In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
superdata = pd.read_csv('covid19_superdata.csv')
```

In [3]:
```
df = pd.concat([superdata.iloc[:, 0:4],superdata.iloc[:,865:1079],
                superdata.iloc[:,1976:2190],
               superdata.iloc[:, -1]], axis=1)
```

In [4]:
```
df.head()
```

Out[4]:

	countyFIPS	County Name	State	StateFIPS	2022-06-01_x	2022-06-02_x	2022-06-03_x	2022-06-04_x	2022-06-05_x	2022-06-06_x	...	2022-12-23_y	2022-12-24_y	2022-12-25_y	2022-12-26_y	2022-12-27_y	2022-12-28_y	2022-12-29_y	2022-12-30_y	2022-12-31_y
0	0	statewide unallocated	AL	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
1	1001	autauga county	AL	1	15969	15978	15978	15978	15978	16032	...	230	230	230	230	230	230	230	230	230
2	1003	baldwin county	AL	1	56580	56648	56648	56648	56648	56895	...	719	719	719	719	719	719	719	719	719
3	1005	barbour county	AL	1	5710	5714	5714	5714	5714	5719	...	103	103	103	103	103	103	103	103	103
4	1007	bibb county	AL	1	6508	6512	6512	6512	6512	6534	...	108	108	108	108	108	108	108	108	108

5 rows × 433 columns

In [5]:
```
def county_state(x):
    return x[0] +"_"+ x[1]
```

In [6]:
```
df["county_state"] = df[["County Name", "State"]].apply(county_state, axis=1)
```

In [7]:
```
df.index=df["county_state"]
```

In [8]:
```
df.head()
```

Out[8]:

	countyFIPS	County Name	State	StateFIPS	2022-06-01_x	2022-06-02_x	2022-06-03_x	2022-06-04_x	2022-06-05_x	2022-06-06_x	...	2022-12-24_y	2022-12-25_y	2022-12-26_y	2022-12-27_y	2022-12-28_y	2022-12-29_y	2022-12-30_y
county_state																		
statewide unallocated_AL	0	statewide unallocated	AL	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0
autauga county_AL	1001	autauga county	AL	1	15969	15978	15978	15978	15978	16032	...	230	230	230	230	230	230	230
baldwin county_AL	1003	baldwin county	AL	1	56580	56648	56648	56648	56648	56895	...	719	719	719	719	719	719	719
barbour county_AL	1005	barbour county	AL	1	5710	5714	5714	5714	5714	5719	...	103	103	103	103	103	103	103
bibb county_AL	1007	bibb county	AL	1	6508	6512	6512	6512	6512	6534	...	108	108	108	108	108	108	108

5 rows × 434 columns

In [9]:
```
superdataT = df.T.copy()
```

In [10]:
```
superdataT["Date"] = superdataT.index
```

In [11]:
```
to_remove = list(superdataT.index[:4])
to_remove.append(superdataT.index[-2])
to_remove.append(superdataT.index[-1])
to_remove
```

Out[11]:
```
['countyFIPS',
 'County Name',
 'State',
 'StateFIPS',
 'population',
 'county_state']
```

In [12]:
```
def new_death(x):
    if x[-2:]=="_x":
        return "new"
    elif x[-2:]=="_y":
        return "death"
```

In [13]:
```
superdataT["new_death"]=superdataT["Date"].apply(new_death)
```

In [14]:
```
superdataT["new_death"].value_counts().index
```

Out[14]:
```
Index(['new', 'death'], dtype='object')
```

In [15]:
```
def clean_date(x,l=to_remove):
        if x in l:
            return np.nan
        else:
            return x[:-2]
```

In [16]:
```
superdataT["Date"] = superdataT["Date"].apply(clean_date,l=to_remove)
```

In [17]:
```
superdataT.head()
```

Out[17]:

county_state	statewide unallocated_AL	autauga county_AL	baldwin county_AL	barbour county_AL	bibb county_AL	blount county_AL	bullock county_AL	butler county_AL	calhoun county_AL	chambers county_AL	...	platte county_WY
countyFIPS	0	1001	1003	1005	1007	1009	1011	1013	1015	1017	...	56031
County Name	statewide unallocated	autauga county	baldwin county	barbour county	bibb county	blount county	bullock county	butler county	calhoun county	chambers county	...	platte county
State	AL	AL	AL	AL	AL	AL	AL	AL	AL	AL	...	WY
StateFIPS	1	1	1	1	1	1	1	1	1	1	...	56
2022-06-01_x	0	15969	56580	5710	6508	15077	2337	5091	32596	8551	...	1929

5 rows × 3191 columns

In [18]:
```
superdataT["Week"]=pd.DatetimeIndex(superdataT["Date"]).week
```

```
C:\Users\amyme\AppData\Local\Temp\ipykernel_14660\801288638.py:1: FutureWarning: weekofyear and week have been deprecated, p
lease use DatetimeIndex.isocalendar().week instead, which returns a Series. To exactly reproduce the behavior of week and we
ekofyear and return an Index, you may call pd.Int64Index(idx.isocalendar().week)
  superdataT["Week"]=pd.DatetimeIndex(superdataT["Date"]).week
```

In [19]:
```
superdataT["Date"]=pd.DatetimeIndex(superdataT["Date"])
```

In [20]:
```
superdataT=superdataT[superdataT["Date"].notnull()]
```

In [21]:
```
superdataT.index = superdataT["Date"]
```

In [22]:
```
superdataT["Total"]=superdataT.iloc[:,:-3].sum(axis=1)
```

In [23]:
```
for feat in superdataT.columns[:-4]:
    superdataT[feat]=superdataT[feat].astype("int")
```

In [24]:
```
superdataT.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 428 entries, 2022-06-01 to 2022-12-31
Columns: 3193 entries, statewide unallocated_AL to Total
dtypes: datetime64[ns](1), float64(2), int32(3189), object(1)
memory usage: 5.2+ MB
```

In [25]:
```
superdataT["Total"]=superdataT.iloc[:,:-4].sum(axis=1)
```

In [26]:
```
superdataT.reset_index(drop=True,inplace=True)
```

In [27]:
```
superdataT["new_death"]=="death"
```

Out[27]:
```
0      False
1      False
2      False
3      False
4      False
       ...  
423     True
424     True
425     True
426     True
427     True
Name: new_death, Length: 428, dtype: bool
```

In [28]:
```
col_NC=pd.Series(superdataT.columns)[pd.Series(superdataT.columns).str.contains("_NC")][1:]
col_NC
```

Out[28]:
```
1920     alamance county_NC
1921    alexander county_NC
1922    alleghany county_NC
1923        anson county_NC
1924         ashe county_NC
                ...        
2015        wayne county_NC
2016       wilkes county_NC
2017       wilson county_NC
2018       yadkin county_NC
2019       yancey county_NC
Name: county_state, Length: 100, dtype: object
```

In [29]:
```
df_NC = superdataT[col_NC].copy()
```

In [30]:
```
df_NC.head()
```

Out[30]:

bertie ty_NC	bladen county_NC	brunswick county_NC	...	vance county_NC	wake county_NC	warren county_NC	washington county_NC	watauga county_NC	wayne county_NC	wilkes county_NC	wilson county_NC	yadkin county_NC	yancey county_NC
4353	9608	31630	...	12222	317226	4502	3091	13215	32084	17994	22266	11315	5574
4353	9608	31630	...	12222	317226	4502	3091	13215	32084	17994	22266	11315	5574
4353	9608	31630	...	12222	317226	4502	3091	13215	32084	17994	22266	11315	5574
4353	9608	31630	...	12222	317226	4502	3091	13215	32084	17994	22266	11315	5574
4353	9608	31630	...	12222	317226	4502	3091	13215	32084	17994	22266	11315	5574

In [31]:
```
df_NC["Total_NC"]=df_NC.sum(axis=1)
df_NC["new_death"]=superdataT["new_death"]
df_NC["Week"]=superdataT["Week"]
```

In [32]:
```
df_NC.head()
```

Out[32]:

county_state	alamance county_NC	alexander county_NC	alleghany county_NC	anson county_NC	ashe county_NC	avery county_NC	beaufort county_NC	bertie county_NC	bladen county_NC	brunswick county_NC	...	washington county_NC	co
0	49188	10600	3041	6672	6575	4697	12939	4353	9608	31630	...	3091	
1	49188	10600	3041	6672	6575	4697	12939	4353	9608	31630	...	3091	
2	49188	10600	3041	6672	6575	4697	12939	4353	9608	31630	...	3091	
3	49188	10600	3041	6672	6575	4697	12939	4353	9608	31630	...	3091	
4	49188	10600	3041	6672	6575	4697	12939	4353	9608	31630	...	3091	

5 rows × 103 columns

In [33]:
```
#df_NC.index[]
```

In [34]:
```
#(df["population"][pd.Series(df.columns).str.contains("_NC")])
```

Normalizing NC Data

In [35]:
```
nc_pop=superdata[superdata.StateFIPS==37].iloc[:,-1:].sum()
```

In [36]:
```
nc_pop.head()
```

Out[36]:
```
population    10488084
dtype: int64
```

In [37]:
```
df_NC.columns
```

Out[37]:
```
Index(['alamance county_NC', 'alexander county_NC', 'alleghany county_NC',
       'anson county_NC', 'ashe county_NC', 'avery county_NC',
       'beaufort county_NC', 'bertie county_NC', 'bladen county_NC',
       'brunswick county_NC',
       ...
       'washington county_NC', 'watauga county_NC', 'wayne county_NC',
       'wilkes county_NC', 'wilson county_NC', 'yadkin county_NC',
       'yancey county_NC', 'Total_NC', 'new_death', 'Week'],
      dtype='object', name='county_state', length=103)
```

In [38]:
```
df_NC.index[:4]
```

Out[38]:
```
RangeIndex(start=0, stop=4, step=1)
```

In [39]:
```
df_NC.iloc[:,2:5]
```

Out[39]:

county_state	alleghany county_NC	anson county_NC	ashe county_NC
0	3041	6672	6575
1	3041	6672	6575
2	3041	6672	6575
3	3041	6672	6575
4	3041	6672	6575
...	...	...	...
423	19	109	91
424	19	109	91
425	19	109	91
426	19	109	91
427	19	109	91

428 rows × 3 columns

In [40]:
```
df_NC.columns
```

Out[40]:
```
Index(['alamance county_NC', 'alexander county_NC', 'alleghany county_NC',
       'anson county_NC', 'ashe county_NC', 'avery county_NC',
       'beaufort county_NC', 'bertie county_NC', 'bladen county_NC',
       'brunswick county_NC',
       ...
       'washington county_NC', 'watauga county_NC', 'wayne county_NC',
       'wilkes county_NC', 'wilson county_NC', 'yadkin county_NC',
       'yancey county_NC', 'Total_NC', 'new_death', 'Week'],
      dtype='object', name='county_state', length=103)
```

In [41]:
```
df_NC.iloc[0,99:103]
```

Out[41]:
```
county_state
yancey county_NC       5574
Total_NC            2772725
new_death               new
Week                   22.0
Name: 0, dtype: object
```

In [42]:
```
watauga county_NC','wayne county_NC','wilkes county_NC','wilson county_NC','yadkin county_NC','yancey county_NC','Total_NC']]
```

In [43]:
```
X_data.head()
```

Out[43]:

