```
In [22]: import pandas as pd
```

```
In [23]: import numpy as np
```

```
In [24]: import seaborn as sns
```

```
In [25]: from matplotlib import pyplot as plt
```

```
In [26]: riverdata = pd.read_csv('river_data.csv', encoding='ISO-8859-1')
```

In [27]:
```
riverdata
```

Out[27]:

	LOCATIONS	STATE	TEMP	DO	pH	CONDUCTIVITY	BOD	NITRATE_N_NITRITE_I
0	AMARAVATI , GUNTUR DIST., A.P	AP	27.6	7.0	7.8	669.0	0.6	0.4
1	GODAVARI AT BASARA, ADILABAD	AP	28.0	5.5	8.1	826.0	1.7	1.0
2	GODAVARI AT BHADRACHALAM D/S BATHING GHAT, KHA...	AP	20.2	5.6	8.0	462.0	0.8	1.0
3	GODAVARI AT BHADRACHALAM U/S BATHING GHAT, KHA...	AP	20.0	6.0	8.1	443.0	0.3	1.0
4	GODAVARI AT BURGAMPAHAD, KHAMMAM	AP	19.8	6.1	7.9	666.0	1.8	0.8
...	...	...	...	...	...	...	...	.
529	KALJANI D/S OF ALIPURDWAR, MUNICIPALITY DISCHA...	WB	28.8	7.6	7.6	141.0	2.4	0.3
530	KAROLA, D/S OF JALPAIGURI, NEAR MIN BHAWAN	WB	27.3	6.7	7.4	166.0	3.0	0.3
531	NABADIP ON GANGA,GHOSHPARA NEAR MONIPURGHAT	WB	27.4	7.4	7.9	353.0	3.3	0.7
532	TEESTA AT SILIGURI	WB	20.8	7.5	7.4	86.0	2.0	0.5
533	TRIBENI ON GANGA, NEAR BURNING GHAT	WB	28.3	6.9	8.0	312.0	2.8	0.8

534 rows × 10 columns

In [10]:

```
riverdata.dtypes
```

Out[10]:

```
LOCATIONS                object
STATE                    object
TEMP                    float64
DO                      float64
pH                      float64
CONDUCTIVITY            float64
BOD                     float64
NITRATE_N_NITRITE_N     float64
FECAL_COLIFORM          float64
TOTAL_COLIFORM          float64
dtype: object
```

In [11]:

```
# Replacing string NAN values with actual NAN value (np.nan)
def convert_to_nan(riverdata):
    n_col = riverdata.shape[1]
    for index in range(n_col):
        riverdata.iloc[:, index]  = riverdata.iloc[:, index].replace("NAN", n
    return riverdata

riverdata = convert_to_nan(riverdata)
```

In [12]:

```
riverdata.columns
```

Out[12]:

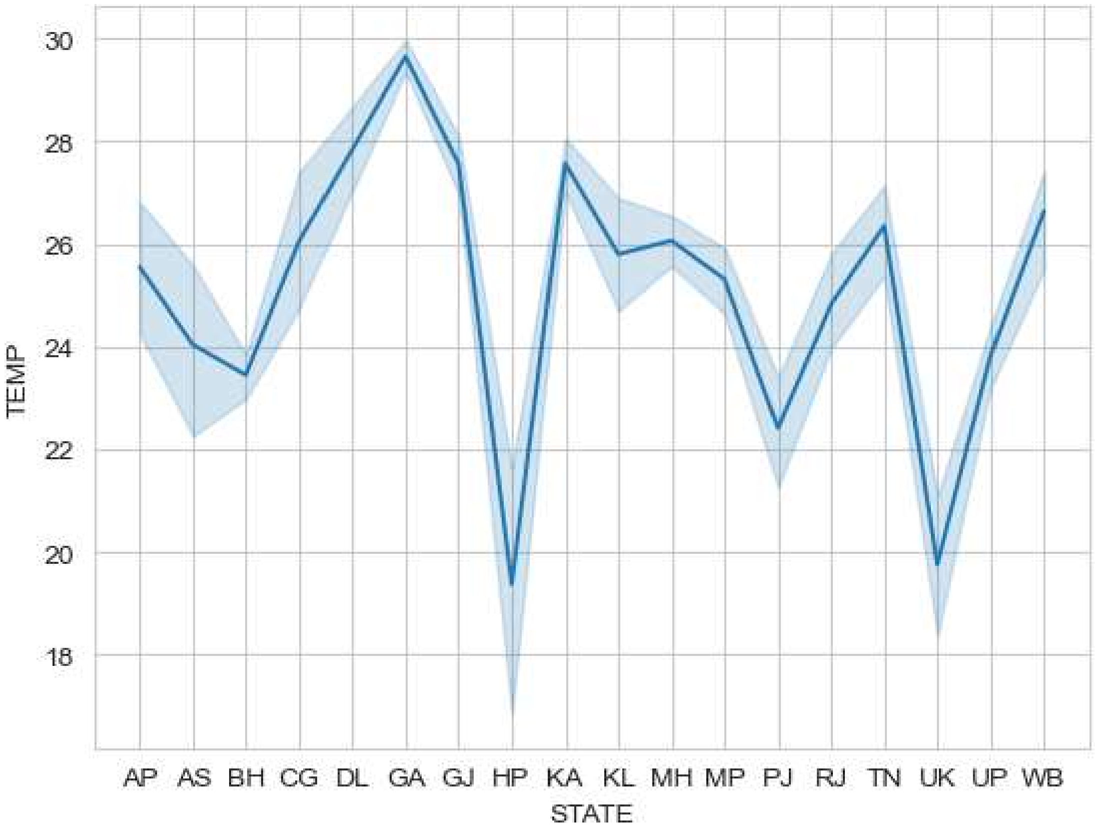
```
Index(['LOCATIONS', 'STATE', 'TEMP', 'DO', 'pH', 'CONDUCTIVITY', 'BOD',
       'NITRATE_N_NITRITE_N', 'FECAL_COLIFORM', 'TOTAL_COLIFORM'],
      dtype='object')
```

In [ ]:

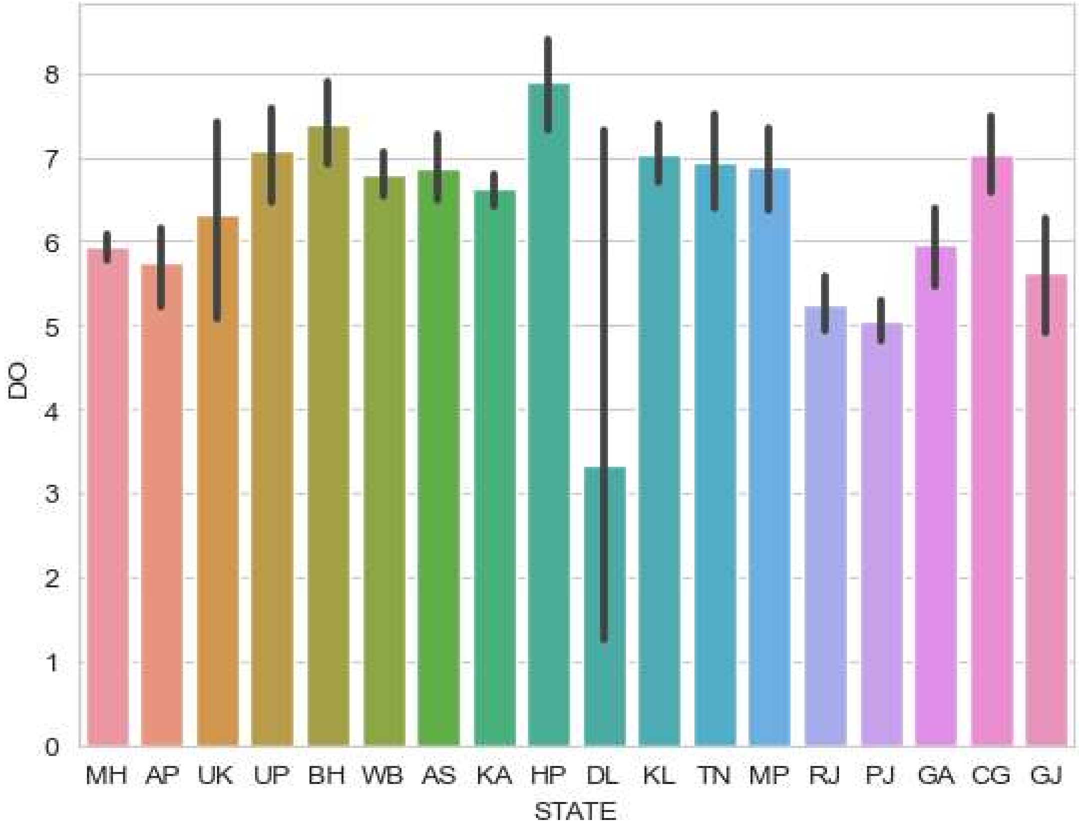
sns.scatterplot(x='STATE', y='LOCATIONS', data=riverdata)

In [57]:
```
sns.lineplot(x='STATE', y='TEMP', data=riverdata)
```

Out[57]:
```
<Axes: xlabel='STATE', ylabel='TEMP'>
```

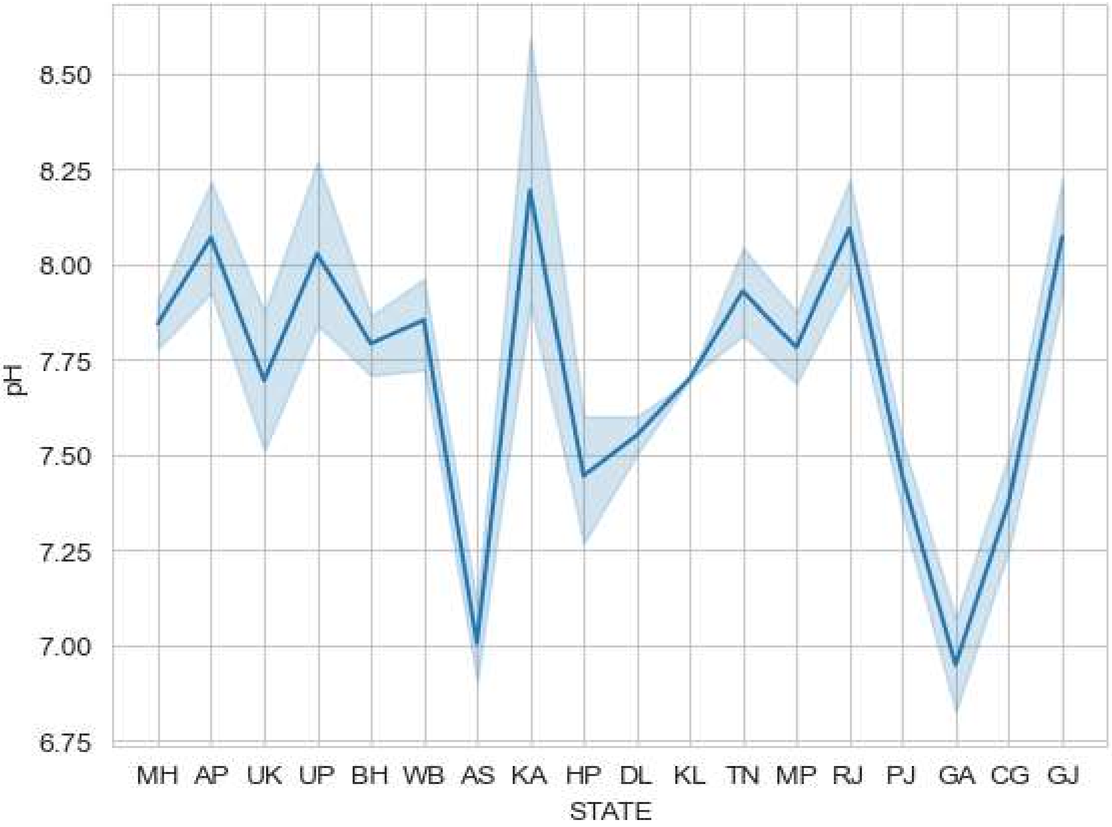


```
In [43]: sns.barplot(x='STATE', y='DO', data=riverdata)
         plt.show()
```

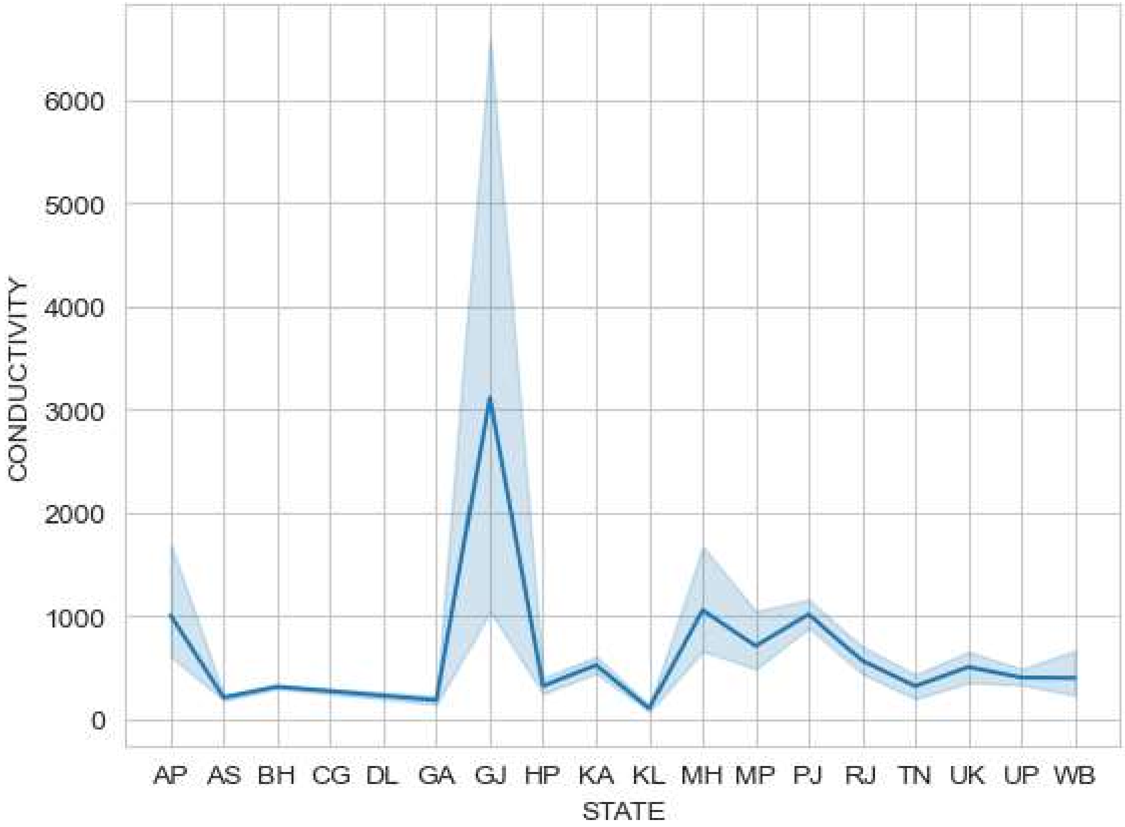


In [53]:

```
sns.lineplot(x='STATE', y='pH', data=riverdata)
plt.show()
```

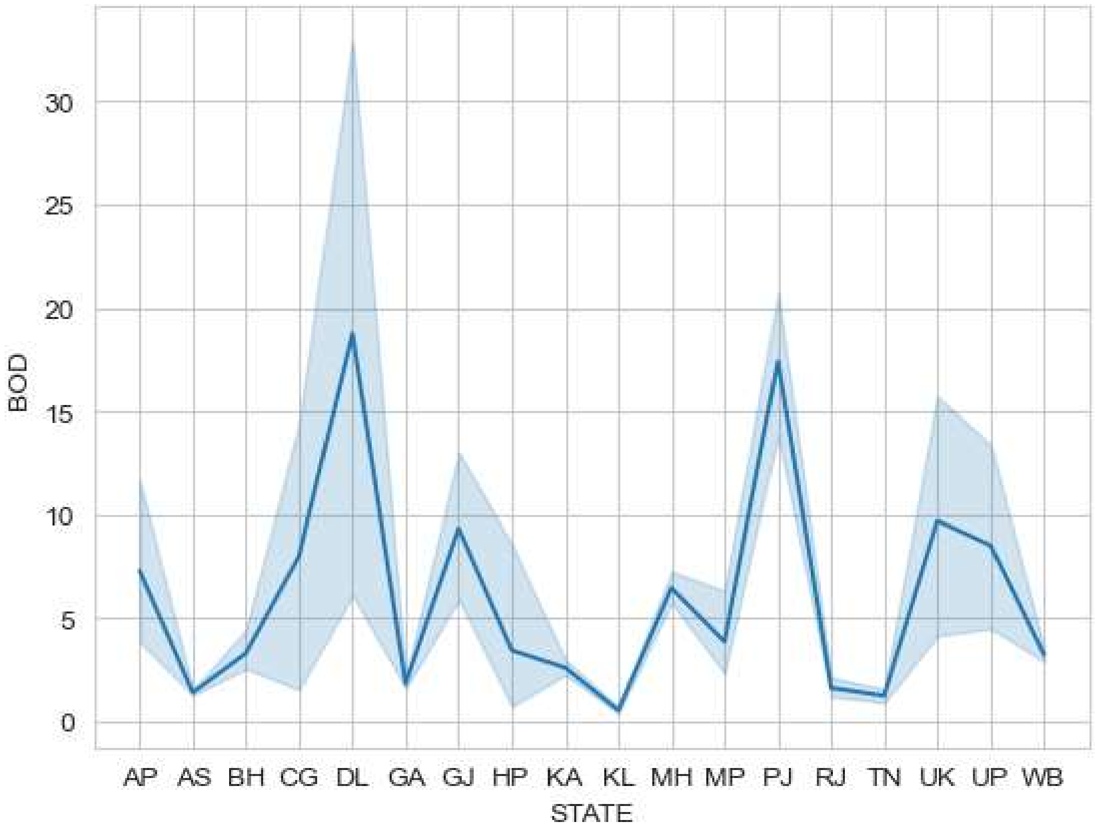


```
In [58]: sns.lineplot(x='STATE', y='CONDUCTIVITY', data=riverdata)
         plt.show()
```

