Project: FBI NICS Gun Data Analysis

Table of Contents

- Introduction
- Data Wrangling
- Exploratory Data Analysis
- Conclusions

Introduction

> This project investigates and analyzs datasets associated with the number of firearms and explosives eligibility checks by month, state and type. The focuse will be on analyzing which states have the highest growth in gun registrations and for which type of gun. Further analysis will look into the overall trend of gun purchases over the years.

In [557…

```
#Importing my libraries into jupyter notebook

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

Data Wrangling

> There are two datasets provided for this project but I will import only one of them- the gun dataset into pandas dataframe in these jupyter notebook as that is the only dataset useful for my analysis.

General Properties

In [992…

```
# Loading my datasets into jupyter notebook
df_gun = pd.read_csv('C:/Users/user/Desktop/ProjectDatasets/gun_data.csv')
```

In [993…

```
#Assessing the structure of the gun data
df_gun.head()
```

Out[993…

	month	state	permit	permit_recheck	handgun	long_gun	other	multiple	admin	prepawn_h

	month	state	permit	permit_recheck	handgun	long_gun	other	multiple	admin	prepawn_h
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 [994...

```
df_gun.tail()
```

Out[994...

	month	state	permit	permit_recheck	handgun	long_gun	other	multiple	admin	prepa
12480	1998-11	Virginia	0.0	NaN	14.0	2.0	NaN	8	0.0	
12481	1998-11	Washington	1.0	NaN	65.0	286.0	NaN	8	1.0	
12482	1998-11	West Virginia	3.0	NaN	149.0	251.0	NaN	5	0.0	
12483	1998-11	Wisconsin	0.0	NaN	25.0	214.0	NaN	2	0.0	
12484	1998-11	Wyoming	8.0	NaN	45.0	49.0	NaN	5	0.0	

5 rows × 27 columns

In [598...

```
gun_df.info()
```

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

In [658…

```
#Assesing the datasets
df_gun.describe()
```

Out[658…

	permit	permit_recheck	handgun	long_gun	other	multiple	
count	12461.000000	1100.000000	12465.000000	12466.000000	5500.000000	12485.000000	12462.
mean	6413.629404	1165.956364	5940.881107	7810.847585	360.471636	268.603364	58.
std	23752.338269	9224.200609	8618.584060	9309.846140	1349.478273	783.185073	604.
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.
25%	0.000000	0.000000	865.000000	2078.250000	17.000000	15.000000	0.
50%	518.000000	0.000000	3059.000000	5122.000000	121.000000	125.000000	0.
75%	4272.000000	0.000000	7280.000000	10380.750000	354.000000	301.000000	0.
max	522188.000000	116681.000000	107224.000000	108058.000000	77929.000000	38907.000000	28083.

8 rows × 25 columns

In [513…

```
df_gun.shape
```

Out[513…

```
(12485, 27)
```

In [514…

```
df_gun.dtypes
```

Out[514…

```
month                         object
state                         object
permit                       float64
permit_recheck               float64
handgun                      float64
long_gun                     float64
```

```
other                          float64
multiple                         int64
admin                          float64
prepawn_handgun                float64
prepawn_long_gun               float64
prepawn_other                  float64
redemption_handgun             float64
redemption_long_gun            float64
redemption_other               float64
returned_handgun               float64
returned_long_gun              float64
returned_other                 float64
rentals_handgun                float64
rentals_long_gun               float64
private_sale_handgun           float64
private_sale_long_gun          float64
private_sale_other             float64
return_to_seller_handgun       float64
return_to_seller_long_gun      float64
return_to_seller_other         float64
totals                           int64
dtype: object
```

In [608…

```
#Checking for duplicated values in the column labelled 'state'
gun_df.state.duplicated().sum()
```

Out[608…

```
12430
```

In [703…

```
gun_df['state'].unique()
```

Out[703…

```
array(['Alabama', 'Alaska', 'Arizona', 'Arkansas', 'California',
       'Colorado', 'Connecticut', 'Delaware', 'District of Columbia',
       'Florida', 'Georgia', 'Guam', 'Hawaii', 'Idaho', 'Illinois',
       'Indiana', 'Iowa', 'Kansas', 'Kentucky', 'Louisiana', 'Maine',
       'Mariana Islands', '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', 'Puerto Rico', 'Rhode Island', 'South Carolina',
       'South Dakota', 'Tennessee', 'Texas', 'Utah', 'Vermont',
       'Virgin Islands', 'Virginia', 'Washington', 'West Virginia',
       'Wisconsin', 'Wyoming'], dtype=object)
```

In [614…

```
#Checking the number of unique values in the 'State' column
gun_df.state.nunique()
```

Out[614…

```
55
```

In [613…

```
#Checking for unique values in the 'Permit' column
gun_df.permit.nunique()
```

Out[613…

```
5390
```

In [682…

```
#Checking for duplicated values in the permit column
```

```
gun_df.permit.duplicated().sum()
```

Out[682…

```
7094
```

About the structure of the Gun Dataset

- Missing Values- The gun data has lots of feature columns with null values in their rows. These columns will be treated during the data cleaning stage.
- Incorrect datatype - Majority of the features are of float datatype and will be converted to type 'int' since gun permit checks and purchases count should not be floating point numbers. The 'month' column has the datatype 'object' which will be changed to datetime datatype.
- Duplicate Values- The entries in the state and other data columns which are duplicated will be ignored in our analysis since the data is a collection of firearm permit checks and purchases over a given period of time.
- Inconsist datatypes - Columns with inconsistent datatypes will be removed

Data Cleaning

In [995…

```
#Changing the datatype for the feature column 'Month'
df_gun['month'] = pd.to_datetime(df_gun['month'])
df_gun.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12485 entries, 0 to 12484
Data columns (total 27 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   month                      12485 non-null  datetime64[ns]
 1   state                      12485 non-null  object
 2   permit                     12461 non-null  float64
 3   permit_recheck             1100 non-null   float64
 4   handgun                    12465 non-null  float64
 5   long_gun                   12466 non-null  float64
 6   other                      5500 non-null   float64
 7   multiple                   12485 non-null  int64
 8   admin                      12462 non-null  float64
 9   prepawn_handgun            10542 non-null  float64
 10  prepawn_long_gun           10540 non-null  float64
 11  prepawn_other              5115 non-null   float64
 12  redemption_handgun         10545 non-null  float64
 13  redemption_long_gun        10544 non-null  float64
 14  redemption_other           5115 non-null   float64
 15  returned_handgun           2200 non-null   float64
 16  returned_long_gun          2145 non-null   float64
 17  returned_other             1815 non-null   float64
 18  rentals_handgun            990 non-null    float64
 19  rentals_long_gun           825 non-null    float64
 20  private_sale_handgun       2750 non-null   float64
 21  private_sale_long_gun      2750 non-null   float64
 22  private_sale_other         2750 non-null   float64
 23  return_to_seller_handgun   2475 non-null   float64
 24  return_to_seller_long_gun  2750 non-null   float64
 25  return_to_seller_other     2255 non-null   float64
 26  totals                     12485 non-null  int64
dtypes: datetime64[ns](1), float64(23), int64(2), object(1)
memory usage: 2.6+ MB
```

In [996…

```
#Rounding up the variables containing the various gun types since these variables will
df_gun['permit'] = round(df_gun['permit'])
df_gun['permit_recheck'] = round(df_gun['permit_recheck'])
df_gun['handgun'] = round(df_gun['handgun'])
df_gun['long_gun'] = round(df_gun['long_gun'])
df_gun['other'] = round(df_gun['other'])
```

In [997…

```
#Calculating mean of values to fill-up columns rows with null values
permit_mean = df_gun['permit'].mean()
permit_recheck_mean= df_gun['permit_recheck'].mean()
handgun_mean= df_gun['handgun'].mean()
long_gun_mean = df_gun['long_gun'].mean()
prepawn_long_gun_mean = df_gun['prepawn_long_gun'].mean()
multiple_mean = df_gun['multiple'].mean()
admin_mean = df_gun['admin'].mean()
prepawn_handgun_mean= df_gun['prepawn_handgun'].mean()
prepawn_long_run_mean = df_gun['prepawn_long_gun'].mean()
other_mean = df_gun['other'].mean()
prepawn_other_mean = df_gun['prepawn_other'].mean()
redemption_handgun_mean = df_gun['redemption_handgun'].mean()
redemption_long_gun_mean = df_gun['redemption_long_gun'].mean()
redemption_other_mean = df_gun['redemption_other'].mean()
returned_handgun_mean = df_gun['returned_handgun'].mean()
returned_long_gun_mean = df_gun['returned_long_gun'].mean()
returned_other_mean = df_gun['returned_other'].mean()
returned_handgun_mean = df_gun['returned_handgun'].mean()
returned_long_gun_mean = df_gun['returned_long_gun'].mean()
returned_other_mean = df_gun['returned_other'].mean()
rentals_handgun_mean = df_gun['rentals_handgun'].mean()
rentals_long_gun_mean = df_gun['rentals_long_gun'].mean()
private_sale_handgun_mean = df_gun['private_sale_handgun'].mean()
private_sale_long_gun_mean = df_gun['private_sale_long_gun'].mean()
private_sale_other_mean = df_gun['private_sale_other'].mean()
return_to_seller_handgun_mean = df_gun['return_to_seller_handgun'].mean()
return_to_seller_long_gun_mean = df_gun['return_to_seller_long_gun'].mean()
return_to_seller_other_mean = df_gun['return_to_seller_other'].mean()
```

In [998…

```
#Filling the missing values for each series with mean values.
df_gun['permit'].fillna(permit_mean, inplace=True)
df_gun['permit_recheck'].fillna(permit_recheck_mean, inplace=True)
df_gun['handgun'].fillna(handgun_mean, inplace=True)
df_gun['long_gun'].fillna(long_gun_mean, inplace=True)
df_gun['prepawn_long_gun'].fillna(prepawn_long_gun_mean, inplace=True)
df_gun['multiple'].fillna(multiple_mean, inplace=True)
df_gun['admin'].fillna(admin_mean, inplace=True)
df_gun['prepawn_handgun'].fillna(prepawn_handgun_mean, inplace=True)
df_gun['other'].fillna(other_mean, inplace=True)
df_gun['prepawn_other'].fillna(prepawn_other_mean, inplace=True)
df_gun['redemption_handgun'].fillna(redemption_handgun_mean, inplace=True)
```

```
df_gun['redemption_long_gun'].fillna(redemption_long_gun_mean, inplace=True)
df_gun['redemption_other'].fillna(redemption_other_mean, inplace=True)
df_gun['returned_handgun'].fillna(returned_handgun_mean, inplace=True)
df_gun['returned_long_gun'].fillna(returned_long_gun_mean, inplace=True)
df_gun['returned_other'].fillna(returned_other_mean, inplace=True)
df_gun['returned_handgun'].fillna(returned_handgun_mean, inplace=True)
df_gun['returned_long_gun'].fillna(returned_long_gun_mean, inplace=True)
df_gun['returned_other'].fillna(returned_other_mean, inplace=True)
df_gun['rentals_handgun'].fillna(rentals_handgun_mean, inplace=True)
df_gun['rentals_long_gun'].fillna(rentals_long_gun_mean, inplace=True)
df_gun['private_sale_handgun'].fillna(private_sale_handgun_mean, inplace=True)
df_gun['private_sale_long_gun'].fillna(private_sale_long_gun_mean, inplace=True)
df_gun['private_sale_other'].fillna(private_sale_other_mean, inplace=True)
df_gun['return_to_seller_handgun'].fillna(return_to_seller_handgun_mean, inplace=True)
df_gun['return_to_seller_long_gun'].fillna(return_to_seller_long_gun_mean, inplace=True
df_gun['return_to_seller_other'].fillna(return_to_seller_other_mean, inplace=True)
```

In [999…

```
#Converting the rounded-up variables above to 'int' datatypes
df_gun['permit'] = df_gun['permit'].astype(int)
df_gun['permit_recheck'] = df_gun['permit_recheck'].astype(int)
df_gun['handgun'] = df_gun['handgun'].astype(int)
df_gun['long_gun'] = df_gun['long_gun'].astype(int)
df_gun['other'] = df_gun['other'].astype(int)
```

In [676…

```
df_gun.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12485 entries, 0 to 12484
Data columns (total 27 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   month                      12485 non-null  datetime64[ns]
 1   state                      12485 non-null  object
 2   permit                     12485 non-null  int32
 3   permit_recheck             12485 non-null  int32
 4   handgun                    12485 non-null  int32
 5   long_gun                   12485 non-null  int32
 6   other                      12485 non-null  int32
 7   multiple                   12485 non-null  int64
 8   admin                      12485 non-null  float64
 9   prepawn_handgun            12485 non-null  float64
 10  prepawn_long_gun           12485 non-null  float64
 11  prepawn_other              12485 non-null  float64
 12  redemption_handgun         12485 non-null  float64
 13  redemption_long_gun        12485 non-null  float64
 14  redemption_other           12485 non-null  float64
 15  returned_handgun           12485 non-null  float64
 16  returned_long_gun          12485 non-null  float64
 17  returned_other             12485 non-null  float64
 18  rentals_handgun            12485 non-null  float64
 19  rentals_long_gun           12485 non-null  float64
 20  private_sale_handgun       12485 non-null  float64
 21  private_sale_long_gun      12485 non-null  float64
 22  private_sale_other         12485 non-null  float64
 23  return_to_seller_handgun   12485 non-null  float64
 24  return_to_seller_long_gun  12485 non-null  float64
 25  return_to_seller_other     12485 non-null  float64
 26  totals                     12485 non-null  int64
dtypes: datetime64[ns](1), float64(18), int32(5), int64(2), object(1)
memory usage: 2.3+ MB
```

In [100…

```
df_gun.head()
```

Out[100…

	month	state	permit	permit_recheck	handgun	long_gun	other	multiple	admin	prepawn_ha
0	2017-09-01	Alabama	16717	0	5734	6320	221	317	0.0	
1	2017-09-01	Alaska	209	2	2320	2930	219	160	0.0	
2	2017-09-01	Arizona	5069	382	11063	7946	920	631	0.0	
3	2017-09-01	Arkansas	2935	632	4347	6063	165	366	51.0	
4	2017-09-01	California	57839	0	37165	24581	2984	0	0.0	

5 rows × 27 columns

In [681…

```
df_gun.duplicated().sum()
```

Out[681…

```
0
```

In [100…

```
#Since we will be analyzing the trend of gun purchases over the years, the date column
df_gun['year'] = pd.DatetimeIndex(df_gun['month']).year
df_gun['month'] =  pd.DatetimeIndex(df_gun['month']).month
```

In [715…

```
df_gun.head()
```

Out[715…

	month	state	permit	permit_recheck	handgun	long_gun	other	multiple	admin	prepawn_ha
0	9	Alabama	16717	0	5734	6320	221	317	0.0	
1	9	Alaska	209	2	2320	2930	219	160	0.0	
2	9	Arizona	5069	382	11063	7946	920	631	0.0	
3	9	Arkansas	2935	632	4347	6063	165	366	51.0	
4	9	California	57839	0	37165	24581	2984	0	0.0	

5 rows × 28 columns

In [100…

```
#Shifting the year column to the first index position
cols = list(df_gun.columns.values)
```

```
cols.pop(cols.index('year'))
df_gun = df_gun[['year']+cols]
```

In [847…

```
df_gun.head()
```

Out[847…

	year	month	state	permit	permit_recheck	handgun	long_gun	other	multiple	admin	...	re
0	2017	9	Alabama	16717	0	5734	6320	221	317	0.0	...	
1	2017	9	Alaska	209	2	2320	2930	219	160	0.0	...	
2	2017	9	Arizona	5069	382	11063	7946	920	631	0.0	...	
3	2017	9	Arkansas	2935	632	4347	6063	165	366	51.0	...	
4	2017	9	California	57839	0	37165	24581	2984	0	0.0	...	

5 rows × 28 columns

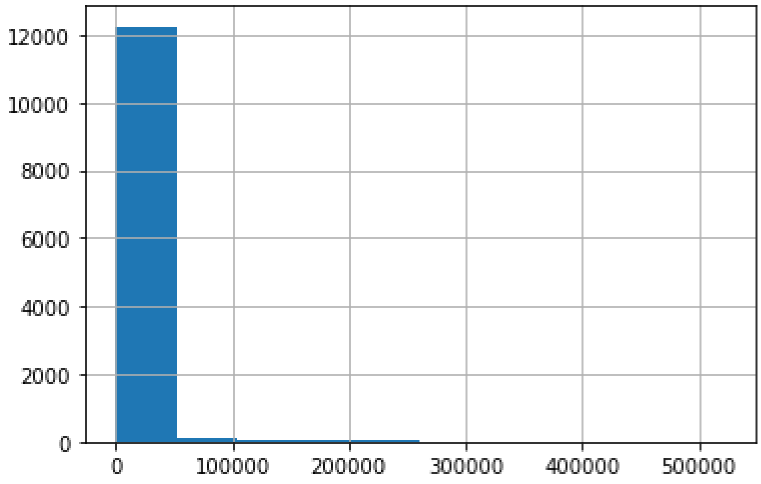
> Now that we have cleaned and assessed the dataset for analysis, lets move on to explore the variables we shall be working with.

EDA with Histogram

> Lets use histogram to explore the distribution of entries in our variables which are most relevant to our analysis

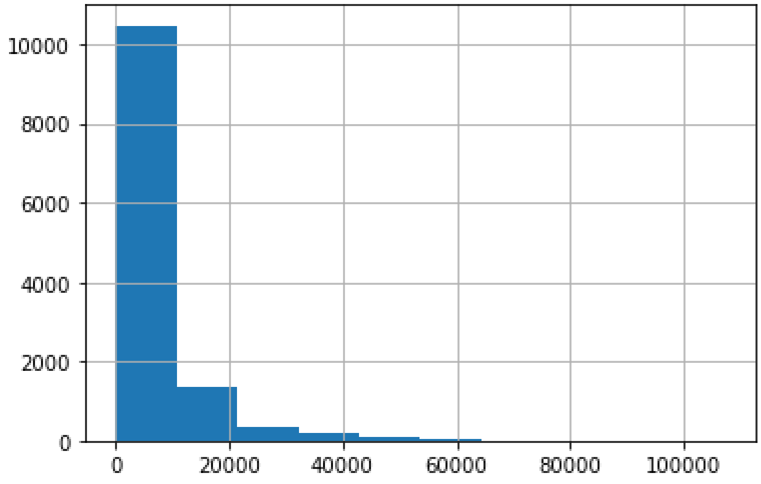
In [100…

```
df_gun.permit.hist();
```



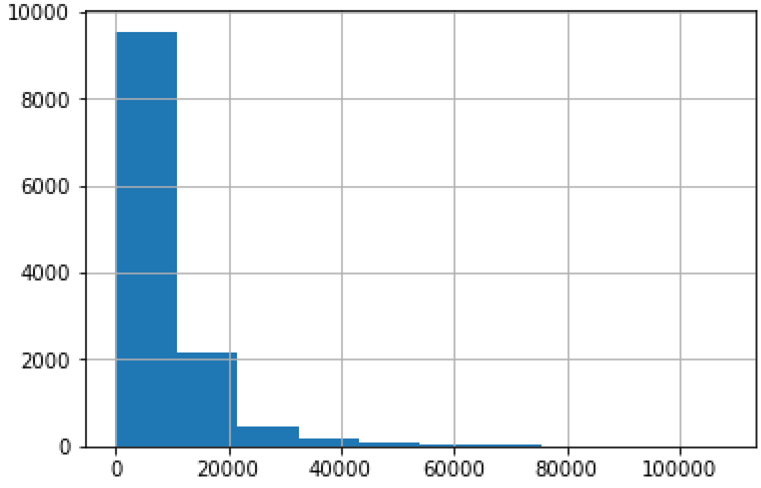
In [100…

```
df_gun.handgun.hist();
```



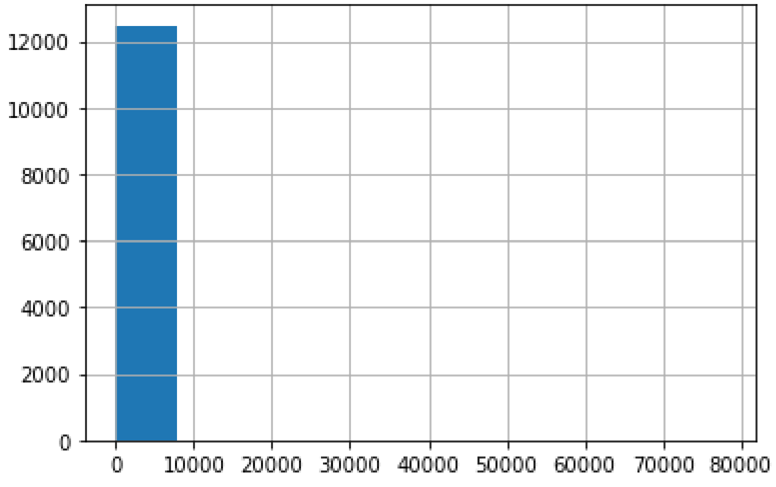
In [797…

```
df_gun.long_gun.hist();
```



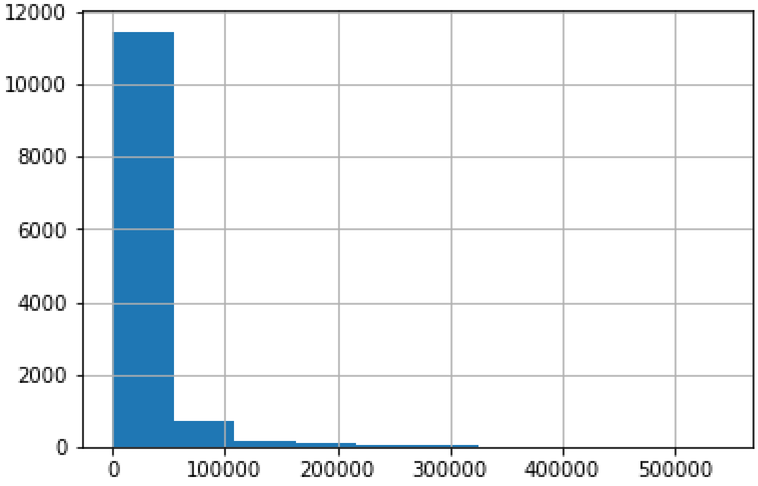
In [100…

```
df_gun.other.hist();
```



In [101…

```
df_gun.totals.hist();
```



Inight Drawn from EDA

> It can be clearly seen that our data is not normally distributed as there is significant skewness to the right indicating that the number of gun purchases are minimal compared to the overall number of both checks and purchases

Exploratory Data Analysis

Research Question 1: Which state has the highest growth rate in Gun registration?

In [100…

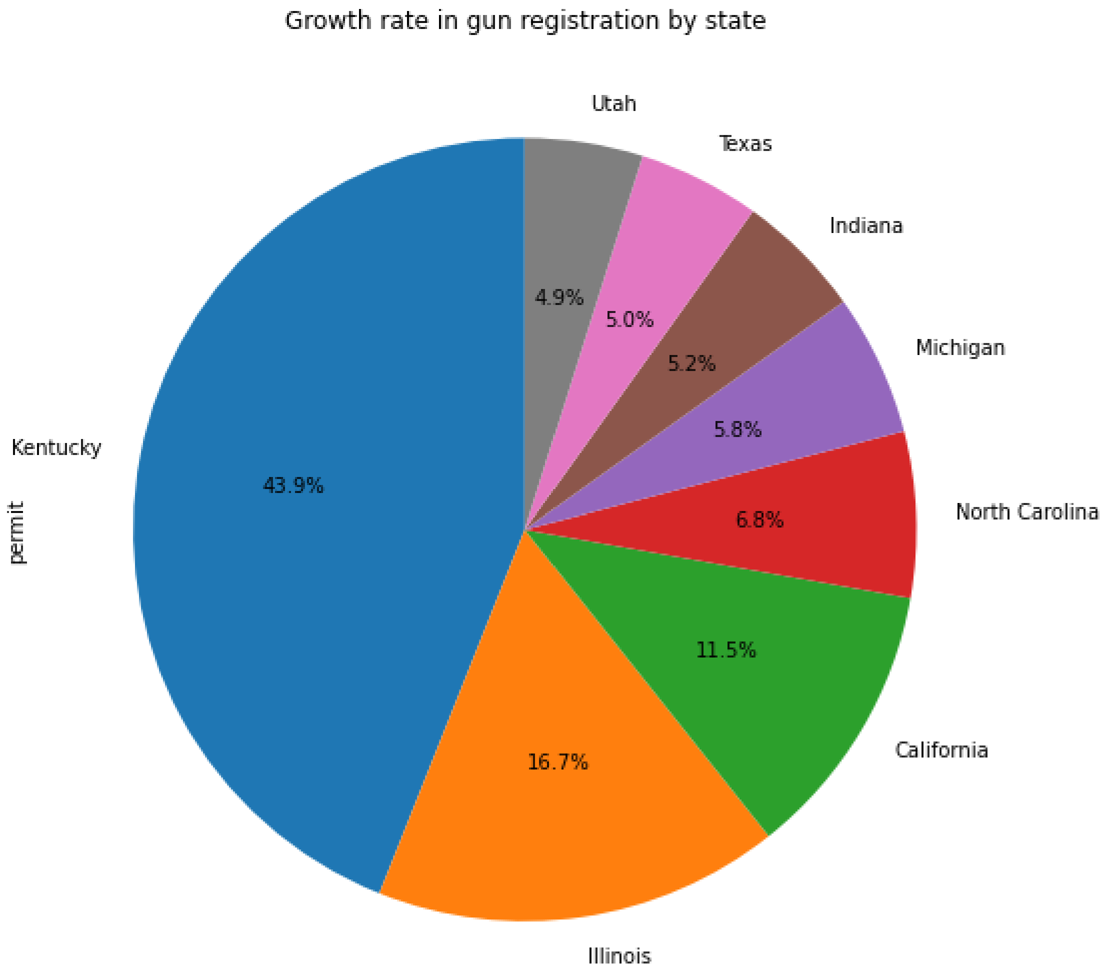
```
registration_means= df_gun.groupby('state')['permit'].mean().round().sort_values(ascend
registration_means
```

Out[100…

```
state
Kentucky          109810.0
Illinois           41844.0
California         28769.0
North Carolina     17062.0
Michigan           14595.0
Indiana            13065.0
Texas              12588.0
Utah               12130.0
Name: permit, dtype: float64
```

In [100…

```
registration_means.plot(kind='pie', autopct='%1.1f%%', startangle=90, figsize=(9,9))
plt.title('Growth rate in gun registration by state');
```



From this visualization, the state with the highest gun registration as defined by the permits given is Kentucky

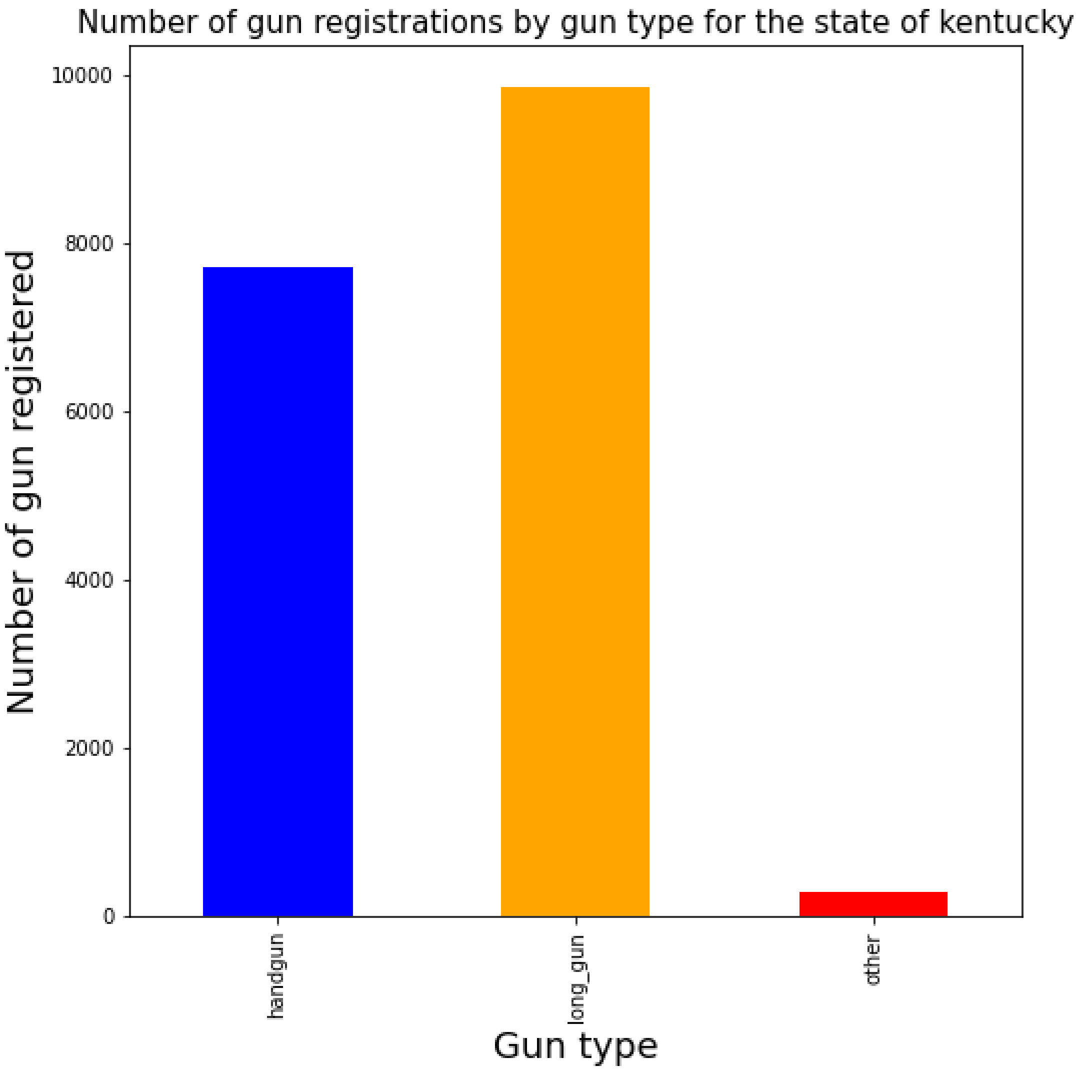
Research Question 2: For the state with the highest Gun registration, what gun type recorded the highest registration?

In [913…

```
#Using query to select (truncate) the data where state is kentucky
df_kentucky = df_gun.query("state == 'Kentucky'")
```

In [912…

```
#Plotting a bar chart to visualize the data variable
colors=['blue', 'orange', 'red']
df_kentucky.iloc[:, 5:8].mean().plot(kind='bar', color=colors, figsize=(8,8))
plt.title('Number of gun registrations by gun type for the state of kentucky', fontsize
plt.xlabel('Gun type', fontsize=18)
plt.ylabel('Number of gun registered', fontsize=18);
```



> This visualization informs us that long gun recorded the highest number of purchases for the state of Kentucky with the highest gun registration

Research Question 2: What month was the highest Gun registration recorded for the State with the highest purchase?.

> Let's visualize the month with the highest recorded permit for gun purchase. This data will be visualized using the column feature 'permit' with the dataset filtered to 'kentucky' state

In [107…

```
df_kentucky_month = df_kentucky.groupby('month', as_index=True)['permit'].mean().round(
df_kentucky_month
```

Out[107…

```
month
3     121943.0
1     117980.0
7     115131.0
12    113458.0
6     111755.0
5     111281.0
4     110944.0
```

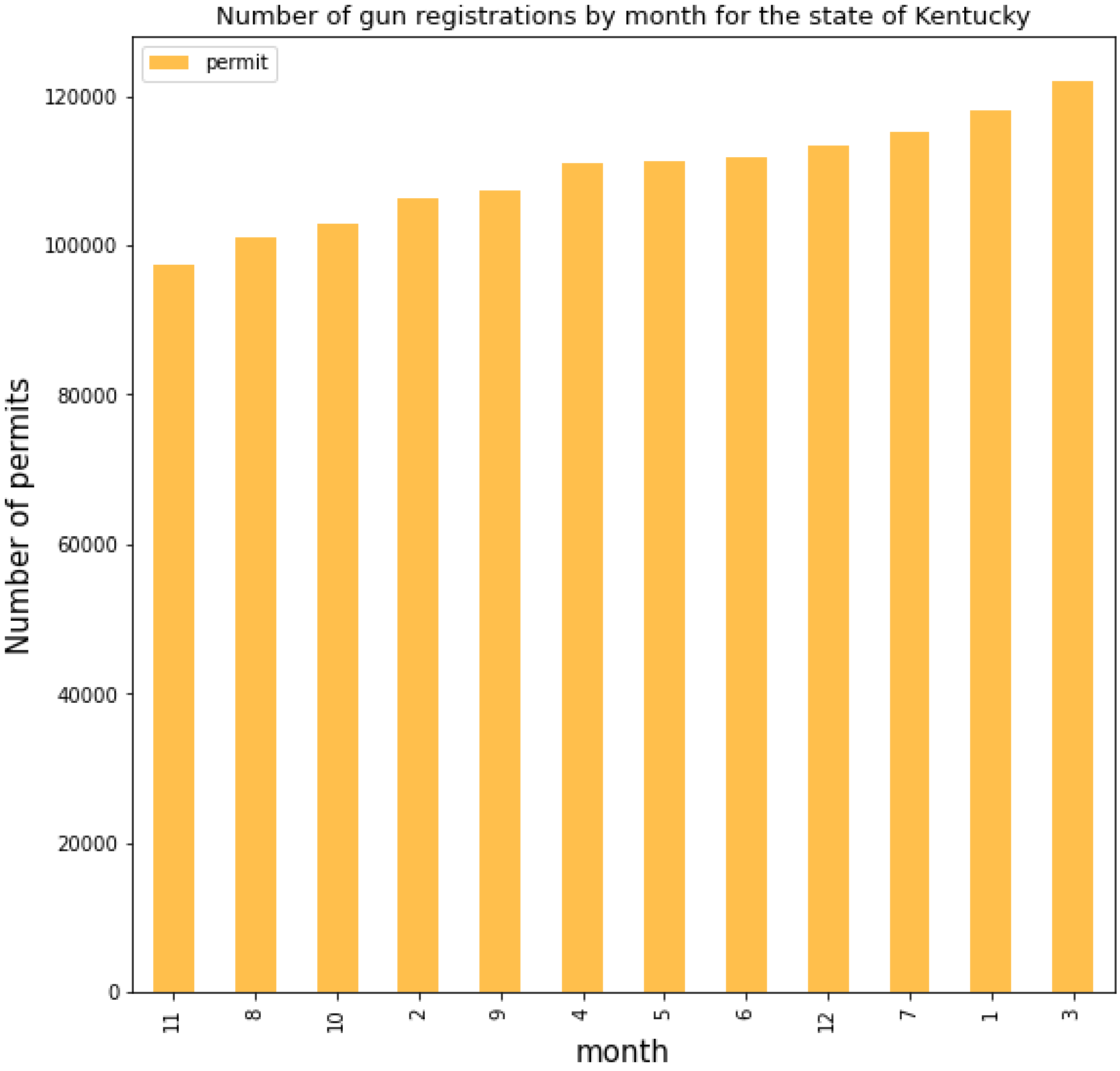
```
9      107437.0
2      106280.0
10     102849.0
8      100984.0
11      97306.0
Name: permit, dtype: float64
```

In [108…

```
#Using pandas groupby function to group the dataset by month to visualize the mean for
df_kentucky_month = df_kentucky.groupby('month', as_index=True)['permit'].mean().round(
```

In [110…

```
#Using the bar plot to visualize the figures calculated
df_kentucky_month.plot(kind='bar', color='orange',tick_label=labels, alpha=.7, figsize=
plt.legend(loc='upper left')
plt.title('Number of gun registrations by month for the state of Kentucky', fontsize=13
plt.ylabel('Number of permits', fontsize=15)
plt.xlabel('month', fontsize=15);
```



> This visualization shows the month of march as having the highest number of gun permit checks and purchases

Research Question 2: What is the overall trend of Gun purchases?

In [104…

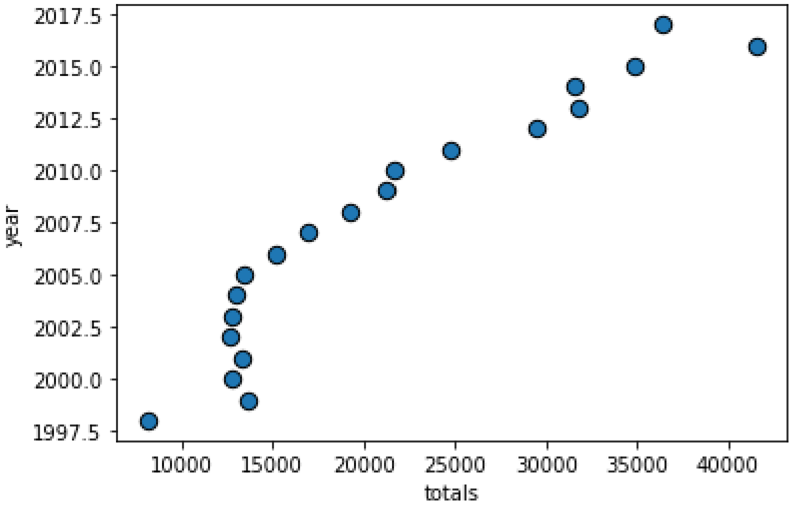
```
#Using the groupby function to extract data for years and totals
df_totals = df_gun.groupby('year', as_index=False)['totals'].mean().round().astype(int)
df_totals
```

Out[104…

	year	totals
0	1998	8108
1	1999	13703
2	2000	12768
3	2001	13364
4	2002	12677
5	2003	12731
6	2004	13000
7	2005	13435
8	2006	15141
9	2007	16913
10	2008	19232
11	2009	21189
12	2010	21698
13	2011	24753
14	2012	29491
15	2013	31768
16	2014	31606
17	2015	34888
18	2016	41554
19	2017	36345

In [111…

```
df_totals.plot(x='totals', y='year', edgecolors="k", s=70, kind='scatter');
```



> From the visualization above, it can deduced that there is an increasing trend of gun eligibility checks and purchases over the years with year-2017 having the peak value, before a slight decline surfaced. It is possible that the preceeding year-2018 might record a decline in gun purchases

Conclusions

> - In conclusion, the city of kentucky has the highest growth in gun purchases for all time. The FBI might have to look at reviewing the eligible to understand the surge in the number of permits given.
> - There is a high demand of long gun compared to other types of gun. This could be a good business insight for Gun shops as they might consider stocking up more of long guns than other types of gun.
> - Permit checks seems to occur the more during the spring than other seasons of the year.
> - The general trend shows that gun purchases is increasing over time

Limitations

> - Two datasets were provided for this project, the census dataset and gun dataset. I decided to limit my analysis to data obtained from the gun dataset.

List of External Resources

> - https://www.geeksforgeeks.org/
> - https://stackoverflow.com/
> - https://www.askpython.com/
> - https://pandas.pydata.org/