bertie ınty_NC	bladen county_NC	brunswick county_NC	...	wake county_NC	warren county_NC	washington county_NC	watauga county_NC	wayne county_NC	wilkes county_NC	wilson county_NC	yadkin county_NC	yancey county_NC	Total_NC
4353	9608	31630	...	317226	4502	3091	13215	32084	17994	22266	11315	5574	2772725
4353	9608	31630	...	317226	4502	3091	13215	32084	17994	22266	11315	5574	2772725
4353	9608	31630	...	317226	4502	3091	13215	32084	17994	22266	11315	5574	2772725
4353	9608	31630	...	317226	4502	3091	13215	32084	17994	22266	11315	5574	2772725
4353	9608	31630	...	317226	4502	3091	13215	32084	17994	22266	11315	5574	2772725

In [44]:
```
#X_data.apply(lambda x: (x-x.min(axis=1))/(x.max(axis=1)-x.min(axis=1))
```

Using Sklearn to normalize [https://www.youtube.com/watch?v=Fw5iiylHzew (https://www.youtube.com/watch?v=Fw5iiylHzew)](https://www.youtube.com/watch?v=Fw5iiylHzew)

In [45]:
```
from sklearn.preprocessing import MinMaxScaler
```

In [46]:
```
scalar = MinMaxScaler()
scalar.fit(X_data)
NC_Data = scalar.transform(X_data)
```

In [47]:
```
NC_Data
```

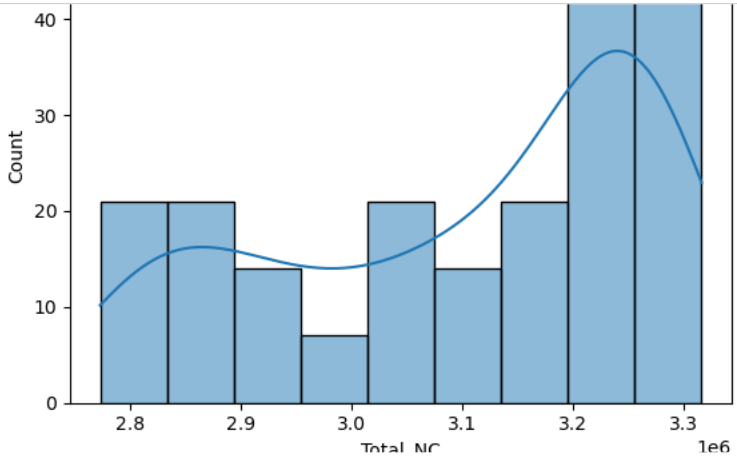
Out[47]:
```
array([[8.35033693e-01, 8.58092035e-01, 8.49241999e-01, ...,
        7.88698654e-01, 8.95762299e-01, 8.34731959e-01],
       [8.35033693e-01, 8.58092035e-01, 8.49241999e-01, ...,
        7.88698654e-01, 8.95762299e-01, 8.34731959e-01],
       [8.35033693e-01, 8.58092035e-01, 8.49241999e-01, ...,
        7.88698654e-01, 8.95762299e-01, 8.34731959e-01],
       ...,
       [1.02878894e-03, 1.31244361e-03, 8.42223470e-04, ...,
        1.40914535e-03, 1.94836824e-03, 9.03965511e-04],
       [1.02878894e-03, 1.31244361e-03, 8.42223470e-04, ...,
        1.40914535e-03, 1.94836824e-03, 9.03965511e-04],
       [1.02878894e-03, 1.31244361e-03, 8.42223470e-04, ...,
        1.40914535e-03, 1.94836824e-03, 9.03965511e-04]])
```

In [48]:
```
#sns.histplot(data=df_NC[df_NC["new_death"]=="new"], x=NC_Data, kde=True)
```

NC Data KDE

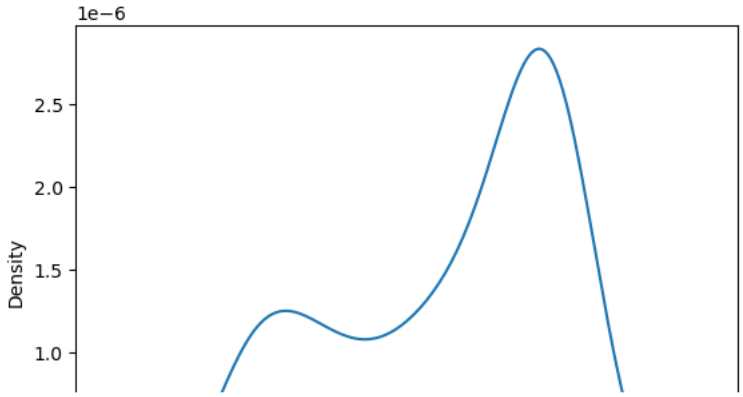
Plotting Distribution

In [49]:
```
sns.histplot(data=df_NC[df_NC["new_death"]=="new"],x="Total_NC", kde = True)
```



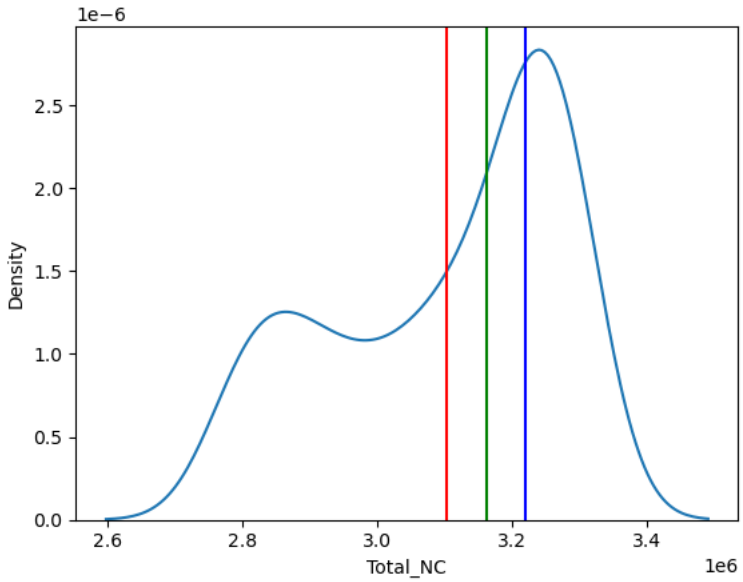
In [50]:
```
sns.kdeplot(data=df_NC[df_NC["new_death"]=="new"],x="Total_NC")
```

Out[50]: <AxesSubplot:xlabel='Total_NC', ylabel='Density'>



In [51]:
```
sns.kdeplot(data=df_NC[df_NC["new_death"]=="new"],x="Total_NC")
plt.axvline(df_NC[df_NC["new_death"]=="new"]["Total_NC"].mean(), color = "red")
plt.axvline(df_NC[df_NC["new_death"]=="new"]["Total_NC"].median(), color = "green")
plt.axvline(df_NC[df_NC["new_death"]=="new"]["Total_NC"].mode()[0], color = "blue")
```

Out[51]: <matplotlib.lines.Line2D at 0x1b15a4bd940>



In [52]:
```
df_NC[df_NC["new_death"]=="new"]["Total_NC"].mean()
```

Out[52]: 3103816.0841121497

In [53]:
```
ncvar=np.var(df_NC[df_NC["new_death"]=="new"]["Total_NC"])
print("Variance of new death cases in NC:", ncvar)
```

```
Variance of new death cases in NC: 28698153394.21727
```

This is a bimodal distribution because it has two peaks. It is skewed to the left. The center is accounted for through the mean, median and mode which is found around 3103816.

Comparing it to VA, GA, SC

VA

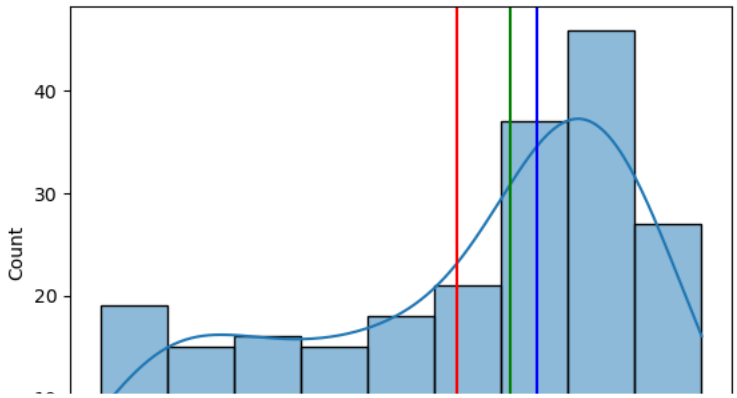
In [70]:
```
col_VA=pd.Series(superdataT.columns)[pd.Series(superdataT.columns).str.contains("_VA")][1:]
col_VA
```

Out[70]:
```
2863        accomack county_VA
2864       albemarle county_VA
2865       alleghany county_VA
2866          amelia county_VA
2867         amherst county_VA
                  ...
2991           suffolk city_VA
2992    virginia beach city_VA
2993        waynesboro city_VA
2994      williamsburg city_VA
2995        winchester city_VA
Name: county_state, Length: 133, dtype: object
```

In [71]:
```
df_VA = superdataT[col_VA].copy()
df_VA["Total_VA"]=df_VA.sum(axis=1)
df_VA["new_death"]=superdataT["new_death"]
df_VA["Week"]=superdataT["Week"]
```

In [89]:
```
sns.histplot(data=df_VA[df_VA["new_death"]=="new"],x="Total_VA", kde = True)
plt.axvline(df_VA[df_VA["new_death"]=="new"]["Total_VA"].mean(), color = "red")
plt.axvline(df_VA[df_VA["new_death"]=="new"]["Total_VA"].median(), color = "green")
plt.axvline(df_VA[df_VA["new_death"]=="new"]["Total_VA"].mode()[0], color = "blue")
```

Out[89]:
```
<matplotlib.lines.Line2D at 0x1b163607ee0>
```



In [90]:
```
df_VA[df_VA["new_death"]=="new"]["Total_VA"].mean()
```

Out[90]:
```
2032707.1074766356
```

GA

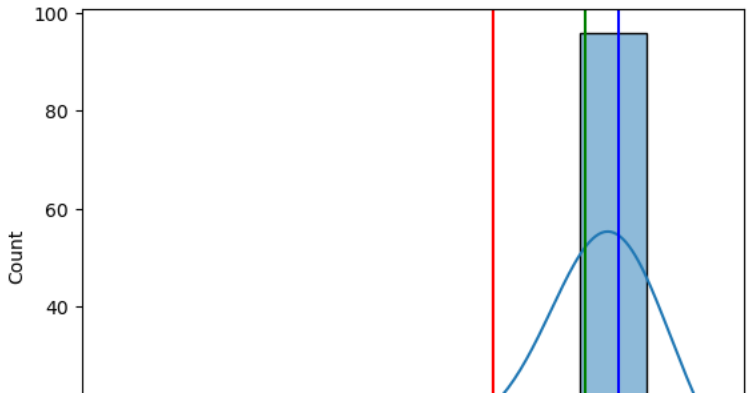
In [73]:
```
col_GA=pd.Series(superdataT.columns)[pd.Series(superdataT.columns).str.contains("_GA")][1:]
col_GA
```

Out[73]:
```
395      appling county_GA
396     atkinson county_GA
397        bacon county_GA
398        baker county_GA
399      baldwin county_GA
              ...
549    whitfield county_GA
550       wilcox county_GA
551       wilkes county_GA
552    wilkinson county_GA
553        worth county_GA
Name: county_state, Length: 159, dtype: object
```

In [78]:
```
df_GA = superdataT[col_GA].copy()
df_GA["Total_GA"]=df_GA.sum(axis=1)
df_GA["Week"]=superdataT["Week"]
df_GA["new_death"]=superdataT["new_death"]
```

In [91]:
```
sns.histplot(data=df_GA[df_GA["new_death"]=="new"],x="Total_GA", kde = True)
plt.axvline(df_GA[df_GA["new_death"]=="new"]["Total_GA"].mean(), color = "red")
plt.axvline(df_GA[df_GA["new_death"]=="new"]["Total_GA"].median(), color = "green")
plt.axvline(df_GA[df_GA["new_death"]=="new"]["Total_GA"].mode()[0], color = "blue")
```

Out[91]:
```
<matplotlib.lines.Line2D at 0x1b1637270d0>
```



In [92]:
```
df_GA[df_GA["new_death"]=="new"]["Total_GA"].mean()
```

Out[92]:
```
2096101.238317757
```

TN

In [84]:
```
col_TN=pd.Series(superdataT.columns)[pd.Series(superdataT.columns).str.contains("_TN")][1:]
col_TN
```

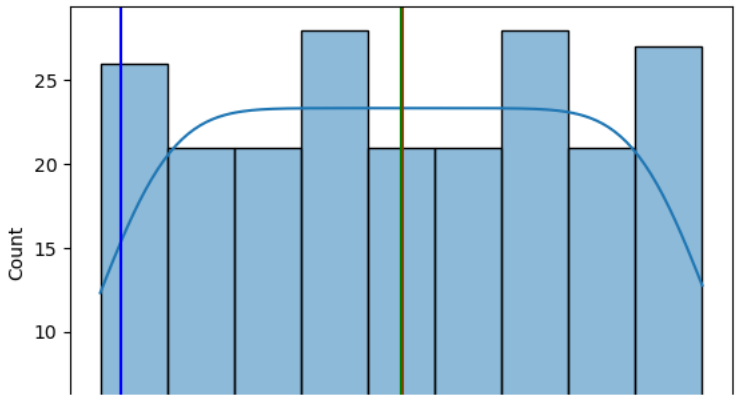
Out[84]:
```
2467      anderson county_TN
2468       bedford county_TN
2469        benton county_TN
2470       bledsoe county_TN
2471        blount county_TN
              ...
2557         wayne county_TN
2558       weakley county_TN
2559         white county_TN
2560    williamson county_TN
2561        wilson county_TN
Name: county_state, Length: 95, dtype: object
```

In [85]:
```
df_TN=superdataT[col_TN].copy()
df_TN["Total_TN"]=df_sc.sum(axis=1)
df_TN["new_death"]=superdataT["new_death"]
df_TN["Week"]=superdataT["Week"]
```

```
C:\Users\amyme\AppData\Local\Temp\ipykernel_14660\3633611855.py:2: 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 colum
ns before calling the reduction.
  df_TN["Total_TN"]=df_sc.sum(axis=1)
```

In [93]:
```
sns.histplot(data=df_TN[df_TN["new_death"]=="new"],x="Total_TN", kde = True)
plt.axvline(df_TN[df_TN["new_death"]=="new"]["Total_TN"].mean(), color = "red")
plt.axvline(df_TN[df_TN["new_death"]=="new"]["Total_TN"].median(), color = "green")
plt.axvline(df_TN[df_TN["new_death"]=="new"]["Total_TN"].mode()[0], color = "blue")
```

Out[93]: <matplotlib.lines.Line2D at 0x1b15a467400>



In [94]:
```
df_TN[df_TN["new_death"]=="new"]["Total_TN"].mean()
```

Out[94]: 2963329.070093458

In [104]:
```
norm_VA = (df_VA[df_VA["new_death"]=="new"]["Total_VA"].mean())/(superdata[superdata.StateFIPS==51].iloc[:,-1:].sum())
print(norm_VA)
```

```
population    0.238147
dtype: float64
```

In [105]:
```
norm_GA = (df_GA[df_GA["new_death"]=="new"]["Total_GA"].mean())/(superdata[superdata.StateFIPS==13].iloc[:,-1:].sum())
print(norm_GA)
```

```
population    0.197421
dtype: float64
```

In [106]:
```
norm_TN = (df_TN[df_TN["new_death"]=="new"]["Total_TN"].mean())/(superdata[superdata.StateFIPS==47].iloc[:,-1:].sum())
print(norm_TN)
```

```
population    0.433922
dtype: float64
```

With the normalized mean values of Virgina having a mean of .238, Georgia having a mean of .197, and Tennesse a mean of .434 we can observe that Georgia had the most pragmatic approach to covid since it had a lower case mean. Tennesse had high cases for a prolonged period of time, as opposed to Georgia, North Carolina and Virgina that had two high points around the holidays and school beginnings which made the distributions bimodal.

In [54]:
```
df2 = pd.read_csv("EconomicCharac.csv")
```

```
C:\Users\amyme\AppData\Local\Temp\ipykernel_14660\2641987803.py:1: DtypeWarning: Columns (2,3,4,5,6,7,8,9,10,11,12,13,14,15,
16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,98,99,100,101,182,183,184,185,186,187,188,189,190,191,192,193,194,195,
196,197,198,199,200,201,202,203,204,205,206,207,208,209,210,211,212,213,214,215,216,217,218,219,220,221,222,223,224,225,226,
227,228,229,230,231,232,233,234,235,236,237,238,239,240,241,242,243,244,245,246,247,248,249,250,251,252,253,254,255,256,257,
258,259,260,261,262,263,264,265,266,267,268,269,270,271,272,273,274,275,276,277,278,279,280,281,286,287,288,289,294,295,296,
297,298,299,300,301,302,303,304,305,306,307,308,309,310,311,312,313,314,315,316,317,318,319,320,321,322,323,324,325,326,327,
328,329,330,331,332,333,334,335,336,337,338,339,340,341,342,343,344,345,346,347,348,349,350,351,352,353,358,359,360,361,366,
367,368,369,370,371,372,373,374,375,376,377,378,379,380,381,382,383,384,385,386,387,388,389,390,391,392,393,394,395,396,397,
398,399,400,401,402,403,404,405,406,407,408,409,410,411,412,413,414,415,416,417,418,419,420,421,422,423,424,425,426,427,428,
429,430,431,432,433,434,435,436,437,438,439,440,441,442,443,444,445,446,447,448,449,450,451,452,453,454,455,456,457,458,459,
460,461,462,463,464,465,466,467,468,469,470,471,472,473,550,551,554,555,556,557,558,559,560,561,562,563,564,565,566,567,568,
569,570,571,572,573,574,575,576,577,578,579,582,583,584,585,730,733,734,735,736,737,738,739,740,741,742,743,744,745,746,747,
748,749,750,751,754,755,756,757,758,759,760,761,762,763,764,765,766,767,768,769,770,771,772,773,774,775,776,777,778,779,780,
781,782,783,784,785,786,787,788,789,790,791,792,793,802,803,804,805,810,811,812,813,818,819,820,821,826,827,828,829,834,835,
836,837,842,843,844,845,846,847,850,851,852,853,854,855,856,857,858,859,860,861,862,863,864,865,866,867,868,869,870,871,872,
873,874,875,876,877,878,879,880,881,882,883,884,885,886,887,888,889,926,929,930,931,932,933,934,935,936,937,938,939,940,941,
942,943,944,945,946,949,950,951,952,953,954,955,958,959,962,965,966,967,968,969,970,971,972,973,974,975,976,977,978,979,980,
981,982,983,986,987,988,989,990,991,992,993,994,995,996,997,998,999,1000,1001,1002,1003,1006,1007,1008,1009,1010,1011,1012,1
013,1014,1015,1016,1017,1018,1019,1020,1021,1022,1023,1024,1025,1026,1027,1028,1029,1030,1031,1032,1033,1034,1035,1036,1037,
1038,1039,1040,1041,1042,1043,1044,1045,1046,1047,1048,1049,1050,1051,1052,1053,1054,1055,1056,1057,1058,1059,1060,1061,106
2,1063,1064,1065,1066,1067,1068,1069,1070,1071,1072,1073,1074,1075,1076,1077,1078,1079,1080,1081,1082,1083,1084,1085,1086,10
87,1088,1089,1090,1091,1092,1093,1094,1095,1096,1097) have mixed types. Specify dtype option on import or set low_memory=Fal
se.
  df2 = pd.read_csv("EconomicCharac.csv")
```

In [55]:
```
df2.columns=df2.iloc[0]
```

In [95]:
```
df2.head()
```

Out[95]:

ercent Margin of !!PERCENTAGE F FAMILIES AND EOPLE WHOSE INCOME IN THE 12 MONTHS IS BELOW THE POVERTY LEVEL!!All eople!!People in families	Annotation of Percent Margin of Error!!PERCENTAGE OF FAMILIES AND PEOPLE WHOSE INCOME IN THE PAST 12 MONTHS IS BELOW THE POVERTY LEVEL!!All people!!People in families	Annotation of Percent!!PERCENTAGE OF FAMILIES AND PEOPLE WHOSE INCOME IN THE PAST 12 MONTHS IS BELOW THE POVERTY LEVEL!!All people!!People in families	Percent!!PERCENTAGE OF FAMILIES AND PEOPLE WHOSE INCOME IN THE PAST 12 MONTHS IS BELOW THE POVERTY LEVEL!!All people!!Unrelated individuals 15 years and over	Annotation of Percent!!PERCENTAGE OF FAMILIES AND PEOPLE WHOSE INCOME IN THE PAST 12 MONTHS IS BELOW THE POVERTY LEVEL!!All people!!Unrelated individuals 15 years and over	Percent Margin of Error!!PERCENTAGE OF FAMILIES AND PEOPLE WHOSE INCOME IN THE PAST 12 MONTHS IS BELOW THE POVERTY LEVEL!!All people!!Unrelated individuals 15 years and over	Annotation of Percent Margin of Error!!PERCENTAGE OF FAMILIES AND PEOPLE WHOSE INCOME IN THE PAST 12 MONTHS IS BELOW THE POVERTY LEVEL!!All people!!Unrelated individuals 15 years and over	NaN
2.9	NaN	NaN	21.7	NaN	4.8	NaN	NaN
2.2	NaN	NaN	21.2	NaN	5.2	NaN	NaN
2.2	NaN	NaN	22.1	NaN	3.8	NaN	NaN
4.1	NaN	NaN	30.1	NaN	7.6	NaN	NaN
2.7	NaN	NaN	23.9	NaN	6.6	NaN	NaN

In [56]:
```
df2["Geographic Area Name"].str.contains("North Carolina")
```

Out[56]:
```
0      False
1      False
2      False
3      False
4      False
       ...
837    False
838    False
839    False
840    False
841    False
Name: Geographic Area Name, Length: 842, dtype: bool
```

In [57]:
```
df2 = df2[df2["Geographic Area Name"].str.contains("North Carolina")].copy()
```

In [58]:
```
df2.index=df2["Geographic Area Name"]
```

In [60]:
```
df_NC.head
```

Out[60]:

county_state	alamance county_NC	alexander county_NC	alleghany county_NC	anson county_NC	ashe county_NC	avery county_NC	beaufort county_NC	bertie county_NC	bladen county_NC	brunswick county_NC	...	washington county_NC	co
0	49188	10600	3041	6672	6575	4697	12939	4353	9608	31630	...	3091	
1	49188	10600	3041	6672	6575	4697	12939	4353	9608	31630	...	3091	
2	49188	10600	3041	6672	6575	4697	12939	4353	9608	31630	...	3091	
3	49188	10600	3041	6672	6575	4697	12939	4353	9608	31630	...	3091	
4	49188	10600	3041	6672	6575	4697	12939	4353	9608	31630	...	3091	

5 rows × 103 columns

In [61]:
```
econNC = df2["Estimate!!EMPLOYMENT STATUS!!Population 16 years and over!!In labor force!!Civilian labor force"].astype("int64
```

In [62]:
```
print(econNC)
```

```
Johnston County, North Carolina         113540
Lincoln County, North Carolina           44263
Mecklenburg County, North Carolina      637790
Moore County, North Carolina             44221
Nash County, North Carolina              48093
New Hanover County, North Carolina      123929
Onslow County, North Carolina            69713
Orange County, North Carolina            77574
Pitt County, North Carolina              86513
Randolph County, North Carolina          69598
Robeson County, North Carolina           46973
Rockingham County, North Carolina        40711
Rowan County, North Carolina             69120
Surry County, North Carolina             32965
Union County, North Carolina            127438
Wake County, North Carolina             633630
Wayne County, North Carolina             51839
Wilkes County, North Carolina            28028
Wilson County, North Carolina            34789
Name: Estimate!!EMPLOYMENT STATUS!!Population 16 years and over!!In labor force!!Civilian labor force, dtype: int64
```

In [63]:
```
econNC.info()
```

```
<class 'pandas.core.series.Series'>
Index: 40 entries, Alamance County, North Carolina to Wilson County, North Carolina
Series name: Estimate!!EMPLOYMENT STATUS!!Population 16 years and over!!In labor force!!Civilian labor force
Non-Null Count  Dtype
--------------  -----
40 non-null     int64
dtypes: int64(1)
memory usage: 640.0+ bytes
```

In [64]:
```
len(econNC)
```

Out[64]:
```
40
```

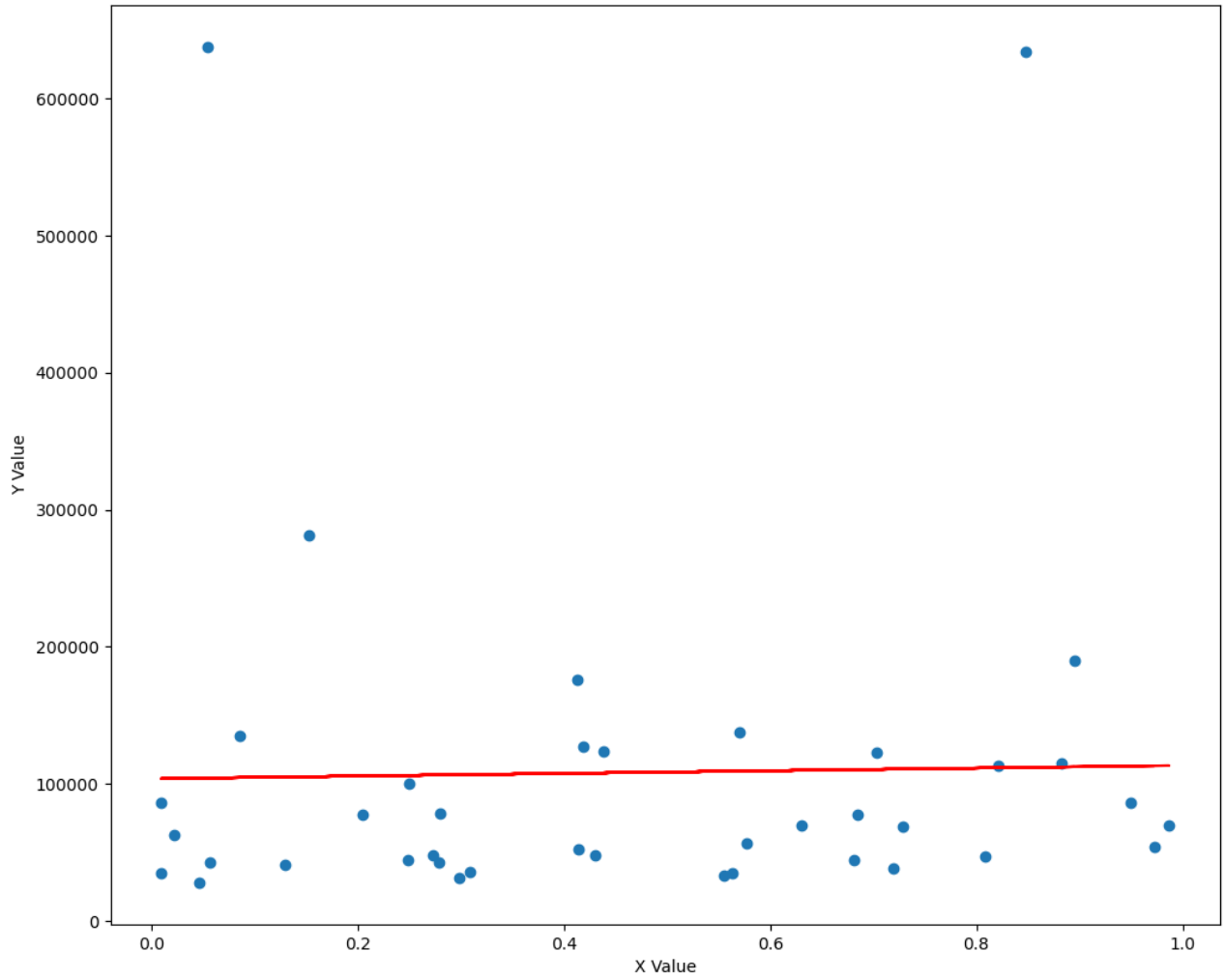
I see that the arrays have to be the same size but I'm not sure how to make them the same size here

In [88]:
```
x1 = NC_Data
y1 = econNC
corr_matrix = np.corrcoef(x1,y1)
corr_coef = corr_matrix[0,1]
print("Correlation coeeficient: ", corr_coef)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_14660\300280163.py in <module>
      1 x1 = NC_Data
      2 y1 = econNC
----> 3 corr_matrix = np.corrcoef(x1,y1)
      4 corr_coef = corr_matrix[0,1]
      5 print("Correlation coeeficient: ", corr_coef)

<__array_function__ internals> in corrcoef(*args, **kwargs)

C:\ProgramData\Anaconda3\lib\site-packages\numpy\lib\function_base.py in corrcoef(x, y, rowvar, bias, ddof, dtype)
   2681         warnings.warn('bias and ddof have no effect and are deprecated',
   2682                       DeprecationWarning, stacklevel=3)
-> 2683     c = cov(x, y, rowvar, dtype=dtype)
   2684     try:
   2685         d = diag(c)

<__array_function__ internals> in cov(*args, **kwargs)

C:\ProgramData\Anaconda3\lib\site-packages\numpy\lib\function_base.py in cov(m, y, rowvar, bias, ddof, fweights, aweights, d
type)
   2475         if not rowvar and y.shape[0] != 1:
   2476             y = y.T
-> 2477         X = np.concatenate((X, y), axis=0)
   2478
   2479     if ddof is None:

<__array_function__ internals> in concatenate(*args, **kwargs)

ValueError: all the input array dimensions for the concatenation axis must match exactly, but along dimension 1, the array a
t index 0 has size 97 and the array at index 1 has size 40
```

It would only work if my other array also had 40 entries

In [66]:
```
X =np.random.rand(40)
Y = X + econNC

m, b = np.polyfit(X, Y, 1)

plt.figure(figsize=(12,10))
plt.scatter(X,Y)
plt.xlabel('X Value')
plt.ylabel('Y Value')
plt.plot(X, m*X + b, 'r-')
plt.show()
print('Correlation of X and Y: %.2f'%np.corrcoef(X, Y)[0, 1])
```




```
Correlation of X and Y: 0.02
```

Here is another attempt at correlation:

In [96]:
```
corr = df_NC[df_NC["new_death"]=="new"]["Total_NC"].corr(df2["Estimate!!EMPLOYMENT STATUS!!Population 16 years and over!!In l
print(corr)
```

```
nan
```

However, there are no null values in the data

In [97]:
```
df2["Estimate!!EMPLOYMENT STATUS!!Population 16 years and over!!In labor force!!Civilian labor force"].isnull().sum()
```

Out[97]: 0

This is another try at correlation:

In [99]:
```
df2["Estimate!!EMPLOYMENT STATUS!!Population 16 years and over!!In labor force!!Civilian labor force"].astype("int64").copy()
```

In [100]:
```
NCcorr = df_NC[df_NC["new_death"]=="new"]["Total_NC"].corr(dfint)
```

In [101]:
```
print(NCcorr)
```

```
nan
```

Hypothesis:

Is employment status negatively correlated to higher covid cases in North Carolina?

Does the PERCENTAGE OF FAMILIES AND PEOPLE WHOSE INCOME IN THE PAST 12 MONTHS IS BELOW THE POVERTY LEVEL increase as cases rise?

Does unemployment rate positively correlated to higher covid cases?

In [ ]:
```

```