Project: Investigate FBI Guns Dataset.

Author: Shayan Bahrampour

Table of Contents

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. The data has been supplemented with state level data from census.gov.

Each column represents the type of transaction submitted to the National Instant Criminal Background Check System (NICS). In addition, each type of transaction is broken down by the type of firearm—handgun, long gun, and other. The types of firearms are defined by the Bureau of Alcohol, Tobacco, Firearms and Explosives as follows:

• *Handgun*—(a) any firearm which has a short stock and is designed to be held and fired by the use of a single hand; and (b) any combination of parts from which a firearm described in paragraph (a) can be assembled.

• *Long* Gun—a weapon designed or redesigned, made or remade, and intended to be fired from the shoulder, and designed or redesigned and made or remade to use the energy of the explosive in (a) a fixed metallic cartridge to fire a single projectile through a rifled bore for each single pull of the trigger; or (b) a fixed shotgun shell to fire through a smooth bore either a number of ball shot or a single projectile for each single pull of the trigger.

• *Other—refers* to frames, receivers, and other firearms that are neither handguns nor long guns (rifles or shotguns), such as firearms having a pistol grip that expel a shotgun shell, or National Firearms Act firearms, including silencers.

The transaction types indicated in yellow on the chart are for background checks initiated by an officially-licensed Federal Firearms Licensee (FFL) or criminal justice/law enforcement agency prior to the issuance of a firearm-related permit or transfer. The indication of “multiple” denotes a background check where more than one type of firearm is associated to a single background check and “admin” denotes the administrative checks that are for other authorized uses of the NICS.

Other types of transactions are explained below:

• *Pre-Pawn—background* checks requested by an officially-licensed FFL on prospective firearm transferees seeking to pledge or pawn a firearm as security for the payment or repayment of money, prior to actually pledging or pawning the firearm.

• *Redemption—background* checks requested by an officially-licensed FFL on prospective firearm transferees attempting to regain possession of a firearm after pledging or pawning a firearm as security at a pawn shop.

• *Returned/Disposition—background* checks requested by criminal justice/law enforcement agencies prior to returning a firearm in its possession to the respective transferee, to ensure the individual is not prohibited.

• *Rentals—background* checks requested by an officially-licensed FFL on prospective firearm transferees attempting to possess a firearm when the firearm is loaned or rented for use off the premises of the business.

• *Private Sale—background* checks requested by an officially-licensed FFL on prospective firearm transferees attempting to possess a firearm from a private party seller who is not an officially-licensed FFL.

• *Return to Seller-Private Sale—background* checks requested by an officially-licensed FFL on prospective firearm transferees attempting to possess a firearm from a private party seller who is not an officially-licensed FFL.

It is important to note that the statistics within this chart represent the number of firearm background checks initiated through the NICS. **They do not represent the number of firearms sold**. Based on varying state laws and purchase scenarios, a one-to-one correlation cannot be made between a firearm background check and a firearm sale.

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

Data Wrangling

> **Tip**: In this section of the report, I will load in the data, check for cleanliness, and then trim and clean dataset for analysis.

General Properties

In [2]:

```
gun_data=pd.read_csv(r'C:\Users\shaw\Desktop\FBI Gun Data\gun_data.csv')
gun_data.head()
```

Out[2]:

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

5 rows × 27 columns

In [3]:

```
print(gun_data.shape)
```

```
(12485, 27)
```

Checking what total is:

In [4]:

```
test=gun_data
col_list= list(gun_data)
col_list.remove('state')
col_list.remove('month')
col_list.remove('totals')
print(*col_list,sep=", ")
test['Final Test']=gun_data[col_list].sum(axis=1)
test.head()
```

```
permit, permit_recheck, handgun, long_gun, other, multiple, admin, prepawn_h
andgun, 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_g
un, private_sale_other, return_to_seller_handgun, return_to_seller_long_gun,
return_to_seller_other
```

Out[4]:

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

5 rows × 28 columns

In [5]:

```
test2=gun_data[['other','prepawn_other'
,'redemption_other','returned_other'
,'private_sale_other','return_to_seller_other']]
test2.head()
```

Out[5]:

	other	prepawn_other	redemption_other	returned_other	private_sale_other	return_to_seller_
0	221.0	2.0	1.0	0.0	3.0	
1	219.0	0.0	2.0	0.0	1.0	
2	920.0	0.0	3.0	0.0	2.0	
3	165.0	0.0	4.0	0.0	0.0	
4	2984.0	0.0	5.0	0.0	0.0	

UPDATE: Total is sumation of every checked backhgrounds(All the columns except "month" and "state")

In [6]:

```
gun_data.describe()
```

Out[6]:

	permit	permit_recheck	handgun	long_gun	other	multipl
count	12461.000000	1100.000000	12465.000000	12466.000000	5500.000000	12485.00000
mean	6413.629404	1165.956364	5940.881107	7810.847585	360.471636	268.60336
std	23752.338269	9224.200609	8618.584060	9309.846140	1349.478273	783.18507
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000
25%	0.000000	0.000000	865.000000	2078.250000	17.000000	15.00000
50%	518.000000	0.000000	3059.000000	5122.000000	121.000000	125.00000
75%	4272.000000	0.000000	7280.000000	10380.750000	354.000000	301.00000
max	522188.000000	116681.000000	107224.000000	108058.000000	77929.000000	38907.00000

8 rows × 26 columns

In [7]:

```
census_data=pd.read_csv(r'C:\Users\shaw\Desktop\FBI Gun Data\U.S. Census Data (2).csv')
census_data.head(3)
```

Out[7]:

	Fact	Fact Note	Alabama	Alaska	Arizona	Arkansas	California	Colorado	Connecticu
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
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
2	Population, percent change - April 1, 2010 (es...	NaN	1.70%	4.50%	8.40%	2.50%	5.40%	10.20%	0.10%

3 rows × 52 columns

In [8]:

```
array1=census_data['Fact']
print(*array1, sep = "\n")
print("##gun DataSet Columns##:"+"\n")
print(*gun_data.columns,sep=", ")
```

```
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,  (V
2016)
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+, 2011-20
15
In civilian labor force, female, percent of population age 16 years+, 2011-2
015
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
nan
NOTE: FIPS Code values are enclosed in quotes to ensure leading zeros remain
intact.
nan
Value Notes
1
nan
Fact Notes
(a)
(b)
(c)
nan
Value Flags
-
D
F
FN
nan
S
X
Z
##gun DataSet Columns##:

month, state, permit, permit_recheck, handgun, long_gun, other, multiple, ad
min, prepawn_handgun, prepawn_long_gun, prepawn_other, redemption_handgun, r
edemption_long_gun, redemption_other, returned_handgun, returned_long_gun, r
eturned_other, rentals_handgun, rentals_long_gun, private_sale_handgun, priv
ate_sale_long_gun, private_sale_other, return_to_seller_handgun, return_to_s
eller_long_gun, return_to_seller_other, totals, Final Test
```

DROP USELESS COLUMNS:

Check what "Fact Note" is:

In [9]:

```
census_factnote_values = census_data[~census_data['Fact Note'].isnull()]
print(*census_factnote_values['Fact Note'],sep=', ')
```

```
(a), (a), (a), (a), (a), (b), (c), (c), (c), (c), (c), (c), Includes data no
t distributed by county., Includes data not distributed by county., Includes
data not distributed by county., Includes data not distributed by county., I
ncludes data not distributed by county., Includes persons reporting only one
race, Hispanics may be of any race, so also are included in applicable race
categories, Economic Census - Puerto Rico data are not comparable to U.S. Ec
onomic Census data, Either no or too few sample observations were available
to compute an estimate, or a ratio of medians cannot be calculated because o
ne or both of the median estimates falls in the lowest or upper interval of
an open ended distribution., Suppressed to avoid disclosure of confidential
information, Fewer than 25 firms, Footnote on this item in place of data, No
t available, Suppressed; does not meet publication standards, Not applicabl
e, Value greater than zero but less than half unit of measure shown
```

UPDATE: MIGHT HAVE SOMETHING USEFUL HERE. SO WE WON'T DROP IT.

Joining Datasets in the right way:

In [10]:

```
population_list=[]
single_list=[]
col_state_list=census_data.columns
col_state_list=col_state_list[2:]
for i in range(len(col_state_list)):
    single_list.append(col_state_list[i]),
    single_list.append(census_data[col_state_list[i]].iloc[0]),
    single_list.append(census_data[col_state_list[i]].iloc[2])
    population_list.append(list(single_list))
    single_list=[]
```

In [11]:

```
cleaned_census=pd.DataFrame(population_list,columns=['state','population','churn rate'])
cleaned_census.head()
```

Out[11]:

	state	population	churn rate
0	Alabama	4,863,300	1.70%
1	Alaska	741,894	4.50%
2	Arizona	6,931,071	8.40%
3	Arkansas	2,988,248	2.50%
4	California	39,250,017	5.40%

In [12]:

```
final_dataset=gun_data.merge(cleaned_census,how='inner',on='state')
final_dataset.drop('Final Test',axis=1,inplace=True)
final_dataset.head()
```

Out[12]:

	month	state	permit	permit_recheck	handgun	long_gun	other	multiple	admin	prep
0	2017-09	Alabama	16717.0	0.0	5734.0	6320.0	221.0	317	0.0	
1	2017-08	Alabama	19733.0	4.0	6289.0	6045.0	216.0	311	0.0	
2	2017-07	Alabama	18042.0	1.0	6046.0	4790.0	224.0	258	0.0	
3	2017-06	Alabama	19508.0	89.0	8275.0	4782.0	254.0	334	0.0	
4	2017-05	Alabama	18538.0	313.0	7198.0	4559.0	254.0	309	0.0	

5 rows × 29 columns

In [13]:

```
final_dataset.permit_recheck.describe()
```

Out[13]:

```
count      1000.000000
mean       1282.552000
std        9667.124288
min           0.000000
25%           0.000000
50%           0.000000
75%           1.000000
max      116681.000000
Name: permit_recheck, dtype: float64
```

Data Cleaning

In [14]:

```
final_dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 11350 entries, 0 to 11349
Data columns (total 29 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   month                       11350 non-null  object 
 1   state                       11350 non-null  object 
 2   permit                      11348 non-null  float64
 3   permit_recheck              1000 non-null   float64
 4   handgun                     11350 non-null  float64
 5   long_gun                    11350 non-null  float64
 6   other                       5000 non-null   float64
 7   multiple                    11350 non-null  int64  
 8   admin                       11348 non-null  float64
 9   prepawn_handgun             9597 non-null   float64
 10  prepawn_long_gun            9595 non-null   float64
 11  prepawn_other               4650 non-null   float64
 12  redemption_handgun          9600 non-null   float64
 13  redemption_long_gun         9598 non-null   float64
 14  redemption_other            4650 non-null   float64
 15  returned_handgun            2000 non-null   float64
 16  returned_long_gun           1950 non-null   float64
 17  returned_other              1650 non-null   float64
 18  rentals_handgun             900 non-null    float64
 19  rentals_long_gun            750 non-null    float64
 20  private_sale_handgun        2500 non-null   float64
 21  private_sale_long_gun       2500 non-null   float64
 22  private_sale_other          2500 non-null   float64
 23  return_to_seller_handgun    2250 non-null   float64
 24  return_to_seller_long_gun   2500 non-null   float64
 25  return_to_seller_other      2050 non-null   float64
 26  totals                      11350 non-null  int64  
 27  population                  11350 non-null  object 
 28  churn rate                  11350 non-null  object 
dtypes: float64(23), int64(2), object(4)
memory usage: 2.6+ MB
```

Exploratory Data Analysis

Which states have the highest registrations in gun?

For finding the answer, We have to compare datas in the same timeline.

In [15]:

```
same_time=final_dataset[(final_dataset.month>='2010-01') & (final_dataset.month<='2016-01')
same_time.head()
```

...

In [16]:

```
list1=[]
pop_list=[]
states = []
check_highest=[]
states_names=[]
for j in range(len(same_time)):
    list1.append(same_time.state.iloc[j])
    list1.append(same_time.totals.iloc[j])
    list1.append(same_time.population.iloc[j])
    list1.append(same_time['churn rate'].iloc[j])
    check_highest.append(list(list1))
    list1=[]

highest = pd.DataFrame(check_highest, columns = ['state', 'total_regs','population','churn'
discover_highest=highest.groupby('state').sum()
for k in range(len(discover_highest)):
    pop_list.append(discover_highest['total_regs'].iloc[k])
for o in range(len(highest)):
    states_names.append(highest.state)


states=np.unique(states_names)
```

In [38]:

```
highest.drop_duplicates(subset='state',keep="last",inplace=True)
```

In [58]:

```
new_frame=[]
new_frame1=[]

for i in range(len(states)):
    new_frame.append(states[i])
    new_frame.append(discover_highest['total_regs'].iloc[i])
    new_frame.append(highest[highest.state==states[i]].population.iloc[0])
    new_frame.append(highest[highest.state==states[i]].churn.iloc[0])
    new_frame1.append(list(new_frame))
    new_frame=[]
answer_highest = pd.DataFrame(new_frame1, columns = ['state', 'total regs',
'current population','population churn rate'])
answer_highest.head()
```

Out[58]:

	state	total regs	current population	population churn rate
0	Alabama	3071977	4,863,300	1.70%
1	Alaska	494183	741,894	4.50%
2	Arizona	1834250	6,931,071	8.40%
3	Arkansas	1435670	2,988,248	2.50%
4	California	7679605	39,250,017	5.40%

In [93]:

```
answer_highest['regs/person']=""
for i in range(len(answer_highest)):
    a=answer_highest['total regs'].iloc[i]
    b=answer_highest['current population'].iloc[i]
    b=int(b.replace(",",""))
    answer_highest['regs/person'].iloc[i]=a/b

answer_highest.head()
```

Out[93]:

	state	total regs	current population	population churn rate	regs/person
0	Alabama	3071977	4,863,300	1.70%	0.631665
1	Alaska	494183	741,894	4.50%	0.66611
2	Arizona	1834250	6,931,071	8.40%	0.264642
3	Arkansas	1435670	2,988,248	2.50%	0.480439
4	California	7679605	39,250,017	5.40%	0.195659

TOP 5 states with highest registration:

In [101]:

```
answer_highest.sort_values(['regs/person'],ascending=[False]).head(5)
```

Out[101]:

	state	total regs	current population	population churn rate	regs/person
16	Kentucky	14847038	4,436,974	2.20%	3.346208
43	Utah	2668120	3,051,217	10.40%	0.874445
47	West Virginia	1309419	1,831,102	-1.20%	0.715099
25	Montana	742617	1,042,520	5.40%	0.712329
1	Alaska	494183	741,894	4.50%	0.66611