> **Tip**: Welcome to the Investigate a Dataset project! You will find tips in quoted sections like this to help organize your approach to your investigation. Before submitting your project, it will be a good idea to go back through your report and remove these sections to make the presentation of your work as tidy as possible. First things first, you might want to double-click this Markdown cell and change the title so that it reflects your dataset and investigation.

Project: Investigate a Dataset (Replace this with something more specific!)

Table of Contents

Introduction

> **Tip**: In this section of the report, provide a brief introduction to the dataset you've selected for analysis. At the end of this section, describe the questions that you plan on exploring over the course of the report. Try to build your report around the analysis of at least one dependent variable and three independent variables.
>
> If you haven't yet selected and downloaded your data, make sure you do that first before coming back here. If you're not sure what questions to ask right now, then make sure you familiarize yourself with the variables and the dataset context for ideas of what to explore.

In [ ]:

```
# Use this cell to set up import statements for all of the pac
kages that you
#   plan to use.

# Remember to include a 'magic word' so that your visualizatio
ns are plotted
#   inline with the notebook. See this page for more:
#   http://ipython.readthedocs.io/en/stable/interactive/magics
.html
```

Data Wrangling

> **Tip**: In this section of the report, you will load in the data, check for cleanliness, and then trim and clean your dataset for analysis. Make sure that you document your steps carefully and justify your cleaning decisions.

General Properties

In [46]:

```
# Load your data and print out a few lines. Perform operations
to inspect data
#   types and look for instances of missing or possibly errant
data.
import pandas as pd
df_GunData = pd.read_excel('gun-data.xlsx')
df_Census_data = pd.read_csv('u.s.-census-data.csv')
import matplotlib.pyplot as plt
```

> **Tip**: You should *not* perform too many operations in each cell. Create cells freely to explore your data. One option that you can take with this project is to do a lot of explorations in an initial notebook. These don't have to be organized, but make sure you use enough comments to understand the purpose of each code cell. Then, after you're done with your analysis, create a duplicate notebook where you will trim the excess and organize your steps so that you have a flowing, cohesive report.
>
> **Tip**: Make sure that you keep your reader informed on the steps that you are taking in your investigation. Follow every code cell, or every set of related code cells, with a markdown cell to describe to the reader what was found in the preceding cell(s). Try to make it so that the reader can then understand what they will be seeing in the following cell(s).

Data Cleaning (Replace this with more specific notes!)

In [22]:

```
# After discussing the structure of the data and any problems that need to be
#   cleaned, perform those cleaning steps in the second part of this section.

#previw the data
df_GunData.head()
```

Out[22]:

	month	state	permit	permit_recheck	handgun	long_gun
0	2017-09	Alabama	16717.0	0.0	5734.0	6320.0
1	2017-09	Alaska	209.0	2.0	2320.0	2930.0
2	2017-09	Arizona	5069.0	382.0	11063.0	7946.0
3	2017-09	Arkansas	2935.0	632.0	4347.0	6063.0
4	2017-09	California	57839.0	0.0	37165.0	24581.0

5 rows × 27 columns

In [ ]:

```

```

In [24]:

```
df_GunData.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12485 entries, 0 to 12484
Data columns (total 27 columns):
month                        12485 non-null object
state                        12485 non-null object
permit                       12461 non-null float6
4
permit_recheck               1100 non-null float64
handgun                      12465 non-null float6
4
long_gun                     12466 non-null float6
4
other                        5500 non-null float64
multiple                     12485 non-null int64
admin                        12462 non-null float6
4
prepawn_handgun              10542 non-null float6
4
prepawn_long_gun             10540 non-null float6
4
prepawn_other                5115 non-null float64
redemption_handgun           10545 non-null float6
4
redemption_long_gun          10544 non-null float6
4
redemption_other             5115 non-null float64
returned_handgun             2200 non-null float64
returned_long_gun            2145 non-null float64
returned_other               1815 non-null float64
rentals_handgun              990 non-null float64
rentals_long_gun             825 non-null float64
private_sale_handgun         2750 non-null float64
private_sale_long_gun        2750 non-null float64
private_sale_other           2750 non-null float64
return_to_seller_handgun     2475 non-null float64
return_to_seller_long_gun    2750 non-null float64
return_to_seller_other       2255 non-null float64
totals                       12485 non-null int64
dtypes: float64(23), int64(2), object(2)
memory usage: 2.6+ MB
```

In [ ]:

In [ ]:

```
```

In [25]:

```
df_Census_data .head()
```

Out[25]:

	Fact	Fact Note	Alabama	Alaska	Arizona	Arkansas	Califo
0	Population estimates, July 1, 2016, (V2016)	NaN	4,863,300	741,894	6,931,071	2,988,248	39,250,
1	Population estimates base, April 1, 2010, (V2...	NaN	4,780,131	710,249	6,392,301	2,916,025	37,254,
2	Population, percent change - April 1, 2010 (es...	NaN	1.70%	4.50%	8.40%	2.50%	5.4
3	Population, Census, April 1, 2010	NaN	4,779,736	710,231	6,392,017	2,915,918	37,253,
4	Persons under 5 years, percent, July 1, 2016, ...	NaN	6.00%	7.30%	6.30%	6.40%	6.3

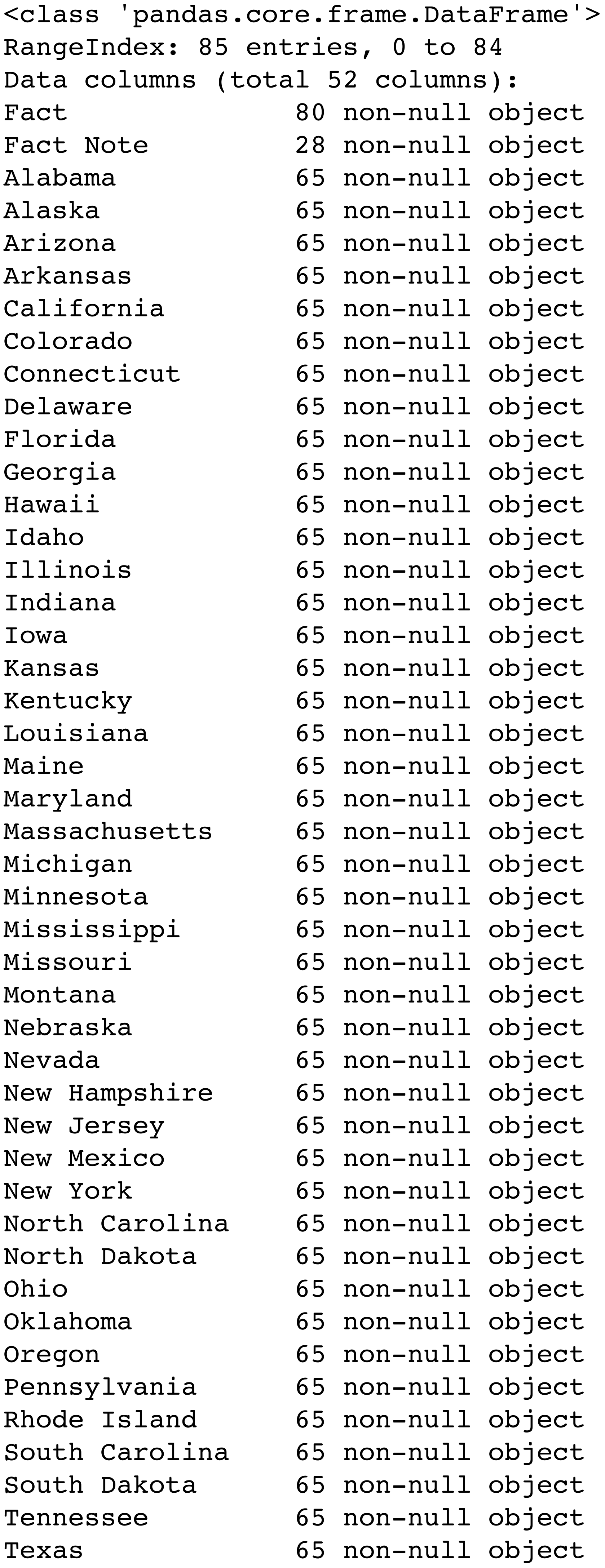
5 rows × 52 columns

In [ ]:

```
```

In [26]:

```
df_Census_data.info()
```



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 85 entries, 0 to 84
Data columns (total 52 columns):
Fact              80 non-null object
Fact Note         28 non-null object
Alabama           65 non-null object
Alaska            65 non-null object
Arizona           65 non-null object
Arkansas          65 non-null object
California        65 non-null object
Colorado          65 non-null object
Connecticut       65 non-null object
Delaware          65 non-null object
Florida           65 non-null object
Georgia           65 non-null object
Hawaii            65 non-null object
Idaho             65 non-null object
Illinois          65 non-null object
Indiana           65 non-null object
Iowa              65 non-null object
Kansas            65 non-null object
Kentucky          65 non-null object
Louisiana         65 non-null object
Maine             65 non-null object
Maryland          65 non-null object
Massachusetts     65 non-null object
Michigan          65 non-null object
Minnesota         65 non-null object
Mississippi       65 non-null object
Missouri          65 non-null object
Montana           65 non-null object
Nebraska          65 non-null object
Nevada            65 non-null object
New Hampshire     65 non-null object
New Jersey        65 non-null object
New Mexico        65 non-null object
New York          65 non-null object
North Carolina    65 non-null object
North Dakota      65 non-null object
Ohio              65 non-null object
Oklahoma          65 non-null object
Oregon            65 non-null object
Pennsylvania      65 non-null object
Rhode Island      65 non-null object
South Carolina    65 non-null object
South Dakota      65 non-null object
Tennessee         65 non-null object
Texas             65 non-null object
```

```
Utah                 65 non-null object
Vermont              65 non-null object
Virginia             65 non-null object
Washington           65 non-null object
West Virginia        65 non-null object
Wisconsin            65 non-null object
Wyoming              65 non-null object
dtypes: object(52)
memory usage: 34.6+ KB
```

In [ ]:

```

```

In [28]:

```
#The gun table is sorted by the date. I want to see the data using the state, so a groupby function is used as follows:
# All I need is the permits, hand gun and long gun
groupedBySatate = df_GunData.groupby('state')['permit','handgun','long_gun'].sum()
```

In [ ]:

```

```

In [29]:

```
#preview the groupby
groupedBySatate
```

Out[29]:

	permit	handgun	long_gun
state			
Alabama	1047441.0	2222037.0	2626029.0
Alaska	13352.0	434602.0	572174.0
Arizona	710509.0	1781468.0	1480762.0
Arkansas	507951.0	965584.0	1663256.0
California	6530543.0	5910880.0	5936770.0
Colorado	426970.0	2528422.0	2726033.0
Connecticut	1595270.0	879260.0	638096.0

Delaware	22720.0	208109.0	242235.0
District of Columbia	330.0	4570.0	605.0
Florida	1088802.0	5909952.0	3829090.0
Georgia	1786708.0	2127777.0	2288386.0
Guam	0.0	7642.0	6035.0
Hawaii	197321.0	3.0	35.0
Idaho	369220.0	471836.0	881447.0
Illinois	9498639.0	2434983.0	2289041.0
Indiana	2965760.0	2181274.0	2336722.0
Iowa	1467974.0	18512.0	773606.0
Kansas	155471.0	931302.0	1289902.0
Kentucky	24926779.0	1753246.0	2239863.0
Louisiana	84376.0	1798366.0	2291899.0
Maine	17260.0	439753.0	716039.0
Mariana Islands	0.0	249.0	182.0
Maryland	345839.0	601656.0	1039895.0
Massachusetts	1365225.0	661861.0	430886.0
Michigan	3313076.0	940859.0	2860539.0
Minnesota	2453637.0	1204844.0	2406917.0
Mississippi	99945.0	1186232.0	1717659.0
Missouri	459241.0	2418598.0	3071938.0
Montana	145138.0	466855.0	959192.0
Nebraska	557192.0	12831.0	542230.0
Nevada	212922.0	761226.0	628341.0
New Hampshire	409328.0	571333.0	587647.0
New Jersey	0.0	575930.0	547655.0
New Mexico	93241.0	794444.0	899343.0
New York	804474.0	988487.0	2719577.0
North Carolina	3872974.0	151110.0	2962831.0
North Dakota	104526.0	209815.0	580539.0
Ohio	372281.0	3791457.0	3646325.0

Oklahoma	9.0	1762896.0	2023098.0
Oregon	36899.0	1646143.0	2019502.0
Pennsylvania	1496008.0	2166025.0	9383642.0
Puerto Rico	0.0	156911.0	31533.0
Rhode Island	0.0	138554.0	121830.0
South Carolina	935319.0	1363203.0	1336777.0
South Dakota	36637.0	315801.0	770281.0
Tennessee	1130468.0	3117004.0	2866345.0
Texas	2857582.0	6493832.0	7651396.0
Utah	2753458.0	567263.0	930350.0
Vermont	0.0	186008.0	285064.0
Virgin Islands	11485.0	3112.0	431.0
Virginia	25688.0	2885493.0	2861010.0
Washington	1668159.0	2025498.0	2050150.0
West Virginia	99018.0	1000020.0	1510211.0
Wisconsin	768091.0	1597173.0	2660589.0
Wyoming	78980.0	280782.0	438096.0

In [ ]:

```

```

In [ ]:

```

```

In [30]:

```
#I want to transpose the table census, so it matches the gun table." Sorted by state"
#where the states will be the rows instead of columns,
#saved it into new variable
df_Census_data_Transposed = df_Census_data.T
```

In [ ]:

```

```

In [31]:

```
#Previw
df_Census_data_Transposed
```

Out[31]:

	0	1	2	3	
Fact	Population estimates, July 1, 2016, (V2016)	Population estimates base, April 1, 2010, (V2...	Population, percent change - April 1, 2010 (es...	Population, Census, April 1, 2010	Perso under year percer July 2016,
Fact Note	NaN	NaN	NaN	NaN	Na
Alabama	4,863,300	4,780,131	1.70%	4,779,736	6.00
Alaska	741,894	710,249	4.50%	710,231	7.30
Arizona	6,931,071	6,392,301	8.40%	6,392,017	6.30
Arkansas	2,988,248	2,916,025	2.50%	2,915,918	6.40
California	39,250,017	37,254,522	5.40%	37,253,956	6.30
Colorado	5,540,545	5,029,324	10.20%	5,029,196	6.10
Connecticut	3,576,452	3,574,114	0.10%	3,574,097	5.20
Delaware	952,065	897,936	6.00%	897,934	5.80
Florida	20,612,439	18,804,592	9.60%	18,801,310	5.50
Georgia	10,310,371	9,688,680	6.40%	9,687,653	6.40
Hawaii	1,428,557	1,360,301	5.00%	1,360,301	6.40
Idaho	1,683,140	1,567,650	7.40%	1,567,582	6.80

Illinois	12,801,539	12,831,574	-0.20%	12,830,632	6.00
Indiana	6,633,053	6,484,136	2.30%	6,483,802	6.40
Iowa	3,134,693	3,046,869	2.90%	3,046,355	6.40
Kansas	2,907,289	2,853,129	1.90%	2,853,118	6.70
Kentucky	4,436,974	4,339,344	2.20%	4,339,367	6.20
Louisiana	4,681,666	4,533,479	3.30%	4,533,372	6.60
Maine	1,331,479	1,328,364	0.20%	1,328,361	4.90
Maryland	6,016,447	5,773,786	4.20%	5,773,552	6.10
Massachusetts	6,811,779	6,547,813	4.00%	6,547,629	5.30
Michigan	9,928,300	9,884,129	0.40%	9,883,640	5.80
Minnesota	5,519,952	5,303,924	4.10%	5,303,925	6.40
Mississippi	2,988,726	2,968,103	0.70%	2,967,297	6.30
Missouri	6,093,000	5,988,928	1.70%	5,988,927	6.10
Montana	1,042,520	989,414	5.40%	989,415	6.00
Nebraska	1,907,116	1,826,334	4.40%	1,826,341	7.00
Nevada	2,940,058	2,700,691	8.90%	2,700,551	6.30
New Hampshire	1,334,795	1,316,461	1.40%	1,316,470	4.80
New Jersey	8,944,469	8,791,953	1.70%	8,791,894	5.80
New Mexico	2081015	2059198	0.011	2059179	0.06
New York	19745289	19378110	0.019	19378102	0.05
North Carolina	10146788	9535688	0.064	9535483	0.0
North Dakota	757952	672591	0.127	672591	0.07
Ohio	11614373	11536727	0.007	11536504	0.0
Oklahoma	3923561	3751615	0.046	3751351	0.06
Oregon	4093465	3831072	0.068	3831074	0.05
Pennsylvania	12784227	12702857	0.006	12702379	0.05
Rhode Island	1056426	1052940	0.003	1052567	0.05
South Carolina	4961119	4625410	0.073	4625364	0.05
South Dakota	865454	814195	0.063	814180	0.07
Tennessee	6651194	6346298	0.048	6346105	0.06

Texas	27,862,596	25,146,100	10.80%	25,145,561	7.20
Utah	3,051,217	2,763,888	10.40%	2,763,885	8.30
Vermont	624,594	625,741	-0.20%	625,741	4.90
Virginia	8,411,808	8,001,041	5.10%	8,001,024	6.10
Washington	7,288,000	6,724,545	8.40%	6,724,540	6.20
West Virginia	1,831,102	1,853,011	-1.20%	1,852,994	5.50
Wisconsin	5,778,708	5,687,289	1.60%	5,686,986	5.80
Wyoming	585,501	563,767	3.90%	563,626	6.50

52 rows × 85 columns

In [ ]:

```

```

In [32]:

```
# Drop the first row that is labeled Fact
df_Census_data_Transposed = df_Census_data_Transposed.drop("Fact" , axis=0)
```

In [ ]:

```

```

In [33]:

```
#Preview
df_Census_data_Transposed
```

Out[33]:

	0	1	2	3	4	
Fact Note	NaN	NaN	NaN	NaN	NaN	
Alabama	4,863,300	4,780,131	1.70%	4,779,736	6.00%	6

Alaska	741,894	710,249	4.50%	710,231	7.30%	7
Arizona	6,931,071	6,392,301	8.40%	6,392,017	6.30%	7
Arkansas	2,988,248	2,916,025	2.50%	2,915,918	6.40%	6
California	39,250,017	37,254,522	5.40%	37,253,956	6.30%	6
Colorado	5,540,545	5,029,324	10.20%	5,029,196	6.10%	6
Connecticut	3,576,452	3,574,114	0.10%	3,574,097	5.20%	5
Delaware	952,065	897,936	6.00%	897,934	5.80%	6
Florida	20,612,439	18,804,592	9.60%	18,801,310	5.50%	5
Georgia	10,310,371	9,688,680	6.40%	9,687,653	6.40%	7
Hawaii	1,428,557	1,360,301	5.00%	1,360,301	6.40%	6
Idaho	1,683,140	1,567,650	7.40%	1,567,582	6.80%	7
Illinois	12,801,539	12,831,574	-0.20%	12,830,632	6.00%	6
Indiana	6,633,053	6,484,136	2.30%	6,483,802	6.40%	6
Iowa	3,134,693	3,046,869	2.90%	3,046,355	6.40%	6
Kansas	2,907,289	2,853,129	1.90%	2,853,118	6.70%	7
Kentucky	4,436,974	4,339,344	2.20%	4,339,367	6.20%	6
Louisiana	4,681,666	4,533,479	3.30%	4,533,372	6.60%	6
Maine	1,331,479	1,328,364	0.20%	1,328,361	4.90%	5
Maryland	6,016,447	5,773,786	4.20%	5,773,552	6.10%	6
Massachusetts	6,811,779	6,547,813	4.00%	6,547,629	5.30%	5
Michigan	9,928,300	9,884,129	0.40%	9,883,640	5.80%	6
Minnesota	5,519,952	5,303,924	4.10%	5,303,925	6.40%	6
Mississippi	2,988,726	2,968,103	0.70%	2,967,297	6.30%	7
Missouri	6,093,000	5,988,928	1.70%	5,988,927	6.10%	6
Montana	1,042,520	989,414	5.40%	989,415	6.00%	6
Nebraska	1,907,116	1,826,334	4.40%	1,826,341	7.00%	7
Nevada	2,940,058	2,700,691	8.90%	2,700,551	6.30%	6
New Hampshire	1,334,795	1,316,461	1.40%	1,316,470	4.80%	5
New Jersey	8,944,469	8,791,953	1.70%	8,791,894	5.80%	6
New Mexico	2081015	2059198	0.011	2059179	0.062	

New York	19745289	19378110	0.019	19378102	0.059	
North Carolina	10146788	9535688	0.064	9535483	0.06	
North Dakota	757952	672591	0.127	672591	0.073	
Ohio	11614373	11536727	0.007	11536504	0.06	
Oklahoma	3923561	3751615	0.046	3751351	0.068	
Oregon	4093465	3831072	0.068	3831074	0.058	
Pennsylvania	12784227	12702857	0.006	12702379	0.056	
Rhode Island	1056426	1052940	0.003	1052567	0.052	
South Carolina	4961119	4625410	0.073	4625364	0.059	
South Dakota	865454	814195	0.063	814180	0.071	
Tennessee	6651194	6346298	0.048	6346105	0.061	
Texas	27,862,596	25,146,100	10.80%	25,145,561	7.20%	7
Utah	3,051,217	2,763,888	10.40%	2,763,885	8.30%	9
Vermont	624,594	625,741	-0.20%	625,741	4.90%	5
Virginia	8,411,808	8,001,041	5.10%	8,001,024	6.10%	6
Washington	7,288,000	6,724,545	8.40%	6,724,540	6.20%	6
West Virginia	1,831,102	1,853,011	-1.20%	1,852,994	5.50%	5
Wisconsin	5,778,708	5,687,289	1.60%	5,686,986	5.80%	6
Wyoming	585,501	563,767	3.90%	563,626	6.50%	7

51 rows × 85 columns

In [ ]:

```
```

In [34]:

```
# Drop the first row that is labeled Fact Note
df_Census_data_Transposed = df_Census_data_Transposed.drop("Fact Note" , axis=0)
```

In [ ]:

```
```

In [35]:

```
#Preview
df_Census_data_Transposed
```

Out[35]:

	0	1	2	3	4	
Alabama	4,863,300	4,780,131	1.70%	4,779,736	6.00%	6
Alaska	741,894	710,249	4.50%	710,231	7.30%	7
Arizona	6,931,071	6,392,301	8.40%	6,392,017	6.30%	7
Arkansas	2,988,248	2,916,025	2.50%	2,915,918	6.40%	6
California	39,250,017	37,254,522	5.40%	37,253,956	6.30%	6
Colorado	5,540,545	5,029,324	10.20%	5,029,196	6.10%	6
Connecticut	3,576,452	3,574,114	0.10%	3,574,097	5.20%	5
Delaware	952,065	897,936	6.00%	897,934	5.80%	6
Florida	20,612,439	18,804,592	9.60%	18,801,310	5.50%	5
Georgia	10,310,371	9,688,680	6.40%	9,687,653	6.40%	7
Hawaii	1,428,557	1,360,301	5.00%	1,360,301	6.40%	6
Idaho	1,683,140	1,567,650	7.40%	1,567,582	6.80%	7
Illinois	12,801,539	12,831,574	-0.20%	12,830,632	6.00%	6
Indiana	6,633,053	6,484,136	2.30%	6,483,802	6.40%	6
Iowa	3,134,693	3,046,869	2.90%	3,046,355	6.40%	6
Kansas	2,907,289	2,853,129	1.90%	2,853,118	6.70%	7
Kentucky	4,436,974	4,339,344	2.20%	4,339,367	6.20%	6
Louisiana	4,681,666	4,533,479	3.30%	4,533,372	6.60%	6
Maine	1,331,479	1,328,364	0.20%	1,328,361	4.90%	5
Maryland	6,016,447	5,773,786	4.20%	5,773,552	6.10%	6
Massachusetts	6,811,779	6,547,813	4.00%	6,547,629	5.30%	5
Michigan	9,928,300	9,884,129	0.40%	9,883,640	5.80%	6
Minnesota	5,519,952	5,303,924	4.10%	5,303,925	6.40%	6
Mississippi	2,988,726	2,968,103	0.70%	2,967,297	6.30%	7
Missouri	6,093,000	5,988,928	1.70%	5,988,927	6.10%	6

Montana	1,042,520	989,414	5.40%	989,415	6.00%	6
Nebraska	1,907,116	1,826,334	4.40%	1,826,341	7.00%	7
Nevada	2,940,058	2,700,691	8.90%	2,700,551	6.30%	6
New Hampshire	1,334,795	1,316,461	1.40%	1,316,470	4.80%	5
New Jersey	8,944,469	8,791,953	1.70%	8,791,894	5.80%	6
New Mexico	2081015	2059198	0.011	2059179	0.062	
New York	19745289	19378110	0.019	19378102	0.059	
North Carolina	10146788	9535688	0.064	9535483	0.06	
North Dakota	757952	672591	0.127	672591	0.073	
Ohio	11614373	11536727	0.007	11536504	0.06	
Oklahoma	3923561	3751615	0.046	3751351	0.068	
Oregon	4093465	3831072	0.068	3831074	0.058	
Pennsylvania	12784227	12702857	0.006	12702379	0.056	
Rhode Island	1056426	1052940	0.003	1052567	0.052	
South Carolina	4961119	4625410	0.073	4625364	0.059	
South Dakota	865454	814195	0.063	814180	0.071	
Tennessee	6651194	6346298	0.048	6346105	0.061	
Texas	27,862,596	25,146,100	10.80%	25,145,561	7.20%	7
Utah	3,051,217	2,763,888	10.40%	2,763,885	8.30%	9
Vermont	624,594	625,741	-0.20%	625,741	4.90%	5
Virginia	8,411,808	8,001,041	5.10%	8,001,024	6.10%	6
Washington	7,288,000	6,724,545	8.40%	6,724,540	6.20%	6
West Virginia	1,831,102	1,853,011	-1.20%	1,852,994	5.50%	5
Wisconsin	5,778,708	5,687,289	1.60%	5,686,986	5.80%	6
Wyoming	585,501	563,767	3.90%	563,626	6.50%	7

50 rows × 85 columns

```
In [ ]:
```

In [36]:

```python
#Rename the column from name 0 to name population
df_Census_data_Transposed.rename({0: "Population"}, axis= 'columns', inplace = True)
```

In [ ]:

```python

```

In [37]:

```python
#Preview
df_Census_data_Transposed
```

Out[37]:

	Population	1	2	3	4	
Alabama	4,863,300	4,780,131	1.70%	4,779,736	6.00%	6
Alaska	741,894	710,249	4.50%	710,231	7.30%	7
Arizona	6,931,071	6,392,301	8.40%	6,392,017	6.30%	7
Arkansas	2,988,248	2,916,025	2.50%	2,915,918	6.40%	6
California	39,250,017	37,254,522	5.40%	37,253,956	6.30%	6
Colorado	5,540,545	5,029,324	10.20%	5,029,196	6.10%	6
Connecticut	3,576,452	3,574,114	0.10%	3,574,097	5.20%	5
Delaware	952,065	897,936	6.00%	897,934	5.80%	6
Florida	20,612,439	18,804,592	9.60%	18,801,310	5.50%	5
Georgia	10,310,371	9,688,680	6.40%	9,687,653	6.40%	7
Hawaii	1,428,557	1,360,301	5.00%	1,360,301	6.40%	6
Idaho	1,683,140	1,567,650	7.40%	1,567,582	6.80%	7
Illinois	12,801,539	12,831,574	-0.20%	12,830,632	6.00%	6
Indiana	6,633,053	6,484,136	2.30%	6,483,802	6.40%	6
Iowa	3,134,693	3,046,869	2.90%	3,046,355	6.40%	6
Kansas	2,907,289	2,853,129	1.90%	2,853,118	6.70%	7
Kentucky	4,436,974	4,339,344	2.20%	4,339,367	6.20%	6
Louisiana	4,681,666	4,533,479	3.30%	4,533,372	6.60%	6

Maine	1,331,479	1,328,364	0.20%	1,328,361	4.90%	5
Maryland	6,016,447	5,773,786	4.20%	5,773,552	6.10%	6
Massachusetts	6,811,779	6,547,813	4.00%	6,547,629	5.30%	5
Michigan	9,928,300	9,884,129	0.40%	9,883,640	5.80%	6
Minnesota	5,519,952	5,303,924	4.10%	5,303,925	6.40%	6
Mississippi	2,988,726	2,968,103	0.70%	2,967,297	6.30%	7
Missouri	6,093,000	5,988,928	1.70%	5,988,927	6.10%	6
Montana	1,042,520	989,414	5.40%	989,415	6.00%	6
Nebraska	1,907,116	1,826,334	4.40%	1,826,341	7.00%	7
Nevada	2,940,058	2,700,691	8.90%	2,700,551	6.30%	6
New Hampshire	1,334,795	1,316,461	1.40%	1,316,470	4.80%	5
New Jersey	8,944,469	8,791,953	1.70%	8,791,894	5.80%	6
New Mexico	2081015	2059198	0.011	2059179	0.062	
New York	19745289	19378110	0.019	19378102	0.059	
North Carolina	10146788	9535688	0.064	9535483	0.06	
North Dakota	757952	672591	0.127	672591	0.073	
Ohio	11614373	11536727	0.007	11536504	0.06	
Oklahoma	3923561	3751615	0.046	3751351	0.068	
Oregon	4093465	3831072	0.068	3831074	0.058	
Pennsylvania	12784227	12702857	0.006	12702379	0.056	
Rhode Island	1056426	1052940	0.003	1052567	0.052	
South Carolina	4961119	4625410	0.073	4625364	0.059	
South Dakota	865454	814195	0.063	814180	0.071	
Tennessee	6651194	6346298	0.048	6346105	0.061	
Texas	27,862,596	25,146,100	10.80%	25,145,561	7.20%	7
Utah	3,051,217	2,763,888	10.40%	2,763,885	8.30%	9
Vermont	624,594	625,741	-0.20%	625,741	4.90%	5
Virginia	8,411,808	8,001,041	5.10%	8,001,024	6.10%	6
Washington	7,288,000	6,724,545	8.40%	6,724,540	6.20%	6
West Virginia	1,831,102	1,853,011	-1.20%	1,852,994	5.50%	5

Wisconsin	5,778,708	5,687,289	1.60%	5,686,986	5.80%	6
Wyoming	585,501	563,767	3.90%	563,626	6.50%	7

50 rows × 85 columns

```
In [ ]:
```

```
In [38]:
```

```
#Dropping the unneeded coulumns
df_Census_data_Transposed = df_Census_data_Transposed.drop(range (1,85) , axis=1)
```

```
In [ ]:
```

```
In [39]:
```

```
#Preview
df_Census_data_Transposed
```

```
Out[39]:
```

	Population
Alabama	4,863,300
Alaska	741,894
Arizona	6,931,071
Arkansas	2,988,248
California	39,250,017
Colorado	5,540,545
Connecticut	3,576,452
Delaware	952,065
Florida	20,612,439
Georgia	10,310,371
Hawaii	1,428,557
Idaho	1,683,140

Illinois	12,801,539
Indiana	6,633,053
Iowa	3,134,693
Kansas	2,907,289
Kentucky	4,436,974
Louisiana	4,681,666
Maine	1,331,479
Maryland	6,016,447
Massachusetts	6,811,779
Michigan	9,928,300
Minnesota	5,519,952
Mississippi	2,988,726
Missouri	6,093,000
Montana	1,042,520
Nebraska	1,907,116
Nevada	2,940,058
New Hampshire	1,334,795
New Jersey	8,944,469
New Mexico	2081015
New York	19745289
North Carolina	10146788
North Dakota	757952
Ohio	11614373
Oklahoma	3923561
Oregon	4093465
Pennsylvania	12784227
Rhode Island	1056426
South Carolina	4961119
South Dakota	865454
Tennessee	6651194
Texas	27,862,596

Utah	3,051,217
Vermont	624,594
Virginia	8,411,808
Washington	7,288,000
West Virginia	1,831,102
Wisconsin	5,778,708
Wyoming	585,501

Exploratory Data Analysis

> **Tip**: Now that you've trimmed and cleaned your data, you're ready to move on to exploration. Compute statistics and create visualizations with the goal of addressing the research questions that you posed in the Introduction section. It is recommended that you be systematic with your approach. Look at one variable at a time, and then follow it up by looking at relationships between variables.

Research Question 1

What is the number of gun permit per capita ?

To do that I will divide the number of gun permit by pupulation.

In [ ]:

```
# Use this, and more code cells, to explore your data. Don't f
orget to add
#   Markdown cells to document your observations and findings.
```

In [40]:

```
# Firat I will merge the needed data into onte table
all_data = pd.merge(df_Census_data_Transposed, groupedBySatate
, right_index= True, left_index= True)
```

In [41]:

```
all_data.head()
```

Out[41]:

	Population	permit	handgun	long_gun
Alabama	4,863,300	1047441.0	2222037.0	2626029.0
Alaska	741,894	13352.0	434602.0	572174.0
Arizona	6,931,071	710509.0	1781468.0	1480762.0
Arkansas	2,988,248	507951.0	965584.0	1663256.0
California	39,250,017	6530543.0	5910880.0	5936770.0

In [ ]:

```
#population is str type, and has comma seperated,
#needs to remove commas then convert the Population column data type from string to float.
#to get the division to work
#Reference:
#https://stackoverflow.com/questions/39125665/cannot-convert-string-to-float-in-pandas-valueerror
```

In [42]:

```
#Converting data type and removing commas
all_data['Population'] = all_data['Population'].apply(lambda x : float(x.split()[0].replace(',', '')))
```

In [43]:

```
#Preview
all_data.dtypes
```

Out[43]:

```
Population    float64
permit        float64
handgun       float64
long_gun      float64
dtype: object
```

In [44]:

```
all_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 50 entries, Alabama to Wyoming
Data columns (total 4 columns):
Population    50 non-null float64
permit        50 non-null float64
handgun       50 non-null float64
long_gun      50 non-null float64
dtypes: float64(4)
memory usage: 2.0+ KB
```

In [55]:

```
#Question number 1: What is the permit per capita ?
all_data['permit_per_capita'] = all_data['Population']/all_data['permit']
```

In [49]:

```
# I need to add colu,n that has all the states so I can plot it against my data

states = ['Alabama','Alaska', 'Arizona','Arkansas', 'California', 'Colorado', 'Connecticut', 'Delaware', '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']
```

In [50]:

```
#Make sure I had all 50 states in the list
print(len(states))
```

```
50
```

In [51]:

```
#Addig the states list as a new column
#Reference
#https://www.geeksforgeeks.org/adding-new-column-to-existing-d
ataframe-in-pandas/
all_data['states']=states
```

In [52]:

```
#preview
all_data.head(30)
```

Out[52]:

```
              Population      permit    handgun   long_gun
Alabama        4863300.0   1047441.0  2222037.0  2626029.0      /
Alaska          741894.0     13352.0   434602.0   572174.0
Arizona        6931071.0    710509.0  1781468.0  1480762.0
Arkansas       2988248.0    507951.0   965584.0  1663256.0      /
California    39250017.0   6530543.0  5910880.0  5936770.0      (
Colorado       5540545.0    426970.0  2528422.0  2726033.0      (
Connecticut    3576452.0   1595270.0   879260.0   638096.0    Cor
Delaware        952065.0     22720.0   208109.0   242235.0      [
Florida       20612439.0   1088802.0  5909952.0  3829090.0
Georgia       10310371.0   1786708.0  2127777.0  2288386.0
Hawaii         1428557.0    197321.0        3.0       35.0
Idaho          1683140.0    369220.0   471836.0   881447.0
Illinois      12801539.0   9498639.0  2434983.0  2289041.0
Indiana        6633053.0   2965760.0  2181274.0  2336722.0
Iowa           3134693.0   1467974.0    18512.0   773606.0
Kansas         2907289.0    155471.0   931302.0  1289902.0
Kentucky       4436974.0  24926779.0  1753246.0  2239863.0      I
Louisiana      4681666.0     84376.0  1798366.0  2291899.0      L
Maine          1331479.0     17260.0   439753.0   716039.0
```

Maryland	6016447.0	345839.0	601656.0	1039895.0	M
Massachusetts	6811779.0	1365225.0	661861.0	430886.0	Massa
Michigan	9928300.0	3313076.0	940859.0	2860539.0	I
Minnesota	5519952.0	2453637.0	1204844.0	2406917.0	M
Mississippi	2988726.0	99945.0	1186232.0	1717659.0	Mi
Missouri	6093000.0	459241.0	2418598.0	3071938.0	
Montana	1042520.0	145138.0	466855.0	959192.0	
Nebraska	1907116.0	557192.0	12831.0	542230.0	N
Nevada	2940058.0	212922.0	761226.0	628341.0	
New Hampshire	1334795.0	409328.0	571333.0	587647.0	Ha
New Jersey	8944469.0	0.0	575930.0	547655.0	Ne

In [69]:

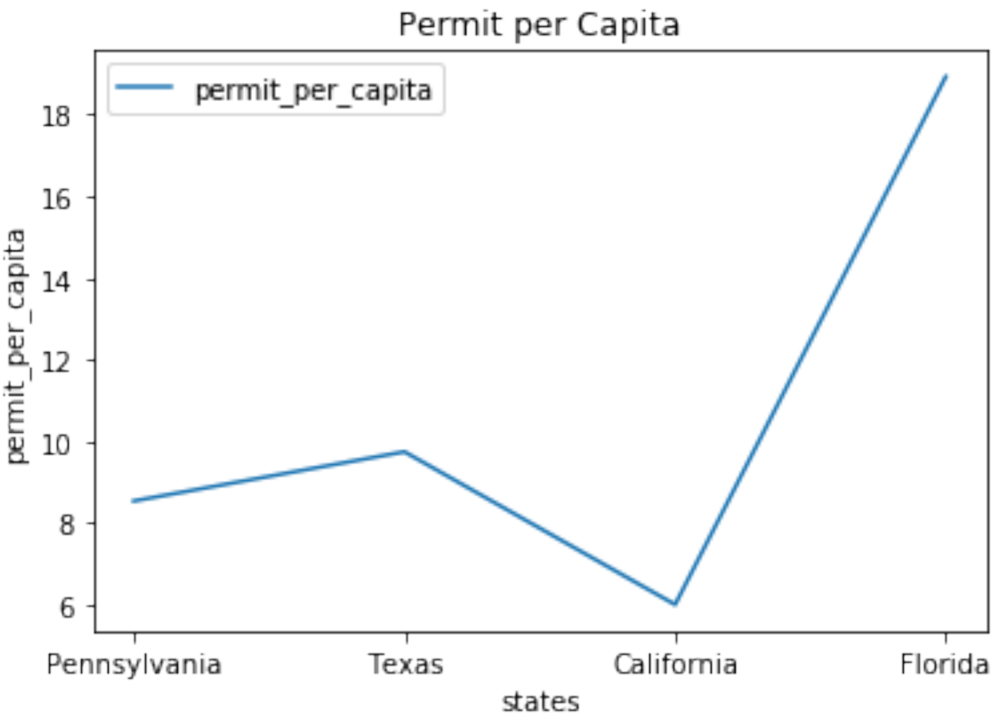
```
# I want to analyze the permit per capit amoung 4 states only,
they are { Pennsylvania , Texas , California, Florida}
# I have used iloc to select the rows I wanted.
#Reference : https://www.shanelynn.ie/using-pandas-dataframe-c
reating-editing-viewing-data-in-python/

four_states = all_data.iloc[0:4, :]
```

In [70]:

```
#Plotting data, to show the difference between the value of permit per capita for each of the four states.

plt.title('Permit per Capita')
plt.xlabel('states')
plt.ylabel('permit_per_capita')
plt.plot(four_states['states'] , four_states['permit_per_capita'] )
plt.legend()
plt.show()
```



In [ ]:

```
#Findings :
# california has the heighst permit per capita amoung the 4 selected states.
```

In [ ]:

```
#Question numbe 2
#What are the top 5 states with hand gun and long gun
#Reference
#https://stackoverflow.com/questions/43859416/finding-top-10-in-a-dataframe-in-pandas
```

In [58]:

```
all_data.sort_values('handgun',ascending=False,inplace=True)
top_five_handgun_states = all_data['states'].head(5)
top_five_handgun_values = all_data['handgun'].head(5)
```

In [59]:

```
#Preview
top_five_handgun_states
top_five_handgun_values
```

Out[59]:

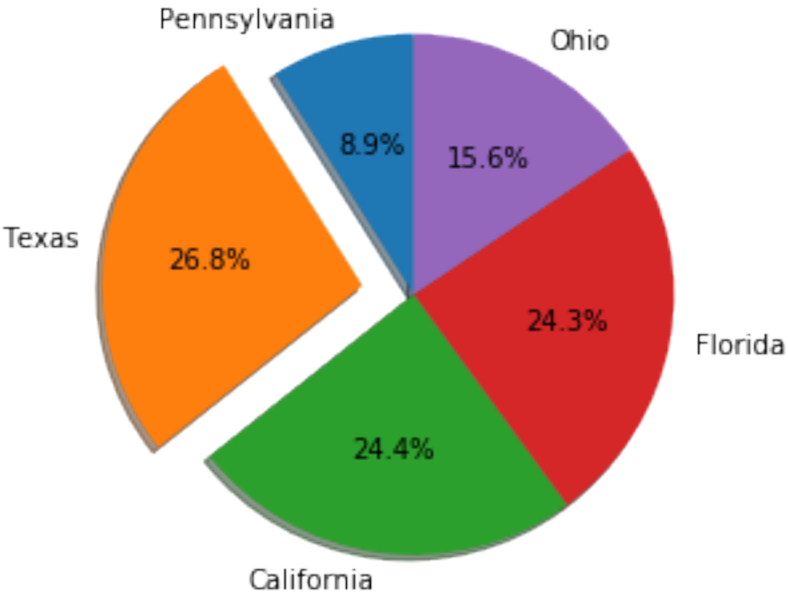
```
Texas         6493832.0
California    5910880.0
Florida       5909952.0
Ohio          3791457.0
Tennessee     3117004.0
Name: handgun, dtype: float64
```

In [71]:

```
#Plotting using pie chart
#Refernce : https://matplotlib.org/examples/pie_and_polar_charts/pie_demo_features.html
# Pie chart, where the slices will be ordered and plotted counter-clockwise:
labels = 'Pennsylvania', 'Texas', 'California', 'Florida' , 'Ohio'
sizes = [2166025.0, 6493832.0, 5910880.0, 5909952.0,3791457.0]
explode = (0, 0.2, 0, 0,0)  # only "explode" the 2nd slice (i.e. 'Texas')

fig1, ax1 = plt.subplots()
ax1.pie(sizes,explode = explode,  labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax1.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

plt.show()
```



In [ ]:

```

```

In [65]:

```
all_data.sort_values('long_gun',ascending=False,inplace=True)
top_five_long_gun_states = all_data['states'].head(5)
top_five_long_gun_values = all_data['long_gun'].head(5)
```

In [72]:

```
#Preview
top_five_long_gun_states
top_five_long_gun_values
```

Out[72]:

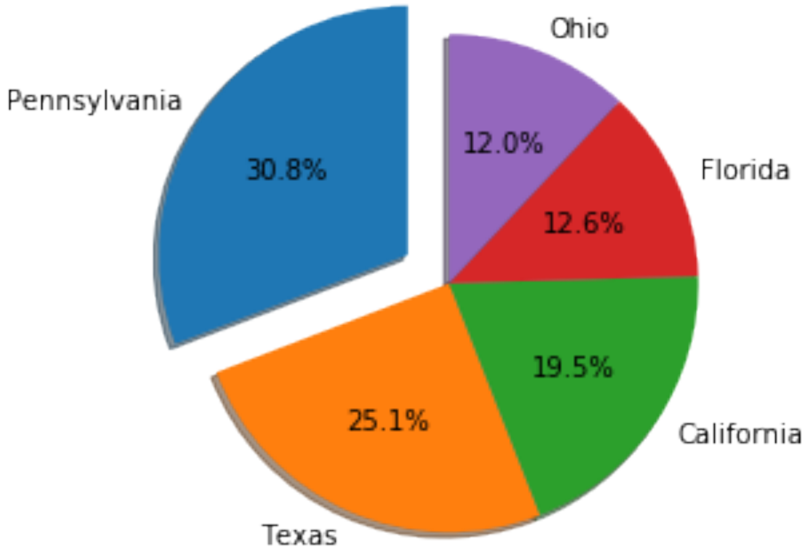
```
Pennsylvania    9383642.0
Texas           7651396.0
California      5936770.0
Florida         3829090.0
Ohio            3646325.0
Name: long_gun, dtype: float64
```

In [74]:

```
#Plotting using pie chart
#Refernce : https://matplotlib.org/examples/pie_and_polar_charts/pie_demo_features.html
# Pie chart, where the slices will be ordered and plotted counter-clockwise:
labels = 'Pennsylvania', 'Texas', 'California', 'Florida' , 'Ohio'
sizes = [9383642.0, 7651396.0, 5936770.0, 3829090.0,3646325.0]
explode = (0.2, 0, 0, 0,0)  # only "explode" the 2nd slice (i.e. 'Texas')

fig1, ax1 = plt.subplots()
ax1.pie(sizes,explode = explode,  labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax1.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

plt.show()
```



In [ ]:

```
#Findings:
# the state that owned the most hand guns is Texas
#while Pennsylvania is the state that is top 1 in owning long gun.
#That might predict that people there are haunting a lot.
```