In [1]:
```
from urllib.request import urlopen
import json
with urlopen('https://raw.githubusercontent.com/plotly/datasets/master/geojson-counties-fips.json') as r
    counties = json.load(response)
```

In [2]:
```
import pandas as pd
import numpy
df = pd.read_csv("County_food_insecurity_2017.csv",
                 dtype={"fips": str})

import plotly.express as px
#df['2017 Food Insecurity Rate'] = df['2017 Food Insecurity Rate'].str.replace(r'\D', '')
#df["2017 Food Insecurity Rate"] = df["2017 Food Insecurity Rate"].astype(float)
df["2017 Food Insecurity Rate"].max()
```

Out[2]:
```
0.363
```

In [3]:
```
df['FIPS_STR']=df["FIPS"].astype(str)
df.loc[df['FIPS_STR'].str.len() == 4, 'FIPS'] = '0'+ df['FIPS_STR']
df.loc[df['FIPS_STR'].str.len() > 4, 'FIPS'] = df['FIPS_STR']
```

In [4]:
```
df.style.format({
    '2017 Food Insecurity Rate': '{:,.2%}'.format,
})
```

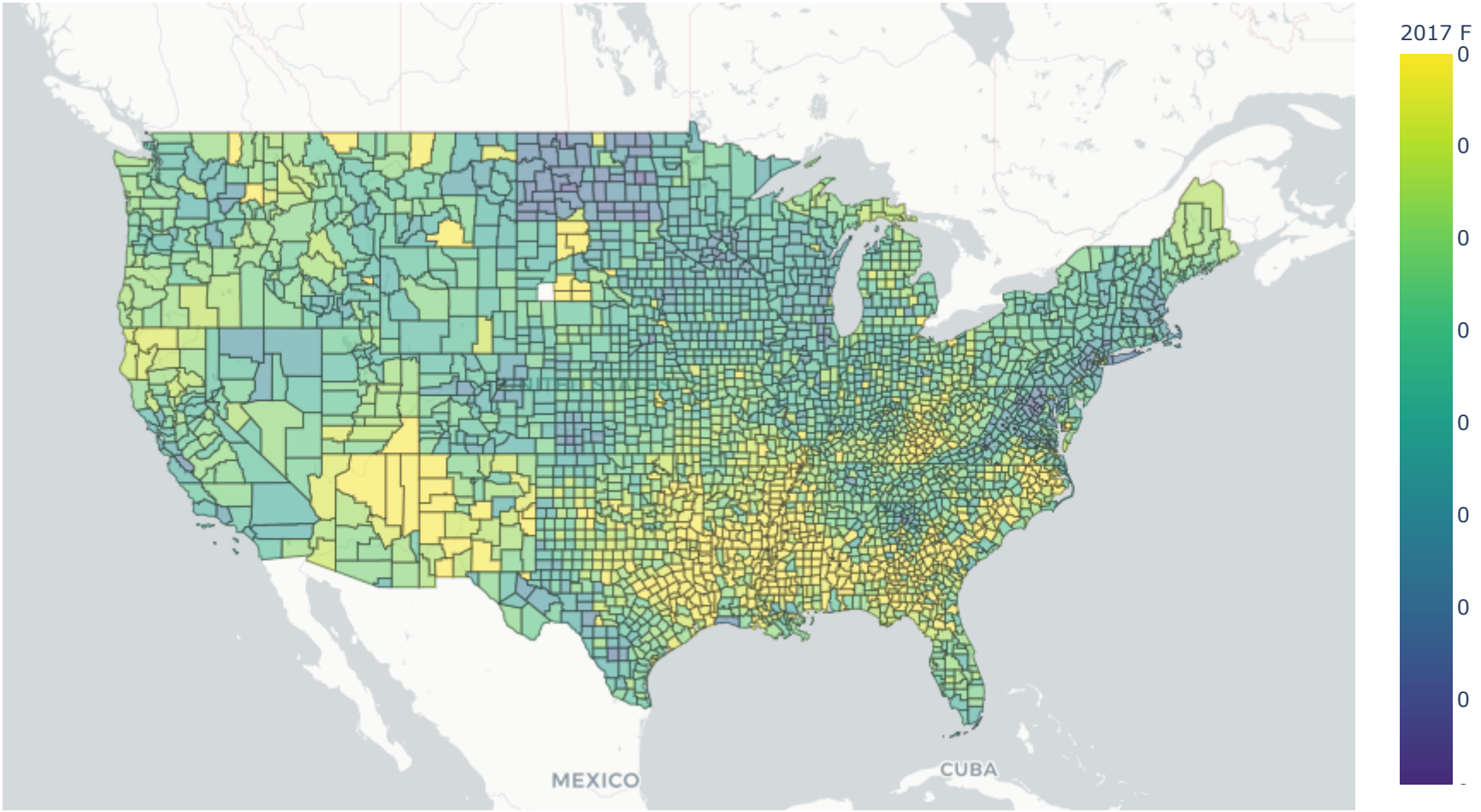
Out[4]:

	FIPS	State	County, State	2017 Food Insecurity Rate	# of Food Insecure Persons in 2017	Low Threshold in state	Low Threshold Type	High Threshold in state	High Threshold Type	% FI ≤ Low Threshold	% FI Btwn Thresholds	% FI > High Threshold
0	01001	AL	Autauga County, Alabama	13.20%	7,270	130%	SNAP	185%	Other Nutrition Program	48.50%	14.40%	37.10%
1	01003	AL	Baldwin County, Alabama	11.60%	23,560	130%	SNAP	185%	Other Nutrition Program	45.50%	14.40%	40.10%
2	01005	AL	Barbour County, Alabama	22.00%	5,760	130%	SNAP	185%	Other Nutrition Program	60.10%	17.10%	22.90%

In [5]:
```
df['FIPS'][0:10]
```

Out[5]:
```
0    01001
1    01003
2    01005
3    01007
4    01009
5    01011
6    01013
7    01015
8    01017
9    01019
Name: FIPS, dtype: object
```

In [6]:
```
import plotly.express as px
fig = px.choropleth_mapbox(df, geojson=counties,
                           locations='FIPS', color='2017 Food Insecurity Rate',
                           color_continuous_scale="Viridis",
                           range_color=(0, .18),
                           mapbox_style="carto-positron",
                           zoom=3, center = {"lat": 37.0902, "lon": -95.7129},
                           opacity=0.5,
                           hover_name='County, State',
                           labels={'Food insequrity in percentages'}
                          )
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```



In [7]:
```
import pandas as pd
import os
os.getcwd()
#url = 'https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_time_ser
df_cd19 = pd.read_csv("time_series_covid19_confirmed_US.csv")
df_cd19['FIPS_STR']=df_cd19["FIPS"].astype(str)
df_cd19['FIPS_STR'] = df_cd19['FIPS_STR'].replace('\.0', '', regex=True)
```

In [8]:
```
df_cd19.loc[df_cd19['FIPS_STR'].str.len() == 4, 'FIPS'] = '0'+ df_cd19['FIPS_STR']
df_cd19.loc[df_cd19['FIPS_STR'].str.len() == 5, 'FIPS'] = df_cd19['FIPS_STR']

df_cd19_rd = df_cd19.loc[df_cd19['FIPS'].str.len() == 5]

#df_cd19_rd['county_state'] = str(df_cd19_rd['Province_State'] + ' ' + df_cd19_rd['Admin2'])
df_cd19_rd['county_state'] = df_cd19_rd['Admin2'] + f", " + df_cd19_rd['Province_State']
df_cd19_rd.head()
```

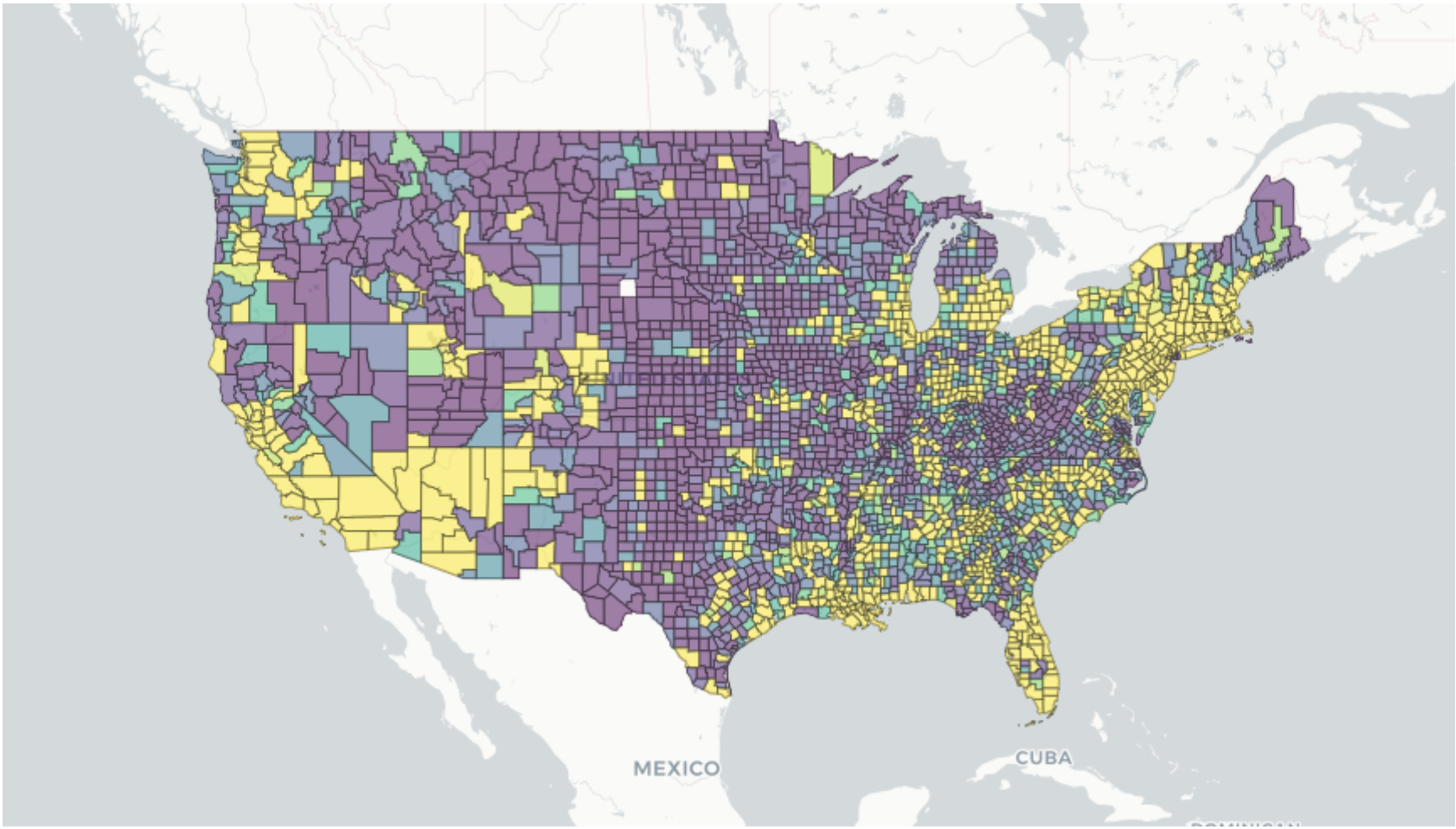
```
/Users/huiyingzheng/opt/anaconda3/envs/PythonData/lib/python3.6/site-packages/ipykernel_launcher.py:7:
SettingWithCopyWarning:


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 (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
```

Out[8]:

	UID	iso2	iso3	code3	FIPS	Admin2	Province_State	Country_Region	Lat	Long_	...	4/13/20	4/14/20	4/15/20	4/
5	84001001	US	USA	840	01001	Autauga	Alabama	US	32.539527	-86.644082	...	19	23	24	
6	84001003	US	USA	840	01003	Baldwin	Alabama	US	30.727750	-87.722071	...	72	87	91	
7	84001005	US	USA	840	01005	Barbour	Alabama	US	31.868263	-85.387129	...	10	11	12	
8	84001007	US	USA	840	01007	Bibb	Alabama	US	32.996421	-87.125115	...	17	17	18	
9	84001009	US	USA	840	01009	Blount	Alabama	US	33.982109	-86.567906	...	14	16	17	

5 rows × 103 columns

In [9]:
```
#import plotly.express as px
fig = px.choropleth_mapbox(df_cd19_rd, geojson=counties,
                           locations='FIPS', color='4/18/20',
                           color_continuous_scale="Viridis",
                           range_color=(0, 50),
                           mapbox_style="carto-positron",
                           zoom=3, center = {"lat": 37.0902, "lon": -95.7129},
                           opacity=0.5,
                           hover_name='county_state',
                           labels={'COVID19 Number of Infections'}
                          )
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```



In [10]:
```
df_cd19_mort = pd.read_csv("time_series_covid19_deaths_US.csv")
df_cd19_mort['FIPS_STR']=df_cd19_mort["FIPS"].astype(str)
df_cd19_mort['FIPS_STR'] = df_cd19_mort['FIPS_STR'].replace('\.0', '', regex=True)
df_cd19_mort['FIPS_STR']
```

Out[10]:
```
0        60
1        66
2        69
3        72
4        78
       ...
3256    nan
3257    nan
3258    nan
3259    nan
3260    nan
Name: FIPS_STR, Length: 3261, dtype: object
```

In [11]:
```
df_cd19_mort.loc[df_cd19_mort['FIPS_STR'].str.len() == 4, 'FIPS'] = '0'+ df_cd19_mort['FIPS_STR']
df_cd19_mort.loc[df_cd19_mort['FIPS_STR'].str.len() == 5, 'FIPS'] = df_cd19_mort['FIPS_STR']

df_cd19_mort_rd = df_cd19_mort.loc[df_cd19['FIPS'].str.len() == 5]

#df_cd19_rd['county_state'] = str(df_cd19_rd['Province_State'] + ' ' + df_cd19_rd['Admin2'])
df_cd19_mort_rd['county_state'] = df_cd19_mort_rd['Admin2'] + f", " + df_cd19_mort_rd['Province_State']
df_cd19_mort_rd.head()
```

```
/Users/huiyingzheng/opt/anaconda3/envs/PythonData/lib/python3.6/site-packages/ipykernel_launcher.py:7:
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 (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
```

Out[11]:

	UID	iso2	iso3	code3	FIPS	Admin2	Province_State	Country_Region	Lat	Long_	...	4/13/20	4/14/20	4/15/20
5	84001001	US	USA	840	01001	Autauga	Alabama	US	32.539527	-86.644082	...	1	1	1
6	84001003	US	USA	840	01003	Baldwin	Alabama	US	30.727750	-87.722071	...	1	2	2
7	84001005	US	USA	840	01005	Barbour	Alabama	US	31.868263	-85.387129	...	0	0	0
8	84001007	US	USA	840	01007	Bibb	Alabama	US	32.996421	-87.125115	...	0	0	0
9	84001009	US	USA	840	01009	Blount	Alabama	US	33.982109	-86.567906	...	0	0	0

5 rows × 104 columns

In [12]:
```
df_cd19_mort_rd.describe()
```

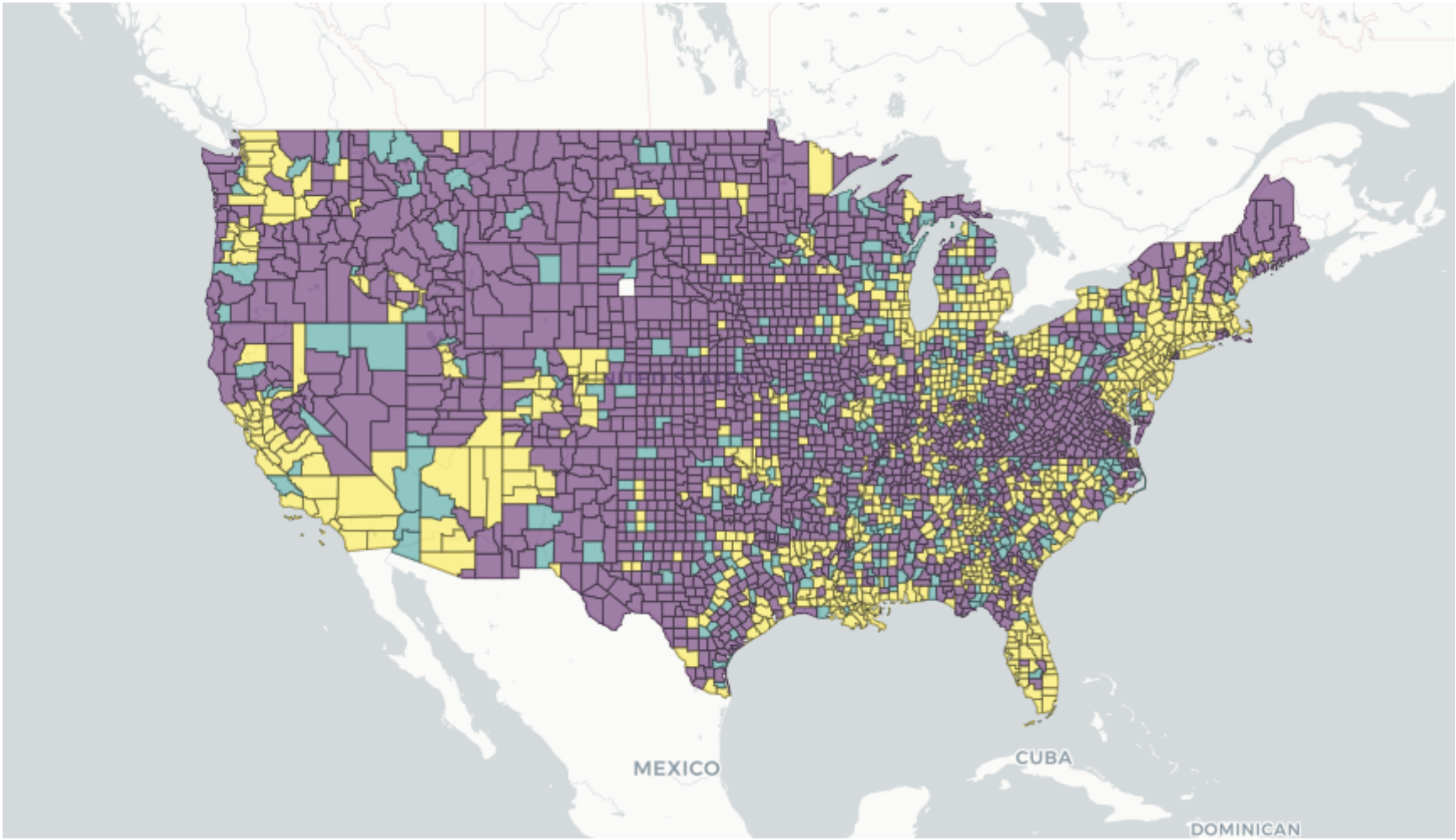
Out[12]:

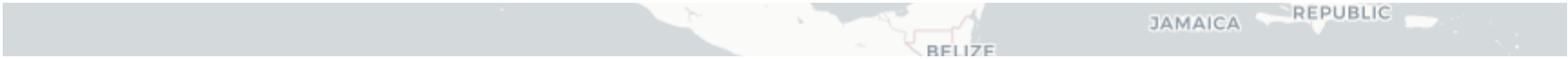
	UID	code3	Lat	Long_	Population	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	...	4/11/20	4
count	3.246000e+03	3246.0	3246.000000	3246.000000	3.246000e+03	3246.0	3246.0	3246.0	3246.0	3246.0	...	3246.000000	3246.0
mean	8.403214e+07	840.0	37.223517	-89.312179	1.024073e+05	0.0	0.0	0.0	0.0	0.0	...	6.285582	6.7
std	1.779447e+04	0.0	8.552691	20.597305	3.424430e+05	0.0	0.0	0.0	0.0	0.0	...	115.602872	125.1
min	8.400100e+07	840.0	0.000000	-164.035380	0.000000e+00	0.0	0.0	0.0	0.0	0.0	...	0.000000	0.0
25%	8.401905e+07	840.0	34.253077	-98.044952	9.831250e+03	0.0	0.0	0.0	0.0	0.0	...	0.000000	0.0
50%	8.403005e+07	840.0	38.154335	-89.827060	2.458750e+04	0.0	0.0	0.0	0.0	0.0	...	0.000000	0.0
75%	8.404701e+07	840.0	41.684874	-82.822027	6.568550e+04	0.0	0.0	0.0	0.0	0.0	...	1.000000	1.0
max	8.410000e+07	840.0	69.314792	0.000000	1.003911e+07	0.0	0.0	0.0	0.0	0.0	...	6367.000000	6898.0

8 rows × 95 columns

In [ ]:
```

```

In [13]:
```
#import plotly.express as px
fig = px.choropleth_mapbox(df_cd19_mort_rd, geojson=counties,
                           locations='FIPS', color='4/18/20',
                           color_continuous_scale="Viridis",
                           range_color=(0, 2),
                           mapbox_style="carto-positron",
                           zoom=3, center = {"lat": 37.0902, "lon": -95.7129},
                           opacity=0.5,
                           hover_name='county_state',
                           labels={'COVID19 Number of Deaths'}
                          )
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```





In [14]:
```
# Total number of counties
df_cd19_mort_rd["FIPS"].count()
```

Out[14]: 3246

In [15]:
```
df_cd19_mort_rd_merge1 = df_cd19_mort_rd[['FIPS','4/19/20','county_state']]
df_cd19_mort_rd_merge1 = df_cd19_mort_rd_merge1.rename(columns={"4/19/20": "4/19/20 Number of Death"})
df_cd19_mort_rd_merge1.head()
```

Out[15]:

	FIPS	4/19/20 Number of Death	county_state
5	01001	2	Autauga, Alabama
6	01003	2	Baldwin, Alabama
7	01005	0	Barbour, Alabama
8	01007	0	Bibb, Alabama
9	01009	0	Blount, Alabama

In [16]:
```
df_cd19_rd_merge1 = df_cd19_rd[['FIPS','4/19/20']]
df_cd19_rd_merge1 = df_cd19_rd_merge1.rename(columns={"4/19/20": "4/19/20 Number of Infected"})
df_cd19_rd_merge1.head()
```

Out[16]:

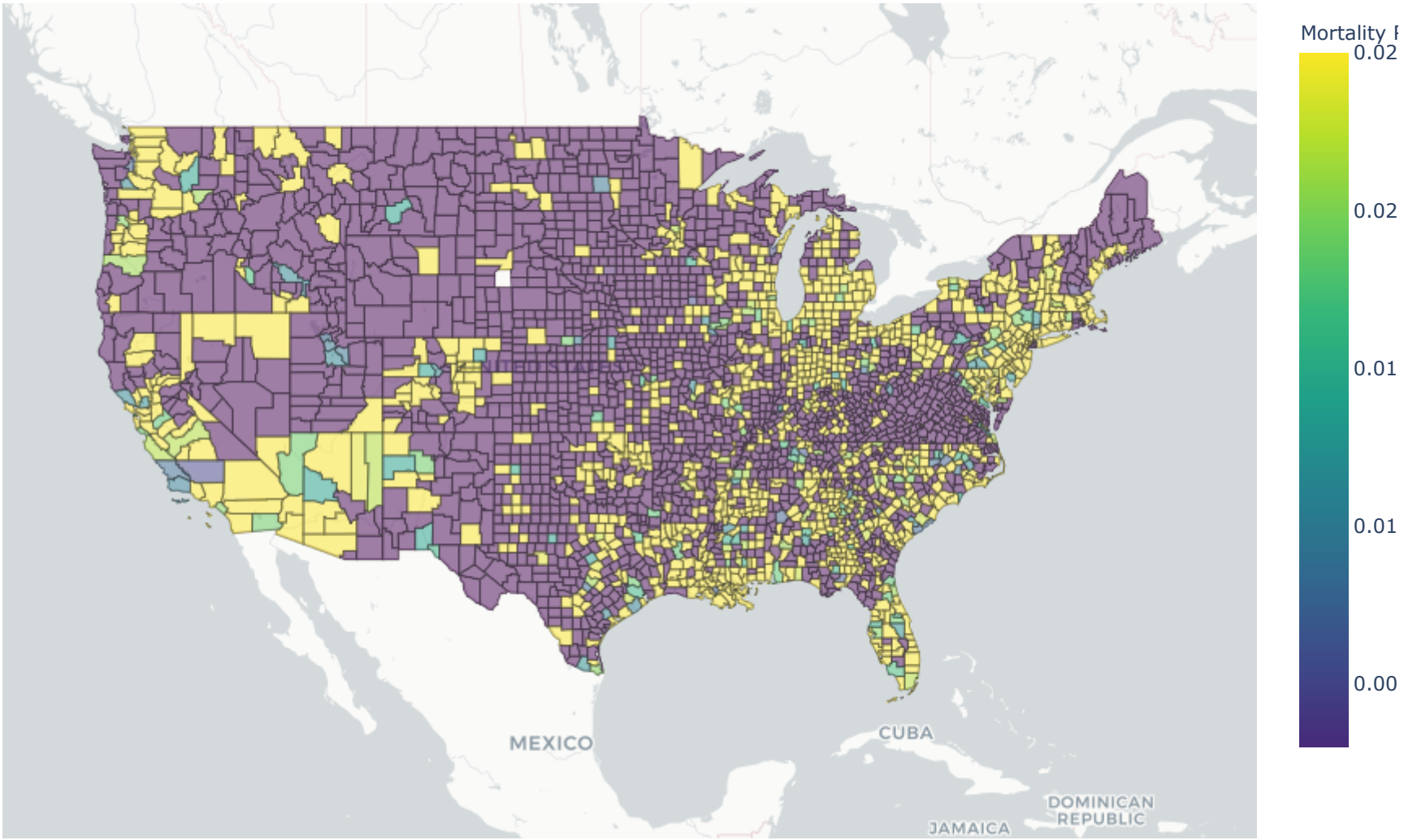
	FIPS	4/19/20 Number of Infected
5	01001	26
6	01003	112
7	01005	20
8	01007	28
9	01009	21

In [17]:
```
df_cd19_mort_final = pd.merge(df_cd19_rd_merge1, df_cd19_mort_rd_merge1, on='FIPS')
df_cd19_mort_final.loc[df_cd19_mort_final['4/19/20 Number of Infected'] > 0, 'Mortality Rate (County Lev
df_cd19_mort_final.loc[df_cd19_mort_final['4/19/20 Number of Infected'] == 0, 'Mortality Rate (County Le
```

In [18]:
```
#import numpy as np
#df_cd19_mort_final = df_cd19_mort_final.replace([np.inf, -np.inf], 0)
df_cd19_mort_final["Mortality Rate (County Level)"] = round(df_cd19_mort_final["Mortality Rate (County I
df_cd19_mort_final["Mortality Rate (County Level)"]
```

Out[18]:
```
0       0.077
1       0.018
2       0.000
3       0.000
4       0.000
       ...
3241    0.000
3242    0.000
3243    0.000
3244    0.000
3245    0.000
Name: Mortality Rate (County Level), Length: 3246, dtype: float64
```

In [19]:
```
fig = px.choropleth_mapbox(df_cd19_mort_final, geojson=counties,
                           locations='FIPS', color='Mortality Rate (County Level)',
                           color_continuous_scale="Viridis",
                           range_color=(0, 0.025),
                           mapbox_style="carto-positron",
                           zoom=3, center = {"lat": 37.0902, "lon": -95.7129},
                           opacity=0.5,
                           hover_name='county_state',
                           labels={'COVID19 Number of Deaths'}
                          )
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```



BELIZE

In [20]:
```
df['2017 Food Insecurity Rate']
df['FIPS']

df_food = df[['FIPS','2017 Food Insecurity Rate']]
df_food
```

Out[20]:

	FIPS	2017 Food Insecurity Rate
0	01001	0.132
1	01003	0.116
2	01005	0.220
3	01007	0.143
4	01009	0.107
...	...	...
3137	56037	0.107
3138	56039	0.097
3139	56041	0.128
3140	56043	0.112
3141	56045	0.131

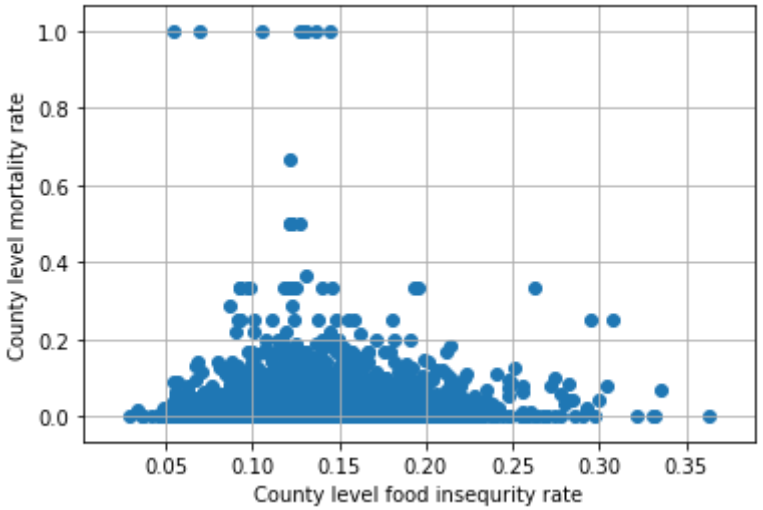
3142 rows × 2 columns

In [21]:
```
df_cd19_food = pd.merge(df_cd19_mort_final, df_food, on='FIPS')
df_cd19_food.head()
df_cd19_food.describe()
```

Out[21]:

	4/19/20 Number of Infected	4/19/20 Number of Death	Mortality Rate (County Level)	2017 Food Insecurity Rate
count	3142.000000	3142.000000	3142.000000	3142.000000
mean	238.917568	12.661044	0.027793	0.132508
std	2752.408561	262.921666	0.070478	0.039603
min	0.000000	0.000000	0.000000	0.029000
25%	2.000000	0.000000	0.000000	0.106000
50%	11.000000	0.000000	0.000000	0.128000
75%	49.750000	2.000000	0.037000	0.152000
max	138700.000000	14451.000000	1.000000	0.363000

In [25]:
```
import matplotlib.pyplot as plt
# Next conduct a regresssion analysis
x_values = df_cd19_food.loc[df_cd19_food["Mortality Rate (County Level)"] <= 1].loc[:,'2017 Food Insecur
y_values = df_cd19_food.loc[df_cd19_food["Mortality Rate (County Level)"] <= 1].loc[:,'Mortality Rate (C
plt.scatter(x_values,y_values)
plt.ylabel("County level mortality rate")
plt.xlabel(f"County level food insequrity rate")
plt.grid()
plt.show()

from scipy.stats import linregress
(slope, intercept, rvalue, pvalue, stderr) = linregress(x_values, y_values)
print(slope, intercept, rvalue, pvalue, stderr)
```




```
0.038585684719504146 0.02268021515718703 0.021681864307566936 0.2243643384617289 0.031751383945363254
```

In [26]:
```
import pandas as pd
import statsmodels.api as sm
from statsmodels.formula.api import poisson
from statsmodels.formula.api import negativebinomial
from statsmodels.genmod.generalized_estimating_equations import GEE
from statsmodels.genmod.cov_struct import (Exchangeable,
    Independence,Autoregressive)
import numpy as np
import matplotlib.pyplot as plt
import math
df_cd19_food.head()
```

Out[26]:

	FIPS	4/19/20 Number of Infected	4/19/20 Number of Death	county_state	Mortality Rate (County Level)	2017 Food Insecurity Rate
0	01001	26	2	Autauga, Alabama	0.077	0.132
1	01003	112	2	Baldwin, Alabama	0.018	0.116
2	01005	20	0	Barbour, Alabama	0.000	0.220
3	01007	28	0	Bibb, Alabama	0.000	0.143
4	01009	21	0	Blount, Alabama	0.000	0.107

In [27]:
```
df_cd19_food.describe()
```

Out[27]:

	4/19/20 Number of Infected	4/19/20 Number of Death	Mortality Rate (County Level)	2017 Food Insecurity Rate
count	3142.000000	3142.000000	3142.000000	3142.000000
mean	238.917568	12.661044	0.027793	0.132508
std	2752.408561	262.921666	0.070478	0.039603
min	0.000000	0.000000	0.000000	0.029000
25%	2.000000	0.000000	0.000000	0.106000
50%	11.000000	0.000000	0.000000	0.128000
75%	49.750000	2.000000	0.037000	0.152000
max	138700.000000	14451.000000	1.000000	0.363000

In [28]:
```
import statsmodels.api as sm
from statsmodels.formula.api import glm

df_cd19_food = df_cd19_food.rename(columns={"4/19/20 Number of Death": "N_MORT",
                                            "4/19/20 Number of Infected": "N_IFTC",
                                            "2017 Food Insecurity Rate": "P_FOOD"})

df_cd19_food_rd = df_cd19_food.dropna()
df_cd19_food_rd = df_cd19_food_rd.loc[df_cd19_food_rd["N_IFTC"]<=150]
df_cd19_food_rd.shape
model = glm('N_MORT ~ P_FOOD', data = df_cd19_food_rd, family = sm.families.Poisson()).fit()
model.summary()
```

Out[28]:

Generalized Linear Model Regression Results

Dep. Variable:	N_MORT	**No. Observations:**	2738
Model:	GLM	**Df Residuals:**	2736
Model Family:	Poisson	**Df Model:**	1
Link Function:	log	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-4296.9
Date:	Sat, 25 Apr 2020	**Deviance:**	6516.3
Time:	11:34:31	**Pearson chi2:**	1.18e+04
No. Iterations:	6		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-0.8245	0.074	-11.098	0.000	-0.970	-0.679
P_FOOD	4.0586	0.504	8.046	0.000	3.070	5.047

In [29]:

```
df_cd19_food = df_cd19_food.rename(columns={"4/19/20 Number of Death": "N_MORT",
                                            "4/19/20 Number of Infected": "N_IFTC",
                                            "2017 Food Insecurity Rate": "P_FOOD"})

df_cd19_food_rd = df_cd19_food.dropna()
df_cd19_food_rd = df_cd19_food_rd.loc[df_cd19_food_rd["N_MORT"]>150]
df_cd19_food_rd.shape
model = glm('N_IFTC ~ P_FOOD', data = df_cd19_food_rd, family = sm.families.Poisson()).fit()
model.summary()
```

Out[29]:

Generalized Linear Model Regression Results

Dep. Variable:	N_IFTC	**No. Observations:**	36
Model:	GLM	**Df Residuals:**	34
Model Family:	Poisson	**Df Model:**	1
Link Function:	log	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-2.9684e+05
Date:	Sat, 25 Apr 2020	**Deviance:**	5.9328e+05
Time:	11:34:51	**Pearson chi2:**	1.40e+06
No. Iterations:	6		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	9.3185	0.004	2222.957	0.000	9.310	9.327
P_FOOD	1.1497	0.036	31.928	0.000	1.079	1.220