          0       0      0          0       0  …           0       0
4          0       0      0          0       0  …           0       0
…        …     …    …        …     …  …         …     …
1138       0       0      0          0       0  …           0       0
1139       0       0      0          0       0  …           0       0
1140       0      13     51         16      11  …           0      40
1141       0       0      0          0       0  …           0       0
1142       0       0      0          0       5  …         249       0

      Tehama  Trinity  Tulare  Tuolumne  Unassigned  Ventura  Yolo  Yuba
0          0        0       0         0           0        0     0     0
1          0        0       0         0           0        0     0     0
2          0        0       0         0           0        0     0     0
3          0        0       0         0           0        0     0     0
4          0        0       0         0           0        0     0     0
…        …      …     …       …         …       …    …   …
1138       0        0       0         0           0        0     0     0
1139       0        0       0         0           0        0     0     0
1140      17        2     247        43           0      269    69    40
1141       0        0       0         0           0        0     0     0
1142       0        0       2         0          -7        5     3     0

[1143 rows x 64 columns]
```

[233]:
```
cases_df['new_case_sums'] = cases_df.sum(axis=1)
```

```
/var/folders/6q/k8jdwbv174s78xj3x3kzvn0w0000gn/T/ipykernel_37936/1710462041.py:1
: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with
'numeric_only=None') is deprecated; in a future version this will raise
TypeError.  Select only valid columns before calling the reduction.
  cases_df['new_case_sums'] = cases_df.sum(axis=1)
```

[234]:
```
cases_df
```

[234]:
```
            key                              value sub_key  sub_value   county  \
0     2020-01-22         {'total': 0, 'new': 0}     new          0  Alameda
1     2020-01-23         {'total': 0, 'new': 0}     new          0  Alameda
2     2020-01-24         {'total': 0, 'new': 0}     new          0  Alameda
3     2020-01-25         {'total': 0, 'new': 0}     new          0  Alameda
4     2020-01-26         {'total': 0, 'new': 0}     new          0  Alameda
…          …                                …     …        …      …
1138  2023-03-05    {'total': 401161, 'new': 0}     new          0  Alameda
1139  2023-03-06    {'total': 401161, 'new': 0}     new          0  Alameda
1140  2023-03-07  {'total': 402158, 'new': 997}     new        997  Alameda
1141  2023-03-08    {'total': 402158, 'new': 0}     new          0  Alameda
```

```
1142  2023-03-09    {'total': 402160, 'new': 2}     new           2  Alameda

      Alpine  Amador  Butte  Calaveras  Colusa  …  Sutter  Tehama  Trinity  \
0          0       0      0          0       0  …       0       0        0
1          0       0      0          0       0  …       0       0        0
2          0       0      0          0       0  …       0       0        0
3          0       0      0          0       0  …       0       0        0
4          0       0      0          0       0  …       0       0        0
…        …     …    …         …     …  …     …    …       …
1138       0       0      0          0       0  …       0       0        0
1139       0       0      0          0       0  …       0       0        0
1140       0      13     51         16      11  …      40      17        2
1141       0       0      0          0       0  …       0       0        0
1142       0       0      0          0       5  …       0       0        0

      Tulare  Tuolumne  Unassigned  Ventura  Yolo  Yuba  new_case_sums
0          0         0           0        0     0     0              0
1          0         0           0        0     0     0              0
2          0         0           0        0     0     0              0
3          0         0           0        0     0     0              0
4          0         0           0        0     0     0              2
…        …       …         …       …   …   …            …
1138       0         0           0        0     0     0              0
1139       0         0           0        0     0     0              0
1140     247        43           0      269    69    40          10786
1141       0         0           0        0     0     0              0
1142       2         0          -7        5     3     0           8734

[1143 rows x 65 columns]
```

[235]:
```
# step 1 pull data from API and form a dataset
```

[236]:
```
region1 = 'Arizona'
```

[237]:
```
url1 = 'https://api.api-ninjas.com/v1/covid19?country={}&region={}'.
 ↪format(country, region1)
```

[238]:
```
response1 = requests.get(url1, headers=headers)
```

[239]:
```
if response1.status_code == 200:
    data1 = response1.json()
    # print(type(data))
    # print(len(data))
    df1 = pd.DataFrame(data1)
else:
    print("Failed to fetch data. Status code:", response1.status_code)
```

[240]:
```
cases_df1 = pd.DataFrame(cases.items(), columns=["key", "value"])
cases_df1
```

[240]:

	key	value
0	2020-01-22	{'total': 0, 'new': 0}
1	2020-01-23	{'total': 0, 'new': 0}
2	2020-01-24	{'total': 0, 'new': 0}
3	2020-01-25	{'total': 0, 'new': 0}
4	2020-01-26	{'total': 0, 'new': 0}
…	…	…
1138	2023-03-05	{'total': 21884, 'new': 0}
1139	2023-03-06	{'total': 21884, 'new': 0}
1140	2023-03-07	{'total': 21924, 'new': 40}
1141	2023-03-08	{'total': 21924, 'new': 0}
1142	2023-03-09	{'total': 21924, 'new': 0}

[1143 rows x 2 columns]

[241]:
```
cases_df1[['sub_key', 'sub_value']] = cases_df1['value'].apply(lambda x: pd.
 ↪Series(list(x.items())[1]))
cases_df1['county']=df1.iloc[0]["county"]
```

[242]:
```
for i in range(1,len(data1)):
    cases1=df1.iloc[i]["cases"]
    cases_df1_tmp = pd.DataFrame(cases.items(), columns=["key", "value"])
    cases_df1_tmp[['sub_key', 'sub_value']] = cases_df1_tmp['value'].
 ↪apply(lambda x: pd.Series(list(x.items())[1]))
    cases_df1[df1.iloc[i]["county"]]=cases_df1_tmp['sub_value']
```

[243]:
```
cases_df1
```

[243]:

	key	value	sub_key	sub_value	county	\
0	2020-01-22	{'total': 0, 'new': 0}	new	0	Apache	
1	2020-01-23	{'total': 0, 'new': 0}	new	0	Apache	
2	2020-01-24	{'total': 0, 'new': 0}	new	0	Apache	
3	2020-01-25	{'total': 0, 'new': 0}	new	0	Apache	
4	2020-01-26	{'total': 0, 'new': 0}	new	0	Apache	
…	…	…	…	…	…	
1138	2023-03-05	{'total': 21884, 'new': 0}	new	0	Apache	
1139	2023-03-06	{'total': 21884, 'new': 0}	new	0	Apache	
1140	2023-03-07	{'total': 21924, 'new': 40}	new	40	Apache	
1141	2023-03-08	{'total': 21924, 'new': 0}	new	0	Apache	
1142	2023-03-09	{'total': 21924, 'new': 0}	new	0	Apache	

	Cochise	Coconino	Gila	Graham	Greenlee	…	Maricopa	Mohave	\
0	0	0	0	0	0	…	0	0	
1	0	0	0	0	0	…	0	0	

```
2           0         0     0       0         0  …         0       0
3           0         0     0       0         0  …         0       0
4           0         0     0       0         0  …         0       0
…         …         …   …     …         …  …         …       …
1138        0         0     0       0         0  …         0       0
1139        0         0     0       0         0  …         0       0
1140       40        40    40      40        40  …        40      40
1141        0         0     0       0         0  …         0       0
1142        0         0     0       0         0  …         0       0

      Navajo  Out of AZ  Pima  Pinal  Santa Cruz  Unassigned  Yavapai  Yuma
0          0          0     0      0           0           0        0     0
1          0          0     0      0           0           0        0     0
2          0          0     0      0           0           0        0     0
3          0          0     0      0           0           0        0     0
4          0          0     0      0           0           0        0     0
…        …         …   …     …         …       …       …   …
1138       0          0     0      0           0           0        0     0
1139       0          0     0      0           0           0        0     0
1140      40         40    40     40          40          40       40    40
1141       0          0     0      0           0           0        0     0
1142       0          0     0      0           0           0        0     0

[1143 rows x 21 columns]
```

[244]:
```
# step 2 get sum of rows and add a new sum column
```

[245]:
```
cases_df1['new_case_sums'] = cases_df1.sum(axis=1)
```

```
/var/folders/6q/k8jdwbv174s78xj3x3kzvn0w0000gn/T/ipykernel_37936/2143349022.py:1
: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with
'numeric_only=None') is deprecated; in a future version this will raise
TypeError.  Select only valid columns before calling the reduction.
  cases_df1['new_case_sums'] = cases_df1.sum(axis=1)
```

[246]:
```
cases_df1
```

[246]:
```
            key                          value sub_key  sub_value  county  \
0    2020-01-22       {'total': 0, 'new': 0}     new          0  Apache
1    2020-01-23       {'total': 0, 'new': 0}     new          0  Apache
2    2020-01-24       {'total': 0, 'new': 0}     new          0  Apache
3    2020-01-25       {'total': 0, 'new': 0}     new          0  Apache
4    2020-01-26       {'total': 0, 'new': 0}     new          0  Apache
…         …                            …     …        …     …
1138 2023-03-05   {'total': 21884, 'new': 0}     new          0  Apache
1139 2023-03-06   {'total': 21884, 'new': 0}     new          0  Apache
1140 2023-03-07  {'total': 21924, 'new': 40}     new         40  Apache
1141 2023-03-08   {'total': 21924, 'new': 0}     new          0  Apache
```

```
1142  2023-03-09   {'total': 21924, 'new': 0}     new          0  Apache

      Cochise  Coconino  Gila  Graham  Greenlee  …  Mohave  Navajo  \
0           0         0     0       0         0  …       0       0
1           0         0     0       0         0  …       0       0
2           0         0     0       0         0  …       0       0
3           0         0     0       0         0  …       0       0
4           0         0     0       0         0  …       0       0
…         …       …   …     …       …  …     …       …
1138        0         0     0       0         0  …       0       0
1139        0         0     0       0         0  …       0       0
1140       40        40    40      40        40  …      40      40
1141        0         0     0       0         0  …       0       0
1142        0         0     0       0         0  …       0       0

      Out of AZ  Pima  Pinal  Santa Cruz  Unassigned  Yavapai  Yuma  \
0             0     0      0           0           0        0     0
1             0     0      0           0           0        0     0
2             0     0      0           0           0        0     0
3             0     0      0           0           0        0     0
4             0     0      0           0           0        0     0
…           …   …     …          …          …      …   …
1138          0     0      0           0           0        0     0
1139          0     0      0           0           0        0     0
1140         40    40     40          40          40       40    40
1141          0     0      0           0           0        0     0
1142          0     0      0           0           0        0     0

      new_case_sums
0                 0
1                 0
2                 0
3                 0
4                 0
…               …
1138              0
1139              0
1140            680
1141              0
1142              0

[1143 rows x 22 columns]
```

[247]:
```
# repeat step 1 and 2
```

[248]:
```
region2 = 'Idaho'
```

[249]:
```
url2 = 'https://api.api-ninjas.com/v1/covid19?country={}&region={}'.
 ↪format(country, region2)
```

[250]:
```
response2 = requests.get(url2, headers=headers)
```

[251]:
```
if response2.status_code == 200:
    data2 = response2.json()
    # print(type(data))
    # print(len(data))
    df2 = pd.DataFrame(data2)
else:
    print("Failed to fetch data. Status code:", response2.status_code)
```

[252]:
```
cases_df2 = pd.DataFrame(cases.items(), columns=["key", "value"])
cases_df2
```

[252]:

	key	value
0	2020-01-22	{'total': 0, 'new': 0}
1	2020-01-23	{'total': 0, 'new': 0}
2	2020-01-24	{'total': 0, 'new': 0}
3	2020-01-25	{'total': 0, 'new': 0}
4	2020-01-26	{'total': 0, 'new': 0}
…	…	…
1138	2023-03-05	{'total': 21884, 'new': 0}
1139	2023-03-06	{'total': 21884, 'new': 0}
1140	2023-03-07	{'total': 21924, 'new': 40}
1141	2023-03-08	{'total': 21924, 'new': 0}
1142	2023-03-09	{'total': 21924, 'new': 0}

[1143 rows x 2 columns]

[253]:
```
for i in range(1,len(data2)):
    cases2=df2.iloc[i]["cases"]
    cases_df2_tmp = pd.DataFrame(cases.items(), columns=["key", "value"])
    cases_df2_tmp[['sub_key', 'sub_value']] = cases_df2_tmp['value'].
 ↪apply(lambda x: pd.Series(list(x.items())[1]))
    cases_df2[df2.iloc[i]["county"]]=cases_df2_tmp['sub_value']
```

[254]:
```
cases_df2[['sub_key', 'sub_value']] = cases_df2['value'].apply(lambda x: pd.
 ↪Series(list(x.items())[1]))
cases_df2['county']=df2.iloc[0]["county"]
```

[255]:
```
cases_df2['new_case_sums'] = cases_df2.sum(axis=1)
```

```
/var/folders/6q/k8jdwbv174s78xj3x3kzvn0w0000gn/T/ipykernel_37936/3685147226.py:1
: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with
'numeric_only=None') is deprecated; in a future version this will raise
TypeError.  Select only valid columns before calling the reduction.
```

```
cases_df2['new_case_sums'] = cases_df2.sum(axis=1)
```

[256]:
```
cases_df2
```

[256]:

	key	value	Adams	Bannock	Bear Lake	\
0	2020-01-22	{'total': 0, 'new': 0}	0	0	0	
1	2020-01-23	{'total': 0, 'new': 0}	0	0	0	
2	2020-01-24	{'total': 0, 'new': 0}	0	0	0	
3	2020-01-25	{'total': 0, 'new': 0}	0	0	0	
4	2020-01-26	{'total': 0, 'new': 0}	0	0	0	
…	…	…	…	…	…	
1138	2023-03-05	{'total': 21884, 'new': 0}	0	0	0	
1139	2023-03-06	{'total': 21884, 'new': 0}	0	0	0	
1140	2023-03-07	{'total': 21924, 'new': 40}	40	40	40	
1141	2023-03-08	{'total': 21924, 'new': 0}	0	0	0	
1142	2023-03-09	{'total': 21924, 'new': 0}	0	0	0	

	Benewah	Bingham	Blaine	Boise	Bonner	…	Shoshone	Teton	\
0	0	0	0	0	0	…	0	0	
1	0	0	0	0	0	…	0	0	
2	0	0	0	0	0	…	0	0	
3	0	0	0	0	0	…	0	0	
4	0	0	0	0	0	…	0	0	
…	…	…	…	…	…	…	…	…	
1138	0	0	0	0	0	…	0	0	
1139	0	0	0	0	0	…	0	0	
1140	40	40	40	40	40	…	40	40	
1141	0	0	0	0	0	…	0	0	
1142	0	0	0	0	0	…	0	0	

	Twin Falls	Unassigned	Valley	Washington	sub_key	sub_value	county	\
0	0	0	0	0	new	0	Ada	
1	0	0	0	0	new	0	Ada	
2	0	0	0	0	new	0	Ada	
3	0	0	0	0	new	0	Ada	
4	0	0	0	0	new	0	Ada	
…	…	…	…	…	…	…	…	
1138	0	0	0	0	new	0	Ada	
1139	0	0	0	0	new	0	Ada	
1140	40	40	40	40	new	40	Ada	
1141	0	0	0	0	new	0	Ada	
1142	0	0	0	0	new	0	Ada	

	new_case_sums
0	0
1	0
2	0

```
3                  0
4                  0
…                  …
1138               0
1139               0
1140            1840
1141               0
1142               0

[1143 rows x 51 columns]
```

[257]:
```
# repeat step 1 and 2
```

[258]:
```
region3 = 'South Dakota'
```

[259]:
```
url3 = 'https://api.api-ninjas.com/v1/covid19?country={}&region={}'.
 ↪format(country, region3)
```

[260]:
```
response3 = requests.get(url3, headers=headers)
```

[261]:
```
if response3.status_code == 200:
    data3 = response3.json()
    # print(type(data))
    # print(len(data))
    df3 = pd.DataFrame(data3)
else:
    print("Failed to fetch data. Status code:", response3.status_code)
```

[262]:
```
cases_df3 = pd.DataFrame(cases.items(), columns=["key", "value"])
```

[263]:
```
cases_df3[['sub_key', 'sub_value']] = cases_df3['value'].apply(lambda x: pd.
 ↪Series(list(x.items())[1]))
cases_df3['county']=df3.iloc[0]["county"]
```

[264]:
```
for i in range(1,len(data3)):
    cases3=df3.iloc[i]["cases"]
    cases_df3_tmp = pd.DataFrame(cases3.items(), columns=["key", "value"])
    cases_df3_tmp[['sub_key', 'sub_value']] = cases_df3_tmp['value'].
 ↪apply(lambda x: pd.Series(list(x.items())[1]))
    cases_df3[df3.iloc[i]["county"]]=cases_df3_tmp['sub_value']
```

[265]:
```
cases_df3['new_case_sums'] = cases_df3.sum(axis=1)
```

```
/var/folders/6q/k8jdwbv174s78xj3x3kzvn0w0000gn/T/ipykernel_37936/1397222048.py:1
: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with
'numeric_only=None') is deprecated; in a future version this will raise
TypeError.  Select only valid columns before calling the reduction.
  cases_df3['new_case_sums'] = cases_df3.sum(axis=1)
```

[266]:
```
cases_df3
```

[266]:

	key	value	sub_key	sub_value	county	\
0	2020-01-22	{'total': 0, 'new': 0}	new	0	Aurora	
1	2020-01-23	{'total': 0, 'new': 0}	new	0	Aurora	
2	2020-01-24	{'total': 0, 'new': 0}	new	0	Aurora	
3	2020-01-25	{'total': 0, 'new': 0}	new	0	Aurora	
4	2020-01-26	{'total': 0, 'new': 0}	new	0	Aurora	
…	…	…	…	…	…	
1138	2023-03-05	{'total': 21884, 'new': 0}	new	0	Aurora	
1139	2023-03-06	{'total': 21884, 'new': 0}	new	0	Aurora	
1140	2023-03-07	{'total': 21924, 'new': 40}	new	40	Aurora	
1141	2023-03-08	{'total': 21924, 'new': 0}	new	0	Aurora	
1142	2023-03-09	{'total': 21924, 'new': 0}	new	0	Aurora	

	Beadle	Bennett	Bon Homme	Brookings	Brown	…	Sully	Todd	Tripp	\
0	0	0	0	0	0	…	0	0	0	
1	0	0	0	0	0	…	0	0	0	
2	0	0	0	0	0	…	0	0	0	
3	0	0	0	0	0	…	0	0	0	
4	0	0	0	0	0	…	0	0	0	
…	…	…	…	…	…	…	…	…	…	
1138	0	0	0	0	0	…	0	0	0	
1139	0	0	0	0	0	…	0	0	0	
1140	0	0	0	0	0	…	0	0	0	
1141	9	2	6	22	43	…	1	7	3	
1142	0	0	0	0	0	…	0	0	0	

	Turner	Unassigned	Union	Walworth	Yankton	Ziebach	new_case_sums
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0
…	…	…	…	…	…	…	
1138	0	0	0	0	0	0	0
1139	0	0	0	0	0	0	0
1140	0	0	0	0	0	0	40
1141	7	0	4	0	21	0	639
1142	0	0	0	0	0	0	0

[1143 rows x 73 columns]

[267]:
```
# repeat step 1 and 2
```

[268]:
```
region4 = 'Minnesota'
```

[269]:
```
url4 = 'https://api.api-ninjas.com/v1/covid19?country={}&region={}'.
 ↪format(country, region4)
```

[270]:
```
response4 = requests.get(url4, headers=headers)
```

[271]:
```
if response4.status_code == 200:
    data4 = response4.json()
    # print(type(data))
    # print(len(data))
    df4 = pd.DataFrame(data4)
else:
    print("Failed to fetch data. Status code:", response4.status_code)
```

[272]:
```
cases_df4 = pd.DataFrame(cases.items(), columns=["key", "value"])
```

[273]:
```
cases_df4[['sub_key', 'sub_value']] = cases_df4['value'].apply(lambda x: pd.
 ↪Series(list(x.items())[1]))
cases_df4['county']=df4.iloc[0]["county"]
```

[274]:
```
for i in range(1,len(data4)):
    cases4=df4.iloc[i]["cases"]
    cases_df4_tmp = pd.DataFrame(cases4.items(), columns=["key", "value"])
    cases_df4_tmp[['sub_key', 'sub_value']] = cases_df4_tmp['value'].
 ↪apply(lambda x: pd.Series(list(x.items())[1]))
    cases_df4[df4.iloc[i]["county"]]=cases_df4_tmp['sub_value']
```

[275]:
```
cases_df4['new_case_sums'] = cases_df4.sum(axis=1)
```

```
/var/folders/6q/k8jdwbv174s78xj3x3kzvn0w0000gn/T/ipykernel_37936/2938808042.py:1
: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with
'numeric_only=None') is deprecated; in a future version this will raise
TypeError.  Select only valid columns before calling the reduction.
  cases_df4['new_case_sums'] = cases_df4.sum(axis=1)
```

[276]:
```
cases_df4
```

[276]:

	key	value	sub_key	sub_value	county	\
0	2020-01-22	{'total': 0, 'new': 0}	new	0	Aitkin	
1	2020-01-23	{'total': 0, 'new': 0}	new	0	Aitkin	
2	2020-01-24	{'total': 0, 'new': 0}	new	0	Aitkin	
3	2020-01-25	{'total': 0, 'new': 0}	new	0	Aitkin	
4	2020-01-26	{'total': 0, 'new': 0}	new	0	Aitkin	
…	…	…	…	…	…	
1138	2023-03-05	{'total': 21884, 'new': 0}	new	0	Aitkin	
1139	2023-03-06	{'total': 21884, 'new': 0}	new	0	Aitkin	
1140	2023-03-07	{'total': 21924, 'new': 40}	new	40	Aitkin	
1141	2023-03-08	{'total': 21924, 'new': 0}	new	0	Aitkin	
1142	2023-03-09	{'total': 21924, 'new': 0}	new	0	Aitkin	

```
      Anoka  Becker  Beltrami  Benton  Big Stone  …  Wabasha  Wadena  \
0         0       0         0       0          0  …        0       0
1         0       0         0       0          0  …        0       0
2         0       0         0       0          0  …        0       0
3         0       0         0       0          0  …        0       0
4         0       0         0       0          0  …        0       0
…       …     …       …     …       …  …       …     …
1138      0       0         0       0          0  …        0       0
1139      0       0         0       0          0  …        0       0
1140      0       0         0       0          0  …        0       0
1141      0       0         0       0          0  …        0       0
1142    207      20        35      39          8  …        6      12

      Waseca  Washington  Watonwan  Wilkin  Winona  Wright  Yellow Medicine  \
0          0           0         0       0       0       0                0
1          0           0         0       0       0       0                0
2          0           0         0       0       0       0                0
3          0           0         0       0       0       0                0
4          0           0         0       0       0       0                0
…        …         …       …     …     …       …
1138       0           0         0       0       0       0                0
1139       0           0         0       0       0       0                0
1140       0           0         0       0       0       0                0
1141       0           0         0       0       0       0                0
1142       4         148         5       6      48      67                3

      new_case_sums
0                 0
1                 0
2                 0
3                 0
4                 0
…               …
1138              0
1139              0
1140             40
1141              0
1142           3596

[1143 rows x 94 columns]
```

[277]:
```
# repeat step 1 and 2
```

[278]:
```
region5 = 'Massachusetts'
```

[279]:
```
url5 = 'https://api.api-ninjas.com/v1/covid19?country={}&region={}'.
 ↪format(country, region5)
```

[280]:
```
response5 = requests.get(url5, headers=headers)
```

[281]:
```
if response5.status_code == 200:
    data5 = response5.json()
    # print(type(data))
    # print(len(data))
    df5 = pd.DataFrame(data5)
else:
    print("Failed to fetch data. Status code:", response5.status_code)
```

[282]:
```
cases_df5 = pd.DataFrame(cases.items(), columns=["key", "value"])
```

[283]:
```
cases_df5[['sub_key', 'sub_value']] = cases_df5['value'].apply(lambda x: pd.
 ↪Series(list(x.items())[1]))
cases_df5['county']=df5.iloc[0]["county"]
```

[284]:
```
for i in range(1,len(data5)):
    cases5=df5.iloc[i]["cases"]
    cases_df5_tmp = pd.DataFrame(cases.items(), columns=["key", "value"])
    cases_df5_tmp[['sub_key', 'sub_value']] = cases_df5_tmp['value'].
 ↪apply(lambda x: pd.Series(list(x.items())[1]))
    cases_df5[df5.iloc[i]["county"]]=cases_df5_tmp['sub_value']
```

[285]:
```
cases_df5['new_case_sums'] = cases_df5.sum(axis=1)
```

```
/var/folders/6q/k8jdwbv174s78xj3x3kzvn0w0000gn/T/ipykernel_37936/3591150679.py:1
: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with
'numeric_only=None') is deprecated; in a future version this will raise
TypeError.  Select only valid columns before calling the reduction.
  cases_df5['new_case_sums'] = cases_df5.sum(axis=1)
```

[286]:
```
cases_df5
```

[286]:

	key	value	sub_key	sub_value	county	\
0	2020-01-22	{'total': 0, 'new': 0}	new	0	Barnstable	
1	2020-01-23	{'total': 0, 'new': 0}	new	0	Barnstable	
2	2020-01-24	{'total': 0, 'new': 0}	new	0	Barnstable	
3	2020-01-25	{'total': 0, 'new': 0}	new	0	Barnstable	
4	2020-01-26	{'total': 0, 'new': 0}	new	0	Barnstable	
…	…	…	…	…	…	
1138	2023-03-05	{'total': 21884, 'new': 0}	new	0	Barnstable	
1139	2023-03-06	{'total': 21884, 'new': 0}	new	0	Barnstable	
1140	2023-03-07	{'total': 21924, 'new': 40}	new	40	Barnstable	
1141	2023-03-08	{'total': 21924, 'new': 0}	new	0	Barnstable	
1142	2023-03-09	{'total': 21924, 'new': 0}	new	0	Barnstable	

```
      Berkshire  Bristol  Dukes  Dukes and Nantucket  Essex  …  Hampshire  \
0             0        0      0                    0      0  …          0
1             0        0      0                    0      0  …          0
2             0        0      0                    0      0  …          0
3             0        0      0                    0      0  …          0
4             0        0      0                    0      0  …          0
…           …      …    …                    …     …  …        …
1138          0        0      0                    0      0  …          0
1139          0        0      0                    0      0  …          0
1140         40       40     40                   40     40  …         40
1141          0        0      0                    0      0  …          0
1142          0        0      0                    0      0  …          0

      Middlesex  Nantucket  Norfolk  Out of MA  Plymouth  Suffolk  Unassigned  \
0             0          0        0          0         0        0           0
1             0          0        0          0         0        0           0
2             0          0        0          0         0        0           0
3             0          0        0          0         0        0           0
4             0          0        0          0         0        0           0
…           …        …       …          …       …      …          …
1138          0          0        0          0         0        0           0
1139          0          0        0          0         0        0           0
1140         40         40       40         40        40       40          40
1141          0          0        0          0         0        0           0
1142          0          0        0          0         0        0           0

      Worcester  new_case_sums
0             0              0
1             0              0
2             0              0
3             0              0
4             0              0
…           …            …
1138          0              0
1139          0              0
1140         40            680
1141          0              0
1142          0              0

[1143 rows x 22 columns]
```

[287]:
```
# step 3 add a new column and drop some columns in each dataframe
```

[288]:
```
cases_df_selected = cases_df[['key', 'new_case_sums']]
cases_df_selected.head()
```

[288]:
```
           key  new_case_sums
0   2020-01-22              0
1   2020-01-23              0
2   2020-01-24              0
3   2020-01-25              0
4   2020-01-26              2
```

[289]:
```
cases_df_selected['State'] = 'CA'
```

```
/var/folders/6q/k8jdwbv174s78xj3x3kzvn0w0000gn/T/ipykernel_37936/1985411014.py:1
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  cases_df_selected['State'] = 'CA'
```

[290]:
```
cases_df_selected
```

[290]:
```
              key  new_case_sums State
0      2020-01-22              0    CA
1      2020-01-23              0    CA
2      2020-01-24              0    CA
3      2020-01-25              0    CA
4      2020-01-26              2    CA
…           …             …    …
1138   2023-03-05              0    CA
1139   2023-03-06              0    CA
1140   2023-03-07          10786    CA
1141   2023-03-08              0    CA
1142   2023-03-09           8734    CA

[1143 rows x 3 columns]
```

[291]:
```
cases_df_selected1 = cases_df1[['key', 'new_case_sums']]
cases_df_selected1.head()
```

[291]:
```
           key  new_case_sums
0   2020-01-22              0
1   2020-01-23              0
2   2020-01-24              0
3   2020-01-25              0
4   2020-01-26              0
```

[292]:
```
cases_df_selected1['State'] = 'AZ'
```

```
/var/folders/6q/k8jdwbv174s78xj3x3kzvn0w0000gn/T/ipykernel_37936/2785569492.py:1
: SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  cases_df_selected1['State'] = 'AZ'
```

[293]:
```
cases_df_selected1
```

[293]:

	key	new_case_sums	State
0	2020-01-22	0	AZ
1	2020-01-23	0	AZ
2	2020-01-24	0	AZ
3	2020-01-25	0	AZ
4	2020-01-26	0	AZ
…	…	…	…
1138	2023-03-05	0	AZ
1139	2023-03-06	0	AZ
1140	2023-03-07	680	AZ
1141	2023-03-08	0	AZ
1142	2023-03-09	0	AZ

[1143 rows x 3 columns]

[294]:
```
cases_df_selected2 = cases_df2[['key', 'new_case_sums']]
cases_df_selected2.head()
```

[294]:

	key	new_case_sums
0	2020-01-22	0
1	2020-01-23	0
2	2020-01-24	0
3	2020-01-25	0
4	2020-01-26	0

[295]:
```
cases_df_selected2['State'] = 'ID'
```

```
/var/folders/6q/k8jdwbv174s78xj3x3kzvn0w0000gn/T/ipykernel_37936/3132477610.py:1
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  cases_df_selected2['State'] = 'ID'
```

[296]:
```
cases_df_selected3 = cases_df3[['key', 'new_case_sums']]
cases_df_selected3.head()
```

[296]:

	key	new_case_sums
0	2020-01-22	0
1	2020-01-23	0
2	2020-01-24	0
3	2020-01-25	0
4	2020-01-26	0

[297]:
```
cases_df_selected3['State'] = 'SD'
```

```
/var/folders/6q/k8jdwbv174s78xj3x3kzvn0w0000gn/T/ipykernel_37936/572683305.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  cases_df_selected3['State'] = 'SD'
```

[298]:
```
cases_df_selected4 = cases_df4[['key', 'new_case_sums']]
cases_df_selected4.head()
```

[298]:

	key	new_case_sums
0	2020-01-22	0
1	2020-01-23	0
2	2020-01-24	0
3	2020-01-25	0
4	2020-01-26	0

[299]:
```
cases_df_selected4['State'] = 'MN'
```

```
/var/folders/6q/k8jdwbv174s78xj3x3kzvn0w0000gn/T/ipykernel_37936/2899550350.py:1
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  cases_df_selected4['State'] = 'MN'
```

[300]:
```
cases_df_selected5 = cases_df5[['key', 'new_case_sums']]
cases_df_selected5.head()
```

[300]:

	key	new_case_sums
0	2020-01-22	0
1	2020-01-23	0
2	2020-01-24	0
3	2020-01-25	0
4	2020-01-26	0

[301]:
```
cases_df_selected5['State'] = 'MA'
```

```
/var/folders/6q/k8jdwbv174s78xj3x3kzvn0w0000gn/T/ipykernel_37936/3005274306.py:1
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  cases_df_selected5['State'] = 'MA'
```

[302]:
```
# step 4 replace header
```

[303]:
```
cases_df_selected.rename(columns={'key': 'Date'}, inplace=True)
```

```
/Users/yuhang/opt/anaconda3/lib/python3.9/site-
packages/pandas/core/frame.py:5039: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return super().rename(
```

[304]:
```
cases_df_selected
```

[304]:

	Date	new_case_sums	State
0	2020-01-22	0	CA
1	2020-01-23	0	CA
2	2020-01-24	0	CA
3	2020-01-25	0	CA
4	2020-01-26	2	CA
…	…	…	…
1138	2023-03-05	0	CA
1139	2023-03-06	0	CA
1140	2023-03-07	10786	CA
1141	2023-03-08	0	CA
1142	2023-03-09	8734	CA

[1143 rows x 3 columns]

[305]:
```
cases_df_selected1.rename(columns={'key': 'Date'}, inplace=True)
```

```
/Users/yuhang/opt/anaconda3/lib/python3.9/site-
packages/pandas/core/frame.py:5039: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return super().rename(
```

[306]:
```
cases_df_selected2.rename(columns={'key': 'Date'}, inplace=True)
```

[307]:
```
cases_df_selected3.rename(columns={'key': 'Date'}, inplace=True)
```

[308]:
```
cases_df_selected4.rename(columns={'key': 'Date'}, inplace=True)
```

[309]:
```
cases_df_selected5.rename(columns={'key': 'Date'}, inplace=True)
```

[310]:
```
cases_df_selected5
```

[310]:

	Date	new_case_sums	State
0	2020-01-22	0	MA
1	2020-01-23	0	MA
2	2020-01-24	0	MA
3	2020-01-25	0	MA
4	2020-01-26	0	MA
…	…	…	…
1138	2023-03-05	0	MA
1139	2023-03-06	0	MA
1140	2023-03-07	680	MA
1141	2023-03-08	0	MA
1142	2023-03-09	0	MA

[1143 rows x 3 columns]

[311]:
```
# step 5 stack datasets
```

[312]:
```
stacked_df = pd.concat([cases_df_selected, cases_df_selected1,␣
↪cases_df_selected2, cases_df_selected3, cases_df_selected4,␣
↪cases_df_selected5], ignore_index=True)
```

[313]:
```
stacked_df
```

[313]:

	Date	new_case_sums	State
0	2020-01-22	0	CA
1	2020-01-23	0	CA
2	2020-01-24	0	CA
3	2020-01-25	0	CA
4	2020-01-26	2	CA
…	…	…	…
6853	2023-03-05	0	MA
6854	2023-03-06	0	MA
6855	2023-03-07	680	MA
6856	2023-03-08	0	MA
6857	2023-03-09	0	MA

[6858 rows x 3 columns]

This week I've been working on pulling data from API, the API I'm working on is a Covid-19 API that provides current and historical Covid-19 data for every country in the world. Available data from this API includes confirmed case counts and deaths, and is updated everyday. My goal is to pull data froom this API and see some correlation between covid 19 vaccination and new cases in states. Since the data are very massive, with data in every country, states and on every day. I tried to pull data in every states, however it took very long to run the code, so I've decided to only select some states to make the analysis. The states I've selected are California, Arizona, Idaho, South Dakota, Minnesota and Massachusetts, which I think would be characteristic states across the timezones in the US.

Firstly, I called the API and get the data to form them into a dataset, but only get the new cases. The API only gave information of new cases in each county within every states. Secondly, I added together the new cases in each county and add a new column to the datasets which contributes the total new cases in each states on each day. Repeat this transformation until I got the datasets for all of the 6 states I've chose. Thirdly, I add a new column for each datasets with the state name abbreviation for future data concatenate with my csv file and drop the rest the columns. Then I replaced header in each datasets for the 'key' column, replace them into 'Date', which represents the date. Last but not least I performed a stack of all the datasets for future data concatenate.

I haven't forseen any risk or prediction of my project during the calling of my API as well as the the tranformation of the datasets. The datasets are from the government and is a creditable source that could trust. The only concern is the accuracy of the data, there might be missing records of the confirmed cases, or not reported. However the group of unreported cases are slightly small that the results won't be affected much.

[314]:
```
file_path = "cases_df_selected_CA.csv"
cases_df_selected.to_csv(file_path, index=False)
```

[315]:
```
file_path1 = "cases_df_selected_AZ.csv"
cases_df_selected1.to_csv(file_path1, index=False)
```

[316]:
```
file_path2 = "cases_df_selected_ID.csv"
cases_df_selected2.to_csv(file_path2, index=False)
```

[317]:
```
file_path3 = "cases_df_selected_SD.csv"
cases_df_selected3.to_csv(file_path3, index=False)
```

[318]:
```
file_path4 = "cases_df_selected_MN.csv"
cases_df_selected4.to_csv(file_path4, index=False)
```

[319]:
```
file_path5 = "cases_df_selected_MA.csv"
cases_df_selected5.to_csv(file_path5, index=False)
```

[320]:
```
file_path6 = "stacked_df.csv"
stacked_df.to_csv(file_path6, index=False)
```

[ ]:
```

```