Assignment 3

Setup

Installing the census, openpyxl and US packages

In [1]:
```
%pip install census us
```

```
Requirement already satisfied: census in /home/nahidanwar/anaconda3/lib/pytho
n3.8/site-packages (0.8.17)
Requirement already satisfied: us in /home/nahidanwar/anaconda3/lib/python3.
8/site-packages (2.0.2)
Requirement already satisfied: requests>=1.1.0 in /home/nahidanwar/anaconda3/
lib/python3.8/site-packages (from census) (2.25.1)
Requirement already satisfied: idna<3,>=2.5 in /home/nahidanwar/anaconda3/li
b/python3.8/site-packages (from requests>=1.1.0->census) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in /home/nahidanwar/anacond
a3/lib/python3.8/site-packages (from requests>=1.1.0->census) (2020.12.5)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /home/nahidanwar/anac
onda3/lib/python3.8/site-packages (from requests>=1.1.0->census) (1.26.4)
Requirement already satisfied: chardet<5,>=3.0.2 in /home/nahidanwar/anaconda
3/lib/python3.8/site-packages (from requests>=1.1.0->census) (4.0.0)
Requirement already satisfied: jellyfish==0.6.1 in /home/nahidanwar/anaconda
3/lib/python3.8/site-packages (from us) (0.6.1)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:
```
conda install openpyxl
```

```
Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.


Note: you may need to restart the kernel to use updated packages.
```

Importing Necessary Python Libraries

In [3]:
```
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
```

Setting up a census object with API key

In [4]:
```
c = Census('425133029e445c12eb64404fc6e09de4fefba42a')
```

Load Census Data

In [5]:
```
df_poverty = pd.DataFrame.from_records(c.acs5.state(('NAME', 'B05010_001E', '
df_poverty.head()
```

Out[5]:

NAME	B05010_001E	B05010_002E	state

	NAME	B05010_001E	B05010_002E	state
0	Alabama	1048560.0	281052.0	01
1	Alaska	179242.0	23963.0	02
2	Arizona	1532525.0	385737.0	04
3	Arkansas	663036.0	179070.0	05
4	California	8778017.0	1945049.0	06

In [6]:
```
df_poverty.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 52 entries, 0 to 51
Data columns (total 4 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   NAME         52 non-null     object 
 1   B05010_001E  52 non-null     float64
 2   B05010_002E  52 non-null     float64
 3   state        52 non-null     object 
dtypes: float64(2), object(2)
memory usage: 1.8+ KB
```

In [7]:
```
df_poverty = df_poverty.rename(columns={'B05010_001E':'TOTAL', 'B05010_002E':
```

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

Out[8]:

	NAME	TOTAL	UNDER-POVERTY	state
0	Alabama	1048560.0	281052.0	01
1	Alaska	179242.0	23963.0	02
2	Arizona	1532525.0	385737.0	04
3	Arkansas	663036.0	179070.0	05
4	California	8778017.0	1945049.0	06

In [9]:
```
df_poverty['POVERTY-RATE'] = df_poverty['UNDER-POVERTY']/df_poverty['TOTAL']
```

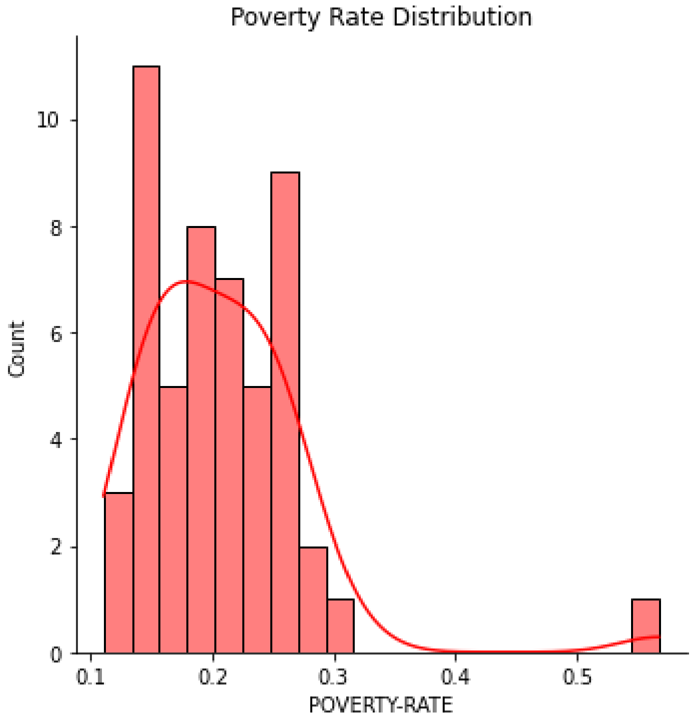
Poverty rate distribution both numerically and graphically

In [10]:
```
df_poverty['POVERTY-RATE'].describe()
```

Out[10]:
```
count    52.000000
mean      0.207473
std       0.070696
min       0.110946
25%       0.153843
50%       0.196620
75%       0.247620
max       0.566731
Name: POVERTY-RATE, dtype: float64
```

In [11]:
```
sns.displot(x = df_poverty['POVERTY-RATE'], color = 'red', edgecolor = 'black
```

Out[11]: <seaborn.axisgrid.FacetGrid at 0x7ff0a1789850>



The above distribution is right-skewed and has small outliers.

In [12]:

```
df_poverty.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 52 entries, 0 to 51
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   NAME           52 non-null     object
 1   TOTAL          52 non-null     float64
 2   UNDER-POVERTY  52 non-null     float64
 3   state          52 non-null     object
 4   POVERTY-RATE   52 non-null     float64
dtypes: float64(3), object(2)
memory usage: 2.2+ KB
```

The relationship of poverty rates between mortality rates of Meningitis and Diarrheal desease

In [13]:

```
df_poverty.rename(columns={'state':'FIPS'}, inplace = True)
```

In [14]:

```
df_poverty.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 52 entries, 0 to 51
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   NAME           52 non-null     object
 1   TOTAL          52 non-null     float64
 2   UNDER-POVERTY  52 non-null     float64
 3   FIPS           52 non-null     object
 4   POVERTY-RATE   52 non-null     float64
dtypes: float64(3), object(2)
memory usage: 2.2+ KB
```

In [15]:
```
df_poverty['FIPS'] = df_poverty['FIPS'].astype(int)
```

In [16]:
```
df_poverty.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 52 entries, 0 to 51
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   NAME           52 non-null     object 
 1   TOTAL          52 non-null     float64
 2   UNDER-POVERTY  52 non-null     float64
 3   FIPS           52 non-null     int64  
 4   POVERTY-RATE   52 non-null     float64
dtypes: float64(3), int64(1), object(1)
memory usage: 2.2+ KB
```

Load Infectious Deseases — GHDx data

In [17]:
```
sheet1 = pd.read_excel('IHME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_NATIONAL_Y2
                       sheet_name = 'Meningitis', header = 1, skipfooter = 2)
```

In [18]:
```
sheet1.head()
```

Out[18]:

	Location	FIPS	Mortality Rate, 1980*	Mortality Rate, 1985*	Mortality Rate, 1990*	Mortality Rate, 1995*	Mortality Rate, 2000*	Mortality Rate, 2005*	Mortality Rate, 2010*	Morta Ra 20:
0	United States	NaN	1.36 (1.30, 1.42)	1.20 (1.15, 1.24)	1.03 (0.99, 1.07)	0.86 (0.83, 0.89)	0.65 (0.62, 0.67)	0.52 (0.51, 0.54)	0.44 (0.42, 0.46)	0 (0. 0.
1	Alabama	1.0	1.49 (1.39, 1.59)	1.37 (1.29, 1.45)	1.24 (1.17, 1.31)	1.08 (1.01, 1.14)	0.83 (0.79, 0.88)	0.69 (0.65, 0.74)	0.60 (0.56, 0.65)	0 (0. 0.
2	Autauga County, Alabama	1001.0	1.34 (1.15, 1.55)	1.25 (1.08, 1.43)	1.16 (0.98, 1.33)	0.96 (0.80, 1.10)	0.71 (0.59, 0.83)	0.61 (0.51, 0.72)	0.51 (0.41, 0.62)	0 (0. 0.
3	Baldwin County, Alabama	1003.0	1.23 (1.08, 1.41)	1.10 (0.98, 1.25)	0.97 (0.85, 1.10)	0.81 (0.71, 0.92)	0.60 (0.52, 0.69)	0.49 (0.42, 0.56)	0.43 (0.36, 0.51)	0 (0. 0.
4	Barbour County, Alabama	1005.0	1.78 (1.54, 2.03)	1.60 (1.39, 1.82)	1.41 (1.21, 1.59)	1.20 (1.02, 1.39)	0.94 (0.81, 1.08)	0.78 (0.66, 0.91)	0.66 (0.55, 0.78)	0 (0. 0.

In [19]:
```
sheet1.rename(columns = {'Mortality Rate, 2014*':'Meningitis_MR'}, inplace =
```

In [20]:
```
M = sheet1[['FIPS','Meningitis_MR']]
```

In [21]:
```
M.head()
```

Out[21]:

	FIPS	Meningitis_MR

	FIPS	Meningitis_MR
0	NaN	0.41 (0.40, 0.43)
1	1.0	0.58 (0.54, 0.64)
2	1001.0	0.51 (0.41, 0.63)
3	1003.0	0.41 (0.34, 0.50)
4	1005.0	0.64 (0.53, 0.76)

In [22]:
```
sheet2 = pd.read_excel('IHME_USA_COUNTY_INFECT_DIS_MORT_1980_2014_NATIONAL_Y2
                    sheet_name = 'Diarrheal diseases', header = 1, skipfoote
```

In [23]:
```
sheet2.rename(columns = {'Mortality Rate, 2014*':'Diarrhea_MR'}, inplace = Tr
```

In [24]:
```
D = sheet2[['FIPS','Diarrhea_MR']]
```

In [25]:
```
D.head()
```

Out[25]:

	FIPS	Diarrhea_MR
0	NaN	2.41 (0.86, 2.67)
1	1.0	2.41 (0.89, 2.70)
2	1001.0	1.89 (0.74, 2.64)
3	1003.0	1.44 (0.55, 1.90)
4	1005.0	2.02 (0.83, 2.87)

In [26]:
```
df = M.merge(D, on = ['FIPS'])
```

In [27]:
```
df.head()
```

Out[27]:

	FIPS	Meningitis_MR	Diarrhea_MR
0	NaN	0.41 (0.40, 0.43)	2.41 (0.86, 2.67)
1	1.0	0.58 (0.54, 0.64)	2.41 (0.89, 2.70)
2	1001.0	0.51 (0.41, 0.63)	1.89 (0.74, 2.64)
3	1003.0	0.41 (0.34, 0.50)	1.44 (0.55, 1.90)
4	1005.0	0.64 (0.53, 0.76)	2.02 (0.83, 2.87)

In [28]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3194 entries, 0 to 3193
Data columns (total 3 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   FIPS           3193 non-null   float64
 1   Meningitis_MR  3194 non-null   object 
 2   Diarrhea_MR    3194 non-null   object 
```

```
dtypes: float64(1), object(2)
memory usage: 99.8+ KB
```

Converting mortality rates for both diseases by excluding the confidence interval that is in the parentheses

In [29]:
```
df['Meningitis_MR'] = df['Meningitis_MR'].str.replace(r'\s*\(.+\)', '', regex
df['Diarrhea_MR'] = df['Diarrhea_MR'].str.replace(r'\s*\(.+\)', '', regex=Tru
```

In [30]:
```
df1 = df_poverty.merge(df, on = ['FIPS'])
```

In [31]:
```
df1.head()
```

Out[31]:

	NAME	TOTAL	UNDER-POVERTY	FIPS	POVERTY-RATE	Meningitis_MR	Diarrhea_MR
0	Alabama	1048560.0	281052.0	1	0.268036	0.58	2.41
1	Alaska	179242.0	23963.0	2	0.133691	0.43	1.34
2	Arizona	1532525.0	385737.0	4	0.251700	0.43	2.55
3	Arkansas	663036.0	179070.0	5	0.270076	0.52	2.02
4	California	8778017.0	1945049.0	6	0.221582	0.31	2.21

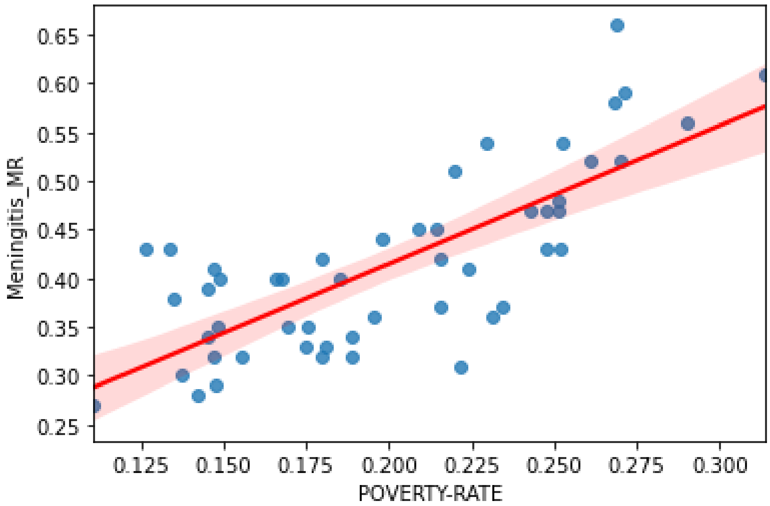
In [32]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 51 entries, 0 to 50
Data columns (total 7 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   NAME            51 non-null     object
 1   TOTAL           51 non-null     float64
 2   UNDER-POVERTY   51 non-null     float64
 3   FIPS            51 non-null     int64
 4   POVERTY-RATE    51 non-null     float64
 5   Meningitis_MR   51 non-null     float64
 6   Diarrhea_MR     51 non-null     float64
dtypes: float64(5), int64(1), object(1)
memory usage: 3.2+ KB
```

Relationship between poverty rate and meningitis mortality rate

In [33]:
```
sns.regplot(data = df1, x = 'POVERTY-RATE', y = 'Meningitis_MR', line_kws={'c
```

Out[33]:
```
<AxesSubplot:xlabel='POVERTY-RATE', ylabel='Meningitis_MR'>
```

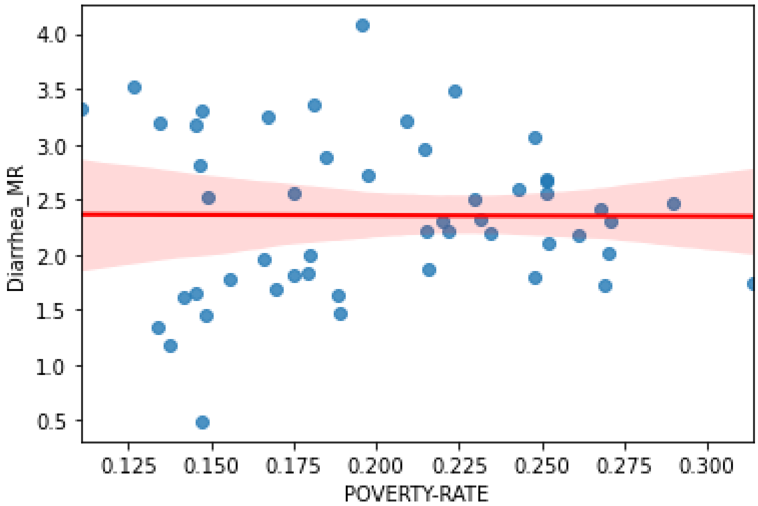


The above pattern clearly shows an uphill linear distribution which indicates that the mortality rate by Meningitis disease increases with the increase of poverty rate.

Relationship between poverty rate and diarrhea mortality rate

In [34]:
```
sns.regplot(data = df1, x = 'POVERTY-RATE', y = 'Diarrhea_MR', line_kws={'col
```

Out[34]: <AxesSubplot:xlabel='POVERTY-RATE', ylabel='Diarrhea_MR'>



The trendline is parallel to X-axis which means that mortality rate by Diarrhea doesn't change(either increase or decrease) much with the change of poverty rate.

Quantify the relationship between poverty rate and meningitis mortality rate by computing correlation coefficients with bootstrapped confidence intervals

Covariance allows us to measure how two variable varies. Correlation coefficient helps us to measure the degree of linear relationshipd between two variables.

In [35]:
```
df1['Meningitis_MR'].cov(df1['POVERTY-RATE'])
```

Out[35]: 0.003501853874082061

In [36]:
```
df1['Meningitis_MR'].corr(df1['POVERTY-RATE'])
```

Out[36]:
```
0.7666266512876146
```

Since here the correlation coefficient is closer to +1 which indicates a strong positive relationship between mortality rate by Meningitis and poverty rate.

In [37]:
```
NBOOT = 10000  #For Meningitis
boot_corrs = np.empty(NBOOT)
for i in range(NBOOT):
    samp = df1.sample(n=len(df1), replace=True)
    boot_corrs[i] = samp['Meningitis_MR'].corr(samp['POVERTY-RATE'])
np.quantile(boot_corrs, [0.025, 0.975])
```

Out[37]:
```
array([0.62659665, 0.86232195])
```

Quantify the relationship between poverty rate and diarrhea mortality rate by computing correlation coefficients with bootstrapped confidence intervals

In [38]:
```
df1['Diarrhea_MR'].cov(df1['POVERTY-RATE'])
```

Out[38]:
```
-0.00021244792825037651
```

In [39]:
```
df1['Diarrhea_MR'].corr(df1['POVERTY-RATE'])
```

Out[39]:
```
-0.00599044853100433
```

Since here the correlation coefficient is slightly less than 0 which indicates a minimal negative association between mortality rate by Diarrhea and poverty rate.

In [40]:
```
NBOOT = 10000  #For Diarrhea
boot_corrs = np.empty(NBOOT)
for i in range(NBOOT):
    samp = df1.sample(n=len(df1), replace=True)
    boot_corrs[i] = samp['Diarrhea_MR'].corr(samp['POVERTY-RATE'])
np.quantile(boot_corrs, [0.025, 0.975])
```

Out[40]:
```
array([-0.28944121,  0.27640077])
```

Load Infant Mortality — CDC Data

In [41]:
```
df_infant = pd.read_csv('undefined.csv')
df_infant.head()
```

Out[41]:

	YEAR	STATE	RATE	DEATHS	URL
0	2019	AL	7.89	449	/nchs/pressroom/states/alabama/al.htm
1	2019	AK	4.81	48	/nchs/pressroom/states/alaska/ak.htm
2	2019	AZ	5.24	429	/nchs/pressroom/states/arizona/az.htm
3	2019	AR	6.9	251	/nchs/pressroom/states/arkansas/ar.htm
4	2019	CA	4.06	1879	/nchs/pressroom/states/california/ca.htm

In [42]:
```
df_infant.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 350 entries, 0 to 349
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   YEAR    350 non-null    int64 
 1   STATE   350 non-null    object
 2   RATE    350 non-null    object
 3   DEATHS  350 non-null    int64 
 4   URL     350 non-null    object
dtypes: int64(2), object(3)
memory usage: 13.8+ KB
```

Computing Infant mortality rate

In [43]:
```
df_infant = df_infant[df_infant['YEAR'] == 2014.0]
```

In [44]:
```
FIPS_code = pd.read_table('state.txt', sep='|')
FIPS_code.head()
```

Out[44]:

	STATE	STUSAB	STATE_NAME	STATENS
0	1	AL	Alabama	1779775
1	2	AK	Alaska	1785533
2	4	AZ	Arizona	1779777
3	5	AR	Arkansas	68085
4	6	CA	California	1779778

In [45]:
```
FIPS_code.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 57 entries, 0 to 56
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   STATE       57 non-null     int64 
 1   STUSAB      57 non-null     object
 2   STATE_NAME  57 non-null     object
 3   STATENS     57 non-null     int64 
dtypes: int64(2), object(2)
memory usage: 1.9+ KB
```

In [46]:
```
FIPS_code.rename(columns = {'STATE':'FIPS'}, inplace = True)
```

In [47]:
```
FIPS_code.rename(columns = {'STUSAB':'STATE'}, inplace = True)
```

Merging infant mortality with the state codes file by state abbreviation to get FIPS codes

In [48]:
```
df2 = df_infant.merge(FIPS_code, on = ['STATE'])
df2.head()
```

Out[48]:

	YEAR	STATE	RATE	DEATHS	URL	FIPS	STATE_NAME	STATEI
0	2014	AL	8.67	515	/nchs/pressroom/states/alabama.htm	1	Alabama	17797

	YEAR	STATE	RATE	DEATHS	URL	FIPS	STATE_NAME	STATEI
1	2014	AK	6.67	76	/nchs/pressroom/states/alaska.htm	2	Alaska	17855
2	2014	AZ	6.1	530	/nchs/pressroom/states/arizona.htm	4	Arizona	17797
3	2014	AR	7.48	288	/nchs/pressroom/states/arkansas.htm	5	Arkansas	680
4	2014	CA	4.32	2173	/nchs/pressroom/states/california.htm	6	California	17797

Merging the resulting table with census data by FIPS code

In [49]:
```
df3 = df_poverty.merge(df2, on = ['FIPS'])
df3.head()
```

Out[49]:

	NAME	TOTAL	UNDER-POVERTY	FIPS	POVERTY-RATE	YEAR	STATE	RATE	DEATHS	
0	Alabama	1048560.0	281052.0	1	0.268036	2014	AL	8.67	515	/nchs/pressr
1	Alaska	179242.0	23963.0	2	0.133691	2014	AK	6.67	76	/nchs/press
2	Arizona	1532525.0	385737.0	4	0.251700	2014	AZ	6.1	530	/nchs/press
3	Arkansas	663036.0	179070.0	5	0.270076	2014	AR	7.48	288	/nchs/pressro
4	California	8778017.0	1945049.0	6	0.221582	2014	CA	4.32	2173	/nchs/pressro

In [50]:
```
df3.rename(columns = {'RATE':'Infant_MR'}, inplace = True)
```

In [51]:
```
df3.head()
```

Out[51]:

	NAME	TOTAL	UNDER-POVERTY	FIPS	POVERTY-RATE	YEAR	STATE	Infant_MR	DEATHS	
0	Alabama	1048560.0	281052.0	1	0.268036	2014	AL	8.67	515	/nchs/pr
1	Alaska	179242.0	23963.0	2	0.133691	2014	AK	6.67	76	/nchs/
2	Arizona	1532525.0	385737.0	4	0.251700	2014	AZ	6.1	530	/nchs/p
3	Arkansas	663036.0	179070.0	5	0.270076	2014	AR	7.48	288	/nchs/pr
4	California	8778017.0	1945049.0	6	0.221582	2014	CA	4.32	2173	/nchs/pr

In [52]:
```
df3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 50 entries, 0 to 49
Data columns (total 12 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   NAME           50 non-null     object 
 1   TOTAL          50 non-null     float64
 2   UNDER-POVERTY  50 non-null     float64
 3   FIPS           50 non-null     int64  
 4   POVERTY-RATE   50 non-null     float64
 5   YEAR           50 non-null     int64  
 6   STATE          50 non-null     object 
 7   Infant_MR      50 non-null     object 
 8   DEATHS         50 non-null     int64  
```

```
 9   URL            50 non-null     object
 10  STATE_NAME     50 non-null     object
 11  STATENS        50 non-null     int64
dtypes: float64(3), int64(4), object(5)
memory usage: 5.1+ KB
```

Converting the data type of the Infant Mortality column

In [53]:
```
df3['Infant_MR'] = df3['Infant_MR'].astype(float)
df3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 50 entries, 0 to 49
Data columns (total 12 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   NAME           50 non-null     object
 1   TOTAL          50 non-null     float64
 2   UNDER-POVERTY  50 non-null     float64
 3   FIPS           50 non-null     int64
 4   POVERTY-RATE   50 non-null     float64
 5   YEAR           50 non-null     int64
 6   STATE          50 non-null     object
 7   Infant_MR      50 non-null     float64
 8   DEATHS         50 non-null     int64
 9   URL            50 non-null     object
 10  STATE_NAME     50 non-null     object
 11  STATENS        50 non-null     int64
dtypes: float64(4), int64(4), object(4)
memory usage: 5.1+ KB
```

In [54]:
```
df3 = df3[['POVERTY-RATE', 'Infant_MR']]
df3.head()
```

Out[54]:

	POVERTY-RATE	Infant_MR
0	0.268036	8.67
1	0.133691	6.67
2	0.251700	6.10
3	0.270076	7.48
4	0.221582	4.32

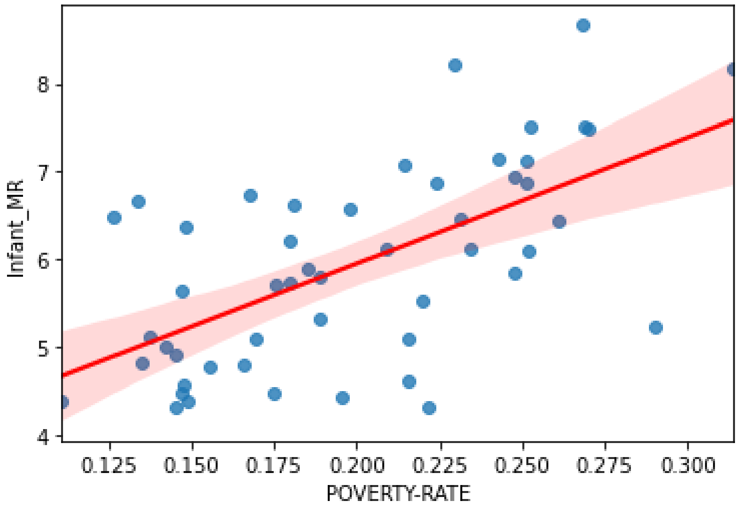
In [55]:
```
df3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 50 entries, 0 to 49
Data columns (total 2 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   POVERTY-RATE  50 non-null     float64
 1   Infant_MR     50 non-null     float64
dtypes: float64(2)
memory usage: 1.2 KB
```

Relationship between poverty rate and Infant mortality rate

In [56]:
```
sns.regplot(data = df3, x = 'POVERTY-RATE', y = 'Infant_MR', line_kws = {'col
```

Out[56]: <AxesSubplot:xlabel='POVERTY-RATE', ylabel='Infant_MR'>



The above pattern shows an uphill linear distribution which indicates that the Infant mortality rate increases with the increase of poverty rate.

Quantify the relationship between poverty rate and infant mortality rate by computing correlation coefficients with bootstrapped confidence intervals

In [57]:
```
(df3['POVERTY-RATE']).cov(df3['Infant_MR'])
```

Out[57]: 0.034733340337397015

In [58]:
```
(df3['POVERTY-RATE']).corr(df3['Infant_MR'])
```

Out[58]: 0.6154666056611091

In [59]:
```
NBOOT = 10000  #For Infant Mortality
boot_corrs = np.empty(NBOOT)
for i in range(NBOOT):
    samp = df3.sample(n=len(df1), replace=True)
    boot_corrs[i] = samp['Infant_MR'].corr(samp['POVERTY-RATE'])
np.quantile(boot_corrs, [0.025, 0.975])
```

Out[59]: array([0.40072362, 0.78549555])

The correlation is significant because there is positive association between the infant mortality rate and poverty rate which indicates that infant mortality rate increases with the increase of poverty rate.

What I learn from these data and limitations of the data and analysis:

In this assignment, I have to use three data sets. The **Poverty data** from the U.S. Census Bureau and **Health data** from the Global Health Data Exchange and the **Infant Mortality data** from US Centers for Disease Control. In all three cases, I only use data for the year **2014**. Before using the data, I installed the python package for the Census API and a U.S. state code data package. For census data, I used the **ACS5** formatted files as instructed. From the census data distribution, I found that most states have a poverty rate of less than 0.3 except for only two states(Mississippi: 0.313809, Puerto Rico: 0.566731).

I need to join data sets among them on FIPS code because I had to find the relationship of diseases mortality rate(Meningitis, Diarrhea) and Infant mortality with respect to the poverty rate so that I can answer the overall question of this assignment which is **Are health outcomes correlated with poverty levels in a community?** In my analysis, I found that mortality rate by Meningitis and infant mortality rate are affected by poverty rate, and they are positively correlated. But there is an exception for mortality rate by Diarrhea which is slightly negative-correlated with poverty rate. Another thing I learned from this data is that I always have to look into the data to skip the unnecessary header and footer if available.

ACS5 is a supplementary annual survey of a population sample carried out by the census bureau, **5-year estimates**. This affects the validity and stability of the data analysis. For this assignment, I had to work with the year 2014, which may vary in the result for all other years. That is why the outcome of the data may not be fully justified for overall years data. To get a more accurate measurement and association, large sample data may play an important role. In some cases, the specific column of a data set has different types while joining the data with another data set. For example, both the census and the infectious disease table use FIPS codes, and to join or merge (subsets of) the two tables by their FIPS code, the census data (initial type is a string) needed to convert into a number (.astype('int')) before joining. Also, in this assignment, the data we used have different formats(some of them are .csv files, some of them are .XLSX, etc.). Therefore, we need various pandas libraries and install python packages to read those different data sets.