CS 533 _ Assignment 2 : Public Health

Author : Farhana Alam

Context and Data

For this assignment, we are going to work with the Census data, with US health data from: http://ghdx.healthdata.org/us-data (http://ghdx.healthdata.org/us-data) . For the census data, we are going to use the American Community Survey (ACS). The 2017 ACS variables are described here: https://api.census.gov/data/2017/acs/acs5/variables.html (https://api.census.gov/data/2017/acs/acs5/variables.html) WE are also going to use the GDP data from the BEA. Also, Infectious Disease Mortality Rates by County data set from: http://ghdx.healthdata.org/record/ihme-data/united-states-infectious-disease-mortality-rates-county-1980-2014 (http://ghdx.healthdata.org/record/ihme-data/united-states-infectious-disease-mortality-rates-county-1980-2014)

Setup

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from census import Census
from us import states
import plotly.figure_factory as ff
import plotly.graph_objects as go
```

After downloading geopandas and plotly turning off those command as comments.

In [2]:

```
##conda install geopandas pyshp shapely
```

In [3]:

```
##conda install -c plotly plotly plotly-geo
```

In [4]:

```
%matplotlib inline
```

In [5]:

```
c = Census('')
```

Project Requirements

Initial Exploration

Loading County files

Using for loop to load all the files.

In [6]:
```
files = ['IHME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_ALABAMA_Y2018M03D27.csv', 'IHME_US
A_COUNTY_INFECT_DIS_MORT_1980_2014_ALASKA_Y2018M03D27.csv',
        'IHME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_ARIZONA_Y2018M03D27.csv', 'IHME_USA
_COUNTY_INFECT_DIS_MORT_1980_2014_ARKANSAS_Y2018M03D27.csv',
        'IHME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_CALIFORNIA_Y2018M03D27.csv', 'IHME_
USA_COUNTY_INFECT_DIS_MORT_1980_2014_COLORADO_Y2018M03D27.csv',
        'IHME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_CONNECTICUT_Y2018M03D27.csv', 'IHME
_USA_COUNTY_INFECT_DIS_MORT_1980_2014_DELAWARE_Y2018M03D27.csv',
        'IHME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_DISTRICT_OF_COLUMBIA_Y2018M03D27.cs
v', 'IHME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_FLORIDA_Y2018M03D27.csv',
        'IHME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_GEORGIA_Y2018M03D27.csv', 'IHME_USA
_COUNTY_INFECT_DIS_MORT_1980_2014_HAWAII_Y2018M03D27.csv',
        'IHME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_IDAHO_Y2018M03D27.csv', 'IHME_USA_C
OUNTY_INFECT_DIS_MORT_1980_2014_ILLINOIS_Y2018M03D27.csv',
        'IHME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_INDIANA_Y2018M03D27.csv', 'IHME_USA
_COUNTY_INFECT_DIS_MORT_1980_2014_IOWA_Y2018M03D27.csv',
        'IHME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_KANSAS_Y2018M03D27.csv','IHME_USA_C
OUNTY_INFECT_DIS_MORT_1980_2014_KENTUCKY_Y2018M03D27.csv',
        'IHME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_LOUISIANA_Y2018M03D27.csv', 'IHME_U
SA_COUNTY_INFECT_DIS_MORT_1980_2014_MAINE_Y2018M03D27.csv',
         'IHME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_MARYLAND_Y2018M03D27.csv', 'IHME_U
SA_COUNTY_INFECT_DIS_MORT_1980_2014_MASSACHUSETTS_Y2018M03D27.csv',
         'IHME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_MICHIGAN_Y2018M03D27.csv', 'IHME_U
SA_COUNTY_INFECT_DIS_MORT_1980_2014_MINNESOTA_Y2018M03D27.csv',
         'IHME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_MISSISSIPPI_Y2018M03D27.csv', 'IHM
E_USA_COUNTY_INFECT_DIS_MORT_1980_2014_MISSOURI_Y2018M03D27.csv',
         'IHME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_MONTANA_Y2018M03D27.csv', 'IHME_US
A_COUNTY_INFECT_DIS_MORT_1980_2014_NEBRASKA_Y2018M03D27.csv',
         'IHME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_NEVADA_Y2018M03D27.csv', 'IHME_USA
_COUNTY_INFECT_DIS_MORT_1980_2014_NEW_HAMPSHIRE_Y2018M03D27.csv',
         'IHME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_NEW_JERSEY_Y2018M03D27.csv', 'IHME
_USA_COUNTY_INFECT_DIS_MORT_1980_2014_NEW_MEXICO_Y2018M03D27.csv',
         'IHME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_NEW_YORK_Y2018M03D27.csv', 'IHME_U
SA_COUNTY_INFECT_DIS_MORT_1980_2014_NORTH_CAROLINA_Y2018M03D27.csv',
         'IHME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_NORTH_DAKOTA_Y2018M03D27.csv', 'IH
ME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_OHIO_Y2018M03D27.csv',
         'IHME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_OKLAHOMA_Y2018M03D27.csv', 'IHME_U
SA_COUNTY_INFECT_DIS_MORT_1980_2014_OREGON_Y2018M03D27.csv',
         'IHME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_PENNSYLVANIA_Y2018M03D27.csv', 'IH
ME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_RHODE_ISLAND_Y2018M03D27.csv',
         'IHME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_SOUTH_CAROLINA_Y2018M03D27.csv',
'IHME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_SOUTH_DAKOTA_Y2018M03D27.csv',
        'IHME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_TENNESSEE_Y2018M03D27.csv', 'IHME_U
SA_COUNTY_INFECT_DIS_MORT_1980_2014_TEXAS_Y2018M03D27.csv',
        'IHME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_UTAH_Y2018M03D27.csv', 'IHME_USA_CO
UNTY_INFECT_DIS_MORT_1980_2014_VERMONT_Y2018M03D27.csv',
        'IHME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_VIRGINIA_Y2018M03D27.csv', 'IHME_US
A_COUNTY_INFECT_DIS_MORT_1980_2014_WASHINGTON_Y2018M03D27.csv',
        'IHME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_WEST_VIRGINIA_Y2018M03D27.csv', 'IH
ME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_WISCONSIN_Y2018M03D27.csv', 'IHME_USA_COUNTY_I
NFECT_DIS_MORT_1980_2014_WYOMING_Y2018M03D27.csv']
```

In [7]:
```
df = pd.concat([pd.read_csv(f) for f in files])
```

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

Out[8]:

	measure_id	measure_name	location_id	location_name	FIPS	cause_id	cause_name	sex_id	sex	age_id	age_na
0	1	Deaths	523	Alabama	1	297	Tuberculosis	1	Male	27	standard
1	1	Deaths	523	Alabama	1	297	Tuberculosis	1	Male	27	standard
2	1	Deaths	523	Alabama	1	297	Tuberculosis	1	Male	27	standard
3	1	Deaths	523	Alabama	1	297	Tuberculosis	1	Male	27	standard
4	1	Deaths	523	Alabama	1	297	Tuberculosis	1	Male	27	standard

The state level FIPS code is from 1 to 56 and county level FIPS code is greater than 56. So, using that logic to split the state and county level data.

For State

```
In [9]: d_state = df.loc[df['FIPS']<=56]
```

We are just considering mortality rate data for state level only for 'year 2014' and for 'both' sex. So, going to split that data.

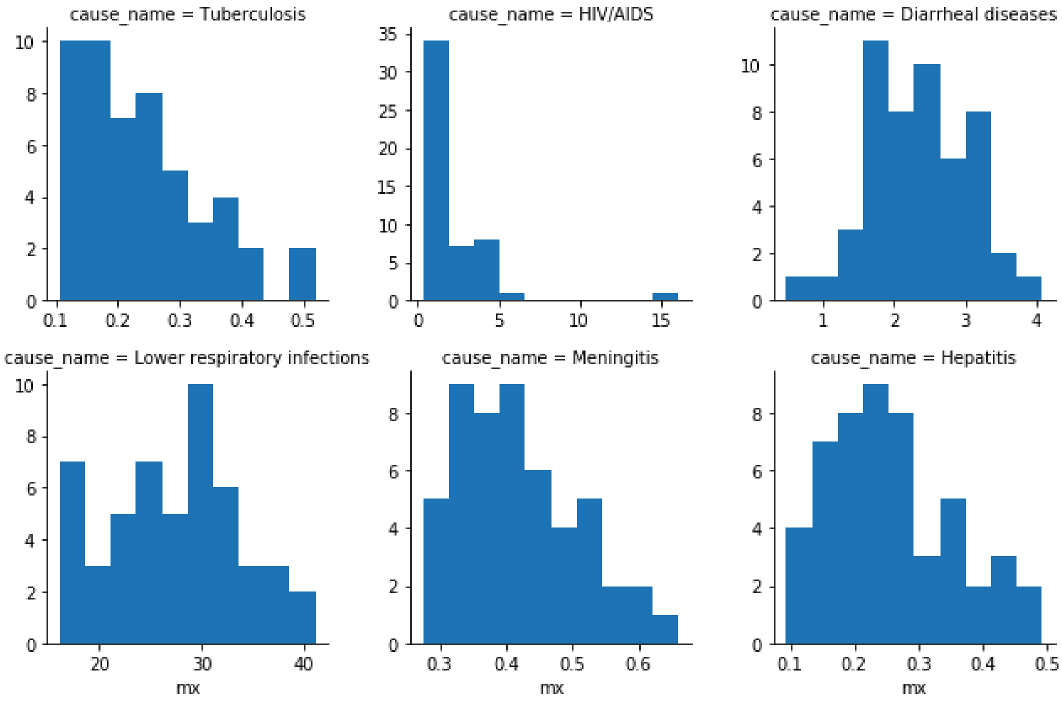
```
In [10]: d_state_both = d_state[d_state.sex_id == 3]
         d_state_2014 = d_state_both.loc[d_state_both['year_id']==2014]
         d_state_2014.head(10)
```

Out[10]:

	measure_id	measure_name	location_id	location_name	FIPS	cause_id	cause_name	sex_id	sex	age_id	ag
104	1	Deaths	523	Alabama	1	297	Tuberculosis	3	Both	27	stan
7244	1	Deaths	523	Alabama	1	298	HIV/AIDS	3	Both	27	stan
14384	1	Deaths	523	Alabama	1	302	Diarrheal diseases	3	Both	27	stan
21524	1	Deaths	523	Alabama	1	322	Lower respiratory infections	3	Both	27	stan
28664	1	Deaths	523	Alabama	1	332	Meningitis	3	Both	27	stan
35804	1	Deaths	523	Alabama	1	400	Hepatitis	3	Both	27	stan
104	1	Deaths	524	Alaska	2	297	Tuberculosis	3	Both	27	stan
3254	1	Deaths	524	Alaska	2	298	HIV/AIDS	3	Both	27	stan
6404	1	Deaths	524	Alaska	2	302	Diarrheal diseases	3	Both	27	stan
9554	1	Deaths	524	Alaska	2	322	Lower respiratory infections	3	Both	27	stan

Histogram of the mortality rate on year 2014 state level (using seaborn Facetgrid)

In [11]:
```
g = sns.FacetGrid(d_state_2014, col="cause_name", col_wrap = 3, sharex=False, sharey=False)
g = g.map(plt.hist, "mx")
```



Pivoting this dataframe for some useful later use.

In [12]:
```
d_stateName_14 = d_state_2014.pivot(index = 'location_name',columns ='cause_name',values='mx')
```

In [13]:
```
d_state_14 = d_state_2014.pivot(index = 'FIPS',columns ='cause_name',values='mx')
d_state_14.head(6)
```

Out[13]:

cause_name	Diarrheal diseases	HIV/AIDS	Hepatitis	Lower respiratory infections	Meningitis	Tuberculosis
FIPS						
1	2.406220	2.585321	0.239779	37.393408	0.582372	0.355667
2	1.343148	0.749104	0.285058	17.639234	0.431981	0.518633
4	2.551890	1.564535	0.335842	18.021531	0.434521	0.258537
5	2.016912	1.786682	0.297565	38.644030	0.520116	0.317000
6	2.205181	2.142319	0.492770	20.695523	0.310342	0.327106
8	1.960395	0.998745	0.279279	23.021514	0.399500	0.192573

Numerical description (mortality rate for diseases on year 2014 by state):

In [14]:
```
d_state_14.describe()
```

Out[14]:

cause_name	Diarrheal diseases	HIV/AIDS	Hepatitis	Lower respiratory infections	Meningitis	Tuberculosis
count	51.000000	51.000000	51.000000	51.000000	51.000000	51.000000
mean	2.356091	2.044772	0.253810	27.163165	0.415890	0.237026
std	0.714691	2.417237	0.101243	6.272960	0.091943	0.101219
min	0.485747	0.338506	0.094369	16.101937	0.274685	0.105845
25%	1.805994	0.747444	0.175431	23.018824	0.347398	0.162280
50%	2.307500	1.262752	0.231361	28.099666	0.402716	0.203494
75%	2.850635	2.480456	0.316040	31.285950	0.469664	0.291722
max	4.083346	16.105768	0.492770	41.107162	0.660877	0.518633

For County

In [15]:
```
d_county = df.loc[df['FIPS']>56]
```

We are just considering mortality rate data for county level only for 'year 2014' and for 'both' sex. So, going to split that data.

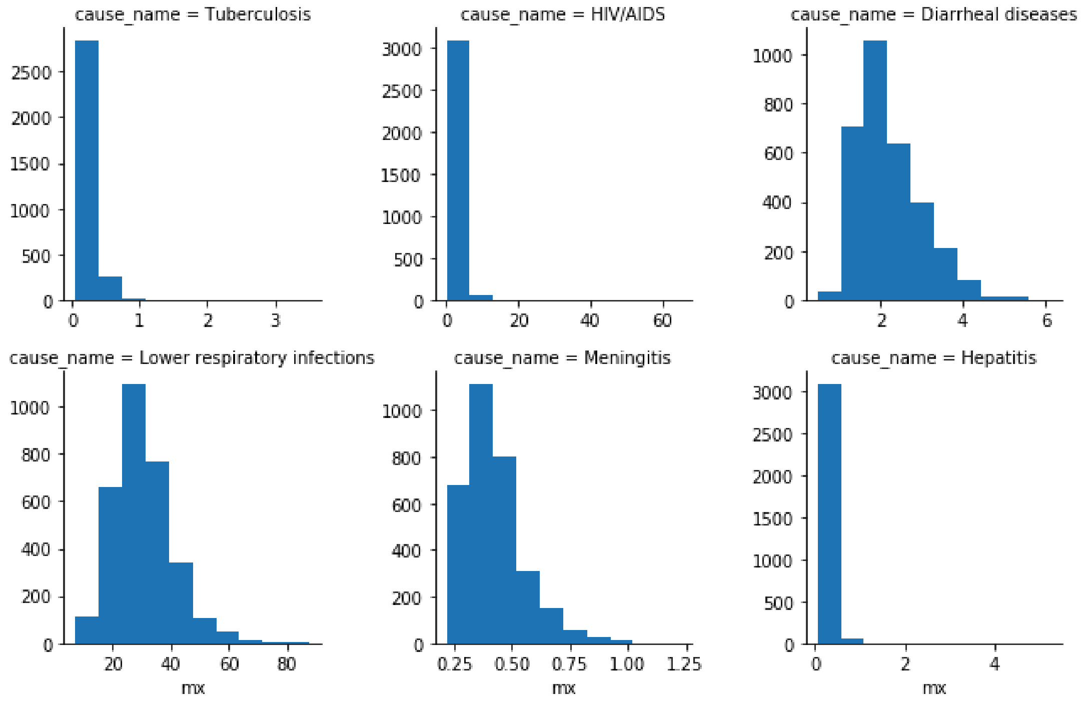
In [16]:
```
d_county_2014 = d_county.loc[d_county['year_id']==2014]
d_county_2014 = d_county_2014[d_county_2014.sex_id == 3]
d_county_2014.tail(6)
```

Out[16]:

	measure_id	measure_name	location_id	location_name	FIPS	cause_id	cause_name	sex_id	sex	age_id	a
14594	1	Deaths	3706	Sublette County	56035	400	Hepatitis	3	Both	27	sta
14699	1	Deaths	3712	Sweetwater County	56037	400	Hepatitis	3	Both	27	sta
14804	1	Deaths	3697	Teton County	56039	400	Hepatitis	3	Both	27	sta
14909	1	Deaths	3714	Uinta County	56041	400	Hepatitis	3	Both	27	sta
15014	1	Deaths	3700	Washakie County	56043	400	Hepatitis	3	Both	27	sta
15119	1	Deaths	3699	Weston County	56045	400	Hepatitis	3	Both	27	sta

Histogram of the mortality rate on year 2014 county level (using seaborn Facetgrid)

In [17]:
```
gc = sns.FacetGrid(d_county_2014, col="cause_name", col_wrap = 3, sharex=False, sharey
= False)
gc = gc.map(plt.hist, "mx")
```



Pivoting this dataframe:

In [18]:
```
d_county_14 = d_county_2014.pivot(index = 'FIPS',columns ='cause_name',values='mx')
d_county_14.head(6)
```

Out[18]:

cause_name	Diarrheal diseases	HIV/AIDS	Hepatitis	Lower respiratory infections	Meningitis	Tuberculosis
FIPS						
1001	1.889825	1.995046	0.197764	34.153983	0.514925	0.309921
1003	1.436522	1.814896	0.210261	21.820851	0.414002	0.197299
1005	2.015340	4.030395	0.201652	33.954553	0.635688	0.416175
1007	2.707614	1.244898	0.200895	41.295744	0.612928	0.261805
1009	3.559936	1.053874	0.207240	54.023856	0.464294	0.258540
1011	2.380902	3.868121	0.221083	30.737632	0.878686	0.580487

Numerical description (mortality rate for diseases on year 2014 by state):

In [19]:

```
d_county_14.describe()
```

Out[19]:

cause_name	Diarrheal diseases	HIV/AIDS	Hepatitis	Lower respiratory infections	Meningitis	Tuberculosis
count	3142.000000	3142.000000	3142.000000	3142.000000	3142.000000	3142.000000
mean	2.210179	1.333057	0.220118	30.337976	0.421084	0.217900
std	0.781120	1.961501	0.155413	10.218042	0.126029	0.174057
min	0.470721	0.154699	0.056586	7.242285	0.217513	0.057117
25%	1.629510	0.447175	0.134477	23.443510	0.330283	0.115154
50%	2.031328	0.731724	0.189628	28.972474	0.399273	0.169081
75%	2.663454	1.437820	0.270877	35.884786	0.477695	0.265586
max	6.139848	64.871216	5.244210	87.696763	1.226104	3.507462

Loading GDP and ACS Data

GDP data

Loading file and formatting gdp data:

In [20]:

```
gdp = pd.read_excel('GCP_Release_1.xlsx', header=3)
gdp.head()
```

Out[20]:

	Unnamed: 0	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	2012	2013	2014	2015
0	01001	Autauga	AL	1.0	All Industries	1383941	1363368	1402516	1539406
1	01001	Autauga	AL	2.0	Private goods-producing industries	286396	310468	323582	346355
2	01001	Autauga	AL	3.0	Private services-providing industries	948490	904599	928438	1037309
3	01001	Autauga	AL	4.0	Government and government enterprises	149055	148301	150496	155742
4	01003	Baldwin	AL	1.0	All Industries	5599194	6365080	6547396	6436107

Just taking data of 'All Industries':

In [21]:

```
gdp.rename(columns={
    'Unnamed: 0': 'FIPS',
    'Unnamed: 1': 'County',
    'Unnamed: 2': 'State',
    'Unnamed: 3': 'LineCode'
}, inplace=True)
gdp = gdp[gdp['LineCode'] == 1.0]
gdp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3113 entries, 0 to 12448
Data columns (total 9 columns):
FIPS           3113 non-null object
County         3113 non-null object
State          3113 non-null object
LineCode       3113 non-null float64
Unnamed: 4     3113 non-null object
2012           3113 non-null object
2013           3113 non-null object
2014           3113 non-null object
2015           3113 non-null object
dtypes: float64(1), object(8)
memory usage: 243.2+ KB
```

In [22]:

```
gdp['FIPS'] = gdp['FIPS'].astype('i8')
```

In [23]:

```
gdp.head()
```

Out[23]:

	FIPS	County	State	LineCode	Unnamed: 4	2012	2013	2014	2015
0	1001	Autauga	AL	1.0	All Industries	1383941	1363368	1402516	1539406
4	1003	Baldwin	AL	1.0	All Industries	5599194	6365080	6547396	6436107
8	1005	Barbour	AL	1.0	All Industries	639833	701750	689212	743779
12	1007	Bibb	AL	1.0	All Industries	297560	325906	329087	322307
16	1009	Blount	AL	1.0	All Industries	632761	701145	688525	819608

For State level GDP

Formating the FIPS code to get the state FIPS code and doing 'groupby' by state FIPS code to get state level GDP.

In [24]:

```
state_gdp = gdp.assign(StateFIPS = gdp['FIPS'] // 1000).groupby('StateFIPS')[[2014]].s
um()
state_gdp.head()
```

Out[24]:

	2014
StateFIPS	
1	194059720
2	55547370
4	284573788
5	117339124
6	2396552112

Numerical description of 2014 State GDP:

In [25]:
```
state_gdp.describe()
```

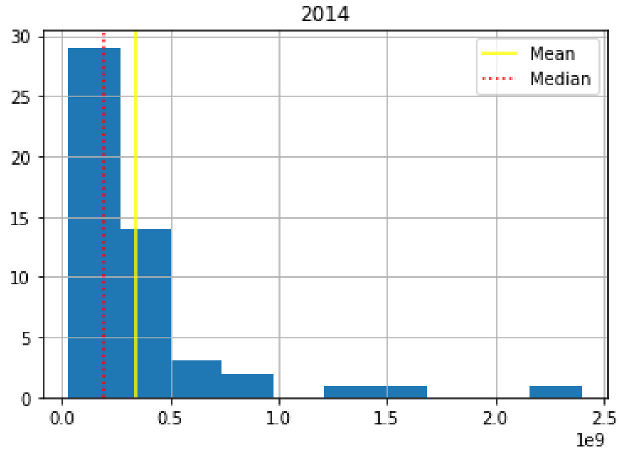
Out[25]:

	2014
count	5.100000e+01
mean	3.413308e+08
std	4.305805e+08
min	2.971440e+07
25%	8.523418e+07
50%	1.940597e+08
75%	4.457034e+08
max	2.396552e+09

Histogram of 2014 State GDP:

In [26]:
```
state_gdp.hist()
plt.axvline(state_gdp[2014].mean(), color='yellow', linestyle = '-', label='Mean')
plt.axvline(state_gdp[2014].median(), color='red', linestyle = ':', label='Median')

plt.legend()
```

Out[26]:
```
<matplotlib.legend.Legend at 0x22a8339fe48>
```



For County Level GDP

In [27]:
```
county_gdp =  gdp[['FIPS', 'County','State', 2014]]
county_gdp.head()
```

Out[27]:

	FIPS	County	State	2014
0	1001	Autauga	AL	1402516
4	1003	Baldwin	AL	6547396
8	1005	Barbour	AL	689212
12	1007	Bibb	AL	329087
16	1009	Blount	AL	688525

Numerical description of 2014 county GDP:

In [28]:

```
#county_gdp.describe()
county_gdp[2014].mean()
```

Out[28]:

```
5591992.558946354
```

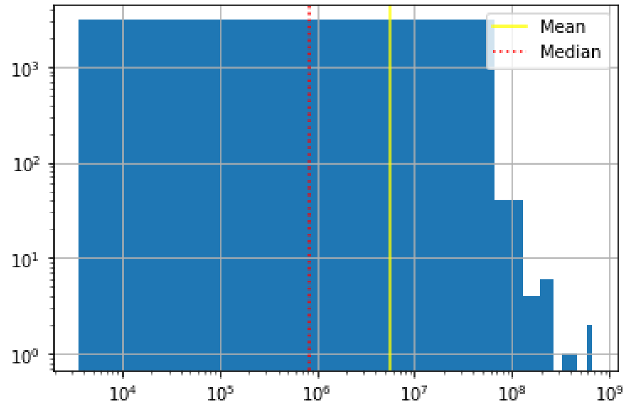
Histogram of 2014 county GDP:

In [29]:

```
county_gdp[2014].hist()
plt.axvline(county_gdp[2014].mean(), color='yellow', linestyle = '-', label='Mean')
plt.axvline(county_gdp[2014].median(), color='red', linestyle = ':', label='Median')
plt.xscale('log')
plt.yscale('log')
plt.legend()
```

Out[29]:

```
<matplotlib.legend.Legend at 0x22a81e91198>
```



ACS data _ For State Level

B00001_001E:an estimate of the total population of a region

C27017_001E:total

C27017_002E:population under the poverty line

B18135_020E : Health Insurance coverage for age 19-64 with no disability

Loading ACS data for state level:

In [30]:

```
st_pop = c.acs.state(('NAME', 'B00001_001E', 'C27017_001E', 'C27017_002E','B18135_020
E'), '*', year=2014)
st_pop = pd.DataFrame.from_records(st_pop)
st_pop.head()
```

Out[30]:

	B00001_001E	B18135_020E	C27017_001E	C27017_002E	NAME	state
0	399744	2005980.0	4687615.0	1294017.0	Alabama	01
1	95089	309645.0	699593.0	114151.0	Alaska	02
2	454290	2700875.0	6399873.0	1711028.0	Arizona	04
3	242391	1133820.0	2857966.0	815516.0	Arkansas	05
4	2749907	16737950.0	37243757.0	9127115.0	California	06

In [31]:

```
st_pop.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 52 entries, 0 to 51
Data columns (total 6 columns):
B00001_001E    52 non-null object
B18135_020E    52 non-null float64
C27017_001E    52 non-null float64
C27017_002E    52 non-null float64
NAME           52 non-null object
state          52 non-null object
dtypes: float64(3), object(3)
memory usage: 2.5+ KB
```

In [32]:

```
st_pop.rename(columns={
    'B00001_001E': 'Est.Total',
    'C27017_001E': 'Total_pop',
    'C27017_002E': 'Under_poverty',
    'B18135_020E': 'Health_Insurance',
    'state': 'FIPS'
}, inplace=True)
st_pop['FIPS'] = st_pop['FIPS'].astype('i8')
st_pop.head()
```

Out[32]:

	Est.Total	Health_Insurance	Total_pop	Under_poverty	NAME	FIPS
0	399744	2005980.0	4687615.0	1294017.0	Alabama	1
1	95089	309645.0	699593.0	114151.0	Alaska	2
2	454290	2700875.0	6399873.0	1711028.0	Arizona	4
3	242391	1133820.0	2857966.0	815516.0	Arkansas	5
4	2749907	16737950.0	37243757.0	9127115.0	California	6

Getting the Fraction of the population below the poverty line and Fraction of the population with health insurance coverage.

In [33]:

```
st_pop['Fraction'] = st_pop['Under_poverty']/st_pop['Total_pop']
st_pop['Health_ins_frac'] = st_pop['Health_Insurance']/st_pop['Total_pop']
```

In [34]:

```
st_pop.head()
```

Out[34]:

	Est.Total	Health_Insurance	Total_pop	Under_poverty	NAME	FIPS	Fraction	Health_ins_frac
0	399744	2005980.0	4687615.0	1294017.0	Alabama	1	0.276050	0.427932
1	95089	309645.0	699593.0	114151.0	Alaska	2	0.163168	0.442607
2	454290	2700875.0	6399873.0	1711028.0	Arizona	4	0.267353	0.422020
3	242391	1133820.0	2857966.0	815516.0	Arkansas	5	0.285348	0.396723
4	2749907	16737950.0	37243757.0	9127115.0	California	6	0.245064	0.449416

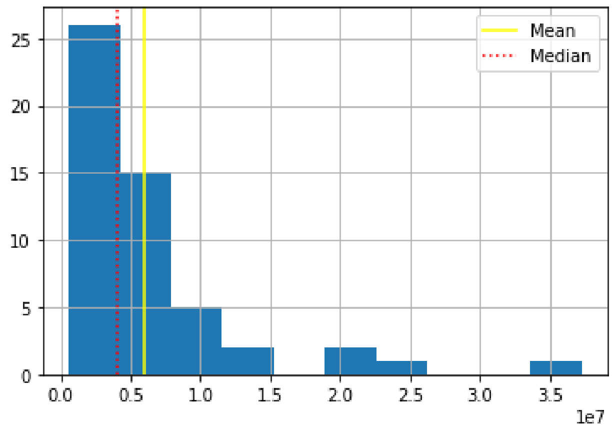
For total population Numerical description and distribution

In [35]:
```
st_pop['Total_pop'].describe()
```

Out[35]:
```
count    5.200000e+01
mean     5.944674e+06
std      6.757276e+06
min      5.592060e+05
25%      1.739724e+06
50%      4.029804e+06
75%      6.497770e+06
max      3.724376e+07
Name: Total_pop, dtype: float64
```

In [36]:
```
st_pop['Total_pop'].hist()
plt.axvline(st_pop['Total_pop'].mean(), color='yellow', linestyle = '-', label='Mean')
plt.axvline(st_pop['Total_pop'].median(), color='red', linestyle = ':', label='Median')

plt.legend()
```

Out[36]:
```
<matplotlib.legend.Legend at 0x22a8350b438>
```



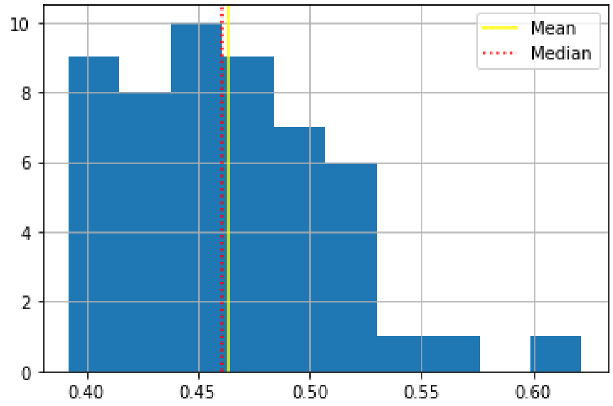
For Fraction of people under poverty line Numerical description and distribution

In [37]:
```
st_pop['Health_ins_frac'].describe()
```

Out[37]:
```
count    52.000000
mean      0.463297
std       0.048395
min       0.391975
25%       0.426847
50%       0.460313
75%       0.495650
max       0.621725
Name: Health_ins_frac, dtype: float64
```

In [38]:
```
st_pop['Health_ins_frac'].hist()
plt.axvline(st_pop['Health_ins_frac'].mean(), color='yellow', linestyle = '-', label
='Mean')
plt.axvline(st_pop['Health_ins_frac'].median(), color='red', linestyle = ':', label='M
edian')

plt.legend()
```

Out[38]:
```
<matplotlib.legend.Legend at 0x22a80e7b208>
```



ACS data _ For County Level

Loading ACS data for county level:

In [39]:
```
cnt_pop = c.acs.state_county(('NAME', 'C27017_001E', 'C27017_002E','B18135_020E'),
'*','*', year=2014)
cnt_pop = pd.DataFrame.from_records(cnt_pop)
cnt_pop.sort_values('state').head()
```

Out[39]:

	B18135_020E	C27017_001E	C27017_002E	NAME	county	state
431	48501.0	117991.0	28535.0	Morgan County, Alabama	103	01
425	7473.0	20227.0	7387.0	Marengo County, Alabama	091	01
424	160352.0	334719.0	68208.0	Madison County, Alabama	089	01
423	8781.0	18338.0	6820.0	Macon County, Alabama	087	01
422	3993.0	10794.0	4276.0	Lowndes County, Alabama	085	01

In [40]:
```
cnt_pop.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3220 entries, 0 to 3219
Data columns (total 6 columns):
B18135_020E    3220 non-null float64
C27017_001E    3220 non-null float64
C27017_002E    3220 non-null float64
NAME           3220 non-null object
county         3220 non-null object
state          3220 non-null object
dtypes: float64(3), object(3)
memory usage: 151.0+ KB
```

In [41]:
```
cnt_pop.rename(columns={
    'B00001_001E': 'Est.Total',
    'C27017_001E': 'Total_pop',
    'C27017_002E': 'Under_poverty',
    'B18135_020E': 'Health_Insurance',
    'county': 'FIPS',
    'state': 'st_FIPS'
}, inplace=True)
cnt_pop['FIPS'] = cnt_pop['FIPS'].astype('i4')
cnt_pop['st_FIPS'] = cnt_pop['st_FIPS'].astype('i4')
cnt_pop.head()
```

Out[41]:

	Health_Insurance	Total_pop	Under_poverty	NAME	FIPS	st_FIPS
0	4228988.0	9815811.0	2724570.0	Los Angeles County, California	37	6
1	52338.0	143740.0	50284.0	Madera County, California	39	6
2	124933.0	249456.0	31489.0	Marin County, California	41	6
3	7646.0	17771.0	3945.0	Mariposa County, California	43	6
4	33668.0	86021.0	26668.0	Mendocino County, California	45	6

Getting the Fraction of the population below the poverty line and Fraction of the population with health insurance coverage.

In [42]:
```
cnt_pop['Fraction'] = cnt_pop['Under_poverty']/cnt_pop['Total_pop']
cnt_pop['Health_Ins_Frac'] = cnt_pop['Health_Insurance']/cnt_pop['Total_pop']
cnt_pop.head()
```

Out[42]:

	Health_Insurance	Total_pop	Under_poverty	NAME	FIPS	st_FIPS	Fraction	Health_Ins_Frac
0	4228988.0	9815811.0	2724570.0	Los Angeles County, California	37	6	0.277570	0.430834
1	52338.0	143740.0	50284.0	Madera County, California	39	6	0.349826	0.364116
2	124933.0	249456.0	31489.0	Marin County, California	41	6	0.126231	0.500822
3	7646.0	17771.0	3945.0	Mariposa County, California	43	6	0.221991	0.430252
4	33668.0	86021.0	26668.0	Mendocino County, California	45	6	0.310017	0.391393

For total population Numerical description and distribution

In [43]:
```
cnt_pop['Total_pop'].describe()
```

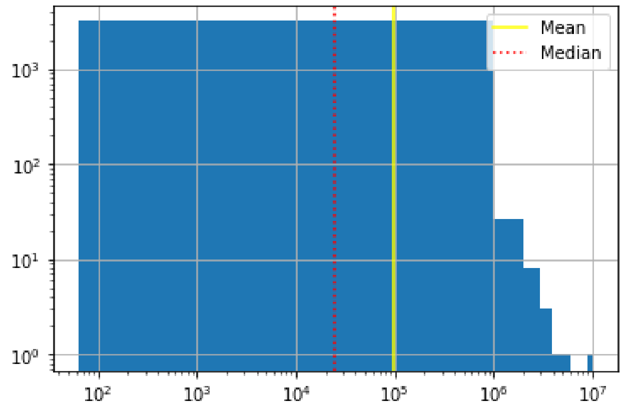
Out[43]:
```
count    3.220000e+03
mean     9.600094e+04
std      3.105923e+05
min      6.400000e+01
25%      1.077800e+04
50%      2.501750e+04
75%      6.391875e+04
max      9.815811e+06
Name: Total_pop, dtype: float64
```

In [44]:

```
cnt_pop['Total_pop'].hist()
plt.axvline(cnt_pop['Total_pop'].mean(), color='yellow', linestyle = '-', label='Mean')
plt.axvline(cnt_pop['Total_pop'].median(), color='red', linestyle = ':', label='Median')
plt.xscale('log')
plt.yscale('log')

plt.legend()
```

Out[44]:

```
<matplotlib.legend.Legend at 0x22a83477278>
```



For Fraction of people under poverty line Numerical description and distribution

In [45]:

```
cnt_pop['Health_Ins_Frac'].describe()
```

Out[45]:

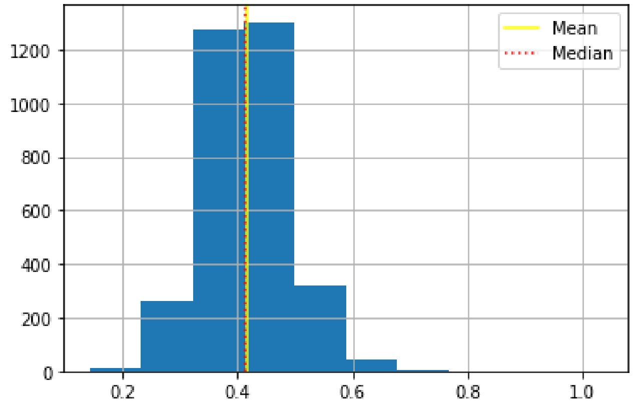
```
count    3220.000000
mean        0.415809
std         0.073479
min         0.142857
25%         0.365642
50%         0.414221
75%         0.461241
max         1.034578
Name: Health_Ins_Frac, dtype: float64
```

In [46]:

```
cnt_pop['Health_Ins_Frac'].hist()
plt.axvline(cnt_pop['Health_Ins_Frac'].mean(), color='yellow', linestyle = '-', label
='Mean')
plt.axvline(cnt_pop['Health_Ins_Frac'].median(), color='red', linestyle = ':', label
='Median')

plt.legend()
```

Out[46]:

```
<matplotlib.legend.Legend at 0x22a83013a90>
```



Spatial Visualization

In this part, we are going to display the state- and county-level Tuberculosis and Lower Respiratory mortality rates on a maps, using Plotly's Choropleth feature.

At state level (for Tuberculosis & Lower Respiratory Infections mortality rates)

Taking the dataframe and assigning there the State postal code.

In [47]:

```
d_stateName_14.head(6)
```

Out[47]:

cause_name	Diarrheal diseases	HIV/AIDS	Hepatitis	Lower respiratory infections	Meningitis	Tuberculosis
location_name						
Alabama	2.406220	2.585321	0.239779	37.393408	0.582372	0.355667
Alaska	1.343148	0.749104	0.285058	17.639234	0.431981	0.518633
Arizona	2.551890	1.564535	0.335842	18.021531	0.434521	0.258537
Arkansas	2.016912	1.786682	0.297565	38.644030	0.520116	0.317000
California	2.205181	2.142319	0.492770	20.695523	0.310342	0.327106
Colorado	1.960395	0.998745	0.279279	23.021514	0.399500	0.192573

In [48]:
```
d_stateName_14 = d_stateName_14.assign(code = ['AL', 'AK', 'AZ','AR','CA','CO','CT','D
E','DC',
                                                          'FL','GA','HI','ID','IL','I
N',
                                                          'IA','KS','KY','LA','ME','M
D','MA','MI','MN','MS','MO','MT',
                                                         'NE','NV','NH','NJ','NM','N
Y','NC','ND','OH',
                                                          'OK','OR','PA','RI','SC','S
D','TN',
                                                         'TX','UT','VT','VA','WA','W
V','WI','WY'])
```

In [49]:
```
new_stName = d_stateName_14.reset_index()
```

In [50]:
```
sp2_st_14 = new_stName[['location_name','code','Tuberculosis','Lower respiratory infec
tions']]
```

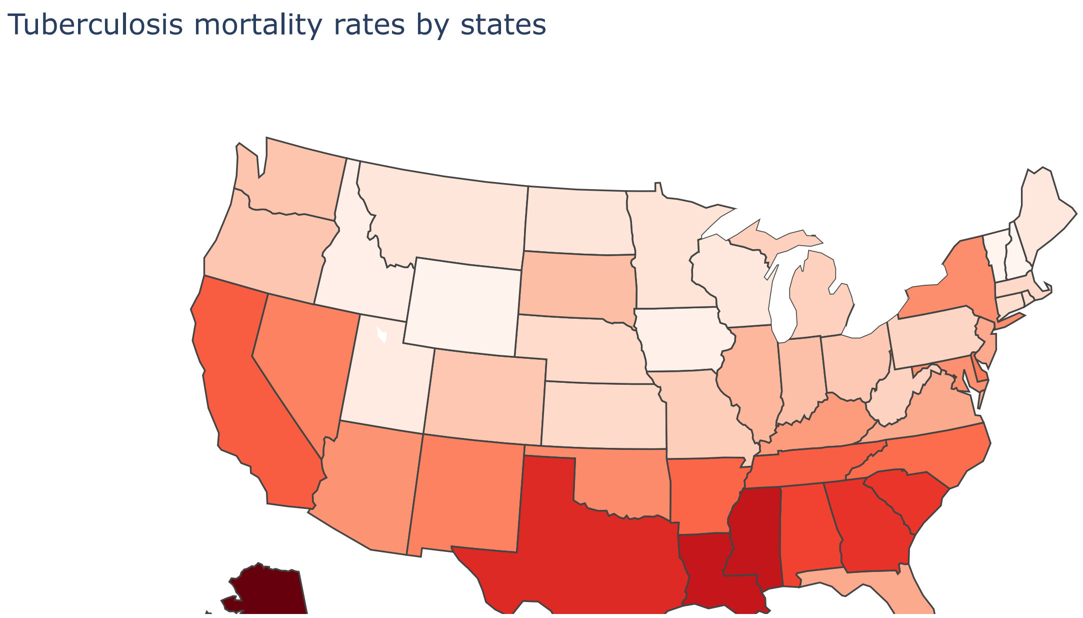
In [51]:
```
sp2_st_14.head()
```

Out[51]:

cause_name	location_name	code	Tuberculosis	Lower respiratory infections
0	Alabama	AL	0.355667	37.393408
1	Alaska	AK	0.518633	17.639234
2	Arizona	AZ	0.258537	18.021531
3	Arkansas	AR	0.317000	38.644030
4	California	CA	0.327106	20.695523

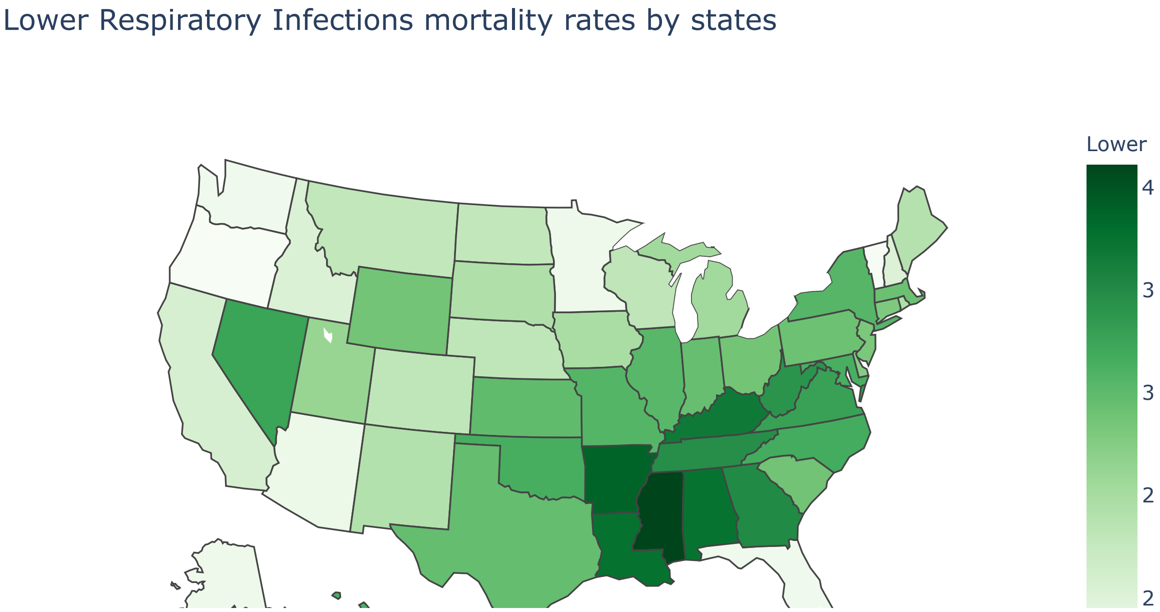
The state-level Tuberculosis mortality rates on a map

```
In [52]: fig = go.Figure(data=go.Choropleth(
             locations=sp2_st_14['code'], # Spatial coordinates
             z = sp2_st_14['Tuberculosis'].astype(float), # Data to be color-coded
             locationmode = 'USA-states', # set of locations match entries in `locations`
             colorscale = 'Reds',
             colorbar_title = "Tuberculosis mortality rates",
         ))

         fig.update_layout(
             title_text = 'Tuberculosis mortality rates by states',
             geo_scope='usa', # limite map scope to USA
         )

         fig.show()
```



The state-level Lower Respiratory Infections mortality rates on a map

In [53]:
```
fig = go.Figure(data=go.Choropleth(
    locations=sp2_st_14['code'], # Spatial coordinates
    z = sp2_st_14['Lower respiratory infections'].astype(float), # Data to be color-co
ded
    locationmode = 'USA-states', # set of locations match entries in `locations`
    colorscale = 'Greens',
    colorbar_title = "Lower Respiratory Infections mortality rates",
))

fig.update_layout(
    title_text = 'Lower Respiratory Infections mortality rates by states',
    geo_scope='usa', # limite map scope to USA
)

fig.show()
```



At county level (for Tuberculosis & Lower Respiratory Infections mortality rates)

In [54]:
```
d_county_14.head(6)
```

Out[54]:

cause_name	Diarrheal diseases	HIV/AIDS	Hepatitis	Lower respiratory infections	Meningitis	Tuberculosis
FIPS						
1001	1.889825	1.995046	0.197764	34.153983	0.514925	0.309921
1003	1.436522	1.814896	0.210261	21.820851	0.414002	0.197299
1005	2.015340	4.030395	0.201652	33.954553	0.635688	0.416175
1007	2.707614	1.244898	0.200895	41.295744	0.612928	0.261805
1009	3.559936	1.053874	0.207240	54.023856	0.464294	0.258540
1011	2.380902	3.868121	0.221083	30.737632	0.878686	0.580487

In [55]:
```
new_cnt = d_county_14.reset_index()
```

In [56]:
```
sp_cnt_14 = new_cnt[['FIPS','Tuberculosis','Lower respiratory infections']]
sp_cnt_14.head()
```

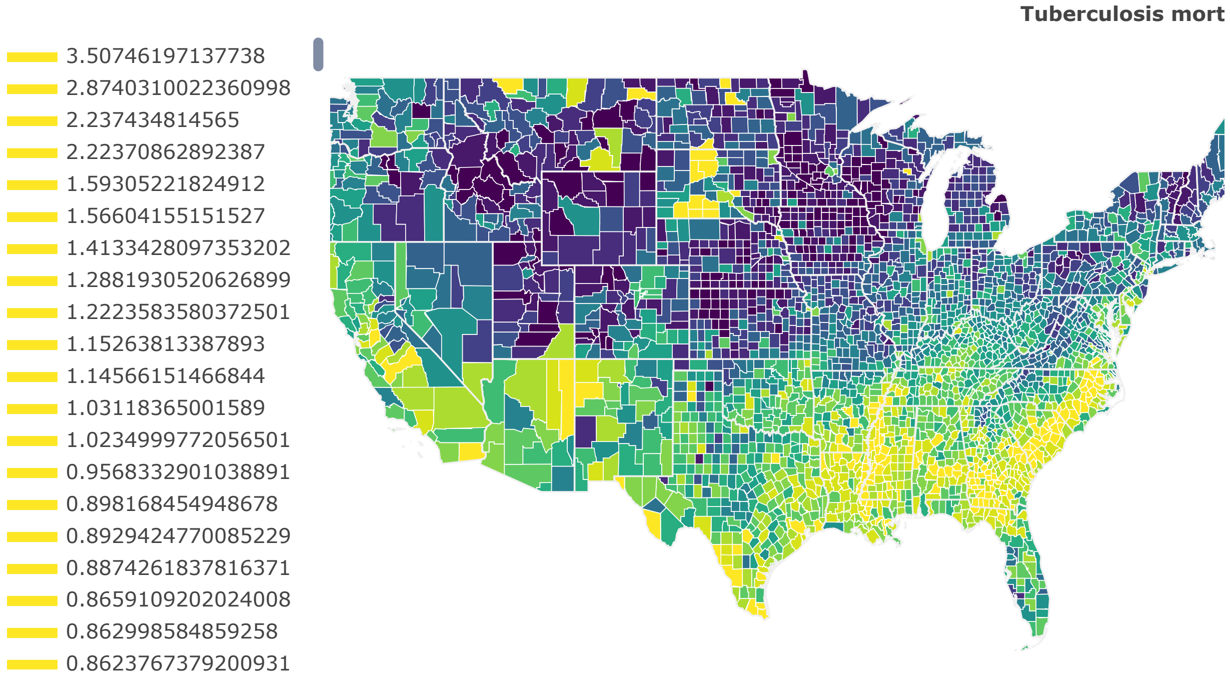
Out[56]:

cause_name	FIPS	Tuberculosis	Lower respiratory infections
0	1001	0.309921	34.153983
1	1003	0.197299	21.820851
2	1005	0.416175	33.954553
3	1007	0.261805	41.295744
4	1009	0.258540	54.023856

The county-level Tuberculosis mortality rates on a map

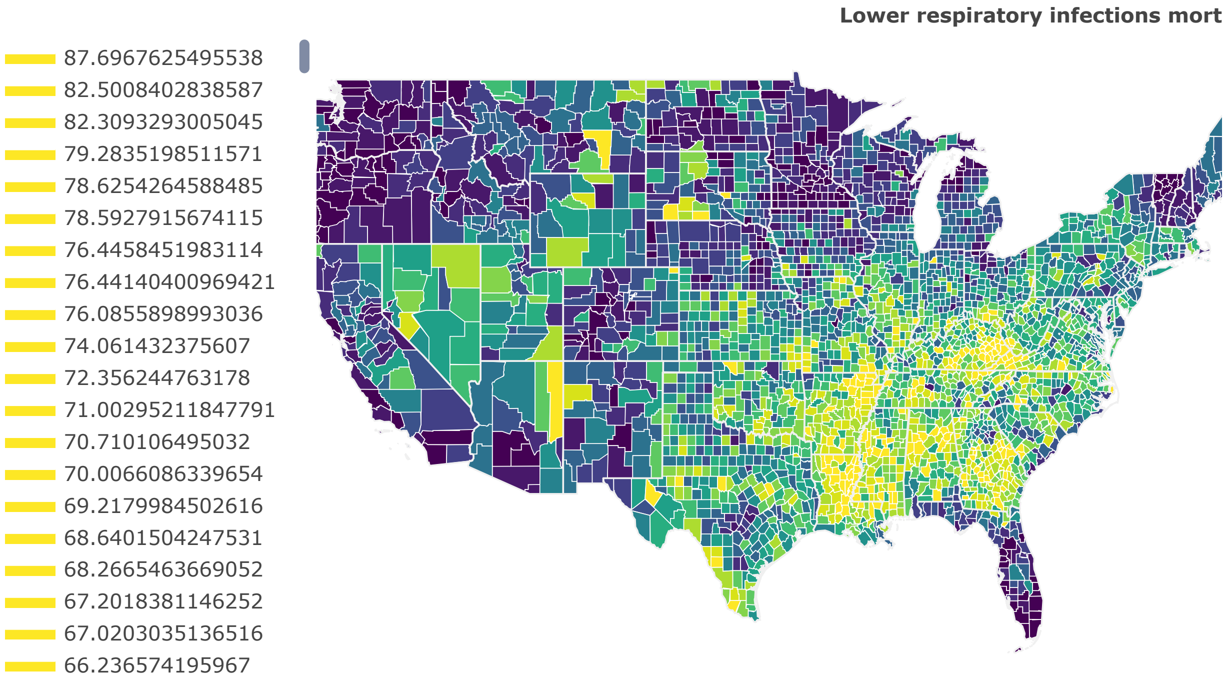
In [57]:
```
## new one
sp_cnt_14['FIPS'] = sp_cnt_14['FIPS'].apply(lambda x: str(x).zfill(3))
values = sp_cnt_14['Tuberculosis'].tolist()
##sp_cnt_14['FIPS']=sp_cnt_14['FIPS'].apply(lambda x: str(x).zfill(3))
fips=sp_cnt_14['FIPS'].tolist()

colorscale = [
   'rgb(68.0, 1.0, 84.0)',
   'rgb(66.0, 64.0, 134.0)',
   'rgb(38.0, 130.0, 142.0)',
   'rgb(63.0, 188.0, 115.0)',
   'rgb(216.0, 226.0, 25.0)'
]

fig = ff.create_choropleth(
   fips=fips, values=values,
 county_outline={'color': 'rgb(255,255,255)', 'width': 0.5},
   legend_title='Tuberculosis mortality rate per county'

)
fig.update_layout(
   legend_x = 0,
   annotations = {'x': -0.12, 'xanchor': 'left'}
)

fig.layout.template = None
fig.show()
```

```
C:\Users\Farhana\Anaconda3\lib\site-packages\pandas\core\frame.py:6692: FutureWarning:

Sorting because non-concatenation axis is not aligned. A future version
of pandas will change to not sort by default.

To accept the future behavior, pass 'sort=False'.

To retain the current behavior and silence the warning, pass 'sort=True'.
```



The county-level Lower Respiratory Infections mortality rates on a map

In [58]:
```
sp_cnt_14['FIPS'] = sp_cnt_14['FIPS'].apply(lambda x: str(x).zfill(3))
values = sp_cnt_14['Lower respiratory infections'].tolist()
##sp_cnt_14['FIPS']=sp_cnt_14['FIPS'].apply(lambda x: str(x).zfill(3))
fips=sp_cnt_14['FIPS'].tolist()

colorscale = [
   'rgb(68.0, 1.0, 84.0)',
   'rgb(66.0, 64.0, 134.0)',
   'rgb(38.0, 130.0, 142.0)',
   'rgb(63.0, 188.0, 115.0)',
   'rgb(216.0, 226.0, 25.0)'
]

fig = ff.create_choropleth(
   fips=fips, values=values,
 county_outline={'color': 'rgb(255,255,255)', 'width': 0.5},
   legend_title='Lower respiratory infections mortality rate per county'

)
fig.update_layout(
   legend_x = 0,
   annotations = {'x': -0.12, 'xanchor': 'left'}
)

fig.layout.template = None
fig.show()
```



Dealing with Time

This part, we need to show the change in state-level mortality rate (latest year - earliest year) for Tuberculosis, Lower Respiratory, and HIV/AIDS on a map.

In [59]:

```
dt_state = d_state_both[['FIPS','location_name','cause_name','cause_id',
                                                 'year_id','sex','mx']]
```

In [60]:

```
dt_state = dt_state[(dt_state['cause_id'] == 297) | (dt_state['cause_id'] == 298) |(dt
_state['cause_id'] == 322)]
```

Taking only 2014 data:

In [61]:

```
dt_state_2014 = dt_state[(dt_state['year_id']==2014)]
dt_state_2014.rename(columns={'mx': 'mx_2014'}, inplace=True)
dt_state_2014= dt_state_2014[['FIPS','location_name', 'cause_name','mx_2014']]
dt_state_2014.head()
```

Out[61]:

	FIPS	location_name	cause_name	mx_2014
104	1	Alabama	Tuberculosis	0.355667
7244	1	Alabama	HIV/AIDS	2.585321
21524	1	Alabama	Lower respiratory infections	37.393408
104	2	Alaska	Tuberculosis	0.518633
3254	2	Alaska	HIV/AIDS	0.749104

Taking only 1980 data:

In [62]:

```
dt_state_1980 = dt_state[(dt_state['year_id']==1980)]
dt_state_1980.rename(columns={'mx': 'mx_1980'}, inplace=True)
dt_state_1980= dt_state_1980[['FIPS','location_name', 'cause_name','mx_1980']]
dt_state_1980.head()
```

Out[62]:

	FIPS	location_name	cause_name	mx_1980
70	1	Alabama	Tuberculosis	1.459400
7210	1	Alabama	HIV/AIDS	0.000000
21490	1	Alabama	Lower respiratory infections	32.602202
70	2	Alaska	Tuberculosis	2.906372
3220	2	Alaska	HIV/AIDS	0.000000

Joining 1980 and 2014 mortality rate data and taking the difference.

In [63]:

```
dt_state_joined = pd.merge(dt_state_1980, dt_state_2014, how = 'outer', on = ['FIP
S','location_name','cause_name'])
```

In [64]:
```
dt_state_joined['mx_diff'] = dt_state_joined['mx_2014']-dt_state_joined['mx_1980']
dt_state_joined= dt_state_joined[['FIPS','location_name', 'cause_name','mx_1980','mx_2
014','mx_diff']]
dt_state_joined.head(6)
```

Out[64]:

	FIPS	location_name	cause_name	mx_1980	mx_2014	mx_diff
0	1	Alabama	Tuberculosis	1.459400	0.355667	-1.103732
1	1	Alabama	HIV/AIDS	0.000000	2.585321	2.585321
2	1	Alabama	Lower respiratory infections	32.602202	37.393408	4.791206
3	2	Alaska	Tuberculosis	2.906372	0.518633	-2.387739
4	2	Alaska	HIV/AIDS	0.000000	0.749104	0.749104
5	2	Alaska	Lower respiratory infections	41.764982	17.639234	-24.125748

Pivoting the data frame and then adding location postal code.

In [65]:
```
dt_state_joined_pivoted = dt_state_joined.pivot(index = 'location_name',
                                                columns ='cause_name',values='mx_dif
f')
```

In [66]:
```
dt_state_joined_pivoted = dt_state_joined_pivoted.reset_index()
dt_state_joined_pivoted.head(6)
```

Out[66]:

cause_name	location_name	HIV/AIDS	Lower respiratory infections	Tuberculosis
0	Alabama	2.585321	4.791206	-1.103732
1	Alaska	0.749104	-24.125748	-2.387739
2	Arizona	1.564535	-19.482137	-1.353445
3	Arkansas	1.786682	3.816518	-0.714531
4	California	2.142319	-19.250765	-1.677673
5	Colorado	0.998745	-18.484465	-1.102421

In [67]:
```
dt_state_joined_pivoted = dt_state_joined_pivoted.assign(code = ['AL', 'AK', 'AZ','A
R','CA','CO','CT','DE','DC',
                                                            'FL','GA','HI','ID','IL','I
N',
                                                            'IA','KS','KY','LA','ME','M
D','MA','MI','MN','MS','MO','MT',
                                                           'NE','NV','NH','NJ','NM','N
Y','NC','ND','OH',
                                                            'OK','OR','PA','RI','SC','S
D','TN',
                                                           'TX','UT','VT','VA','WA','W
V','WI','WY'])
```

In [68]:

```
dt_state_joined_pivoted.head(6)
```

Out[68]:

cause_name	location_name	HIV/AIDS	Lower respiratory infections	Tuberculosis	code
0	Alabama	2.585321	4.791206	-1.103732	AL
1	Alaska	0.749104	-24.125748	-2.387739	AK
2	Arizona	1.564535	-19.482137	-1.353445	AZ
3	Arkansas	1.786682	3.816518	-0.714531	AR
4	California	2.142319	-19.250765	-1.677673	CA
5	Colorado	0.998745	-18.484465	-1.102421	CO

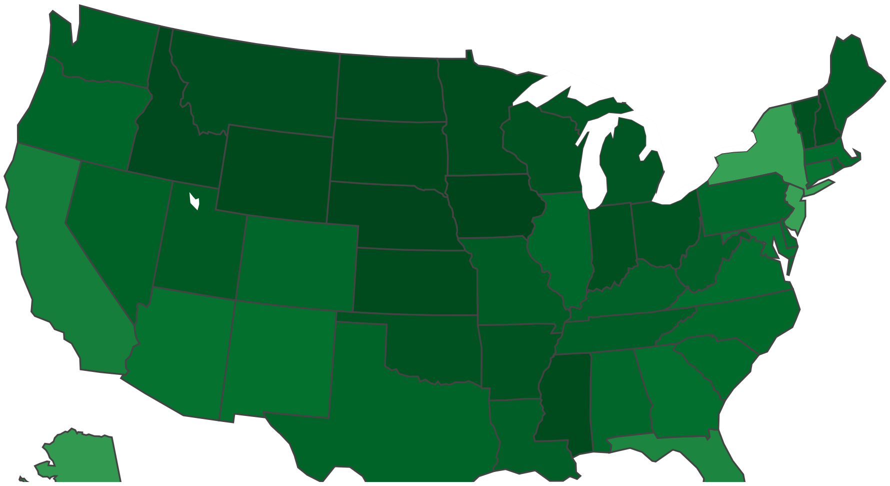
Plotting on a Map

The change in state-level mortality rate (1980 - 2014) for Tuberculosis

In [69]:

```python
fig = go.Figure(data=go.Choropleth(
    locations=dt_state_joined_pivoted['code'], # Spatial coordinates
    z = dt_state_joined_pivoted['Tuberculosis'].astype(float), # Data to be color-coded
    locationmode = 'USA-states', # set of locations match entries in `locations`
    colorscale = 'Greens',
    colorbar_title = "Tuberculosis Rate chane",
))

fig.update_layout(
    title_text = 'Tuberculosis mortality rate change by State over 1980-2014',
    geo_scope='usa', # limite map scope to USA
)

fig.show()
```

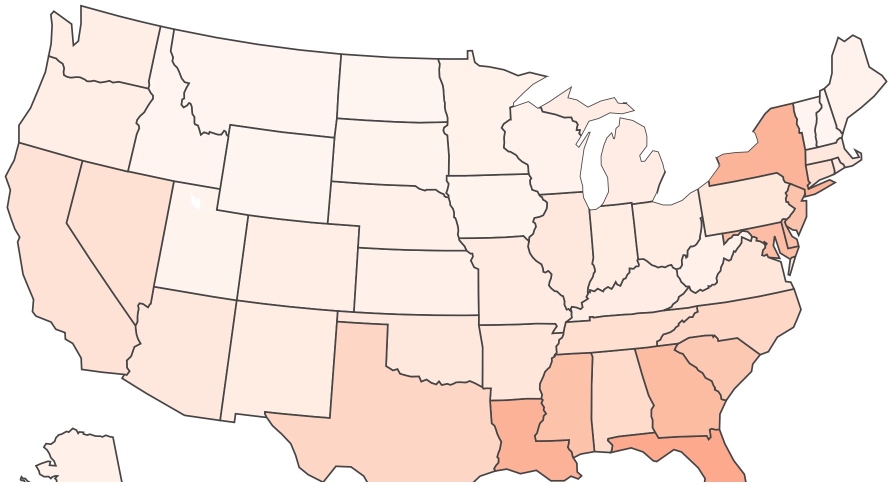
Tuberculosis mortality rate change by State over 1980-2014



The change in state-level mortality rate (1980 - 2014) for HIV/AIDS

```
In [70]: fig = go.Figure(data=go.Choropleth(
             locations=dt_state_joined_pivoted['code'], # Spatial coordinates
             z = dt_state_joined_pivoted['HIV/AIDS'].astype(float), # Data to be color-coded
             locationmode = 'USA-states', # set of locations match entries in `locations`
             colorscale = 'Reds',
             colorbar_title = "HIV/AIDS Rate chane",
         ))

         fig.update_layout(
             title_text = 'HIV/AIDS mortality rate change by State over 1980-2014',
             geo_scope='usa', # limite map scope to USA
         )

         fig.show()
```

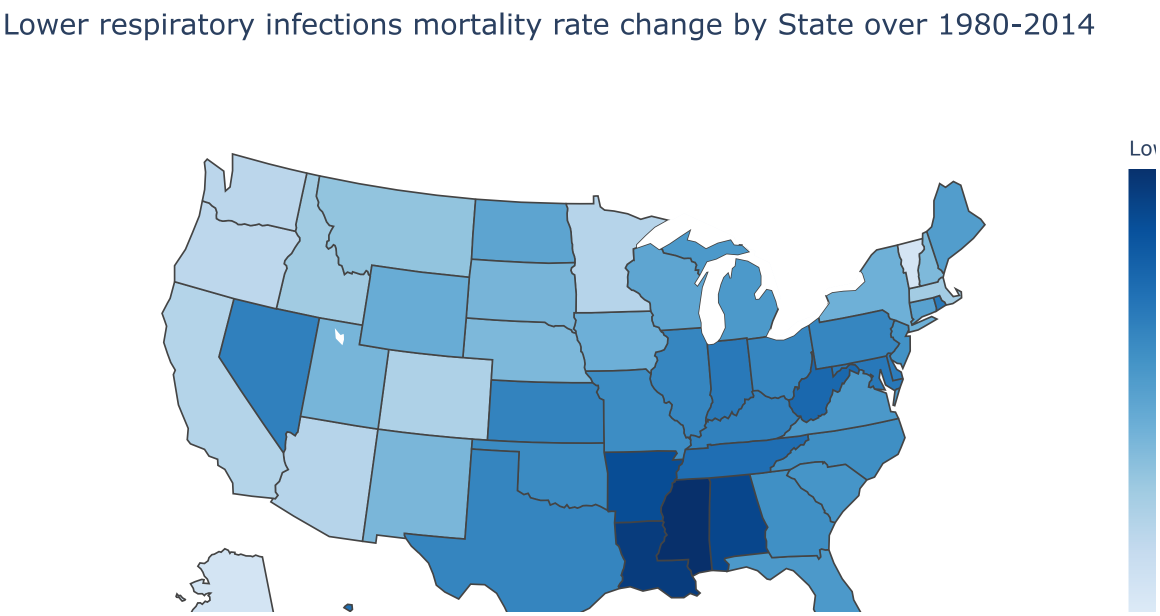
HIV/AIDS mortality rate change by State over 1980-2014



The change in state-level mortality rate (1980 - 2014) for Lower Respiratory Infections

In [71]:

```
fig = go.Figure(data=go.Choropleth(
    locations=dt_state_joined_pivoted['code'], # Spatial coordinates
    z = dt_state_joined_pivoted['Lower respiratory infections'].astype(float), # Data
to be color-coded
    locationmode = 'USA-states', # set of locations match entries in `locations`
    colorscale = 'Blues',
    colorbar_title = "Lower respiratory infections Rate chane",
))

fig.update_layout(
    title_text = 'Lower respiratory infections mortality rate change by State over 198
0-2014',
    geo_scope='usa', # limite map scope to USA
)

fig.show()
```


Lower respiratory infections mortality rate change by State over 1980-2014
Lo

Works on United States File

Loading United states Data file of mortality rates of diseases and splitting data for both sex:

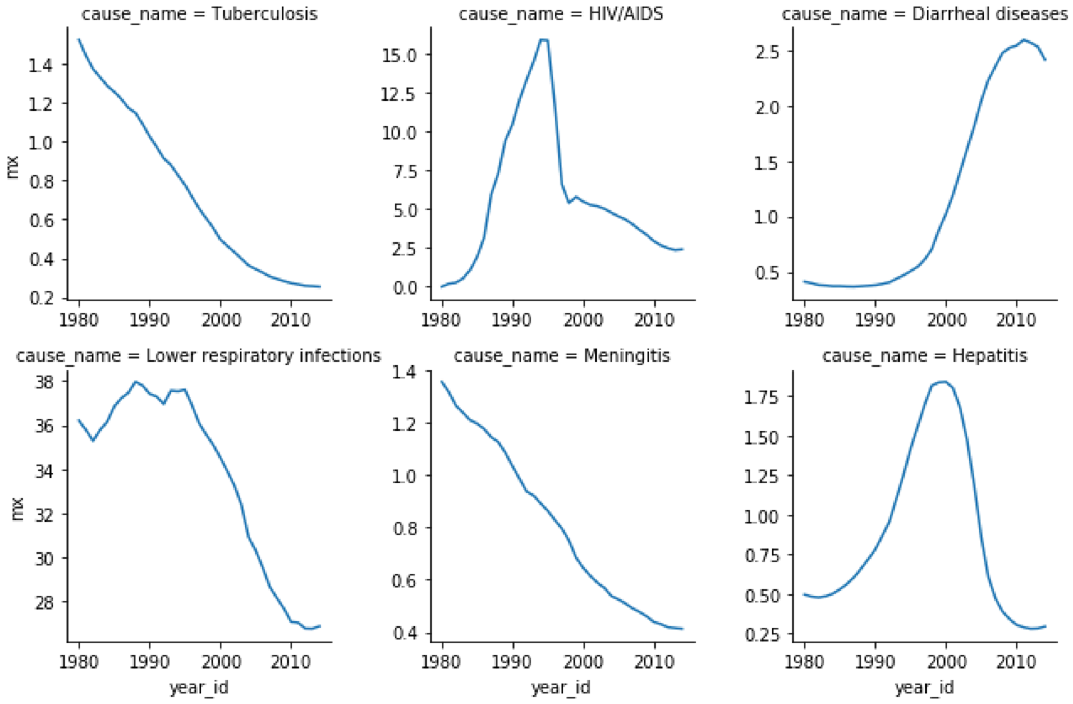
In [72]:
```
d_us = pd.read_csv('IHME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_UNITED_STATES_Y2018M03D27.csv')
d_us_both = d_us[d_us.sex_id == 3]
d_us_both = d_us_both[['location_name','cause_name','year_id','mx']]
d_us_both.head()
```

Out[72]:

	location_name	cause_name	year_id	mx
70	United States	Tuberculosis	1980	1.523290
71	United States	Tuberculosis	1981	1.440302
72	United States	Tuberculosis	1982	1.373205
73	United States	Tuberculosis	1983	1.330390
74	United States	Tuberculosis	1984	1.286244

Plotting this data:

In [73]:
```
g1 = sns.FacetGrid(d_us_both, col="cause_name", col_wrap = 3, sharex=False, sharey =False)
g1 = g1.map(plt.plot,"year_id", "mx")
```



for male and female

Formating United states Data of mortality rates of diseases and taking data for male and female:

In [74]:
```
d_us_sex = d_us[d_us.sex_id != 3]
d_us_sex = d_us_sex[['location_name','cause_name','sex','sex_id','year_id','mx']]
d_us_sex.tail()
```

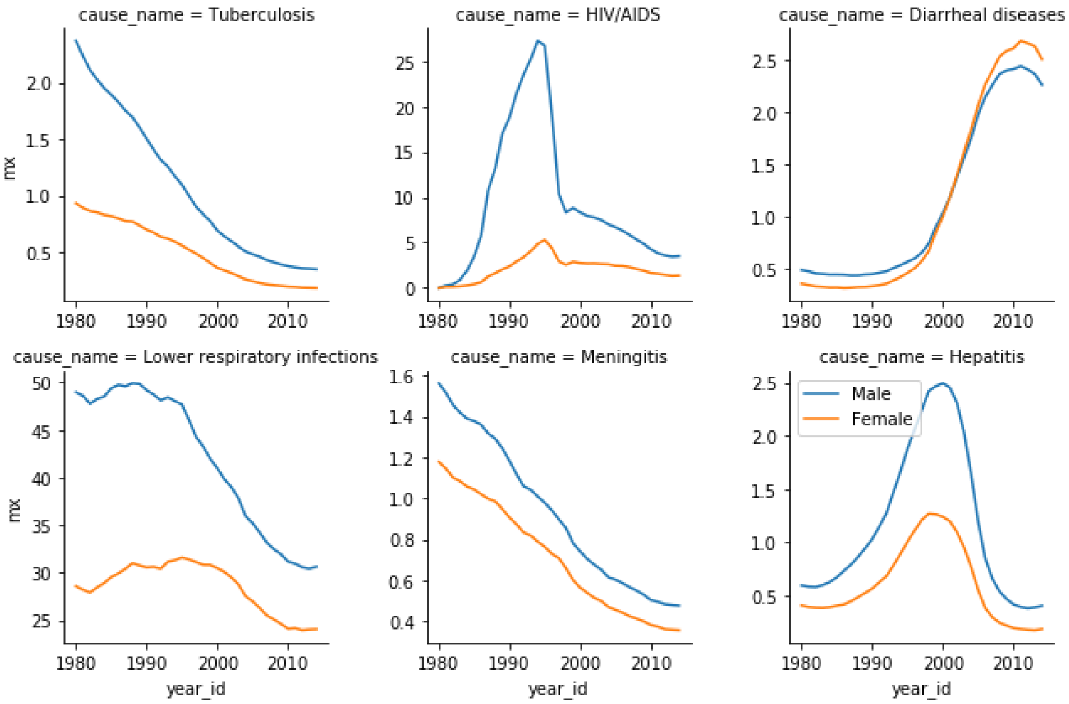
Out[74]:

	location_name	cause_name	sex	sex_id	year_id	mx
590	United States	Hepatitis	Female	2	2010	0.197970
591	United States	Hepatitis	Female	2	2011	0.188651
592	United States	Hepatitis	Female	2	2012	0.181888
593	United States	Hepatitis	Female	2	2013	0.177715
594	United States	Hepatitis	Female	2	2014	0.188535

Plotting this data:

In [75]:
```
gx = sns.FacetGrid(d_us_sex, col="cause_name", hue ='sex',col_wrap = 3, sharex=False,
sharey =False)
gx = gx.map(plt.plot,"year_id", "mx")
plt.legend()
```

Out[75]:
```
<matplotlib.legend.Legend at 0x22ad071afd0>
```



Modeling Relationships (with Lower respiratory infections)

In this section, we are going to look at the relationship between Lower Respiratory mortality rate and each of the Total population, Fraction of the population below the poverty line, and Fraction of the population with health insurance coverage, at both state and county levels. So , we are going to joining some data frame used here before of population and mortality rates.

For State Level

In [76]:
```
st_pop.head()
```

Out[76]:

	Est.Total	Health_Insurance	Total_pop	Under_poverty	NAME	FIPS	Fraction	Health_ins_frac
0	399744	2005980.0	4687615.0	1294017.0	Alabama	1	0.276050	0.427932
1	95089	309645.0	699593.0	114151.0	Alaska	2	0.163168	0.442607
2	454290	2700875.0	6399873.0	1711028.0	Arizona	4	0.267353	0.422020
3	242391	1133820.0	2857966.0	815516.0	Arkansas	5	0.285348	0.396723
4	2749907	16737950.0	37243757.0	9127115.0	California	6	0.245064	0.449416

In [77]:
```
d_state_14.head()
```

Out[77]:

cause_name	Diarrheal diseases	HIV/AIDS	Hepatitis	Lower respiratory infections	Meningitis	Tuberculosis
FIPS						
1	2.406220	2.585321	0.239779	37.393408	0.582372	0.355667
2	1.343148	0.749104	0.285058	17.639234	0.431981	0.518633
4	2.551890	1.564535	0.335842	18.021531	0.434521	0.258537
5	2.016912	1.786682	0.297565	38.644030	0.520116	0.317000
6	2.205181	2.142319	0.492770	20.695523	0.310342	0.327106

In [78]:
```
d_state_14 = d_state_14.reset_index()
```

In [79]:
```
sp_st_14 = d_state_14[['FIPS','Tuberculosis','Lower respiratory infections']]
```

In [80]:
```
sp_st_14.head()
```

Out[80]:

cause_name	FIPS	Tuberculosis	Lower respiratory infections
0	1	0.355667	37.393408
1	2	0.518633	17.639234
2	4	0.258537	18.021531
3	5	0.317000	38.644030
4	6	0.327106	20.695523

In [81]:
```
sp_st_14.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51 entries, 0 to 50
Data columns (total 3 columns):
FIPS                            51 non-null int64
Tuberculosis                    51 non-null float64
Lower respiratory infections    51 non-null float64
dtypes: float64(2), int64(1)
memory usage: 1.3 KB
```

In [82]:
```
sp_st_14['FIPS'] = sp_st_14['FIPS'].astype('i4')
```

In [83]:
```
joined_st = pd.merge(sp_st_14, st_pop, how = 'left', on = ['FIPS'])
joined_st.head()
```

Out[83]:

	FIPS	Tuberculosis	Lower respiratory infections	Est.Total	Health_Insurance	Total_pop	Under_poverty	NAME	Fraction	Health_in
0	1	0.355667	37.393408	399744	2005980.0	4687615.0	1294017.0	Alabama	0.276050	0.4
1	2	0.518633	17.639234	95089	309645.0	699593.0	114151.0	Alaska	0.163168	0.4
2	4	0.258537	18.021531	454290	2700875.0	6399873.0	1711028.0	Arizona	0.267353	0.4
3	5	0.317000	38.644030	242391	1133820.0	2857966.0	815516.0	Arkansas	0.285348	0.3
4	6	0.327106	20.695523	2749907	16737950.0	37243757.0	9127115.0	California	0.245064	0.4

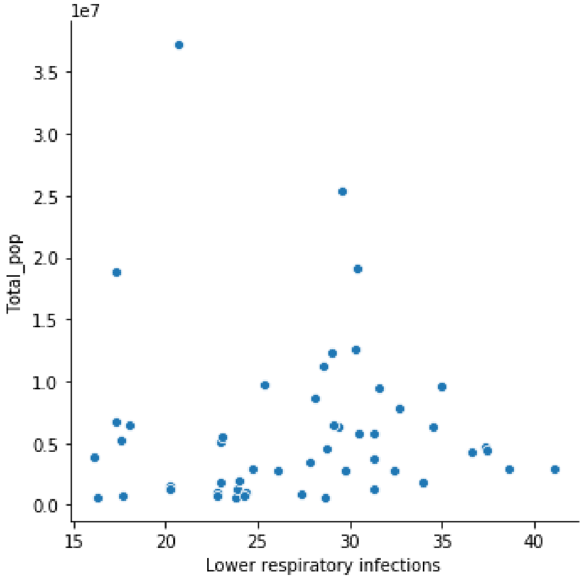
In [84]:
```
joined_st = joined_st[['FIPS','NAME','Total_pop','Under_poverty','Fraction', 'Health_Insurance','Health_ins_frac','Lower respiratory infections']]
joined_st.head()
```

Out[84]:

	FIPS	NAME	Total_pop	Under_poverty	Fraction	Health_Insurance	Health_ins_frac	Lower respiratory infections
0	1	Alabama	4687615.0	1294017.0	0.276050	2005980.0	0.427932	37.393408
1	2	Alaska	699593.0	114151.0	0.163168	309645.0	0.442607	17.639234
2	4	Arizona	6399873.0	1711028.0	0.267353	2700875.0	0.422020	18.021531
3	5	Arkansas	2857966.0	815516.0	0.285348	1133820.0	0.396723	38.644030
4	6	California	37243757.0	9127115.0	0.245064	16737950.0	0.449416	20.695523

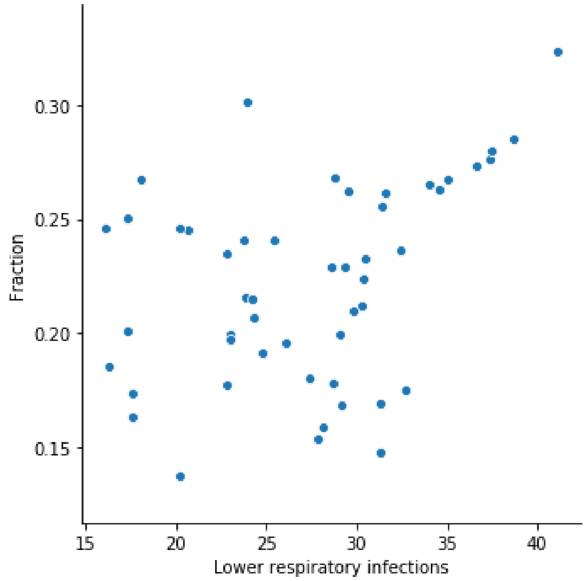
Relationship with Total population

In [85]:
```
g_pop = sns.relplot(x = "Lower respiratory infections", y = "Total_pop", data = joined_st)
```



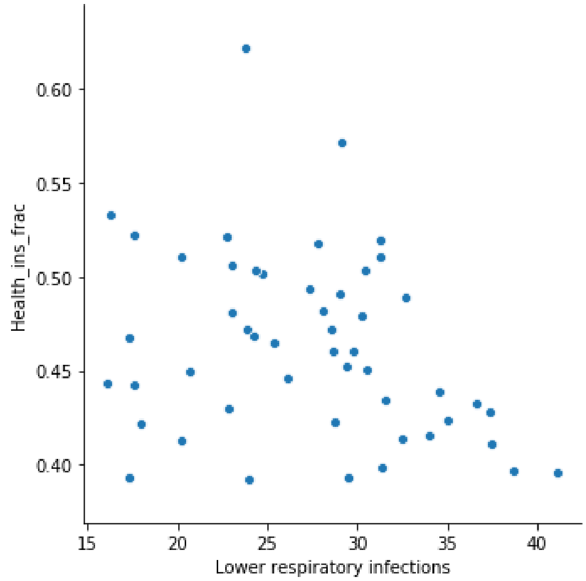
Relationship with Fraction of the population below the poverty line

In [86]:
```
g_fraction = sns.relplot(x = "Lower respiratory infections", y = "Fraction", data = jo
ined_st)
```



Relationship with Fraction of the population with health insurance coverage

In [87]:
```
g_fraction = sns.relplot(x = "Lower respiratory infections", y = "Health_ins_frac", da
ta = joined_st)
```



For County Level

In [88]:

```python
cnt_pop.sort_values('st_FIPS').head()
```

Out[88]:

	Health_Insurance	Total_pop	Under_poverty	NAME	FIPS	st_FIPS	Fraction	Health_Ins_Frac
431	48501.0	117991.0	28535.0	Morgan County, Alabama	103	1	0.241840	0.411057
425	7473.0	20227.0	7387.0	Marengo County, Alabama	91	1	0.365205	0.369457
424	160352.0	334719.0	68208.0	Madison County, Alabama	89	1	0.203777	0.479065
423	8781.0	18338.0	6820.0	Macon County, Alabama	87	1	0.371905	0.478842
422	3993.0	10794.0	4276.0	Lowndes County, Alabama	85	1	0.396146	0.369928

In [89]:

```python
cnt_pop.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3220 entries, 0 to 3219
Data columns (total 8 columns):
Health_Insurance    3220 non-null float64
Total_pop           3220 non-null float64
Under_poverty       3220 non-null float64
NAME                3220 non-null object
FIPS                3220 non-null int32
st_FIPS             3220 non-null int32
Fraction            3220 non-null float64
Health_Ins_Frac     3220 non-null float64
dtypes: float64(5), int32(2), object(1)
memory usage: 176.2+ KB
```

Formating county level FIPS to get match:

In [90]:

```python
##cnt_pop['st_FIPS'] = cnt_pop['st_FIPS'].astype('i4')
```

In [91]:

```python
cnt_pop['FIPS'] = cnt_pop['FIPS']+1000*cnt_pop['st_FIPS']
```

In [92]:

```python
cnt_pop.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3220 entries, 0 to 3219
Data columns (total 8 columns):
Health_Insurance    3220 non-null float64
Total_pop           3220 non-null float64
Under_poverty       3220 non-null float64
NAME                3220 non-null object
FIPS                3220 non-null int32
st_FIPS             3220 non-null int32
Fraction            3220 non-null float64
Health_Ins_Frac     3220 non-null float64
dtypes: float64(5), int32(2), object(1)
memory usage: 176.2+ KB
```

In [93]:

```python
sp_cnt_14.head()
```

Out[93]:

cause_name	FIPS	Tuberculosis	Lower respiratory infections
0	1001	0.309921	34.153983
1	1003	0.197299	21.820851
2	1005	0.416175	33.954553
3	1007	0.261805	41.295744
4	1009	0.258540	54.023856

In [94]:

```
sp_cnt_14.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3142 entries, 0 to 3141
Data columns (total 3 columns):
FIPS                            3142 non-null object
Tuberculosis                    3142 non-null float64
Lower respiratory infections    3142 non-null float64
dtypes: float64(2), object(1)
memory usage: 73.7+ KB
```

Joining two dataframe on FIPS:

In [95]:

```
sp_cnt_14['FIPS'] = sp_cnt_14['FIPS'].astype('i4')
```

In [96]:

```
sp_cnt_14.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3142 entries, 0 to 3141
Data columns (total 3 columns):
FIPS                            3142 non-null int32
Tuberculosis                    3142 non-null float64
Lower respiratory infections    3142 non-null float64
dtypes: float64(2), int32(1)
memory usage: 61.4 KB
```

In [97]:

```
joined_cnt = pd.merge(sp_cnt_14, cnt_pop, how = 'left', on = ['FIPS'])
```

In [98]:

```
joined_cnt.head()
```

Out[98]:

	FIPS	Tuberculosis	Lower respiratory infections	Health_Insurance	Total_pop	Under_poverty	NAME	st_FIPS	Fraction	Health_Ins_
0	1001	0.309921	34.153983	23480.0	54156.0	10982.0	Autauga County, Alabama	1.0	0.202785	0.43
1	1003	0.197299	21.820851	78069.0	188357.0	40286.0	Baldwin County, Alabama	1.0	0.213881	0.41
2	1005	0.416175	33.954553	8858.0	24215.0	8274.0	Barbour County, Alabama	1.0	0.341689	0.36
3	1007	0.261805	41.295744	8692.0	21095.0	6548.0	Bibb County, Alabama	1.0	0.310405	0.41
4	1009	0.258540	54.023856	23576.0	57052.0	15925.0	Blount County, Alabama	1.0	0.279131	0.41

In [99]:

```
joined_cnt = joined_cnt[['FIPS','NAME','Total_pop','Under_poverty','Fraction', 'Health_Insurance','Health_Ins_Frac','Lower respiratory infections']]
joined_cnt.head(50)
```

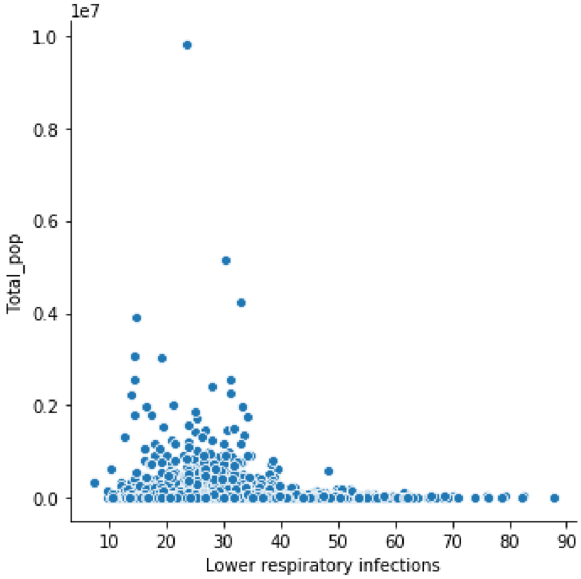
Out[99]:

	FIPS	NAME	Total_pop	Under_poverty	Fraction	Health_Insurance	Health_Ins_Frac	Lower respiratory infections
0	1001	Autauga County, Alabama	54156.0	10982.0	0.202785	23480.0	0.433562	34.153983
1	1003	Baldwin County, Alabama	188357.0	40286.0	0.213881	78069.0	0.414474	21.820851
2	1005	Barbour County, Alabama	24215.0	8274.0	0.341689	8858.0	0.365806	33.954553
3	1007	Bibb County, Alabama	21095.0	6548.0	0.310405	8692.0	0.412041	41.295744
4	1009	Blount County, Alabama	57052.0	15925.0	0.279131	23576.0	0.413237	54.023856
5	1011	Bullock County, Alabama	10153.0	3490.0	0.343741	3698.0	0.364227	30.737632
6	1013	Butler County, Alabama	20186.0	7870.0	0.389874	7485.0	0.370802	36.836894
7	1015	Calhoun County, Alabama	113928.0	35442.0	0.311091	47442.0	0.416421	61.549387
8	1017	Chambers County, Alabama	33645.0	12210.0	0.362907	12753.0	0.379046	31.094510
9	1019	Cherokee County, Alabama	25554.0	7475.0	0.292518	9845.0	0.385263	44.222744
10	1021	Chilton County, Alabama	43275.0	13219.0	0.305465	16830.0	0.388908	44.680139
11	1023	Choctaw County, Alabama	13360.0	3820.0	0.285928	4612.0	0.345210	29.405039
12	1025	Clarke County, Alabama	25008.0	8523.0	0.340811	9551.0	0.381918	27.313601
13	1027	Clay County, Alabama	13293.0	4137.0	0.311216	5304.0	0.399007	46.045348
14	1029	Cleburne County, Alabama	14828.0	3558.0	0.239951	5462.0	0.368357	46.734294
15	1031	Coffee County, Alabama	48551.0	13262.0	0.273156	19713.0	0.406027	29.080495
16	1033	Colbert County, Alabama	53940.0	14900.0	0.276233	21986.0	0.407601	37.486897
17	1035	Conecuh County, Alabama	12928.0	5925.0	0.458308	3921.0	0.303295	32.163829
18	1037	Coosa County, Alabama	10953.0	3713.0	0.338994	4235.0	0.386652	33.602889
19	1039	Covington County, Alabama	37219.0	11377.0	0.305677	13966.0	0.375238	48.572249
20	1041	Crenshaw County, Alabama	13684.0	4089.0	0.298816	5566.0	0.406752	46.189496
21	1043	Cullman County, Alabama	79538.0	22539.0	0.283374	31030.0	0.390128	33.161023
22	1045	Dale County, Alabama	47035.0	12720.0	0.270437	18524.0	0.393834	26.196303
23	1047	Dallas County, Alabama	42215.0	19558.0	0.463295	15472.0	0.366505	49.672806
24	1049	DeKalb County, Alabama	70246.0	21615.0	0.307704	26279.0	0.374100	34.549186
25	1051	Elmore County, Alabama	74474.0	14789.0	0.198579	33562.0	0.450654	33.242317
26	1053	Escambia County, Alabama	35574.0	13865.0	0.389751	12228.0	0.343734	32.487128

	FIPS	NAME	Total_pop	Under_poverty	Fraction	Health_Insurance	Health_Ins_Frac	Lower respiratory infections
27	1055	Etowah County, Alabama	102228.0	30618.0	0.299507	39419.0	0.385599	56.196489
28	1057	Fayette County, Alabama	16766.0	5411.0	0.322736	6286.0	0.374925	47.925454
29	1059	Franklin County, Alabama	31305.0	10507.0	0.335633	11422.0	0.364862	43.790860
30	1061	Geneva County, Alabama	26338.0	8836.0	0.335485	9688.0	0.367834	24.276235
31	1063	Greene County, Alabama	8750.0	4602.0	0.525943	3054.0	0.349029	42.981052
32	1065	Hale County, Alabama	15084.0	5638.0	0.373774	5546.0	0.367674	42.372049
33	1067	Henry County, Alabama	17057.0	4141.0	0.242774	6445.0	0.377851	24.045944
34	1069	Houston County, Alabama	101741.0	28404.0	0.279179	40992.0	0.402905	21.959268
35	1071	Jackson County, Alabama	52438.0	15986.0	0.304855	19872.0	0.378962	34.960660
36	1073	Jefferson County, Alabama	644032.0	170087.0	0.264097	292404.0	0.454021	39.547775
37	1075	Lamar County, Alabama	14101.0	4454.0	0.315864	5343.0	0.378909	44.744290
38	1077	Lauderdale County, Alabama	90730.0	23828.0	0.262625	40434.0	0.445652	33.018169
39	1079	Lawrence County, Alabama	33506.0	8027.0	0.239569	13587.0	0.405509	35.656095
40	1081	Lee County, Alabama	141277.0	44722.0	0.316555	75323.0	0.533158	34.979634
41	1083	Limestone County, Alabama	82885.0	17143.0	0.206829	36415.0	0.439344	27.435009
42	1085	Lowndes County, Alabama	10794.0	4276.0	0.396146	3993.0	0.369928	31.343562
43	1087	Macon County, Alabama	18338.0	6820.0	0.371905	8781.0	0.478842	37.588477
44	1089	Madison County, Alabama	334719.0	68208.0	0.203777	160352.0	0.479065	31.129898
45	1091	Marengo County, Alabama	20227.0	7387.0	0.365205	7473.0	0.369457	35.217389
46	1093	Marion County, Alabama	29593.0	9001.0	0.304160	10836.0	0.366168	40.093555
47	1095	Marshall County, Alabama	92975.0	28004.0	0.301199	33891.0	0.364517	39.638569
48	1097	Mobile County, Alabama	403388.0	117021.0	0.290095	165952.0	0.411395	36.031143
49	1099	Monroe County, Alabama	22261.0	8631.0	0.387718	8200.0	0.368357	37.042465

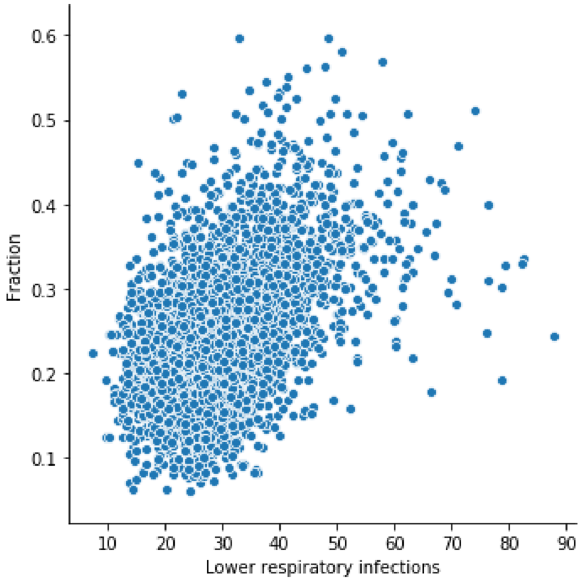
Relationship with Total population

```
In [100]: gc_pop = sns.relplot(x = "Lower respiratory infections", y = "Total_pop", data = joine
d_cnt)
```



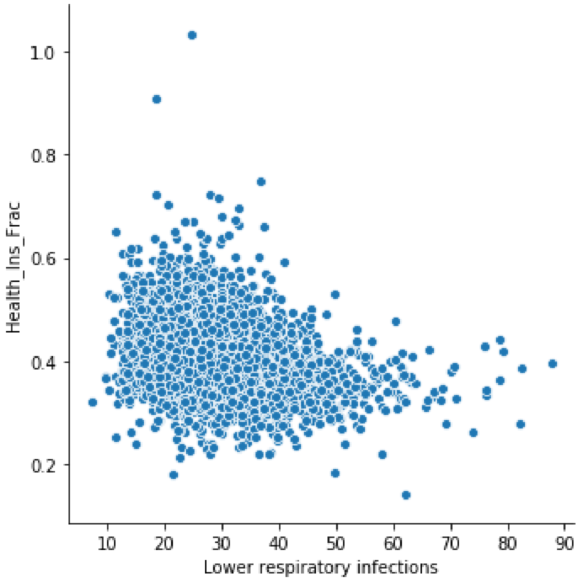
Relationship with Fraction of the population below the poverty line

```
In [101]: gc_fraction = sns.relplot(x = "Lower respiratory infections", y = "Fraction", data = j
oined_cnt)
```



Relationship with Fraction of the population with health insurance coverage

```
In [102]: g_fraction = sns.relplot(x = "Lower respiratory infections", y = "Health_Ins_Frac", da
ta = joined_cnt)
```



```
In [ ]:
```