FBI Gun

October 26, 2020

1 Investigate a Dataset: FBI Gun Data

1.1 Table of Contants

1.2 1. Introduction

The data comes from the FBI's National Instant Criminal Background Check System. The NICS is used by 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. he data has been supplemented with state level data from https://www.google.com/url?q=https://www.census.gov/&sa=D&ust=1532469042127000.

1.2.1 1.1 The questions handled in this project:

1. What census data is most associated with high gun per capita?

2. Which states have had the highest growth in gun registrations?

3. What is the overall trend of gun purchases?

4. which year had the most checks, and by which state. and which had the least?

1.3 2. Data Wrangling

In this part I'm going to import all the needed libraries, and the two datasets (Gun dataset and Census dataset). At first, I'm going to try and get to know my dataset, how many variables are there? what are the dimensions? what are the types that each dataset contains?. Then, I'm going to check whether the datasets need cleaning and trimming or not.

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.ticker import FuncFormatter
%matplotlib inline
```

1.3.1 2.1 Getting to know the Gun dataset

[2]:
```
## loading the Gun dataset and displaying the first 10 rows
url='https://github.com/BuzzFeedNews/nics-firearm-background-checks/blob/master/
 ↪data/nics-firearm-background-checks.csv?raw=true'
gun1=pd.read_csv(url)
gun1.head(10)
```

[2]:

	month	state	permit	permit_recheck	handgun	long_gun	\
0	2020-09	Alabama	33228.0	642.0	23455.0	17369.0	
1	2020-09	Alaska	388.0	2.0	3275.0	3333.0	
2	2020-09	Arizona	8786.0	1198.0	23996.0	12094.0	
3	2020-09	Arkansas	3686.0	554.0	9214.0	8003.0	
4	2020-09	California	32998.0	0.0	61258.0	36638.0	
5	2020-09	Colorado	10309.0	24.0	24260.0	15873.0	
6	2020-09	Connecticut	9845.0	416.0	6101.0	2284.0	
7	2020-09	Delaware	470.0	0.0	3501.0	1909.0	
8	2020-09	District of Columbia	901.0	0.0	531.0	11.0	
9	2020-09	Florida	34496.0	0.0	76685.0	29317.0	

	other	multiple	admin	prepawn_handgun	…	returned_other	\
0	1633.0	981	0.0	35.0	…	0.0	
1	345.0	201	0.0	1.0	…	0.0	
2	1963.0	1873	0.0	16.0	…	0.0	
3	505.0	383	10.0	8.0	…	0.0	
4	7815.0	0	0.0	0.0	…	0.0	
5	1919.0	1721	0.0	0.0	…	2.0	
6	1443.0	0	1.0	0.0	…	0.0	
7	176.0	141	0.0	0.0	…	0.0	

```
8     0.0         1    0.0             0.0  …            25.0
9  5523.0      3174    0.0            13.0  …            26.0

   rentals_handgun  rentals_long_gun  private_sale_handgun  \
0              0.0               0.0                  30.0
1              0.0               0.0                   8.0
2              0.0               0.0                  39.0
3              0.0               0.0                   3.0
4              0.0               0.0                   0.0
5              0.0               0.0                   0.0
6              0.0               0.0                   1.0
7              0.0               0.0                  84.0
8              0.0               0.0                   0.0
9              0.0               0.0                 482.0

   private_sale_long_gun  private_sale_other  return_to_seller_handgun  \
0                   19.0                 8.0                       1.0
1                   16.0                 2.0                       1.0
2                   13.0                 5.0                       0.0
3                   12.0                 3.0                       0.0
4                    0.0                 0.0                       0.0
5                    0.0                 0.0                       0.0
6                    0.0                 0.0                       0.0
7                   37.0                 4.0                       1.0
8                    0.0                 0.0                       0.0
9                  268.0                65.0                      42.0

   return_to_seller_long_gun  return_to_seller_other  totals
0                        2.0                     0.0   80478
1                        1.0                     0.0    7897
2                        0.0                     0.0   51287
3                        0.0                     0.0   24043
4                        0.0                     0.0  139313
5                        0.0                     0.0   54479
6                        0.0                     0.0   20091
7                        0.0                     0.0    6381
8                        0.0                     0.0    1469
9                       31.0                     2.0  154982

[10 rows x 27 columns]
```

[3]:
```
#Displaing the last 10 rows
gun1.tail(10)
```

[3]:
```
         month           state  permit  permit_recheck  handgun  long_gun  \
14455  1998-11       Tennessee     0.0             NaN     19.0      85.0
14456  1998-11           Texas     0.0             NaN   1384.0    1349.0
```

```
14457  1998-11            Utah     0.0            NaN     98.0     169.0
14458  1998-11         Vermont     0.0            NaN     23.0      35.0
14459  1998-11  Virgin Islands     0.0            NaN      0.0       0.0
14460  1998-11        Virginia     0.0            NaN     14.0       2.0
14461  1998-11      Washington     1.0            NaN     65.0     286.0
14462  1998-11   West Virginia     3.0            NaN    149.0     251.0
14463  1998-11       Wisconsin     0.0            NaN     25.0     214.0
14464  1998-11         Wyoming     8.0            NaN     45.0      49.0

       other  multiple  admin  prepawn_handgun  …  returned_other  \
14455    NaN         3    0.0              NaN  …             NaN
14456    NaN        60    1.0              NaN  …             NaN
14457    NaN         0    0.0              NaN  …             NaN
14458    NaN         0    1.0              NaN  …             NaN
14459    NaN         0    0.0              NaN  …             NaN
14460    NaN         8    0.0              NaN  …             NaN
14461    NaN         8    1.0              NaN  …             NaN
14462    NaN         5    0.0              NaN  …             NaN
14463    NaN         2    0.0              NaN  …             NaN
14464    NaN         5    0.0              NaN  …             NaN

       rentals_handgun  rentals_long_gun  private_sale_handgun  \
14455              NaN               NaN                   NaN
14456              NaN               NaN                   NaN
14457              NaN               NaN                   NaN
14458              NaN               NaN                   NaN
14459              NaN               NaN                   NaN
14460              NaN               NaN                   NaN
14461              NaN               NaN                   NaN
14462              NaN               NaN                   NaN
14463              NaN               NaN                   NaN
14464              NaN               NaN                   NaN

       private_sale_long_gun  private_sale_other  return_to_seller_handgun  \
14455                    NaN                 NaN                       NaN
14456                    NaN                 NaN                       NaN
14457                    NaN                 NaN                       NaN
14458                    NaN                 NaN                       NaN
14459                    NaN                 NaN                       NaN
14460                    NaN                 NaN                       NaN
14461                    NaN                 NaN                       NaN
14462                    NaN                 NaN                       NaN
14463                    NaN                 NaN                       NaN
14464                    NaN                 NaN                       NaN

       return_to_seller_long_gun  return_to_seller_other  totals
14455                        NaN                     NaN     107
```

```
14456                          NaN                          NaN    2794
14457                          NaN                          NaN     267
14458                          NaN                          NaN      59
14459                          NaN                          NaN       0
14460                          NaN                          NaN      24
14461                          NaN                          NaN     361
14462                          NaN                          NaN     408
14463                          NaN                          NaN     241
14464                          NaN                          NaN     107

[10 rows x 27 columns]
```

The past two cells make me familiar with most of the variables and entries. It is a quick look at the dataset.

[4]:
```
#finding out the dimensions of the dataset
gun1.shape
```

[4]:
```
(14465, 27)
```

There are 14465 rows and 27 columns in the dataset

[5]:
```
#getting more information about the dataset
gun1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14465 entries, 0 to 14464
Data columns (total 27 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   month                    14465 non-null  object
 1   state                    14465 non-null  object
 2   permit                   14441 non-null  float64
 3   permit_recheck           3080 non-null   float64
 4   handgun                  14445 non-null  float64
 5   long_gun                 14446 non-null  float64
 6   other                    7480 non-null   float64
 7   multiple                 14465 non-null  int64
 8   admin                    14442 non-null  float64
 9   prepawn_handgun          12522 non-null  float64
 10  prepawn_long_gun         12520 non-null  float64
 11  prepawn_other            7095 non-null   float64
 12  redemption_handgun       12525 non-null  float64
 13  redemption_long_gun      12524 non-null  float64
 14  redemption_other         7095 non-null   float64
 15  returned_handgun         4180 non-null   float64
 16  returned_long_gun        4125 non-null   float64
 17  returned_other           3795 non-null   float64
```

```
 18  rentals_handgun            2970 non-null   float64
 19  rentals_long_gun           2805 non-null   float64
 20  private_sale_handgun       4730 non-null   float64
 21  private_sale_long_gun      4730 non-null   float64
 22  private_sale_other         4730 non-null   float64
 23  return_to_seller_handgun   4455 non-null   float64
 24  return_to_seller_long_gun  4730 non-null   float64
 25  return_to_seller_other     4235 non-null   float64
 26  totals                     14465 non-null  int64
dtypes: float64(23), int64(2), object(2)
memory usage: 3.0+ MB
```

The **info()** function provides us with a lot of useful information about the dataset.

There are 14465 entries starting from 0 to 14464, and 27 columns. Also, the names of all the columns are shown, along with their data type and the number of the nonnull entries. And at the end, it states all the data types in the dataset and how many of them are there.

[6]:
```
# Calculating the number of missing values in each column
for x in range(1):
    print(gun1.isnull().sum())
```

```
month                          0
state                          0
permit                        24
permit_recheck             11385
handgun                       20
long_gun                      19
other                       6985
multiple                       0
admin                         23
prepawn_handgun             1943
prepawn_long_gun            1945
prepawn_other               7370
redemption_handgun          1940
redemption_long_gun         1941
redemption_other            7370
returned_handgun           10285
returned_long_gun          10340
returned_other             10670
rentals_handgun            11495
rentals_long_gun           11660
private_sale_handgun        9735
private_sale_long_gun       9735
private_sale_other          9735
return_to_seller_handgun   10010
return_to_seller_long_gun   9735
return_to_seller_other     10230
totals                         0
```

```
dtype: int64
```

I made a function to calculate the number of the missing values in each column, just in case I needed this information later.

[7]:
```
# taking a look at some statistical details about the dataset
gun1.describe()
```

[7]:

	permit	permit_recheck	handgun	long_gun	\
count	14441.000000	3080.000000	14445.000000	14446.000000	
mean	7160.660134	6852.150649	6814.501973	7873.343555	
std	26264.594279	45588.312382	10112.058750	9164.382320	
min	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	1011.000000	2150.000000	
50%	757.000000	0.000000	3432.000000	5203.000000	
75%	5298.000000	47.000000	8269.000000	10603.500000	
max	522188.000000	626794.000000	147714.000000	108058.000000	

	other	multiple	admin	prepawn_handgun	\
count	7480.000000	14465.000000	14442.000000	12522.000000	
mean	505.399465	292.524162	53.917601	5.144146	
std	1354.099433	778.159456	569.589979	11.365380	
min	0.000000	0.000000	0.000000	0.000000	
25%	27.000000	14.000000	0.000000	0.000000	
50%	168.000000	133.000000	0.000000	0.000000	
75%	513.000000	324.000000	0.000000	5.000000	
max	77929.000000	38907.000000	28083.000000	164.000000	

	prepawn_long_gun	prepawn_other	…	returned_other	rentals_handgun	\
count	12520.000000	7095.000000	…	3795.000000	2970.000000	
mean	7.589137	0.338689	…	2.293281	0.149158	
std	15.965987	1.314143	…	17.808110	1.013038	
min	0.000000	0.000000	…	0.000000	0.000000	
25%	0.000000	0.000000	…	0.000000	0.000000	
50%	1.000000	0.000000	…	0.000000	0.000000	
75%	8.000000	0.000000	…	1.000000	0.000000	
max	269.000000	49.000000	…	592.000000	13.000000	

	rentals_long_gun	private_sale_handgun	private_sale_long_gun	\
count	2805.000000	4730.000000	4730.000000	
mean	0.154724	24.418182	19.497252	
std	0.957151	90.060423	71.277555	
min	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	
50%	0.000000	0.000000	0.000000	
75%	0.000000	12.000000	12.000000	
max	12.000000	1299.000000	993.000000	

```
       private_sale_other  return_to_seller_handgun  \
count         4730.000000               4455.000000
mean             2.758351                  0.860606
std             11.575735                  4.340017
min              0.000000                  0.000000
25%              0.000000                  0.000000
50%              0.000000                  0.000000
75%              1.000000                  0.000000
max            232.000000                 70.000000

       return_to_seller_long_gun  return_to_seller_other         totals
count                4730.000000             4235.000000   14465.000000
mean                    0.851374                0.115939   24812.636364
std                     3.719907                0.427026   42351.474525
min                     0.000000                0.000000       0.000000
25%                     0.000000                0.000000    4963.000000
50%                     0.000000                0.000000   13184.000000
75%                     0.000000                0.000000   28295.000000
max                    62.000000                4.000000  714424.000000

[8 rows x 25 columns]
```

The **describe()** function provide us with some statistical details about each column in the dataset. Such as **count**, which counts the number of non-null values in each field. Also, there's **mean**, which calculates the mean of each column. **std**, which calculates the standard deviation of the values of each column. **min** and **max**, which are the minimum and miximum value respectively in each column. **25%**, **50%** and **75%** are the Quartile of each column, **25%** is the lower percentile, **50%** percentile is same as the median, and **75%** is the upper percentile.

[8]:
```
# Displaying the the columns' names; to have a better look
gun1.columns.values
```

[8]:
```
array(['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'], dtype=object)
```

As we can see, it created an array with the columns' names. Also, it provided me with the data type of these names; which is object.

1.3.2 2.2 Getting to know the Census dataset

I'm going to repeat the steps I did in **Getting to know the Gun dataset**. So, there will be no comments here, unless there's something that needs to be commented on.

[9]:
```
url='https://d17h27t6h515a5.cloudfront.net/topher/2017/November/5a0a554c_u.s.
↪-census-data/u.s.-census-data.csv'
cen=pd.read_csv(url)
cen.head(10)
```

[9]:

	Fact	Fact Note	Alabama	Alaska	Arizona	Arkansas	California	Colorado	Connecticut	Delaware	…	South Dakota	Tennessee	Texas	Utah	Vermont	Virginia	Washington	West Virginia	Wisconsin	Wyoming
0	Population estimates, July 1, 2016, (V2016)	NaN	4,863,300	741,894	6,931,071	2,988,248	39,250,017	5,540,545	3,576,452	952,065	…	865454	6651194	27,862,596	3,051,217	624,594	8,411,808	7,288,000	1,831,102	5,778,708	585,501
1	Population estimates base, April 1, 2010, (V2…	NaN	4,780,131	710,249	6,392,301	2,916,025	37,254,522	5,029,324	3,574,114	897,936	…	814195	6346298	25,146,100	2,763,888	625,741	8,001,041				
2	Population, percent change - April 1, 2010 (es…	NaN	1.70%	4.50%	8.40%	2.50%	5.40%	10.20%	0.10%	6.00%	…	0.063	0.048	10.80%	10.40%	-0.20%	5.10%				
3	Population, Census, April 1, 2010	NaN	4,779,736	710,231	6,392,017	2,915,918	37,253,956	5,029,196	3,574,097	897,934	…	814180	6346105	25,145,561	2,763,885	625,741	8,001,024				
4	Persons under 5 years, percent, July 1, 2016, …	NaN	6.00%	7.30%	6.30%	6.40%	6.30%	6.10%	5.20%	5.80%	…	0.071	0.061	7.20%	8.30%	4.90%	6.10%				
5	Persons under 5 years, percent, April 1, 2010	NaN	6.40%	7.60%	7.10%	6.80%	6.80%	6.80%	5.70%	6.20%	…	0.073	0.064	7.70%	9.50%	5.10%	6.40%				
6	Persons under 18 years, percent, July 1, 2016,…	NaN	22.60%	25.20%	23.50%	23.60%	23.20%	22.80%	21.10%	21.50%	…	0.246	0.226	26.20%	30.20%	19.00%	22.20%				
7	Persons under 18 years, percent, April 1, 2010	NaN	23.70%	26.40%	25.50%	24.40%	25.00%	24.40%	22.90%	22.90%	…	0.249	0.236	27.30%	31.50%	20.70%	23.20%				
8	Persons 65 years and over, percent, July 1, 2…	NaN	16.10%	10.40%	16.90%	16.30%	13.60%	13.40%	16.10%	17.50%	…	0.16	0.157	12.00%	10.50%	18.10%	14.60%				
9	Persons 65 years and over, percent, April 1, 2010	NaN	13.80%	7.70%	13.80%	14.40%	11.40%	10.90%	14.20%	14.40%	…	0.143	0.134	10.30%	9.00%	14.60%	12.20%				

```
1  6,724,545     1,853,011  5,687,289  563,767
2      8.40%        -1.20%      1.60%    3.90%
3  6,724,540     1,852,994  5,686,986  563,626
4      6.20%         5.50%      5.80%    6.50%
5      6.50%         5.60%      6.30%    7.10%
6     22.40%        20.50%     22.30%   23.70%
7     23.50%        20.90%     23.60%   24.00%
8     14.80%        18.80%     16.10%   15.00%
9     12.30%        16.00%     13.70%   12.40%

[10 rows x 52 columns]
```

[10]:
```
cen.tail(10)
```

[10]:
```
           Fact                                                Fact Note Alabama  \
75          NaN                                                      NaN     NaN
76  Value Flags                                                      NaN     NaN
77            -  Either no or too few sample observations were …     NaN
78            D  Suppressed to avoid disclosure of confidential…     NaN
79            F                                Fewer than 25 firms     NaN
80           FN            Footnote on this item in place of data     NaN
81          NaN                                     Not available     NaN
82            S   Suppressed; does not meet publication standards     NaN
83            X                                   Not applicable     NaN
84            Z  Value greater than zero but less than half uni…     NaN

   Alaska Arizona Arkansas California Colorado Connecticut Delaware  …  \
75    NaN     NaN      NaN        NaN      NaN         NaN      NaN  …
76    NaN     NaN      NaN        NaN      NaN         NaN      NaN  …
77    NaN     NaN      NaN        NaN      NaN         NaN      NaN  …
78    NaN     NaN      NaN        NaN      NaN         NaN      NaN  …
79    NaN     NaN      NaN        NaN      NaN         NaN      NaN  …
80    NaN     NaN      NaN        NaN      NaN         NaN      NaN  …
81    NaN     NaN      NaN        NaN      NaN         NaN      NaN  …
82    NaN     NaN      NaN        NaN      NaN         NaN      NaN  …
83    NaN     NaN      NaN        NaN      NaN         NaN      NaN  …
84    NaN     NaN      NaN        NaN      NaN         NaN      NaN  …

   South Dakota Tennessee Texas Utah Vermont Virginia Washington  \
75          NaN       NaN   NaN  NaN     NaN      NaN        NaN
76          NaN       NaN   NaN  NaN     NaN      NaN        NaN
77          NaN       NaN   NaN  NaN     NaN      NaN        NaN
78          NaN       NaN   NaN  NaN     NaN      NaN        NaN
79          NaN       NaN   NaN  NaN     NaN      NaN        NaN
80          NaN       NaN   NaN  NaN     NaN      NaN        NaN
81          NaN       NaN   NaN  NaN     NaN      NaN        NaN
82          NaN       NaN   NaN  NaN     NaN      NaN        NaN
```

```
83          NaN       NaN   NaN  NaN     NaN      NaN        NaN
84          NaN       NaN   NaN  NaN     NaN      NaN        NaN

   West Virginia Wisconsin Wyoming
75           NaN       NaN     NaN
76           NaN       NaN     NaN
77           NaN       NaN     NaN
78           NaN       NaN     NaN
79           NaN       NaN     NaN
80           NaN       NaN     NaN
81           NaN       NaN     NaN
82           NaN       NaN     NaN
83           NaN       NaN     NaN
84           NaN       NaN     NaN

[10 rows x 52 columns]
```

[11]:
```
cen.shape
```

[11]:
```
(85, 52)
```

[12]:
```
cen.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 85 entries, 0 to 84
Data columns (total 52 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Fact            80 non-null     object
 1   Fact Note       28 non-null     object
 2   Alabama         65 non-null     object
 3   Alaska          65 non-null     object
 4   Arizona         65 non-null     object
 5   Arkansas        65 non-null     object
 6   California      65 non-null     object
 7   Colorado        65 non-null     object
 8   Connecticut     65 non-null     object
 9   Delaware        65 non-null     object
 10  Florida         65 non-null     object
 11  Georgia         65 non-null     object
 12  Hawaii          65 non-null     object
 13  Idaho           65 non-null     object
 14  Illinois        65 non-null     object
 15  Indiana         65 non-null     object
 16  Iowa            65 non-null     object
 17  Kansas          65 non-null     object
 18  Kentucky        65 non-null     object
```

```
 19  Louisiana       65 non-null     object
 20  Maine           65 non-null     object
 21  Maryland        65 non-null     object
 22  Massachusetts   65 non-null     object
 23  Michigan        65 non-null     object
 24  Minnesota       65 non-null     object
 25  Mississippi     65 non-null     object
 26  Missouri        65 non-null     object
 27  Montana         65 non-null     object
 28  Nebraska        65 non-null     object
 29  Nevada          65 non-null     object
 30  New Hampshire   65 non-null     object
 31  New Jersey      65 non-null     object
 32  New Mexico      65 non-null     object
 33  New York        65 non-null     object
 34  North Carolina  65 non-null     object
 35  North Dakota    65 non-null     object
 36  Ohio            65 non-null     object
 37  Oklahoma        65 non-null     object
 38  Oregon          65 non-null     object
 39  Pennsylvania    65 non-null     object
 40  Rhode Island    65 non-null     object
 41  South Carolina  65 non-null     object
 42  South Dakota    65 non-null     object
 43  Tennessee       65 non-null     object
 44  Texas           65 non-null     object
 45  Utah            65 non-null     object
 46  Vermont         65 non-null     object
 47  Virginia        65 non-null     object
 48  Washington      65 non-null     object
 49  West Virginia   65 non-null     object
 50  Wisconsin       65 non-null     object
 51  Wyoming         65 non-null     object
dtypes: object(52)
memory usage: 34.7+ KB
```

[13]:
```
for x in range(1):
    print(cen.isnull().sum())
```

```
Fact              5
Fact Note        57
Alabama          20
Alaska           20
Arizona          20
Arkansas         20
California       20
Colorado         20
Connecticut      20
```

```
Delaware          20
Florida           20
Georgia           20
Hawaii            20
Idaho             20
Illinois          20
Indiana           20
Iowa              20
Kansas            20
Kentucky          20
Louisiana         20
Maine             20
Maryland          20
Massachusetts     20
Michigan          20
Minnesota         20
Mississippi       20
Missouri          20
Montana           20
Nebraska          20
Nevada            20
New Hampshire     20
New Jersey        20
New Mexico        20
New York          20
North Carolina    20
North Dakota      20
Ohio              20
Oklahoma          20
Oregon            20
Pennsylvania      20
Rhode Island      20
South Carolina    20
South Dakota      20
Tennessee         20
Texas             20
Utah              20
Vermont           20
Virginia          20
Washington        20
West Virginia     20
Wisconsin         20
Wyoming           20
dtype: int64
```

[14]:
```
cen.describe()
```

[14]:

	Fact	Fact Note	Alabama	Alaska	Arizona	Arkansas	\
count	80	28	65	65	65	65	
unique	80	15	65	64	64	64	
top	Households, 2011-2015	(c)	4,863,300	7.30%	50.30%	50.90%	
freq	1	6	1	2	2	2	

	California	Colorado	Connecticut	Delaware	…	South Dakota	Tennessee	\
count	65	65	65	65	…	65	65	
unique	63	64	63	64	…	65	64	
top	6.80%	3.30%	5.70%	51.60%	…	0.023	0.048	
freq	2	2	2	2	…	1	2	

	Texas	Utah	Vermont	Virginia	Washington	West Virginia	Wisconsin	\
count	65	65	65	65	65	65	65	
unique	64	64	63	65	65	64	65	
top	50.40%	2.50%	625,741	7.70%	8.90%	1.50%	4.80%	
freq	2	2	2	1	1	2	1	

	Wyoming
count	65
unique	64
top	7.10%
freq	2

[4 rows x 52 columns]

Since this dataset contains object data only, the result of the **describe()** function included **count** which is as mentioned before, counts the number of non-null values in a column. Also, **unique**, which is the number of unique values in each column. **top**, which is the most common value. And **freq**, which is the most common value's frequency.

[15]:
```
cen.columns.values
```

[15]:
```
array(['Fact', 'Fact Note', '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'], dtype=object)
```

1.3.3 2.3 Data cleaning

In this part, I'm going to move backwards and forwards between the two datasets while I'm cleaning. I'm going to document every step, and provide some comments in order to make keeping track of the cleaning process easier.

Note : Since some states in the Gun dataset were not included in the Census dataset, it would be better to remove them; so it could be easer to merge them later and answer the questions.

Cesus dataset

[16]:
```
#Making a copy of the original dataset, and droping the first 2 columns
cen2= cen.copy()
cen2=cen2.drop(['Fact','Fact Note'], axis=1)
```

I always prefer to create a copy of my original dataset and experiment on it. And when I come up with a satisfying result, I apply it on the original dataset, or simply take the output if it is what I am looking for. Here, I created a copy of the dataset to work on it, and extract the missing states (with the help of the Gun dataset). And the first step toward that is by deleting the **'Fact'** and **'Fact Note'** columns, so only the states columns will remain. I'm doing this because in the Gun dataset, there's a column with the name **state** that contain all the fifty states of America. But in the Census dataset, there's no such column. Yet, there's columns with the names of each state. So, if I want to extract the missing states, I will first have to create a new column in the census dataset and fill it with the names of the other states' columns.

[17]:
```
cen2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 85 entries, 0 to 84
Data columns (total 50 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Alabama         65 non-null     object
 1   Alaska          65 non-null     object
 2   Arizona         65 non-null     object
 3   Arkansas        65 non-null     object
 4   California      65 non-null     object
 5   Colorado        65 non-null     object
 6   Connecticut     65 non-null     object
 7   Delaware        65 non-null     object
 8   Florida         65 non-null     object
 9   Georgia         65 non-null     object
 10  Hawaii          65 non-null     object
 11  Idaho           65 non-null     object
 12  Illinois        65 non-null     object
 13  Indiana         65 non-null     object
 14  Iowa            65 non-null     object
 15  Kansas          65 non-null     object
 16  Kentucky        65 non-null     object
 17  Louisiana       65 non-null     object
```

```
 18  Maine           65 non-null     object
 19  Maryland        65 non-null     object
 20  Massachusetts   65 non-null     object
 21  Michigan        65 non-null     object
 22  Minnesota       65 non-null     object
 23  Mississippi     65 non-null     object
 24  Missouri        65 non-null     object
 25  Montana         65 non-null     object
 26  Nebraska        65 non-null     object
 27  Nevada          65 non-null     object
 28  New Hampshire   65 non-null     object
 29  New Jersey      65 non-null     object
 30  New Mexico      65 non-null     object
 31  New York        65 non-null     object
 32  North Carolina  65 non-null     object
 33  North Dakota    65 non-null     object
 34  Ohio            65 non-null     object
 35  Oklahoma        65 non-null     object
 36  Oregon          65 non-null     object
 37  Pennsylvania    65 non-null     object
 38  Rhode Island    65 non-null     object
 39  South Carolina  65 non-null     object
 40  South Dakota    65 non-null     object
 41  Tennessee       65 non-null     object
 42  Texas           65 non-null     object
 43  Utah            65 non-null     object
 44  Vermont         65 non-null     object
 45  Virginia        65 non-null     object
 46  Washington      65 non-null     object
 47  West Virginia   65 non-null     object
 48  Wisconsin       65 non-null     object
 49  Wyoming         65 non-null     object
dtypes: object(50)
memory usage: 33.3+ KB
```

I tend to use the **info()** function after each alteration I make, in order to check my data and see the changes. And, as we can see, the first two columns have been dropped successfully

[18]:
```
#creating a new column, and fill it with the names of the other columns.
cen2['state2'] = pd.Series(cen2.columns.values)
```

[19]:
```
cen2.head()
```

[19]:

	Alabama	Alaska	Arizona	Arkansas	California	Colorado	\
0	4,863,300	741,894	6,931,071	2,988,248	39,250,017	5,540,545	
1	4,780,131	710,249	6,392,301	2,916,025	37,254,522	5,029,324	
2	1.70%	4.50%	8.40%	2.50%	5.40%	10.20%	
3	4,779,736	710,231	6,392,017	2,915,918	37,253,956	5,029,196	

```
4      6.00%    7.30%      6.30%      6.40%       6.30%      6.10%

  Connecticut Delaware     Florida     Georgia  … Tennessee       Texas  \
0   3,576,452  952,065  20,612,439  10,310,371  …   6651194  27,862,596
1   3,574,114  897,936  18,804,592   9,688,680  …   6346298  25,146,100
2       0.10%    6.00%       9.60%       6.40%  …     0.048      10.80%
3   3,574,097  897,934  18,801,310   9,687,653  …   6346105  25,145,561
4       5.20%    5.80%       5.50%       6.40%  …     0.061       7.20%

         Utah  Vermont   Virginia Washington West Virginia  Wisconsin  Wyoming  \
0  3,051,217  624,594  8,411,808  7,288,000     1,831,102  5,778,708  585,501
1  2,763,888  625,741  8,001,041  6,724,545     1,853,011  5,687,289  563,767
2     10.40%   -0.20%      5.10%      8.40%        -1.20%      1.60%    3.90%
3  2,763,885  625,741  8,001,024  6,724,540     1,852,994  5,686,986  563,626
4      8.30%    4.90%      6.10%      6.20%         5.50%      5.80%    6.50%

       state2
0     Alabama
1      Alaska
2     Arizona
3    Arkansas
4  California

[5 rows x 51 columns]
```

[20]:
```
cen2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 85 entries, 0 to 84
Data columns (total 51 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Alabama         65 non-null     object
 1   Alaska          65 non-null     object
 2   Arizona         65 non-null     object
 3   Arkansas        65 non-null     object
 4   California      65 non-null     object
 5   Colorado        65 non-null     object
 6   Connecticut     65 non-null     object
 7   Delaware        65 non-null     object
 8   Florida         65 non-null     object
 9   Georgia         65 non-null     object
 10  Hawaii          65 non-null     object
 11  Idaho           65 non-null     object
 12  Illinois        65 non-null     object
 13  Indiana         65 non-null     object
 14  Iowa            65 non-null     object
```

```
 15  Kansas          65 non-null     object
 16  Kentucky        65 non-null     object
 17  Louisiana       65 non-null     object
 18  Maine           65 non-null     object
 19  Maryland        65 non-null     object
 20  Massachusetts   65 non-null     object
 21  Michigan        65 non-null     object
 22  Minnesota       65 non-null     object
 23  Mississippi     65 non-null     object
 24  Missouri        65 non-null     object
 25  Montana         65 non-null     object
 26  Nebraska        65 non-null     object
 27  Nevada          65 non-null     object
 28  New Hampshire   65 non-null     object
 29  New Jersey      65 non-null     object
 30  New Mexico      65 non-null     object
 31  New York        65 non-null     object
 32  North Carolina  65 non-null     object
 33  North Dakota    65 non-null     object
 34  Ohio            65 non-null     object
 35  Oklahoma        65 non-null     object
 36  Oregon          65 non-null     object
 37  Pennsylvania    65 non-null     object
 38  Rhode Island    65 non-null     object
 39  South Carolina  65 non-null     object
 40  South Dakota    65 non-null     object
 41  Tennessee       65 non-null     object
 42  Texas           65 non-null     object
 43  Utah            65 non-null     object
 44  Vermont         65 non-null     object
 45  Virginia        65 non-null     object
 46  Washington      65 non-null     object
 47  West Virginia   65 non-null     object
 48  Wisconsin       65 non-null     object
 49  Wyoming         65 non-null     object
 50  state2          50 non-null     object
dtypes: object(51)
memory usage: 34.0+ KB
```

I created a new field and named it **state2**, and it contains only the names of the other columns, which are the names of the states in Census dataset.

[21]:
```
#dropping all the columns except state2 column.
cen2.drop(cen2.columns.difference(['state2']), 1, inplace=True)
```

[22]:
```
cen2.columns.values
```

[22]:
```
array(['state2'], dtype=object)
```

[23]:
```
cen2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 85 entries, 0 to 84
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   state2  50 non-null     object
dtypes: object(1)
memory usage: 808.0+ bytes
```

I dropped all the columns except **state2** column. Because in this part, I'm only going to need this column from this dataset for comparison and extraction.

[24]:
```
#dropping all the null values in case there's any.
cen2 = cen2.dropna()
cen2
```

[24]:

	state2
0	Alabama
1	Alaska
2	Arizona
3	Arkansas
4	California
5	Colorado
6	Connecticut
7	Delaware
8	Florida
9	Georgia
10	Hawaii
11	Idaho
12	Illinois
13	Indiana
14	Iowa
15	Kansas
16	Kentucky
17	Louisiana
18	Maine
19	Maryland
20	Massachusetts
21	Michigan
22	Minnesota
23	Mississippi
24	Missouri
25	Montana
26	Nebraska
27	Nevada
28	New Hampshire

```
29      New Jersey
30      New Mexico
31        New York
32  North Carolina
33    North Dakota
34            Ohio
35        Oklahoma
36          Oregon
37    Pennsylvania
38    Rhode Island
39  South Carolina
40    South Dakota
41       Tennessee
42           Texas
43            Utah
44         Vermont
45        Virginia
46      Washington
47   West Virginia
48       Wisconsin
49         Wyoming
```

Gun dataset I'm going to repeat the steps I did above, so there will be no comments here, unless there's something that needs to be commented on.

[25]:
```
Gun= gun1.copy()
```

[26]:
```
Gun['state1'] = pd.Series(gun1['state'].unique())
```

[27]:
```
Gun.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14465 entries, 0 to 14464
Data columns (total 28 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   month                      14465 non-null  object
 1   state                      14465 non-null  object
 2   permit                     14441 non-null  float64
 3   permit_recheck             3080 non-null   float64
 4   handgun                    14445 non-null  float64
 5   long_gun                   14446 non-null  float64
 6   other                      7480 non-null   float64
 7   multiple                   14465 non-null  int64
 8   admin                      14442 non-null  float64
 9   prepawn_handgun            12522 non-null  float64
 10  prepawn_long_gun           12520 non-null  float64
```

```
 11  prepawn_other              7095 non-null   float64
 12  redemption_handgun         12525 non-null  float64
 13  redemption_long_gun        12524 non-null  float64
 14  redemption_other           7095 non-null   float64
 15  returned_handgun           4180 non-null   float64
 16  returned_long_gun          4125 non-null   float64
 17  returned_other             3795 non-null   float64
 18  rentals_handgun            2970 non-null   float64
 19  rentals_long_gun           2805 non-null   float64
 20  private_sale_handgun       4730 non-null   float64
 21  private_sale_long_gun      4730 non-null   float64
 22  private_sale_other         4730 non-null   float64
 23  return_to_seller_handgun   4455 non-null   float64
 24  return_to_seller_long_gun  4730 non-null   float64
 25  return_to_seller_other     4235 non-null   float64
 26  totals                     14465 non-null  int64
 27  state1                     55 non-null     object
dtypes: float64(23), int64(2), object(3)
memory usage: 3.1+ MB
```

Despit that the gun1 dataset already has a **state** column that contais all the states, I'm creating a new column with the name **state1** that contains only the unique ones; since **state** column contains the states repeated 14465 times based on the date. So, in **state1** each state will be mentioned once only.

[28]:
```
Gun.drop(Gun.columns.difference(['state1']), 1, inplace=True)
```

[29]:
```
Gun.columns.values
```

[29]:
```
array(['state1'], dtype=object)
```

[30]:
```
Gun.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14465 entries, 0 to 14464
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   state1  55 non-null     object
dtypes: object(1)
memory usage: 113.1+ KB
```

[31]:
```
Gun = Gun.dropna()
```

[32]:
```
Gun.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 55 entries, 0 to 54
Data columns (total 1 columns):
```

```
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   state1  55 non-null     object
dtypes: object(1)
memory usage: 880.0+ bytes
```

Merging Gun and cen2 In this part I'm going to figure out what are the missing states.

[33]:
```
Gun
```

[33]:

	state1
0	Alabama
1	Alaska
2	Arizona
3	Arkansas
4	California
5	Colorado
6	Connecticut
7	Delaware
8	District of Columbia
9	Florida
10	Georgia
11	Guam
12	Hawaii
13	Idaho
14	Illinois
15	Indiana
16	Iowa
17	Kansas
18	Kentucky
19	Louisiana
20	Maine
21	Mariana Islands
22	Maryland
23	Massachusetts
24	Michigan
25	Minnesota
26	Mississippi
27	Missouri
28	Montana
29	Nebraska
30	Nevada
31	New Hampshire
32	New Jersey
33	New Mexico
34	New York
35	North Carolina

36	North Dakota
37	Ohio
38	Oklahoma
39	Oregon
40	Pennsylvania
41	Puerto Rico
42	Rhode Island
43	South Carolina
44	South Dakota
45	Tennessee
46	Texas
47	Utah
48	Vermont
49	Virgin Islands
50	Virginia
51	Washington
52	West Virginia
53	Wisconsin
54	Wyoming

[34]:
```
cen2
```

[34]:

	state2
0	Alabama
1	Alaska
2	Arizona
3	Arkansas
4	California
5	Colorado
6	Connecticut
7	Delaware
8	Florida
9	Georgia
10	Hawaii
11	Idaho
12	Illinois
13	Indiana
14	Iowa
15	Kansas
16	Kentucky
17	Louisiana
18	Maine
19	Maryland
20	Massachusetts
21	Michigan
22	Minnesota
23	Mississippi

```
24        Missouri
25         Montana
26        Nebraska
27          Nevada
28   New Hampshire
29      New Jersey
30      New Mexico
31        New York
32  North Carolina
33    North Dakota
34            Ohio
35        Oklahoma
36          Oregon
37    Pennsylvania
38    Rhode Island
39  South Carolina
40    South Dakota
41       Tennessee
42           Texas
43            Utah
44         Vermont
45        Virginia
46      Washington
47   West Virginia
48       Wisconsin
49         Wyoming
```

[35]:
```
# merging the two datasets
merging = Gun.merge(cen2, how='left', left_on=['state1'], right_on=['state2'])
merging
```

[35]:

	state1	state2
0	Alabama	Alabama
1	Alaska	Alaska
2	Arizona	Arizona
3	Arkansas	Arkansas
4	California	California
5	Colorado	Colorado
6	Connecticut	Connecticut
7	Delaware	Delaware
8	District of Columbia	NaN
9	Florida	Florida
10	Georgia	Georgia
11	Guam	NaN
12	Hawaii	Hawaii
13	Idaho	Idaho
14	Illinois	Illinois

15	Indiana	Indiana
16	Iowa	Iowa
17	Kansas	Kansas
18	Kentucky	Kentucky
19	Louisiana	Louisiana
20	Maine	Maine
21	Mariana Islands	NaN
22	Maryland	Maryland
23	Massachusetts	Massachusetts
24	Michigan	Michigan
25	Minnesota	Minnesota
26	Mississippi	Mississippi
27	Missouri	Missouri
28	Montana	Montana
29	Nebraska	Nebraska
30	Nevada	Nevada
31	New Hampshire	New Hampshire
32	New Jersey	New Jersey
33	New Mexico	New Mexico
34	New York	New York
35	North Carolina	North Carolina
36	North Dakota	North Dakota
37	Ohio	Ohio
38	Oklahoma	Oklahoma
39	Oregon	Oregon
40	Pennsylvania	Pennsylvania
41	Puerto Rico	NaN
42	Rhode Island	Rhode Island
43	South Carolina	South Carolina
44	South Dakota	South Dakota
45	Tennessee	Tennessee
46	Texas	Texas
47	Utah	Utah
48	Vermont	Vermont
49	Virgin Islands	NaN
50	Virginia	Virginia
51	Washington	Washington
52	West Virginia	West Virginia
53	Wisconsin	Wisconsin
54	Wyoming	Wyoming

The **merge()** function merged the two datasets using only keys from left frame (Gun), it's just like SQL outter join on preserving key order. It merged left on **state1** and right on **state2**. the output of this is a dataframe with both **state1** and **state2** next to each other, and if there's a value in **state1** that's not in **state2** it get replaced with **NaN**. Which is exactly what I'm looking for :)

[36]:
```
#extracting the NaN values only
Gun[merging['state2'].isna()]
```

[36]:

	state1
8	District of Columbia
11	Guam
21	Mariana Islands
41	Puerto Rico
49	Virgin Islands

And you're looking at the states that were not included in the Census dataset :)

[37]:
```
#dropping these states from gun1
gun1 = gun1[gun1.state != 'Guam']
gun1 = gun1[gun1.state != 'Puerto Rico']
gun1 = gun1[gun1.state != 'District of Columbia']
gun1 = gun1[gun1.state != 'Virgin Islands']
gun1 = gun1[gun1.state != 'Mariana Islands']
```

[38]:
```
gun1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 13150 entries, 0 to 14464
Data columns (total 27 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   month                     13150 non-null  object
 1   state                     13150 non-null  object
 2   permit                    13148 non-null  float64
 3   permit_recheck            2800 non-null   float64
 4   handgun                   13150 non-null  float64
 5   long_gun                  13150 non-null  float64
 6   other                     6800 non-null   float64
 7   multiple                  13150 non-null  int64
 8   admin                     13148 non-null  float64
 9   prepawn_handgun           11397 non-null  float64
 10  prepawn_long_gun          11395 non-null  float64
 11  prepawn_other             6450 non-null   float64
 12  redemption_handgun        11400 non-null  float64
 13  redemption_long_gun       11398 non-null  float64
 14  redemption_other          6450 non-null   float64
 15  returned_handgun          3800 non-null   float64
 16  returned_long_gun         3750 non-null   float64
 17  returned_other            3450 non-null   float64
 18  rentals_handgun           2700 non-null   float64
 19  rentals_long_gun          2550 non-null   float64
 20  private_sale_handgun      4300 non-null   float64
 21  private_sale_long_gun     4300 non-null   float64
```

```
 22  private_sale_other          4300 non-null   float64
 23  return_to_seller_handgun    4050 non-null   float64
 24  return_to_seller_long_gun   4300 non-null   float64
 25  return_to_seller_other      3850 non-null   float64
 26  totals                      13150 non-null  int64
dtypes: float64(23), int64(2), object(2)
memory usage: 2.8+ MB
```

I eliminated the extracted five states from the **Gun** dataset along with any information related to them.

[39]:
```
#I'm going to create a table with the data needed to match the Census dataset␣
 ↪and prepare it for merging later.
GunCapita= gun1[['month','state','totals']]
GunCapita.head()
```

[39]:

	month	state	totals
0	2020-09	Alabama	80478
1	2020-09	Alaska	7897
2	2020-09	Arizona	51287
3	2020-09	Arkansas	24043
4	2020-09	California	139313

[40]:
```
GunCapita.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 13150 entries, 0 to 14464
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   month   13150 non-null  object
 1   state   13150 non-null  object
 2   totals  13150 non-null  int64
dtypes: int64(1), object(2)
memory usage: 410.9+ KB
```

I created a new dataframe with only the **month**, **state** and **totals** columns, and it would help me answer one of my questions later.

[41]:
```
#convert the month column from object to datetime
GunCapita['month'] = pd.to_datetime(gun1['month'])
```

```
<ipython-input-41-faa50fca02f0>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  GunCapita['month'] = pd.to_datetime(gun1['month'])
```

[42]:
```
GunCapita.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 13150 entries, 0 to 14464
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   month   13150 non-null  datetime64[ns]
 1   state   13150 non-null  object
 2   totals  13150 non-null  int64
dtypes: datetime64[ns](1), int64(1), object(1)
memory usage: 410.9+ KB
```

[43]:
```
GunCapita.head()
```

[43]:

	month	state	totals
0	2020-09-01	Alabama	80478
1	2020-09-01	Alaska	7897
2	2020-09-01	Arizona	51287
3	2020-09-01	Arkansas	24043
4	2020-09-01	California	139313

Since the **month** column was an **object**, I converted it into **date** so I can extract the needed data easly.

[44]:
```
#I'm going to extract the data of the 1st of July,2016 then do the same with␣
 ↪the 1st of April,2010
GUN2016= GunCapita[GunCapita['month']== '2016-07-01']
GUN2016.head()
```

[44]:

	month	state	totals
2750	2016-07-01	Alabama	48927
2751	2016-07-01	Alaska	6793
2752	2016-07-01	Arizona	34496
2753	2016-07-01	Arkansas	19378
2754	2016-07-01	California	190218

I extracted the 1st of July, 2016 data from the Gun dataset to match the one in the Census dataset.

[45]:
```
GUN2010= GunCapita[GunCapita['month']== '2010-04-01']
GUN2010.head()
```

[45]:

	month	state	totals
6875	2010-04-01	Alabama	20791
6876	2010-04-01	Alaska	6411
6877	2010-04-01	Arizona	16578
6878	2010-04-01	Arkansas	14563
6879	2010-04-01	California	80750

I extracted the 1st of April, 2010 data from the Gun dataset to match the one in the Census dataset.

[46]:
```
#check for null values in the columns that matters to me
gun1.isnull().any()
```

[46]:
```
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
```

Since there's no null values in **month**, **state**, **handgun**, **long_gun**, **multiple** and **totals** columns, that leaves me only with **other** column to take care of. But it won't be necessary, since these missing values will not affect my answers.

[47]:
```
#creating a new copy of the Census dataset
CEN=cen.copy()
```

[48]:
```
CEN.head()
```

[48]:

	Fact	Fact Note	Alabama
0	Population estimates, July 1, 2016, (V2016)	NaN	4,863,300
1	Population estimates base, April 1, 2010, (V2…	NaN	4,780,131
2	Population, percent change - April 1, 2010 (es…	NaN	1.70%
3	Population, Census, April 1, 2010	NaN	4,779,736

```
4  Persons under 5 years, percent, July 1, 2016, …       NaN      6.00%

    Alaska    Arizona   Arkansas  California   Colorado Connecticut Delaware  \
0  741,894  6,931,071  2,988,248  39,250,017  5,540,545   3,576,452  952,065
1  710,249  6,392,301  2,916,025  37,254,522  5,029,324   3,574,114  897,936
2    4.50%      8.40%      2.50%       5.40%     10.20%       0.10%    6.00%
3  710,231  6,392,017  2,915,918  37,253,956  5,029,196   3,574,097  897,934
4    7.30%      6.30%      6.40%       6.30%      6.10%       5.20%    5.80%

   … South Dakota Tennessee       Texas       Utah  Vermont   Virginia  \
0  …       865454   6651194  27,862,596  3,051,217  624,594  8,411,808
1  …       814195   6346298  25,146,100  2,763,888  625,741  8,001,041
2  …        0.063     0.048      10.80%     10.40%   -0.20%      5.10%
3  …       814180   6346105  25,145,561  2,763,885  625,741  8,001,024
4  …        0.071     0.061       7.20%      8.30%    4.90%      6.10%

  Washington West Virginia  Wisconsin  Wyoming
0  7,288,000     1,831,102  5,778,708  585,501
1  6,724,545     1,853,011  5,687,289  563,767
2      8.40%        -1.20%      1.60%    3.90%
3  6,724,540     1,852,994  5,686,986  563,626
4      6.20%         5.50%      5.80%    6.50%

[5 rows x 52 columns]
```

[49]:
```
#dropping 'Fact Note' column since it won't be needed
CEN=CEN.drop('Fact Note',axis=1)
```

[50]:
```
#setting 'Fact' column as an index
CEN.set_index('Fact',inplace=True)
```

[51]:
```
#applying the transpose method and resetting the index
CEN = CEN.T.reset_index()
```

[52]:
```
CEN.head()
```

[52]:
```
Fact       index Population estimates, July 1, 2016,  (V2016)  \
0        Alabama                                      4,863,300
1         Alaska                                        741,894
2        Arizona                                      6,931,071
3       Arkansas                                      2,988,248
4     California                                     39,250,017

Fact Population estimates base, April 1, 2010,  (V2016)  \
0                                          4,780,131
1                                            710,249
2                                          6,392,301
```

```
3                                                2,916,025
4                                               37,254,522

Fact Population, percent change - April 1, 2010 (estimates base) to July 1,
2016,  (V2016)  \
0                                                    1.70%
1                                                    4.50%
2                                                    8.40%
3                                                    2.50%
4                                                    5.40%

Fact Population, Census, April 1, 2010  \
0                             4,779,736
1                               710,231
2                             6,392,017
3                             2,915,918
4                            37,253,956

Fact Persons under 5 years, percent, July 1, 2016,  (V2016)  \
0                                                    6.00%
1                                                    7.30%
2                                                    6.30%
3                                                    6.40%
4                                                    6.30%

Fact Persons under 5 years, percent, April 1, 2010  \
0                                               6.40%
1                                               7.60%
2                                               7.10%
3                                               6.80%
4                                               6.80%

Fact Persons under 18 years, percent, July 1, 2016,  (V2016)  \
0                                                   22.60%
1                                                   25.20%
2                                                   23.50%
3                                                   23.60%
4                                                   23.20%

Fact Persons under 18 years, percent, April 1, 2010  \
0                                               23.70%
1                                               26.40%
2                                               25.50%
3                                               24.40%
4                                               25.00%

Fact Persons 65 years and over, percent,  July 1, 2016,  (V2016)  …  NaN  \
```

```
0                                                 16.10%           …  NaN
1                                                 10.40%           …  NaN
2                                                 16.90%           …  NaN
3                                                 16.30%           …  NaN
4                                                 13.60%           …  NaN

Fact Value Flags    -    D    F   FN  NaN    S    X    Z
0            NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN
1            NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN
2            NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN
3            NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN
4            NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN

[5 rows x 86 columns]
```

What I did here is simply making the rows column and the columns rows, so the dataset would be more presentable to me.

[53]:
```
#renameing the 'index' column to 'state' so it would make more sense
CEN=CEN.rename(columns={'index':'State'})
```

[54]:
```
CEN.head()
```

[54]:
```
Fact       State Population estimates, July 1, 2016,  (V2016)  \
0        Alabama                                     4,863,300
1         Alaska                                       741,894
2        Arizona                                     6,931,071
3       Arkansas                                     2,988,248
4     California                                    39,250,017

Fact Population estimates base, April 1, 2010,  (V2016)  \
0                                             4,780,131
1                                               710,249
2                                             6,392,301
3                                             2,916,025
4                                            37,254,522

Fact Population, percent change - April 1, 2010 (estimates base) to July 1,
2016,  (V2016)  \
0                                                 1.70%
1                                                 4.50%
2                                                 8.40%
3                                                 2.50%
4                                                 5.40%

Fact Population, Census, April 1, 2010  \
0                            4,779,736
```

```
1                              710,231
2                            6,392,017
3                            2,915,918
4                           37,253,956

Fact Persons under 5 years, percent, July 1, 2016,  (V2016)  \
0                                                      6.00%
1                                                      7.30%
2                                                      6.30%
3                                                      6.40%
4                                                      6.30%

Fact Persons under 5 years, percent, April 1, 2010  \
0                                                6.40%
1                                                7.60%
2                                                7.10%
3                                                6.80%
4                                                6.80%

Fact Persons under 18 years, percent, July 1, 2016,  (V2016)  \
0                                                     22.60%
1                                                     25.20%
2                                                     23.50%
3                                                     23.60%
4                                                     23.20%

Fact Persons under 18 years, percent, April 1, 2010  \
0                                                23.70%
1                                                26.40%
2                                                25.50%
3                                                24.40%
4                                                25.00%

Fact Persons 65 years and over, percent,  July 1, 2016,  (V2016)  …  NaN  \
0                                                     16.10%          …  NaN
1                                                     10.40%          …  NaN
2                                                     16.90%          …  NaN
3                                                     16.30%          …  NaN
4                                                     13.60%          …  NaN

Fact Value Flags    -    D    F   FN  NaN    S    X    Z
0            NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN
1            NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN
2            NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN
3            NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN
4            NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN
```

```
[5 rows x 86 columns]
```

[55]:
```
CEN.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 86 columns):
 #   Column
Non-Null Count  Dtype
---  ------
--------------  -----
 0   State
50 non-null     object
 1   Population estimates, July 1, 2016,  (V2016)
50 non-null     object
 2   Population estimates base, April 1, 2010,  (V2016)
50 non-null     object
 3   Population, percent change - April 1, 2010 (estimates base) to July 1,
2016,  (V2016)   50 non-null     object
 4   Population, Census, April 1, 2010
50 non-null     object
 5   Persons under 5 years, percent, July 1, 2016,  (V2016)
50 non-null     object
 6   Persons under 5 years, percent, April 1, 2010
50 non-null     object
 7   Persons under 18 years, percent, July 1, 2016,  (V2016)
50 non-null     object
 8   Persons under 18 years, percent, April 1, 2010
50 non-null     object
 9   Persons 65 years and over, percent,  July 1, 2016,  (V2016)
50 non-null     object
 10  Persons 65 years and over, percent, April 1, 2010
50 non-null     object
 11  Female persons, percent,  July 1, 2016,  (V2016)
50 non-null     object
 12  Female persons, percent, April 1, 2010
50 non-null     object
 13  White alone, percent, July 1, 2016,  (V2016)
50 non-null     object
 14  Black or African American alone, percent, July 1, 2016,  (V2016)
50 non-null     object
 15  American Indian and Alaska Native alone, percent, July 1, 2016,  (V2016)
50 non-null     object
 16  Asian alone, percent, July 1, 2016,  (V2016)
50 non-null     object
 17  Native Hawaiian and Other Pacific Islander alone, percent, July 1, 2016,
(V2016)       50 non-null     object
```

```
 18  Two or More Races, percent, July 1, 2016,  (V2016)
50 non-null     object
 19  Hispanic or Latino, percent, July 1, 2016,  (V2016)
50 non-null     object
 20  White alone, not Hispanic or Latino, percent, July 1, 2016,  (V2016)
50 non-null     object
 21  Veterans, 2011-2015
50 non-null     object
 22  Foreign born persons, percent, 2011-2015
50 non-null     object
 23  Housing units,  July 1, 2016,  (V2016)
50 non-null     object
 24  Housing units, April 1, 2010
50 non-null     object
 25  Owner-occupied housing unit rate, 2011-2015
50 non-null     object
 26  Median value of owner-occupied housing units, 2011-2015
50 non-null     object
 27  Median selected monthly owner costs -with a mortgage, 2011-2015
50 non-null     object
 28  Median selected monthly owner costs -without a mortgage, 2011-2015
50 non-null     object
 29  Median gross rent, 2011-2015
50 non-null     object
 30  Building permits, 2016
50 non-null     object
 31  Households, 2011-2015
50 non-null     object
 32  Persons per household, 2011-2015
50 non-null     object
 33  Living in same house 1 year ago, percent of persons age 1 year+, 2011-2015
50 non-null     object
 34  Language other than English spoken at home, percent of persons age 5
years+, 2011-2015  50 non-null     object
 35  High school graduate or higher, percent of persons age 25 years+, 2011-2015
50 non-null     object
 36  Bachelor's degree or higher, percent of persons age 25 years+, 2011-2015
50 non-null     object
 37  With a disability, under age 65 years, percent, 2011-2015
50 non-null     object
 38  Persons  without health insurance, under age 65 years, percent
50 non-null     object
 39  In civilian labor force, total, percent of population age 16 years+,
2011-2015          50 non-null     object
 40  In civilian labor force, female, percent of population age 16 years+,
2011-2015         50 non-null     object
 41  Total accommodation and food services sales, 2012 ($1,000)
50 non-null     object
```

```
 42  Total health care and social assistance receipts/revenue, 2012 ($1,000)
50 non-null     object
 43  Total manufacturers shipments, 2012 ($1,000)
50 non-null     object
 44  Total merchant wholesaler sales, 2012 ($1,000)
50 non-null     object
 45  Total retail sales, 2012 ($1,000)
50 non-null     object
 46  Total retail sales per capita, 2012
50 non-null     object
 47  Mean travel time to work (minutes), workers age 16 years+, 2011-2015
50 non-null     object
 48  Median household income (in 2015 dollars), 2011-2015
50 non-null     object
 49  Per capita income in past 12 months (in 2015 dollars), 2011-2015
50 non-null     object
 50  Persons in poverty, percent
50 non-null     object
 51  Total employer establishments, 2015
50 non-null     object
 52  Total employment, 2015
50 non-null     object
 53  Total annual payroll, 2015 ($1,000)
50 non-null     object
 54  Total employment, percent change, 2014-2015
50 non-null     object
 55  Total nonemployer establishments, 2015
50 non-null     object
 56  All firms, 2012
50 non-null     object
 57  Men-owned firms, 2012
50 non-null     object
 58  Women-owned firms, 2012
50 non-null     object
 59  Minority-owned firms, 2012
50 non-null     object
 60  Nonminority-owned firms, 2012
50 non-null     object
 61  Veteran-owned firms, 2012
50 non-null     object
 62  Nonveteran-owned firms, 2012
50 non-null     object
 63  Population per square mile, 2010
50 non-null     object
 64  Land area in square miles, 2010
50 non-null     object
 65  FIPS Code
50 non-null     object
```

```
 66  nan
0 non-null      object
 67  NOTE: FIPS Code values are enclosed in quotes to ensure leading zeros
remain intact.    0 non-null      object
 68  nan
0 non-null      object
 69  Value Notes
0 non-null      object
 70  1
0 non-null      object
 71  nan
0 non-null      object
 72  Fact Notes
0 non-null      object
 73  (a)
0 non-null      object
 74  (b)
0 non-null      object
 75  (c)
0 non-null      object
 76  nan
0 non-null      object
 77  Value Flags
0 non-null      object
 78  -
0 non-null      object
 79  D
0 non-null      object
 80  F
0 non-null      object
 81  FN
0 non-null      object
 82  nan
0 non-null      object
 83  S
0 non-null      object
 84  X
0 non-null      object
 85  Z
0 non-null      object
dtypes: object(86)
memory usage: 33.7+ KB
```

[56]:
```
#dropping the unneeded columns
CEN.drop(CEN.iloc[:, 66:86], inplace = True, axis = 1)
```

[57]:
```
CEN.head()
```

[57]:
Fact	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)
0	Alabama	4,863,300	4,780,131	1.70%	4,779,736	6.00%	6.40%	22.60%
1	Alaska	741,894	710,249	4.50%	710,231	7.30%	7.60%	25.20%
2	Arizona	6,931,071	6,392,301	8.40%	6,392,017	6.30%	7.10%	23.50%
3	Arkansas	2,988,248	2,916,025	2.50%	2,915,918	6.40%	6.80%	
4	California	39,250,017	37,254,522	5.40%	37,253,956	6.30%	6.80%	

```
3                                                 23.60%
4                                                 23.20%

Fact Persons under 18 years, percent, April 1, 2010  \
0                                              23.70%
1                                              26.40%
2                                              25.50%
3                                              24.40%
4                                              25.00%

Fact Persons 65 years and over, percent,  July 1, 2016,  (V2016)  …  \
0                                                 16.10%           …
1                                                 10.40%           …
2                                                 16.90%           …
3                                                 16.30%           …
4                                                 13.60%           …

Fact All firms, 2012 Men-owned firms, 2012 Women-owned firms, 2012  \
0            374,153               203,604                 137,630
1             68,032                35,402                  22,141
2            499,926               245,243                 182,425
3            231,959               123,158                  75,962
4          3,548,449             1,852,580               1,320,085

Fact Minority-owned firms, 2012 Nonminority-owned firms, 2012  \
0                         92,219                         272,651
1                         13,688                          51,147
2                        135,313                         344,981
3                         35,982                         189,029
4                      1,619,857                       1,819,107

Fact Veteran-owned firms, 2012 Nonveteran-owned firms, 2012  \
0                        41,943                       316,984
1                         7,953                        56,091
2                        46,780                       427,582
3                        25,915                       192,988
4                       252,377                     3,176,341

Fact Population per square mile, 2010 Land area in square miles, 2010  \
0                                  94.4                       50,645.33
1                                   1.2                      570,640.95
2                                  56.3                      113,594.08
3                                    56                       52,035.48
4                                 239.1                      155,779.22

Fact FIPS Code
0         "01"
```

```
1         "02"
2         "04"
3         "05"
4         "06"

[5 rows x 66 columns]
```

I removed the unneded columns starting from 66 till 85

[58]:
```
#replacing NaN values with 0.0
CEN=CEN.fillna(0.0)
```

[59]:
```
# checking if there's any missing value, and if yes I want to see it
CE = CEN[CEN.isna().any(axis=1)]
CE
```

[59]:
```
Empty DataFrame
Columns: [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), 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,  (V2016), 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+, 2011-2015, In civilian labor force, female, percent of population age 16
years+, 2011-2015, 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]
Index: []

[0 rows x 66 columns]
```

Since the output has 0 rows, that means there's no missing value.

Problem has been taken care of :)

What I'm doing next is converting the needed columns from object to float.

[60]:
```
#creating a new dataframe with only these columns
CEN16= CEN[['State', 'Population estimates, July 1, 2016,  (V2016)']]
CEN16.head()
```

[60]:

Fact	State	Population estimates, July 1, 2016, (V2016)
0	Alabama	4,863,300
1	Alaska	741,894
2	Arizona	6,931,071
3	Arkansas	2,988,248
4	California	39,250,017

[61]:
```
#first, instead of (,) I'll put an empty('') as to eliminate the (,)
CEN16['Population estimates, July 1, 2016,  (V2016)'].
 ↪replace(to_replace=',',value=r'',regex=True,inplace=True)
```

```
/Users/yara/opt/anaconda3/envs/second_project/lib/python3.8/site-
packages/pandas/core/series.py:4563: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return super().replace(
```

[62]:
```
#checking the changes
CEN16.head()
```

[62]:
Fact	State	Population estimates, July 1, 2016, (V2016)
0	Alabama	4863300
1	Alaska	741894
2	Arizona	6931071
3	Arkansas	2988248
4	California	39250017

[63]:
```
#Secons, I'm going to convert the Population estimates, July 1, 2016, (V2016)␣
↪column from object to float
CEN16['Population estimates, July 1, 2016,  (V2016)']= pd.
↪to_numeric(CEN16['Population estimates, July 1, 2016,  (V2016)'],␣
↪downcast='float', errors='ignore')
```

```
<ipython-input-63-4082256a6a1b>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  CEN16['Population estimates, July 1, 2016,  (V2016)']=
pd.to_numeric(CEN16['Population estimates, July 1, 2016,  (V2016)'],
downcast='float', errors='ignore')
```

[64]:
```
#checking the changes
CEN16.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 2 columns):
 #   Column                                      Non-Null Count  Dtype
---  ------                                      --------------  -----
 0   State                                       50 non-null     object
 1   Population estimates, July 1, 2016,  (V2016)  50 non-null     float32
dtypes: float32(1), object(1)
memory usage: 728.0+ bytes
```

[65]:
```
CEN16.head()
```

[65]:
Fact	State	Population estimates, July 1, 2016, (V2016)
0	Alabama	4863300.0
1	Alaska	741894.0
2	Arizona	6931071.0
3	Arkansas	2988248.0
4	California	39250016.0

[66]:
```
#getting the sum of the 'Population estimates, July 1, 2016,  (V2016)' column
Cen16=CEN16['Population estimates, July 1, 2016,  (V2016)'].sum()
```

```
Cen16
```

[66]:
```
322446340.0
```

Next, I'm going to repeat the previous steps with 2010 data

[67]:
```
CEN10= CEN[['State','Population estimates base, April 1, 2010,  (V2016)']]
CEN10
```

[67]:

Fact	State	Population estimates base, April 1, 2010, (V2016)
0	Alabama	4,780,131
1	Alaska	710,249
2	Arizona	6,392,301
3	Arkansas	2,916,025
4	California	37,254,522
5	Colorado	5,029,324
6	Connecticut	3,574,114
7	Delaware	897,936
8	Florida	18,804,592
9	Georgia	9,688,680
10	Hawaii	1,360,301
11	Idaho	1,567,650
12	Illinois	12,831,574
13	Indiana	6,484,136
14	Iowa	3,046,869
15	Kansas	2,853,129
16	Kentucky	4,339,344
17	Louisiana	4,533,479
18	Maine	1,328,364
19	Maryland	5,773,786
20	Massachusetts	6,547,813
21	Michigan	9,884,129
22	Minnesota	5,303,924
23	Mississippi	2,968,103
24	Missouri	5,988,928
25	Montana	989,414
26	Nebraska	1,826,334
27	Nevada	2,700,691
28	New Hampshire	1,316,461
29	New Jersey	8,791,953
30	New Mexico	2059198
31	New York	19378110
32	North Carolina	9535688
33	North Dakota	672591
34	Ohio	11536727
35	Oklahoma	3751615
36	Oregon	3831072

37	Pennsylvania	12702857
38	Rhode Island	1052940
39	South Carolina	4625410
40	South Dakota	814195
41	Tennessee	6346298
42	Texas	25,146,100
43	Utah	2,763,888
44	Vermont	625,741
45	Virginia	8,001,041
46	Washington	6,724,545
47	West Virginia	1,853,011
48	Wisconsin	5,687,289
49	Wyoming	563,767

[68]:
```
CEN10['Population estimates base, April 1, 2010,  (V2016)'].
↪replace(to_replace=',',value=r'',regex=True,inplace=True)
```

```
/Users/yara/opt/anaconda3/envs/second_project/lib/python3.8/site-
packages/pandas/core/series.py:4563: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return super().replace(
```

[69]:
```
CEN10.head()
```

[69]:

Fact	State	Population estimates base, April 1, 2010, (V2016)
0	Alabama	4780131
1	Alaska	710249
2	Arizona	6392301
3	Arkansas	2916025
4	California	37254522

[70]:
```
CEN10['Population estimates base, April 1, 2010,  (V2016)']= pd.
↪to_numeric(CEN10['Population estimates base, April 1, 2010,  (V2016)'],␣
↪downcast='float', errors='ignore')
```

```
<ipython-input-70-c19d831c209f>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  CEN10['Population estimates base, April 1, 2010,  (V2016)']=
pd.to_numeric(CEN10['Population estimates base, April 1, 2010,  (V2016)'],
downcast='float', errors='ignore')
```

[71]:
```
CEN10.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 2 columns):
 #   Column                                          Non-Null Count  Dtype
---  ------                                          --------------  -----
 0   State                                           50 non-null     object
 1   Population estimates base, April 1, 2010,  (V2016)  50 non-null     float32
dtypes: float32(1), object(1)
memory usage: 728.0+ bytes
```

[72]:
```
CEN10.head()
```

[72]:

Fact	State	Population estimates base, April 1, 2010, (V2016)
0	Alabama	4780131.0
1	Alaska	710249.0
2	Arizona	6392301.0
3	Arkansas	2916025.0
4	California	37254520.0

[73]:
```
Cen10 = CEN10['Population estimates base, April 1, 2010,  (V2016)'].sum()
Cen10
```

[73]:
```
308156350.0
```

Merging In this part, I'm going to merge the datasets.

Preparing the datasets for merging:

[74]:
```
CEN16.reset_index(drop=True).head()
```

[74]:

Fact	State	Population estimates, July 1, 2016, (V2016)
0	Alabama	4863300.0
1	Alaska	741894.0
2	Arizona	6931071.0
3	Arkansas	2988248.0
4	California	39250016.0

[75]:
```
CEN10.reset_index(drop=True).head()
```

[75]:

Fact	State	Population estimates base, April 1, 2010, (V2016)
0	Alabama	4780131.0
1	Alaska	710249.0
2	Arizona	6392301.0
3	Arkansas	2916025.0
4	California	37254520.0

[76]:
```
GUN2010= GUN2010.reset_index(drop=True)
GUN2010.head()
```

[76]:

	month	state	totals
0	2010-04-01	Alabama	20791
1	2010-04-01	Alaska	6411
2	2010-04-01	Arizona	16578
3	2010-04-01	Arkansas	14563
4	2010-04-01	California	80750

[77]:
```
GUN2016=GUN2016.reset_index(drop= True)
GUN2016.head()
```

[77]:

	month	state	totals
0	2016-07-01	Alabama	48927
1	2016-07-01	Alaska	6793
2	2016-07-01	Arizona	34496
3	2016-07-01	Arkansas	19378
4	2016-07-01	California	190218

[78]:
```
#merging the Census 2010 and 2016 data to use it in answering question1
CenMerged = CEN10.merge(CEN16, on='State', how='left')
CenMerged.head()
```

[78]:

Fact	State	Population estimates base, April 1, 2010, (V2016)	Population estimates, July 1, 2016, (V2016)
0	Alabama	4780131.0	4863300.0
1	Alaska	710249.0	741894.0
2	Arizona	6392301.0	6931071.0
3	Arkansas	2916025.0	2988248.0
4	California	37254520.0	39250016.0

[79]:
```
CenMerged.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 50 entries, 0 to 49
Data columns (total 3 columns):
 #   Column                                            Non-Null Count  Dtype
---  ------                                            --------------  -----
 0   State                                             50 non-null     object
 1   Population estimates base, April 1, 2010,  (V2016)  50 non-null     float32
```

```
 2   Population estimates, July 1, 2016,  (V2016)        50 non-null     float32
dtypes: float32(2), object(1)
memory usage: 1.2+ KB
```

[80]:
```
##merging the Gun 2010 and 2016 data to use it in answering question1
GunMerged= GUN2010.merge(GUN2016, on='state', how='left')
GunMerged.head()
```

[80]:

	month_x	state	totals_x	month_y	totals_y
0	2010-04-01	Alabama	20791	2016-07-01	48927
1	2010-04-01	Alaska	6411	2016-07-01	6793
2	2010-04-01	Arizona	16578	2016-07-01	34496
3	2010-04-01	Arkansas	14563	2016-07-01	19378
4	2010-04-01	California	80750	2016-07-01	190218

[81]:
```
GunMerged.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 50 entries, 0 to 49
Data columns (total 5 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   month_x   50 non-null     datetime64[ns]
 1   state     50 non-null     object
 2   totals_x  50 non-null     int64
 3   month_y   50 non-null     datetime64[ns]
 4   totals_y  50 non-null     int64
dtypes: datetime64[ns](2), int64(2), object(1)
memory usage: 2.3+ KB
```

[82]:
```
# a method to convert whatever value to percentage value
def prcnt(x):
    percentage= x*100
    return percentage
```

I created a method that calculates percentage of a value. I will needed it later.

1.4 3. Exploratory Data Analysis

1.4.1 3.1 Question 1: gun per capita

What is the approximated number of commoners gun registration per capita in 2010 and 2016

[83]:
```
#Gun per capita in 2010
Capita10= GunMerged['totals_x'].sum()/CenMerged['Population estimates base,␣
 ↪April 1, 2010,  (V2016)'].sum()
Capita10
```

[83]:
```
0.003978133152355075
```

What I did here is getting the sum of the **totals_x** column (which is the totals of the year 2010), then divide it by the sum of the **Population estimates base, April 1, 2010, (V2016)** column. The result is the number of gun registration per capita in 2010.

[84]:
```
#getting the percentage
prcnt(Capita10)
```

[84]:
```
0.39781331523550745
```

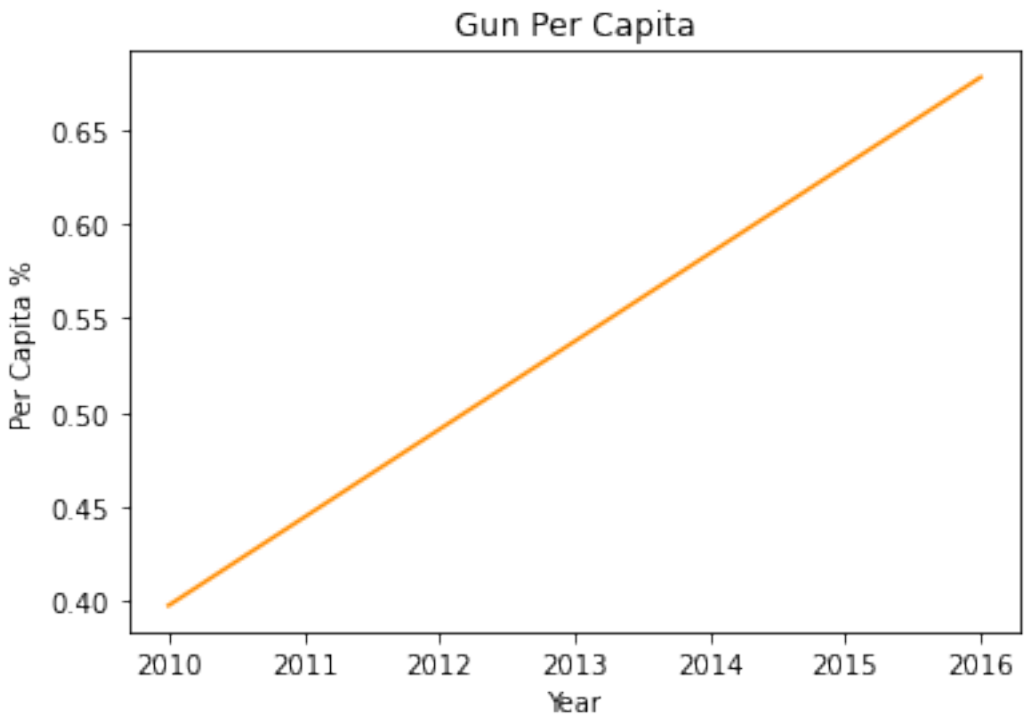
[85]:
```
#Gun per capita in 2016
Capita16= GunMerged['totals_y'].sum()/CenMerged['Population estimates, July 1,␣
↪2016,  (V2016)'].sum()
Capita16
```

[85]:
```
0.00677793404977627
```

[86]:
```
#getting the percentage
prcnt(Capita16)
```

[86]:
```
0.677793404977627
```

[87]:
```
#ploting the relationship of gun per capita for 2010 and 2016
PerCapita= [0.39781331523550745, 0.677793404977627]
Year= [2010,2016]

plt.title('Gun Per Capita')
plt.xlabel('Year')
plt.ylabel('Per Capita %')
plt.plot(Year, PerCapita,'darkorange')
plt.show()
```



1.4.2 Answers:

- 2010's gun per capita is 0.39781331523550745%
- 2016's gun per capita is 0.677793404977627%

And as the chart shows, it has increased since 2010 by approximetly 70%

1.4.3 3.2 Question 2: gun registrations

Which states have had the highest growth in gun registrations?

[116]:
```
gun1.head()
```

[116]:

	month	state	permit	permit_recheck	handgun	long_gun	other	\
0	2020-09	Alabama	33228.0	642.0	23455.0	17369.0	1633.0	
1	2020-09	Alaska	388.0	2.0	3275.0	3333.0	345.0	
2	2020-09	Arizona	8786.0	1198.0	23996.0	12094.0	1963.0	
3	2020-09	Arkansas	3686.0	554.0	9214.0	8003.0	505.0	
4	2020-09	California	32998.0	0.0	61258.0	36638.0	7815.0	

	multiple	admin	prepawn_handgun	…	returned_other	rentals_handgun	\
0	981	0.0	35.0	…	0.0	0.0	
1	201	0.0	1.0	…	0.0	0.0	
2	1873	0.0	16.0	…	0.0	0.0	
3	383	10.0	8.0	…	0.0	0.0	

```
4         0    0.0              0.0  …             0.0              0.0

   rentals_long_gun  private_sale_handgun  private_sale_long_gun  \
0               0.0                  30.0                   19.0
1               0.0                   8.0                   16.0
2               0.0                  39.0                   13.0
3               0.0                   3.0                   12.0
4               0.0                   0.0                    0.0

   private_sale_other  return_to_seller_handgun  return_to_seller_long_gun  \
0                 8.0                       1.0                        2.0
1                 2.0                       1.0                        1.0
2                 5.0                       0.0                        0.0
3                 3.0                       0.0                        0.0
4                 0.0                       0.0                        0.0

   return_to_seller_other  totals
0                     0.0   80478
1                     0.0    7897
2                     0.0   51287
3                     0.0   24043
4                     0.0  139313

[5 rows x 27 columns]
```

[89]:
```
#craeting a dataframe that copies only the dates that starts with 2020 from the Gun datset
G2020 = gun1[gun1['month'].str.match('2020')]
G2020.head()
```

[89]:
```
     month       state   permit  permit_recheck  handgun  long_gun   other  \
0  2020-09     Alabama  33228.0           642.0  23455.0   17369.0  1633.0
1  2020-09      Alaska    388.0             2.0   3275.0    3333.0   345.0
2  2020-09     Arizona   8786.0          1198.0  23996.0   12094.0  1963.0
3  2020-09    Arkansas   3686.0           554.0   9214.0    8003.0   505.0
4  2020-09  California  32998.0             0.0  61258.0   36638.0  7815.0

   multiple  admin  prepawn_handgun  …  returned_other  rentals_handgun  \
0       981    0.0             35.0  …             0.0              0.0
1       201    0.0              1.0  …             0.0              0.0
2      1873    0.0             16.0  …             0.0              0.0
3       383   10.0              8.0  …             0.0              0.0
4         0    0.0              0.0  …             0.0              0.0

   rentals_long_gun  private_sale_handgun  private_sale_long_gun  \
0               0.0                  30.0                   19.0
1               0.0                   8.0                   16.0
```

```
2                0.0                  39.0                   13.0
3                0.0                   3.0                   12.0
4                0.0                   0.0                    0.0

   private_sale_other  return_to_seller_handgun  return_to_seller_long_gun  \
0                 8.0                       1.0                        2.0
1                 2.0                       1.0                        1.0
2                 5.0                       0.0                        0.0
3                 3.0                       0.0                        0.0
4                 0.0                       0.0                        0.0

   return_to_seller_other  totals
0                     0.0   80478
1                     0.0    7897
2                     0.0   51287
3                     0.0   24043
4                     0.0  139313

[5 rows x 27 columns]
```

[90]:
```
G2020.tail()
```

[90]:
```
       month          state   permit  permit_recheck  handgun  long_gun  \
490  2020-01       Virginia    360.0           361.0  36391.0   21424.0
491  2020-01     Washington  15767.0           429.0  18390.0    8842.0
492  2020-01  West Virginia   2697.0           135.0   6543.0    5664.0
493  2020-01      Wisconsin  11587.0           731.0  14521.0    9835.0
494  2020-01        Wyoming    550.0             1.0   2135.0    1656.0

      other  multiple  admin  prepawn_handgun  …  returned_other  \
490  9713.0         0    4.0              0.0  …             2.0
491  3813.0        51   24.0              1.0  …            14.0
492   422.0       432    1.0             11.0  …             1.0
493  1168.0        49    0.0              0.0  …             3.0
494   185.0       107    1.0              0.0  …             0.0

     rentals_handgun  rentals_long_gun  private_sale_handgun  \
490              0.0               0.0                   0.0
491              0.0               0.0                 271.0
492              0.0               0.0                   9.0
493              0.0               0.0                   0.0
494              0.0               0.0                  68.0

     private_sale_long_gun  private_sale_other  return_to_seller_handgun  \
490                    0.0                0.0                       0.0
491                  208.0               84.0                       1.0
492                    4.0                1.0                       0.0
```

```
493                  12.0                  0.0                       0.0
494                  13.0                  0.0                       0.0

     return_to_seller_long_gun  return_to_seller_other  totals
490                        0.0                     0.0   68420
491                        7.0                     1.0   49714
492                        0.0                     0.0   17974
493                        0.0                     0.0   38349
494                        2.0                     0.0    4984

[5 rows x 27 columns]
```

This way I only extracted the data of the year 2020.

Source: https://stackoverflow.com/questions/17957890/pandas-select-from-dataframe-using-startswith

[91]:
```
#sorting the total in a descending way based on the totals column
G2020.sort_values(by=['totals'], axis = 0, ascending = False).head(30)
```

[91]:

	month	state	permit	permit_recheck	handgun	long_gun	other	\
124	2020-07	Illinois	42167.0	626794.0	31853.0	12250.0	0.0	
179	2020-06	Illinois	73508.0	566780.0	49101.0	14913.0	0.0	
234	2020-05	Illinois	25190.0	598361.0	26854.0	9959.0	0.0	
344	2020-03	Illinois	39780.0	541682.0	44112.0	17127.0	0.0	
289	2020-04	Illinois	25857.0	563741.0	30500.0	11184.0	0.0	
399	2020-02	Illinois	13769.0	572881.0	22743.0	9761.0	0.0	
454	2020-01	Illinois	17551.0	561765.0	22567.0	9588.0	0.0	
69	2020-08	Illinois	39261.0	504311.0	28864.0	13330.0	0.0	
14	2020-09	Illinois	33632.0	344401.0	27645.0	13848.0	0.0	
403	2020-02	Kentucky	1993.0	365823.0	15294.0	8763.0	668.0	
183	2020-06	Kentucky	1175.0	351397.0	27051.0	11707.0	993.0	
238	2020-05	Kentucky	466.0	357737.0	20881.0	11275.0	823.0	
458	2020-01	Kentucky	2314.0	340741.0	12022.0	7558.0	704.0	
128	2020-07	Kentucky	1481.0	255511.0	20962.0	10578.0	947.0	
376	2020-03	Texas	35688.0	0.0	147714.0	61827.0	5897.0	
348	2020-03	Kentucky	2394.0	182977.0	28268.0	15315.0	937.0	
211	2020-06	Texas	41486.0	0.0	121926.0	45845.0	6857.0	
156	2020-07	Texas	58415.0	0.0	96689.0	41060.0	6431.0	
174	2020-06	Florida	20676.0	0.0	133285.0	39111.0	7341.0	
70	2020-08	Indiana	1199.0	156039.0	28843.0	16235.0	2226.0	
293	2020-04	Kentucky	1555.0	165016.0	20712.0	12196.0	820.0	
15	2020-09	Indiana	1254.0	155556.0	26530.0	15978.0	1955.0	
321	2020-04	Texas	38817.0	0.0	93980.0	40921.0	5668.0	
339	2020-03	Florida	16713.0	0.0	117900.0	38365.0	5017.0	
119	2020-07	Florida	33826.0	0.0	106745.0	33102.0	7751.0	
101	2020-08	Texas	50347.0	0.0	77807.0	43770.0	5975.0	
266	2020-05	Texas	41584.0	0.0	83977.0	36456.0	5393.0	

```
46   2020-09     Texas  46746.0            0.0   68804.0   46463.0  4888.0
125  2020-07   Indiana   1117.0       123289.0   32970.0   15066.0  2395.0
64   2020-08   Florida  35826.0            0.0   85345.0   30914.0  6509.0

     multiple  admin  prepawn_handgun  …  returned_other  rentals_handgun  \
124      1360    0.0              0.0  …             0.0              0.0
179      2102    0.0              0.0  …             0.0              0.0
234      1106    0.0              0.0  …             0.0              0.0
344      2408    0.0              0.0  …             0.0              0.0
289      1282    0.0              0.0  …             0.0              0.0
399       913    0.0              0.0  …             0.0              0.0
454       940    0.0              0.0  …             0.0              0.0
69       1458    0.0              0.0  …             0.0              0.0
14       1504    0.0              0.0  …             0.0              0.0
403      1096    1.0             34.0  …             0.0              0.0
183       957    3.0             19.0  …             0.0              0.0
238       933    1.0             13.0  …             0.0              0.0
458       655    0.0             27.0  …             0.0              0.0
128       759    1.0             18.0  …             0.0              0.0
376      8286    0.0            138.0  …             0.0              0.0
348      1615    2.0             32.0  …             0.0              0.0
211      3968    7.0             61.0  …             1.0              0.0
156      3366   22.0             76.0  …             0.0              0.0
174      4846    0.0             12.0  …            13.0              0.0
70       1038   15.0              4.0  …             0.0              0.0
293      1173    0.0             30.0  …             1.0              0.0
15        916   25.0              5.0  …             1.0              0.0
321      4490    2.0            104.0  …             0.0              0.0
339      6073    0.0             26.0  …             5.0              0.0
119      3832    0.0             13.0  …            25.0              0.0
101      2967    0.0             72.0  …             0.0              0.0
266      3142    3.0             78.0  …             0.0              0.0
46       2928    3.0             91.0  …             1.0              0.0
125      1083   26.0              4.0  …             1.0              0.0
64       3193    0.0              9.0  …            30.0              0.0

     rentals_long_gun  private_sale_handgun  private_sale_long_gun  \
124               0.0                   0.0                    0.0
179               0.0                   0.0                    0.0
234               0.0                   0.0                    0.0
344               0.0                   0.0                    0.0
289               0.0                   0.0                    0.0
399               0.0                   0.0                    0.0
454               0.0                   0.0                    0.0
69                0.0                   0.0                    0.0
14                0.0                   0.0                    0.0
403               0.0                  37.0                   24.0
```

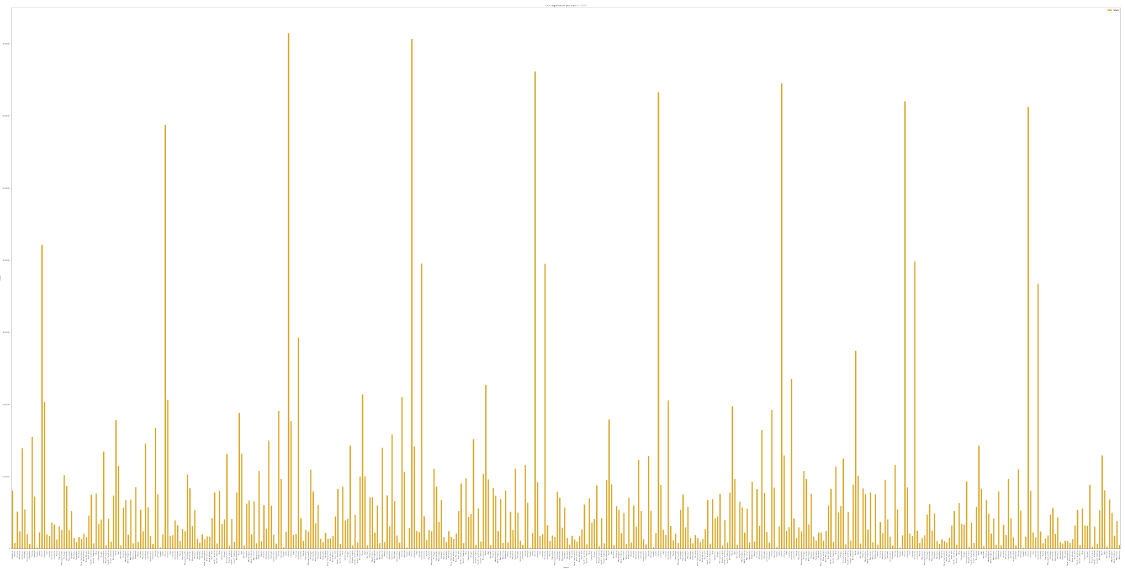
183	0.0	24.0	15.0
238	0.0	11.0	19.0
458	0.0	30.0	28.0
128	0.0	32.0	16.0
376	0.0	78.0	77.0
348	0.0	32.0	16.0
211	0.0	81.0	36.0
156	0.0	81.0	39.0
174	0.0	428.0	200.0
70	0.0	50.0	42.0
293	0.0	21.0	11.0
15	0.0	68.0	36.0
321	0.0	77.0	43.0
339	0.0	316.0	184.0
119	0.0	522.0	265.0
101	0.0	74.0	37.0
266	0.0	80.0	47.0
46	0.0	95.0	65.0
125	0.0	106.0	53.0
64	0.0	548.0	261.0

	private_sale_other	return_to_seller_handgun	return_to_seller_long_gun	\
124	0.0	0.0	0.0	
179	0.0	0.0	0.0	
234	0.0	0.0	0.0	
344	0.0	0.0	0.0	
289	0.0	0.0	0.0	
399	0.0	0.0	0.0	
454	0.0	0.0	0.0	
69	0.0	0.0	0.0	
14	0.0	0.0	0.0	
403	3.0	0.0	1.0	
183	1.0	0.0	0.0	
238	2.0	1.0	2.0	
458	3.0	1.0	1.0	
128	4.0	3.0	2.0	
376	14.0	2.0	0.0	
348	1.0	1.0	2.0	
211	8.0	3.0	3.0	
156	10.0	3.0	5.0	
174	66.0	43.0	31.0	
70	16.0	2.0	1.0	
293	2.0	0.0	1.0	
15	25.0	2.0	4.0	
321	7.0	1.0	2.0	
339	37.0	54.0	60.0	
119	100.0	36.0	28.0	

```
101                11.0                    0.0                    2.0
266                 5.0                    2.0                    2.0
46                  4.0                    0.0                    1.0
125                12.0                    5.0                    1.0
64                 91.0                   36.0                   34.0

     return_to_seller_other  totals
124                     0.0  714424
179                     0.0  706404
234                     0.0  661470
344                     0.0  645109
289                     0.0  632564
399                     0.0  620067
454                     0.0  612411
69                      0.0  587224
14                      0.0  421030
403                     0.0  398247
183                     0.0  395188
238                     1.0  394969
458                     1.0  367301
128                     1.0  292534
376                     1.0  274211
348                     0.0  235305
211                     2.0  227232
156                     0.0  213969
174                     1.0  210415
70                      0.0  206454
293                     0.0  205536
15                      0.0  203253
321                     0.0  197343
339                     3.0  192234
119                     1.0  190975
101                     0.0  188091
266                     0.0  179192
46                      0.0  178136
125                     0.0  176861
64                      0.0  167138

[30 rows x 27 columns]
```

[92]:
```
#plotting the relationship and finding the highest gun ownership in 2020 based␣
 ↪on the states
G2020.plot(x='state', y='totals', kind='bar',color='goldenrod',figsize =␣
 ↪(100,50), xlabel='States', ylabel='Totals',title='Gun registration per state␣
 ↪in 2020' )
```

[92]: <AxesSubplot:title={'center':'Gun registration per state in 2020'},
xlabel='States', ylabel='Totals'>



We can see from the graph that **Illinois** state is the highest state with gun ownership in 2020.

Next, I'm going to do the same with the years 2019, and 2018 to compare the three years.

[93]:
```
G2019=gun1[gun1['month'].str.match('2019')]
G2019.head()
```

[93]:

	month	state	permit	permit_recheck	handgun	long_gun	other	\
495	2019-12	Alabama	33683.0	485.0	33020.0	25882.0	1560.0	
496	2019-12	Alaska	386.0	13.0	3455.0	3080.0	397.0	
497	2019-12	Arizona	4536.0	649.0	18178.0	10423.0	1560.0	
498	2019-12	Arkansas	2339.0	638.0	8671.0	10957.0	531.0	
499	2019-12	California	29544.0	0.0	40637.0	30780.0	4883.0	

	multiple	admin	prepawn_handgun	…	returned_other	rentals_handgun	\
495	1382	0.0	42.0	…	0.0	0.0	
496	215	0.0	2.0	…	0.0	0.0	
497	1109	0.0	8.0	…	2.0	0.0	
498	458	3.0	10.0	…	0.0	0.0	
499	0	0.0	0.0	…	0.0	0.0	

	rentals_long_gun	private_sale_handgun	private_sale_long_gun	\
495	0.0	53.0	31.0	
496	0.0	18.0	13.0	
497	0.0	20.0	6.0	
498	0.0	10.0	16.0	

```
499              0.0                   0.0                   0.0

     private_sale_other  return_to_seller_handgun  return_to_seller_long_gun  \
495                 8.0                       1.0                        6.0
496                 1.0                       0.0                        0.0
497                 2.0                       0.0                        0.0
498                 3.0                       0.0                        1.0
499                 0.0                       0.0                        0.0

     return_to_seller_other  totals
495                     0.0  100100
496                     0.0    7840
497                     0.0   38171
498                     0.0   25699
499                     0.0  106600

[5 rows x 27 columns]
```

[94]:
```
G2019.tail()
```

[94]:
```
        month          state   permit  permit_recheck  handgun  long_gun  \
1150  2019-01       Virginia   1061.0           118.0  20600.0   12813.0
1151  2019-01     Washington  18889.0            53.0  17487.0   10889.0
1152  2019-01  West Virginia   2438.0             0.0   5355.0    4964.0
1153  2019-01      Wisconsin  11124.0           300.0  12204.0    9154.0
1154  2019-01        Wyoming    357.0            11.0   1530.0    1621.0

       other  multiple  admin  prepawn_handgun  …  returned_other  \
1150  3209.0         0    0.0              0.0  …             0.0
1151  3622.0       863    7.0             15.0  …            13.0
1152   379.0       382    4.0              8.0  …             0.0
1153  1310.0        43    0.0              0.0  …             3.0
1154   175.0        99    2.0              0.0  …             0.0

      rentals_handgun  rentals_long_gun  private_sale_handgun  \
1150              0.0               0.0                   0.0
1151              0.0               0.0                 776.0
1152              0.0               0.0                  15.0
1153              0.0               0.0                   0.0
1154              0.0               0.0                   3.0

      private_sale_long_gun  private_sale_other  return_to_seller_handgun  \
1150                    0.0                 0.0                       0.0
1151                  622.0                85.0                      15.0
1152                    9.0                 0.0                       0.0
1153                    4.0                 0.0                       0.0
1154                    4.0                 1.0                       0.0
```

```
      return_to_seller_long_gun  return_to_seller_other  totals
1150                        0.0                     0.0   37857
1151                        5.0                     2.0   56051
1152                        0.0                     0.0   15320
1153                        0.0                     0.0   34545
1154                        0.0                     0.0    4075

[5 rows x 27 columns]
```

[95]:
```
G2019.sort_values(by=['totals'], axis = 0, ascending = False).head(50)
```

[95]:

	month	state	permit	permit_recheck	handgun	long_gun	\
509	2019-12	Illinois	15241.0	493794.0	24792.0	13258.0	
729	2019-08	Illinois	13542.0	448997.0	19105.0	9281.0	
674	2019-09	Illinois	13011.0	442109.0	17639.0	9512.0	
619	2019-10	Illinois	12414.0	437512.0	17482.0	10040.0	
564	2019-11	Illinois	12356.0	403014.0	21597.0	12115.0	
784	2019-07	Illinois	10764.0	405293.0	15782.0	6889.0	
949	2019-04	Illinois	14158.0	360461.0	21307.0	8748.0	
513	2019-12	Kentucky	1160.0	369253.0	16185.0	14471.0	
568	2019-11	Kentucky	1372.0	360423.0	12094.0	12686.0	
894	2019-05	Illinois	13168.0	342314.0	17976.0	7587.0	
953	2019-04	Kentucky	361715.0	0.0	10133.0	5989.0	
1008	2019-03	Kentucky	350354.0	0.0	15646.0	8110.0	
1004	2019-03	Illinois	18595.0	313540.0	29438.0	13143.0	
839	2019-06	Illinois	12246.0	335793.0	17041.0	7321.0	
843	2019-06	Kentucky	150731.0	202171.0	8520.0	4608.0	
898	2019-05	Kentucky	315756.0	0.0	7899.0	4585.0	
1114	2019-01	Illinois	16976.0	281803.0	21587.0	10082.0	
1063	2019-02	Kentucky	303739.0	0.0	12027.0	7182.0	
788	2019-07	Kentucky	1627.0	301200.0	8059.0	4658.0	
733	2019-08	Kentucky	1628.0	291623.0	10802.0	6592.0	
1118	2019-01	Kentucky	290432.0	0.0	8170.0	6213.0	
623	2019-10	Kentucky	2238.0	279272.0	8759.0	8890.0	
895	2019-05	Indiana	110.0	266081.0	15062.0	8146.0	
678	2019-09	Kentucky	1675.0	265396.0	8526.0	6502.0	
840	2019-06	Indiana	168.0	247044.0	15272.0	7963.0	
1059	2019-02	Illinois	14499.0	139042.0	24870.0	11381.0	
541	2019-12	Texas	25456.0	0.0	69732.0	53386.0	
1115	2019-01	Indiana	107476.0	21459.0	17265.0	10423.0	
1036	2019-03	Texas	34837.0	0.0	55222.0	31747.0	
596	2019-11	Texas	23710.0	0.0	57252.0	41942.0	
1091	2019-02	Texas	37469.0	0.0	49130.0	28709.0	
761	2019-08	Texas	29242.0	0.0	49804.0	36943.0	
504	2019-12	Florida	15238.0	0.0	67435.0	31580.0	
994	2019-03	California	46812.0	0.0	42551.0	28146.0	

```
706   2019-09         Texas   27236.0            0.0  42688.0   35840.0
651   2019-10         Texas   28826.0            0.0  41994.0   33957.0
1146  2019-01         Texas   37167.0            0.0  39240.0   27842.0
829   2019-06    California   39303.0            0.0  38712.0   27050.0
939   2019-04    California   46004.0            0.0  37484.0   25895.0
999   2019-03       Florida   21078.0            0.0  59615.0   19429.0
884   2019-05    California   45237.0            0.0  36314.0   24718.0
926   2019-05         Texas   28928.0            0.0  38714.0   23587.0
724   2019-08       Florida   19468.0            0.0  54215.0   18910.0
559   2019-11       Florida   14070.0            0.0  54799.0   24497.0
499   2019-12    California   29544.0            0.0  40637.0   30780.0
981   2019-04         Texas   28638.0            0.0  38994.0   24009.0
950   2019-04       Indiana   41080.0        33562.0  18528.0    9718.0
1030  2019-03  Pennsylvania   27864.0            0.0  60068.0   16921.0
1054  2019-02       Florida   18437.0            0.0  51711.0   17833.0
871   2019-06         Texas   25136.0            0.0  37592.0   22608.0

       other  multiple   admin  prepawn_handgun  …  returned_other  \
509      0.0      1107     0.0              0.0  …             0.0
729      0.0       790     0.0              0.0  …             0.0
674      0.0       891     0.0              0.0  …             0.0
619      0.0       881     0.0              0.0  …             0.0
564      0.0      1002     0.0              0.0  …             0.0
784      0.0       730     0.0              0.0  …             0.0
949      0.0       790     0.0              0.0  …             0.0
513    729.0       753    42.0             31.0  …             0.0
568    528.0       738     5.0             29.0  …             0.0
894      0.0       726     0.0              0.0  …             0.0
953    390.0       452     4.0             18.0  …             0.0
1008   534.0       781     5.0             15.0  …             0.0
1004     0.0      1302     0.0              0.0  …             0.0
839      0.0       765     0.0              0.0  …             0.0
843    384.0       514     5.0             23.0  …             0.0
898    343.0       511     2.0             17.0  …             0.0
1114     0.0       877     0.0              0.0  …             0.0
1063   443.0       932     3.0             40.0  …             0.0
788    370.0       451     0.0             34.0  …             0.0
733    475.0       592     1.0             32.0  …             0.0
1118   510.0       450     4.0             22.0  …             0.0
623    455.0       612     0.0             22.0  …             0.0
895   1431.0       619    22.0              3.0  …             0.0
678    489.0       527     0.0             19.0  …             0.0
840   1368.0       705    23.0              7.0  …             0.0
1059     0.0      1094     0.0              0.0  …             0.0
541   4713.0      3398     4.0             56.0  …             0.0
1115  1738.0       681    32.0              7.0  …             1.0
1036  4092.0      2938    63.0             87.0  …             0.0
```

```
596   3592.0      3519     0.0             81.0  …             0.0
1091  3663.0      3149     5.0             88.0  …             0.0
761   3855.0      2652     1.0             90.0  …             1.0
504   4458.0      3150     0.0             36.0  …            25.0
994   4299.0         0     1.0              0.0  …             0.0
706   3736.0      2477     4.0             78.0  …             0.0
651   3423.0      2464     1.0             74.0  …             0.0
1146  3727.0      2179     2.0             66.0  …             0.0
829   4856.0         0  5040.0              0.0  …             0.0
939   4204.0         0     0.0              0.0  …             0.0
999   3869.0      2526     0.0             41.0  …             2.0
884   4700.0         0     0.0              0.0  …             0.0
926   3118.0      3993     0.0             72.0  …             0.0
724   4711.0      2632     0.0             38.0  …            14.0
559   3666.0      2924     0.0             41.0  …            27.0
499   4883.0         0     0.0              0.0  …             0.0
981   3451.0      2167     0.0             72.0  …             0.0
950   1493.0       686    22.0              8.0  …             0.0
1030    29.0         0   300.0              0.0  …             0.0
1054  3649.0      2315     4.0             50.0  …             3.0
871   3351.0      4335     2.0             61.0  …             0.0

      rentals_handgun  rentals_long_gun  private_sale_handgun  \
509               0.0               0.0                   0.0
729               0.0               0.0                   0.0
674               0.0               0.0                   0.0
619               0.0               0.0                   0.0
564               0.0               0.0                   0.0
784               0.0               0.0                   0.0
949               0.0               0.0                   0.0
513               0.0               0.0                  31.0
568               0.0               0.0                  20.0
894               0.0               0.0                   0.0
953               0.0               0.0                  24.0
1008              0.0               0.0                  33.0
1004              0.0               0.0                   0.0
839               0.0               0.0                   0.0
843               0.0               0.0                  14.0
898               0.0               0.0                  21.0
1114              0.0               0.0                   0.0
1063              0.0               0.0                  26.0
788               0.0               0.0                  21.0
733               0.0               0.0                  33.0
1118              0.0               0.0                  22.0
623               0.0               0.0                  17.0
895               0.0               0.0                  62.0
678               0.0               0.0                  26.0
```

840	0.0	0.0	46.0
1059	0.0	0.0	0.0
541	0.0	0.0	61.0
1115	0.0	0.0	45.0
1036	0.0	0.0	65.0
596	0.0	0.0	60.0
1091	0.0	0.0	67.0
761	0.0	0.0	76.0
504	0.0	0.0	292.0
994	0.0	0.0	0.0
706	0.0	0.0	79.0
651	0.0	0.0	64.0
1146	0.0	0.0	69.0
829	0.0	0.0	0.0
939	0.0	0.0	0.0
999	0.0	0.0	245.0
884	0.0	0.0	0.0
926	0.0	0.0	43.0
724	0.0	0.0	271.0
559	0.0	0.0	197.0
499	0.0	0.0	0.0
981	0.0	0.0	45.0
950	0.0	0.0	37.0
1030	0.0	0.0	0.0
1054	0.0	0.0	220.0
871	0.0	0.0	58.0

	private_sale_long_gun	private_sale_other	return_to_seller_handgun	\
509	0.0	0.0	0.0	
729	0.0	0.0	0.0	
674	0.0	0.0	0.0	
619	0.0	0.0	0.0	
564	0.0	0.0	0.0	
784	0.0	0.0	0.0	
949	0.0	0.0	0.0	
513	32.0	3.0	1.0	
568	27.0	0.0	0.0	
894	0.0	0.0	0.0	
953	23.0	2.0	1.0	
1008	26.0	1.0	0.0	
1004	0.0	0.0	0.0	
839	0.0	0.0	0.0	
843	14.0	1.0	0.0	
898	19.0	1.0	1.0	
1114	0.0	0.0	0.0	
1063	22.0	4.0	1.0	
788	7.0	0.0	0.0	

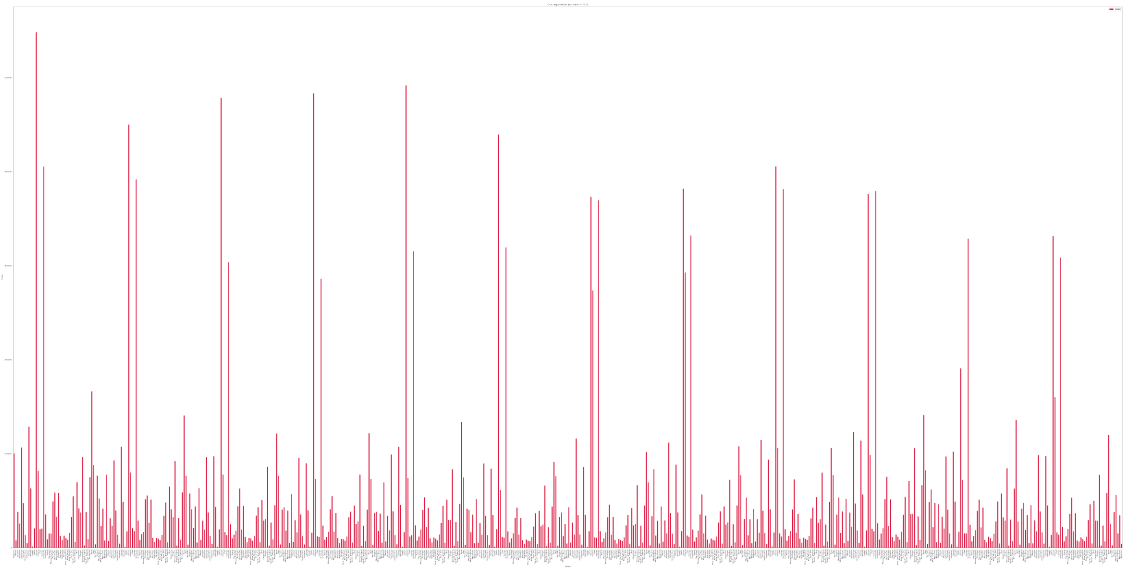
733	22.0	3.0	0.0
1118	13.0	4.0	0.0
623	18.0	2.0	1.0
895	29.0	19.0	1.0
678	20.0	4.0	0.0
840	22.0	36.0	2.0
1059	0.0	0.0	0.0
541	60.0	9.0	0.0
1115	50.0	19.0	1.0
1036	57.0	6.0	3.0
596	47.0	6.0	3.0
1091	50.0	10.0	1.0
761	47.0	11.0	3.0
504	172.0	39.0	68.0
994	0.0	0.0	0.0
706	58.0	10.0	2.0
651	54.0	6.0	1.0
1146	58.0	13.0	1.0
829	0.0	0.0	0.0
939	0.0	0.0	0.0
999	118.0	46.0	49.0
884	0.0	0.0	0.0
926	24.0	8.0	2.0
724	137.0	41.0	32.0
559	138.0	29.0	53.0
499	0.0	0.0	0.0
981	29.0	7.0	0.0
950	30.0	16.0	0.0
1030	0.0	0.0	0.0
1054	107.0	41.0	45.0
871	19.0	9.0	3.0

	return_to_seller_long_gun	return_to_seller_other	totals
509	0.0	0.0	548192
729	0.0	0.0	491715
674	0.0	0.0	483162
619	0.0	0.0	478329
564	0.0	0.0	450084
784	0.0	0.0	439458
949	0.0	0.0	405464
513	1.0	2.0	405291
568	2.0	0.0	391747
894	0.0	0.0	381771
953	1.0	0.0	381411
1008	0.0	0.0	379268
1004	0.0	0.0	376018
839	0.0	0.0	373166

```
843                    1.0                    0.0  369622
898                    1.0                    0.0  332076
1114                   0.0                    0.0  331325
1063                   1.0                    0.0  328572
788                    1.0                    0.0  319270
733                    1.0                    0.0  315348
1118                   1.0                    0.0  308573
623                    0.0                    0.0  303650
895                    2.0                    0.0  292587
678                    2.0                    1.0  286064
840                    4.0                    0.0  273660
1059                   0.0                    0.0  190886
541                    1.0                    1.0  166241
1115                   3.0                    0.0  160140
1036                   0.0                    0.0  141056
596                    3.0                    0.0  140598
1091                   2.0                    0.0  135931
761                    2.0                    0.0  133647
504                   57.0                    0.0  128789
994                    0.0                    0.0  123056
706                    0.0                    0.0  121718
651                    4.0                    3.0  121299
1146                   0.0                    0.0  119902
829                    0.0                    0.0  115987
939                    0.0                    0.0  114466
999                   30.0                    1.0  114001
884                    0.0                    0.0  111816
926                    3.0                    0.0  107763
724                   45.0                    1.0  107318
559                   39.0                    0.0  107274
499                    0.0                    0.0  106600
981                    0.0                    0.0  106112
950                    1.0                    0.0  106096
1030                   0.0                    0.0  105892
1054                  32.0                    1.0  101991
871                    2.0                    2.0  101762

[50 rows x 27 columns]
```

[96]:
```
#plotting the relationship and finding the highest gun ownership in 2019 based␣
↪on the states
G2019.plot(x='state', y='totals', kind='bar',color='crimson',figsize =␣
↪(100,50), xlabel='States', ylabel='Totals',title='Gun registration per state␣
↪in 2019' )
```

[96]:
```
<AxesSubplot:title={'center':'Gun registration per state in 2019'},
xlabel='States', ylabel='Totals'>
```



We can see from the graph that **Illinois** state is the highest state with gun ownership in 2019.

[97]:
```
G2018=gun1[gun1['month'].str.match('2018')]
G2018.head()
```

[97]:

	month	state	permit	permit_recheck	handgun	long_gun	other	\
1155	2018-12	Alabama	26029.0	242.0	11463.0	13578.0	464.0	
1156	2018-12	Alaska	215.0	11.0	2956.0	3278.0	312.0	
1157	2018-12	Arizona	4732.0	394.0	15533.0	11436.0	1451.0	
1158	2018-12	Arkansas	2498.0	806.0	8138.0	11736.0	369.0	
1159	2018-12	California	30754.0	0.0	40445.0	34180.0	3808.0	

	multiple	admin	prepawn_handgun	…	returned_other	rentals_handgun	\
1155	417	0.0	11.0	…	0.0	0.0	
1156	155	0.0	0.0	…	0.0	0.0	
1157	874	0.0	10.0	…	2.0	0.0	
1158	398	58.0	9.0	…	0.0	0.0	
1159	0	0.0	0.0	…	0.0	0.0	

	rentals_long_gun	private_sale_handgun	private_sale_long_gun	\
1155	0.0	18.0	12.0	
1156	0.0	11.0	10.0	
1157	0.0	15.0	12.0	
1158	0.0	14.0	30.0	
1159	0.0	0.0	0.0	

	private_sale_other	return_to_seller_handgun	return_to_seller_long_gun	\
1155	2.0	1.0	2.0	
1156	2.0	0.0	0.0	

```
1157                1.0                     1.0                     0.0
1158                3.0                     3.0                     2.0
1159                0.0                     0.0                     0.0

      return_to_seller_other  totals
1155                     1.0   54383
1156                     0.0    7251
1157                     1.0   36275
1158                     0.0   26272
1159                     0.0  110017

[5 rows x 27 columns]
```

[98]:
```
G2018.tail()
```

[98]:
```
        month          state   permit  permit_recheck  handgun  long_gun  \
1810  2018-01       Virginia   1109.0             2.0  19443.0   12248.0
1811  2018-01     Washington  18112.0            76.0  16327.0    9811.0
1812  2018-01  West Virginia   2838.0             0.0   5128.0    4452.0
1813  2018-01      Wisconsin  16462.0             0.0  10484.0    8337.0
1814  2018-01        Wyoming    384.0            11.0   1460.0    1534.0

       other  multiple  admin  prepawn_handgun  …  returned_other  \
1810  2374.0         0    0.0              0.0  …             0.0
1811  2170.0       623    3.0             16.0  …             7.0
1812   316.0       308   23.0             24.0  …             1.0
1813   945.0        49    0.0              0.0  …             1.0
1814   138.0        91    4.0              1.0  …             0.0

      rentals_handgun  rentals_long_gun  private_sale_handgun  \
1810              0.0               0.0                   0.0
1811              0.0               0.0                 919.0
1812              0.0               0.0                  25.0
1813              0.0               0.0                   0.0
1814              0.0               0.0                  11.0

      private_sale_long_gun  private_sale_other  return_to_seller_handgun  \
1810                    0.0                 0.0                       0.0
1811                  666.0                51.0                       6.0
1812                   28.0                 2.0                       0.0
1813                   35.0                 0.0                       0.0
1814                    9.0                 0.0                       1.0

      return_to_seller_long_gun  return_to_seller_other  totals
1810                        0.0                     0.0   35206
1811                        9.0                     0.0   51432
1812                        1.0                     1.0   15102
```

```
1813                      0.0                     0.0   36791
1814                      3.0                     0.0    3971

[5 rows x 27 columns]
```

[99]:
```
G2018.sort_values(by=['totals'], axis = 0, ascending = False).head(30)
```

[99]:

	month	state	permit	permit_recheck	handgun	long_gun	\
1228	2018-11	Kentucky	393953.0	0.0	10506.0	12670.0	
1668	2018-03	Kentucky	390708.0	0.0	14642.0	10303.0	
1723	2018-02	Kentucky	387471.0	0.0	13917.0	9302.0	
1283	2018-10	Kentucky	394208.0	0.0	7065.0	8001.0	
1393	2018-08	Kentucky	393679.0	0.0	7250.0	6241.0	
1503	2018-06	Kentucky	395679.0	0.0	7130.0	4686.0	
1613	2018-04	Kentucky	390433.0	0.0	8893.0	6662.0	
1338	2018-09	Kentucky	393160.0	0.0	6603.0	6014.0	
1173	2018-12	Kentucky	371756.0	0.0	14189.0	14309.0	
1778	2018-01	Kentucky	384989.0	0.0	8202.0	5736.0	
1558	2018-05	Kentucky	379571.0	0.0	7620.0	4696.0	
1448	2018-07	Kentucky	379409.0	0.0	6306.0	4537.0	
1169	2018-12	Illinois	12795.0	246015.0	26943.0	15297.0	
1224	2018-11	Illinois	14214.0	244550.0	22978.0	14038.0	
1664	2018-03	Illinois	24775.0	196253.0	32326.0	17008.0	
1389	2018-08	Illinois	14583.0	199766.0	17992.0	9858.0	
1609	2018-04	Illinois	18258.0	184764.0	23977.0	10883.0	
1279	2018-10	Illinois	13700.0	194077.0	17916.0	10755.0	
1554	2018-05	Illinois	13239.0	193052.0	18631.0	8397.0	
1334	2018-09	Illinois	12726.0	189108.0	16749.0	9393.0	
1499	2018-06	Illinois	12711.0	186091.0	17755.0	7827.0	
1444	2018-07	Illinois	13684.0	160026.0	16840.0	7743.0	
1719	2018-02	Illinois	21425.0	124056.0	25626.0	12938.0	
1774	2018-01	Illinois	18881.0	118688.0	21821.0	9850.0	
1201	2018-12	Texas	28161.0	0.0	62560.0	56919.0	
1696	2018-03	Texas	37551.0	0.0	60134.0	38853.0	
1751	2018-02	Texas	36801.0	0.0	55634.0	35889.0	
1256	2018-11	Texas	29090.0	0.0	44763.0	43276.0	
1659	2018-03	Florida	27957.0	0.0	64230.0	30487.0	
1654	2018-03	California	50714.0	0.0	47425.0	32189.0	

	other	multiple	admin	prepawn_handgun	…	returned_other	\
1228	437.0	888	1.0	25.0	…	0.0	
1668	818.0	727	5.0	26.0	…	0.0	
1723	532.0	1001	4.0	30.0	…	0.0	
1283	360.0	523	7.0	18.0	…	0.0	
1393	306.0	490	4.0	18.0	…	0.0	
1503	304.0	523	0.0	22.0	…	0.0	
1613	439.0	488	3.0	16.0	…	0.0	

```
1338   300.0       448     1.0             25.0  …             0.0
1173   581.0       618     1.0             16.0  …             0.0
1778   323.0       394     0.0             16.0  …             0.0
1558   344.0       434     3.0             10.0  …             0.0
1448   295.0       341     2.0             17.0  …             0.0
1169     0.0       981     0.0              0.0  …             0.0
1224     0.0      1130     0.0              0.0  …             0.0
1664     0.0      1352     0.0              0.0  …             0.0
1389     0.0       665     0.0              0.0  …             0.0
1609     0.0       892     0.0              0.0  …             0.0
1279     0.0       776     0.0              0.0  …             0.0
1554     0.0       644     0.0              0.0  …             0.0
1334     0.0       719     0.0              0.0  …             0.0
1499     0.0       622     0.0              0.0  …             0.0
1444     0.0       678     0.0              0.0  …             0.0
1719     0.0      1243     0.0              0.0  …             0.0
1774     0.0       767     0.0              0.0  …             0.0
1201  4482.0      3138     1.0             90.0  …             0.0
1696  5634.0      3340  2558.0             97.0  …             0.0
1751  4158.0      3712  3496.0             91.0  …             0.0
1256  3437.0      9604     0.0             92.0  …             0.0
1659  6732.0      2808     1.0             10.0  …             1.0
1654  4802.0         0     0.0              0.0  …             0.0

      rentals_handgun  rentals_long_gun  private_sale_handgun  \
1228              0.0               0.0                  27.0
1668              0.0               0.0                  31.0
1723              0.0               0.0                  23.0
1283              0.0               0.0                  10.0
1393              0.0               0.0                  30.0
1503              0.0               0.0                  26.0
1613              0.0               0.0                   9.0
1338              0.0               0.0                  25.0
1173              0.0               0.0                  43.0
1778              0.0               0.0                  17.0
1558              0.0               0.0                  23.0
1448              0.0               0.0                  18.0
1169              0.0               0.0                   0.0
1224              0.0               0.0                   0.0
1664              0.0               0.0                   0.0
1389              0.0               0.0                   0.0
1609              0.0               0.0                   0.0
1279              0.0               0.0                   0.0
1554              0.0               0.0                   0.0
1334              0.0               0.0                   0.0
1499              0.0               0.0                   0.0
1444              0.0               0.0                   0.0
```

1719	0.0	0.0	0.0
1774	0.0	0.0	0.0
1201	0.0	0.0	102.0
1696	0.0	0.0	147.0
1751	0.0	0.0	124.0
1256	0.0	0.0	73.0
1659	0.0	0.0	24.0
1654	0.0	0.0	0.0

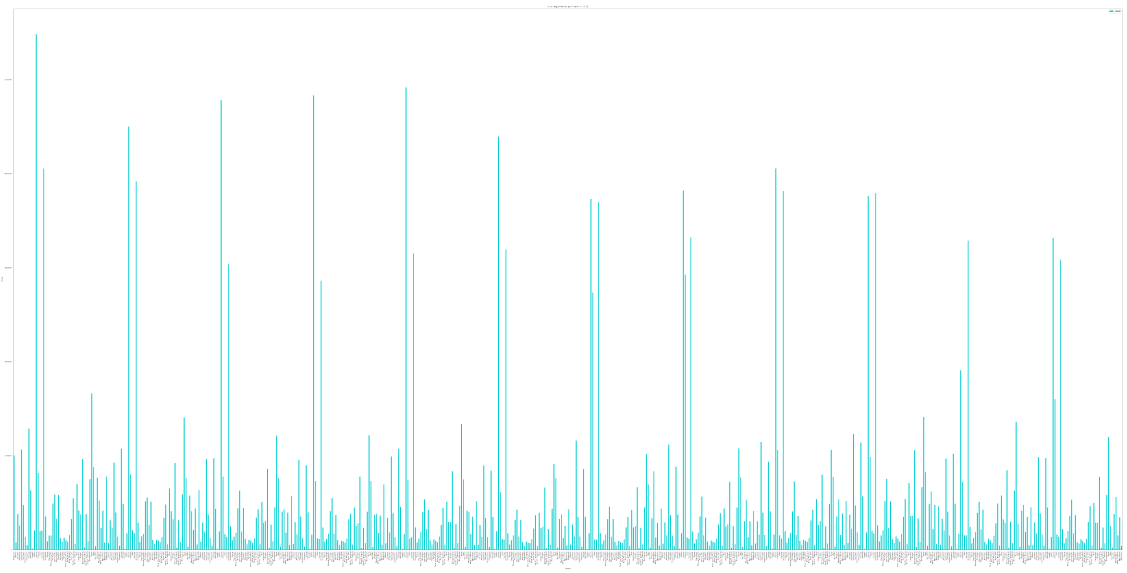
	private_sale_long_gun	private_sale_other	return_to_seller_handgun	\
1228	27.0	3.0	0.0	
1668	33.0	6.0	0.0	
1723	36.0	5.0	0.0	
1283	22.0	1.0	1.0	
1393	45.0	1.0	0.0	
1503	25.0	2.0	1.0	
1613	23.0	2.0	0.0	
1338	23.0	0.0	0.0	
1173	39.0	1.0	0.0	
1778	25.0	2.0	1.0	
1558	13.0	0.0	0.0	
1448	24.0	1.0	2.0	
1169	0.0	0.0	0.0	
1224	0.0	0.0	0.0	
1664	0.0	0.0	0.0	
1389	0.0	0.0	0.0	
1609	0.0	0.0	0.0	
1279	0.0	0.0	0.0	
1554	0.0	0.0	0.0	
1334	0.0	0.0	0.0	
1499	0.0	0.0	0.0	
1444	0.0	0.0	0.0	
1719	0.0	0.0	0.0	
1774	0.0	0.0	0.0	
1201	80.0	13.0	4.0	
1696	96.0	15.0	3.0	
1751	92.0	16.0	2.0	
1256	98.0	9.0	3.0	
1659	15.0	0.0	0.0	
1654	0.0	0.0	0.0	

	return_to_seller_long_gun	return_to_seller_other	totals
1228	1.0	0.0	422343
1668	2.0	1.0	421247
1723	1.0	0.0	417418
1283	0.0	0.0	413568
1393	1.0	0.0	411800

```
1503                    0.0                    0.0  411606
1613                    0.0                    1.0  409485
1338                    1.0                    0.0  409385
1173                    0.0                    0.0  404143
1778                    0.0                    0.0  402376
1558                    1.0                    0.0  395487
1448                    0.0                    0.0  393583
1169                    0.0                    0.0  302031
1224                    0.0                    0.0  296910
1664                    0.0                    0.0  271714
1389                    0.0                    0.0  242864
1609                    0.0                    0.0  238774
1279                    0.0                    0.0  237224
1554                    0.0                    0.0  233963
1334                    0.0                    0.0  228695
1499                    0.0                    0.0  225006
1444                    0.0                    0.0  198971
1719                    0.0                    0.0  185288
1774                    0.0                    0.0  170007
1201                    5.0                    0.0  165244
1696                    6.0                    0.0  161200
1751                    2.0                    0.0  155942
1256                    2.0                    0.0  141813
1659                    0.0                    0.0  137997
1654                    0.0                    0.0  136228

[30 rows x 27 columns]
```

[100]:
```
#plotting the relationship and finding the highest gun ownership in 2018 based␣
↪on the states
G2019.plot(x='state', y='totals', kind='bar',color='darkturquoise',figsize =␣
↪(100,50), xlabel='States', ylabel='Totals',title='Gun registration per state␣
↪in 2018' )
```

[100]:
```
<AxesSubplot:title={'center':'Gun registration per state in 2018'},
xlabel='States', ylabel='Totals'>
```



we can see from the graph that **Kentucky** state is the highest state with gun ownership in 2018.

1.4.4 Answers:

The states with the highest gun registration in 2018:

1. Kentucky.

2. Illinois.

3. Texas.

4. Florida.

5. California.

The states with the highest gun registration in 2019:

1. Illinois.

2. Kentucky.

3. Indiana.

4. Texas.

5. Florida.

The states with the highest gun registration in 2020:

the dataset stops at september, so these are the highest till now.

1. Illinois.

2. Kentucky.

3. Texas.

4. Florida.

5. Indiana.

And as we can see, these five states (Illinois, Kentucky, Texas, Florida and Indiana) are the top 5 for three years.

1.4.5 3.3 Question 3: Overall gun trend

What is the overall trend of gun purchases?

[101]:
```
#in 2020
G2020.head()
```

[101]:

	month	state	permit	permit_recheck	handgun	long_gun	other	multiple	admin	prepawn_handgun	…	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
0	2020-09	Alabama	33228.0	642.0	23455.0	17369.0	1633.0	981	0.0	35.0	…	0.0	0.0	0.0	30.0	19.0	8.0	1.0	2.0	0.0	80478
1	2020-09	Alaska	388.0	2.0	3275.0	3333.0	345.0	201	0.0	1.0	…	0.0	0.0	0.0	8.0	16.0	2.0	1.0	1.0	0.0	7897
2	2020-09	Arizona	8786.0	1198.0	23996.0	12094.0	1963.0	1873	0.0	16.0	…	0.0	0.0	0.0	39.0	13.0	5.0	0.0	0.0	0.0	51287
3	2020-09	Arkansas	3686.0	554.0	9214.0	8003.0	505.0	383	10.0	8.0	…	0.0	0.0	0.0	3.0	12.0	3.0	0.0	0.0	0.0	24043
4	2020-09	California	32998.0	0.0	61258.0	36638.0	7815.0	0	0.0	0.0	…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	139313

```
[5 rows x 27 columns]
```

The types on firearms are **handgun**, **long gun**, **other** and **multiple**. I'm going to find the sum of each one of them so I can plot the relationship and find out which firearme is moset purchased.

[102]:
```
G2020['handgun'].sum()
```

[102]:
```
9000541.0
```

[103]:
```
G2020['long_gun'].sum()
```

[103]:
```
4891755.0
```

[104]:
```
G2020['other'].sum()
```

[104]:
```
636067.0
```

[105]:
```
G2020['multiple'].sum()
```

[105]:
```
297110
```

[106]:
```
#creating a list of the firearms types
Type= ['Handgun', 'Long Gun', 'Other', 'Multiple']
```

[107]:
```
#creating a list of sum result
bought=[9000541, 4891755, 636067, 297110]
```

I created the two lists in order to plot the relationship.

[108]:
```
#to determine the x axis ticks
Type2=np.arange(len(Type))
Type2
```

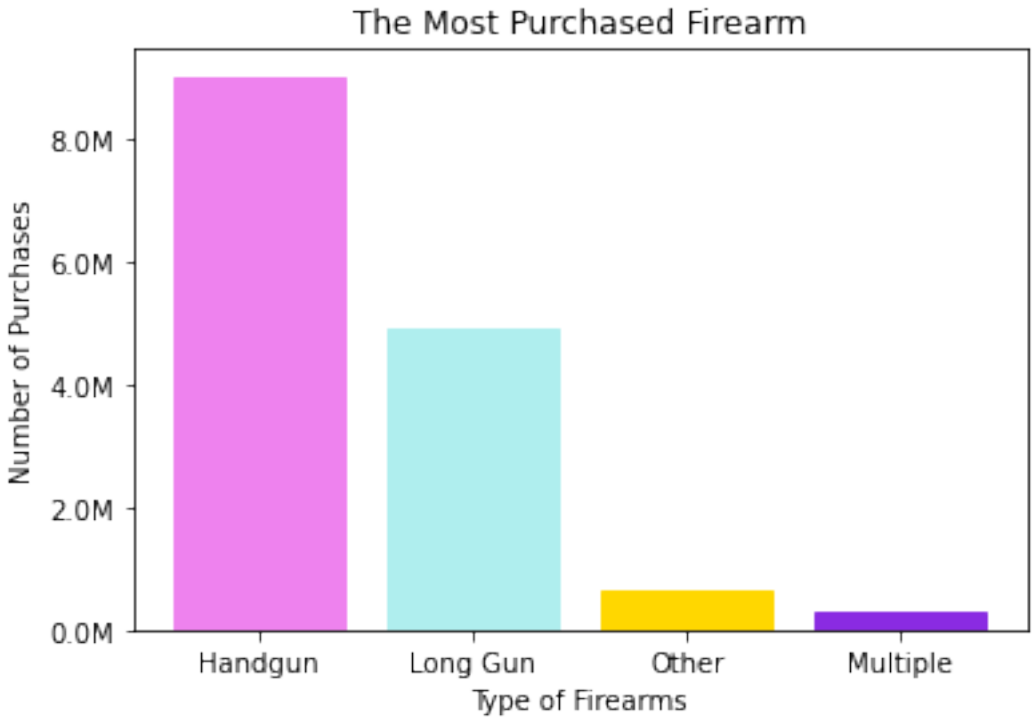
[108]:
```
array([0, 1, 2, 3])
```

[109]:
```
#this code is not necessary, I only wanted to present the values on the y axis␣
 ↪in a better way that's all.
def Million(Type2, pos):
    'The two args are the value and tick position'
    return '%1.1fM' % (Type2 * 1e-6)
```

[110]:
```
Format = FuncFormatter(Million)
```

[111]:
```
fig, ax = plt.subplots()
ax.yaxis.set_major_formatter(Format)
plt.title('The Most Purchased Firearm')
plt.ylabel('Number of Purchases')
plt.xlabel('Type of Firearms')
```

```
color=plt.bar(Type2, bought)
color[0].set_color('violet')
color[1].set_color('paleturquoise')
color[2].set_color('gold')
color[3].set_color('blueviolet')
plt.xticks(Type2, Type)
plt.show()
```



source: https://matplotlib.org/3.1.0/gallery/ticks_and_spines/custom_ticker1.html#sphx-glr-gallery-ticks-and-spines-custom-ticker1-py

source: https://www.semicolonworld.com/question/58048/setting-different-bar-color-in-matplotlib-python

source: https://matplotlib.org/3.1.0/gallery/color/named_colors.html

1.4.6 Answers:

The most purchased guns in the USA from the begining of this year (which is 2020) till now, arranged from highest to lowest:

1. Handguns, with about 9000541 checks.

2. Long guns, with about 4891755 checks.

3. Other, with about 636067 checks.

4. Multiple, whith about 297110 checks.

1.4.7 3.4 Question 4: the year with the highest and lowest checks

which year had the most checks, and by which state. and which had the least?

[112]:
```
#the most
gun1.sort_values(by=['totals'], axis = 0, ascending = False).head(100)
```

[112]:

	month	state	permit	permit_recheck	handgun	long_gun	\
124	2020-07	Illinois	42167.0	626794.0	31853.0	12250.0	
179	2020-06	Illinois	73508.0	566780.0	49101.0	14913.0	
234	2020-05	Illinois	25190.0	598361.0	26854.0	9959.0	
344	2020-03	Illinois	39780.0	541682.0	44112.0	17127.0	
289	2020-04	Illinois	25857.0	563741.0	30500.0	11184.0	
…	…	…	…	…	…	…	
3139	2015-12	California	119166.0	NaN	74399.0	48762.0	
4088	2014-07	Kentucky	232847.0	NaN	6375.0	5428.0	
5023	2013-02	Kentucky	205714.0	NaN	19072.0	13980.0	
4968	2013-03	Kentucky	210474.0	NaN	14921.0	12647.0	
1389	2018-08	Illinois	14583.0	199766.0	17992.0	9858.0	

	other	multiple	admin	prepawn_handgun	…	returned_other	\
124	0.0	1360	0.0	0.0	…	0.0	
179	0.0	2102	0.0	0.0	…	0.0	
234	0.0	1106	0.0	0.0	…	0.0	
344	0.0	2408	0.0	0.0	…	0.0	
289	0.0	1282	0.0	0.0	…	0.0	
…	…	…	…	…	…	…	
3139	9583.0	0	0.0	0.0	…	0.0	
4088	151.0	484	0.0	1.0	…	NaN	
5023	321.0	1246	1.0	13.0	…	NaN	
4968	337.0	735	0.0	6.0	…	NaN	
1389	0.0	665	0.0	0.0	…	0.0	

	rentals_handgun	rentals_long_gun	private_sale_handgun	\
124	0.0	0.0	0.0	
179	0.0	0.0	0.0	
234	0.0	0.0	0.0	
344	0.0	0.0	0.0	
289	0.0	0.0	0.0	
…	…	…	…	
3139	NaN	NaN	0.0	
4088	NaN	NaN	0.0	
5023	NaN	NaN	NaN	
4968	0.0	NaN	NaN	
1389	0.0	0.0	0.0	

```
      private_sale_long_gun  private_sale_other  return_to_seller_handgun  \
124                     0.0                0.0                       0.0
179                     0.0                0.0                       0.0
234                     0.0                0.0                       0.0
344                     0.0                0.0                       0.0
289                     0.0                0.0                       0.0
…                       …                  …                         …
3139                    0.0                0.0                       0.0
4088                    0.0                0.0                       0.0
5023                    NaN                NaN                       NaN
4968                    NaN                NaN                       NaN
1389                    0.0                0.0                       0.0

      return_to_seller_long_gun  return_to_seller_other  totals
124                         0.0                     0.0  714424
179                         0.0                     0.0  706404
234                         0.0                     0.0  661470
344                         0.0                     0.0  645109
289                         0.0                     0.0  632564
…                           …                       …    …
3139                        0.0                     0.0  252946
4088                        0.0                     0.0  248979
5023                        NaN                     NaN  247260
4968                        NaN                     NaN  243253
1389                        0.0                     0.0  242864

[100 rows x 27 columns]
```

[113]:
```
#the least
gun1.sort_values(by=['totals'], axis = 0).head(100)
```

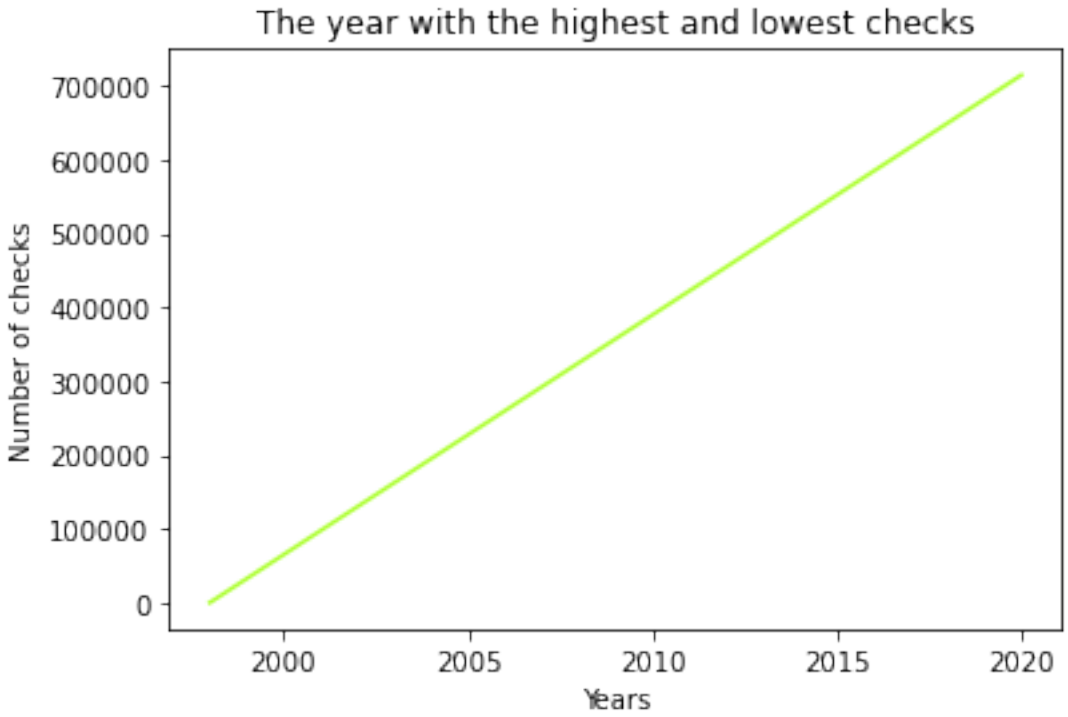
[113]:
```
         month           state  permit  permit_recheck  handgun  long_gun  \
14453  1998-11  South Carolina     0.0             NaN      0.0       6.0
14450  1998-11    Pennsylvania     0.0             NaN      5.0       8.0
14424  1998-11        Illinois     4.0             NaN      0.0      18.0
14460  1998-11        Virginia     0.0             NaN     14.0       2.0
14422  1998-11          Hawaii    27.0             NaN      0.0       1.0
…        …              …     …              …       …       …
11342  2003-07          Hawaii   584.0             NaN      0.0       0.0
11617  2003-02          Hawaii   584.0             NaN      0.0       0.0
12277  2002-02          Hawaii   586.0             NaN      0.0       0.0
12937  2001-02          Hawaii   587.0             NaN      0.0       0.0
14413  1998-11        Arkansas     0.0             NaN    149.0     429.0

       other  multiple  admin  prepawn_handgun  …  returned_other  \
14453    NaN         0    0.0              NaN  …             NaN
14450    NaN         4    0.0              NaN  …             NaN
```

```
14424    NaN          0    0.0              NaN  …              NaN
14460    NaN          8    0.0              NaN  …              NaN
14422    NaN          0    0.0              NaN  …              NaN
…        …        …    …                …  …              …
11342    NaN          0    0.0              0.0  …              NaN
11617    NaN          0    0.0              0.0  …              NaN
12277    NaN          0    0.0              0.0  …              NaN
12937    NaN          0    0.0              NaN  …              NaN
14413    NaN         11    0.0              NaN  …              NaN

       rentals_handgun  rentals_long_gun  private_sale_handgun  \
14453              NaN               NaN                   NaN
14450              NaN               NaN                   NaN
14424              NaN               NaN                   NaN
14460              NaN               NaN                   NaN
14422              NaN               NaN                   NaN
…                  …                 …                     …
11342              NaN               NaN                   NaN
11617              NaN               NaN                   NaN
12277              NaN               NaN                   NaN
12937              NaN               NaN                   NaN
14413              NaN               NaN                   NaN

       private_sale_long_gun  private_sale_other  return_to_seller_handgun  \
14453                    NaN                 NaN                       NaN
14450                    NaN                 NaN                       NaN
14424                    NaN                 NaN                       NaN
14460                    NaN                 NaN                       NaN
14422                    NaN                 NaN                       NaN
…                        …                   …                         …
11342                    NaN                 NaN                       NaN
11617                    NaN                 NaN                       NaN
12277                    NaN                 NaN                       NaN
12937                    NaN                 NaN                       NaN
14413                    NaN                 NaN                       NaN

       return_to_seller_long_gun  return_to_seller_other  totals
14453                        NaN                     NaN       6
14450                        NaN                     NaN      17
14424                        NaN                     NaN      22
14460                        NaN                     NaN      24
14422                        NaN                     NaN      28
…                            …                       …        …
11342                        NaN                     NaN     584
11617                        NaN                     NaN     584
12277                        NaN                     NaN     586
12937                        NaN                     NaN     587
```

```
14413                          NaN                          NaN      589

[100 rows x 27 columns]
```

[114]:
```
year=[1998,2020]
data=[gun1.totals.min(), gun1.totals.max()]
```

[115]:
```
plt.title('The year with the highest and lowest checks')
plt.xlabel('Years')
plt.ylabel('Number of checks')
plt.plot(year, data,'greenyellow')
plt.show()
```



1.4.8 Answers:

So far, **Illinois** state has the highest number of checks since the program initiated in 1998. It registered 714424 checks in July, 2020.

On the other hand, **South Carolina** state registered the lowest number of checks with only 6 checks in November,1998.

And as the figure shows, it has increased significantly since 1998 by about 11.9×10^6%.

1.5 4. Conclusion

In conclusion, both datasets share similar columns, such as the date and the states columns. The Analysis process showed that the gun registration per capita is equal to 0.3978133152355074 5% in 2010, while in 2016 it is equal to 0.677793404977627%. And that shows that the gun registration per capita has increased since 2010 by approximetly 70%.

Moreover, states such as Illinois, Kentucky, Texas, Florida, Indiana and California kept the record of being the top states with high gun registration in 2018, 2019 and 2020. Furthermore, the type of firearms that the citizens of the United States of America tend to purchase in 2020 are primarily handguns, then comes the long guns. Furthermore, since the FBI's National Instant Criminal Background Check System initiated in 1998, Illinois state registered the highest number of checkes in July, 2020 with 714424 checks. While South Carolina registered the lowest number of checkes in November, 1998, with only 6 checks.

There were quite few limitations and challenges in both datasets. For the Census dataset, there ware missing data that might contribute in the analysis accuracy, such as the five addressed colonies; District of Columbia, Guam, Mariana Islands, Puerto Rico and Virgin Islands. It would have been better if they were included in order to calculate the gun per capita appropriately and check the results. Moreover, Kentucky and Illinois are probably outliers among the the other states of U.S. The reason behind this assumption is that after the observations Kentucky and Illinois's records are noticeably higher compared to the other states in 2018, 2019 and 2020. From my perspective, it would be advisable and reasonable to verify the Kentucky and Illinois data again, especially Illinois since it has the strictest gun control in the U.S. Finally, at the time of this project the gun dataset was up to date, but not complete. I think it would be better to updata the dataset once the year is over and check if the results would change.

1.6 5. General Resources

- Data Wrangling with Python Tips and Tools to Make Your Life Easier by Jacqueline Kazil, Katharine Jarmul.
- Python for Data Analysis Data Wrangling with Pandas, NumPy, and IPython by Wes McKinney.
- https://matplotlib.org/3.3.2/index.html
- https://pandas.pydata.org/pandas-docs/stable/index.html