Investigate a Dataset Project

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.

1. What is the most popular gun type?

2. Which state has had the highest growth in gun registrations?

3. What is the overall trend of gun purchases?

In [4]:

```
import datetime
```

In [5]:

```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns

import ast

%matplotlib inline
```

Data Wrangling

read gun_data.csv into pandas dataframe

```
In [6]: dfgun = pd.read_csv('gun_data.csv')
        dfgun.head()
```

Out[6]:

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

5 rows × 27 columns

dfcensus = pd.read_csv('U.S. Census Data.csv') dfcensus.head()

In [7]:
```
dfcensus = pd.read_csv('u.s.-census-data.csv')
dfcensus.head()
```

Out[7]:

	Fact	Fact Note	Alabama	Alaska	Arizona	Arkansas	California	Colorado	Connecticut
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	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
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 rows × 52 columns

Display the shape of the dataframe. It shows 12485 rows and 27 columns

In [8]:
```
dfgun.shape
```

Out[8]:
```
(12485, 27)
```

In [9]:
```
dfcensus.shape
```

Out[9]:
```
(85, 52)
```

Confirm that there are no duplicated rows in either dataset

In [10]:
```
dfgun.duplicated().sum()
```

Out[10]:
```
0
```

In [11]:
```
dfcensus.duplicated().sum()
```

Out[11]:
```
3
```

Remove the 3 duplicated rows found in the Census Data

In [12]:
```
dfcensus.drop_duplicates(inplace = True)
```

In [13]:
```
dfcensus.duplicated().sum()
```

Out[13]:
```
0
```

Confirm duplicates have been removed

Exploring both datasets further

In [14]:
```
dfgun.describe()
```

Out[14]:

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

8 rows × 25 columns

In [15]:
```
dfgun.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
```

Converting to correct data types

In [16]:
```
dfgun['month'] = pd.to_datetime(dfgun['month'])
```

In [17]:
```
dfgun['multiple'] = pd.to_numeric(dfgun['multiple']).astype(float)
dfgun.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  float64       
 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(24), int64(1), object(1)
memory usage: 2.6+ MB
```

Getting rid of unnecessary columns in gun dataset

In [18]:

```
column_name = dfgun.columns[15:26]

dfgun = dfgun.drop(columns=column_name)
dfgun.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12485 entries, 0 to 12484
Data columns (total 16 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  float64
 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  totals               12485 non-null  int64
dtypes: datetime64[ns](1), float64(13), int64(1), object(1)
memory usage: 1.5+ MB
```

The function below takes a list of columns to drop and a dataframe as the agruments and drops the specified colums.

In [21]:

```
def drop(col_list,dfgun):
    for i in col_list:
        dfgun.drop(dfgun.columns[dfgun.columns.str.contains('^'+i)], axis =1, 
inplace = True)
```

In [22]:

```
drop(['admin','prepawn_handgun','prepawn_long_gun','prepawn_other','redemption
_handgun','redemption_long_gun','redemption_other'],dfgun)
dfgun.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12485 entries, 0 to 12484
Data columns (total 9 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  float64       
 8   totals          12485 non-null  int64         
dtypes: datetime64[ns](1), float64(6), int64(1), object(1)
memory usage: 878.0+ KB
```

In [23]:

```
dfcensus.describe()
```

Out[23]:

	Fact	Fact Note	Alabama	Alaska	Arizona	Arkansas	California	Colorado	Connectic
count	80	28	65	65	65	65	65	65	
unique	80	15	65	64	64	64	63	64	
top	Nonveteran-owned firms, 2012	(c)	5.20%	7.30%	50.30%	50.90%	6.80%	3.30%	0.1
freq	1	6	1	2	2	2	2	2	

4 rows × 52 columns

Confirm that all states are present in census and gun datasets

In [24]:

```
Census_index = dfcensus.iloc[0].index
Census_index
```

Out[24]:

```
Index(['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')
```

```
In [25]: Gun_index = dfgun.groupby('state').sum().index
         Gun_index
```

```
Out[25]: Index(['Alabama', 'Alaska', 'Arizona', 'Arkansas', 'California', 'Colorado',
                'Connecticut', 'Delaware', 'District of Columbia', 'Florida', 'Georgi
         a',
                '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 Virgini
         a',
                'Wisconsin', 'Wyoming'],
               dtype='object', name='state')
```

```
In [26]: len(Census_index[2:])
```

```
Out[26]: 50
```

```
In [27]: len(Gun_index[0:])
```

```
Out[27]: 55
```

The gun index appears to be longer than the census index.

I'll use a for loop to find the items not present in the census index

```
In [28]: for s in Gun_index:
             if s not in Census_index:
                 print(s)
```

```
District of Columbia
Guam
Mariana Islands
Puerto Rico
Virgin Islands
```

```
In [29]: dfgun = dfgun[dfgun.state != 'District of Columbia']
         dfgun = dfgun[dfgun.state != 'Virgin Islands']
         dfgun = dfgun[dfgun.state != 'Guam']
         dfgun = dfgun[dfgun.state != 'Puerto Rico']
         dfgun = dfgun[dfgun.state != 'Mariana Islands']
```

After dropping DC and the territories, they no longer appear in the gun index.

In [30]:

```
Gun_index = dfgun.groupby('state').sum().index
len(Gun_index[0:])
```

Out[30]:

```
50
```

In [31]:

```
dfcensus.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 82 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
```

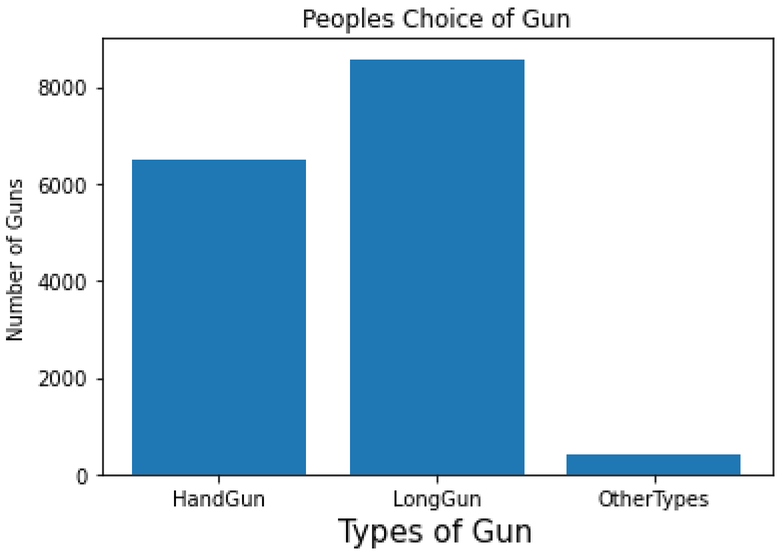
```
dtypes: object(52)
memory usage: 34.0+ KB
```

Dropping the Fact Note column

In [32]:
```
dfcensus = dfcensus.drop(['Fact Note'], axis=1)
```

Research Question #1: What is the most popular gun type?

In [33]:
```
hand = dfgun['handgun'].mean()
long = dfgun['long_gun'].mean()
other = dfgun['other'].mean()
plt.bar([1,2,3], [hand, long, other],tick_label=['HandGun','LongGun','OtherTyp
es'])
plt.figsize=(20,10)
plt.title('Peoples Choice of Gun', fontsize=12)
plt.xlabel('Types of Gun', fontsize=15)
plt.ylabel('Number of Guns', fontsize=10);
```



Answer: Long Guns are the most popular type of gun

Research Question #2: Which state has had the highest growth in gun registrations?

In [34]:
```
total_bystate = dfgun.groupby('state')
```

In [35]:
```
state_sum = total_bystate.sum()
```

In [36]:
```
state_total = state_sum['totals']
```

In [37]:
```
state_total.head()
```

Out[37]:
```
state
Alabama        6706079
Alaska         1137643
Arizona        4425714
Arkansas       3752633
California    19014063
Name: totals, dtype: int64
```

In [38]:
```
state_highgrowth = dfgun.groupby(['month', 'state'])['totals'].sum()
```

In [39]:
```
dfgun = dfgun.sort_values(['totals'], ascending=False)
```

In [40]:
```
max_date = dfgun['month'].max()
min_date = dfgun['month'].min()
```

In [41]:
```
state_highgrowth_total = state_highgrowth.loc[max_date] - state_highgrowth.loc
[min_date]
state_highgrowth_total.idxmax()
```

Out[41]:
```
'Kentucky'
```

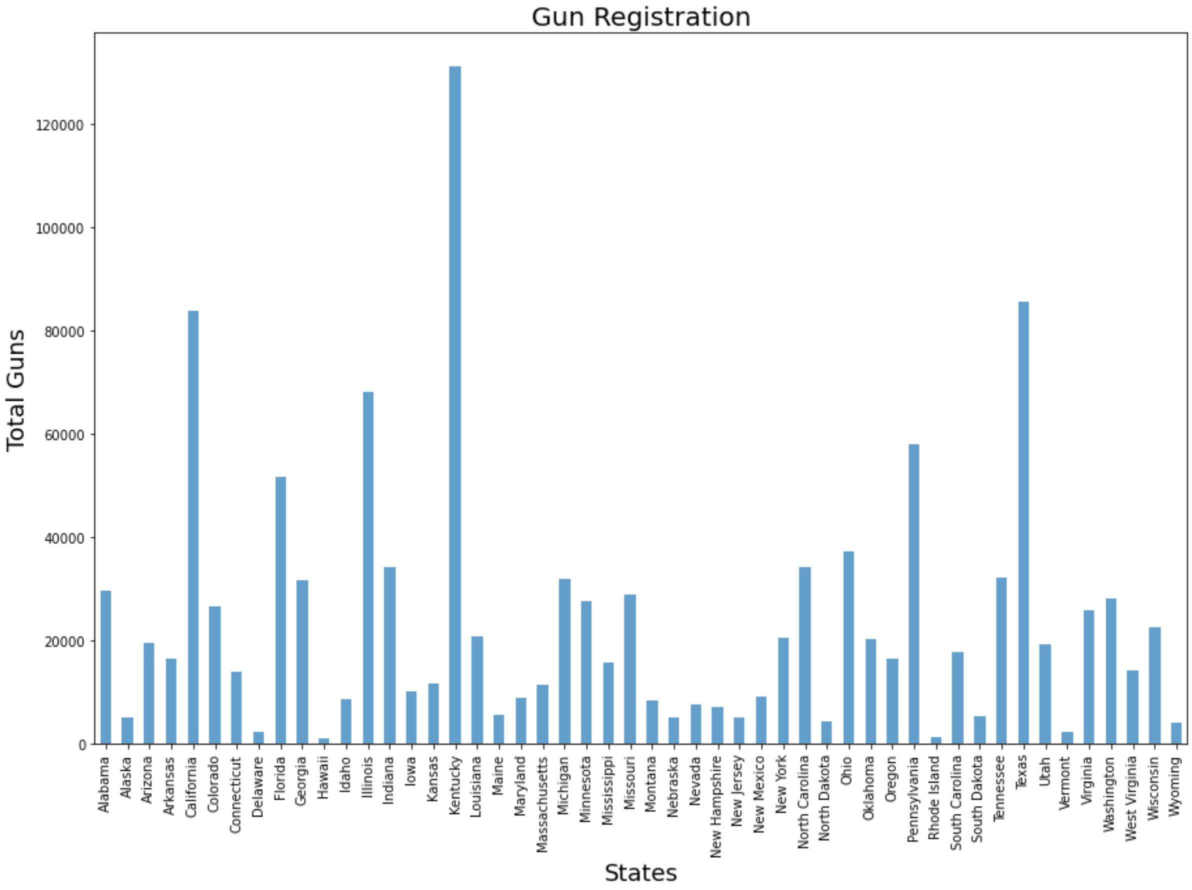
Total guns in Kentucky

In [42]:
```
state_highgrowth_total.loc['Kentucky']
```

Out[42]:
```
397866
```

In [43]:

```
dfgun.groupby('state')['totals'].mean().plot(kind='bar', figsize=(15,10), alph
a=.7)
plt.xlabel('States', fontsize=18)
plt.ylabel('Total Guns', fontsize=18)
plt.title('Gun Registration',fontsize=20);
```



Answer: Kentucky is the state with the highest growth in gun registration

Research Question #2: What is the overall trend of gun purchases over time?

In [44]:

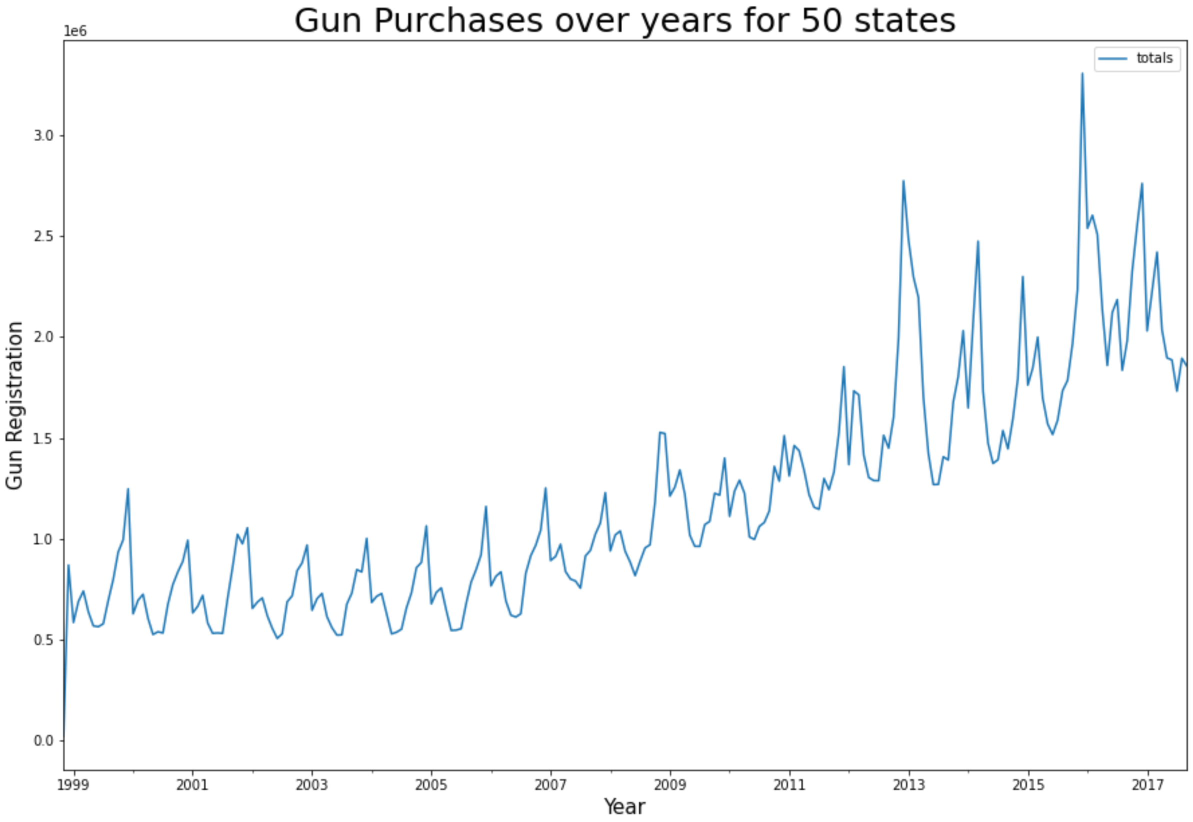
```
dfgun_totals = dfgun[['month','totals']]
dfgun_totals.set_index('month', inplace = True)

dfgun_totals = dfgun_totals[::-1]

gun_totals_groupby_month = dfgun_totals.groupby('month').sum()
```

In [46]:

```
ax = gun_totals_groupby_month.plot(kind='line',figsize=(15,10))
ax.set_title('Gun Purchases over years for 50 states', fontsize=25)
ax.set_xlabel('Year', fontsize=15)
ax.set_ylabel('Gun Registration', fontsize=15);
```



The graph above clearly shows an upward trend

Conclusion:

Limitations:

In [ ]:

```
In the census data there was no data for DC and US territories.

Data was seperated into two tables which affected the process of analysis. Add
itionally, the population data was only recorded for 2010 and 2016.

The dataset of gun data has many null values, which I felt I could not remove
since it would skew the data potentially causing the analysis to be incorrect.

Having the gun and census datasets in separate files and formats was a limitat
ion for me. I was not able to combine the dataset to answer potentially more i
nteresting questions.
```

In conclusion, I was able to answer each of the research questions posed. I would have guessed that hand guns would be more popular than long guns, but that is not the case. The data shows that long guns are far more popular than hand guns.

The State of Kentucky has the highest Gun Registrations of any state in the US. The cause could be due to more lenient laws regarding firearms in that state and this would be interesting to investigate further.

There is a definiite upward trend upward of gun purchases as shown in the last vizualization. There seems to be a strong pattern in the peaks and valleys from year to year and this would be interesting to investigat further as well.

In [ ]: