# roquelaure-project-NICS

October 12, 2022

# 1 Project: Investigation of the FBI NICS dataset

# 2 by Julien Roquelaure

## 2.1 Table of Contents



## Introduction

The main dataset 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.

In addition, we download a U.S. census dataset for comparison against the NICS data.

The datasets can be found here:

NICS

Census

Let us import the necessary libraries first:

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In this project, we want to answer the followong questions:

1. What are the country-wide trends about gun purchases?
2. How do states differ from one another in this respect?
3. By using the details of each state's demographics, can we find associations with gun purchases?

## Data Wrangling

### 2.1.1 General Properties

In this section, I will load 2 datasets: - the NICS data, which contains the number of firearm checks by month, state, and type. - the U.S. census data, which contains demographics data about each state

[2]:
```
nics = pd.read_excel('gun_data.xlsx')
census = pd.read_csv('U.S. Census Data.csv')
```

[3]:
```
nics.shape
```

[3]:
```
(12485, 27)
```

[4]:
```
nics.head()
```

[4]:
```
     month       state   permit  permit_recheck  handgun  long_gun   other  \
0  2017-09     Alabama  16717.0             0.0   5734.0    6320.0   221.0
1  2017-09      Alaska    209.0             2.0   2320.0    2930.0   219.0
2  2017-09     Arizona   5069.0           382.0  11063.0    7946.0   920.0
3  2017-09    Arkansas   2935.0           632.0   4347.0    6063.0   165.0
4  2017-09  California  57839.0             0.0  37165.0   24581.0  2984.0

   multiple  admin  prepawn_handgun  …  returned_other  rentals_handgun  \
0       317    0.0             15.0  …             0.0              0.0
1       160    0.0              5.0  …             0.0              0.0
2       631    0.0             13.0  …             0.0              0.0
3       366   51.0             12.0  …             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                   9.0                   16.0
1               0.0                  17.0                   24.0
2               0.0                  38.0                   12.0
3               0.0                  13.0                   23.0
4               0.0                   0.0                    0.0

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

   return_to_seller_other  totals
0                     3.0   32019
1                     0.0    6303
2                     0.0   28394
3                     1.0   17747
```

```
4                      0.0  123506

[5 rows x 27 columns]
```

[5]:
```
nics.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
```

[6]:
```
census.shape
```

[6]: (85, 52)

[7]:
```
census.head()
```

[7]:
```
                                                  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]
```

[8]:
```
census.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
```

### 2.1.2 Data cleaning for the Census dataset

Already we notice two things about the census dataset:

- the data is arranged with states in columns, and variables in rows, so we will transpose the table.
- there are several lines (and one column) of notes, that might be useful for reference but not for computation and visualization, so we get rid of them.

[9]:
```
census.drop(['Fact Note'], axis=1, inplace=True)
```

[10]:
```
census.rename(columns={'Fact': 'State'}, inplace=True)
```

[11]:
```
census = census.T
```

We use the first row as our column name and drop it afterwards:

[12]:
```
census.columns = census.iloc[0]
census.drop('State', inplace=True)
```

We drop all the columns with the different notes and comments:

[13]:
```
census.dropna(axis=1, inplace=True)
```

[14]:
```
census.head()
```

[14]:
```
State      Population estimates, July 1, 2016,  (V2016)  \
Alabama                                      4,863,300
Alaska                                         741,894
Arizona                                      6,931,071
Arkansas                                     2,988,248
California                                  39,250,017

State      Population estimates base, April 1, 2010,  (V2016)  \
Alabama                                            4,780,131
Alaska                                               710,249
Arizona                                            6,392,301
Arkansas                                           2,916,025
California                                        37,254,522

State      Population, percent change - April 1, 2010 (estimates base) to July
1, 2016,  (V2016)  \
Alabama                                                 1.70%
Alaska                                                  4.50%
Arizona                                                 8.40%
Arkansas                                                2.50%
California                                              5.40%
```

```
State      Population, Census, April 1, 2010  \
Alabama                        4,779,736
Alaska                           710,231
Arizona                        6,392,017
Arkansas                       2,915,918
California                    37,253,956

State      Persons under 5 years, percent, July 1, 2016,  (V2016)  \
Alabama                                                 6.00%
Alaska                                                  7.30%
Arizona                                                 6.30%
Arkansas                                                6.40%
California                                              6.30%

State      Persons under 5 years, percent, April 1, 2010  \
Alabama                                             6.40%
Alaska                                              7.60%
Arizona                                             7.10%
Arkansas                                            6.80%
California                                          6.80%

State      Persons under 18 years, percent, July 1, 2016,  (V2016)  \
Alabama                                                 22.60%
Alaska                                                  25.20%
Arizona                                                 23.50%
Arkansas                                                23.60%
California                                              23.20%

State      Persons under 18 years, percent, April 1, 2010  \
Alabama                                             23.70%
Alaska                                              26.40%
Arizona                                             25.50%
Arkansas                                            24.40%
California                                          25.00%

State      Persons 65 years and over, percent,  July 1, 2016,  (V2016)  \
Alabama                                                 16.10%
Alaska                                                  10.40%
Arizona                                                 16.90%
Arkansas                                                16.30%
California                                              13.60%

State      Persons 65 years and over, percent, April 1, 2010  …  \
Alabama                                                13.80%  …
Alaska                                                  7.70%  …
Arizona                                                13.80%  …
Arkansas                                               14.40%  …
```

```
California                                             11.40%  …

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

State      Minority-owned firms, 2012 Nonminority-owned firms, 2012  \
Alabama                        92,219                       272,651
Alaska                         13,688                        51,147
Arizona                       135,313                       344,981
Arkansas                       35,982                       189,029
California                  1,619,857                     1,819,107

State      Veteran-owned firms, 2012 Nonveteran-owned firms, 2012  \
Alabama                       41,943                      316,984
Alaska                         7,953                       56,091
Arizona                       46,780                      427,582
Arkansas                      25,915                      192,988
California                   252,377                    3,176,341

State      Population per square mile, 2010 Land area in square miles, 2010  \
Alabama                                94.4                       50,645.33
Alaska                                  1.2                      570,640.95
Arizona                                56.3                      113,594.08
Arkansas                                 56                       52,035.48
California                            239.1                      155,779.22

State      FIPS Code
Alabama         "01"
Alaska          "02"
Arizona         "04"
Arkansas        "05"
California      "06"

[5 rows x 65 columns]
```

Now, we have the demographics for the 50 states. But we have 65 features, which is a lot. Let's take a look at which ones we could use:

[15]:
```
census.columns
```

[15]:
```
Index(['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'],
   dtype='object', name='State')
```

The U.S. census occurs every 10 years. So we have 2010 measured populations and 2016 estimates.

For this project I will trust the estimates and keep only the 2016 values when possible.

For my analysis I will select the following variables (with their new names):

Population and density: - Population estimates, July 1, 2016, (V2016) as 'population' - Land area in square miles, 2010, as 'land_area'

Age: - Persons under 18 years, percent, July 1, 2016, (V2016) as 'under_18y_percent' - Persons 65 years and over, percent, July 1, 2016, (V2016) as 'over_65y_percent'

Gender: - Female persons, percent, July 1, 2016, (V2016) as 'female_percent'

Race: - Black or African American alone, percent, July 1, 2016, (V2016) as 'black_percent' - American Indian and Alaska Native alone, percent, July 1, 2016, (V2016) as 'native_american_percent' - Asian alone, percent, July 1, 2016, (V2016) as 'asian_percent' - Native Hawaiian and Other Pacific Islander alone, percent, July 1, 2016, (V2016) as 'native_pacific_percent' - Hispanic or Latino, percent, July 1, 2016, (V2016) as 'hispanic_percent' - White alone, not Hispanic or Latino, percent, July 1, 2016, (V2016) as 'white_percent'

Military status: - Veterans, 2011-2015 as 'veterans'

Foreign birth status: - Foreign born persons, percent, 2011-2015 as 'foreign_born_percent'

Education: - High school graduate or higher, percent of persons age 25 years+, 2011-2015 as 'hs_percent' - Bachelor's degree or higher, percent of persons age 25 years+, 2011-2015 as 'bachelor_percent'

Employment and income: - Total employment, 2015 as 'employment' - Per capita income in past 12 months (in 2015 dollars), 2011-2015 as 'per_capita_income'

I discarded several variables that seemed less relevant for individuals like sales, number of firms or land area.

In each category of features, I tried to keep only a handful of variables in order to not clutter the project.

[16]:
```
census = census[['Population estimates base, April 1, 2010,  (V2016)',
         'Land area in square miles, 2010',
         'Persons under 18 years, percent, July 1, 2016,  (V2016)',
         'Persons 65 years and over, percent,  July 1, 2016,  (V2016)',
         'Female persons, 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)',
         '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',
         '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",
         'Total employment, 2015',
         'Per capita income in past 12 months (in 2015 dollars), 2011-2015'
        ]]
```

[17]:
```
census = census.rename(columns = {
     'Population estimates base, April 1, 2010,  (V2016)': 'population',
     'Land area in square miles, 2010': 'land_area',
     'Persons under 18 years, percent, July 1, 2016,  (V2016)':␣
 ↪'under_18y_percent',
         'Persons 65 years and over, percent,  July 1, 2016,  (V2016)':␣
 ↪'over_65y_percent',
         'Female persons, percent,  July 1, 2016,  (V2016)': 'female_percent',
         'Black or African American alone, percent, July 1, 2016,  (V2016)':␣
 ↪'black_percent',
         'American Indian and Alaska Native alone, percent, July 1, 2016, ␣
 ↪(V2016)': 'native_american_percent',
         'Asian alone, percent, July 1, 2016,  (V2016)': 'asian_percent',
         'Native Hawaiian and Other Pacific Islander alone, percent, July 1,␣
 ↪2016,  (V2016)': 'native_pacific_percent',
         'Hispanic or Latino, percent, July 1, 2016,  (V2016)':␣
 ↪'hispanic_percent',
         'White alone, not Hispanic or Latino, percent, July 1, 2016,  (V2016)':␣
 ↪'white_percent',
         'Veterans, 2011-2015': 'veterans',
         'Foreign born persons, percent, 2011-2015': 'foreign_born_percent',
```

```
        'High school graduate or higher, percent of persons age 25 years+,␣
↪2011-2015': 'hs_percent',
        "Bachelor's degree or higher, percent of persons age 25 years+,␣
↪2011-2015": 'bachelor_percent',
        'Total employment, 2015': 'employment',
        'Per capita income in past 12 months (in 2015 dollars), 2011-2015':␣
↪'per_capita_income'
})
```

[18]:
```
census.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 50 entries, Alabama to Wyoming
Data columns (total 17 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   population               50 non-null     object
 1   land_area                50 non-null     object
 2   under_18y_percent        50 non-null     object
 3   over_65y_percent         50 non-null     object
 4   female_percent           50 non-null     object
 5   black_percent            50 non-null     object
 6   native_american_percent  50 non-null     object
 7   asian_percent            50 non-null     object
 8   native_pacific_percent   50 non-null     object
 9   hispanic_percent         50 non-null     object
 10  white_percent            50 non-null     object
 11  veterans                 50 non-null     object
 12  foreign_born_percent     50 non-null     object
 13  hs_percent               50 non-null     object
 14  bachelor_percent         50 non-null     object
 15  employment               50 non-null     object
 16  per_capita_income        50 non-null     object
dtypes: object(17)
memory usage: 7.0+ KB
```

Now, we see that all the values are of string data type. So we need to transform all of them to numbers.

First we need to strip the strings of their special characters, or non-numerical values:

(Note that in some fields a Z value is a placeholder for a very small nonzero value, that we choose to put to 0)

[19]:
```
for c in ['population', 'land_area', 'veterans', 'employment',␣
↪'per_capita_income']:
    census[c] = census[c].apply(lambda x: x.replace(',',''))
```

```
for c in census.columns.values[census.columns.str.contains('percent')]:
    census[c] = census[c].apply(lambda x: x.replace('Z', '0'))
    census[c] = census[c].apply(lambda x: x.replace('%',''))

census['per_capita_income'] = census['per_capita_income'].apply(lambda x: x.
 ↪replace('$',''))
```

[20]:
```
for c in census.columns.values[census.columns.str.contains('percent')]:
    census[c] = census[c].apply(lambda x: float(x))
```

[21]:
```
for c in ['land_area', 'per_capita_income']:
    census[c] = census[c].apply(lambda x: float(x))

for c in ['population', 'veterans', 'employment']:
    census[c] = census[c].apply(lambda x: int(x))
```

[22]:
```
census.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 50 entries, Alabama to Wyoming
Data columns (total 17 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   population               50 non-null     int64
 1   land_area                50 non-null     float64
 2   under_18y_percent        50 non-null     float64
 3   over_65y_percent         50 non-null     float64
 4   female_percent           50 non-null     float64
 5   black_percent            50 non-null     float64
 6   native_american_percent  50 non-null     float64
 7   asian_percent            50 non-null     float64
 8   native_pacific_percent   50 non-null     float64
 9   hispanic_percent         50 non-null     float64
 10  white_percent            50 non-null     float64
 11  veterans                 50 non-null     int64
 12  foreign_born_percent     50 non-null     float64
 13  hs_percent               50 non-null     float64
 14  bachelor_percent         50 non-null     float64
 15  employment               50 non-null     int64
 16  per_capita_income        50 non-null     float64
dtypes: float64(14), int64(3)
memory usage: 7.0+ KB
```

[23]:
```
census.head()
census.shape
```

[23]: (50, 17)

Now I am satisfied with my census dataset. The state serves as the index. And we have 17 numerical features. They have a more manageable format, and the names ends with "percent" when the value is a percentage.

### 2.1.3 Data cleaning for the NICS dataset

Now I want to compare which states and territories are common to the census and NICS dataset.

[24]:
```
nics.state.unique()
```

[24]:
```
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)
```

[25]:
```
nics.shape[0] / nics[nics.state.isin(census.index.values)].shape[0]
nics.state.nunique()
```

[25]: 55

So, in addition to the 50 states, the NICS contains data on 5 territories. The census has data for the 50 States, so I will drop the NICS Territories data.

[26]:
```
nics = nics[nics.state.isin(census.index.values)]
```

[27]:
```
# nics.shape
# nics.fillna(0).describe()
# nics.info()
nics.describe()
```

[27]:

| | permit | permit_recheck | handgun | long_gun | \ |
|---|---|---|---|---|---|
| count | 11348.000000 | 1000.000000 | 11350.000000 | 11350.000000 | |
| mean | 7041.630331 | 1282.552000 | 6509.303877 | 8575.439648 | |
| std | 24801.129677 | 9667.124288 | 8829.284061 | 9416.217660 | |
| min | 0.000000 | 0.000000 | 0.000000 | 0.000000 | |
| 25% | 0.000000 | 0.000000 | 1327.250000 | 2778.000000 | |
| 50% | 814.000000 | 0.000000 | 3622.500000 | 5893.000000 | |

```
75%      5137.750000       1.000000    7987.750000   11021.500000
max    522188.000000  116681.000000  107224.000000  108058.000000

             other     multiple         admin  prepawn_handgun  \
count  5000.000000  11350.000000  11348.000000      9597.000000
mean    396.052400    295.059471     64.675097         5.301969
std    1410.425364    816.710594    633.514277        11.321981
min       0.000000      0.000000      0.000000         0.000000
25%      37.000000     41.000000      0.000000         0.000000
50%     148.000000    151.000000      0.000000         0.000000
75%     393.000000    328.750000      0.000000         5.000000
max   77929.000000  38907.000000  28083.000000       164.000000

       prepawn_long_gun  prepawn_other  …  returned_other  rentals_handgun  \
count       9595.000000    4650.000000  …     1650.000000       900.000000
mean           8.604482       0.181720  …        1.130303         0.084444
std           17.067140       1.107227  …        4.587865         0.665018
min            0.000000       0.000000  …        0.000000         0.000000
25%            0.000000       0.000000  …        0.000000         0.000000
50%            2.000000       0.000000  …        0.000000         0.000000
75%            9.000000       0.000000  …        0.000000         0.000000
max          269.000000      49.000000  …       64.000000        12.000000

       rentals_long_gun  private_sale_handgun  private_sale_long_gun  \
count        750.000000           2500.000000            2500.000000
mean           0.096000             16.427200              12.763200
std            0.703878             74.529346              56.771827
min            0.000000              0.000000               0.000000
25%            0.000000              0.000000               0.000000
50%            0.000000              0.000000               0.000000
75%            0.000000              4.000000               5.000000
max           12.000000           1017.000000             777.000000

       private_sale_other  return_to_seller_handgun  \
count         2500.000000               2250.000000
mean             1.131600                  0.439556
std              4.673652                  1.511227
min              0.000000                  0.000000
25%              0.000000                  0.000000
50%              0.000000                  0.000000
75%              0.000000                  0.000000
max             71.000000                 28.000000

       return_to_seller_long_gun  return_to_seller_other        totals
count                2500.000000             2050.000000  11350.000000
mean                    0.484400                0.115610  23734.978502
std                     1.596113                0.446012  33437.577310
```

```
min                    0.000000               0.000000       6.000000
25%                    0.000000               0.000000    6472.000000
50%                    0.000000               0.000000   14050.000000
75%                    0.000000               0.000000   27537.000000
max                   17.000000               4.000000  541978.000000

[8 rows x 25 columns]
```

The NICS dataset runs from November 1998 to September 2017.

I will create a subset of the NICS data, based only on the last 5 years for comparison with the census: "nics_5y".

The NICS dataset is too wide to see all columns in a describe method, so I will split the columns in two parts to see all the fearures.

[28]:
```
nics_5y = nics[nics['month']>'2012-09']
```

[29]:
```
n_row, n_col = nics.shape
nics.iloc[:, 0:int(n_col/2)].describe()
```

[29]:
```
              permit  permit_recheck        handgun       long_gun  \
count   11348.000000     1000.000000   11350.000000   11350.000000
mean     7041.630331     1282.552000    6509.303877    8575.439648
std     24801.129677     9667.124288    8829.284061    9416.217660
min         0.000000        0.000000       0.000000       0.000000
25%         0.000000        0.000000    1327.250000    2778.000000
50%       814.000000        0.000000    3622.500000    5893.000000
75%      5137.750000        1.000000    7987.750000   11021.500000
max    522188.000000   116681.000000  107224.000000  108058.000000

              other      multiple         admin  prepawn_handgun  \
count   5000.000000  11350.000000  11348.000000      9597.000000
mean     396.052400    295.059471     64.675097         5.301969
std     1410.425364    816.710594    633.514277        11.321981
min        0.000000      0.000000      0.000000         0.000000
25%       37.000000     41.000000      0.000000         0.000000
50%      148.000000    151.000000      0.000000         0.000000
75%      393.000000    328.750000      0.000000         5.000000
max    77929.000000  38907.000000  28083.000000       164.000000

       prepawn_long_gun  prepawn_other  redemption_handgun
count       9595.000000    4650.000000         9600.000000
mean           8.604482       0.181720          448.087604
std           17.067140       1.107227          810.116491
min            0.000000       0.000000            0.000000
25%            0.000000       0.000000            0.000000
50%            2.000000       0.000000           74.000000
```

```
75%             9.000000         0.000000          647.000000
max           269.000000        49.000000        10046.000000
```

[30]:
```
nics.iloc[:, int(n_col/2):].describe()
```

[30]:
```
       redemption_long_gun  redemption_other  returned_handgun  \
count          9598.000000       4650.000000       2000.000000
mean            658.392582          1.995699         32.573500
std             978.295189          4.760378         85.381401
min               0.000000          0.000000          0.000000
25%               0.000000          0.000000          0.000000
50%             235.000000          0.000000          0.000000
75%             977.000000          2.000000         20.000000
max            8831.000000         79.000000        603.000000

       returned_long_gun  returned_other  rentals_handgun  rentals_long_gun  \
count        1950.000000     1650.000000       900.000000        750.000000
mean            8.313333        1.130303         0.084444          0.096000
std            23.040103        4.587865         0.665018          0.703878
min             0.000000        0.000000         0.000000          0.000000
25%             0.000000        0.000000         0.000000          0.000000
50%             0.000000        0.000000         0.000000          0.000000
75%             5.000000        0.000000         0.000000          0.000000
max           168.000000       64.000000        12.000000         12.000000

       private_sale_handgun  private_sale_long_gun  private_sale_other  \
count           2500.000000            2500.000000         2500.000000
mean              16.427200              12.763200            1.131600
std               74.529346              56.771827            4.673652
min                0.000000               0.000000            0.000000
25%                0.000000               0.000000            0.000000
50%                0.000000               0.000000            0.000000
75%                4.000000               5.000000            0.000000
max             1017.000000             777.000000           71.000000

       return_to_seller_handgun  return_to_seller_long_gun  \
count               2250.000000                2500.000000
mean                   0.439556                   0.484400
std                    1.511227                   1.596113
min                    0.000000                   0.000000
25%                    0.000000                   0.000000
50%                    0.000000                   0.000000
75%                    0.000000                   0.000000
max                   28.000000                  17.000000

       return_to_seller_other        totals
count             2050.000000  11350.000000
```

```
mean                 0.115610   23734.978502
std                  0.446012   33437.577310
min                  0.000000       6.000000
25%                  0.000000    6472.000000
50%                  0.000000   14050.000000
75%                  0.000000   27537.000000
max                  4.000000  541978.000000
```

Among the different categories, I will keep only the columns where the mean of at least one subcategory is above 100, which means that there are more than 100 background checks per state per month on average.

Therefore, I will drop the admin, prepawn, returned, rentals, private_sale, return_to_seller. It makes sense because they are either about giving away a weapon, administrative or an uncommon way of acquiring one.

[31]:
```
nics.loc[:, 'permit':'totals'].mean() > 100
```

[31]:
```
permit                        True
permit_recheck                True
handgun                       True
long_gun                      True
other                         True
multiple                      True
admin                        False
prepawn_handgun              False
prepawn_long_gun             False
prepawn_other                False
redemption_handgun            True
redemption_long_gun           True
redemption_other             False
returned_handgun             False
returned_long_gun            False
returned_other               False
rentals_handgun              False
rentals_long_gun             False
private_sale_handgun         False
private_sale_long_gun        False
private_sale_other           False
return_to_seller_handgun     False
return_to_seller_long_gun    False
return_to_seller_other       False
totals                        True
dtype: bool
```

[32]:
```
nics.isna().sum()
```

[32]:
```
month                          0
state                          0
permit                         2
permit_recheck             10350
handgun                        0
long_gun                       0
other                       6350
multiple                       0
admin                          2
prepawn_handgun             1753
prepawn_long_gun            1755
prepawn_other               6700
redemption_handgun          1750
redemption_long_gun         1752
redemption_other            6700
returned_handgun            9350
returned_long_gun           9400
returned_other              9700
rentals_handgun            10450
rentals_long_gun           10600
private_sale_handgun        8850
private_sale_long_gun       8850
private_sale_other          8850
return_to_seller_handgun    9100
return_to_seller_long_gun   8850
return_to_seller_other      9300
totals                         0
dtype: int64
```

Among the remaining columns, for the purpose of the analysis, I decide to drop the permit_recheck because of the missing values, and the redemption category, in order to concentrate on permits and types of guns.

[33]:
```
nics[nics['permit'].isna()]
```

[33]:
```
        month      state  permit  permit_recheck  handgun  long_gun  other  \
7279  2006-09  Louisiana     NaN             NaN   5948.0   10836.0    NaN
7310  2006-09   Virginia     NaN             NaN   6935.0   11023.0    NaN

      multiple  admin  prepawn_handgun  …  returned_other  rentals_handgun  \
7279       253    0.0              5.0  …             NaN              NaN
7310       242    0.0              0.0  …             NaN              NaN

      rentals_long_gun  private_sale_handgun  private_sale_long_gun  \
7279               NaN                   NaN                    NaN
7310               NaN                   NaN                    NaN
```

```
     private_sale_other  return_to_seller_handgun  return_to_seller_long_gun  \
7279                NaN                       NaN                        NaN
7310                NaN                       NaN                        NaN

     return_to_seller_other  totals
7279                    NaN   19080
7310                    NaN   18200

[2 rows x 27 columns]
```

[34]:
```python
nics[nics.other.isna()].head(1)
```

[34]:
```
        month    state  permit  permit_recheck  handgun  long_gun  other  \
5500  2009-05  Alabama     0.0             NaN   8829.0    7878.0    NaN

     multiple  admin  prepawn_handgun  …  returned_other  rentals_handgun  \
5500      559    0.0             11.0  …             NaN              NaN

     rentals_long_gun  private_sale_handgun  private_sale_long_gun  \
5500              NaN                   NaN                    NaN

     private_sale_other  return_to_seller_handgun  return_to_seller_long_gun  \
5500                NaN                       NaN                        NaN

     return_to_seller_other  totals
5500                    NaN   20277

[1 rows x 27 columns]
```

I will fill the other missing values with 0. There are about 2 missing permit values, and the "other" category which is filled after May 2009 seemingly.

[35]:
```python
nics = nics[['month', 'state', 'permit', 'handgun', 'long_gun', 'other',␣
 ↪'multiple', 'totals']].fillna(0)
nics_5y = nics_5y[['month', 'state', 'permit', 'handgun', 'long_gun', 'other',␣
 ↪'multiple', 'totals']].fillna(0)
```

[36]:
```python
nics.isna().sum()
```

[36]:
```
month       0
state       0
permit      0
handgun     0
long_gun    0
other       0
multiple    0
totals      0
```

```
dtype: int64
```

## Exploratory Data Analysis

After the cleanup, in this section, we want to answer the following questions:

1. What are the country-wide trend of gun purchases during the last 20 years?
2. What are the states that give rise to the most NICS checks?
3. What census data is most associated with high gun per capita?

A caveat of this analysis is that we will use NICS background, not gun purchases per se. Because of the varying state laws and purchase scenarios, there is not a one-to-one correspondance between background checks and gun purchases.

Nevertheless, in this project, we will consider it a good enough proxy.

### 2.1.4 Country-wide trend of gun purchases

In the following, we use a groupby method to sum the data of the 50 states and see the time evolution.

[37]:
```
nics_country = nics.groupby('month').sum()
nics_5y_country = nics_5y_country = nics_5y.groupby('month').sum()
```

[38]:
```
nics_country.tail()
```

[38]:

| month | permit | handgun | long_gun | other | multiple | totals |
|---|---|---|---|---|---|---|
| 2017-05 | 830898.0 | 550551.0 | 327885.0 | 29297.0 | 16926 | 1896910 |
| 2017-06 | 800512.0 | 567630.0 | 330890.0 | 29690.0 | 17030 | 1886240 |
| 2017-07 | 754798.0 | 478844.0 | 322020.0 | 26838.0 | 15682 | 1731550 |
| 2017-08 | 789893.0 | 506098.0 | 396659.0 | 27583.0 | 18221 | 1894569 |
| 2017-09 | 761620.0 | 477315.0 | 417126.0 | 26897.0 | 17612 | 1856214 |

[39]:
```
nics_5y_country.tail()
```

[39]:

| month | permit | handgun | long_gun | other | multiple | totals |
|---|---|---|---|---|---|---|
| 2017-05 | 830898.0 | 550551.0 | 327885.0 | 29297.0 | 16926 | 1896910 |
| 2017-06 | 800512.0 | 567630.0 | 330890.0 | 29690.0 | 17030 | 1886240 |
| 2017-07 | 754798.0 | 478844.0 | 322020.0 | 26838.0 | 15682 | 1731550 |
| 2017-08 | 789893.0 | 506098.0 | 396659.0 | 27583.0 | 18221 | 1894569 |
| 2017-09 | 761620.0 | 477315.0 | 417126.0 | 26897.0 | 17612 | 1856214 |

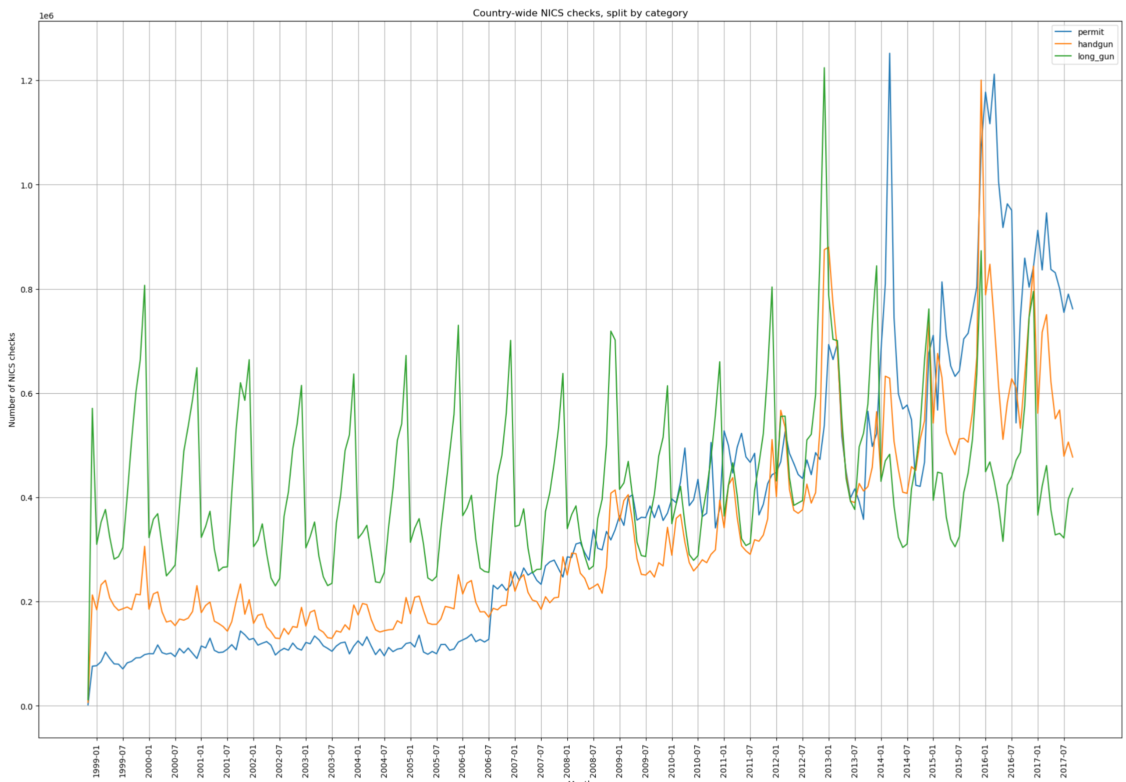
[40]:
```
n_month = nics.shape[0]/50
```

First, I want to plot the data for each gun category for the last 20 years.

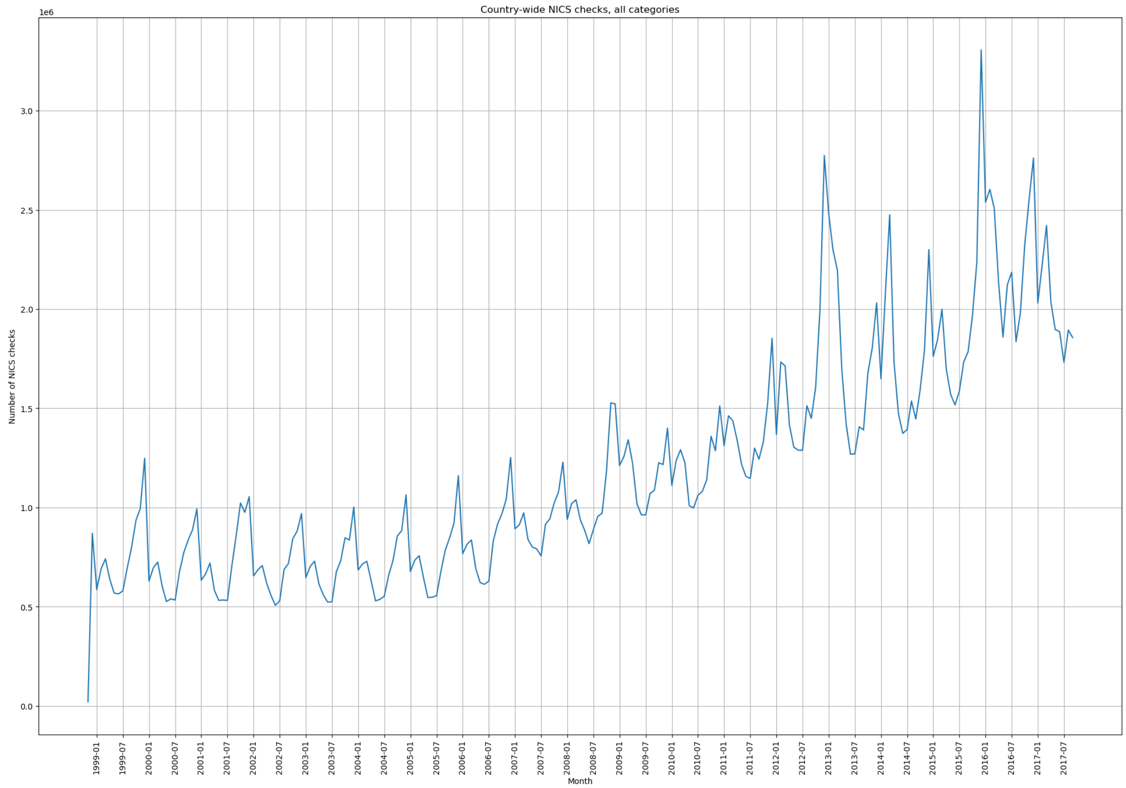
I noticed that my x ticks were cluttered, so I decided to display them vertically and every 6 months, starting January 1999.

The “other” and “multiple” (even if multiply by a small integer, which is a reasonable expectation of “multiple”) categories seem to be a small part of the overall background checks, si I dropped them from the figure.

[41]:
```
plt.figure(figsize=[24,16])
plt.plot(nics_country.index, nics_country.loc[:, 'permit':'long_gun'])
plt.xlabel('Month')
plt.ylabel('Number of NICS checks')
plt.legend(nics_country.columns)
plt.title('Country-wide NICS checks, split by category')
plt.xticks(np.arange(2, n_month, step=6), rotation='vertical')
plt.grid(True)
plt.show()
```

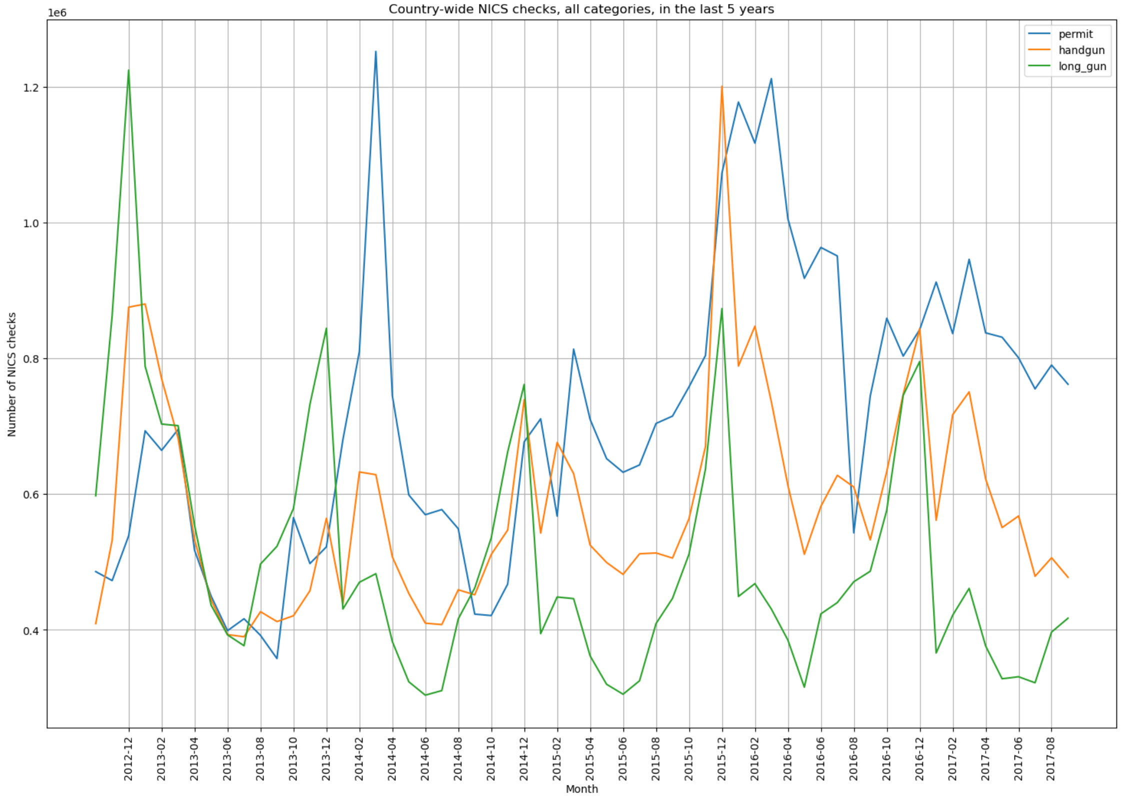



[42]:
```
plt.figure(figsize=[24,16])
plt.plot(nics_country.index, nics_country['totals'])
plt.xlabel('Month')
plt.ylabel('Number of NICS checks')
plt.title('Country-wide NICS checks, all categories')
plt.xticks(np.arange(2, n_month, step=6), rotation='vertical')
plt.grid(True)
plt.show()
```




Zoom-in on the last 5 years:

[43]:
```
plt.figure(figsize=[18,12])
plt.plot(nics_5y_country.index, nics_5y_country.loc[:, 'permit':'long_gun'])
plt.xlabel('Month')
plt.ylabel('Number of NICS checks')
plt.legend(nics_5y_country.columns)
plt.title('Country-wide NICS checks, all categories, in the last 5 years')
plt.xticks(np.arange(2, 60, step=2), rotation='vertical')
plt.grid(True)
plt.show()
```




We notice first that the overall gun purchases seems to increase for the last 10 years. This increase is driven by permit and handgun triggered background checks. The long gun purchases seem more stable over time.

There is an interesting periodic pattern, with a peak in December and a trough in the Summer, for gun purchases.

Permits seem to follow along but more irregularly, with for example a sudden increase in 2006, maybe triggered by new laws.

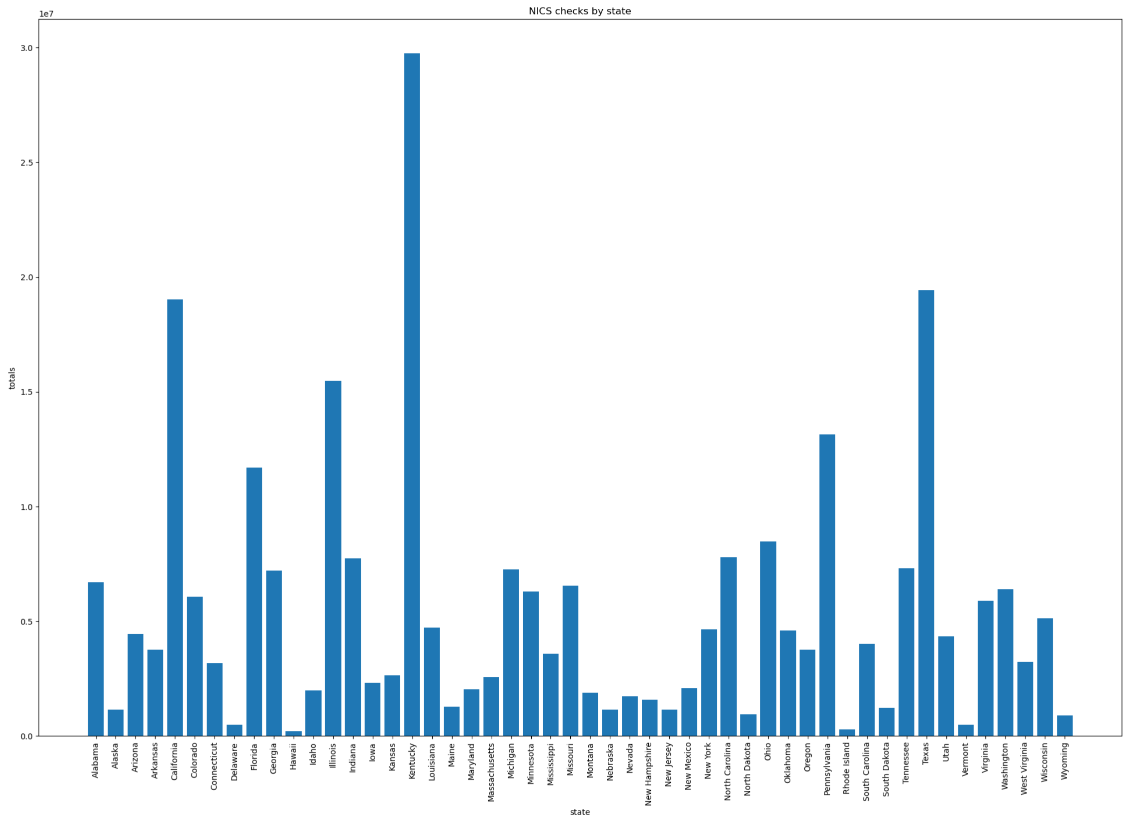
#### 2.1.5 State-level gun purchases

Now, let us look at the state level. We use a groupby method on our 20-year and 5-year datasets.

[44]:
```
nics_states = nics.groupby('state').sum()
nics_5y_states = nics_5y.groupby('state').sum()
```

[45]:
```
plt.figure(figsize=[24,16])
plt.bar(nics_states.index, nics_states.totals)
plt.xlabel('state')
plt.ylabel('totals')
plt.title('NICS checks by state')
```

```
plt.xticks(rotation='vertical')
plt.show()
```




At first look, we notice that for obvious reasons, the background checks are correlated, with population size, so let us look at the per capita data.

[46]:
```
nics_per_capita = nics_states.join(census['population'])
nics_5y_per_capita = nics_5y_states.join(census['population'])
```

[47]:
```
for c in nics_per_capita.columns:
    nics_per_capita.loc[:, c] = nics_per_capita.loc[:, c] /␣
 ↪nics_per_capita['population']
    nics_5y_per_capita.loc[:, c] = nics_5y_per_capita.loc[:, c] /␣
 ↪nics_5y_per_capita['population']
```

[48]:
```
nics_sorted = nics_per_capita.sort_values('totals')
nics_5y_sorted = nics_5y_per_capita.sort_values('totals')
```

[49]:
```
nics_5y_sorted
```

[49]:

| state | permit | handgun | long_gun | other | multiple | totals | \ |
|---|---|---|---|---|---|---|---|
| Hawaii | 0.058027 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.058100 | |
| New Jersey | 0.000000 | 0.033330 | 0.025340 | 0.001287 | 0.000000 | 0.059957 | |
| New York | 0.011731 | 0.029167 | 0.050123 | 0.002327 | 0.000515 | 0.096103 | |
| Rhode Island | 0.000000 | 0.058588 | 0.045509 | 0.003136 | 0.008919 | 0.117129 | |
| Maryland | 0.018633 | 0.058158 | 0.061998 | 0.001277 | 0.000209 | 0.142885 | |
| Massachusetts | 0.074675 | 0.050360 | 0.028517 | 0.003782 | 0.001600 | 0.159418 | |
| Nebraska | 0.139183 | 0.004259 | 0.077593 | 0.000611 | 0.000165 | 0.225742 | |
| California | 0.093827 | 0.065447 | 0.058177 | 0.009176 | 0.000000 | 0.228227 | |
| Nevada | 0.047976 | 0.108296 | 0.073261 | 0.005514 | 0.006645 | 0.252128 | |
| Michigan | 0.109877 | 0.067469 | 0.069049 | 0.002404 | 0.001244 | 0.254935 | |
| Iowa | 0.190489 | 0.003935 | 0.060915 | 0.000566 | 0.000101 | 0.260856 | |
| Delaware | 0.025302 | 0.122362 | 0.106429 | 0.005321 | 0.004860 | 0.268953 | |
| Georgia | 0.097230 | 0.088581 | 0.067207 | 0.002451 | 0.003670 | 0.282330 | |
| Arizona | 0.053783 | 0.114455 | 0.080111 | 0.007409 | 0.005533 | 0.282858 | |
| Vermont | 0.000000 | 0.131011 | 0.143305 | 0.006885 | 0.006004 | 0.287624 | |
| Virginia | 0.003210 | 0.165978 | 0.128768 | 0.007196 | 0.000000 | 0.305185 | |
| Florida | 0.057898 | 0.149542 | 0.079868 | 0.007761 | 0.005964 | 0.313266 | |
| Texas | 0.057890 | 0.117722 | 0.101180 | 0.006344 | 0.006317 | 0.317049 | |
| Ohio | 0.032269 | 0.152852 | 0.106071 | 0.007602 | 0.006620 | 0.318008 | |
| Kansas | 0.043516 | 0.133861 | 0.134923 | 0.006929 | 0.007432 | 0.347816 | |
| North Carolina | 0.226958 | 0.007233 | 0.093707 | 0.003816 | 0.001447 | 0.361224 | |
| Maine | 0.012993 | 0.153146 | 0.170386 | 0.007665 | 0.007831 | 0.365675 | |
| New Mexico | 0.027646 | 0.157747 | 0.135391 | 0.009404 | 0.008719 | 0.371123 | |
| South Carolina | 0.121210 | 0.125969 | 0.092834 | 0.005660 | 0.005378 | 0.377882 | |
| Oregon | 0.007849 | 0.197954 | 0.172449 | 0.000000 | 0.000004 | 0.378560 | |
| Connecticut | 0.186753 | 0.130708 | 0.069414 | 0.003868 | 0.000000 | 0.391543 | |
| Louisiana | 0.018605 | 0.165571 | 0.162423 | 0.008963 | 0.008560 | 0.393691 | |
| Wisconsin | 0.101084 | 0.142885 | 0.138690 | 0.006330 | 0.000431 | 0.394410 | |
| Pennsylvania | 0.111774 | 0.170148 | 0.121803 | 0.000406 | 0.000000 | 0.405963 | |
| Mississippi | 0.022777 | 0.164827 | 0.159493 | 0.004988 | 0.007873 | 0.416263 | |
| Washington | 0.124171 | 0.143739 | 0.107718 | 0.012583 | 0.005361 | 0.423372 | |
| Arkansas | 0.091476 | 0.125743 | 0.147867 | 0.003632 | 0.007417 | 0.442363 | |
| Missouri | 0.052693 | 0.207106 | 0.166769 | 0.010285 | 0.009945 | 0.474924 | |
| Colorado | 0.066852 | 0.213688 | 0.171121 | 0.011415 | 0.013315 | 0.477744 | |
| Idaho | 0.125931 | 0.126332 | 0.175147 | 0.006269 | 0.007141 | 0.477836 | |
| Oklahoma | 0.000000 | 0.212904 | 0.182843 | 0.013605 | 0.014492 | 0.483878 | |
| Tennessee | 0.110895 | 0.218842 | 0.161176 | 0.002545 | 0.007735 | 0.511268 | |
| Utah | 0.316774 | 0.078206 | 0.093864 | 0.003822 | 0.002719 | 0.511682 | |
| Minnesota | 0.276586 | 0.107420 | 0.133764 | 0.006377 | 0.004185 | 0.536666 | |
| New Hampshire | 0.151330 | 0.221087 | 0.160014 | 0.005101 | 0.000126 | 0.540769 | |
| North Dakota | 0.079893 | 0.155024 | 0.286010 | 0.006856 | 0.007916 | 0.555308 | |
| Illinois | 0.301987 | 0.104700 | 0.064610 | 0.000000 | 0.004162 | 0.564405 | |
| Wyoming | 0.058540 | 0.204451 | 0.238469 | 0.009756 | 0.011657 | 0.565432 | |
| South Dakota | 0.044997 | 0.197002 | 0.310359 | 0.012335 | 0.012172 | 0.603803 | |
| Alaska | 0.018626 | 0.258097 | 0.272851 | 0.018026 | 0.016500 | 0.619881 | |

```
Alabama         0.215606  0.188065  0.169161  0.006733  0.008845  0.636395
West Virginia   0.051377  0.232692  0.243540  0.007879  0.014503  0.645149
Montana         0.079286  0.183774  0.288947  0.007675  0.013978  0.662223
Indiana         0.359730  0.183859  0.131920  0.009657  0.006322  0.697599
Kentucky        3.135595  0.143398  0.131777  0.003524  0.008196  3.476291

                population
state
Hawaii                 1.0
New Jersey             1.0
New York               1.0
Rhode Island           1.0
Maryland               1.0
Massachusetts          1.0
Nebraska               1.0
California             1.0
Nevada                 1.0
Michigan               1.0
Iowa                   1.0
Delaware               1.0
Georgia                1.0
Arizona                1.0
Vermont                1.0
Virginia               1.0
Florida                1.0
Texas                  1.0
Ohio                   1.0
Kansas                 1.0
North Carolina         1.0
Maine                  1.0
New Mexico             1.0
South Carolina         1.0
Oregon                 1.0
Connecticut            1.0
Louisiana              1.0
Wisconsin              1.0
Pennsylvania           1.0
Mississippi            1.0
Washington             1.0
Arkansas               1.0
Missouri               1.0
Colorado               1.0
Idaho                  1.0
Oklahoma               1.0
Tennessee              1.0
Utah                   1.0
Minnesota              1.0
```

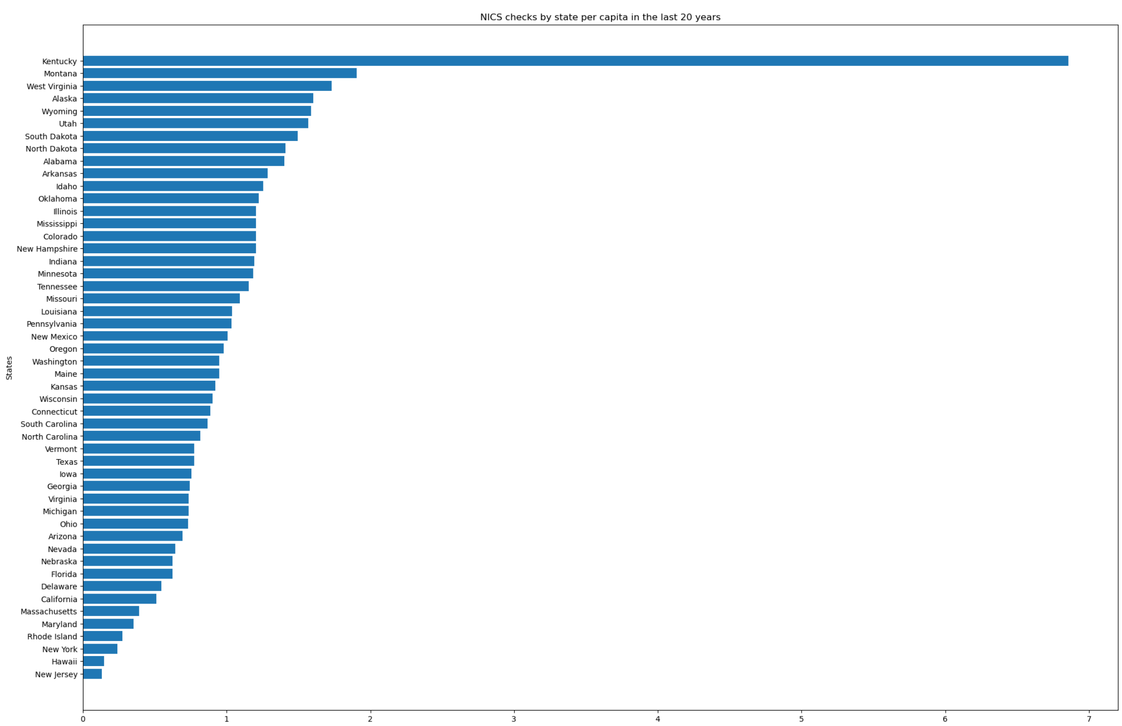
```
New Hampshire           1.0
North Dakota            1.0
Illinois                1.0
Wyoming                 1.0
South Dakota            1.0
Alaska                  1.0
Alabama                 1.0
West Virginia           1.0
Montana                 1.0
Indiana                 1.0
Kentucky                1.0
```

Now we have per capita data, with the sanity check of the population ratio of 1.

[50]:
```
# New function to replace repetitive cells (replaced cells in comments):

def barhplot(dataset, years):
    plt.figure(figsize=[24,16])
    plt.barh(dataset.index, dataset.totals)
    plt.xlabel("NICS checks per capita")
    plt.ylabel("States")
    plt.title("NICS checks by state per capita in the last {} years".
 ↪format(years))
    plt.show()
```

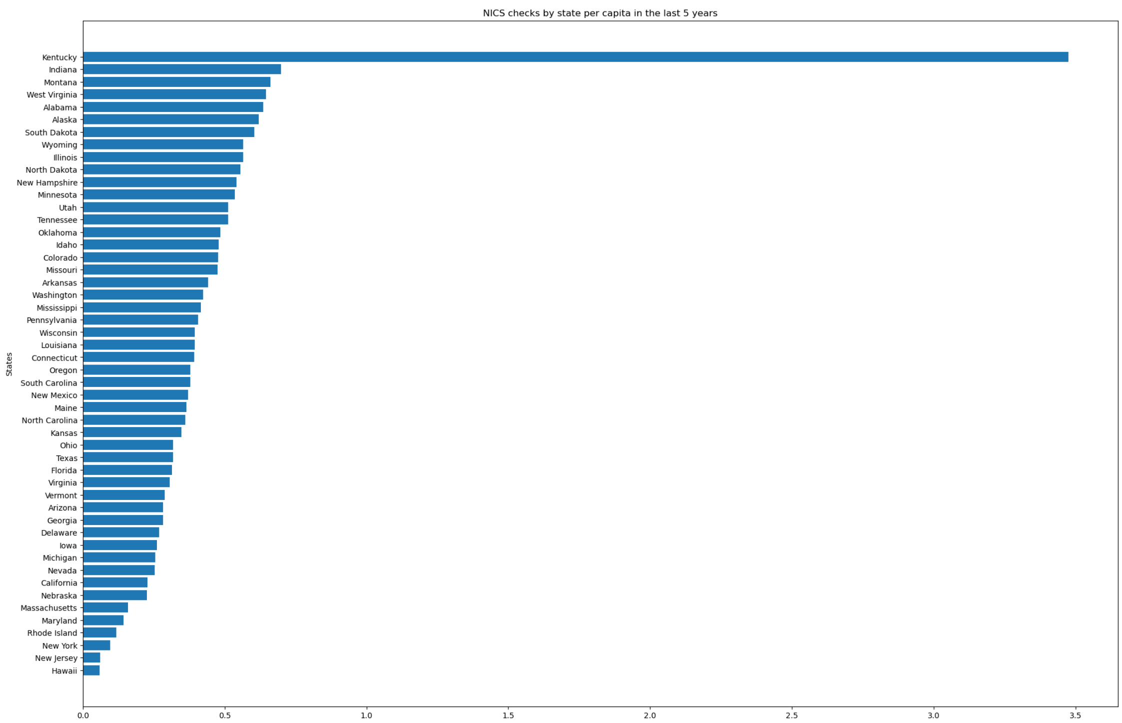
[51]:
```
barhplot(nics_sorted, 20)
```



[52]:
```
# plt.figure(figsize=[24,16])
# plt.barh(nics_sorted.index, nics_sorted.totals)
# plt.xlabel('NICS checks per capita')
# plt.ylabel('States')
# plt.title('NICS checks by state per capita in the last 20 years')

# plt.show()
```

[53]:
```
barhplot(nics_5y_sorted, 5)
```




[54]:
```
# plt.figure(figsize=[24,16])
# plt.barh(nics_5y_sorted.index, nics_5y_sorted.totals)
# plt.xlabel('NICS checks per capita')
# plt.ylabel('States')
# plt.title('NICS checks by state per capita in the last 5 years')

# plt.show()
```

The data are similar for the 2 time intervals. Kentucky is a clear outlier. This is explained in the notes coming with the dataset: "Kentucky runs a new check on each concealed carry license holder each month".

So let's rerun the analysis with only the handgun and long gun variables.

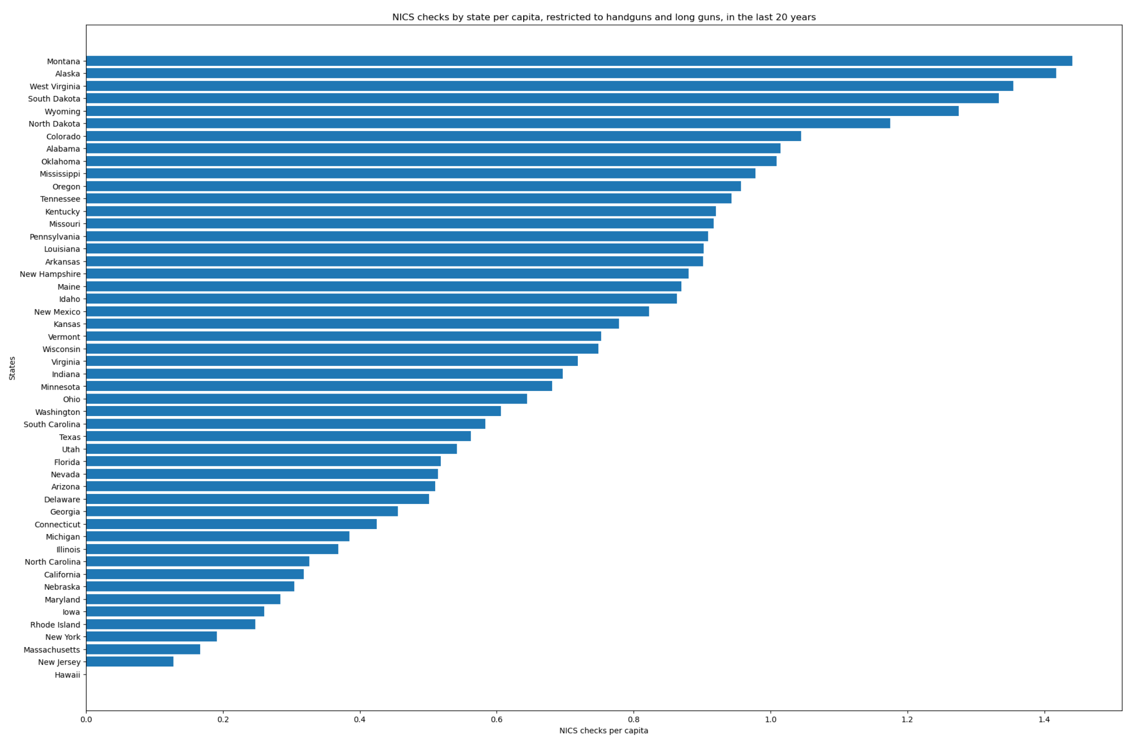
[55]:
```
nics_per_capita['total_gun'] = nics_per_capita['handgun'] +␣
↪nics_per_capita['long_gun']
nics_5y_per_capita['total_gun'] = nics_5y_per_capita['handgun'] +␣
↪nics_5y_per_capita['long_gun']
```

[56]:
```
nics_sorted = nics_per_capita.sort_values('total_gun')
nics_5y_sorted = nics_5y_per_capita.sort_values('total_gun')
```

[57]:
```
# New function to replace repetitive cells (replaced cells in comments):

def barhplot_guns(dataset, years):
    plt.figure(figsize=[24,16])
    plt.barh(dataset.index, dataset.total_gun)
    plt.xlabel("NICS checks per capita")
    plt.ylabel("States")
    plt.title("NICS checks by state per capita, restricted to handguns and long␣
↪guns, in the last {} years".format(years))
    plt.show()
```

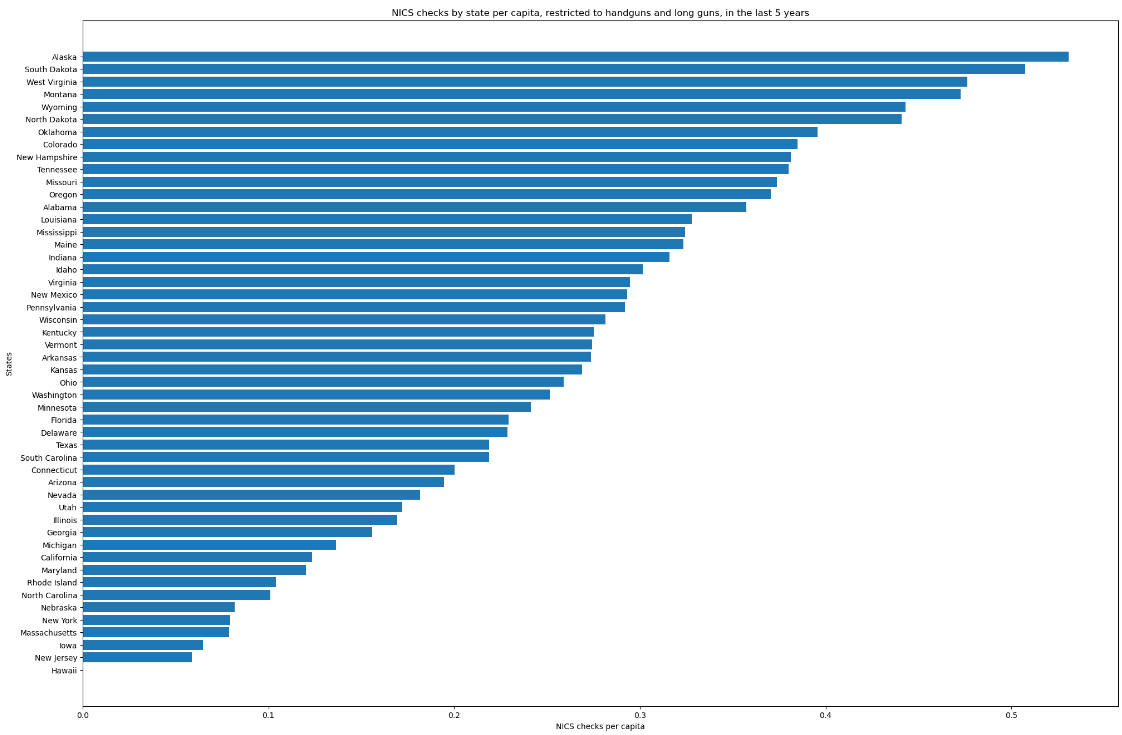
[58]:
```
barhplot_guns(nics_sorted, 20)
```



[59]:
```
# plt.figure(figsize=[24,16])
# plt.barh(nics_sorted.index, nics_sorted.total_gun)
# plt.xlabel('NICS checks per capita')
# plt.ylabel('States')
# plt.title('NICS checks by state per capita, restricted to handguns and long␣
↪guns in the last 20 years')

# plt.show()
```

[60]:
```
barhplot_guns(nics_5y_sorted, 5)
```




[61]:
```
# plt.figure(figsize=[24,16])
# plt.barh(nics_5y_sorted.index, nics_5y_sorted.total_gun)
# plt.xlabel('NICS checks per capita')
# plt.ylabel('States')
# plt.title('NICS checks by state per capita, restricted to handguns and long␣
↪guns, in the last 5 years')

# plt.show()
```

The time frame doesn't seem to matter much. States seem consistent in their gun purchases trends.

Hawai is an outlier with very few gun purchases per capita. Urban states seem on the lower end of gun purchases: New Jersey, Massachusetts, New York, Rhode Island, Maryland… But some rural states are also present: Nebraska, Iowa.

On the high end, we see mainly rural states: Alaska, the Dakotas, West Virginia, Wyoming, Montana...

Let us now see with the next question if we can pin down the relevant demographic variables.

### 2.1.6 Correlations of gun purchases with census variables

[62]:
```
df = census
```

I will transform the following variables:

- a new variable population_density = population / land_area
- verterans_percent = veterans / population * 100
- employment_percent = employment / population * 100

So that the variables are not dependent on the population size.

[63]:
```
df['population_density'] = df['population'] / df['land_area']
df['veterans_percent'] = df['veterans'] / df['population'] * 100
df['employment_percent'] = df['employment'] / df['population'] * 100
```

[64]:
```
# df.head(5)
df.describe()
```

[64]:
```
State    population      land_area  under_18y_percent  over_65y_percent  \
count  5.000000e+01      50.000000          50.000000          50.00000
mean   6.163127e+06   70636.887800          17.425780          11.81044
std    6.848463e+06   85815.678218           9.936137           6.84738
min    5.637670e+05    1033.810000           0.197000           0.14500
25%    1.833003e+06   36741.167500          19.025000          10.42500
50%    4.436412e+06   53891.280000          22.250000          15.00000
75%    6.680362e+06   81225.725000          23.450000          16.10000
max    3.725452e+07  570640.950000          30.200000          19.90000

State  female_percent  black_percent  native_american_percent  asian_percent  \
count       50.000000      50.000000                 50.00000       50.00000
mean        38.511740       8.360900                  1.18026        3.63118
std         21.584467       9.917937                  2.36307        5.79106
min          0.487000       0.020000                  0.00300        0.01500
25%         48.000000       0.650000                  0.22500        0.80000
50%         50.300000       4.700000                  0.60000        2.35000
75%         50.900000      12.400000                  1.17500        4.55000
max         51.600000      37.700000                 15.20000       37.70000

State  native_pacific_percent  hispanic_percent  white_percent      veterans  \
count               50.000000         50.000000      50.000000  5.000000e+01
mean                 0.360360          8.874680      52.779080  4.015940e+05
```

```
std                  1.444674          9.805721      32.872755  3.831585e+05
min                  0.000000          0.036000       0.381000  4.470800e+04
25%                  0.002000          1.525000      26.000000  1.332715e+05
50%                  0.100000          6.250000      62.050000  3.020175e+05
75%                  0.100000         11.575000      79.600000  4.949490e+05
max                 10.200000         39.100000      93.500000  1.777410e+06

State  foreign_born_percent  hs_percent  bachelor_percent    employment  \
count              50.00000   50.000000         50.000000  5.000000e+01
mean                7.07704   67.424100         22.363160  2.471459e+06
std                 6.75162   37.882927         13.383697  2.733595e+06
min                 0.03000    0.842000          0.241000  2.198810e+05
25%                 1.65000   81.825000         19.575000  5.888890e+05
50%                 4.80000   87.850000         27.200000  1.606934e+06
75%                11.22500   90.625000         31.075000  3.040622e+06
max                27.00000   92.800000         40.500000  1.432538e+07

State  per_capita_income  population_density  veterans_percent  \
count          50.000000           50.000000         50.000000
mean        28491.780000          194.981476          7.173003
std          4103.284534          261.118884          1.213013
min         21057.000000            1.244651          4.275887
25%         25443.750000           44.433008          6.485900
50%         27669.500000           98.724207          7.293251
75%         30977.000000          209.500803          7.997617
max         38803.000000         1195.497687          9.760380

State  employment_percent
count           50.000000
mean            40.095804
std              4.548393
min             30.413977
25%             37.776170
50%             40.206493
75%             42.237600
max             54.400520
```

[65]:
```
df = df.join(nics_sorted.total_gun)
```

[66]:
```
df.corr()['total_gun']
```

[66]:
```
population               -0.379411
land_area                 0.358228
under_18y_percent        -0.088005
over_65y_percent         -0.086718
female_percent           -0.104708
black_percent            -0.146017
```

```
native_american_percent    0.343734
asian_percent             -0.456416
native_pacific_percent    -0.269873
hispanic_percent          -0.347722
white_percent              0.100698
veterans                  -0.333765
foreign_born_percent      -0.528301
hs_percent                -0.090830
bachelor_percent          -0.218145
employment                -0.391453
per_capita_income         -0.343633
population_density        -0.579734
veterans_percent           0.573048
employment_percent        -0.210771
total_gun                  1.000000
Name: total_gun, dtype: float64
```

[67]:
```
df.corr()['total_gun'][df.corr()['total_gun']>.3]
```

[67]:
```
land_area                  0.358228
native_american_percent    0.343734
veterans_percent           0.573048
total_gun                  1.000000
Name: total_gun, dtype: float64
```

[68]:
```
df.corr()['total_gun'][df.corr()['total_gun']<-.3]
```

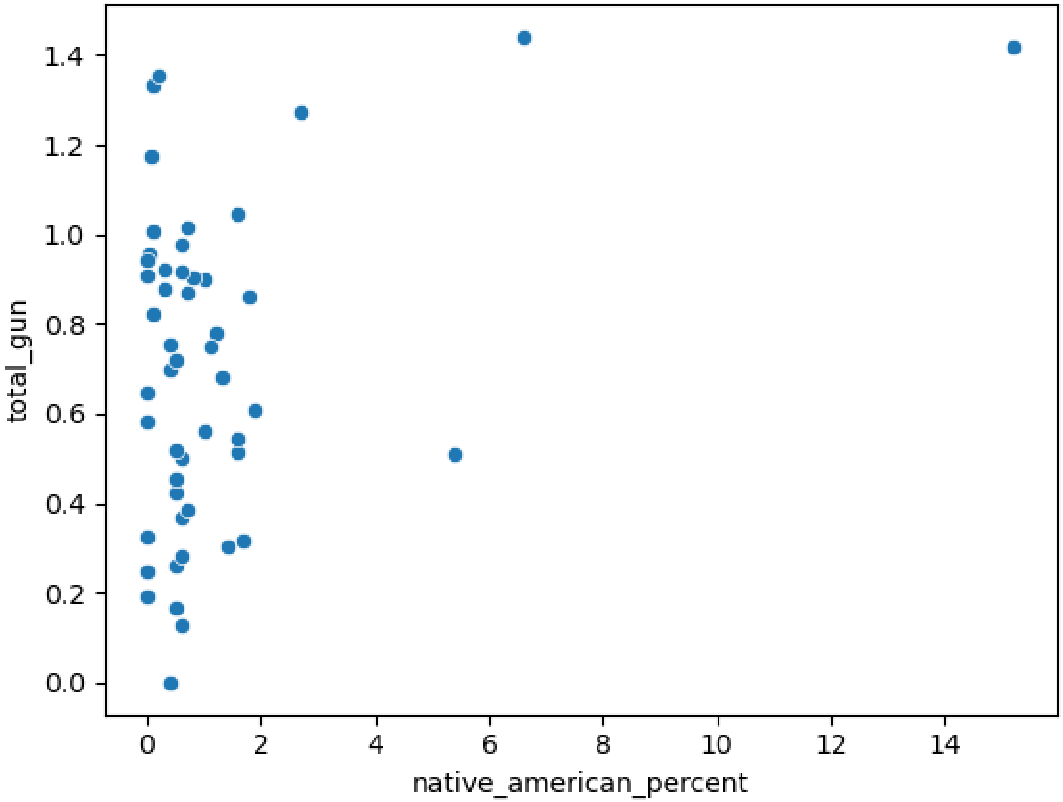
[68]:
```
population              -0.379411
asian_percent           -0.456416
hispanic_percent        -0.347722
veterans                -0.333765
foreign_born_percent    -0.528301
employment              -0.391453
per_capita_income       -0.343633
population_density      -0.579734
Name: total_gun, dtype: float64
```

We see that gun per capita is correlated with 3 variables: land_area, native_american_percent, veterans_percent.

The native_american_percent correlation may be misleading because their share is low in most states. So I am unsure if the correlation is relevant.

[69]:
```
sns.scatterplot(x=df.native_american_percent, y=df.total_gun)
```

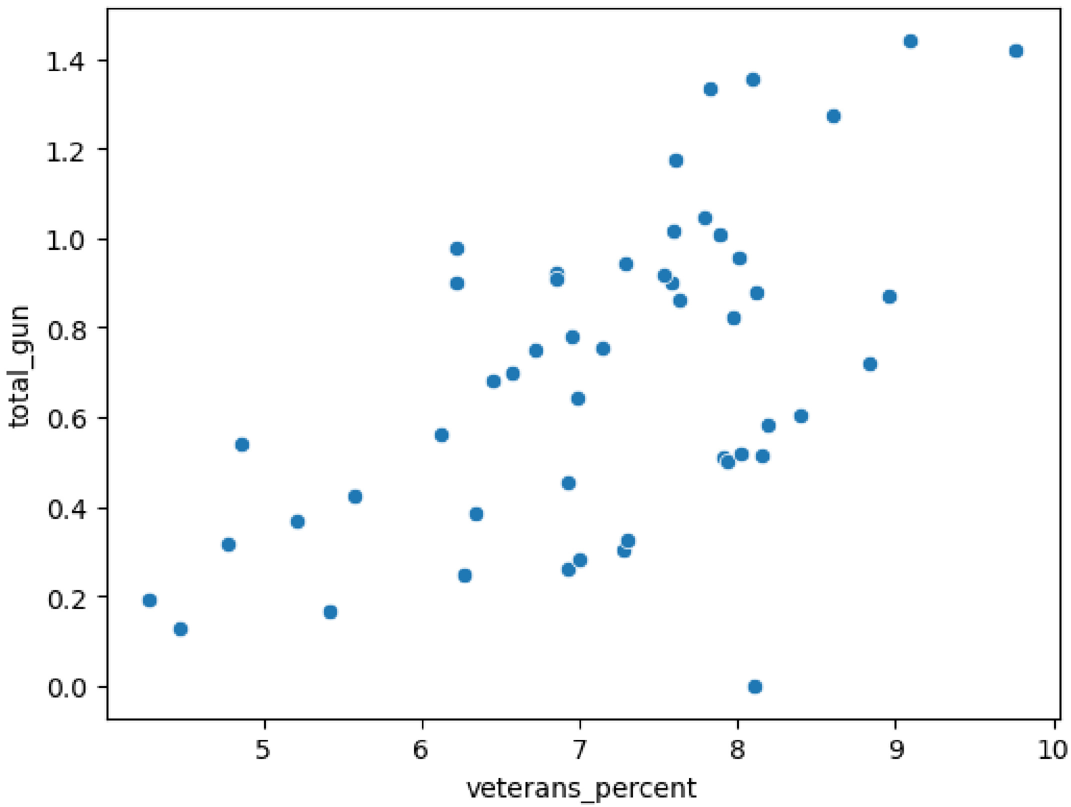
[69]:
```
<AxesSubplot:xlabel='native_american_percent', ylabel='total_gun'>
```




veterans_percent has the stronger positive correlation of 0.57. It is sensible that people familiar with guns and with self defense have the inclination to own guns. The negative correlation with total veterans number may be caused by the bias caused by population size.

[70]:
```
sns.scatterplot(x=df.veterans_percent, y=df.total_gun)
```

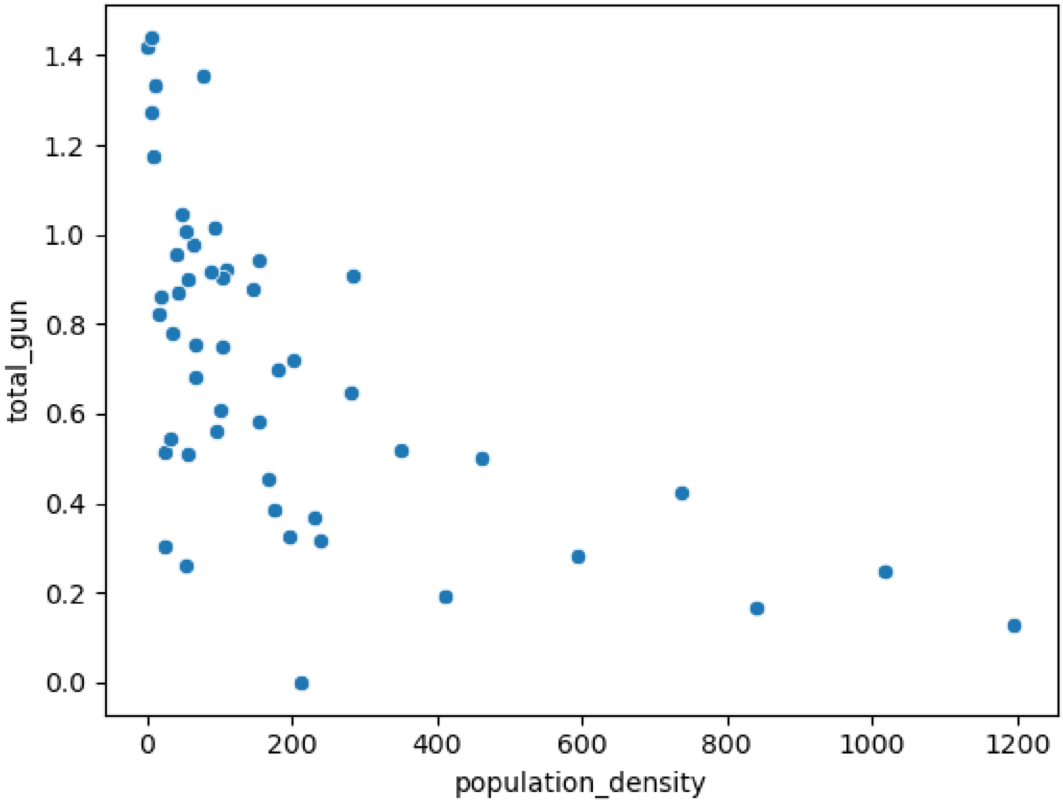
[70]: `<AxesSubplot:xlabel='veterans_percent', ylabel='total_gun'>`




Land area makes sense but I think this is even more obvious when we will at the negative correlation with population size and especially strong negative correlation of -0.58 of gun ownership with population density. It may have to do with the rural aspect, where we can imagine that in large and wild lands, people may be more inclined to protect themselves, in contrast with urban settings where police forces have more oversight, and people can rely more on their neighbors.

[71]:
```
sns.scatterplot(x=df.population_density, y=df.total_gun)
```

[71]: <AxesSubplot:xlabel='population_density', ylabel='total_gun'>



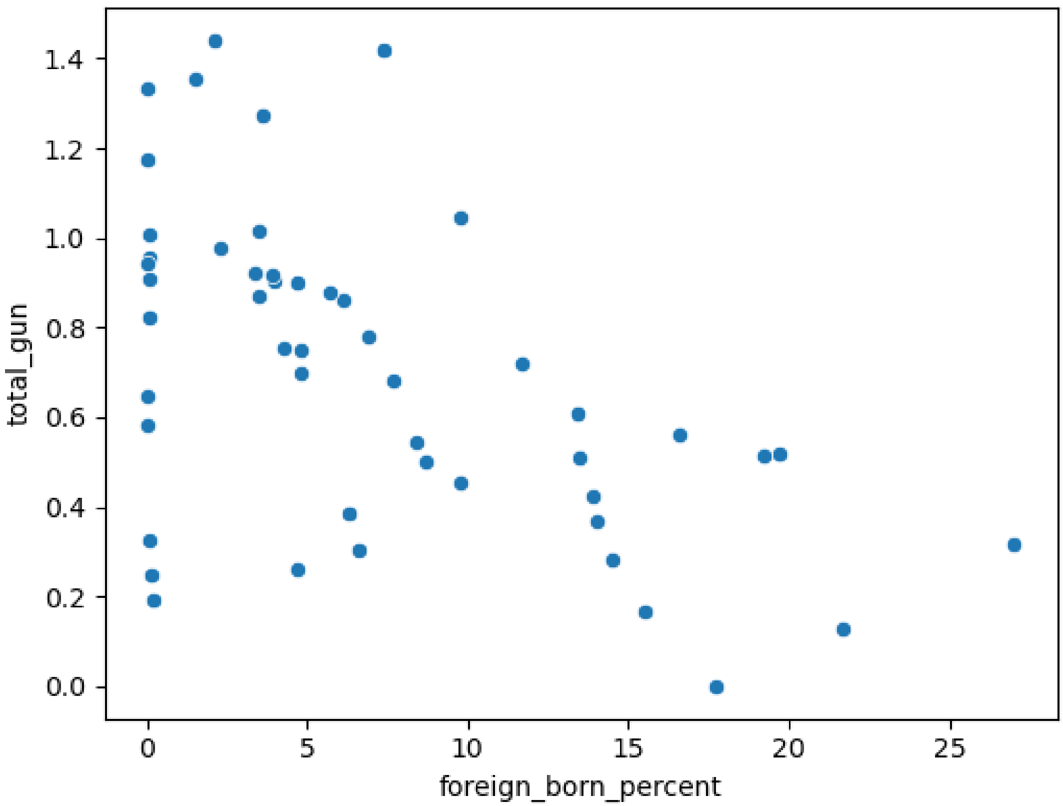

Two demographics variable: Hispanic and Asian proportions of population are negatively correlated with gun ownership. It may indicates either a lesser affinity of these populations with gun ownership.

Or, given that Asian and Hispanic immigration are the more recent immmigration in the U.S.A., and that foreign_born_percent is strongly negatively correlated (-0.53) with gun ownership, it may be that foreigners are less inclined towards gun ownership that demographics from early immigration to the U.S.A.

Another possible confounder is the rural/urban divide, given that immigrants usually cluster around cities, and less in sparsely populated areas.

[72]:
```
sns.scatterplot(x=df.foreign_born_percent, y=df.total_gun)
```

[72]: <AxesSubplot:xlabel='foreign_born_percent', ylabel='total_gun'>




Finally, per_capita income is negatively correlated with gun purchases, along with employment, even if for the latter the magnitude of the correlation diminish if we consider employment rate. There may be a pattern of people earning more living in more secure settings, having higher trust in their communities. Or it may be confounded by the higher population density or foreign born share associated with urban settings (where people earn more on average).

## Conclusions

In this project, we analysed the NICS background checks dataset against data from the U.S. census.

### 2.1.7 Limitations

The big caveat of this analysis is that background checks are only a proxy of gun purchases in America. There are a lot of differences in regulations across the 50 states, and private transactions go under the radar, along with illegal puchases.

We also selected a subset of the census data, in order to facilitate the analysis, there may be left over relevant variables in the census dataset.

As usual with every analysis based on correlation, we can't assume a causal link between one correlated variable and the other. But they can be a starting point for further investigation.

### 2.1.8 Results

The sales seem to be increasing, with a noticeable steady rise since around 2008. Interestingly, sales seem to follow a periodic patterns, with disturbances possibly caused by changes in law or some events.

Gun purchasing, and we may assume, gun ownership vary wildy across the 50 States. The most salient correlations are: - a positive one with the share of veterans in a population. - a negative one with the share of foreign born people. - a negative one with population density. Along with weaker correlations that are trickier to interpret due to the bias caused by, I think, these more relevant variables.

The data is up to 2016-2017, it would be interesting what the trends are for the last 5 years, especially given the many events that happened in the U.S. and worldwide.

[ ]:

```

```