Project: FBI Gun Investigation and Data Analysis

Table of Contents ¶

Introduction

> The source of the data is FBI's National Instant Criminal Background Check System (NICS). This system is used to determine whether a prospective buyer is eligible to buy firearms or explosives. Gun shops call into this system to ensure that each customer does not have a criminal record or isn't otherwise ineligible to make a purchase. The data has been supplemented with state level data from census.gov.

About the data:

> The NICS data is found in a single sheet of an .xlsx file. It contains the number of firearm checks by month, state, and type. The U.S. census data is found in a .csv file. It contains several variables at the state level. Most variables just have one data point per state (2016), but a few have data for more than one year.
>
> Questions asked:
>
> 1. Which state has the highest total gun purchases in April 2000 and April 2010?
> 2. What is per capita firearm sales for all states in April 2010 vs July 2016?

In [420]:
```
# Use this cell to set up import statements for all of the packages that you p
lan to use.
import pandas as pd
import numpy as np
import datetime

# Remember to include a 'magic word' so that your visualizations are plotted
# inline with the notebook. See this page for more:
#   http://ipython.readthedocs.io/en/stable/interactive/magics.html
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
sns.set_style('darkgrid')
```

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 [421]:
```
# 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.
df_census = pd.read_csv('U.S. Census Data.csv')
df_gun = pd.read_excel('gun_data.xlsx')
```

In [422]:
```
# Printing out a few lines of both the datasets
display(df_census.head())
display(df_gun.head())
```

	State	Population estimates, July 1, 2016, (V2016)	Population estimates base, April 1, 2010, (V2016)	Population, percent change - April 1, 2010 (estimates base) to July 1, 2016, (V2016)	Population, Census, April 1, 2010	Persons under 5 years, percent, July 1, 2016, (V2016)	Persons under 5 years, percent, April 1, 2010	Persons under 18 years, percent, July 1, 2016, (V2016)	Perso unc yea perce April 20
0	Alabama	4863300	4780131	0.017	4779736	0.060	0.064	0.226	0.2
1	Alaska	741894	710249	0.045	710231	0.073	0.076	0.252	0.2
2	Arizona	6931071	6392301	0.084	6392017	0.063	0.071	0.235	0.2
3	Arkansas	2988248	2916025	0.025	2915918	0.064	0.068	0.236	0.2
4	California	39250017	37254522	0.054	37253956	0.063	0.068	0.232	0.2

5 rows × 66 columns

	month	state	permit	permit_recheck	handgun	long_gun	other	multiple	admin	prep
0	2017-09	Alabama	16717.0	0.0	5734.0	6320.0	221.0	317	0.0	
1	2017-09	Alaska	209.0	2.0	2320.0	2930.0	219.0	160	0.0	
2	2017-09	Arizona	5069.0	382.0	11063.0	7946.0	920.0	631	0.0	
3	2017-09	Arkansas	2935.0	632.0	4347.0	6063.0	165.0	366	51.0	
4	2017-09	California	57839.0	0.0	37165.0	24581.0	2984.0	0	0.0	

5 rows × 27 columns

In [423]:
```
df_census.set_index('State',inplace = True)
df_census
```

Out[423]:

State	Population estimates, July 1, 2016, (V2016)	Population estimates base, April 1, 2010, (V2016)	Population, percent change - April 1, 2010 (estimates base) to July 1, 2016, (V2016)	Population, Census, April 1, 2010	Persons under 5 years, percent, July 1, 2016, (V2016)	Persons under 5 years, percent, April 1, 2010	Persons under 18 years, percent, July 1, 2016, (V2016)	Pe Y pe A
Alabama	4863300	4780131	0.017	4779736	0.060	0.064	0.226	
Alaska	741894	710249	0.045	710231	0.073	0.076	0.252	
Arizona	6931071	6392301	0.084	6392017	0.063	0.071	0.235	
Arkansas	2988248	2916025	0.025	2915918	0.064	0.068	0.236	
California	39250017	37254522	0.054	37253956	0.063	0.068	0.232	
Colorado	5540545	5029324	0.102	5029196	0.061	0.068	0.228	
Connecticut	3576452	3574114	0.001	3574097	0.052	0.057	0.211	
Delaware	952065	897936	0.060	897934	0.058	0.062	0.215	
Florida	20612439	18804592	0.096	18801310	0.055	0.057	0.201	
Georgia	10310371	9688680	0.064	9687653	0.064	0.071	0.244	
Hawaii	1428557	1360301	0.050	1360301	0.064	0.064	0.216	
Idaho	1683140	1567650	0.074	1567582	0.068	0.078	0.260	
Illinois	12801539	12831574	-0.002	12830632	0.060	0.065	0.229	
Indiana	6633053	6484136	0.023	6483802	0.064	0.067	0.238	
Iowa	3134693	3046869	0.029	3046355	0.064	0.066	0.233	
Kansas	2907289	2853129	0.019	2853118	0.067	0.072	0.246	
Kentucky	4436974	4339344	0.022	4339367	0.062	0.065	0.228	
Louisiana	4681666	4533479	0.033	4533372	0.066	0.069	0.238	
Maine	1331479	1328364	0.002	1328361	0.049	0.052	0.191	
Maryland	6016447	5773786	0.042	5773552	0.061	0.063	0.224	
Massachusetts	6811779	6547813	0.040	6547629	0.053	0.056	0.202	
Michigan	9928300	9884129	0.004	9883640	0.058	0.060	0.221	
Minnesota	5519952	5303924	0.041	5303925	0.064	0.067	0.233	
Mississippi	2988726	2968103	0.007	2967297	0.063	0.071	0.241	
Missouri	6093000	5988928	0.017	5988927	0.061	0.065	0.228	
Montana	1042520	989414	0.054	989415	0.060	0.063	0.218	
Nebraska	1907116	1826334	0.044	1826341	0.070	0.072	0.248	
Nevada	2940058	2700691	0.089	2700551	0.063	0.069	0.230	
New Hampshire	1334795	1316461	0.014	1316470	0.048	0.053	0.195	

	Population estimates, July 1, 2016, (V2016)	Population estimates base, April 1, 2010, (V2016)	Population, percent change - April 1, 2010 (estimates base) to July 1, 2016, (V2016)	Population, Census, April 1, 2010	Persons under 5 years, percent, July 1, 2016, (V2016)	Persons under 5 years, percent, April 1, 2010	Persons under 18 years, percent, July 1, 2016, (V2016)	Pe ... pe A
State								
New Jersey	8944469	8791953	0.017	8791894	0.058	0.062	0.222	
New Mexico	2081015	2059198	0.011	2059179	0.062	0.070	0.236	
New York	19745289	19378110	0.019	19378102	0.059	0.060	0.212	
North Carolina	10146788	9535688	0.064	9535483	0.060	0.066	0.227	
North Dakota	757952	672591	0.127	672591	0.073	0.066	0.233	
Ohio	11614373	11536727	0.007	11536504	0.060	0.062	0.225	
Oklahoma	3923561	3751615	0.046	3751351	0.068	0.070	0.245	
Oregon	4093465	3831072	0.068	3831074	0.058	0.062	0.212	
Pennsylvania	12784227	12702857	0.006	12702379	0.056	0.057	0.209	
Rhode Island	1056426	1052940	0.003	1052567	0.052	0.055	0.197	
South Carolina	4961119	4625410	0.073	4625364	0.059	0.065	0.221	
South Dakota	865454	814195	0.063	814180	0.071	0.073	0.246	
Tennessee	6651194	6346298	0.048	6346105	0.061	0.064	0.226	
Texas	27862596	25146100	0.108	25145561	0.072	0.077	0.262	
Utah	3051217	2763888	0.104	2763885	0.083	0.095	0.302	
Vermont	624594	625741	-0.002	625741	0.049	0.051	0.190	
Virginia	8411808	8001041	0.051	8001024	0.061	0.064	0.222	
Washington	7288000	6724545	0.084	6724540	0.062	0.065	0.224	
West Virginia	1831102	1853011	-0.012	1852994	0.055	0.056	0.205	
Wisconsin	5778708	5687289	0.016	5686986	0.058	0.063	0.223	
Wyoming	585501	563767	0.039	563626	0.065	0.071	0.237	

50 rows × 65 columns

In [424]:
```
#Inspecting the data and getting information on the dimensions of the  datafra
mes
display("Census dimensions: " +str(df_census.shape))
display("Gun Dimensions:" + str(df_gun.shape))
```

```
'Census dimensions: (50, 65)'

'Gun Dimensions:(12485, 27)'
```

In [425]:
```
#Printing concise summary of both the dataframes helping in knowing the datatypes and null vales
df_census.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 50 entries, Alabama to Wyoming
Data columns (total 65 columns):
 #   Column
Non-Null Count  Dtype
---  ------
--------------  -----
 0   Population estimates, July 1, 2016,  (V2016)
50 non-null     int64
 1   Population estimates base, April 1, 2010,  (V2016)
50 non-null     int64
 2   Population, percent change - April 1, 2010 (estimates base) to July 1, 2
016,  (V2016)   50 non-null     float64
 3   Population, Census, April 1, 2010
50 non-null     int64
 4   Persons under 5 years, percent, July 1, 2016,  (V2016)
50 non-null     float64
 5   Persons under 5 years, percent, April 1, 2010
50 non-null     float64
 6   Persons under 18 years, percent, July 1, 2016,  (V2016)
50 non-null     float64
 7   Persons under 18 years, percent, April 1, 2010
50 non-null     float64
 8   Persons 65 years and over, percent,  July 1, 2016,  (V2016)
50 non-null     float64
 9   Persons 65 years and over, percent, April 1, 2010
50 non-null     float64
 10  Female persons, percent,  July 1, 2016,  (V2016)
50 non-null     float64
 11  Female persons, percent, April 1, 2010
50 non-null     float64
 12  White alone, percent, July 1, 2016,  (V2016)
50 non-null     float64
 13  Black or African American alone, percent, July 1, 2016,  (V2016)
50 non-null     float64
 14  American Indian and Alaska Native alone, percent, July 1, 2016,  (V2016)
50 non-null     float64
 15  Asian alone, percent, July 1, 2016,  (V2016)
50 non-null     float64
 16  Native Hawaiian and Other Pacific Islander alone, percent, July 1, 2016,
(V2016)       50 non-null     object
 17  Two or More Races, percent, July 1, 2016,  (V2016)
50 non-null     float64
 18  Hispanic or Latino, percent, July 1, 2016,  (V2016)
50 non-null     float64
 19  White alone, not Hispanic or Latino, percent, July 1, 2016,  (V2016)
50 non-null     float64
 20  Veterans, 2011-2015
50 non-null     int64
 21  Foreign born persons, percent, 2011-2015
50 non-null     float64
 22  Housing units,  July 1, 2016,  (V2016)
50 non-null     int64
 23  Housing units, April 1, 2010
50 non-null     int64
 24  Owner-occupied housing unit rate, 2011-2015
50 non-null     float64
```

```
 25  Median value of owner-occupied housing units, 2011-2015
50 non-null     int64
 26  Median selected monthly owner costs -with a mortgage, 2011-2015
50 non-null     int64
 27  Median selected monthly owner costs -without a mortgage, 2011-2015
50 non-null     int64
 28  Median gross rent, 2011-2015
50 non-null     int64
 29  Building permits, 2016
50 non-null     int64
 30  Households, 2011-2015
50 non-null     int64
 31  Persons per household, 2011-2015
50 non-null     float64
 32  Living in same house 1 year ago, percent of persons age 1 year+, 2011-20
15              50 non-null     float64
 33  Language other than English spoken at home, percent of persons age 5 yea
rs+, 2011-2015  50 non-null     float64
 34  High school graduate or higher, percent of persons age 25 years+, 2011-2
015             50 non-null     float64
 35  Bachelor's degree or higher, percent of persons age 25 years+, 2011-2015
50 non-null     float64
 36  With a disability, under age 65 years, percent, 2011-2015
50 non-null     float64
 37  Persons  without health insurance, under age 65 years, percent
50 non-null     float64
 38  In civilian labor force, total, percent of population age 16 years+, 201
1-2015          50 non-null     float64
 39  In civilian labor force, female, percent of population age 16 years+, 20
11-2015         50 non-null     float64
 40  Total accommodation and food services sales, 2012 ($1,000)
50 non-null     int64
 41  Total health care and social assistance receipts/revenue, 2012 ($1,000)
50 non-null     int64
 42  Total manufacturers shipments, 2012 ($1,000)
50 non-null     object
 43  Total merchant wholesaler sales, 2012 ($1,000)
50 non-null     int64
 44  Total retail sales, 2012 ($1,000)
50 non-null     int64
 45  Total retail sales per capita, 2012
50 non-null     int64
 46  Mean travel time to work (minutes), workers age 16 years+, 2011-2015
50 non-null     float64
 47  Median household income (in 2015 dollars), 2011-2015
50 non-null     int64
 48  Per capita income in past 12 months (in 2015 dollars), 2011-2015
50 non-null     int64
 49  Persons in poverty, percent
50 non-null     float64
 50  Total employer establishments, 2015
50 non-null     int64
 51  Total employment, 2015
50 non-null     int64
 52  Total annual payroll, 2015 ($1,000)
50 non-null     int64
 53  Total employment, percent change, 2014-2015
```

```
50 non-null     object
 54  Total nonemployer establishments, 2015
50 non-null     int64
 55  All firms, 2012
50 non-null     int64
 56  Men-owned firms, 2012
50 non-null     int64
 57  Women-owned firms, 2012
50 non-null     int64
 58  Minority-owned firms, 2012
50 non-null     int64
 59  Nonminority-owned firms, 2012
50 non-null     int64
 60  Veteran-owned firms, 2012
50 non-null     int64
 61  Nonveteran-owned firms, 2012
50 non-null     int64
 62  Population per square mile, 2010
50 non-null     float64
 63  Land area in square miles, 2010
50 non-null     float64
 64  FIPS Code
50 non-null     object
dtypes: float64(31), int64(30), object(4)
memory usage: 25.8+ KB
```

In [426]:
```
#Printing concise summary of both the dataframes helping in knowing the datatypes and null vales
df_gun.info();
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12485 entries, 0 to 12484
Data columns (total 27 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   month                      12485 non-null  object 
 1   state                      12485 non-null  object 
 2   permit                     12461 non-null  float64
 3   permit_recheck             1100 non-null   float64
 4   handgun                    12465 non-null  float64
 5   long_gun                   12466 non-null  float64
 6   other                      5500 non-null   float64
 7   multiple                   12485 non-null  int64  
 8   admin                      12462 non-null  float64
 9   prepawn_handgun            10542 non-null  float64
 10  prepawn_long_gun           10540 non-null  float64
 11  prepawn_other              5115 non-null   float64
 12  redemption_handgun         10545 non-null  float64
 13  redemption_long_gun        10544 non-null  float64
 14  redemption_other           5115 non-null   float64
 15  returned_handgun           2200 non-null   float64
 16  returned_long_gun          2145 non-null   float64
 17  returned_other             1815 non-null   float64
 18  rentals_handgun            990 non-null    float64
 19  rentals_long_gun           825 non-null    float64
 20  private_sale_handgun       2750 non-null   float64
 21  private_sale_long_gun      2750 non-null   float64
 22  private_sale_other         2750 non-null   float64
 23  return_to_seller_handgun   2475 non-null   float64
 24  return_to_seller_long_gun  2750 non-null   float64
 25  return_to_seller_other     2255 non-null   float64
 26  totals                     12485 non-null  int64  
dtypes: float64(23), int64(2), object(2)
memory usage: 2.6+ MB
```

In [427]:
```
#Giving a general descriptive statistics of the dataframe
df_census.describe()
```

Out[427]:

	Population estimates, July 1, 2016, (V2016)	Population estimates base, April 1, 2010, (V2016)	Population, percent change - April 1, 2010 (estimates base) to July 1, 2016, (V2016)	Population, Census, April 1, 2010	Persons under 5 years, percent, July 1, 2016, (V2016)	Persons under 5 years, percent, April 1, 2010	Persons under 18 years, percent, July 1, 2016, (V2016)
count	5.000000e+01	5.000000e+01	50.000000	5.000000e+01	50.000000	50.000000	50.00000
mean	6.448927e+06	6.163127e+06	0.041800	6.162876e+06	0.061600	0.065460	0.22750
std	7.271769e+06	6.848463e+06	0.033811	6.848235e+06	0.006612	0.007579	0.01977
min	5.855010e+05	5.637670e+05	-0.012000	5.636260e+05	0.048000	0.051000	0.19000
25%	1.850106e+06	1.833003e+06	0.016250	1.833004e+06	0.058000	0.062000	0.21650
50%	4.559320e+06	4.436412e+06	0.040500	4.436370e+06	0.061000	0.065000	0.22750
75%	7.198768e+06	6.680362e+06	0.063750	6.680312e+06	0.064000	0.069750	0.23675
max	3.925002e+07	3.725452e+07	0.127000	3.725396e+07	0.083000	0.095000	0.30200

8 rows × 61 columns

In [428]:
```
#Giving a general descriptive statistics of the dataframe
df_gun.describe()
```

Out[428]:

	permit	permit_recheck	handgun	long_gun	other	multiple
count	12461.000000	1100.000000	12465.000000	12466.000000	5500.000000	12485.000000
mean	6413.629404	1165.956364	5940.881107	7810.847585	360.471636	268.603364
std	23752.338269	9224.200609	8618.584060	9309.846140	1349.478273	783.185073
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	865.000000	2078.250000	17.000000	15.000000
50%	518.000000	0.000000	3059.000000	5122.000000	121.000000	125.000000
75%	4272.000000	0.000000	7280.000000	10380.750000	354.000000	301.000000
max	522188.000000	116681.000000	107224.000000	108058.000000	77929.000000	38907.000000

8 rows × 25 columns

In [429]:
```
#Looking into the column names in the dataframe
for col in df_census.columns:
    print(col)
```

```
Population estimates, July 1, 2016,  (V2016)
Population estimates base, April 1, 2010,  (V2016)
Population, percent change - April 1, 2010 (estimates base) to July 1, 2016,
(V2016)
Population, Census, April 1, 2010
Persons under 5 years, percent, July 1, 2016,  (V2016)
Persons under 5 years, percent, April 1, 2010
Persons under 18 years, percent, July 1, 2016,  (V2016)
Persons under 18 years, percent, April 1, 2010
Persons 65 years and over, percent,  July 1, 2016,  (V2016)
Persons 65 years and over, percent, April 1, 2010
Female persons, percent,  July 1, 2016,  (V2016)
Female persons, percent, April 1, 2010
White alone, percent, July 1, 2016,  (V2016)
Black or African American alone, percent, July 1, 2016,  (V2016)
American Indian and Alaska Native alone, percent, July 1, 2016,  (V2016)
Asian alone, percent, July 1, 2016,  (V2016)
Native Hawaiian and Other Pacific Islander alone, percent, July 1, 2016,  (V2
016)
Two or More Races, percent, July 1, 2016,  (V2016)
Hispanic or Latino, percent, July 1, 2016,  (V2016)
White alone, not Hispanic or Latino, percent, July 1, 2016,  (V2016)
Veterans, 2011-2015
Foreign born persons, percent, 2011-2015
Housing units,  July 1, 2016,  (V2016)
Housing units, April 1, 2010
Owner-occupied housing unit rate, 2011-2015
Median value of owner-occupied housing units, 2011-2015
Median selected monthly owner costs -with a mortgage, 2011-2015
Median selected monthly owner costs -without a mortgage, 2011-2015
Median gross rent, 2011-2015
Building permits, 2016
Households, 2011-2015
Persons per household, 2011-2015
Living in same house 1 year ago, percent of persons age 1 year+, 2011-2015
Language other than English spoken at home, percent of persons age 5 years+,
2011-2015
High school graduate or higher, percent of persons age 25 years+, 2011-2015
Bachelor's degree or higher, percent of persons age 25 years+, 2011-2015
With a disability, under age 65 years, percent, 2011-2015
Persons  without health insurance, under age 65 years, percent
In civilian labor force, total, percent of population age 16 years+, 2011-201
5
In civilian labor force, female, percent of population age 16 years+, 2011-20
15
Total accommodation and food services sales, 2012 ($1,000)
Total health care and social assistance receipts/revenue, 2012 ($1,000)
Total manufacturers shipments, 2012 ($1,000)
Total merchant wholesaler sales, 2012 ($1,000)
Total retail sales, 2012 ($1,000)
Total retail sales per capita, 2012
Mean travel time to work (minutes), workers age 16 years+, 2011-2015
Median household income (in 2015 dollars), 2011-2015
Per capita income in past 12 months (in 2015 dollars), 2011-2015
Persons in poverty, percent
Total employer establishments, 2015
Total employment, 2015
```

```
Total annual payroll, 2015 ($1,000)
Total employment, percent change, 2014-2015
Total nonemployer establishments, 2015
All firms, 2012
Men-owned firms, 2012
Women-owned firms, 2012
Minority-owned firms, 2012
Nonminority-owned firms, 2012
Veteran-owned firms, 2012
Nonveteran-owned firms, 2012
Population per square mile, 2010
Land area in square miles, 2010
FIPS Code
```

In [430]:

```
#looking into the column names for the gun dataframe
for col in df_gun.columns:
    print(col)
```

```
month
state
permit
permit_recheck
handgun
long_gun
other
multiple
admin
prepawn_handgun
prepawn_long_gun
prepawn_other
redemption_handgun
redemption_long_gun
redemption_other
returned_handgun
returned_long_gun
returned_other
rentals_handgun
rentals_long_gun
private_sale_handgun
private_sale_long_gun
private_sale_other
return_to_seller_handgun
return_to_seller_long_gun
return_to_seller_other
totals
```

After manually analyzing both the datasets carefully in Excel, data for a few of the states is not very consistent in the datasets we are considering.Some of the states aren't present in the Gun data as compared to the Census data.The states include: Guam, District of Columbia, Puerto Rico, Mariana Islands, Virgin Islands. We can eliminate the data so as to avoid any confusions in the analysis.

In [431]:
```
#Removing all the unnecessary state data from the guns data set
print(df_gun.state.nunique())

df_gun = df_gun[df_gun.state != 'Guam']
df_gun = df_gun[df_gun.state != 'District of Columbia']
df_gun = df_gun[df_gun.state != 'Puerto Rico']
df_gun = df_gun[df_gun.state != 'Mariana Islands']
df_gun = df_gun[df_gun.state != 'Virgin Islands']
```

```
55
```

In [432]:
```
df_census.dtypes
```

Out[432]:
```
Population estimates, July 1, 2016,  (V2016)
int64
Population estimates base, April 1, 2010,  (V2016)
int64
Population, percent change - April 1, 2010 (estimates base) to July 1, 2016,
(V2016)    float64
Population, Census, April 1, 2010
int64
Persons under 5 years, percent, July 1, 2016,  (V2016)
float64

...
Veteran-owned firms, 2012
int64
Nonveteran-owned firms, 2012
int64
Population per square mile, 2010
float64
Land area in square miles, 2010
float64
FIPS Code
object
Length: 65, dtype: object
```

In [433]:
```
df_gun.dtypes
```

Out[433]:
```
month                          object
state                          object
permit                        float64
permit_recheck                float64
handgun                       float64
long_gun                      float64
other                         float64
multiple                        int64
admin                         float64
prepawn_handgun               float64
prepawn_long_gun              float64
prepawn_other                 float64
redemption_handgun            float64
redemption_long_gun           float64
redemption_other              float64
returned_handgun              float64
returned_long_gun             float64
returned_other                float64
rentals_handgun               float64
rentals_long_gun              float64
private_sale_handgun          float64
private_sale_long_gun         float64
private_sale_other            float64
return_to_seller_handgun      float64
return_to_seller_long_gun     float64
return_to_seller_other        float64
totals                          int64
dtype: object
```

In [434]:
```
#Check for duplicates in the Guns data
sum(df_gun.duplicated())
```

Out[434]:
```
0
```

In [435]:
```
#Check for duplicates in the Census data
sum(df_census.duplicated())
```

Out[435]:
```
0
```

In [436]:
```
#Checking for NaN values
df_census.isnull().any().any(), sum(df_census.isnull().any())
```

Out[436]:
```
(False, 0)
```

In [437]:
```
df_census.isnull().any()
```

Out[437]:
```
Population estimates, July 1, 2016,  (V2016)
False
Population estimates base, April 1, 2010,  (V2016)
False
Population, percent change - April 1, 2010 (estimates base) to July 1, 2016,
(V2016)    False
Population, Census, April 1, 2010
False
Persons under 5 years, percent, July 1, 2016,  (V2016)
False

...
Veteran-owned firms, 2012
False
Nonveteran-owned firms, 2012
False
Population per square mile, 2010
False
Land area in square miles, 2010
False
FIPS Code
False
Length: 65, dtype: bool
```

In [438]:
```
df_gun.isnull().any().any(), sum(df_gun.isnull().any())
```

Out[438]:
```
(True, 21)
```

In [439]:
```
df_gun.isnull().any()
```

Out[439]:
```
month                        False
state                        False
permit                        True
permit_recheck                True
handgun                      False
long_gun                     False
other                         True
multiple                     False
admin                         True
prepawn_handgun               True
prepawn_long_gun              True
prepawn_other                 True
redemption_handgun            True
redemption_long_gun           True
redemption_other              True
returned_handgun              True
returned_long_gun             True
returned_other                True
rentals_handgun               True
rentals_long_gun              True
private_sale_handgun          True
private_sale_long_gun         True
private_sale_other            True
return_to_seller_handgun      True
return_to_seller_long_gun     True
return_to_seller_other        True
totals                       False
dtype: bool
```

In [440]:
```
df_census.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 50 entries, Alabama to Wyoming
Data columns (total 65 columns):
 #   Column
Non-Null Count  Dtype
---  ------
--------------  -----
 0   Population estimates, July 1, 2016,  (V2016)
50 non-null     int64
 1   Population estimates base, April 1, 2010,  (V2016)
50 non-null     int64
 2   Population, percent change - April 1, 2010 (estimates base) to July 1, 2
016,  (V2016)   50 non-null     float64
 3   Population, Census, April 1, 2010
50 non-null     int64
 4   Persons under 5 years, percent, July 1, 2016,  (V2016)
50 non-null     float64
 5   Persons under 5 years, percent, April 1, 2010
50 non-null     float64
 6   Persons under 18 years, percent, July 1, 2016,  (V2016)
50 non-null     float64
 7   Persons under 18 years, percent, April 1, 2010
50 non-null     float64
 8   Persons 65 years and over, percent,  July 1, 2016,  (V2016)
50 non-null     float64
 9   Persons 65 years and over, percent, April 1, 2010
50 non-null     float64
 10  Female persons, percent,  July 1, 2016,  (V2016)
50 non-null     float64
 11  Female persons, percent, April 1, 2010
50 non-null     float64
 12  White alone, percent, July 1, 2016,  (V2016)
50 non-null     float64
 13  Black or African American alone, percent, July 1, 2016,  (V2016)
50 non-null     float64
 14  American Indian and Alaska Native alone, percent, July 1, 2016,  (V2016)
50 non-null     float64
 15  Asian alone, percent, July 1, 2016,  (V2016)
50 non-null     float64
 16  Native Hawaiian and Other Pacific Islander alone, percent, July 1, 2016,
(V2016)       50 non-null     object
 17  Two or More Races, percent, July 1, 2016,  (V2016)
50 non-null     float64
 18  Hispanic or Latino, percent, July 1, 2016,  (V2016)
50 non-null     float64
 19  White alone, not Hispanic or Latino, percent, July 1, 2016,  (V2016)
50 non-null     float64
 20  Veterans, 2011-2015
50 non-null     int64
 21  Foreign born persons, percent, 2011-2015
50 non-null     float64
 22  Housing units,  July 1, 2016,  (V2016)
50 non-null     int64
 23  Housing units, April 1, 2010
50 non-null     int64
 24  Owner-occupied housing unit rate, 2011-2015
50 non-null     float64
```

```
 25  Median value of owner-occupied housing units, 2011-2015
50 non-null     int64
 26  Median selected monthly owner costs -with a mortgage, 2011-2015
50 non-null     int64
 27  Median selected monthly owner costs -without a mortgage, 2011-2015
50 non-null     int64
 28  Median gross rent, 2011-2015
50 non-null     int64
 29  Building permits, 2016
50 non-null     int64
 30  Households, 2011-2015
50 non-null     int64
 31  Persons per household, 2011-2015
50 non-null     float64
 32  Living in same house 1 year ago, percent of persons age 1 year+, 2011-20
15              50 non-null     float64
 33  Language other than English spoken at home, percent of persons age 5 yea
rs+, 2011-2015  50 non-null     float64
 34  High school graduate or higher, percent of persons age 25 years+, 2011-2
015             50 non-null     float64
 35  Bachelor's degree or higher, percent of persons age 25 years+, 2011-2015
50 non-null     float64
 36  With a disability, under age 65 years, percent, 2011-2015
50 non-null     float64
 37  Persons  without health insurance, under age 65 years, percent
50 non-null     float64
 38  In civilian labor force, total, percent of population age 16 years+, 201
1-2015          50 non-null     float64
 39  In civilian labor force, female, percent of population age 16 years+, 20
11-2015         50 non-null     float64
 40  Total accommodation and food services sales, 2012 ($1,000)
50 non-null     int64
 41  Total health care and social assistance receipts/revenue, 2012 ($1,000)
50 non-null     int64
 42  Total manufacturers shipments, 2012 ($1,000)
50 non-null     object
 43  Total merchant wholesaler sales, 2012 ($1,000)
50 non-null     int64
 44  Total retail sales, 2012 ($1,000)
50 non-null     int64
 45  Total retail sales per capita, 2012
50 non-null     int64
 46  Mean travel time to work (minutes), workers age 16 years+, 2011-2015
50 non-null     float64
 47  Median household income (in 2015 dollars), 2011-2015
50 non-null     int64
 48  Per capita income in past 12 months (in 2015 dollars), 2011-2015
50 non-null     int64
 49  Persons in poverty, percent
50 non-null     float64
 50  Total employer establishments, 2015
50 non-null     int64
 51  Total employment, 2015
50 non-null     int64
 52  Total annual payroll, 2015 ($1,000)
50 non-null     int64
 53  Total employment, percent change, 2014-2015
```

```
50 non-null     object
 54  Total nonemployer establishments, 2015
50 non-null     int64
 55  All firms, 2012
50 non-null     int64
 56  Men-owned firms, 2012
50 non-null     int64
 57  Women-owned firms, 2012
50 non-null     int64
 58  Minority-owned firms, 2012
50 non-null     int64
 59  Nonminority-owned firms, 2012
50 non-null     int64
 60  Veteran-owned firms, 2012
50 non-null     int64
 61  Nonveteran-owned firms, 2012
50 non-null     int64
 62  Population per square mile, 2010
50 non-null     float64
 63  Land area in square miles, 2010
50 non-null     float64
 64  FIPS Code
50 non-null     object
dtypes: float64(31), int64(30), object(4)
memory usage: 25.8+ KB
```

In [441]:
```
df_gun.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 11350 entries, 0 to 12484
Data columns (total 27 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   month                      11350 non-null  object
 1   state                      11350 non-null  object
 2   permit                     11348 non-null  float64
 3   permit_recheck             1000 non-null   float64
 4   handgun                    11350 non-null  float64
 5   long_gun                   11350 non-null  float64
 6   other                      5000 non-null   float64
 7   multiple                   11350 non-null  int64
 8   admin                      11348 non-null  float64
 9   prepawn_handgun            9597 non-null   float64
 10  prepawn_long_gun           9595 non-null   float64
 11  prepawn_other              4650 non-null   float64
 12  redemption_handgun         9600 non-null   float64
 13  redemption_long_gun        9598 non-null   float64
 14  redemption_other           4650 non-null   float64
 15  returned_handgun           2000 non-null   float64
 16  returned_long_gun          1950 non-null   float64
 17  returned_other             1650 non-null   float64
 18  rentals_handgun            900 non-null    float64
 19  rentals_long_gun           750 non-null    float64
 20  private_sale_handgun       2500 non-null   float64
 21  private_sale_long_gun      2500 non-null   float64
 22  private_sale_other         2500 non-null   float64
 23  return_to_seller_handgun   2250 non-null   float64
 24  return_to_seller_long_gun  2500 non-null   float64
 25  return_to_seller_other     2050 non-null   float64
 26  totals                     11350 non-null  int64
dtypes: float64(23), int64(2), object(2)
memory usage: 2.4+ MB
```

Data Cleaning

In [442]:
```
# 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.

# We are starting the cleaning process by checking the duplicates and getting
 rid of them, if any
df_census.drop_duplicates(inplace = True)
sum(df_census.duplicated())
```

Out[442]: 0

In [443]:
```
df_gun.drop_duplicates(inplace = True)
sum(df_gun.duplicated())
```

Out[443]: 0

In [444]:
```
#Now we will fill all the empty / NA values using the fillna function
df_census.fillna('No Records', inplace = True)
df_census.isnull().any()
```

Out[444]:
```
Population estimates, July 1, 2016,  (V2016)
False
Population estimates base, April 1, 2010,  (V2016)
False
Population, percent change - April 1, 2010 (estimates base) to July 1, 2016,
(V2016)    False
Population, Census, April 1, 2010
False
Persons under 5 years, percent, July 1, 2016,  (V2016)
False

...
Veteran-owned firms, 2012
False
Nonveteran-owned firms, 2012
False
Population per square mile, 2010
False
Land area in square miles, 2010
False
FIPS Code
False
Length: 65, dtype: bool
```

In [445]:
```
df_gun.fillna(df_gun.mean(), inplace = True )
df_gun.isnull().any()
```

Out[445]:
```
month                        False
state                        False
permit                       False
permit_recheck               False
handgun                      False
long_gun                     False
other                        False
multiple                     False
admin                        False
prepawn_handgun              False
prepawn_long_gun             False
prepawn_other                False
redemption_handgun           False
redemption_long_gun          False
redemption_other             False
returned_handgun             False
returned_long_gun            False
returned_other               False
rentals_handgun              False
rentals_long_gun             False
private_sale_handgun         False
private_sale_long_gun        False
private_sale_other           False
return_to_seller_handgun     False
return_to_seller_long_gun    False
return_to_seller_other       False
totals                       False
dtype: bool
```

In [446]:
```
df_gun.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 11350 entries, 0 to 12484
Data columns (total 27 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   month                      11350 non-null  object
 1   state                      11350 non-null  object
 2   permit                     11350 non-null  float64
 3   permit_recheck             11350 non-null  float64
 4   handgun                    11350 non-null  float64
 5   long_gun                   11350 non-null  float64
 6   other                      11350 non-null  float64
 7   multiple                   11350 non-null  int64
 8   admin                      11350 non-null  float64
 9   prepawn_handgun            11350 non-null  float64
 10  prepawn_long_gun           11350 non-null  float64
 11  prepawn_other              11350 non-null  float64
 12  redemption_handgun         11350 non-null  float64
 13  redemption_long_gun        11350 non-null  float64
 14  redemption_other           11350 non-null  float64
 15  returned_handgun           11350 non-null  float64
 16  returned_long_gun          11350 non-null  float64
 17  returned_other             11350 non-null  float64
 18  rentals_handgun            11350 non-null  float64
 19  rentals_long_gun           11350 non-null  float64
 20  private_sale_handgun       11350 non-null  float64
 21  private_sale_long_gun      11350 non-null  float64
 22  private_sale_other         11350 non-null  float64
 23  return_to_seller_handgun   11350 non-null  float64
 24  return_to_seller_long_gun  11350 non-null  float64
 25  return_to_seller_other     11350 non-null  float64
 26  totals                     11350 non-null  int64
dtypes: float64(23), int64(2), object(2)
memory usage: 2.4+ MB
```

In [447]:
```
#Noticing the names of the column names, we can bring the column names to a si
ngle format

df_census.rename(columns = lambda x:x.lower(), inplace = True)
df_census
```

Out[447]:

State	population estimates, july 1, 2016, (v2016)	population estimates base, april 1, 2010, (v2016)	population, percent change - april 1, 2010 (estimates base) to july 1, 2016, (v2016)	population, census, april 1, 2010	persons under 5 years, percent, july 1, 2016, (v2016)	persons under 5 years, percent, april 1, 2010	persons under 18 years, percent, july 1, 2016, (v2016)	pers u y per ap
Alabama	4863300	4780131	0.017	4779736	0.060	0.064	0.226	(
Alaska	741894	710249	0.045	710231	0.073	0.076	0.252	(
Arizona	6931071	6392301	0.084	6392017	0.063	0.071	0.235	(
Arkansas	2988248	2916025	0.025	2915918	0.064	0.068	0.236	(
California	39250017	37254522	0.054	37253956	0.063	0.068	0.232	(
Colorado	5540545	5029324	0.102	5029196	0.061	0.068	0.228	(
Connecticut	3576452	3574114	0.001	3574097	0.052	0.057	0.211	(
Delaware	952065	897936	0.060	897934	0.058	0.062	0.215	(
Florida	20612439	18804592	0.096	18801310	0.055	0.057	0.201	(
Georgia	10310371	9688680	0.064	9687653	0.064	0.071	0.244	(
Hawaii	1428557	1360301	0.050	1360301	0.064	0.064	0.216	(
Idaho	1683140	1567650	0.074	1567582	0.068	0.078	0.260	(
Illinois	12801539	12831574	-0.002	12830632	0.060	0.065	0.229	(
Indiana	6633053	6484136	0.023	6483802	0.064	0.067	0.238	(
Iowa	3134693	3046869	0.029	3046355	0.064	0.066	0.233	(
Kansas	2907289	2853129	0.019	2853118	0.067	0.072	0.246	(
Kentucky	4436974	4339344	0.022	4339367	0.062	0.065	0.228	(
Louisiana	4681666	4533479	0.033	4533372	0.066	0.069	0.238	(
Maine	1331479	1328364	0.002	1328361	0.049	0.052	0.191	(
Maryland	6016447	5773786	0.042	5773552	0.061	0.063	0.224	(
Massachusetts	6811779	6547813	0.040	6547629	0.053	0.056	0.202	(
Michigan	9928300	9884129	0.004	9883640	0.058	0.060	0.221	(
Minnesota	5519952	5303924	0.041	5303925	0.064	0.067	0.233	(
Mississippi	2988726	2968103	0.007	2967297	0.063	0.071	0.241	(
Missouri	6093000	5988928	0.017	5988927	0.061	0.065	0.228	(
Montana	1042520	989414	0.054	989415	0.060	0.063	0.218	(
Nebraska	1907116	1826334	0.044	1826341	0.070	0.072	0.248	(
Nevada	2940058	2700691	0.089	2700551	0.063	0.069	0.230	(
New Hampshire	1334795	1316461	0.014	1316470	0.048	0.053	0.195	(

State	population estimates, july 1, 2016, (v2016)	population estimates base, april 1, 2010, (v2016)	population, percent change - april 1, 2010 (estimates base) to july 1, 2016, (v2016)	population, census, april 1, 2010	persons under 5 years, percent, july 1, 2016, (v2016)	persons under 5 years, percent, april 1, 2010	persons under 18 years, percent, july 1, 2016, (v2016)	pers u y perc ap
New Jersey	8944469	8791953	0.017	8791894	0.058	0.062	0.222	(
New Mexico	2081015	2059198	0.011	2059179	0.062	0.070	0.236	(
New York	19745289	19378110	0.019	19378102	0.059	0.060	0.212	(
North Carolina	10146788	9535688	0.064	9535483	0.060	0.066	0.227	(
North Dakota	757952	672591	0.127	672591	0.073	0.066	0.233	(
Ohio	11614373	11536727	0.007	11536504	0.060	0.062	0.225	(
Oklahoma	3923561	3751615	0.046	3751351	0.068	0.070	0.245	(
Oregon	4093465	3831072	0.068	3831074	0.058	0.062	0.212	(
Pennsylvania	12784227	12702857	0.006	12702379	0.056	0.057	0.209	(
Rhode Island	1056426	1052940	0.003	1052567	0.052	0.055	0.197	(
South Carolina	4961119	4625410	0.073	4625364	0.059	0.065	0.221	(
South Dakota	865454	814195	0.063	814180	0.071	0.073	0.246	(
Tennessee	6651194	6346298	0.048	6346105	0.061	0.064	0.226	(
Texas	27862596	25146100	0.108	25145561	0.072	0.077	0.262	(
Utah	3051217	2763888	0.104	2763885	0.083	0.095	0.302	(
Vermont	624594	625741	-0.002	625741	0.049	0.051	0.190	(
Virginia	8411808	8001041	0.051	8001024	0.061	0.064	0.222	(
Washington	7288000	6724545	0.084	6724540	0.062	0.065	0.224	(
West Virginia	1831102	1853011	-0.012	1852994	0.055	0.056	0.205	(
Wisconsin	5778708	5687289	0.016	5686986	0.058	0.063	0.223	(
Wyoming	585501	563767	0.039	563626	0.065	0.071	0.237	(

50 rows × 65 columns

In [448]:
```
df_gun
```

Out[448]:

	month	state	permit	permit_recheck	handgun	long_gun	other	multiple	ad
0	2017-09	Alabama	16717.0	0.000	5734.0	6320.0	221.0000	317	
1	2017-09	Alaska	209.0	2.000	2320.0	2930.0	219.0000	160	
2	2017-09	Arizona	5069.0	382.000	11063.0	7946.0	920.0000	631	
3	2017-09	Arkansas	2935.0	632.000	4347.0	6063.0	165.0000	366	[illegible]
4	2017-09	California	57839.0	0.000	37165.0	24581.0	2984.0000	0	
...	...	...	...	...	...	...	...	...	
12480	1998-11	Virginia	0.0	1282.552	14.0	2.0	396.0524	8	
12481	1998-11	Washington	1.0	1282.552	65.0	286.0	396.0524	8	
12482	1998-11	West Virginia	3.0	1282.552	149.0	251.0	396.0524	5	
12483	1998-11	Wisconsin	0.0	1282.552	25.0	214.0	396.0524	2	
12484	1998-11	Wyoming	8.0	1282.552	45.0	49.0	396.0524	5	

11350 rows × 27 columns

In [449]:
```
#For dealing with dates, we will bring them to the standard format using the datetime package
df_gun.month = pd.to_datetime(df_gun['month'], errors='coerce')


df_gun
```

Out[449]:

	month	state	permit	permit_recheck	handgun	long_gun	other	multiple	ad
0	2017-09-01	Alabama	16717.0	0.000	5734.0	6320.0	221.0000	317	
1	2017-09-01	Alaska	209.0	2.000	2320.0	2930.0	219.0000	160	
2	2017-09-01	Arizona	5069.0	382.000	11063.0	7946.0	920.0000	631	
3	2017-09-01	Arkansas	2935.0	632.000	4347.0	6063.0	165.0000	366	[illegible]
4	2017-09-01	California	57839.0	0.000	37165.0	24581.0	2984.0000	0	
...	...	...	...	...	...	...	...	...	
12480	1998-11-01	Virginia	0.0	1282.552	14.0	2.0	396.0524	8	
12481	1998-11-01	Washington	1.0	1282.552	65.0	286.0	396.0524	8	
12482	1998-11-01	West Virginia	3.0	1282.552	149.0	251.0	396.0524	5	
12483	1998-11-01	Wisconsin	0.0	1282.552	25.0	214.0	396.0524	2	
12484	1998-11-01	Wyoming	8.0	1282.552	45.0	49.0	396.0524	5	

11350 rows × 27 columns

Exploratory Data Analysis

Research Question 1: Which state has the highest total gun purchases in April 2005 and April 2010?

In [450]:
```
# Use this, and more code cells, to explore your data. Don't forget to add
#   Markdown cells to document your observations and findings.

#We will create a new df with all the columns that will be required to perform the analysis

df_gun_q1 = df_gun[['month','state','totals']]

df_gun_q1
```

Out[450]:

	month	state	totals
0	2017-09-01	Alabama	32019
1	2017-09-01	Alaska	6303
2	2017-09-01	Arizona	28394
3	2017-09-01	Arkansas	17747
4	2017-09-01	California	123506
...	...	...	...
12480	1998-11-01	Virginia	24
12481	1998-11-01	Washington	361
12482	1998-11-01	West Virginia	408
12483	1998-11-01	Wisconsin	241
12484	1998-11-01	Wyoming	107

11350 rows × 3 columns

In [451]:
```
df_gun_q1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 11350 entries, 0 to 12484
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   month   11350 non-null  datetime64[ns]
 1   state   11350 non-null  object
 2   totals  11350 non-null  int64
dtypes: datetime64[ns](1), int64(1), object(1)
memory usage: 354.7+ KB
```

In [452]:
```
#Now lets extract just the April 2000 and April 2010 data
q1_2005 = df_gun_q1[df_gun_q1['month'] == "2005-04-01"]
display(q1_2005.head())

q1_2010 = df_gun_q1[df_gun_q1['month'] == "2010-04-01"]
display(q1_2010.head())
```

	month	state	totals
8195	2005-04-01	Alabama	14099
8196	2005-04-01	Alaska	3843
8197	2005-04-01	Arizona	13659
8198	2005-04-01	Arkansas	10027
8199	2005-04-01	California	49679

	month	state	totals
4895	2010-04-01	Alabama	20791
4896	2010-04-01	Alaska	6411
4897	2010-04-01	Arizona	16578
4898	2010-04-01	Arkansas	14563
4899	2010-04-01	California	80750

In [453]:
```
print(q1_2005.describe())
print('\n')
print(q1_2010.describe())
```

```
             totals
count     50.000000
mean   12932.920000
std    11216.245617
min      659.000000
25%     5188.750000
50%     9859.000000
75%    16423.750000
max    49679.000000


            totals
count      50.0000
mean    24517.7400
std     34280.1673
min       963.0000
25%      7197.5000
50%     15242.5000
75%     26335.5000
max    211261.0000
```

In [454]:

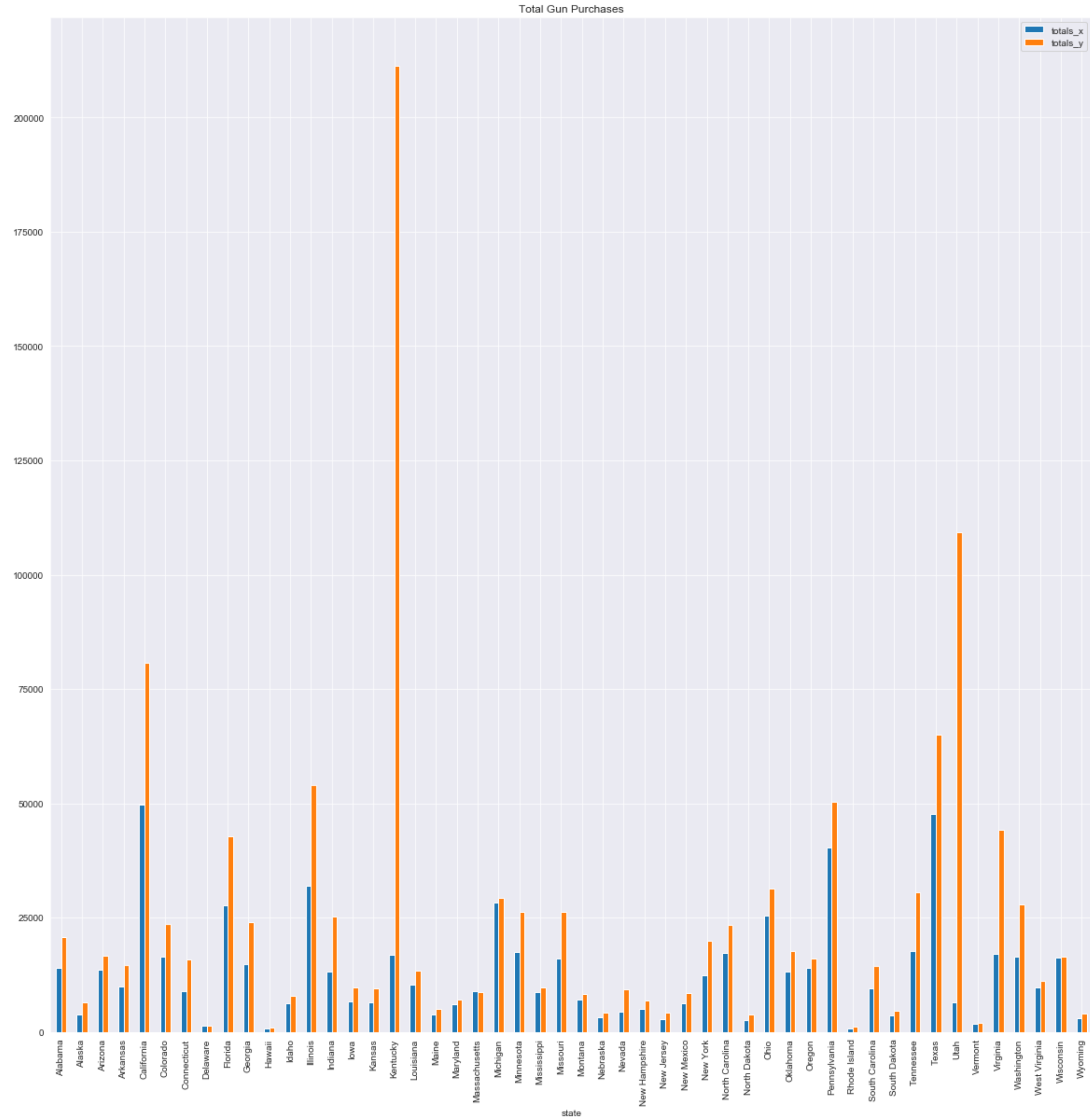
```
#We will now merge both of the df created into a single df usig merge to do fu
rther analysis

q1_combined = q1_2005.merge(q1_2010, on = 'state', how = 'outer')
q1_combined.head()
```

Out[454]:

	month_x	state	totals_x	month_y	totals_y
0	2005-04-01	Alabama	14099	2010-04-01	20791
1	2005-04-01	Alaska	3843	2010-04-01	6411
2	2005-04-01	Arizona	13659	2010-04-01	16578
3	2005-04-01	Arkansas	10027	2010-04-01	14563
4	2005-04-01	California	49679	2010-04-01	80750

In [456]:
```
q1_combined.plot(x = 'state',title = 'Total Gun Purchases',
                 y = ['totals_x','totals_y'],kind = 'bar', figsize = (20,20));
```



In [457]:
```
kent_2005 = q1_combined[q1_combined['state'] == 'Kentucky']['totals_x']
kent_2010 = q1_combined[q1_combined['state'] == 'Kentucky']['totals_y']

print('The total number of firearms sold in state of Ketucky in 2000 is:',kent
_2005);
print('The total number of firearms sold in state of Ketucky in 2010 is:',kent
_2010);
```

```
The total number of firearms sold in state of Ketucky in 2000 is: 16    16961
Name: totals_x, dtype: int64
The total number of firearms sold in state of Ketucky in 2010 is: 16    21126
1
Name: totals_y, dtype: int64
```

Insight:

> After looking through the overall firearms sold in the year of 2005 and compared to the year of 2010, there has been a very significant increase in the 5-year span. There has been an increase of 194,300 firearms in 2010 compared to 2005

Research Question 2:

What is per capita firearm sales for all states in April 2010 vs July 2016?

In [458]:
```
# To perform analysis on the question above, we will require some data from bo
th the dataframes

df_q2 = df_gun[['state','month','totals']]


df_q2 = df_q2[df_q2['month'] == '2016-07-01']

df_q2
```

Out[458]:

	state	month	totals
770	Alabama	2016-07-01	48927
771	Alaska	2016-07-01	6793
772	Arizona	2016-07-01	34496
773	Arkansas	2016-07-01	19378
774	California	2016-07-01	190218
775	Colorado	2016-07-01	43094
776	Connecticut	2016-07-01	29755
777	Delaware	2016-07-01	4494
779	Florida	2016-07-01	125208
780	Georgia	2016-07-01	49183
782	Hawaii	2016-07-01	1565
783	Idaho	2016-07-01	12154
784	Illinois	2016-07-01	168227
785	Indiana	2016-07-01	88340
786	Iowa	2016-07-01	11937
787	Kansas	2016-07-01	14140
788	Kentucky	2016-07-01	363085
789	Louisiana	2016-07-01	41063
790	Maine	2016-07-01	7702
792	Maryland	2016-07-01	12228
793	Massachusetts	2016-07-01	20480
794	Michigan	2016-07-01	40142
795	Minnesota	2016-07-01	43368
796	Mississippi	2016-07-01	21907
797	Missouri	2016-07-01	46637
798	Montana	2016-07-01	9869
799	Nebraska	2016-07-01	5429
800	Nevada	2016-07-01	11785
801	New Hampshire	2016-07-01	13068
802	New Jersey	2016-07-01	10074
803	New Mexico	2016-07-01	12219
804	New York	2016-07-01	29513
805	North Carolina	2016-07-01	44123
806	North Dakota	2016-07-01	5470
807	Ohio	2016-07-01	63148

	state	month	totals
808	Oklahoma	2016-07-01	25946
809	Oregon	2016-07-01	24813
810	Pennsylvania	2016-07-01	86137
812	Rhode Island	2016-07-01	2368
813	South Carolina	2016-07-01	32730
814	South Dakota	2016-07-01	7406
815	Tennessee	2016-07-01	57653
816	Texas	2016-07-01	127207
817	Utah	2016-07-01	17608
818	Vermont	2016-07-01	2674
820	Virginia	2016-07-01	43574
821	Washington	2016-07-01	47887
822	West Virginia	2016-07-01	16791
823	Wisconsin	2016-07-01	38922
824	Wyoming	2016-07-01	4585

In [460]:
```
#Drop the indexes from the between and get new indexes
df_q2.reset_index(drop = True ,inplace = True)

df_q2
```

Out[460]:

	state	month	totals
0	Alabama	2016-07-01	48927
1	Alaska	2016-07-01	6793
2	Arizona	2016-07-01	34496
3	Arkansas	2016-07-01	19378
4	California	2016-07-01	190218
5	Colorado	2016-07-01	43094
6	Connecticut	2016-07-01	29755
7	Delaware	2016-07-01	4494
8	Florida	2016-07-01	125208
9	Georgia	2016-07-01	49183
10	Hawaii	2016-07-01	1565
11	Idaho	2016-07-01	12154
12	Illinois	2016-07-01	168227
13	Indiana	2016-07-01	88340
14	Iowa	2016-07-01	11937
15	Kansas	2016-07-01	14140
16	Kentucky	2016-07-01	363085
17	Louisiana	2016-07-01	41063
18	Maine	2016-07-01	7702
19	Maryland	2016-07-01	12228
20	Massachusetts	2016-07-01	20480
21	Michigan	2016-07-01	40142
22	Minnesota	2016-07-01	43368
23	Mississippi	2016-07-01	21907
24	Missouri	2016-07-01	46637
25	Montana	2016-07-01	9869
26	Nebraska	2016-07-01	5429
27	Nevada	2016-07-01	11785
28	New Hampshire	2016-07-01	13068
29	New Jersey	2016-07-01	10074
30	New Mexico	2016-07-01	12219
31	New York	2016-07-01	29513
32	North Carolina	2016-07-01	44123
33	North Dakota	2016-07-01	5470
34	Ohio	2016-07-01	63148

	state	month	totals
35	Oklahoma	2016-07-01	25946
36	Oregon	2016-07-01	24813
37	Pennsylvania	2016-07-01	86137
38	Rhode Island	2016-07-01	2368
39	South Carolina	2016-07-01	32730
40	South Dakota	2016-07-01	7406
41	Tennessee	2016-07-01	57653
42	Texas	2016-07-01	127207
43	Utah	2016-07-01	17608
44	Vermont	2016-07-01	2674
45	Virginia	2016-07-01	43574
46	Washington	2016-07-01	47887
47	West Virginia	2016-07-01	16791
48	Wisconsin	2016-07-01	38922
49	Wyoming	2016-07-01	4585

In [473]:

```
#We are adding the Population of July 2016 in the newly created df
df_q2['population_july2016'] = df_census['population estimates, july 1, 2016, (v2016)'].values

df_q2.head()
```

Out[473]:

	state	month	totals	population_july2016
0	Alabama	2016-07-01	48927	4863300
1	Alaska	2016-07-01	6793	741894
2	Arizona	2016-07-01	34496	6931071
3	Arkansas	2016-07-01	19378	2988248
4	California	2016-07-01	190218	39250017

In [466]:
```
#we are performing the same steps to create a new dataframe for April 2010
df_q2_2010 = df_gun[['state','month','totals']]

df_q2_2010 = df_q2_2010[df_q2_2010['month']=='2010-04-01']

df_q2_2010.reset_index(drop=True, inplace=True)

df_q2_2010
```

Out[466]:

	state	month	totals
0	Alabama	2010-04-01	20791
1	Alaska	2010-04-01	6411
2	Arizona	2010-04-01	16578
3	Arkansas	2010-04-01	14563
4	California	2010-04-01	80750
5	Colorado	2010-04-01	23609
6	Connecticut	2010-04-01	15922
7	Delaware	2010-04-01	1439
8	Florida	2010-04-01	42794
9	Georgia	2010-04-01	24065
10	Hawaii	2010-04-01	963
11	Idaho	2010-04-01	7814
12	Illinois	2010-04-01	53929
13	Indiana	2010-04-01	25232
14	Iowa	2010-04-01	9720
15	Kansas	2010-04-01	9529
16	Kentucky	2010-04-01	211261
17	Louisiana	2010-04-01	13373
18	Maine	2010-04-01	5073
19	Maryland	2010-04-01	6992
20	Massachusetts	2010-04-01	8748
21	Michigan	2010-04-01	29383
22	Minnesota	2010-04-01	26351
23	Mississippi	2010-04-01	9702
24	Missouri	2010-04-01	26289
25	Montana	2010-04-01	8367
26	Nebraska	2010-04-01	4141
27	Nevada	2010-04-01	9294
28	New Hampshire	2010-04-01	6911
29	New Jersey	2010-04-01	4215
30	New Mexico	2010-04-01	8599
31	New York	2010-04-01	19906
32	North Carolina	2010-04-01	23378
33	North Dakota	2010-04-01	3726
34	Ohio	2010-04-01	31312

	state	month	totals
35	Oklahoma	2010-04-01	17750
36	Oregon	2010-04-01	16031
37	Pennsylvania	2010-04-01	50249
38	Rhode Island	2010-04-01	1199
39	South Carolina	2010-04-01	14441
40	South Dakota	2010-04-01	4561
41	Tennessee	2010-04-01	30453
42	Texas	2010-04-01	65012
43	Utah	2010-04-01	109391
44	Vermont	2010-04-01	2053
45	Virginia	2010-04-01	44137
46	Washington	2010-04-01	27816
47	West Virginia	2010-04-01	11180
48	Wisconsin	2010-04-01	16471
49	Wyoming	2010-04-01	4013

In [472]:

```
df_q2_2010['population_april2010'] = df_census['population estimates base, apr
il 1, 2010,  (v2016)'].values

df_q2_2010.head()
```

Out[472]:

	state	month	totals	population_april2010
0	Alabama	2010-04-01	20791	4780131
1	Alaska	2010-04-01	6411	710249
2	Arizona	2010-04-01	16578	6392301
3	Arkansas	2010-04-01	14563	2916025
4	California	2010-04-01	80750	37254522

In [477]:
```
#We will now try to calculate the per capita income percentage by summing all
 the states
#and then try showing a graph of the changed from 2010 till 2016


df_q2['totals_2010'] = df_q2_2010['totals'].values
df_q2['population_april2010'] = df_q2_2010['population_april2010'].values
df_q2
```

Out[477]:

	state	month	totals	population_july2016	totals_2010	population_april2010
0	Alabama	2016-07-01	48927	4863300	20791	4780131
1	Alaska	2016-07-01	6793	741894	6411	710249
2	Arizona	2016-07-01	34496	6931071	16578	6392301
3	Arkansas	2016-07-01	19378	2988248	14563	2916025
4	California	2016-07-01	190218	39250017	80750	37254522
5	Colorado	2016-07-01	43094	5540545	23609	5029324
6	Connecticut	2016-07-01	29755	3576452	15922	3574114
7	Delaware	2016-07-01	4494	952065	1439	897936
8	Florida	2016-07-01	125208	20612439	42794	18804592
9	Georgia	2016-07-01	49183	10310371	24065	9688680
10	Hawaii	2016-07-01	1565	1428557	963	1360301
11	Idaho	2016-07-01	12154	1683140	7814	1567650
12	Illinois	2016-07-01	168227	12801539	53929	12831574
13	Indiana	2016-07-01	88340	6633053	25232	6484136
14	Iowa	2016-07-01	11937	3134693	9720	3046869
15	Kansas	2016-07-01	14140	2907289	9529	2853129
16	Kentucky	2016-07-01	363085	4436974	211261	4339344
17	Louisiana	2016-07-01	41063	4681666	13373	4533479
18	Maine	2016-07-01	7702	1331479	5073	1328364
19	Maryland	2016-07-01	12228	6016447	6992	5773786
20	Massachusetts	2016-07-01	20480	6811779	8748	6547813
21	Michigan	2016-07-01	40142	9928300	29383	9884129
22	Minnesota	2016-07-01	43368	5519952	26351	5303924
23	Mississippi	2016-07-01	21907	2988726	9702	2968103
24	Missouri	2016-07-01	46637	6093000	26289	5988928
25	Montana	2016-07-01	9869	1042520	8367	989414
26	Nebraska	2016-07-01	5429	1907116	4141	1826334
27	Nevada	2016-07-01	11785	2940058	9294	2700691
28	New Hampshire	2016-07-01	13068	1334795	6911	1316461
29	New Jersey	2016-07-01	10074	8944469	4215	8791953
30	New Mexico	2016-07-01	12219	2081015	8599	2059198
31	New York	2016-07-01	29513	19745289	19906	19378110
32	North Carolina	2016-07-01	44123	10146788	23378	9535688
33	North Dakota	2016-07-01	5470	757952	3726	672591
34	Ohio	2016-07-01	63148	11614373	31312	11536727

	state	month	totals	population_july2016	totals_2010	population_april2010
35	Oklahoma	2016-07-01	25946	3923561	17750	3751615
36	Oregon	2016-07-01	24813	4093465	16031	3831072
37	Pennsylvania	2016-07-01	86137	12784227	50249	12702857
38	Rhode Island	2016-07-01	2368	1056426	1199	1052940
39	South Carolina	2016-07-01	32730	4961119	14441	4625410
40	South Dakota	2016-07-01	7406	865454	4561	814195
41	Tennessee	2016-07-01	57653	6651194	30453	6346298
42	Texas	2016-07-01	127207	27862596	65012	25146100
43	Utah	2016-07-01	17608	3051217	109391	2763888
44	Vermont	2016-07-01	2674	624594	2053	625741
45	Virginia	2016-07-01	43574	8411808	44137	8001041
46	Washington	2016-07-01	47887	7288000	27816	6724545
47	West Virginia	2016-07-01	16791	1831102	11180	1853011
48	Wisconsin	2016-07-01	38922	5778708	16471	5687289
49	Wyoming	2016-07-01	4585	585501	4013	563767

In [501]:

```
#Now let us calculate the capita of 2010 and 2016

capita_2010 = df_q2.totals_2010.sum()/ df_q2.population_april2010.sum()
print('The per capita firearms for April 2010 is:', capita_2010)

capita_2016 = df_q2.totals.sum()/ df_q2.population_july2016.sum()
print('The per capita firearms for July 2016 is:', capita_2016)
```

```
The per capita firearms for April 2010 is: 0.003978133320178106
The per capita firearms for July 2016 is: 0.006777933902633841
```

In [502]:

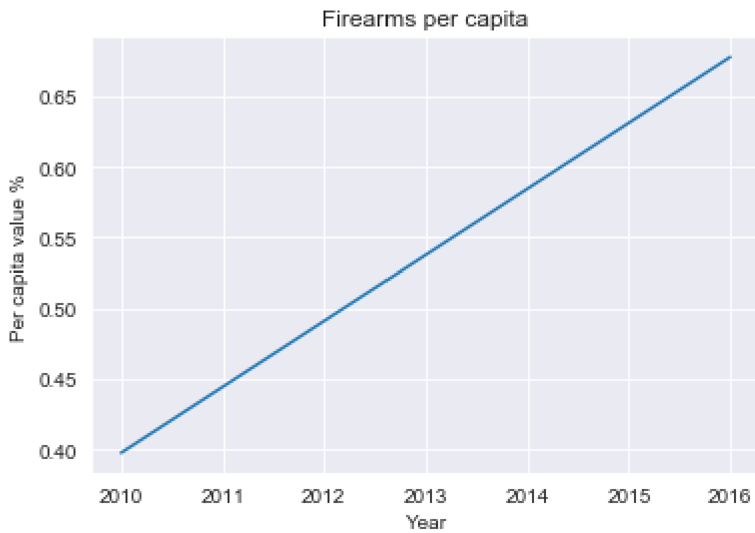
```
#To make it easier to perform calculations and analysis, we'll multiply both t
he variables by 100
capita_2010 = capita_2010*100
capita_2016 = capita_2016*100

print('New capita_2010:',capita_2010)
print('New capita_2016:',capita_2016)
```

```
New capita_2010: 0.3978133320178106
New capita_2016: 0.677793390263384
```

In [504]:
```
#now we plot the graph to note the changes

years = [2010,2016]
per_capita = [capita_2010,capita_2016]
plt.plot(years,per_capita);
plt.title('Firearms per capita');
plt.xlabel('Year');
plt.ylabel('Per capita value %');
```



Insights:

> The graph shows us the increase in per capita ownership of firearms. The per-capita value (percentage value) in April 2010 is 0.38 and it goes on to 0.678 in July 2016.

Conclusions

A few conclusions could be made using the questions asked to the database:

1. After comparing the gun data between the years 2005 and 2010, there has been a significant increase in firearm purchases. We compared the data between April 2005 and April 2010, coming to a conclusion that the state of Kentucky has the highest increase with a number of 194,300 firearms in the 5-year span.
2. The second graph helps us in understanding the increase in per capita ownership of firearms. The per-capita value (percentage value) in April 2010 is 0.38 and it goes on to 0.678 in July 2016.

Number of things to be considered for the question 2 visual:

> 1. This chart is a nation-wide chart and does give us a birds-eye view.
> 2. At state level this graph's slope will vary and hence the insights cannot be generalized.
> 3. Also, we are using 2 point of time (i.e. April 2010 and July 2016), whereas we have no information about how this per capita relationship will vary over the 6 years between 2010 and 2016. Again, this insight cannot be generalized.

In [1]:
```
from subprocess import call
call(['python', '-m', 'nbconvert', 'Investigate_a_Dataset.ipynb'])
```

Out[1]: 0

In [ ]: