# BANA Assignment 1

## Team Members:


## Aaron Liu

## Pranav Belmannu

## Maithilee Nagesh Kulkarni

## Gautam Mohan Babu


## In this project we are cleaning, analysing and visualizing the data on causes of death over the years from 1999 to 2016 in the United States of America. The data has been cleaned to accomodate 50 states and the causes of deaths along with their statistical data. This data was further analysed using python.

### 1.) Import important packages like pandas and numpy

In [16]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

### 2.) Store your directory and file names in a variable for code reusability

In [17]:
```
location='C://Downloads//'
file1='NCHS_-_Leading_Causes_of_Death__United_States.csv'
file2='nst-est2018-01.xlsx'
```

### 3.) Reading the file into data frames

In [18]:
```
df1=pd.read_csv(location+file1)
df2=pd.read_excel(location+file2,header=None)
```

### 4.) Exploring and viewing the data

In [19]:
```
df1.head()
```

Out[19]:

| | Year | 113 Cause Name | Cause Name | State | Deaths | Age-adjusted Death Rate |
|---|---|---|---|---|---|---|
| **0** | 2012 | Nephritis, nephrotic syndrome and nephrosis (N... | Kidney disease | Vermont | 21 | 2.6 |
| **1** | 2016 | Nephritis, nephrotic syndrome and nephrosis (N... | Kidney disease | Vermont | 30 | 3.7 |
| **2** | 2013 | Nephritis, nephrotic syndrome and nephrosis (N... | Kidney disease | Vermont | 30 | 3.8 |
| **3** | 2000 | Intentional self-harm (suicide) (*U03,X60-X84,... | Suicide | District of Columbia | 23 | 3.8 |
| **4** | 2014 | Nephritis, nephrotic syndrome and nephrosis (N... | Kidney disease | Arizona | 325 | 4.1 |

In [20]:
```
df2.head()
```

Out[20]:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | table with row headers in column A and column ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **1** | Table 1. Annual Estimates of the Resident Popu... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **2** | Geographic Area | 2010-04-01 00:00:00 | NaN | Population Estimate (as of July 1) | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **3** | NaN | Census | Estimates Base | 2010 | 2011.0 | 2012.0 | 2013.0 | 2014.0 | 2015.0 | 2016.0 | 2017.0 |
| **4** | United States | 308745538 | 308758105 | 309326085 | 311580009.0 | 313874218.0 | 316057727.0 | 318386421.0 | 320742673.0 | 323071342.0 | 325147121.0 |

In [21]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10296 entries, 0 to 10295
Data columns (total 6 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   Year                   10296 non-null  int64  
 1   113 Cause Name         10296 non-null  object 
 2   Cause Name             10296 non-null  object 
 3   State                  10296 non-null  object 
 4   Deaths                 10296 non-null  int64  
 5   Age-adjusted Death Rate  10296 non-null  float64
dtypes: float64(1), int64(2), object(3)
memory usage: 482.8+ KB
```

In [22]:
```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 67 entries, 0 to 66
Data columns (total 12 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   0       65 non-null     object 
 1   1       59 non-null     object 
 2   2       58 non-null     object 
 3   3       59 non-null     object 
 4   4       58 non-null     float64
 5   5       58 non-null     float64
 6   6       58 non-null     float64
 7   7       58 non-null     float64
 8   8       58 non-null     float64
 9   9       58 non-null     float64
 10  10      58 non-null     float64
 11  11      58 non-null     float64
dtypes: float64(8), object(4)
memory usage: 6.4+ KB
```

# Question 1 Sub Question 1

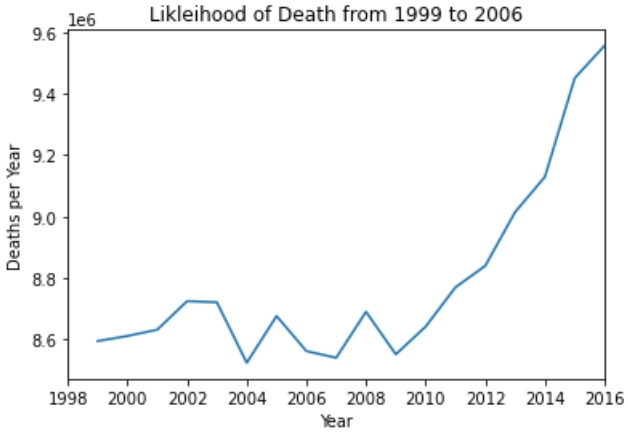
## Are Americans facing increasing, decreasing, or steady likelihood of death?

In [23]:
```
df1.head()
# Group by the year and sum all the deaths for each year
df1_likelihood=df1.groupby('Year')['Deaths'].agg('sum')

df1_likelihood
```

Out[23]:Year
1999 8594450
2000 8611090
2001 8631566
2002 8724520
2003 8720806
2004 8523496
2005 8675996
2006 8561880
2007 8540026
2008 8689930
2009 8551328
2010 8641568
2011 8769558
2012 8839734
2013 9014608
2014 9129652
2015 9451294
2016 9556734
Name: Deaths, dtype: int64

In [24]:
```
# Plot the data
plt.figure()
plt.ylabel("Deaths per Year")
plt.xlim(1998,2016)
plt.xlabel("Year")
plt.title("Likleihood of Death from 1999 to 2016")
plt.plot(df1_likelihood.index,df1_likelihood.values)
plt.show()
```




The above plot shows that the number of deaths per year has been fluctuating but gradually increased from 2009 to 2016. This confirms that Americans are facing increasing likelihood of deaths post 2010

Further Questions that can be asked:

1. Is the likelyhood ratio remained same accross states?
2. Are there are specific causes that caused gradual increase in deaths?
3. What causes have been prominent in the increase of deaths from 1999?

# Question 1 Sub Question 2

## What are top 4 leading causes of death?

In [10]:
```
### Get the Years column sorted
df1_sorted_by_year = df1.sort_values(by=['Year'])
df1_sorted_by_year # 10296 rows
### Get all the unique causes: 11 Unique Causes (936 each)
df1_sorted_by_year.iloc[0:, 1].value_counts()
"""
All Causes                                                      936
Alzheimer's disease (G30)                                       936
Chronic lower respiratory diseases (J40-J47)                    936
Nephritis, nephrotic syndrome and nephrosis (N00-N07,N17-N19,N25-N27)    936
Malignant neoplasms (C00-C97)                                   936
Accidents (unintentional injuries) (V01-X59,Y85-Y86)            936
Diabetes mellitus (E10-E14)                                     936
Cerebrovascular diseases (I60-I69)                              936
Influenza and pneumonia (J09-J18)                               936
Intentional self-harm (suicide) (*U03,X60-X84,Y87.0)            936
Diseases of heart (I00-I09,I11,I13,I20-I51)                     936
Name: 113 Cause Name, dtype: int64
"""
## Display only these three columns
df1_causes_and_death = df1_sorted_by_year[['113 Cause Name', 'Cause Name', 'Deaths']]
```

```
# df1_causes_and_death

#########################################################################
### The Below was used to help understand one type of cause of death
## Find the unique count of the causes
# df1_causes_and_death.iloc[0:, 1].value_counts()
## Get the deaths for a specific cause name
# clrd_deaths = df1_causes_and_death[df1_causes_and_death.iloc[0:,1].isin(['CLRD'])]
# clrd_deaths
## Get the cumulative count: 4869452 Deaths by CLRD
# clrd_deaths.iloc[0:,2].agg('sum')
#########################################################################

## Use pivot table to help aggregate all the causes and summing their deaths
total_deaths_per_causes = pd.pivot_table(df1_causes_and_death,
                                         values='Deaths',
                                         index=['Cause Name'],
                                         #columns=['113 Cause Name'],
                                         aggfunc=np.sum)
#total_deaths_per_causes
## Sort the pivot table in descending order
deaths_descending = total_deaths_per_causes.sort_values(by=['Deaths'], ascending=False)

# deaths_descending
## Display the top 4
top_4_deaths = deaths_descending.iloc[1:5]
top_4_deaths.values

## Alternative Method to getting the 4 leading causes of death
cause_of_death = df1_causes_and_death.groupby('Cause Name')
sum_cause_of_death = cause_of_death['Deaths'].agg('sum')
top_4_deaths = sum_cause_of_death.sort_values(ascending=False).iloc[1:5]
top_4_deaths
```

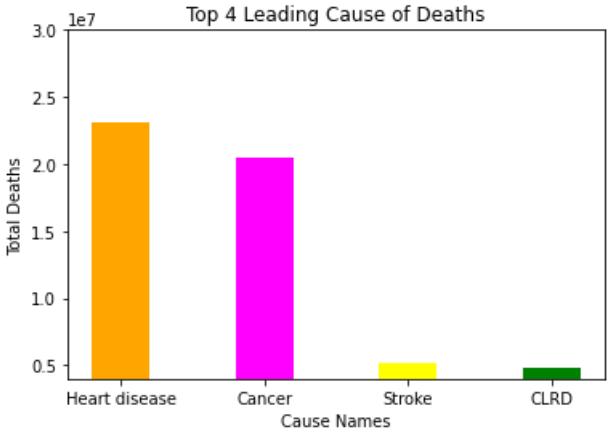
Out[10]:
```
Cause Name
Heart disease    23150366
Cancer           20489072
Stroke           5160280
CLRD              4869452
Name: Deaths, dtype: int64
```

The above table shows that the top four leading causes of deaths for Americans are Heart Diseases, cancer, Stroke and CLRD.

## Plotting this for variation purposes

In [11]:
```
# Plot the data

plt.figure()
plt.ylabel("Total Deaths")
plt.xlabel("Cause Names")
plt.ylim(4000000,30000000)
plt.title("Top 4 Leading Cause of Deaths")
plt.bar(top_4_deaths.index, top_4_deaths.values, width=0.4, color=['orange', 'magenta', 'yellow', 'green'])
plt.show()
```




# Question 1 Sub Question 3

## Do individual states show the same four leading causes of death??

In [12]:
```
df1.head()
##Picking up data only for deaths,states and cause name
df1_deaths_states_all=df1.loc[:,['Deaths','State','Cause Name']]

###Sort the data with death
```

```
df1_deaths_states_all=df1_deaths_states_all.sort_values(by=['State','Cause Name'],ascending=False)
df1_deaths_states_all=df1_deaths_states_all[df1_deaths_states_all['State']!='United States']
df1_deaths_states_all=df1_deaths_states_all[df1_deaths_states_all['Cause Name']!='All causes']
df1_deaths_states_all=df1_deaths_states_all.groupby(['State','Cause Name'],as_index=False)['Deaths'].agg(sum)
pd.set_option('display.max_rows', 530)
df1_deaths_states_all
df1_deaths_states_all = df1_deaths_states_all.sort_values(by=['State','Deaths'],ascending=False)
count = 4
curr_count = 0
state_dic = {}
for index, row in df1_deaths_states_all.iterrows():
    #print(row['State'], row['Cause Name'])
    if row['State'] in state_dic:
        if curr_count < count:
            state_dic[row['State']].append(row['Cause Name'])
            curr_count += 1
        else:
            continue
    else:
        state_dic[row['State']] = []
        curr_count = 0
state_dic
state_df = pd.DataFrame(state_dic)
state_df[state_df.isin(['Heart disease', 'Cancer', 'Stroke', 'CLRD'])]
```

Out[12]:

| | Wyoming | Wisconsin | West Virginia | Washington | Virginia | Vermont | Utah | Texas | Tennessee | South Dakota | ... | Florida | District of Columbia | Delaware | Connecticut |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | Cancer | Cancer | Cancer | Heart disease | Cancer | Heart disease | Cancer | Cancer | Cancer | Cancer | ... | Cancer | Cancer | Cancer | Cancer |
| **1** | CLRD | Stroke | CLRD | Stroke | Stroke | CLRD | NaN | Stroke | Stroke | Stroke | ... | CLRD | Stroke | CLRD | Stroke |
| **2** | NaN | NaN | NaN | CLRD | CLRD | NaN | Stroke | NaN | CLRD | CLRD | ... | Stroke | NaN | Stroke | CLRD |
| **3** | Stroke | CLRD | Stroke | NaN | NaN | Stroke | CLRD | CLRD | NaN | NaN | ... | NaN | NaN | NaN | NaN |

4 rows × 51 columns

As we can see from above table, All states have any three out of four leading causes of death in each state.

Further Questions can be asked are: what are individual four causes of death for each state? what position is the left out leading cause of death in each state?

# Question 1 Sub Question 4

## Are there year-by-year changes in the four leading causes of death nationwide?

In [13]:
```
df1.head()

df1_yearly_leading_causes=df1.loc[:,['Year','Cause Name','Deaths']]
df1_yearly_leading_causes=df1.groupby(['Cause Name','Year'],as_index=False)['Deaths'].agg(sum)
df1_yearly_leading_causes=pd.DataFrame(df1_yearly_leading_causes)
df1_yearly_leading_causes=df1_yearly_leading_causes[df1_yearly_leading_causes['Cause Name'].isin(top_4_deaths.index)]
df1_yearly_leading_causes
```

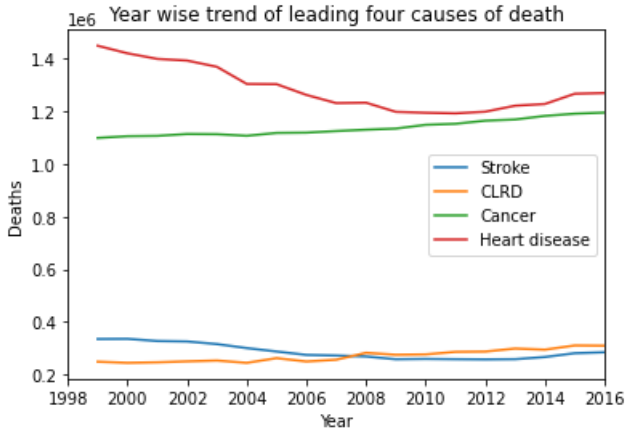
Out[13]:

| | Cause Name | Year | Deaths |
|---|---|---|---|
| **36** | CLRD | 1999 | 248362 |
| **37** | CLRD | 2000 | 244018 |
| **38** | CLRD | 2001 | 246026 |
| **39** | CLRD | 2002 | 249632 |
| **40** | CLRD | 2003 | 252764 |
| **41** | CLRD | 2004 | 243974 |
| **42** | CLRD | 2005 | 261866 |
| **43** | CLRD | 2006 | 249166 |
| **44** | CLRD | 2007 | 255848 |
| **45** | CLRD | 2008 | 282180 |
| **46** | CLRD | 2009 | 274706 |
| **47** | CLRD | 2010 | 276160 |
| **48** | CLRD | 2011 | 285886 |
| **49** | CLRD | 2012 | 286978 |
| **50** | CLRD | 2013 | 298410 |
| **51** | CLRD | 2014 | 294202 |

| | Cause Name | Year | Deaths |
|---|---|---|---|
| **52** | CLRD | 2015 | 310082 |
| **53** | CLRD | 2016 | 309192 |
| **54** | Cancer | 1999 | 1099676 |
| **55** | Cancer | 2000 | 1106182 |
| **56** | Cancer | 2001 | 1107536 |
| **57** | Cancer | 2002 | 1114542 |
| **58** | Cancer | 2003 | 1113804 |
| **59** | Cancer | 2004 | 1107776 |
| **60** | Cancer | 2005 | 1118624 |
| **61** | Cancer | 2006 | 1119776 |
| **62** | Cancer | 2007 | 1125750 |
| **63** | Cancer | 2008 | 1130938 |
| **64** | Cancer | 2009 | 1135256 |
| **65** | Cancer | 2010 | 1149486 |
| **66** | Cancer | 2011 | 1153382 |
| **67** | Cancer | 2012 | 1165246 |
| **68** | Cancer | 2013 | 1169762 |
| **69** | Cancer | 2014 | 1183400 |
| **70** | Cancer | 2015 | 1191860 |
| **71** | Cancer | 2016 | 1196076 |
| **90** | Heart disease | 1999 | 1450384 |
| **91** | Heart disease | 2000 | 1421520 |
| **92** | Heart disease | 2001 | 1400284 |
| **93** | Heart disease | 2002 | 1393894 |
| **94** | Heart disease | 2003 | 1370178 |
| **95** | Heart disease | 2004 | 1304972 |
| **96** | Heart disease | 2005 | 1304182 |
| **97** | Heart disease | 2006 | 1263272 |
| **98** | Heart disease | 2007 | 1232134 |
| **99** | Heart disease | 2008 | 1233656 |
| **100** | Heart disease | 2009 | 1198826 |
| **101** | Heart disease | 2010 | 1195378 |
| **102** | Heart disease | 2011 | 1193154 |
| **103** | Heart disease | 2012 | 1199422 |
| **104** | Heart disease | 2013 | 1222210 |
| **105** | Heart disease | 2014 | 1228696 |
| **106** | Heart disease | 2015 | 1267684 |
| **107** | Heart disease | 2016 | 1270520 |
| **144** | Stroke | 1999 | 334732 |
| **145** | Stroke | 2000 | 335322 |
| **146** | Stroke | 2001 | 327076 |
| **147** | Stroke | 2002 | 325344 |
| **148** | Stroke | 2003 | 315378 |
| **149** | Stroke | 2004 | 300148 |

| | Cause Name | Year | Deaths |
|---|---|---|---|
| 150 | Stroke | 2005 | 267158 |
| 151 | Stroke | 2006 | 274238 |
| 152 | Stroke | 2007 | 271904 |
| 153 | Stroke | 2008 | 268296 |
| 154 | Stroke | 2009 | 257684 |
| 155 | Stroke | 2010 | 258952 |
| 156 | Stroke | 2011 | 257864 |
| 157 | Stroke | 2012 | 257092 |
| 158 | Stroke | 2013 | 257956 |
| 159 | Stroke | 2014 | 266206 |
| 160 | Stroke | 2015 | 280646 |
| 161 | Stroke | 2016 | 284284 |

In [14]:
```
plt.figure()
plt.ylabel("Deaths")
plt.xlim(1998,2016)
plt.xlabel("Year")
plt.title("Year wise trend of leading four causes of death")
plt.plot(df1_yearly_leading_causes[df1_yearly_leading_causes['Cause Name']=='Stroke']['Year'],df1_yearly_leading_causes[df1_yearly_leading_cause
plt.plot(df1_yearly_leading_causes[df1_yearly_leading_causes['Cause Name']=='CLRD']['Year'],df1_yearly_leading_causes[df1_yearly_leading_cause
plt.plot(df1_yearly_leading_causes[df1_yearly_leading_causes['Cause Name']=='Cancer']['Year'],df1_yearly_leading_causes[df1_yearly_leading_caus
plt.plot(df1_yearly_leading_causes[df1_yearly_leading_causes['Cause Name']=='Heart disease']['Year'],df1_yearly_leading_causes[df1_yearly_leadin
plt.legend(loc='center right')
plt.show()
```




Here there are few observations to make:- 1.Cancer and CLRD are steadily increasing over the years 2.Both Heart diseases and strokes have come down slightly over the years

Questions can be asked are:- 1.Is the increase same accross all states? 2.Are there other causes that have increased the death count over the years? 3.Are there other causes that have decreased the death count over the years?

## Q2. Normalization of population or standardization data involves following procedures

### 1.) Drop first two cells for data cleaning purposes

### 2.) pick headers and add it back appropriately

### 3.) Remove census and estimates which cannot be used in further analysis

### 4.) drop NA values which would disrupt the operations

### 5.) bring all the datatypes into int because population cannot be in fractions

In [15]:
```
## Pull the column title "Geographic Area"
label = df2.iloc[2,0]

## Pull the remainder row containing the header
headers = df2.iloc[3:4]

## Pull the values within the rows
headers.values

## Create new column header array
newColumnHeaders = []

## Add the first value to it
newColumnHeaders.append(label)
```

```
## Add the remainder values to it
for x in headers.values[0]:
    if isinstance(x, float):
        if np.isnan(x):
            continue
        newColumnHeaders.append(int(x))
        continue
    newColumnHeaders.append(x)
newColumnHeaders
## Make the clean data frame with the values from United States to Puerto Rico
cleanDF = pd.DataFrame(df2.iloc[4:-5])
## Assign the new column headers
cleanDF.columns = newColumnHeaders
## Get rid of row number 60 as it was just an empty row
cleanDF = cleanDF.dropna()
cleanDF=cleanDF.drop(['Census','Estimates Base'],axis=1)
cleanDF.set_index('Geographic Area')
cleanDF.transpose()
cleanDF
state_col = cleanDF.iloc[0:,0]
#state_col['Geographic Area'].apply(lambda x: x[0].upper() + x[1:])

df_state = pd.DataFrame(state_col)
cleanDF = cleanDF.iloc[0:, 1:].astype(int)
df_state.merge(cleanDF, left_index=True, right_index=True)
```

Out[15]:

| | Geographic Area | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 | 2018 |
|---|---|---|---|---|---|---|---|---|---|---|
| 4 | United States | 309326085 | 311580009 | 313874218 | 316057727 | 318386421 | 320742673 | 323071342 | 325147121 | 327167434 |
| 5 | Northeast | 55380645 | 55600532 | 55776729 | 55907823 | 56015864 | 56047587 | 56058789 | 56072676 | 56111079 |
| 6 | Midwest | 66974749 | 67152631 | 67336937 | 67564135 | 67752238 | 67869139 | 67996917 | 68156035 | 68308744 |
| 7 | South | 114867066 | 116039399 | 117271075 | 118393244 | 119657737 | 121037542 | 122401186 | 123598424 | 124753948 |
| 8 | West | 72103625 | 72787447 | 73489477 | 74192525 | 74960582 | 75788405 | 76614450 | 77319986 | 77993663 |
| 9 | .Alabama | 4785448 | 4798834 | 4815564 | 4830460 | 4842481 | 4853160 | 4864745 | 4875120 | 4887871 |
| 10 | .Alaska | 713906 | 722038 | 730399 | 737045 | 736307 | 737547 | 741504 | 739786 | 737438 |
| 11 | .Arizona | 6407774 | 6473497 | 6556629 | 6634999 | 6733840 | 6833596 | 6945452 | 7048876 | 7171646 |
| 12 | .Arkansas | 2921978 | 2940407 | 2952109 | 2959549 | 2967726 | 2978407 | 2990410 | 3002997 | 3013825 |
| 13 | .California | 37320903 | 37641823 | 37960782 | 38280824 | 38625139 | 38953142 | 39209127 | 39399349 | 39557045 |
| 14 | .Colorado | 5048281 | 5121771 | 5193721 | 5270482 | 5351218 | 5452107 | 5540921 | 5615902 | 5695564 |
| 15 | .Connecticut | 3579125 | 3588023 | 3594395 | 3594915 | 3594783 | 3587509 | 3578674 | 3573880 | 3572665 |
| 16 | .Delaware | 899595 | 907316 | 915188 | 923638 | 932596 | 941413 | 949216 | 957078 | 967171 |
| 17 | .District of Columbia | 605085 | 619602 | 634725 | 650431 | 662513 | 675254 | 686575 | 695691 | 702455 |
| 18 | .Florida | 18845785 | 19093352 | 19326230 | 19563166 | 19860330 | 20224249 | 20629982 | 20976812 | 21299325 |
| 19 | .Georgia | 9711810 | 9801578 | 9901496 | 9973326 | 10069001 | 10181111 | 10304763 | 10413055 | 10519475 |
| 20 | .Hawaii | 1363963 | 1379252 | 1394905 | 1408453 | 1414862 | 1422484 | 1428105 | 1424203 | 1420491 |
| 21 | .Idaho | 1570773 | 1583828 | 1595441 | 1611530 | 1631479 | 1651523 | 1682930 | 1718904 | 1754208 |
| 22 | .Illinois | 12840762 | 12867291 | 12884119 | 12898269 | 12888962 | 12864342 | 12826895 | 12786196 | 12741080 |
| 23 | .Indiana | 6490436 | 6516045 | 6537640 | 6568367 | 6593533 | 6608296 | 6633344 | 6660082 | 6691878 |
| 24 | .Iowa | 3050767 | 3066054 | 3076097 | 3093078 | 3109504 | 3121460 | 3131785 | 3143637 | 3156145 |
| 25 | .Kansas | 2858213 | 2869035 | 2885361 | 2893510 | 2900896 | 2909502 | 2911263 | 2910689 | 2911505 |
| 26 | .Kentucky | 4348200 | 4369488 | 4386381 | 4404817 | 4414483 | 4425999 | 4438229 | 4453874 | 4468402 |
| 27 | .Louisiana | 4544532 | 4575184 | 4600814 | 4624577 | 4644204 | 4664851 | 4678215 | 4670818 | 4659978 |
| 28 | .Maine | 1327632 | 1328150 | 1327691 | 1328196 | 1330760 | 1328484 | 1331370 | 1335063 | 1338404 |
| 29 | .Maryland | 5788642 | 5838991 | 5887072 | 5923704 | 5958165 | 5986717 | 6004692 | 6024891 | 6042718 |
| 30 | .Massachusetts | 6566431 | 6613149 | 6663158 | 6713944 | 6763652 | 6795891 | 6826022 | 6863246 | 6902149 |
| 31 | .Michigan | 9877535 | 9881521 | 9896930 | 9913349 | 9930589 | 9932573 | 9951890 | 9976447 | 9995915 |
| 32 | .Minnesota | 5310843 | 5345668 | 5376550 | 5413693 | 5451522 | 5482503 | 5523409 | 5568155 | 5611179 |
| 33 | .Mississippi | 2970536 | 2978470 | 2983767 | 2988797 | 2990623 | 2988693 | 2988298 | 2989663 | 2986530 |
| 34 | .Missouri | 5995976 | 6009641 | 6024081 | 6040658 | 6056293 | 6071745 | 6087203 | 6108612 | 6126452 |
| 35 | .Montana | 990722 | 997221 | 1003754 | 1013564 | 1021891 | 1030503 | 1040863 | 1053090 | 1062305 |
| 36 | .Nebraska | 1829536 | 1840538 | 1853323 | 1865414 | 1879522 | 1891507 | 1905924 | 1917575 | 1929268 |
| 37 | .Nevada | 2702464 | 2712799 | 2744566 | 2776972 | 2819012 | 2868666 | 2919772 | 2972405 | 3034392 |
| 38 | .New Hampshire | 1316777 | 1319815 | 1323962 | 1326408 | 1333223 | 1336294 | 1342373 | 1349767 | 1356458 |
| 39 | .New Jersey | 8799624 | 8827783 | 8845483 | 8858362 | 8866780 | 8870869 | 8874516 | 8888543 | 8908520 |

| | Geographic Area | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 | 2018 |
|---|---|---|---|---|---|---|---|---|---|---|
| 40 | .New Mexico | 2064588 | 2080395 | 2087549 | 2092792 | 2090342 | 2090211 | 2092789 | 2093395 | 2095428 |
| 41 | .New York | 19400080 | 19498514 | 19574549 | 19628043 | 19656330 | 19661411 | 19641589 | 19590719 | 19542209 |
| 42 | .North Carolina | 9574293 | 9656754 | 9749123 | 9843599 | 9933944 | 10033079 | 10156679 | 10270800 | 10383620 |
| 43 | .North Dakota | 674710 | 685136 | 701116 | 721999 | 737382 | 754022 | 754353 | 755176 | 760077 |
| 44 | .Ohio | 11539327 | 11543463 | 11548369 | 11576576 | 11602973 | 11617850 | 11635003 | 11664129 | 11689442 |
| 45 | .Oklahoma | 3759632 | 3787821 | 3818600 | 3853205 | 3878367 | 3909831 | 3926769 | 3932640 | 3943079 |
| 46 | .Oregon | 3837532 | 3871728 | 3899118 | 3922908 | 3964106 | 4016918 | 4091404 | 4146592 | 4190713 |
| 47 | .Pennsylvania | 12711158 | 12744583 | 12766827 | 12776621 | 12789101 | 12785759 | 12783538 | 12790447 | 12807060 |
| 48 | .Rhode Island | 1053938 | 1053536 | 1054601 | 1055122 | 1056017 | 1056173 | 1057063 | 1056486 | 1057315 |
| 49 | .South Carolina | 4635656 | 4671422 | 4717112 | 4764153 | 4823793 | 4892253 | 4958235 | 5021219 | 5084127 |
| 50 | .South Dakota | 816165 | 823484 | 833496 | 842270 | 849088 | 853933 | 862890 | 873286 | 882235 |
| 51 | .Tennessee | 6355301 | 6397410 | 6451281 | 6493432 | 6540826 | 6590808 | 6645011 | 6708794 | 6770010 |
| 52 | .Texas | 25242679 | 25646227 | 26089620 | 26489464 | 26977142 | 27486814 | 27937492 | 28322717 | 28701845 |
| 53 | .Utah | 2775334 | 2814216 | 2853467 | 2897927 | 2937399 | 2982497 | 3042613 | 3103118 | 3161105 |
| 54 | .Vermont | 625880 | 626979 | 626063 | 626212 | 625218 | 625197 | 623644 | 624525 | 626299 |
| 55 | .Virginia | 8023680 | 8100469 | 8185229 | 8253053 | 8312076 | 8362907 | 8410946 | 8465207 | 8517685 |
| 56 | .Washington | 6742902 | 6821655 | 6892876 | 6962906 | 7052439 | 7163543 | 7294680 | 7425432 | 7535591 |
| 57 | .West Virginia | 1854214 | 1856074 | 1856764 | 1853873 | 1849467 | 1841996 | 1830929 | 1817048 | 1805832 |
| 58 | .Wisconsin | 5690479 | 5704755 | 5719855 | 5736952 | 5751974 | 5761406 | 5772958 | 5792051 | 5813568 |
| 59 | .Wyoming | 564483 | 567224 | 576270 | 582123 | 582548 | 585668 | 584290 | 578934 | 577737 |
| 61 | Puerto Rico | 3721525 | 3678732 | 3634488 | 3593077 | 3534874 | 3473166 | 3406495 | 3325001 | 3195153 |