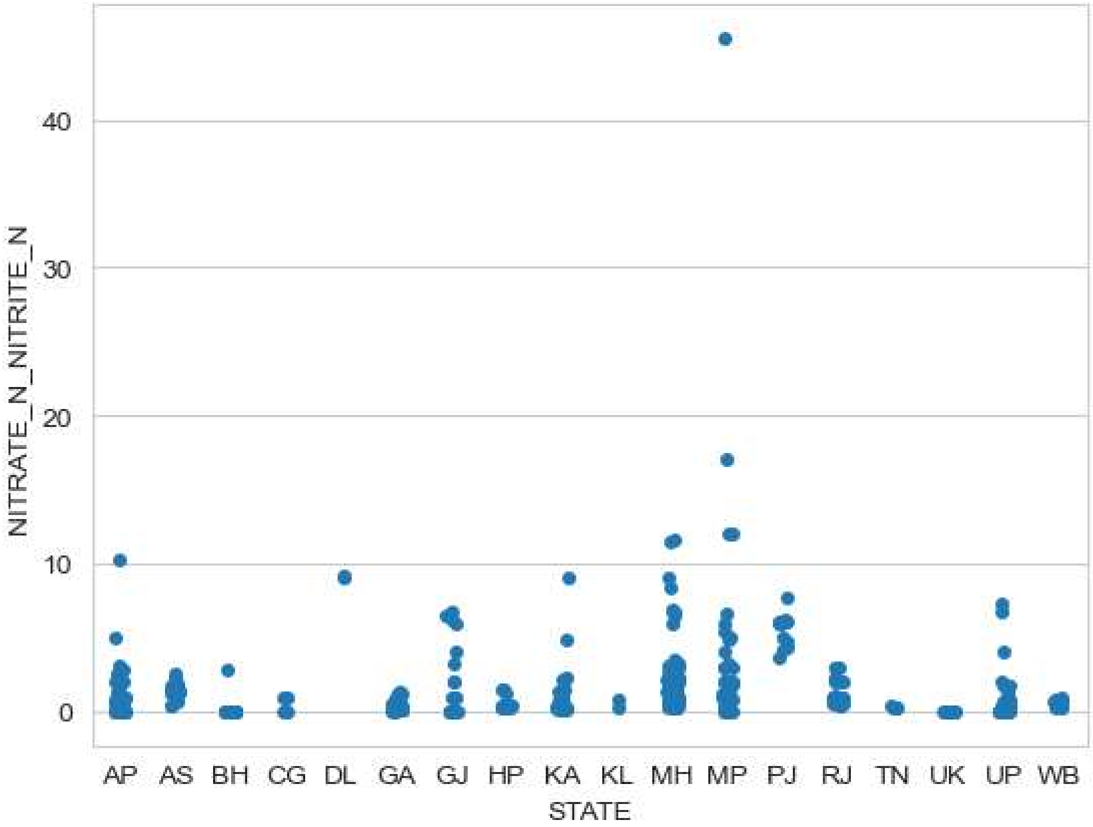


In [59]:

```
sns.lineplot(x='STATE', y='BOD', data=riverdata)
plt.show()
```

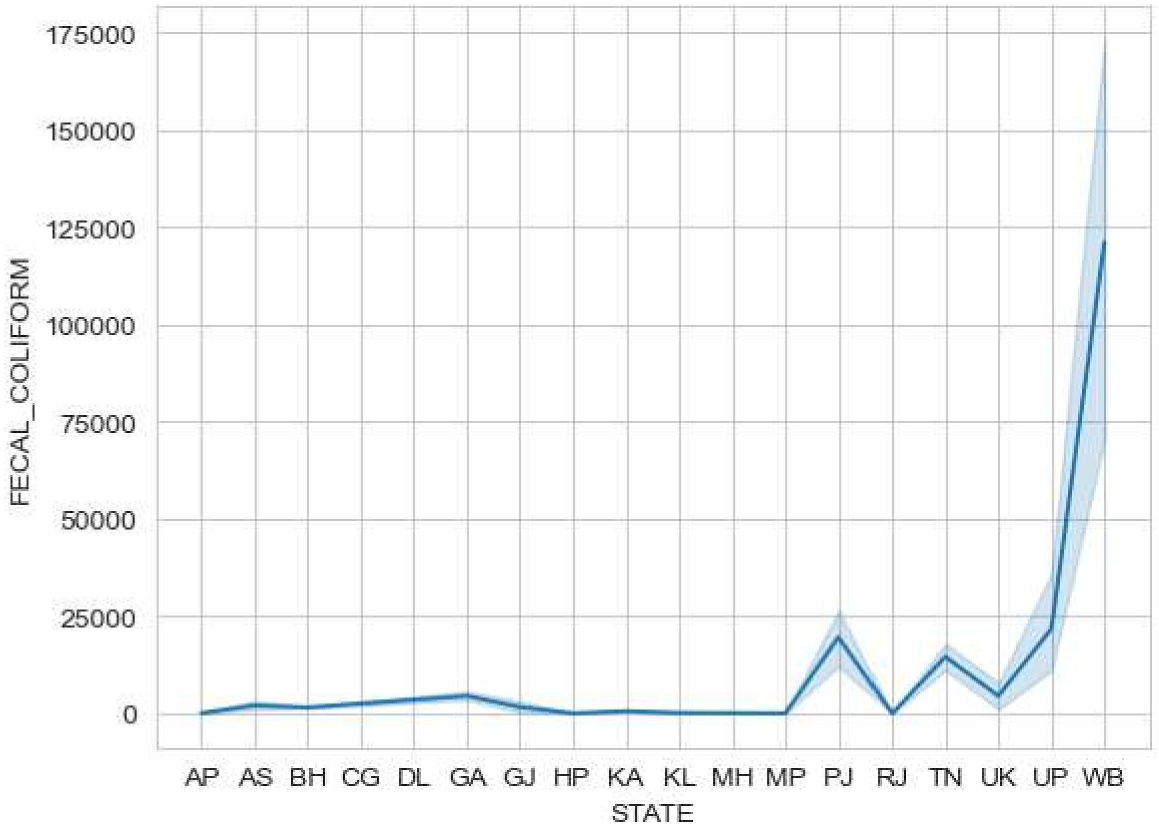


In [71]:
```
sns.stripplot(x='STATE', y='NITRATE_N_NITRITE_N', data=riverdata)
plt.show()
```

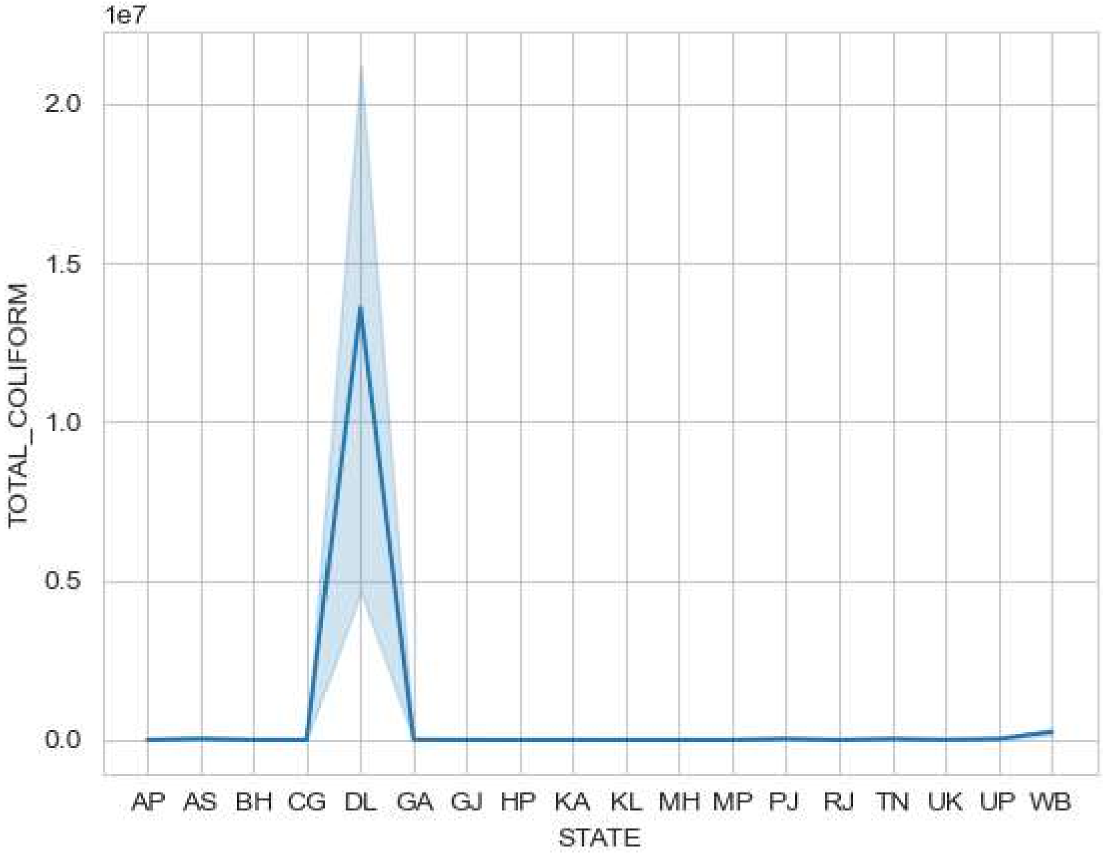


In [61]:

```
sns.lineplot(x='STATE', y='FECAL_COLIFORM', data=riverdata)
plt.show()
```



In [72]:
```
sns.lineplot(x='STATE', y='TOTAL_COLIFORM', data=riverdata)
plt.show()
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: