Health Care

In [38]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline
```

1. Retrieving DataSets

In [85]:

```
#Population by age and sex
pop_agesex=pd.read_csv("F:/PythonTraining/Python_CaseStudy/PCS_HealthCare/Popul_estim_sex_a

#Population by race and sex
pop_racesex=pd.read_csv("F:/PythonTraining/Python_CaseStudy/PCS_HealthCare/Popul_estim_sex_

#Flu State weeks
flu_week=pd.read_csv("F:/PythonTraining/Python_CaseStudy/PCS_HealthCare/flu_states_weeks.cs

#SeasonFlue
flu_season=pd.read_csv("F:/PythonTraining/Python_CaseStudy/PCS_HealthCare/StateDatabySeason
```

2. Statistics Population by Age

In [6]:

```
pop_agesex.head()
```

Out[6]:

	STATE_ABR	SUMLEV	REGION	DIVISION	STATE	NAME	SEX	AGE	ESTBASE2010_CIV	F
0	US	10	0	0	0	United States	0	0	3944160	
1	US	10	0	0	0	United States	0	1	3978090	
2	US	10	0	0	0	United States	0	2	4096939	
3	US	10	0	0	0	United States	0	3	4119051	
4	US	10	0	0	0	United States	0	4	4063186	

In [22]:

```
pop_agesex.columns
```

Out[22]:

```
Index(['STATE_ABR', 'SUMLEV', 'REGION', 'DIVISION', 'STATE', 'NAME', 'SEX',
       'AGE', 'ESTBASE2010_CIV', 'POPEST2010_CIV', 'POPEST2011_CIV',
       'POPEST2012_CIV', 'POPEST2013_CIV', 'POPEST2014_CIV', 'POPEST2015_CI
V',
       'POPEST2016_CIV', 'POPEST2017_CIV'],
      dtype='object')
```

In [21]:

```
#Checking unique values for age
print("The unique values are: ", len(pop_agesex.AGE.unique()))
pop_agesex.AGE.unique()
```

```
The unique values are:  87
```

Out[21]:

```
array([  0,   1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  11,  12,
        13,  14,  15,  16,  17,  18,  19,  20,  21,  22,  23,  24,  25,
        26,  27,  28,  29,  30,  31,  32,  33,  34,  35,  36,  37,  38,
        39,  40,  41,  42,  43,  44,  45,  46,  47,  48,  49,  50,  51,
        52,  53,  54,  55,  56,  57,  58,  59,  60,  61,  62,  63,  64,
        65,  66,  67,  68,  69,  70,  71,  72,  73,  74,  75,  76,  77,
        78,  79,  80,  81,  82,  83,  84,  85, 999], dtype=int64)
```

In [26]:

```
#Subset of dataset
pop_agesex_sub1=pop_agesex[['NAME', 'SEX', 'AGE', 'ESTBASE2010_CIV', 'POPEST2010_CIV', 'POP
                 'POPEST2012_CIV', 'POPEST2013_CIV', 'POPEST2014_CIV', 'POPEST2015_CIV', 'P

#Setting index
pop_agesex_sub1=pop_agesex_sub1.set_index("NAME")
```

2.1 Population by age and sex basic statistics

In [42]:

```
pop_agesex_sub1.describe()
```

Out[42]:

	SEX	AGE	ESTBASE2010_CIV	POPEST2010_CIV	POPEST2011_CIV	PO
count	13572.000000	13572.000000	1.357200e+04	1.357200e+04	1.357200e+04	
mean	1.000000	53.494253	1.812720e+05	1.816111e+05	1.829806e+05	
std	0.816527	104.905228	3.319803e+06	3.326021e+06	3.351248e+06	
min	0.000000	0.000000	3.880000e+02	3.850000e+02	3.780000e+02	
25%	0.000000	21.000000	1.093475e+04	1.098175e+04	1.106625e+04	
50%	1.000000	43.000000	2.840700e+04	2.851500e+04	2.879250e+04	
75%	2.000000	65.000000	6.280075e+04	6.280300e+04	6.348950e+04	
max	2.000000	999.000000	3.075280e+08	3.081032e+08	3.104266e+08	

2.2 Population by age and sex correlations

In [40]:

```
pop_agesex_sub1.corr()
```

Out[40]:

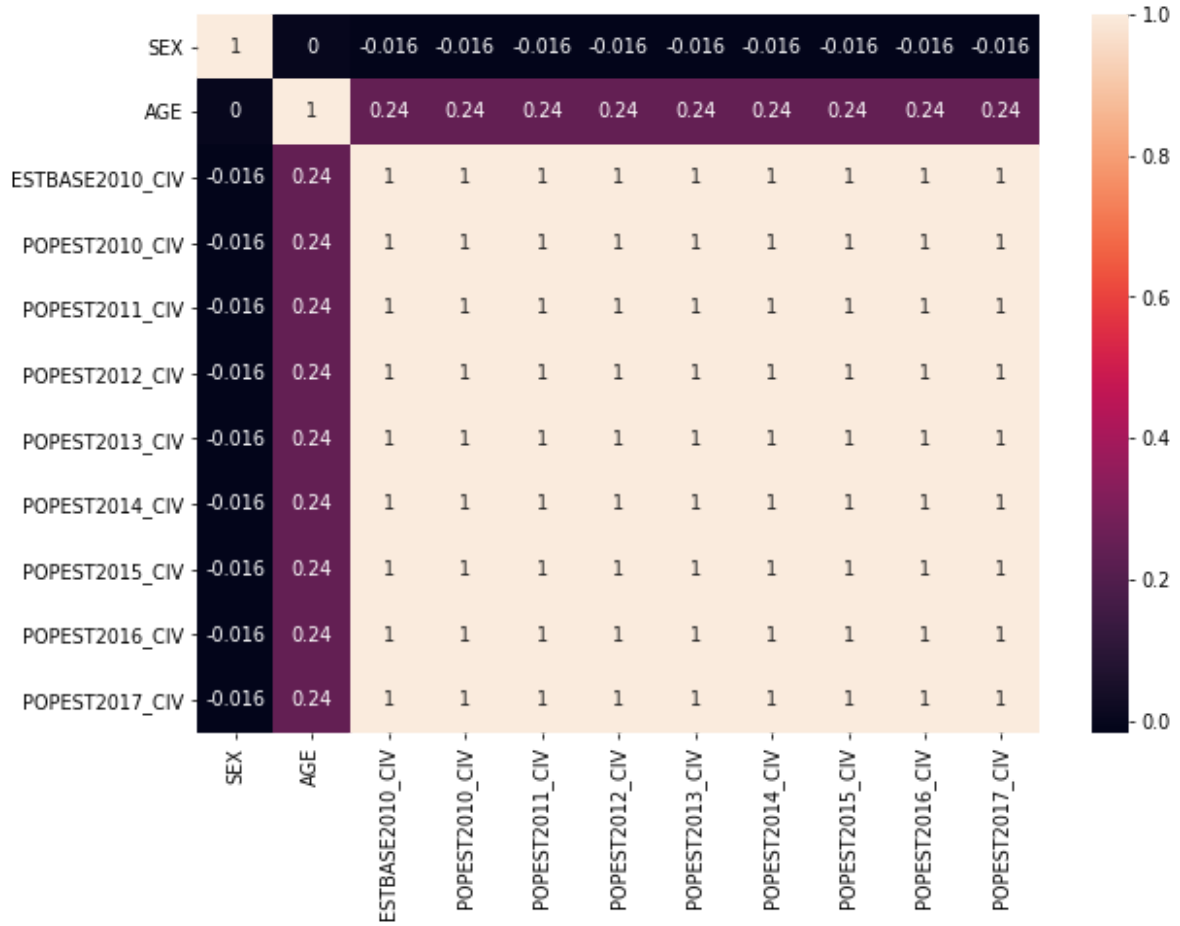
	SEX	AGE	ESTBASE2010_CIV	POPEST2010_CIV	POPEST2011_CIV
SEX	1.000000	0.000000	-0.016391	-0.016392	-0.016395
AGE	0.000000	1.000000	0.241865	0.241876	0.241915
ESTBASE2010_CIV	-0.016391	0.241865	1.000000	1.000000	0.999989
POPEST2010_CIV	-0.016392	0.241876	1.000000	1.000000	0.999991
POPEST2011_CIV	-0.016395	0.241915	0.999989	0.999991	1.000000
POPEST2012_CIV	-0.016400	0.241959	0.999979	0.999981	0.999991
POPEST2013_CIV	-0.016404	0.242000	0.999967	0.999969	0.999981
POPEST2014_CIV	-0.016409	0.242038	0.999953	0.999956	0.999969
POPEST2015_CIV	-0.016415	0.242072	0.999936	0.999940	0.999956
POPEST2016_CIV	-0.016418	0.242114	0.999915	0.999920	0.999939
POPEST2017_CIV	-0.016422	0.242161	0.999895	0.999899	0.999921

In [39]:

```
#The matrix of correlation
fig = plt.figure(figsize=(10,7), dpi=70)
sns.heatmap(pop_agesex_sub1.corr(), annot=True)
```

Out[39]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x268d3622cf8>
```



3. Statistics Population and Race

In [93]:

```
pop_racesex.head()
```

Out[93]:

ME	SEX	ORIGIN	RACE	AGE	CENSUS2010POP	ESTIMATESBASE2010	POPESTIMATE2010	PC
ama	0	0	1	0	37991	37991	37794	
ama	0	0	1	1	38150	38150	38035	
ama	0	0	1	2	39738	39738	39576	
ama	0	0	1	3	39827	39828	39808	
ama	0	0	1	4	39353	39353	39495	

In [45]:

```
pop_racesex.columns
```

Out[45]:

```
Index(['STATE_ABR', 'SUMLEV', 'REGION', 'DIVISION', 'STATE', 'NAME', 'SEX',
       'ORIGIN', 'RACE', 'AGE', 'CENSUS2010POP', 'ESTIMATESBASE2010',
       'POPESTIMATE2010', 'POPESTIMATE2011', 'POPESTIMATE2012',
       'POPESTIMATE2013', 'POPESTIMATE2014', 'POPESTIMATE2015',
       'POPESTIMATE2016', 'POPESTIMATE2017'],
      dtype='object')
```

In [50]:

```
#Subset of dataset
pop_racesex_sub1=pop_racesex[['NAME', 'SEX', "ORIGIN", "RACE", 'AGE', 'CENSUS2010POP', 'EST
                             'POPESTIMATE2010', 'POPESTIMATE2011', 'POPESTIMATE2012',
                              'POPESTIMATE2013', 'POPESTIMATE2014', 'POPESTIMATE2015',
                             'POPESTIMATE2016', 'POPESTIMATE2017']]

#Setting index
pop_racesex_sub1=pop_racesex_sub1.set_index("NAME")
```

3.1 Population by race and sex basic statistics

In [52]:

```
pop_racesex_sub1.describe()
```

Out[52]:

	SEX	ORIGIN	RACE	AGE	CENSUS2010POP	ESTIMA
count	236844.000000	236844.000000	236844.000000	236844.000000	236844.000000	2
mean	1.000000	1.000000	3.500000	42.500000	5214.327372	
std	0.816498	0.816498	1.707829	24.824436	18111.535290	
min	0.000000	0.000000	1.000000	0.000000	0.000000	
25%	0.000000	0.000000	2.000000	21.000000	29.000000	
50%	1.000000	1.000000	3.500000	42.500000	200.000000	
75%	2.000000	2.000000	5.000000	64.000000	1767.000000	
max	2.000000	2.000000	6.000000	85.000000	499596.000000	4

3.2 Population by age and sex correlations

In [54]:

```
pop_racesex_sub1.corr()
```

Out[54]:

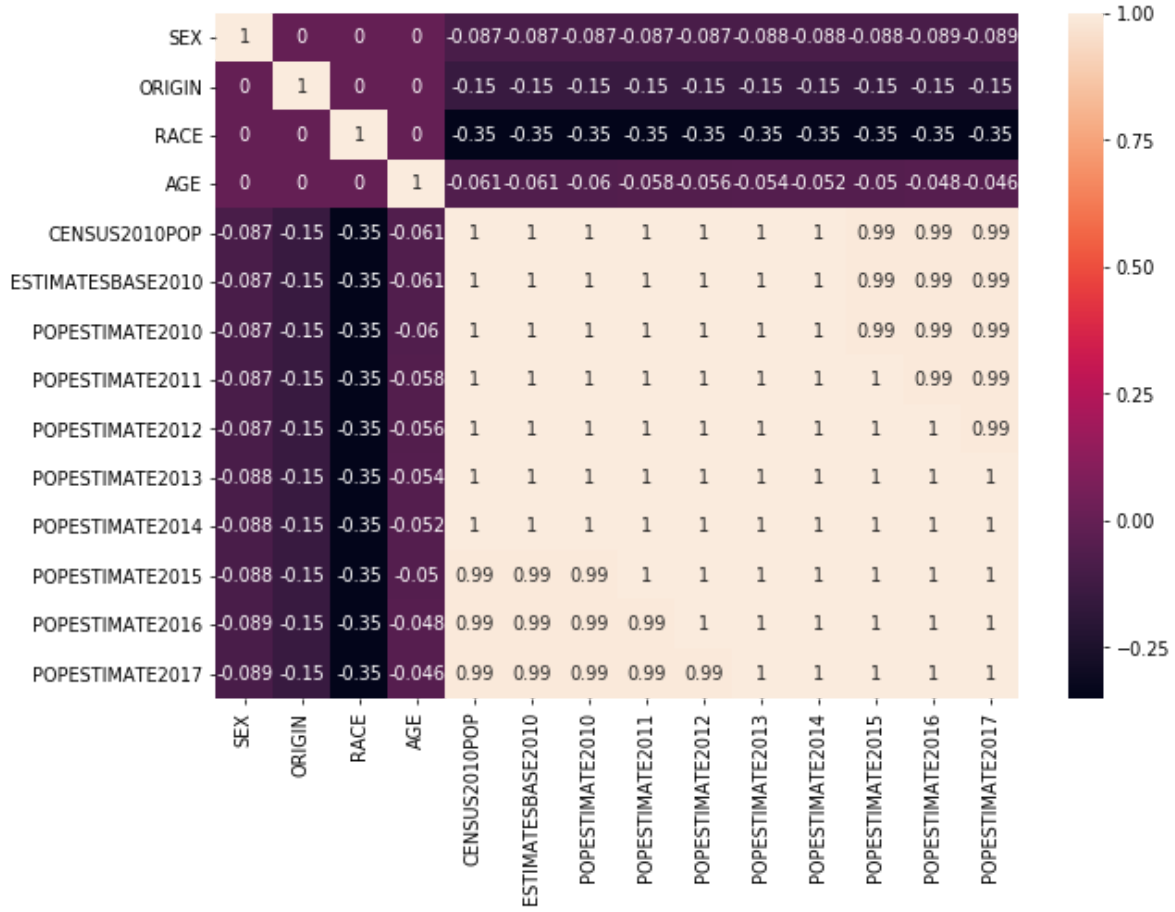
POPESTIMATE2010	POPESTIMATE2011	POPESTIMATE2012	POPESTIMATE2013	POPESTIMATE20
-0.086745	-0.087028	-0.087335	-0.087664	-0.0879
-0.147492	-0.147504	-0.147561	-0.147663	-0.1477
-0.350015	-0.349902	-0.349795	-0.349774	-0.3496
-0.060481	-0.058446	-0.056304	-0.054284	-0.0522
0.999943	0.998744	0.997671	0.996506	0.9953
0.999943	0.998744	0.997672	0.996506	0.9953
1.000000	0.998997	0.997930	0.996798	0.9956
0.998997	1.000000	0.999012	0.997970	0.9968
0.997930	0.999012	1.000000	0.999038	0.9980
0.996798	0.997970	0.999038	1.000000	0.9990
0.995673	0.996838	0.998008	0.999053	1.0000
0.994204	0.995711	0.996882	0.998026	0.9990
0.992421	0.994265	0.995772	0.996911	0.9980
0.990586	0.992542	0.994379	0.995835	0.9969

In [55]:

```
#The matrix of correlation
fig = plt.figure(figsize=(10,7), dpi=70)
sns.heatmap(pop_racesex_sub1.corr(), annot=True)
```

Out[55]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x268d42adf60>
```



4. Statistics Flu by Week

In [56]:

```
flu_week.head()
```

Out[56]:

:GION TYPE	REGION	YEAR	WEEK	% WEIGHTED ILI	%UNWEIGHTED ILI	AGE 0-4	AGE 25-49	AGE 25-64	AGE 5-24	AGE 50-64	AGE 65
States	Alabama	2010	40	X	2.13477	X	X	X	X	X	X
States	Alaska	2010	40	X	0.875146	X	X	X	X	X	X
States	Arizona	2010	40	X	0.674721	X	X	X	X	X	X
States	Arkansas	2010	40	X	0.696056	X	X	X	X	X	X
States	California	2010	40	X	1.95412	X	X	X	X	X	X

In [75]:

```
#Transforming to numeric
flu_week.ILITOTAL=pd.to_numeric(flu_week.ILITOTAL, errors="coerce")
flu_week["NUM. OF PROVIDERS"]=pd.to_numeric(flu_week["NUM. OF PROVIDERS"], errors="coerce")
flu_week["TOTAL PATIENTS"]=pd.to_numeric(flu_week["TOTAL PATIENTS"], errors="coerce")
```

In [77]:

```
flu_week.dtypes
```

Out[77]:

```
REGION TYPE
object
REGION
object
YEAR
int64
WEEK
int64
% WEIGHTED ILI
object
%UNWEIGHTED ILI
object
AGE 0-4
object
AGE 25-49
object
AGE 25-64
object
AGE 5-24
object
AGE 50-64
object
AGE 65
object
ILITOTAL
float64
NUM. OF PROVIDERS
float64
TOTAL PATIENTS
float64
PERCENTAGE OF VISITS FOR INFLUENZA-LIKE-ILLNESS REPORTED BY SENTINEL PROVIDE
RS    float64
Unnamed: 16
float64
dtype: object
```

4.1 Statistics Flu by Week

In [79]:

```
flu_week.describe()
```

Out[79]:

	YEAR	WEEK	ILITOTAL	NUM. OF PROVIDERS	TOTAL PATIENTS	PERCENTAGE OF VISITS FOR INFLUENZA-LIKE-ILLNESS REPORTED BY SENTINEL PROVIDERS
count	23768.000000	23768.000000	23324.000000	23324.000000	23324.000000	0.0
mean	2014.535342	26.531807	300.248328	33.008832	14203.332190	NaN
std	2.473321	15.367787	620.131337	30.265550	19311.350886	NaN
min	2010.000000	1.000000	0.000000	0.000000	0.000000	NaN
25%	2012.000000	13.000000	21.000000	13.000000	3129.000000	NaN
50%	2015.000000	26.500000	86.000000	23.000000	6445.000000	NaN
75%	2017.000000	40.000000	291.000000	45.000000	16936.000000	NaN
max	2019.000000	53.000000	11452.000000	214.000000	121056.000000	NaN

In [80]:

```
flu_week.corr()
```

Out[80]:

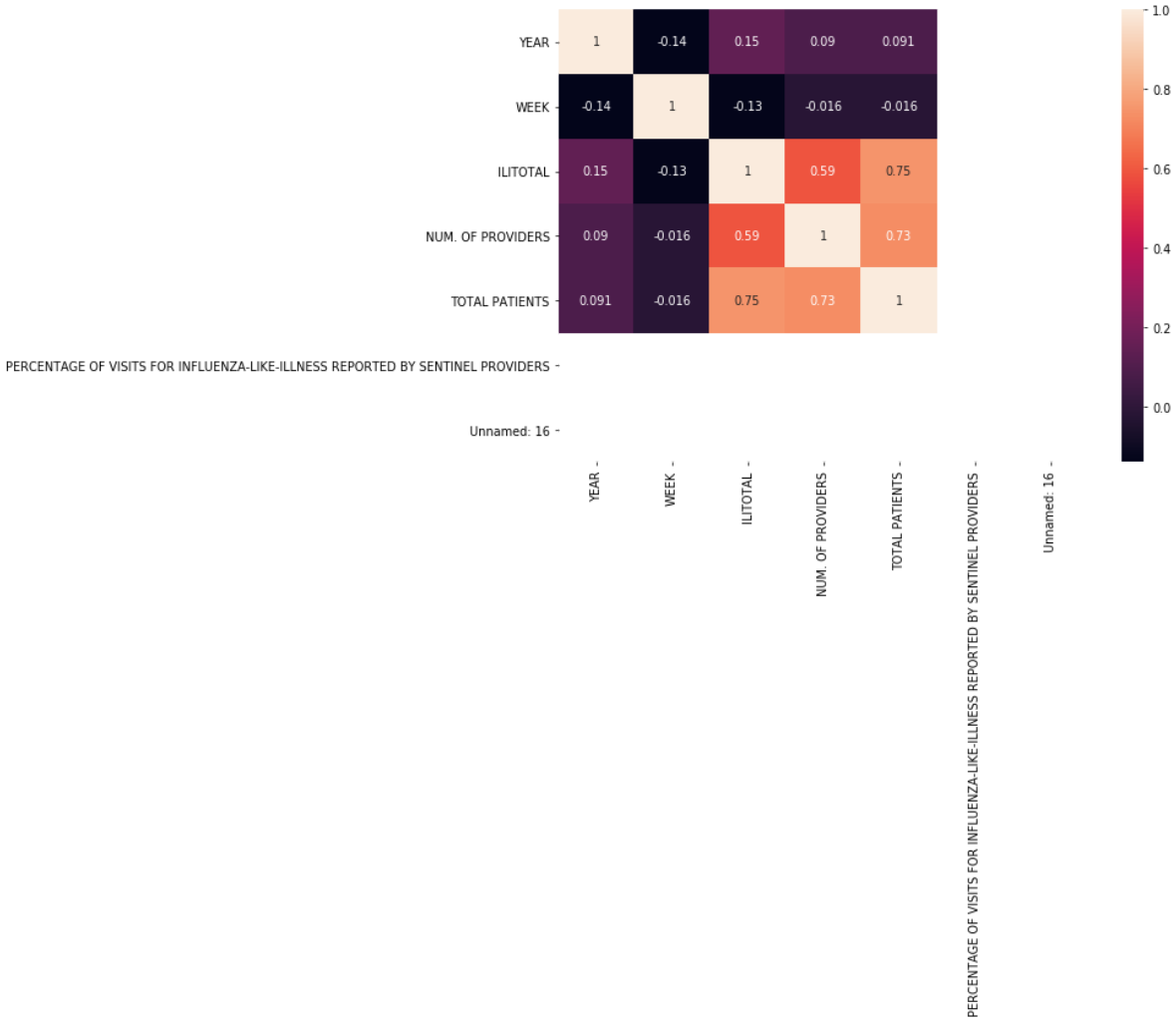
	YEAR	WEEK	ILITOTAL	NUM. OF PROVIDERS	TOTAL PATIENTS	PERCENTAGE OF VISITS FOR INFLUENZA-LIKE-ILLNESS REPORTED BY SENTINEL PROVIDERS	Unname
YEAR	1.000000	-0.139441	0.148271	0.090003	0.091220	NaN	Na
WEEK	-0.139441	1.000000	-0.131352	-0.015669	-0.016282	NaN	Na
ILITOTAL	0.148271	-0.131352	1.000000	0.594892	0.749716	NaN	Na
NUM. OF PROVIDERS	0.090003	-0.015669	0.594892	1.000000	0.730400	NaN	Na
TOTAL PATIENTS	0.091220	-0.016282	0.749716	0.730400	1.000000	NaN	Na
PERCENTAGE OF VISITS FOR INFLUENZA-LIKE-ILLNESS REPORTED BY SENTINEL PROVIDERS	NaN	NaN	NaN	NaN	NaN	NaN	Na
Unnamed: 16	NaN	NaN	NaN	NaN	NaN	NaN	Na

In [81]:

```
#The matrix of correlation
fig = plt.figure(figsize=(10,7), dpi=70)
sns.heatmap(flu_week.corr(), annot=True)
```

Out[81]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x268d7e51898>
```



In [89]:

```
flu_week.YEAR.unique()
```

Out[89]:

```
array([2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019],
      dtype=int64)
```

5. Statistics Flu by Season

In [86]:

```
flu_season.head()
```

Out[86]:

	STATENAME	URL	WEBSITE	ACTIVITY LEVEL	ACTI LE LA
0	Alaska	http://dhss.alaska.gov/dph/Epi/id/Pages/influe...	Influenza Surveillance Report	Level 1	Mi
1	Arizona	http://www.azdhs.gov/phs/oids/epi/flu/index.htm	Influenza & RSV Surveillance	Level 1	Mi
2	Arkansas	http://www.healthy.arkansas.gov/programs-servi...	Communicable Disease and Immunizations	Level 6	Mod
3	California	https://www.cdph.ca.gov/Programs/CID/DCDC/Page...	Influenza (Flu)	Level 1	Mi
4	Colorado	https://www.colorado.gov/pacific/cdphe/influenza	Influenza Surveillance	Level 1	Mi

Auxiliar table to join tables and perform statistical analysis

5. Joint Analysis

In [95]:

```
pop_agesex.NAME.unique()
```

Out[95]:

```
array(['United States', 'Alabama', 'Alaska', 'Arizona', 'Arkansas',
       'California', 'Colorado', 'Connecticut', 'Delaware',
       'District of Columbia', 'Florida', 'Georgia', 'Hawaii', 'Idaho',
       'Illinois', 'Indiana', 'Iowa', 'Kansas', 'Kentucky', 'Louisiana',
       'Maine', 'Maryland', 'Massachusetts', 'Michigan', 'Minnesota',
       'Mississippi', 'Missouri', 'Montana', 'Nebraska', 'Nevada',
       'New Hampshire', 'New Jersey', 'New Mexico', 'New York',
       'North Carolina', 'North Dakota', 'Ohio', 'Oklahoma', 'Oregon',
       'Pennsylvania', 'Rhode Island', 'South Carolina', 'South Dakota',
       'Tennessee', 'Texas', 'Utah', 'Vermont', 'Virginia', 'Washington',
       'West Virginia', 'Wisconsin', 'Wyoming'], dtype=object)
```

In [87]:

```
flu_week.REGION.unique()
```

Out[87]:

```
array(['Alabama', 'Alaska', 'Arizona', 'Arkansas', 'California',
       'Colorado', 'Connecticut', 'Delaware', 'District of Columbia',
       'Florida', 'Georgia', 'Hawaii', 'Idaho', 'Illinois', 'Indiana',
       'Iowa', 'Kansas', 'Kentucky', 'Louisiana', 'Maine', 'Maryland',
       'Massachusetts', 'Michigan', 'Minnesota', 'Mississippi',
       'Missouri', 'Montana', 'Nebraska', 'Nevada', 'New Hampshire',
       'New Jersey', 'New Mexico', 'New York', 'North Carolina',
       'North Dakota', 'Ohio', 'Oklahoma', 'Oregon', 'Pennsylvania',
       'Rhode Island', 'South Carolina', 'South Dakota', 'Tennessee',
       'Texas', 'Utah', 'Vermont', 'Virginia', 'Washington',
       'West Virginia', 'Wisconsin', 'Wyoming', 'New York City',
       'Virgin Islands', 'Puerto Rico'], dtype=object)
```

In [105]:

```
pop_racesex2=pop_racesex.groupby("NAME").sum()
pop_racesex2
```

Out[105]:

NAME	SUMLEV	REGION	DIVISION	STATE	SEX	ORIGIN	RACE	AGE	CENSUS2
Alabama	185760	13932	27864	4644	4644	4644	16254	197370	1
Alaska	185760	18576	41796	9288	4644	4644	16254	197370	
Arizona	185760	18576	37152	18576	4644	4644	16254	197370	2
Arkansas	185760	13932	32508	23220	4644	4644	16254	197370	1
California	185760	18576	41796	27864	4644	4644	16254	197370	14
Colorado	185760	18576	37152	37152	4644	4644	16254	197370	2
Connecticut	185760	4644	4644	41796	4644	4644	16254	197370	1
Delaware	185760	13932	23220	46440	4644	4644	16254	197370	
District of Columbia	185760	13932	23220	51084	4644	4644	16254	197370	
Florida	185760	13932	23220	55728	4644	4644	16254	197370	7
Georgia	185760	13932	23220	60372	4644	4644	16254	197370	3
Hawaii	185760	18576	41796	69660	4644	4644	16254	197370	
Idaho	185760	18576	37152	74304	4644	4644	16254	197370	
Illinois	185760	9288	13932	78948	4644	4644	16254	197370	5
Indiana	185760	9288	13932	83592	4644	4644	16254	197370	2
Iowa	185760	9288	18576	88236	4644	4644	16254	197370	1
Kansas	185760	9288	18576	92880	4644	4644	16254	197370	1
Kentucky	185760	13932	27864	97524	4644	4644	16254	197370	1
Louisiana	185760	13932	32508	102168	4644	4644	16254	197370	1
Maine	185760	4644	4644	106812	4644	4644	16254	197370	
Maryland	185760	13932	23220	111456	4644	4644	16254	197370	2
Massachusetts	185760	4644	4644	116100	4644	4644	16254	197370	2
Michigan	185760	9288	13932	120744	4644	4644	16254	197370	3
Minnesota	185760	9288	18576	125388	4644	4644	16254	197370	2
Mississippi	185760	13932	27864	130032	4644	4644	16254	197370	1
Missouri	185760	9288	18576	134676	4644	4644	16254	197370	2
Montana	185760	18576	37152	139320	4644	4644	16254	197370	
Nebraska	185760	9288	18576	143964	4644	4644	16254	197370	
Nevada	185760	18576	37152	148608	4644	4644	16254	197370	1
New Hampshire	185760	4644	4644	153252	4644	4644	16254	197370	
New Jersey	185760	4644	9288	157896	4644	4644	16254	197370	3
New Mexico	185760	18576	37152	162540	4644	4644	16254	197370	

NAME	SUMLEV	REGION	DIVISION	STATE	SEX	ORIGIN	RACE	AGE	CENSUS2
New York	185760	4644	9288	167184	4644	4644	16254	197370	7
North Carolina	185760	13932	23220	171828	4644	4644	16254	197370	3
North Dakota	185760	9288	18576	176472	4644	4644	16254	197370	
Ohio	185760	9288	13932	181116	4644	4644	16254	197370	4
Oklahoma	185760	13932	32508	185760	4644	4644	16254	197370	1
Oregon	185760	18576	41796	190404	4644	4644	16254	197370	1
Pennsylvania	185760	4644	9288	195048	4644	4644	16254	197370	5
Rhode Island	185760	4644	4644	204336	4644	4644	16254	197370	
South Carolina	185760	13932	23220	208980	4644	4644	16254	197370	1
South Dakota	185760	9288	18576	213624	4644	4644	16254	197370	
Tennessee	185760	13932	27864	218268	4644	4644	16254	197370	2
Texas	185760	13932	32508	222912	4644	4644	16254	197370	10
Utah	185760	18576	37152	227556	4644	4644	16254	197370	1
Vermont	185760	4644	4644	232200	4644	4644	16254	197370	
Virginia	185760	13932	23220	236844	4644	4644	16254	197370	3
Washington	185760	18576	41796	246132	4644	4644	16254	197370	2
West Virginia	185760	13932	23220	250776	4644	4644	16254	197370	
Wisconsin	185760	9288	13932	255420	4644	4644	16254	197370	2
Wyoming	185760	18576	37152	260064	4644	4644	16254	197370	

In [ ]:

```
### Notes: Check stack and unstack table

#http://www.datasciencemadesimple.com/reshape-using-stack-unstack-function-pandas-python/
```

In [ ]:

```
### Next Steps..............
```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: