Projet_1_1

August 4, 2022

1 Études des données provennant de l'enquête du National Instant Criminal Check System (NCIS) du FBI sur l'achat et le contrôle des armes à feu.

2 Introduction:

> Le NICS est utilisé pour déterminer si un acheteur potentiel est éligible pour acheter des armes à feu ou explosifs.Les armureries font appel à ce système pour s'assurer que chaque client n'a pas de casier judiciaire ou n'est pas autrement inéligible pour faire un achat.
>
> Les données contiennent le nombre de contrôles d'armes à feu par mois, États et type d'armes.Pour étudier l'ensembles des données provenant de l'enquête mener par le NCIS sur l'achat et le contrôle d'armes à feu nous allons explorer les données du fichier **gun_data.csv**. Notre analyse sera focaliser sur les réponses que l'on apportera aux questions posées dans les lignes qui suives.

3 Questions:

Pour etudier l'ensembles des données; nous tenterons de poser ces questions suivantes; 1. Quelles sont les types d'armes les plus achetés en moyenne ? 2. Quels États ont connu la plus forte croissance dans enregistrements d'armes à feu ? 3. Quelle est la tendance générale des armes à feu ?
> Dans les lignes qui suives nous allons proceder à la préparation des données

4 Préparations des données

> Dans cette phase nous allons importer les modules necessaire pour l'analyse des données,charger les données,inspecter les données et néttoyer les données.

4.0.1 Importations de l'ensemble des paquages et modules requisent pour le projet

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

```
%matplotlib inline
```

Dans cette cellule nous avons importer l'ensembles des modules qui nous serons utile pour la réussite de ce projet

4.0.2 Chargement des données

[3]:
```
#load dataset
df_gun=pd.read_csv('data/gun_data.csv')
```

4.0.3 Inspections des données

Entête

[4]:
```
# entête du dataset
df_gun.head(10)
```

[4]:
```
     month                 state   permit  permit_recheck  handgun  long_gun  \
0  2017-09               Alabama  16717.0             0.0   5734.0    6320.0
1  2017-09                Alaska    209.0             2.0   2320.0    2930.0
2  2017-09               Arizona   5069.0           382.0  11063.0    7946.0
3  2017-09              Arkansas   2935.0           632.0   4347.0    6063.0
4  2017-09            California  57839.0             0.0  37165.0   24581.0
5  2017-09              Colorado   4356.0             0.0  15751.0   13448.0
6  2017-09           Connecticut   4343.0           673.0   4834.0    1993.0
7  2017-09              Delaware    275.0             0.0   1414.0    1538.0
8  2017-09  District of Columbia      1.0             0.0     56.0       4.0
9  2017-09               Florida  10784.0             0.0  39199.0   17949.0

    other  multiple  admin  prepawn_handgun  …  returned_other  \
0   221.0       317    0.0             15.0  …             0.0
1   219.0       160    0.0              5.0  …             0.0
2   920.0       631    0.0             13.0  …             0.0
3   165.0       366   51.0             12.0  …             0.0
4  2984.0         0    0.0              0.0  …             0.0
5  1007.0      1062    0.0              0.0  …             1.0
6   274.0         0    0.0              0.0  …             0.0
7    66.0        68    0.0              0.0  …             0.0
8     0.0         0    0.0              0.0  …             0.0
9  2319.0      1721    1.0             18.0  …             0.0

   rentals_handgun  rentals_long_gun  private_sale_handgun  \
0              0.0               0.0                   9.0
1              0.0               0.0                  17.0
2              0.0               0.0                  38.0
3              0.0               0.0                  13.0
4              0.0               0.0                   0.0
5              0.0               0.0                   0.0
```

```
6              0.0                0.0                  0.0
7              0.0                0.0                 55.0
8              0.0                0.0                  0.0
9              0.0                0.0                 11.0

   private_sale_long_gun  private_sale_other  return_to_seller_handgun  \
0                   16.0                 3.0                       0.0
1                   24.0                 1.0                       0.0
2                   12.0                 2.0                       0.0
3                   23.0                 0.0                       0.0
4                    0.0                 0.0                       0.0
5                    0.0                 0.0                       0.0
6                    0.0                 0.0                       0.0
7                   34.0                 3.0                       1.0
8                    0.0                 0.0                       0.0
9                    9.0                 0.0                       0.0

   return_to_seller_long_gun  return_to_seller_other  totals
0                        0.0                     3.0   32019
1                        0.0                     0.0    6303
2                        0.0                     0.0   28394
3                        2.0                     1.0   17747
4                        0.0                     0.0  123506
5                        0.0                     0.0   35873
6                        0.0                     0.0   12117
7                        2.0                     0.0    3502
8                        0.0                     0.0      61
9                        1.0                     0.0   77390

[10 rows x 27 columns]
```

Ici vous voyez le resumer des 10 premiéres lignes du fichiers **gun_data.csv**

Tailles

[5]:
```
## La tailes,le nombres de lignes et de colonnes
print('Nombre de ligne et de colonne {}'.format(df_gun.shape))
print("Taille du fichier {}".format(df_gun.size))
```

```
Nombre de ligne et de colonne (12485, 27)
Taille du fichier 337095
```

Le dataset comptes en totale 12484 lignes,27 colonnes et une taille de 337095

Les colonnes

[8]:
```
#les colonnes du dataset
df_gun.columns
```

[8]:
```
Index(['month', 'state', 'permit', 'permit_recheck', 'handgun', 'long_gun',
       'other', 'multiple', 'admin', 'prepawn_handgun', 'prepawn_long_gun',
       'prepawn_other', 'redemption_handgun', 'redemption_long_gun',
       'redemption_other', 'returned_handgun', 'returned_long_gun',
       'returned_other', 'rentals_handgun', 'rentals_long_gun',
       'private_sale_handgun', 'private_sale_long_gun', 'private_sale_other',
       'return_to_seller_handgun', 'return_to_seller_long_gun',
       'return_to_seller_other', 'totals'],
      dtype='object')
```

Ici on a l'ensembles des colonnes du dataset

Typages des données

[9]:
```
#les differentes types des colonnes
df_gun.dtypes
```

[9]:
```
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
```

Vous voyez les differentes types de l'ensembles des variables du dataset; On a des entiers **(int64)**, des flottants **(float64)** et des chaines de caractéres **(object)**

Details sur les données

[10]:
```
# les details avec la fontions info() de Pandas
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  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
```

Cette cellules vous montre en details les differentes colonnes du dataset, le nombre de valeurs non-null et le types de chaque colonne. Parexemple ici la colonne **prepawn_handgun** est de type float et à **10542** valeurs non-null

Description ou statistique sur les données

[11]:
```
#statistique descriptives
df_gun.describe()
```

[11]:

```
              permit  permit_recheck        handgun       long_gun  \
count   12461.000000     1100.000000   12465.000000   12466.000000
mean     6413.629404     1165.956364    5940.881107    7810.847585
std     23752.338269     9224.200609    8618.584060    9309.846140
min         0.000000        0.000000       0.000000       0.000000
25%         0.000000        0.000000     865.000000    2078.250000
50%       518.000000        0.000000    3059.000000    5122.000000
75%      4272.000000        0.000000    7280.000000   10380.750000
max    522188.000000   116681.000000  107224.000000  108058.000000

              other      multiple         admin  prepawn_handgun  \
count   5500.000000  12485.000000  12462.000000     10542.000000
mean     360.471636    268.603364     58.898090         4.828021
std     1349.478273    783.185073    604.814818        10.907756
min        0.000000      0.000000      0.000000         0.000000
25%       17.000000     15.000000      0.000000         0.000000
50%      121.000000    125.000000      0.000000         0.000000
75%      354.000000    301.000000      0.000000         5.000000
max    77929.000000  38907.000000  28083.000000       164.000000

       prepawn_long_gun  prepawn_other  …  returned_other  rentals_handgun  \
count      10540.000000    5115.000000  …     1815.000000       990.000000
mean           7.834156       0.165591  …        1.027548         0.076768
std           16.468028       1.057105  …        4.386296         0.634503
min            0.000000       0.000000  …        0.000000         0.000000
25%            0.000000       0.000000  …        0.000000         0.000000
50%            1.000000       0.000000  …        0.000000         0.000000
75%            8.000000       0.000000  …        0.000000         0.000000
max          269.000000      49.000000  …       64.000000        12.000000

       rentals_long_gun  private_sale_handgun  private_sale_long_gun  \
count        825.000000           2750.000000            2750.000000
mean           0.087273             14.936000              11.602909
std            0.671649             71.216021              54.253090
min            0.000000              0.000000               0.000000
25%            0.000000              0.000000               0.000000
50%            0.000000              0.000000               0.000000
75%            0.000000              2.000000               4.000000
max           12.000000           1017.000000             777.000000

       private_sale_other  return_to_seller_handgun  \
count         2750.000000               2475.000000
mean             1.030182                  0.402020
std              4.467843                  1.446568
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                2750.000000             2255.000000   12485.000000
mean                    0.441818                0.105987   21595.725911
std                     1.528223                0.427363   32591.418387
min                     0.000000                0.000000       0.000000
25%                     0.000000                0.000000    4638.000000
50%                     0.000000                0.000000   12399.000000
75%                     0.000000                0.000000   25453.000000
max                    17.000000                4.000000  541978.000000

[8 rows x 25 columns]
```

On a les statistique descriptive comme le moyenne, la mediane, l'ecartype etc, des colonnes de types numeriques

4.0.4 Nettoyages des données

Check des valeurs manquantes

[13]:
```python
#check des valeurs manquantes
print(df_gun.isna().sum())
```

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

```
return_to_seller_long_gun     9735
return_to_seller_other       10230
totals                           0
dtype: int64
```

Oups! On constate que l'on a beaucoup de valeur manquante parexemple la colone **permit_recheck** a 11385 valeur manquantes. On va proceder par la suppression des valeurs manquantes

[15]:
```
# je copy le dataframe df_gun dans df_gun_cp
df_gun_cp=df_gun.copy()
df_gun_cp.head(8)
```

[15]:
```
     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
5  2017-09     Colorado   4356.0             0.0  15751.0   13448.0  1007.0
6  2017-09  Connecticut   4343.0           673.0   4834.0    1993.0   274.0
7  2017-09     Delaware    275.0             0.0   1414.0    1538.0    66.0

   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
5      1062    0.0              0.0  …             1.0              0.0
6         0    0.0              0.0  …             0.0              0.0
7        68    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
5               0.0                   0.0                    0.0
6               0.0                   0.0                    0.0
7               0.0                  55.0                   34.0

   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
5                  0.0                        0.0                        0.0
6                  0.0                        0.0                        0.0
7                  3.0                        1.0                        2.0

   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                     0.0   35873
6                     0.0   12117
7                     0.0    3502

[8 rows x 27 columns]
```

Cette copie nous permettra de pouvoir la comparaison des deux dataframes apres le néttoyages des données

Suppression des valeurs manquantes

[16]:
```
# suppression des valeurs manquantes
df_gun_cp.dropna(inplace=True)
sum(df_gun_cp.isnull().sum())
```

[16]:
```
0
```

Ici on vient de supprimer toutes les valeurs manquantes

Checks des valeus dupliquées

[17]:
```
# Valeur dupliqué
df_gun_cp.duplicated().sum()
```

[17]:
```
0
```

Le dataset n'as pas de valeurs dupliquées

Checks des valeurs uniques

[18]:
```
# Valeurs unique
df_gun_cp.nunique()
```

[18]:
```
month                   14
state                   55
permit                 655
permit_recheck         140
handgun                721
```

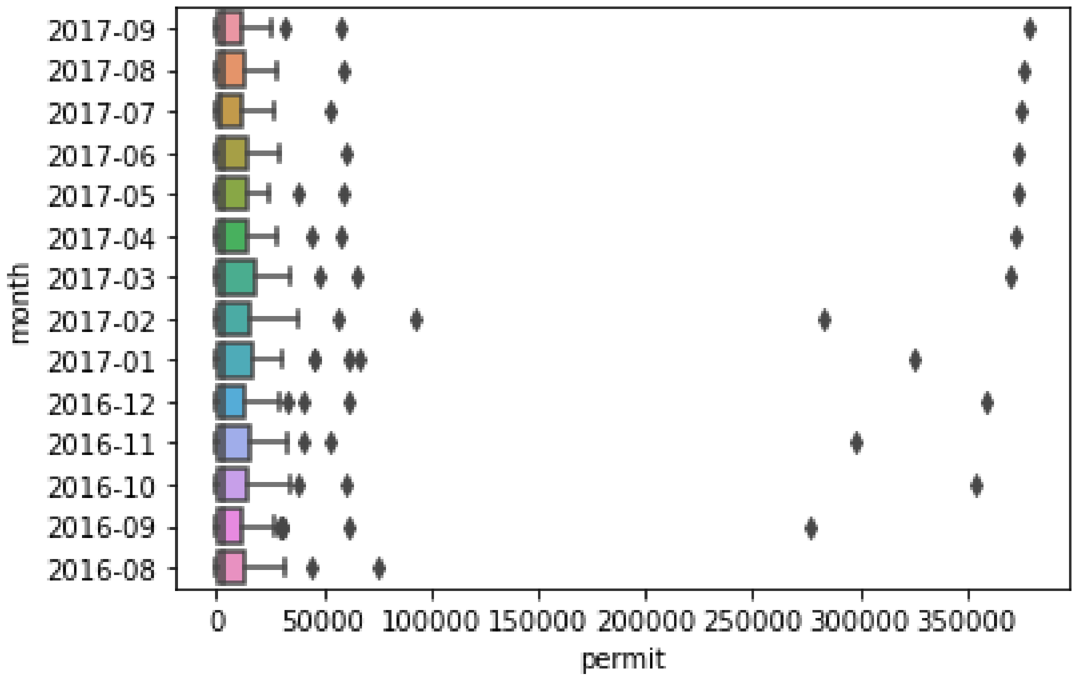
```
long_gun                     703
other                        503
multiple                     427
admin                         54
prepawn_handgun               38
prepawn_long_gun              33
prepawn_other                  7
redemption_handgun           427
redemption_long_gun          446
redemption_other              40
returned_handgun             172
returned_long_gun             85
returned_other                27
rentals_handgun                8
rentals_long_gun               8
private_sale_handgun          97
private_sale_long_gun         90
private_sale_other            38
return_to_seller_handgun      14
return_to_seller_long_gun     13
return_to_seller_other         5
totals                       763
dtype: int64
```

Checks des valeurs abérrentes:

Ici on va utilisé le **Diagramme en moustâche** avec **seaborn.boxplot**. J'ai utilisé le Boxplot pour detecter les valeurs aberrantes dans l'ensemble des données et de pouvoir supprimé les valeurs extrêmes afin de faciliter l'exploration de l'analyse des données.Les graphes suivantes de resumés les variables de maniére simple et visuel, d'identifier les valeurs aberrentes et de comprendre la repartition des observations.

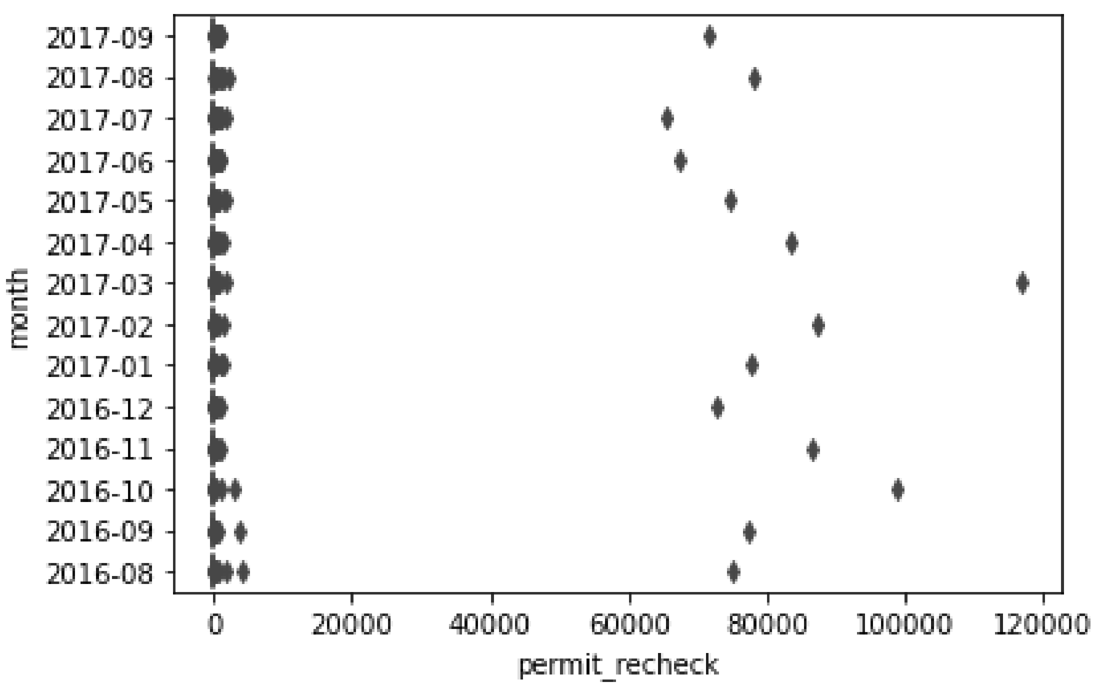
[19]:
```
def var_extreme(arg):
    sns.boxplot(x=arg,y='month',data=df_gun_cp)
```

[20]:
```
var_extreme('permit')
```



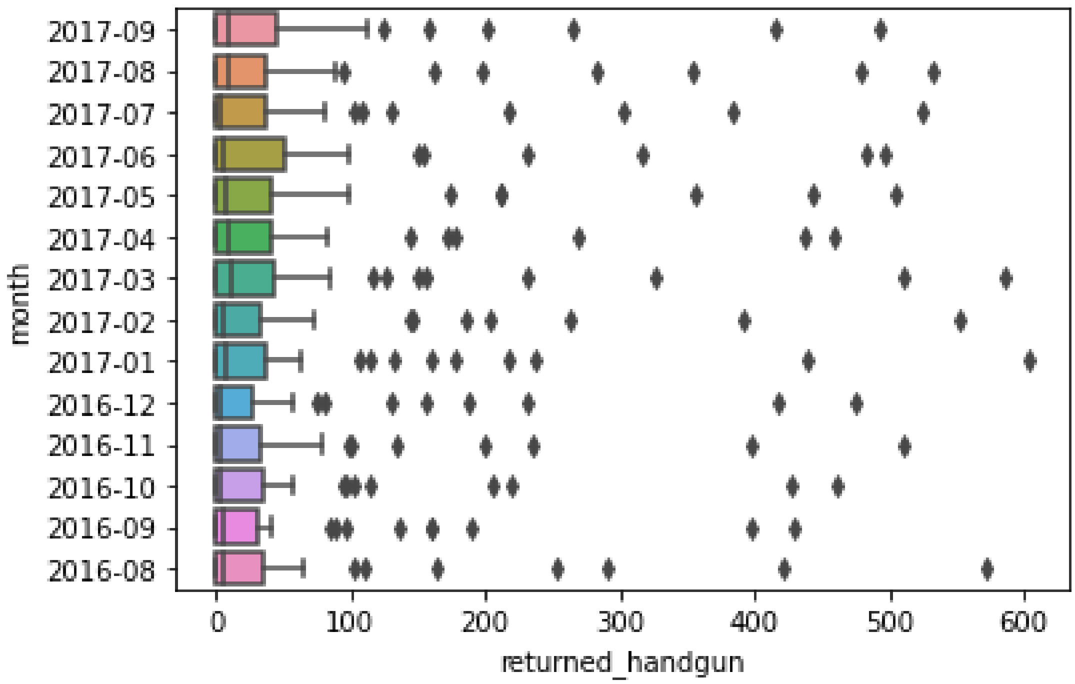
Repartition du **permit** en fonction du mois **month**

[21]:
```
var_extreme('permit_recheck')
```

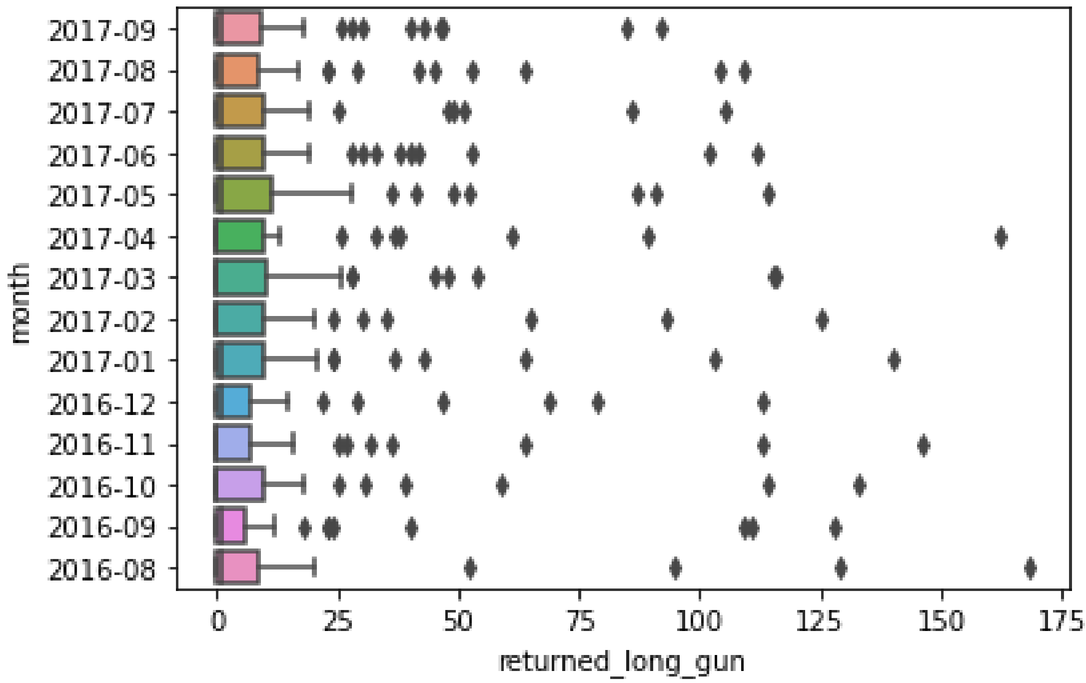


Repartition du **permit_recheck** en fonction de **month**

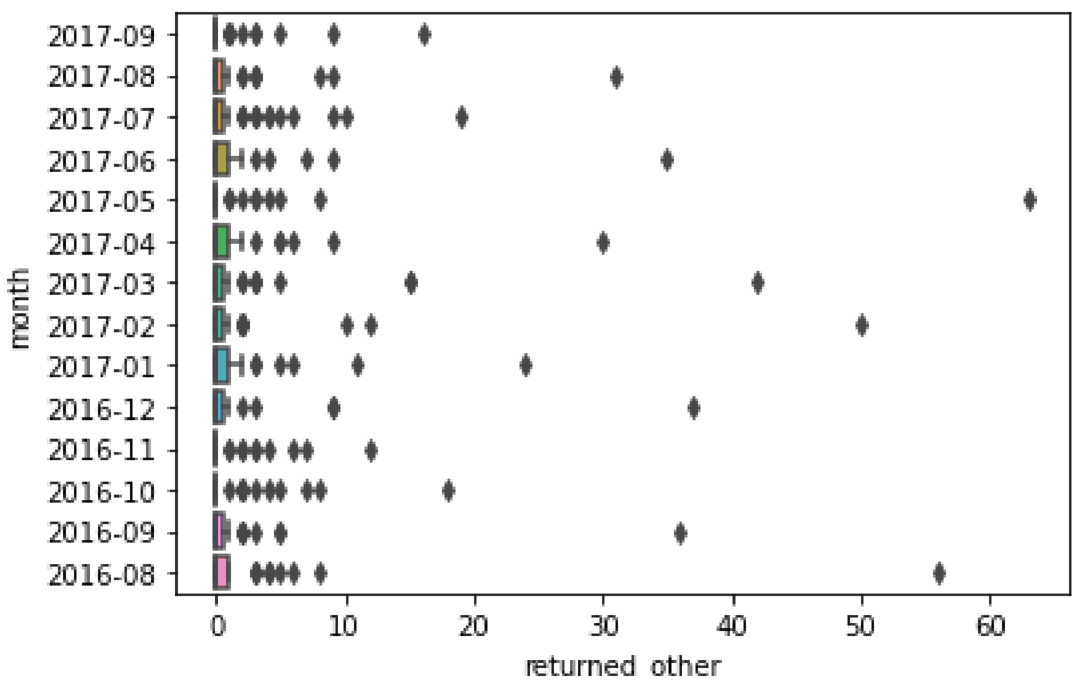
[22]:
```
var_extreme('returned_handgun')
```



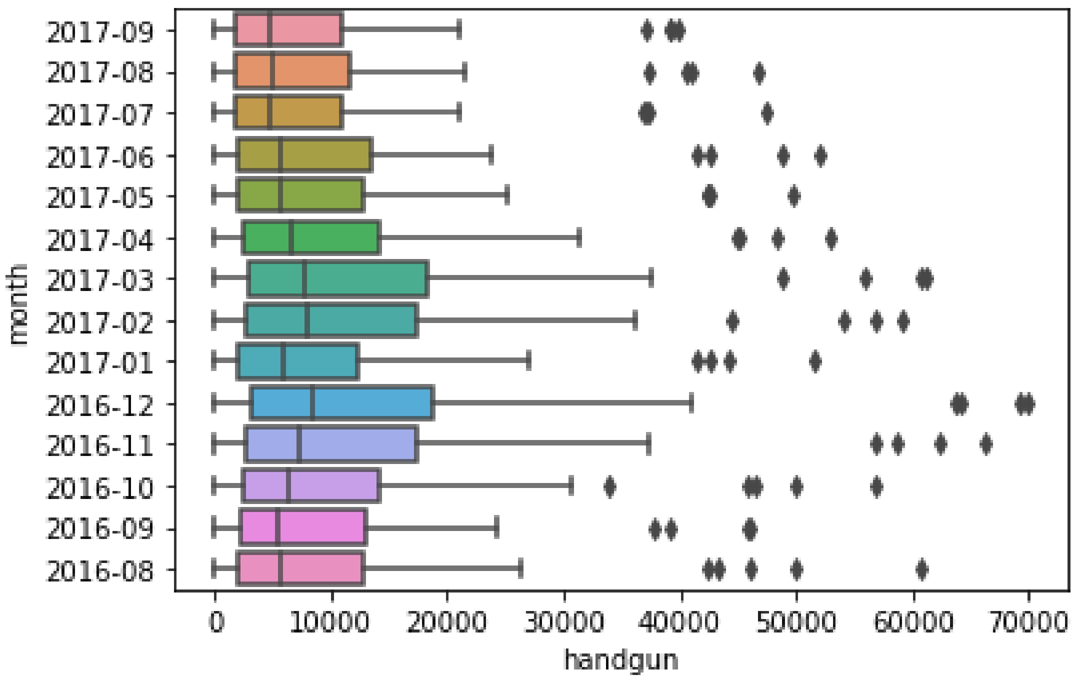
[23]:
```
var_extreme('returned_long_gun')
```



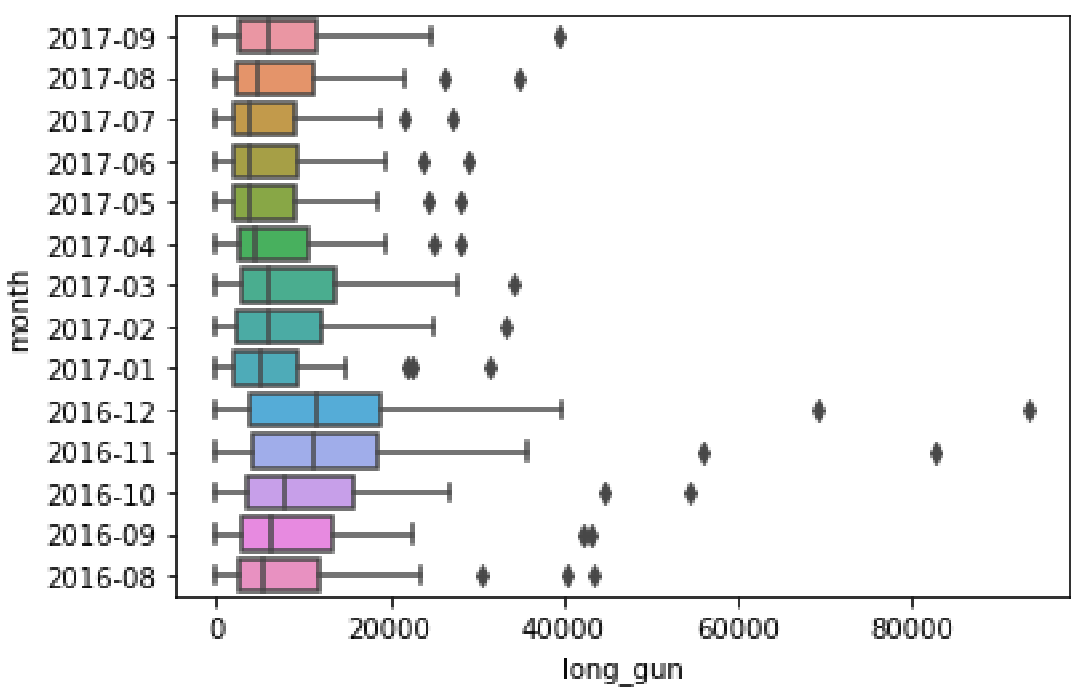

```
[24]: var_extreme('returned_other')
```



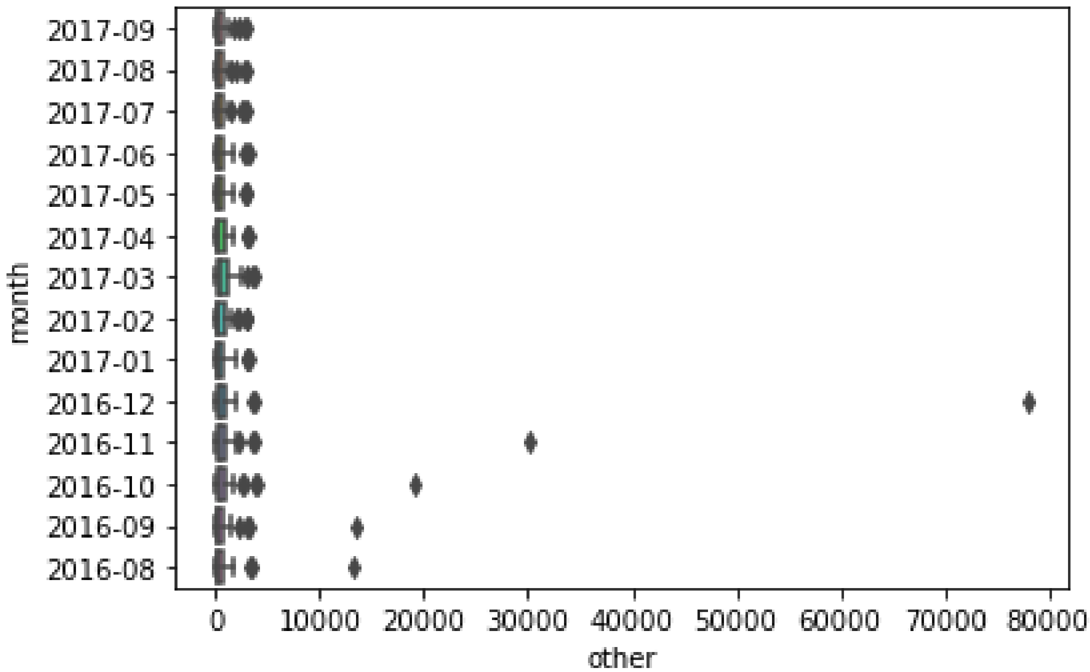
[25]:
```
var_extreme('handgun')
```



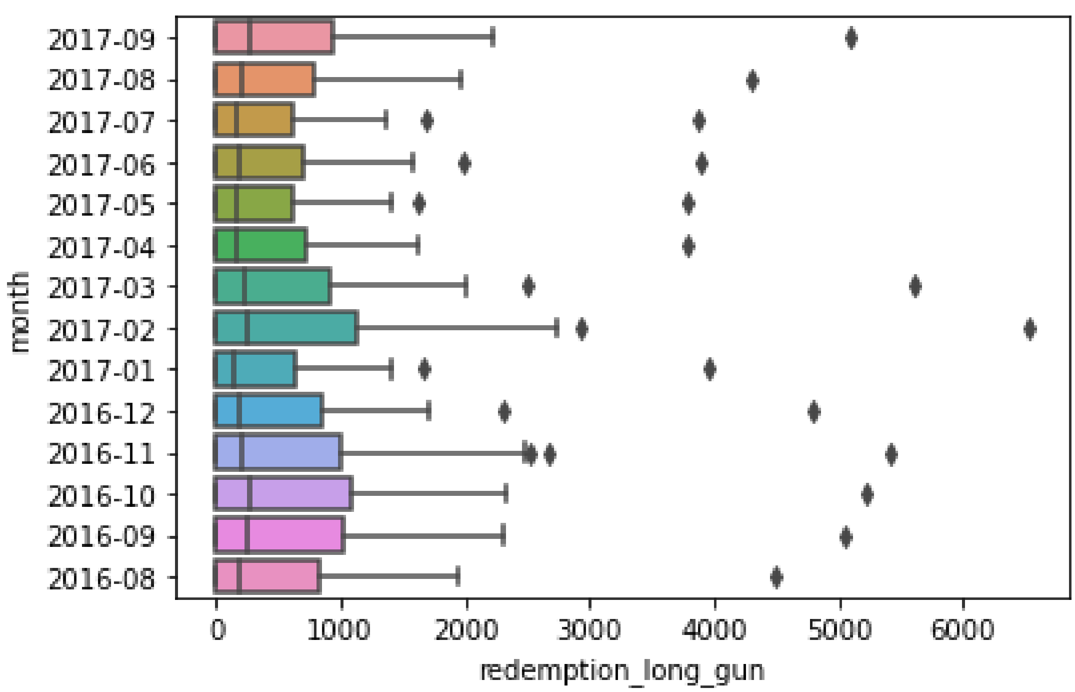
[26]:
```
var_extreme('long_gun')
```



[27]:
```
var_extreme('other')
```

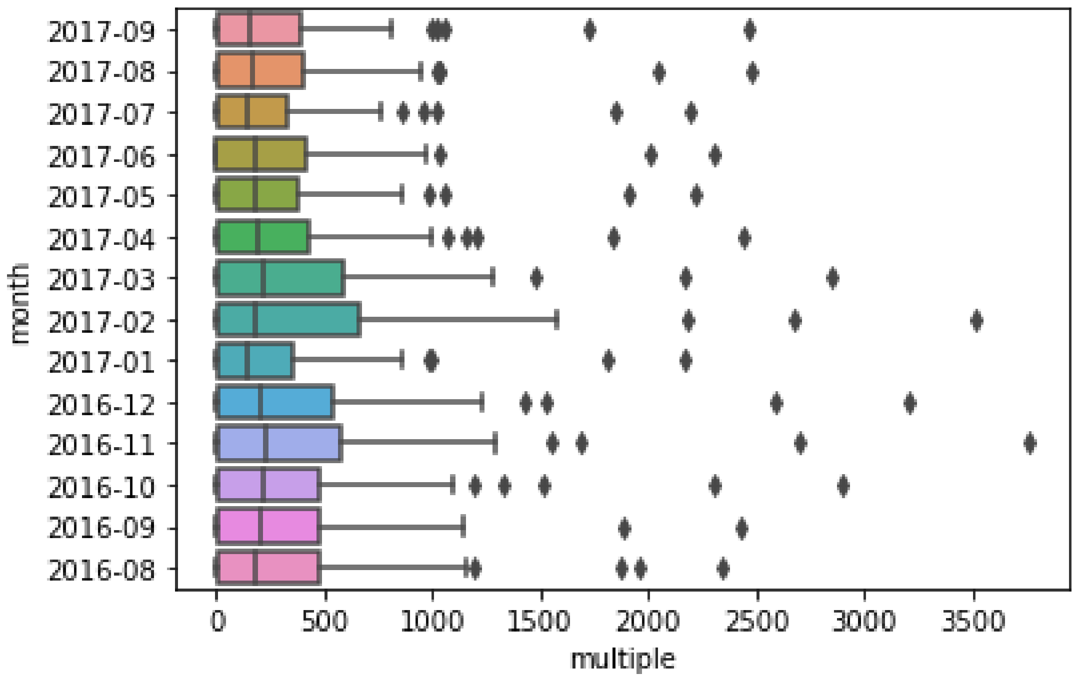


[28]:
```
var_extreme('redemption_long_gun')
```



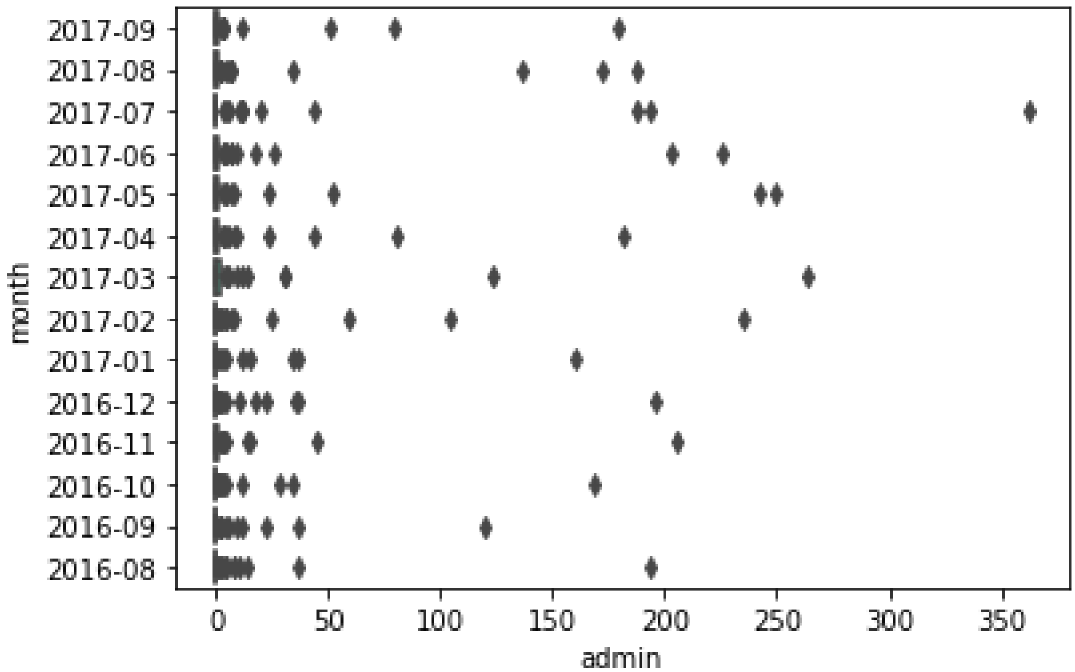
[29]:

```
var_extreme('multiple')
```

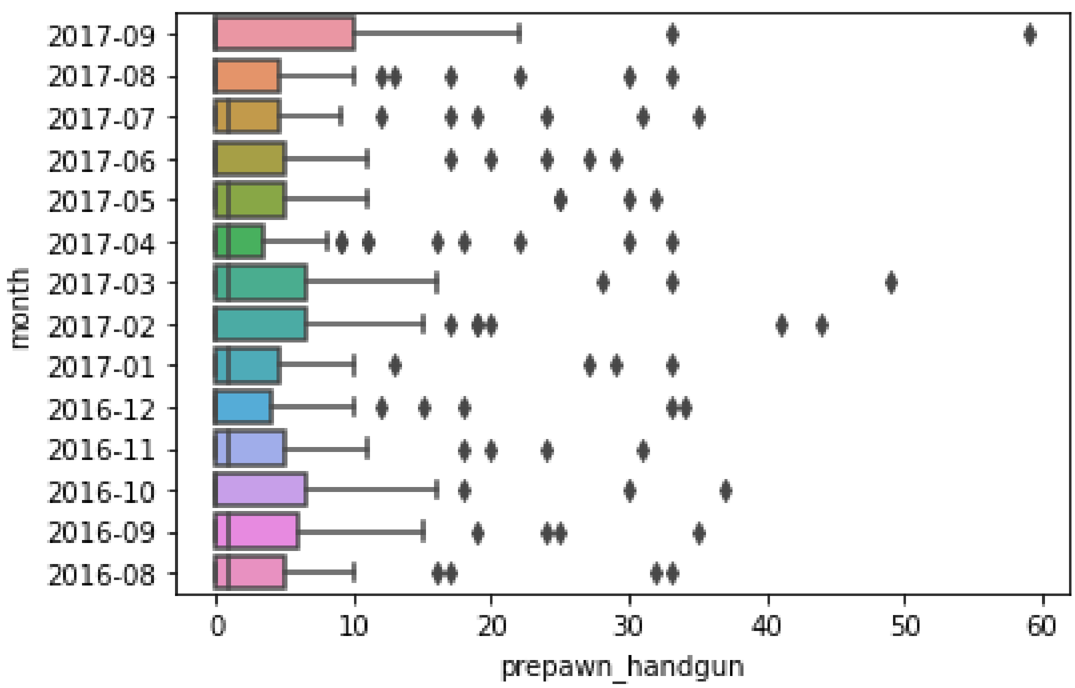


[30]:

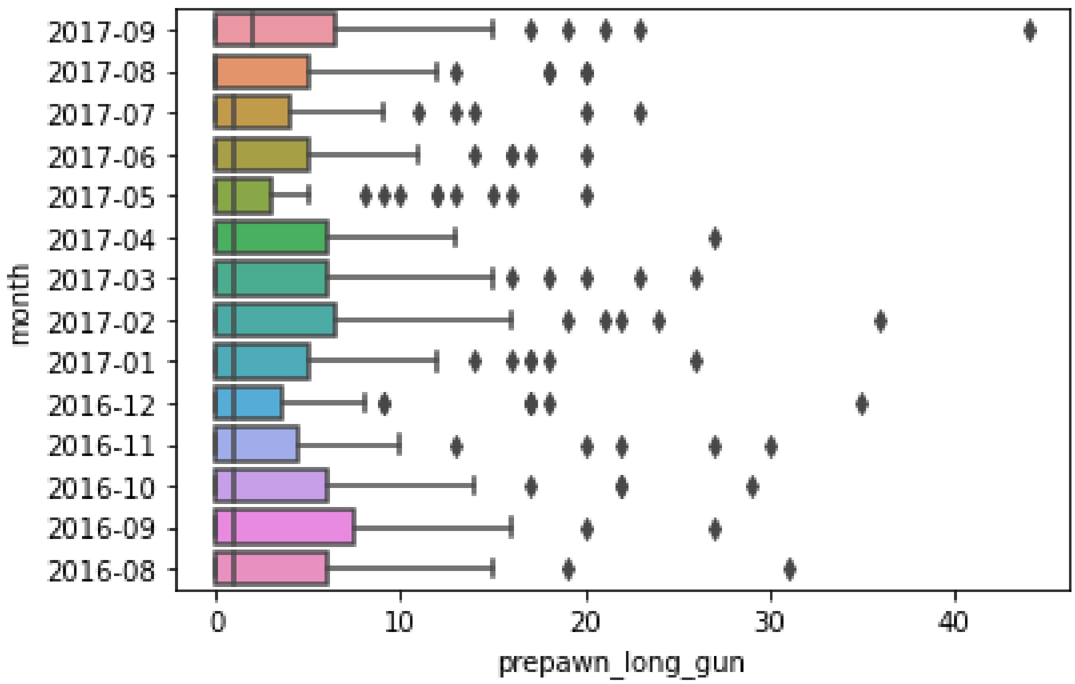
```
var_extreme('admin')
```



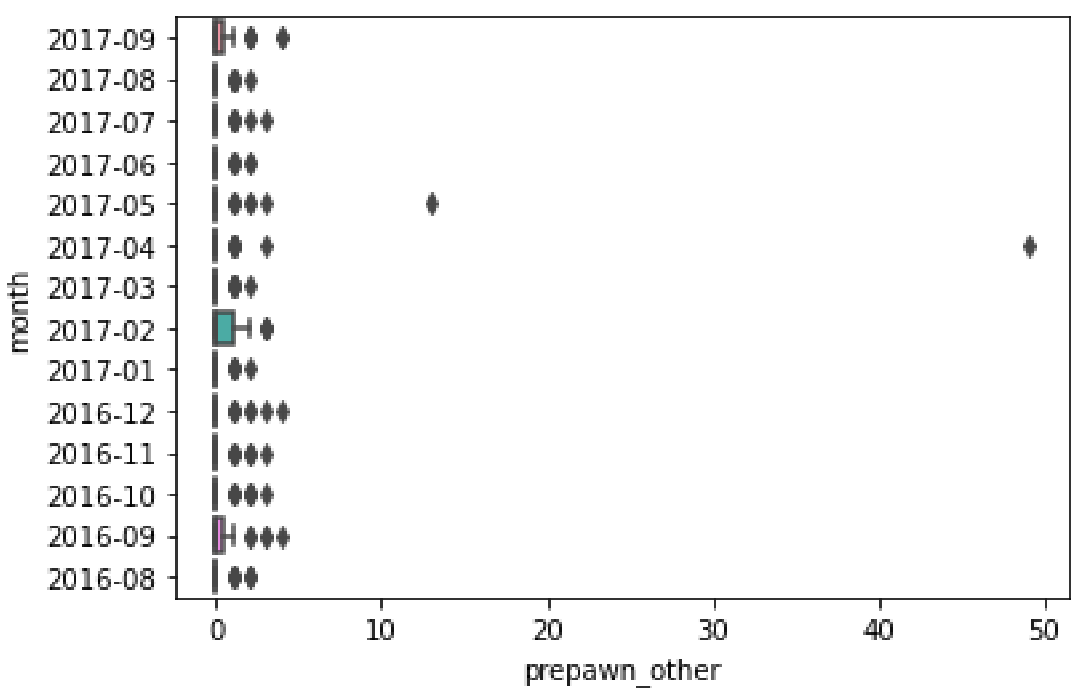
[31]:
```
var_extreme('prepawn_handgun')
```



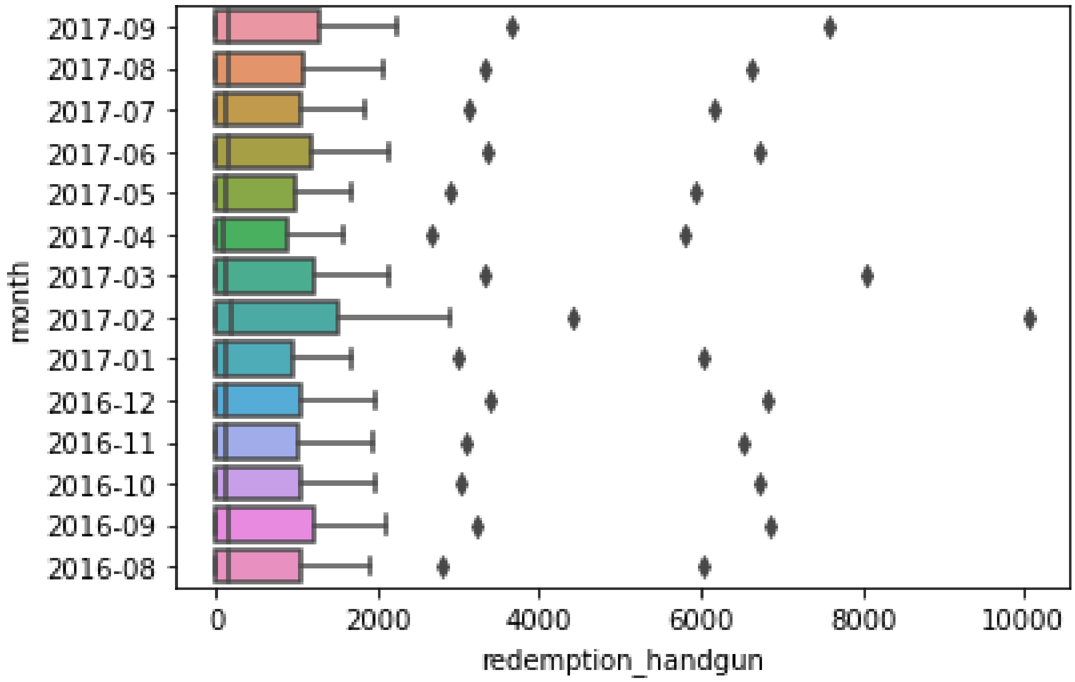
[32]:
```
var_extreme('prepawn_long_gun')
```



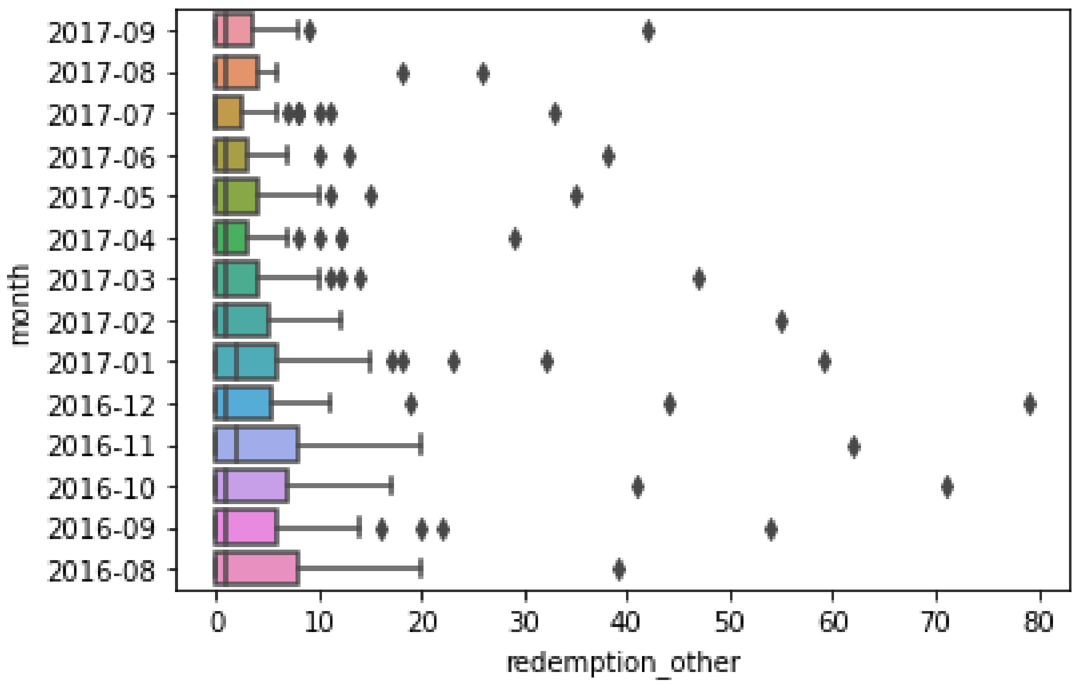
[33]:
```
var_extreme('prepawn_other')
```



[34]:
```
var_extreme('redemption_handgun')
```

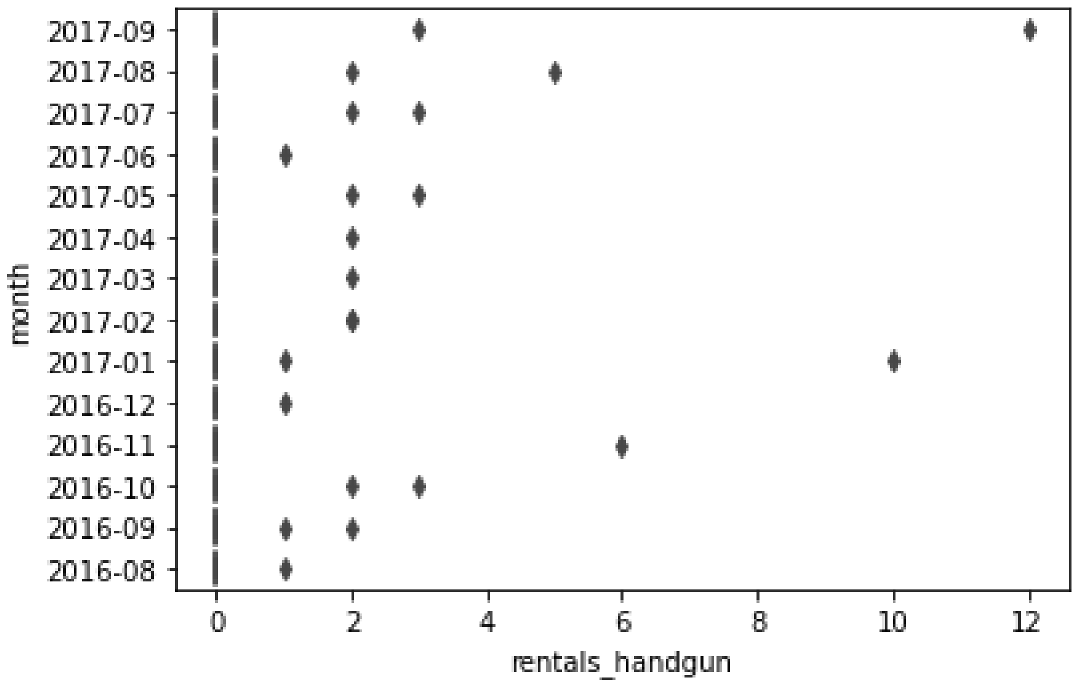


[35]:
```
var_extreme('redemption_other')
```



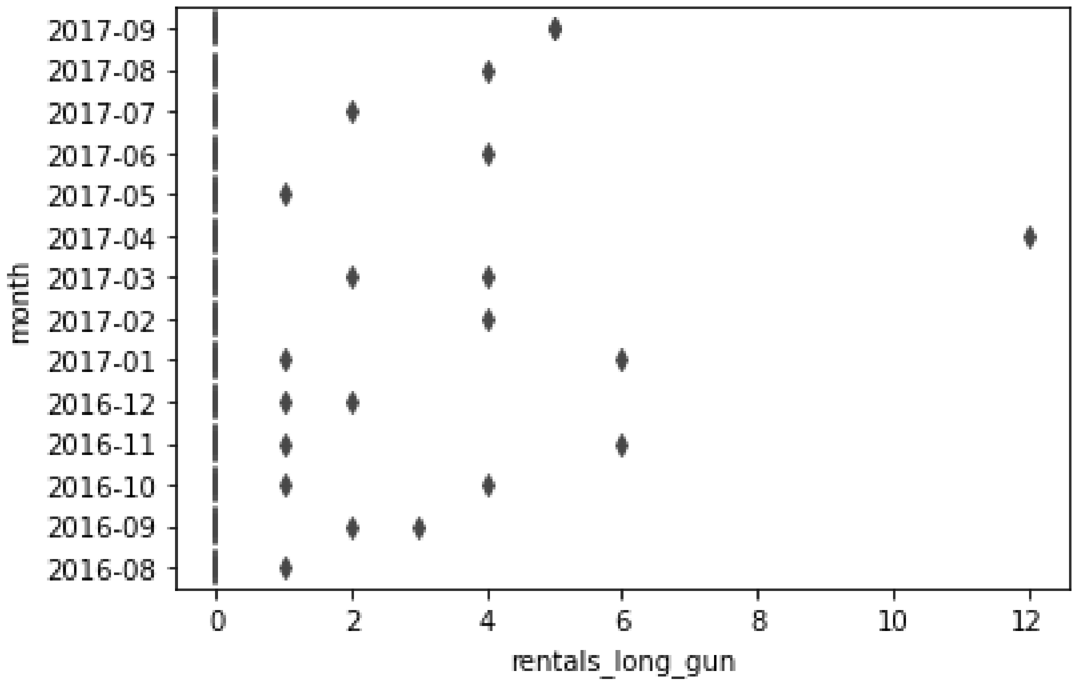
[36]:

```
var_extreme('rentals_handgun')
```

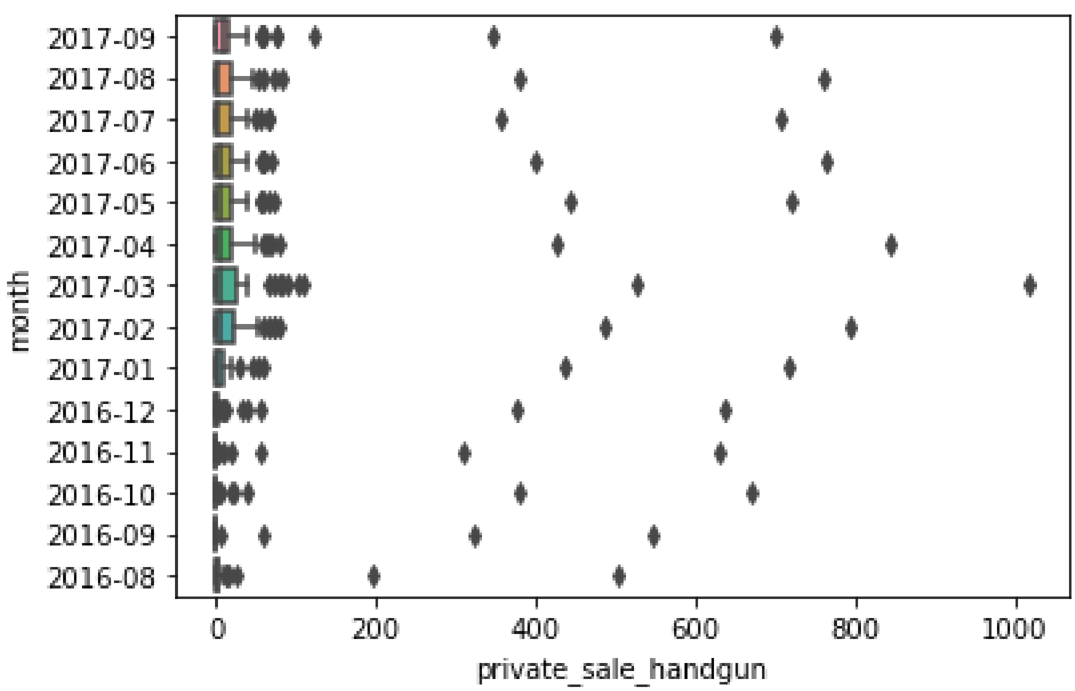


[37]:

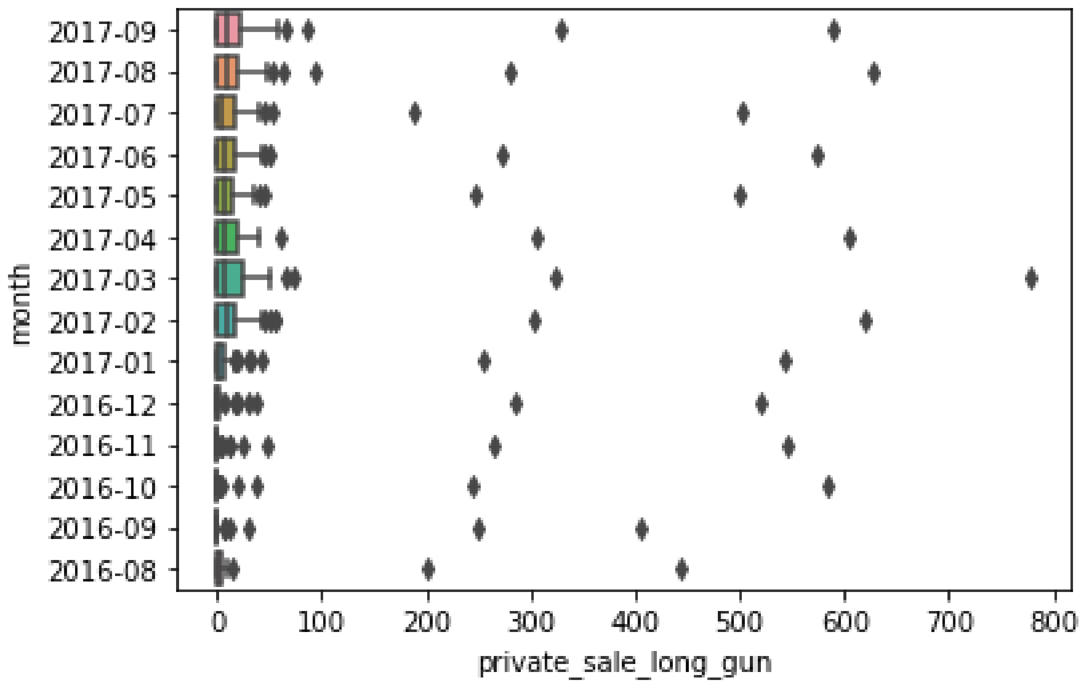
```
var_extreme('rentals_long_gun')
```



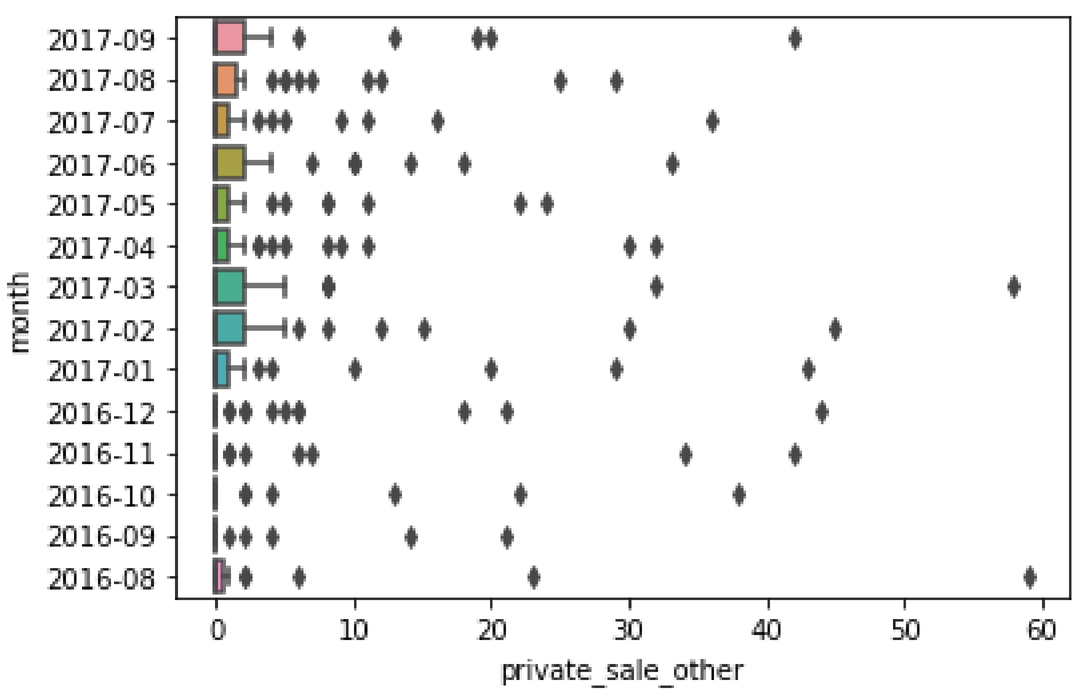
[38]:
```
var_extreme('private_sale_handgun')
```



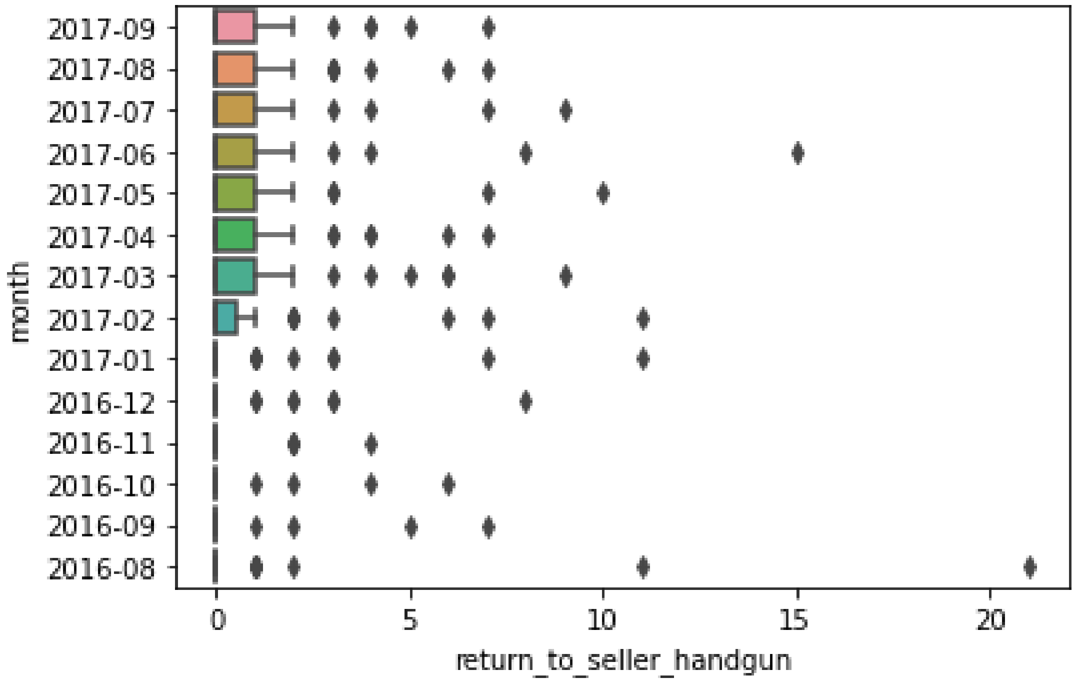
```
[39]: var_extreme('private_sale_long_gun')
```



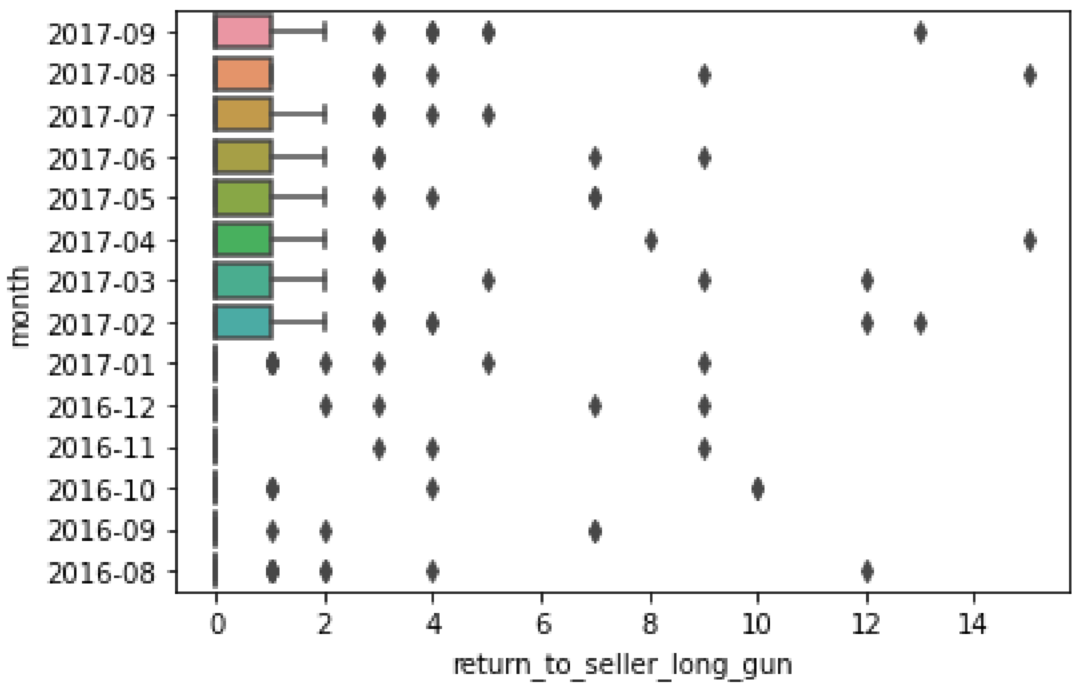

```
[40]: var_extreme('private_sale_other')
```



[41]:
```
var_extreme('return_to_seller_handgun')
```

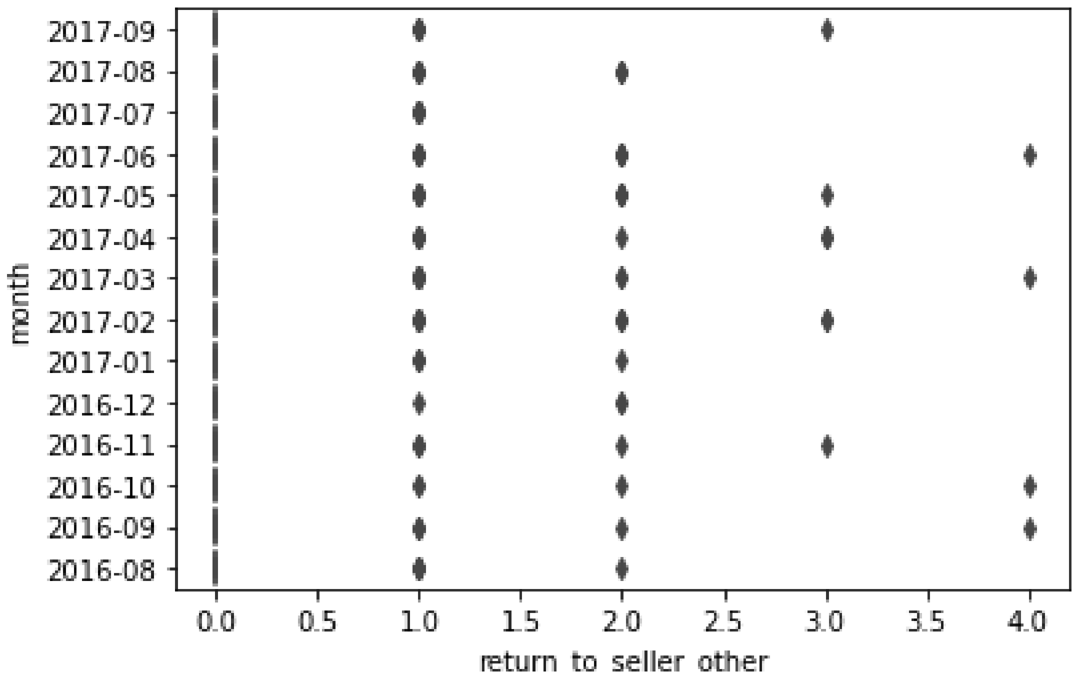


[42]:
```
var_extreme('return_to_seller_long_gun')
```



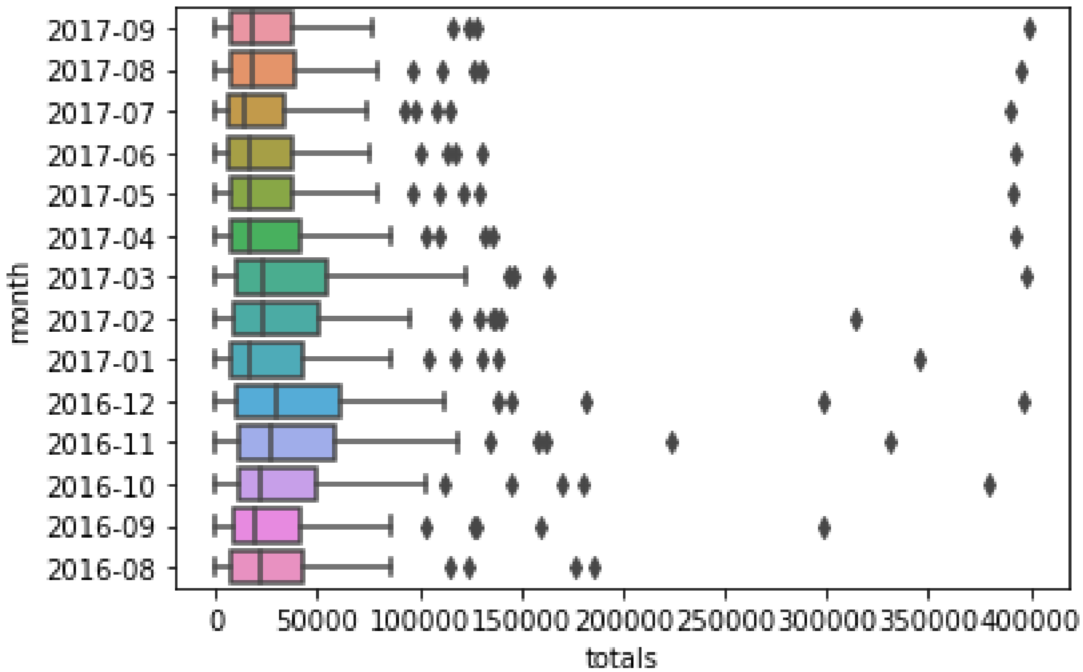
[43]:

```
var_extreme('return_to_seller_other')
```



[44]:

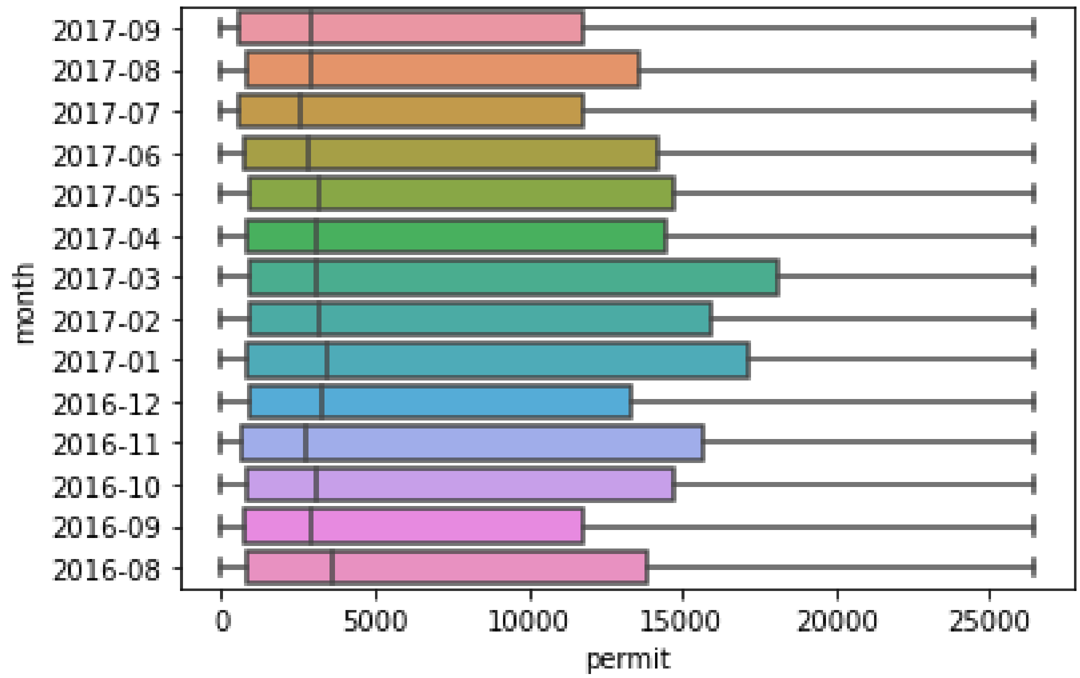
```
var_extreme('totals')
```



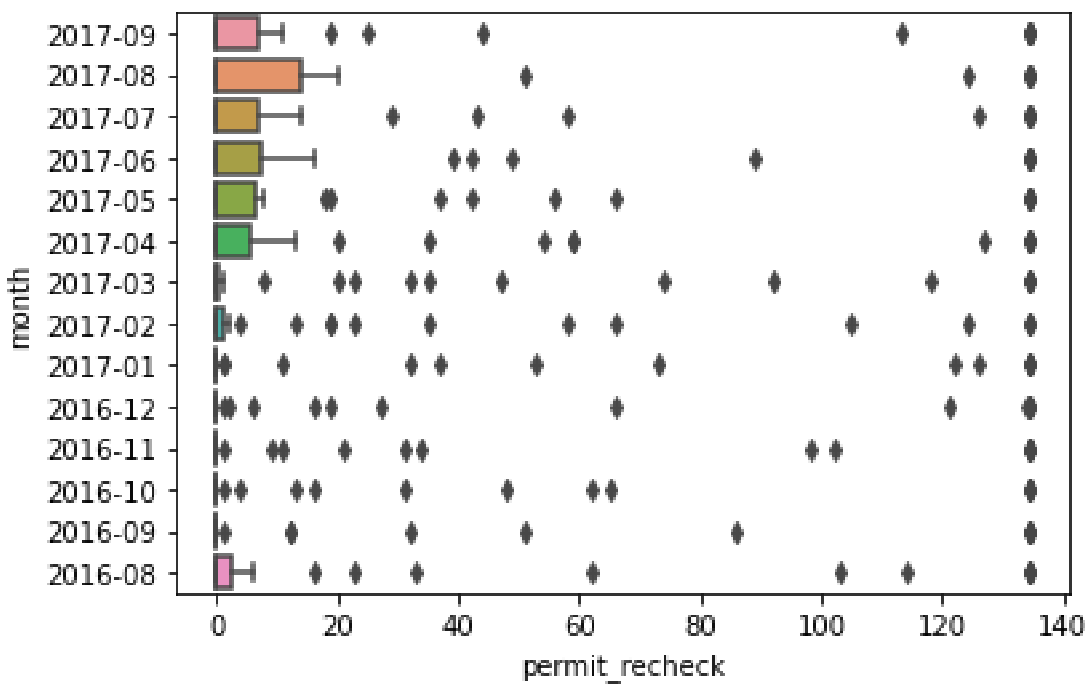
Utilisation de la methode des quantiles pour supprimer les valeurs extrêmes

[45]:
```
# methode des quantiles
def methode_quartiles(col):
    g10 =df_gun_cp[col].quantile(0.10)
    g90 =df_gun_cp[col].quantile(0.90)
    df_gun_cp[col] = np.where(df_gun_cp[col]< g10, g10, df_gun_cp[col])
    df_gun_cp[col] = np.where(df_gun_cp[col]> g90, g90, df_gun_cp[col])
    sns.boxplot(x=col,y='month', data=df_gun_cp)
    plt.show()
```

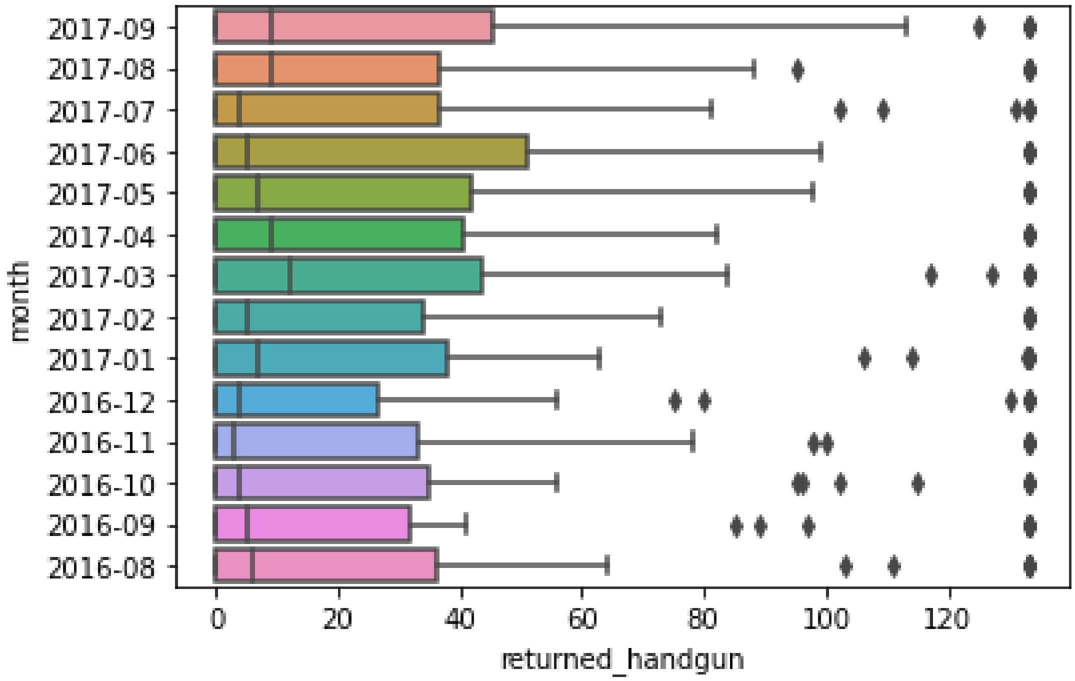
[46]:
```
methode_quartiles('permit')
```



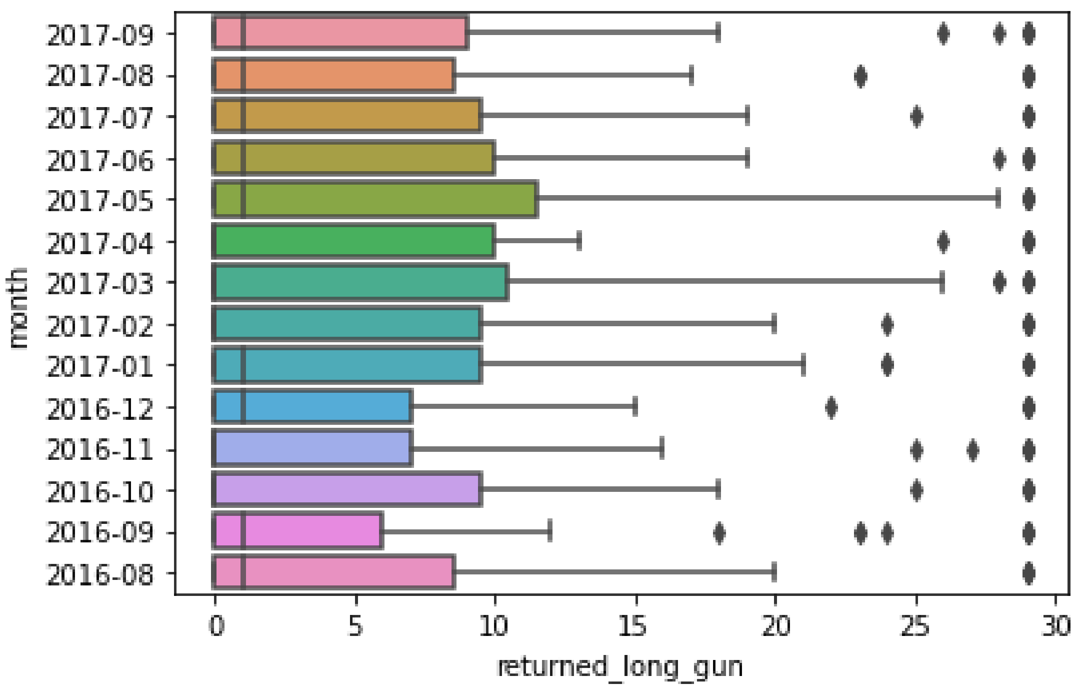
[47]:
```
methode_quartiles('permit_recheck')
```



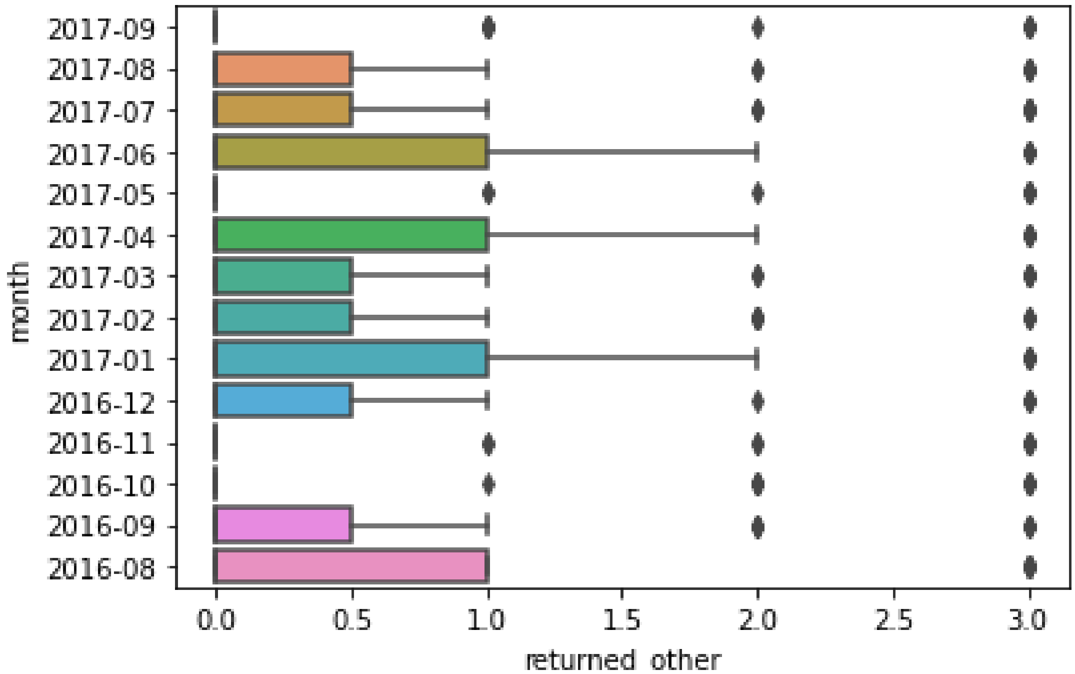
[48]:
```
methode_quartiles('returned_handgun')
```



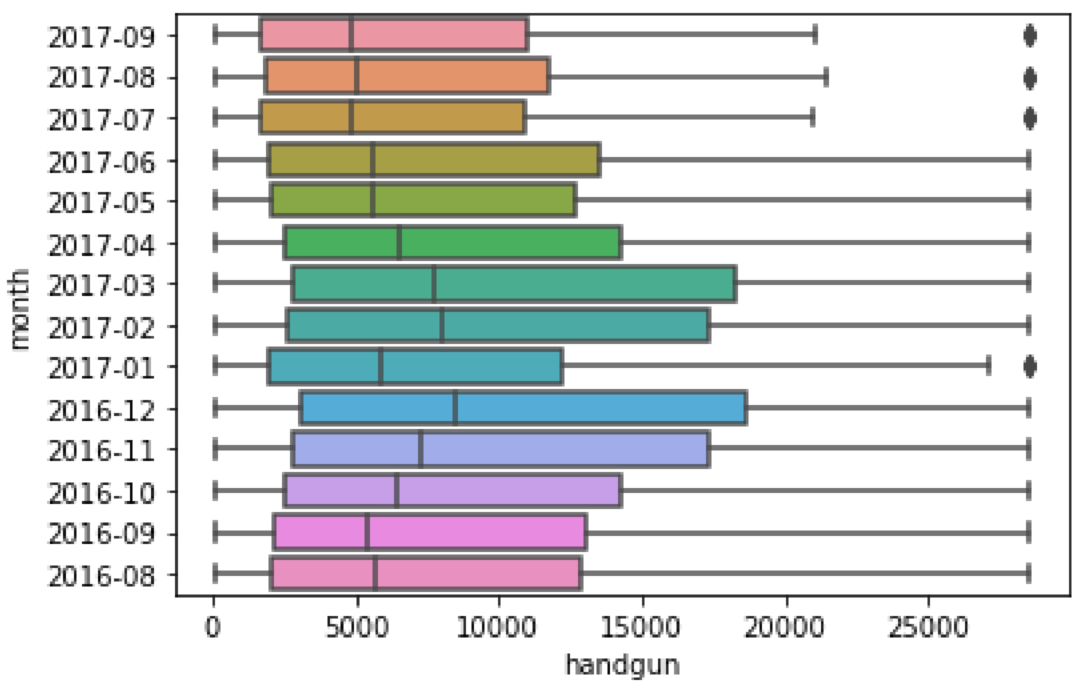
[49]:
```
methode_quartiles('returned_long_gun')
```



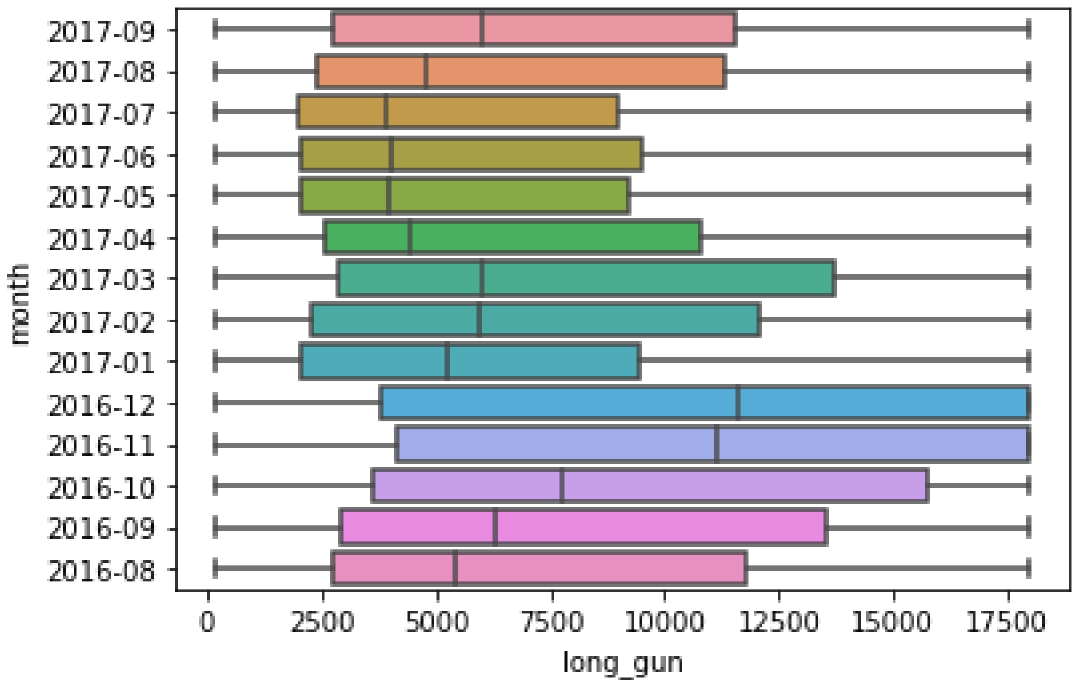
```
[50]: methode_quartiles('returned_other')
```



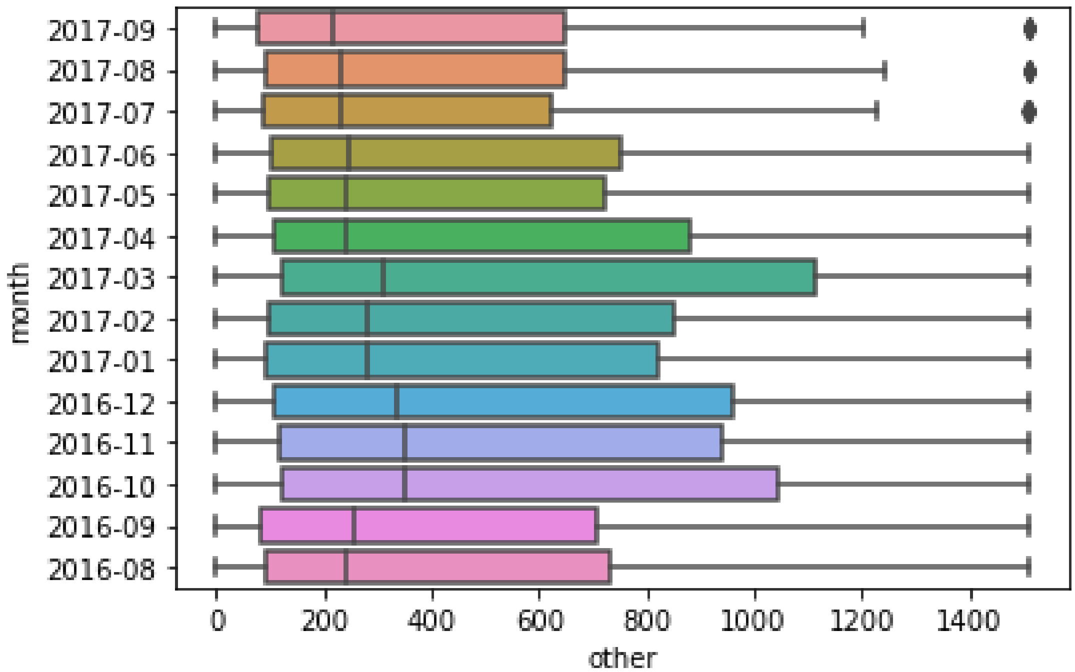

```
[51]: methode_quartiles('handgun')
```



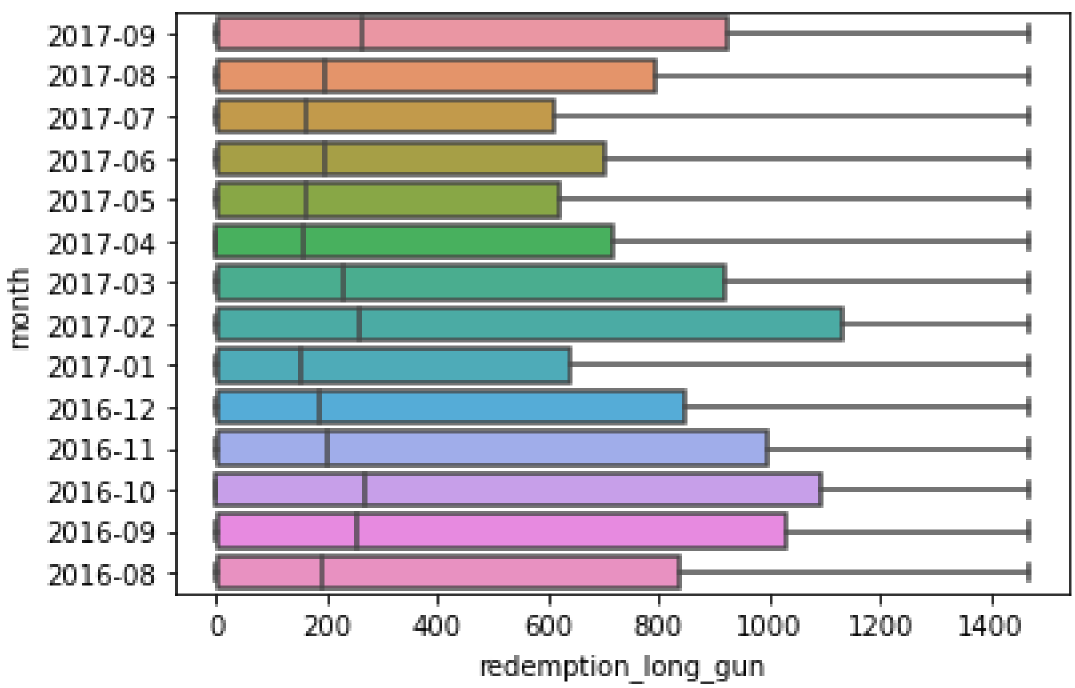
[52]:
```
methode_quartiles('long_gun')
```



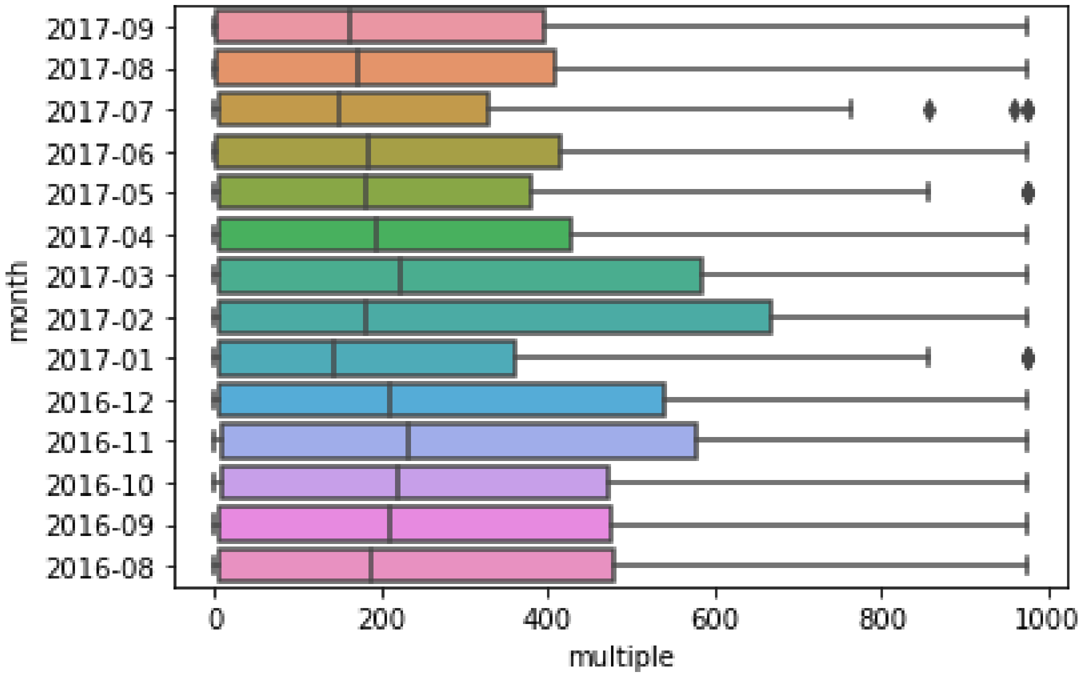
[53]:
```
methode_quartiles('other')
```



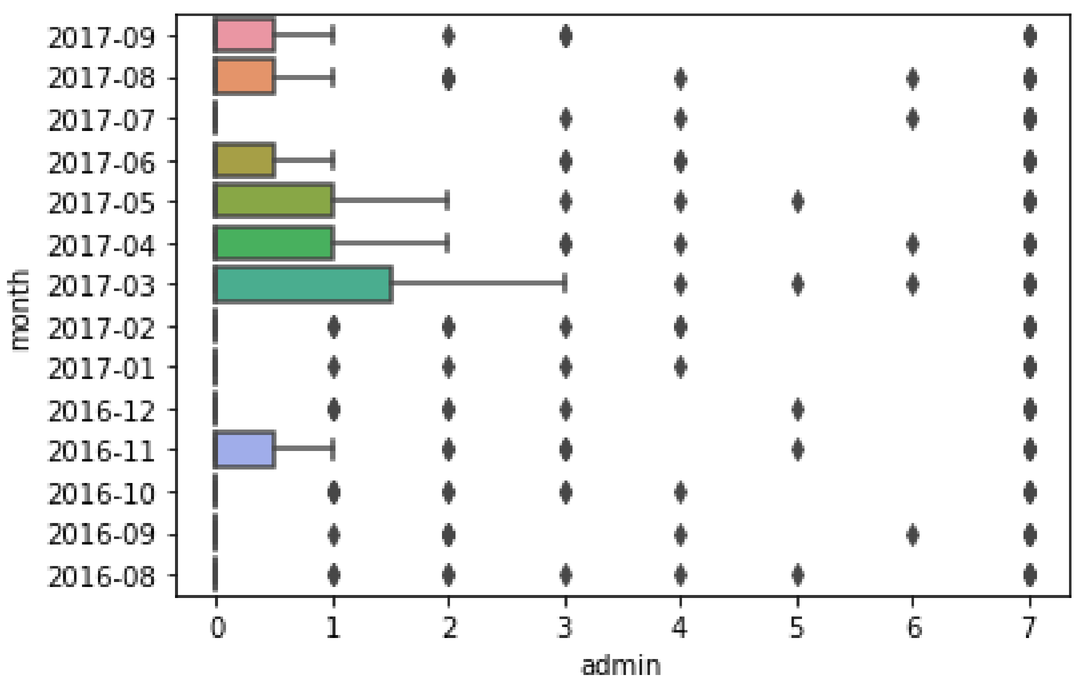
[54]:
```
methode_quartiles('redemption_long_gun')
```



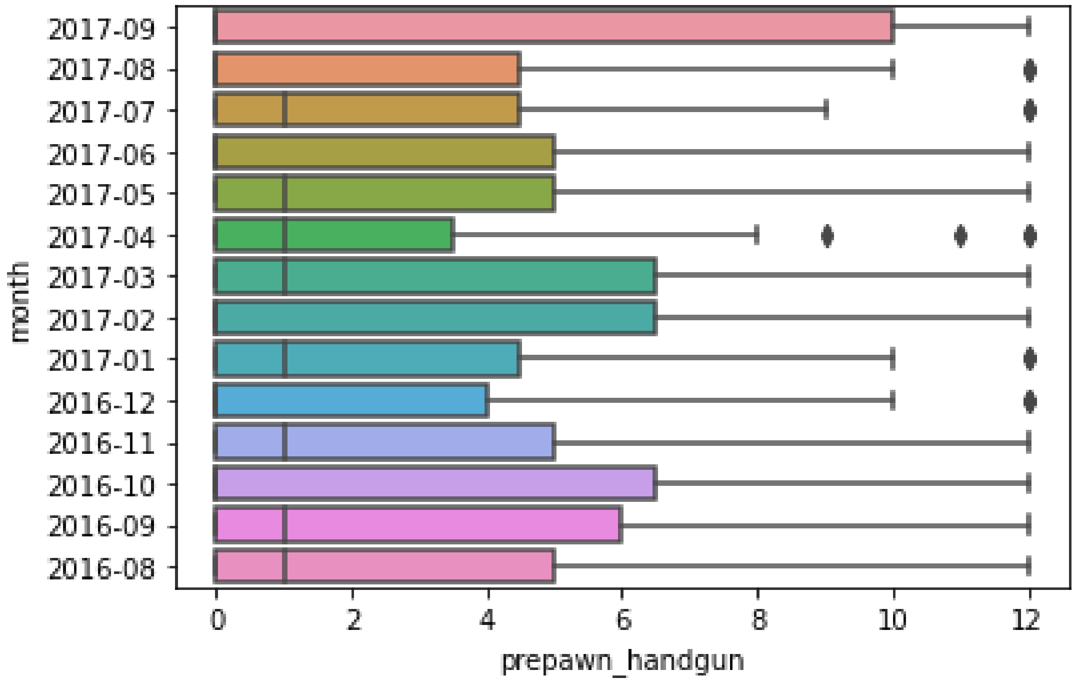
[55]:
```
methode_quartiles('multiple')
```



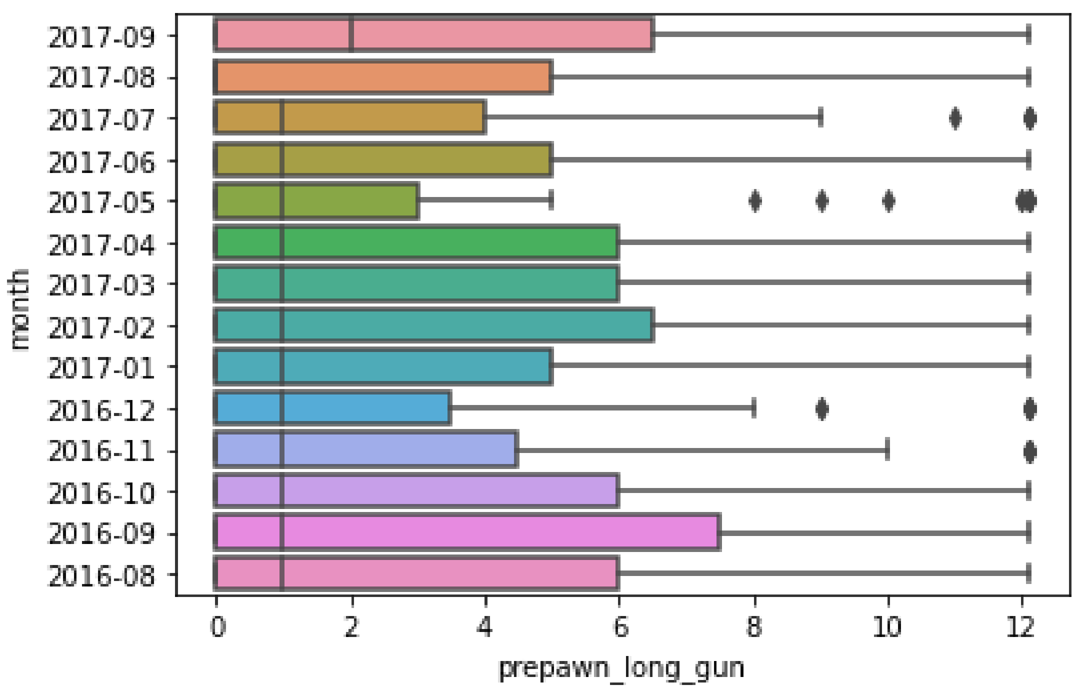
[56]:
```
methode_quartiles('admin')
```



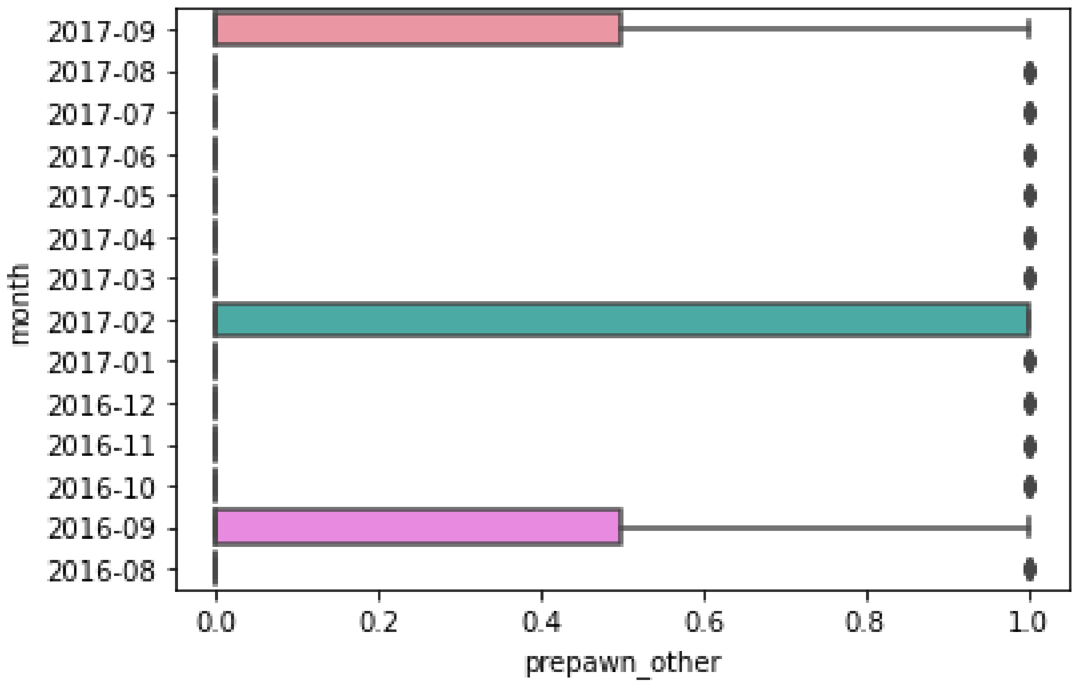
```
[57]: methode_quartiles('prepawn_handgun')
```



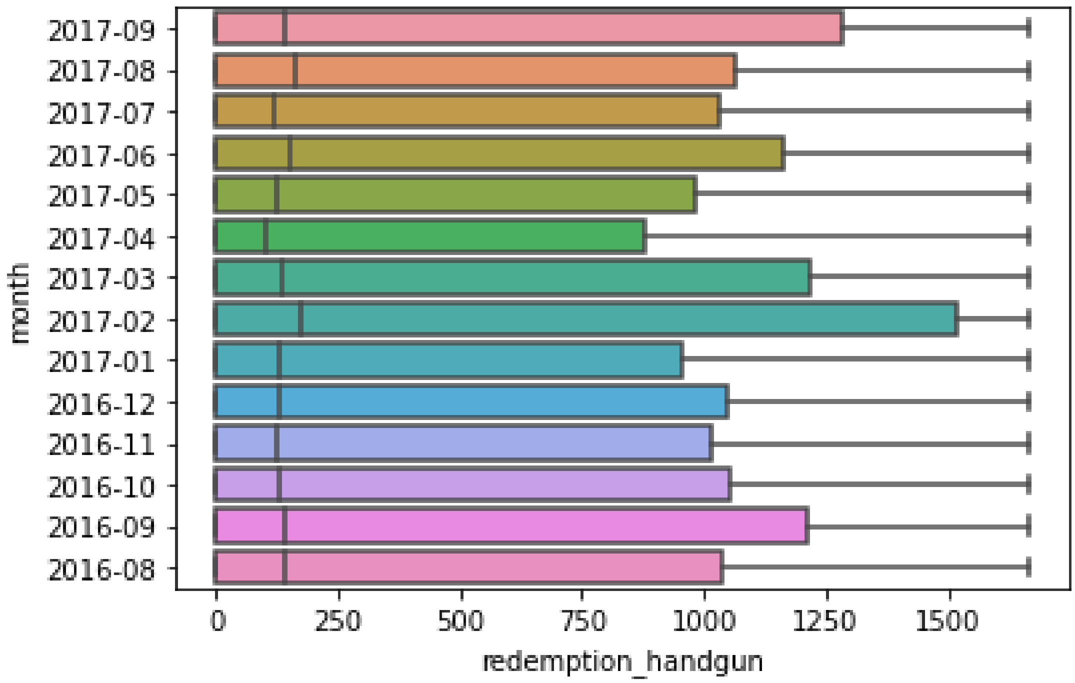

```
[58]: methode_quartiles('prepawn_long_gun')
```



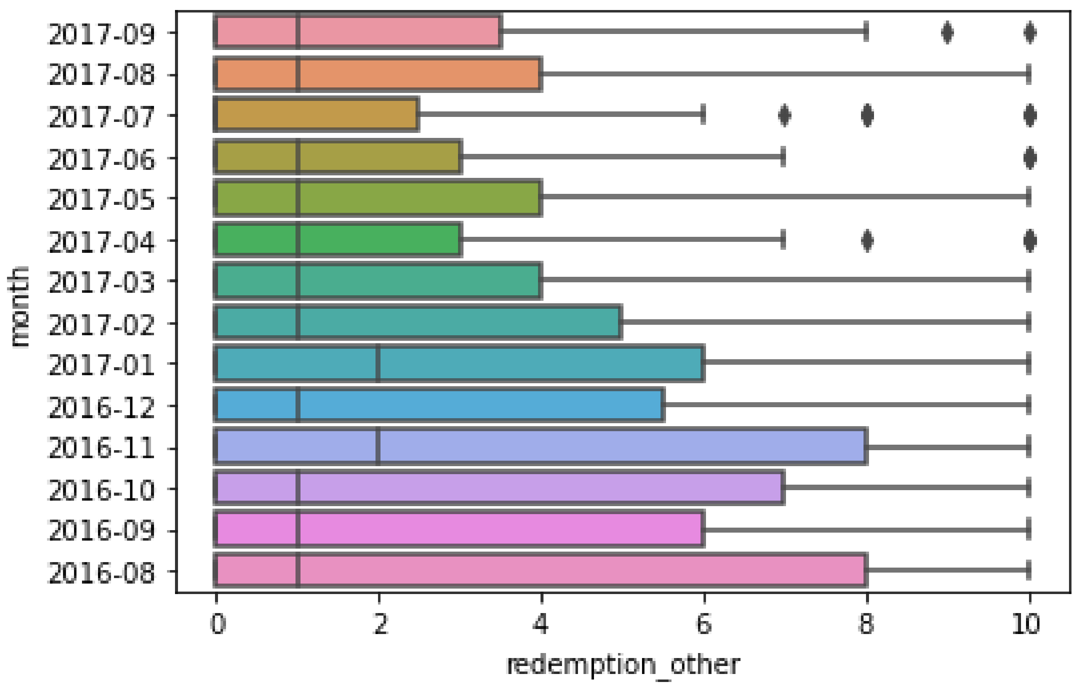
[59]:
```
methode_quartiles('prepawn_other')
```



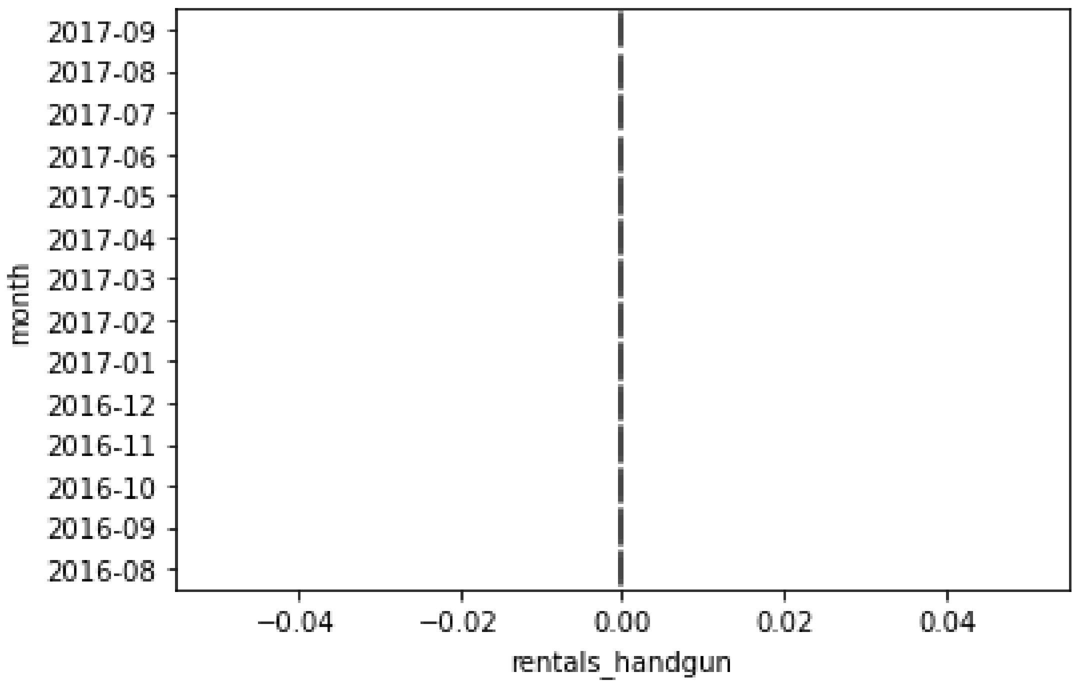
[60]:
```
methode_quartiles('redemption_handgun')
```



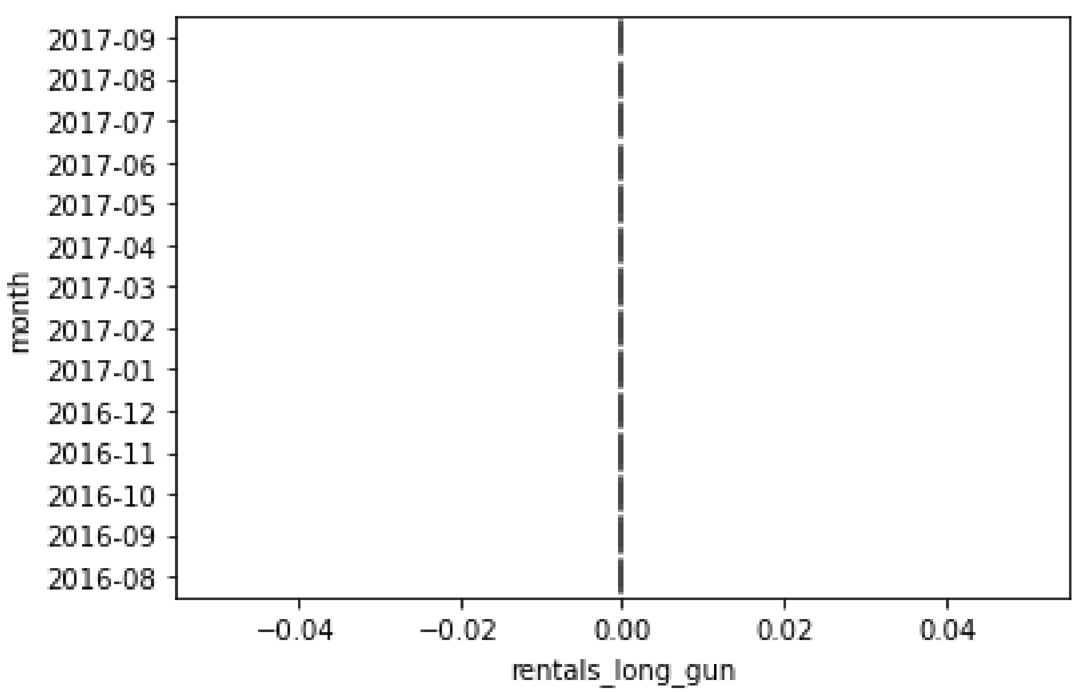
[61]:
```
methode_quartiles('redemption_other')
```



```
[62]: methode_quartiles('rentals_handgun')
```

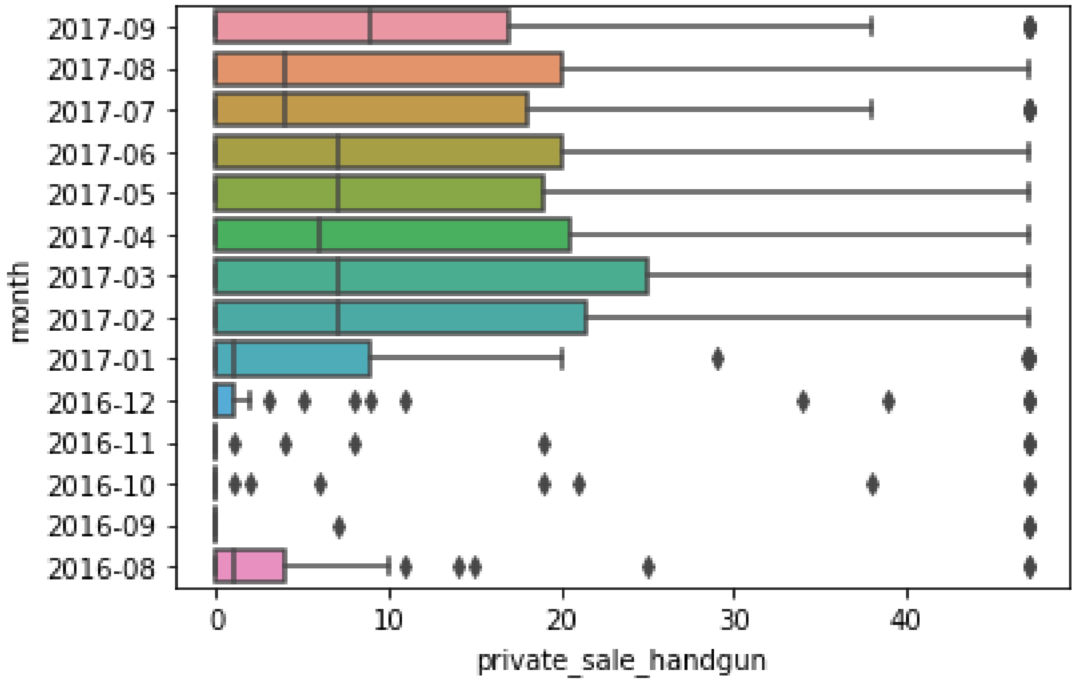



```
[63]: methode_quartiles('rentals_long_gun')
```



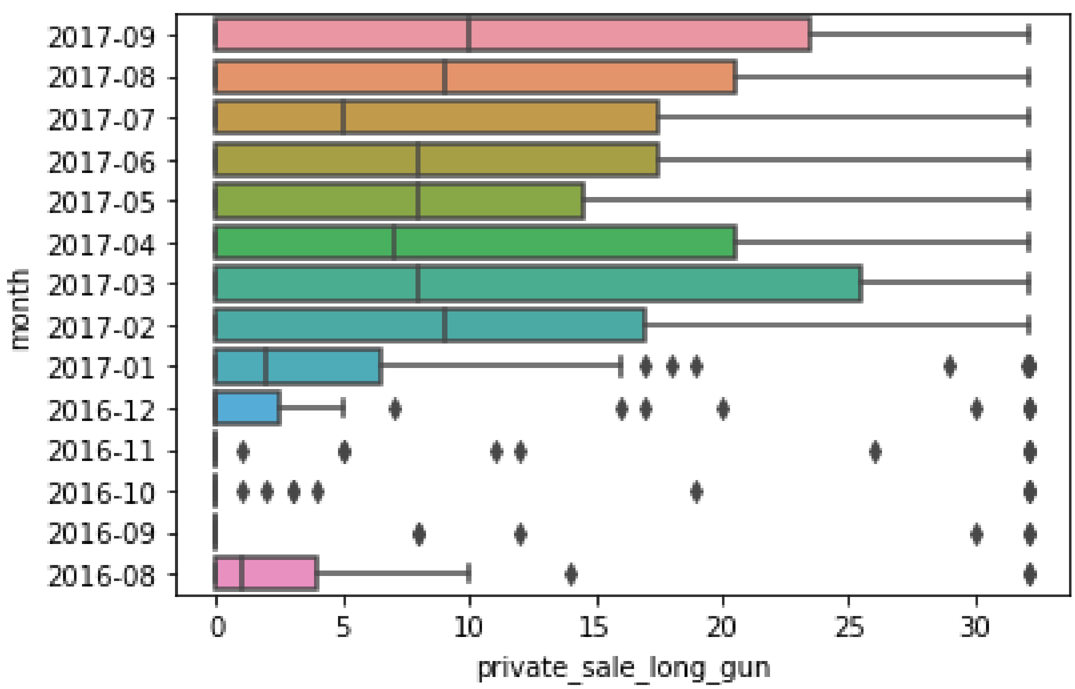
[64]:

```
methode_quartiles('private_sale_handgun')
```

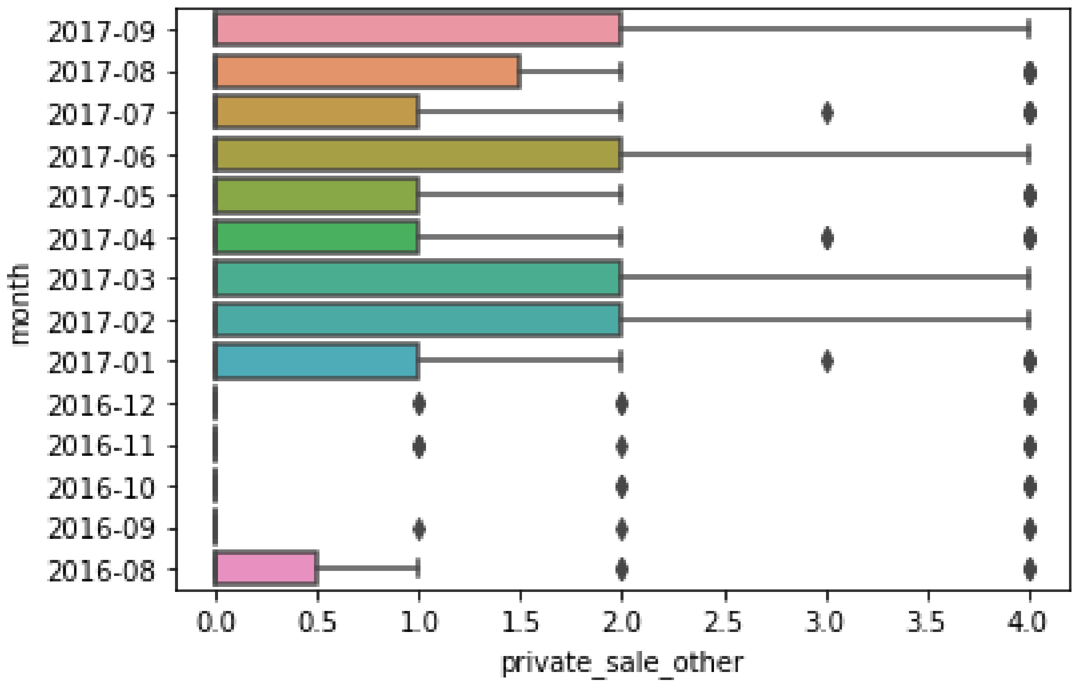


[65]:

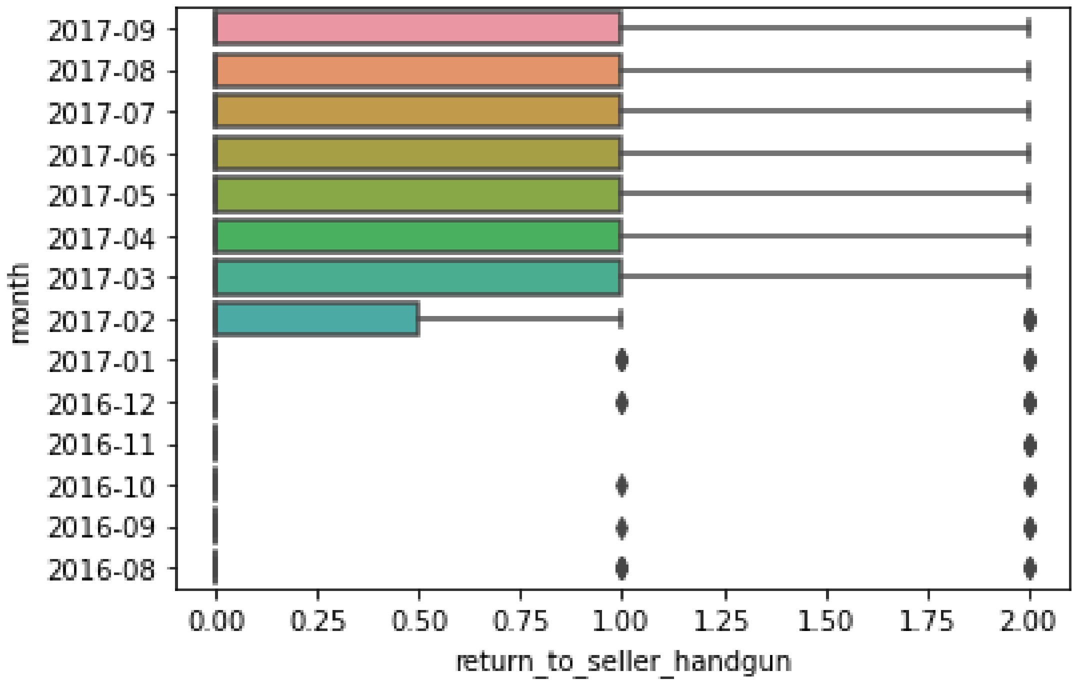
```
methode_quartiles('private_sale_long_gun')
```



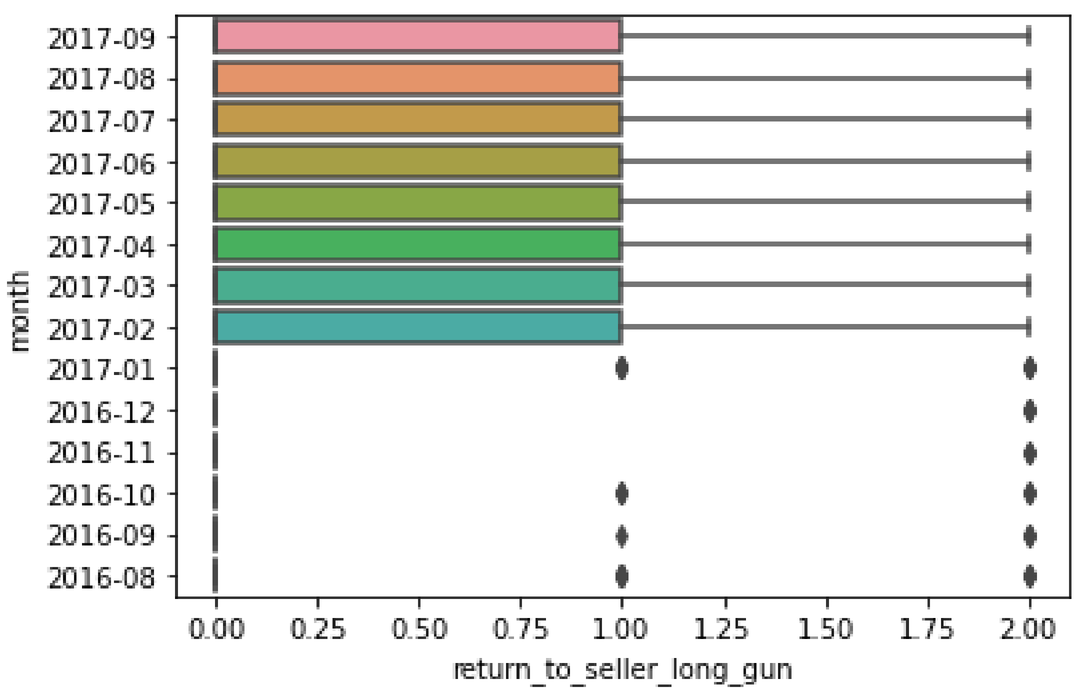
[66]:
```
methode_quartiles('private_sale_other')
```



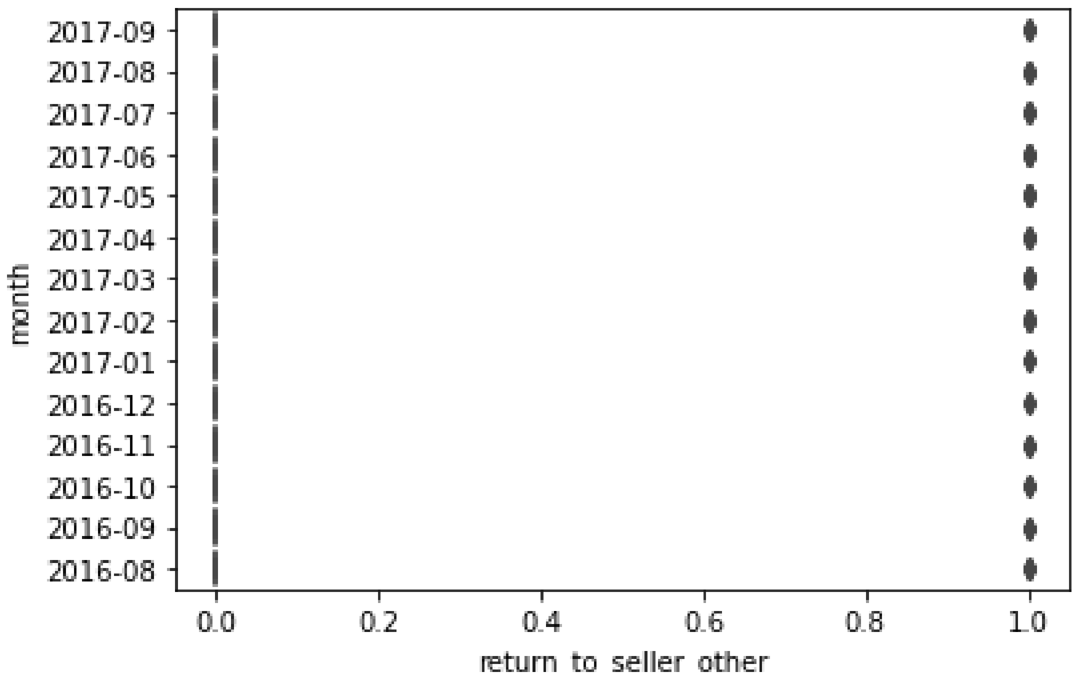
[67]:
```
methode_quartiles('return_to_seller_handgun')
```



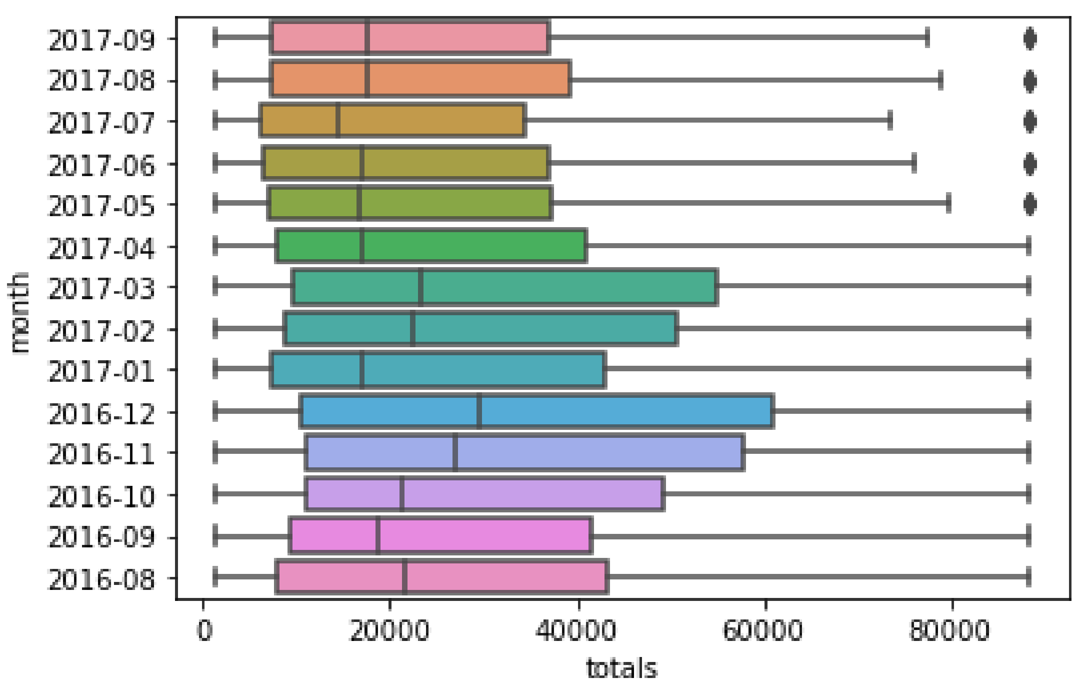
[68]:
```
methode_quartiles('return_to_seller_long_gun')
```



[69]:
```
methode_quartiles('return_to_seller_other')
```



[70]:
```
methode_quartiles('totals')
```



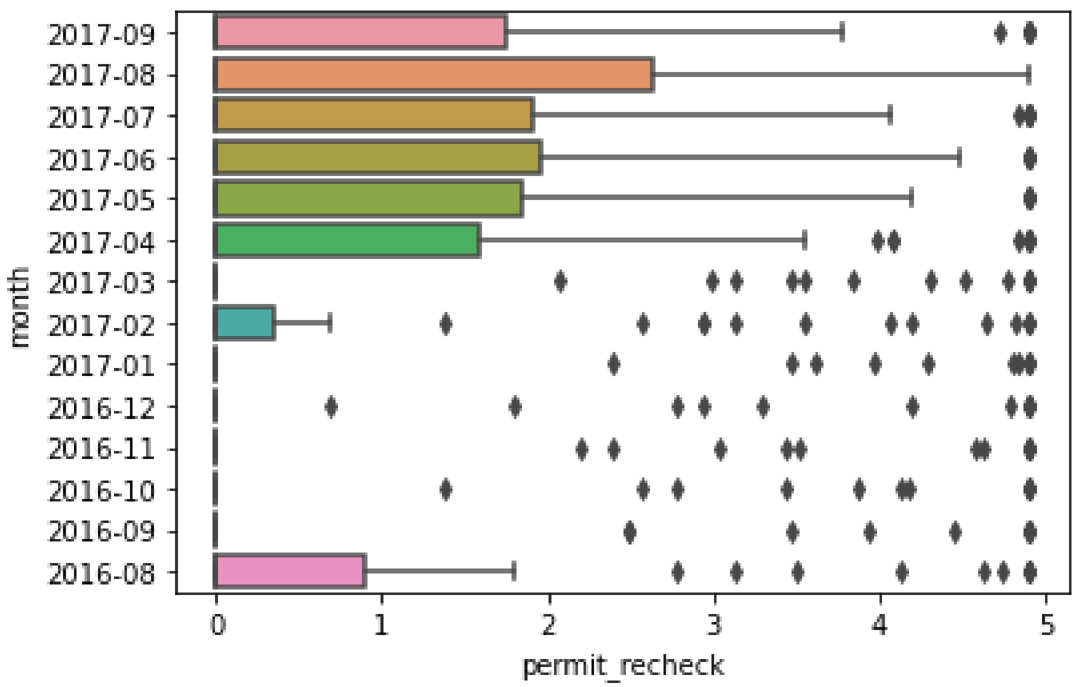
suppression des valeurs extêmes avec la transformation algorithmique methode des log

Ici on traite simplement les variables dont leurs valeurs extrêmes n'ont pas été nettoyer par la methode des quartiles.

[71]:
```
#fonction avec log
def changement_dechelle(col):
    print("Transformation algorithmique")
    #les fonction log se trouve dans numpy
    df_gun_cp[col] = df_gun_cp[col]
    df_gun_cp[col] = df_gun_cp[col].map(lambda i: np.log(i) if i>0 else 0)
    sns.boxplot(x=col, y='month',data=df_gun_cp)
    plt.show()
```

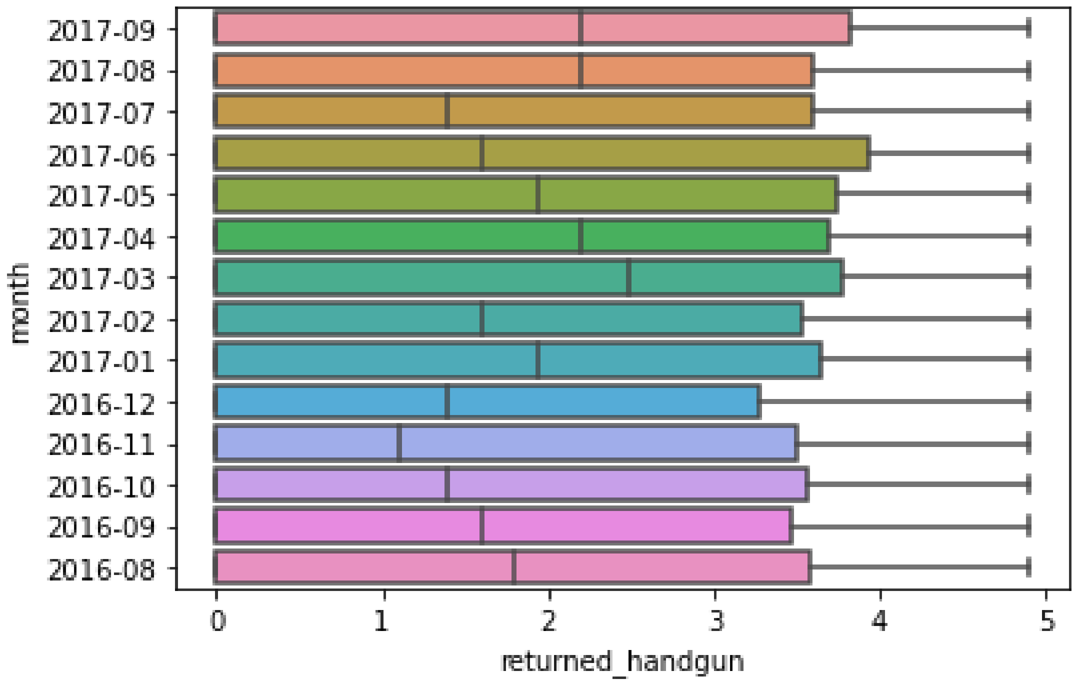
[72]:
```
changement_dechelle('permit_recheck')
```

```
Transformation algorithmique
```



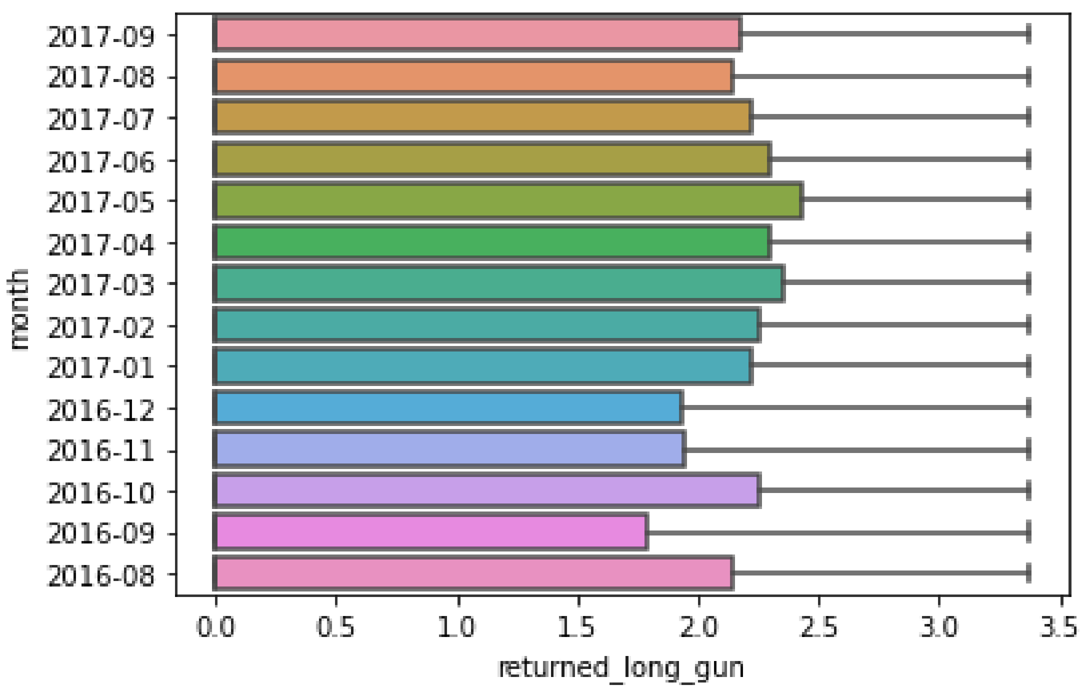
[73]:
```
changement_dechelle('returned_handgun')
```

```
Transformation algorithmique
```



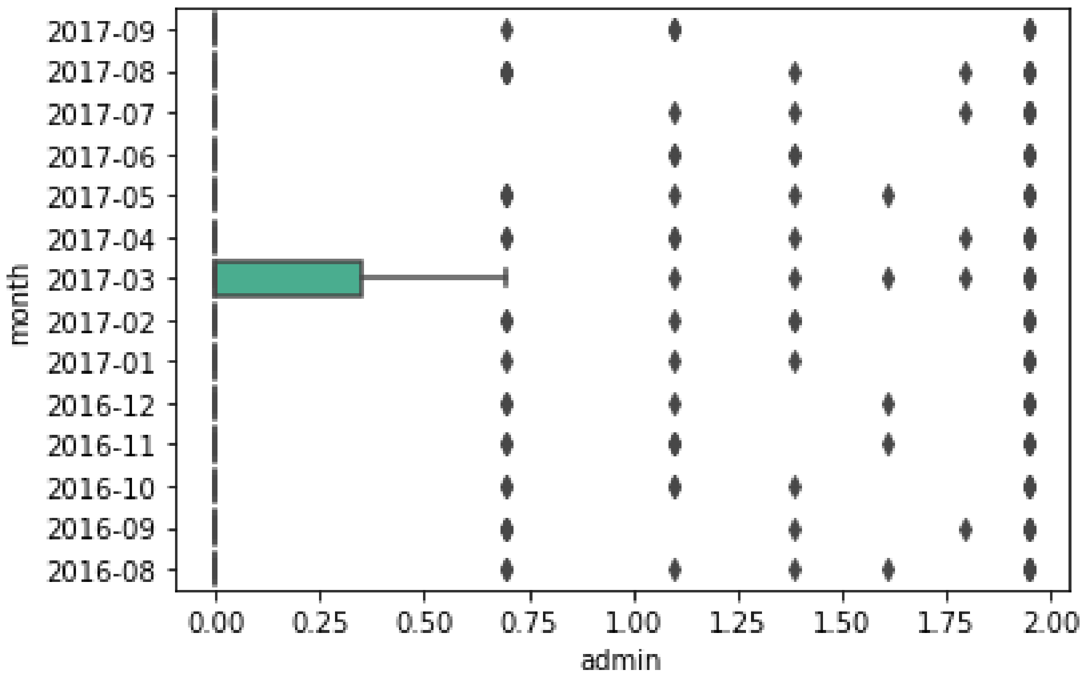
[74]:
```
changement_dechelle('returned_long_gun')
```

```
Transformation algorithmique
```



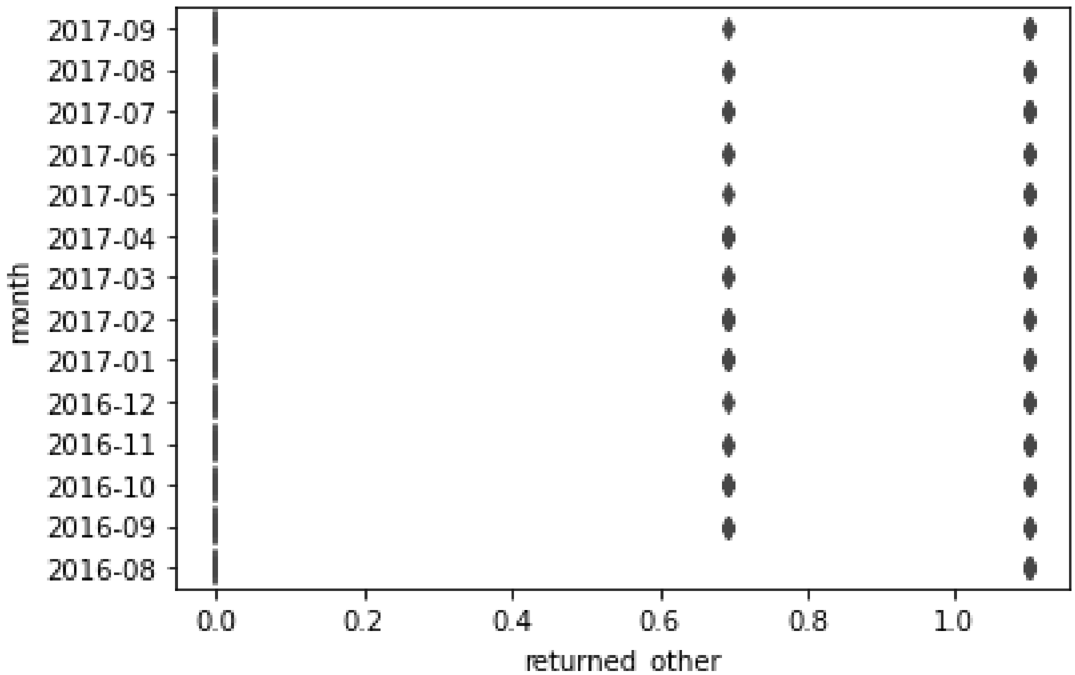
[75]:
```
changement_dechelle('admin')
```

```
Transformation algorithmique
```



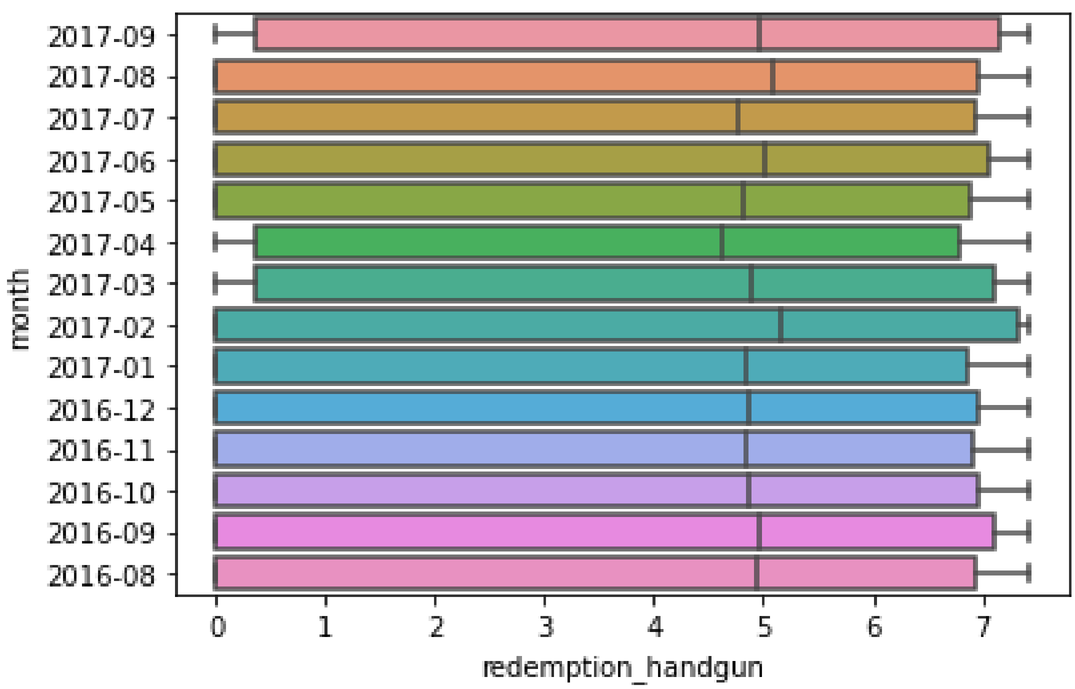
[76]:
```
changement_dechelle('returned_other')
```

```
Transformation algorithmique
```



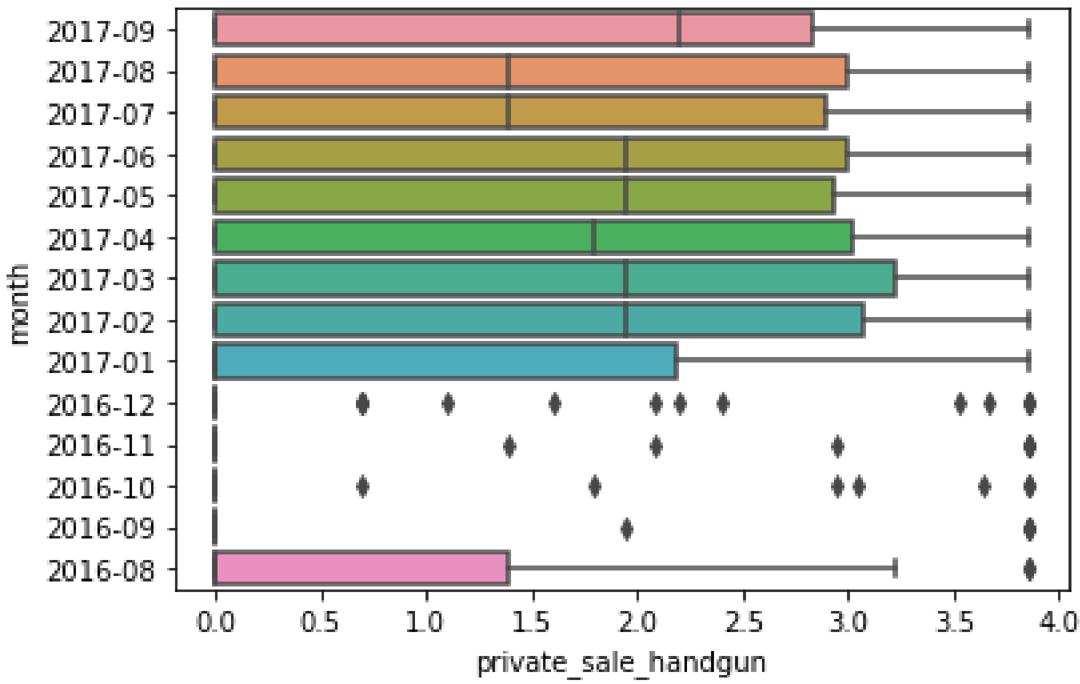
[77]:
```
changement_dechelle('redemption_handgun')
```

```
Transformation algorithmique
```



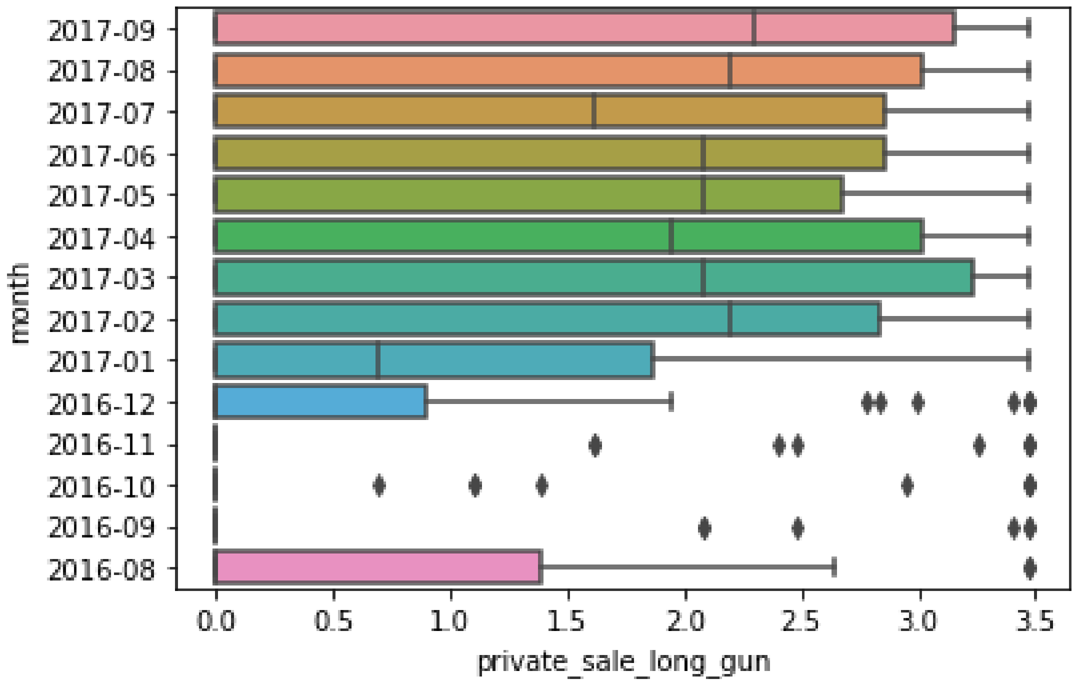
[78]:
```
changement_dechelle('private_sale_handgun')
```

Transformation algorithmique



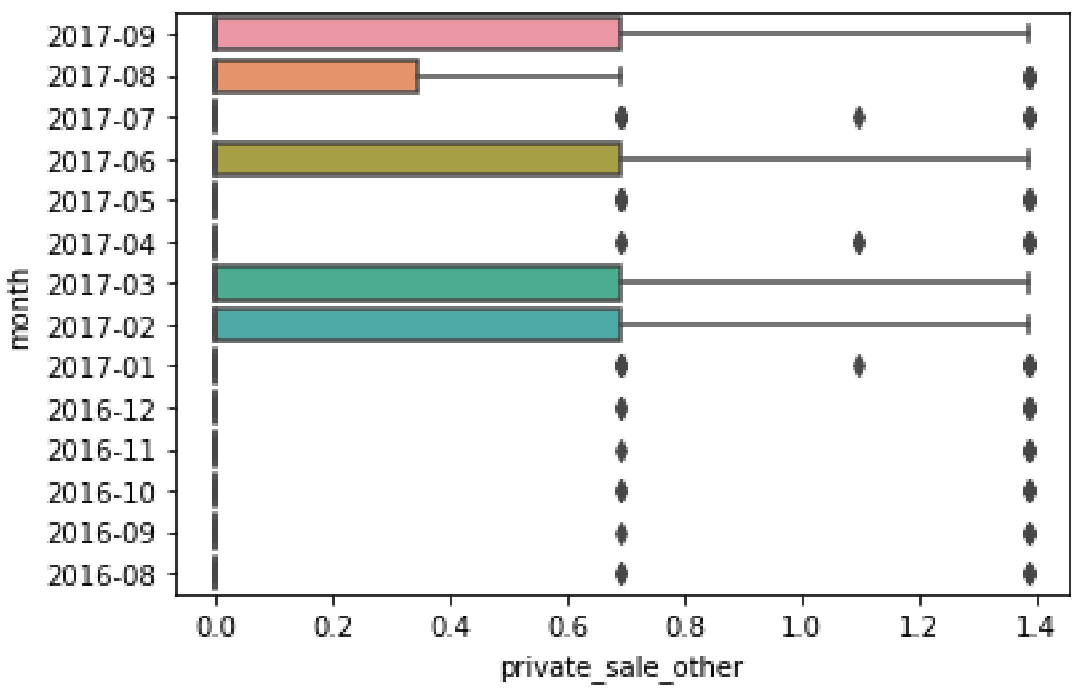
[79]:
```
changement_dechelle('private_sale_long_gun')
```

Transformation algorithmique



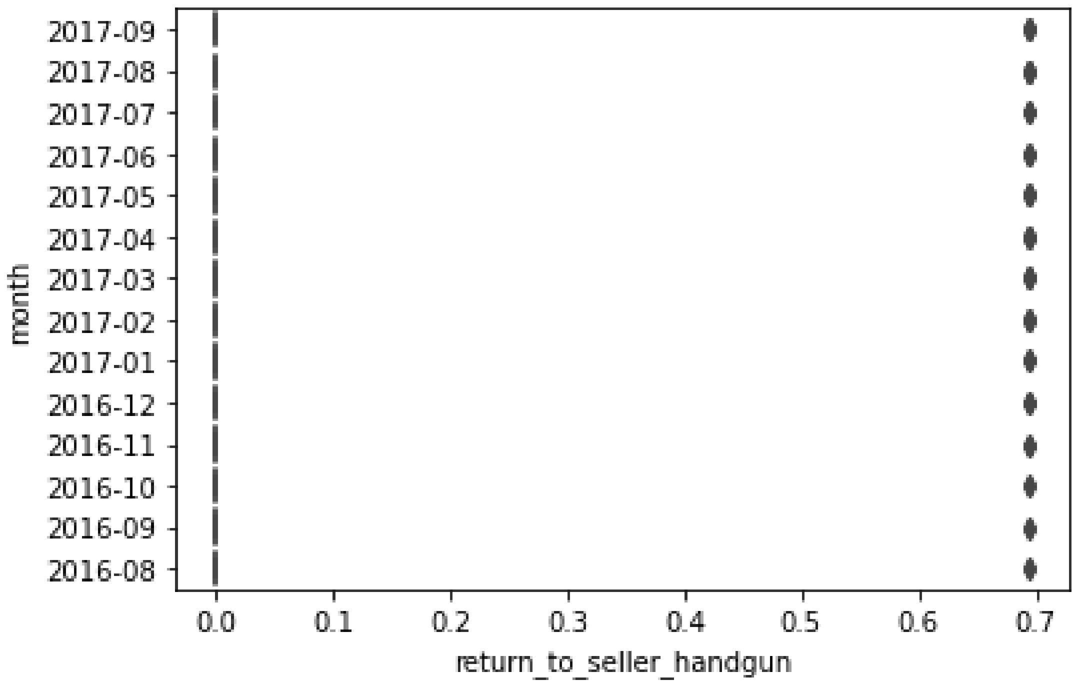
[80]:
```
changement_dechelle('private_sale_other')
```

```
Transformation algorithmique
```



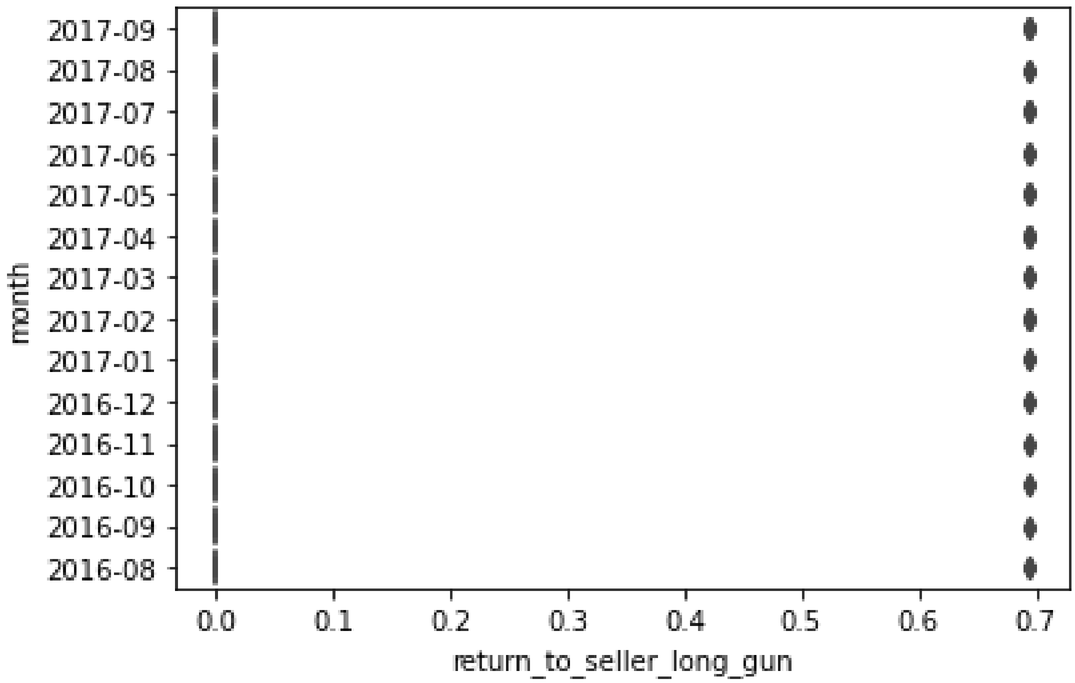
[81]:
```
changement_dechelle('return_to_seller_handgun')
```

```
Transformation algorithmique
```



[82]:
```
changement_dechelle('return_to_seller_long_gun')
```

```
Transformation algorithmique
```



Suppression des valeurs extrêmes

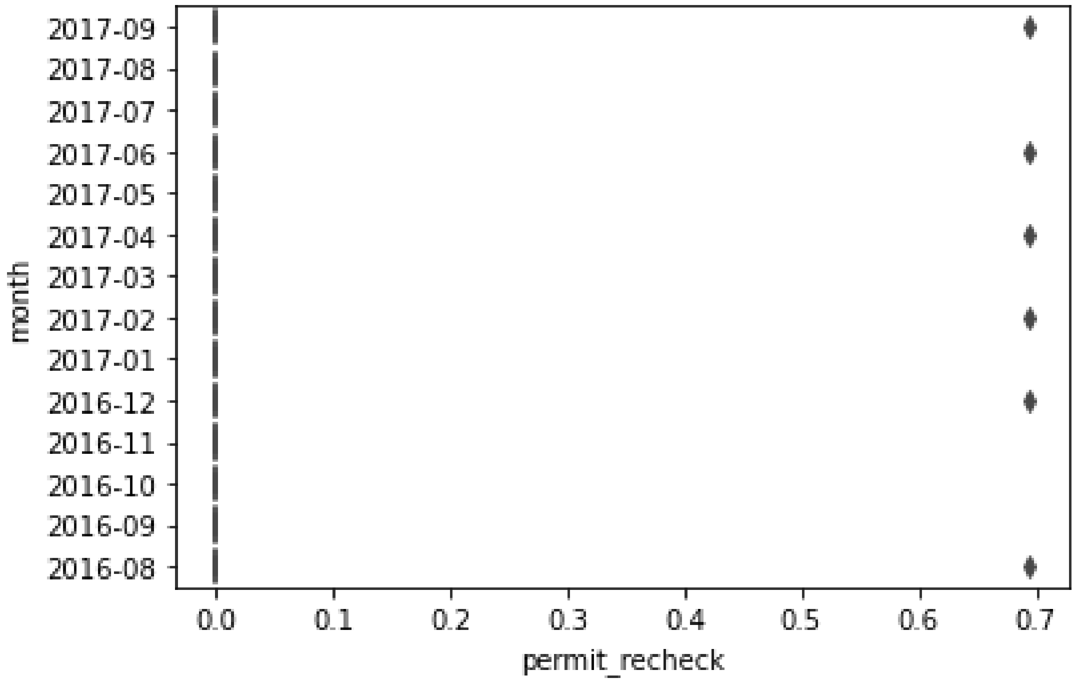
On doit supprimer les valeurs extrêmes des variables n'ont pas été propre apres l'utilisation de la methode des **quartiles** et de la methode **log**

[83]:
```
# fonction pour supprimer les valeurs extrêmes
def suppression_valeur(col):
    #Supprimer les valeurs externes(Conserver les valeurs interquartiles)
    print("Suppression des valeurs externes(Conservation des valeurs␣
 ↪interquartiles) \n")
    q1 = df_gun_cp[col].quantile(0.25)
    q3 = df_gun_cp[col].quantile(0.75)
    index = df_gun_cp[(df_gun_cp[col]<q1) | (df_gun_cp[col] >q3)].index
    print(index)
    df_gun_cp.drop(index, inplace =True)
    sns.boxplot(x=col,y='month', data=df_gun_cp)
    plt.show()
```

[84]:
```
suppression_valeur('permit_recheck')
```

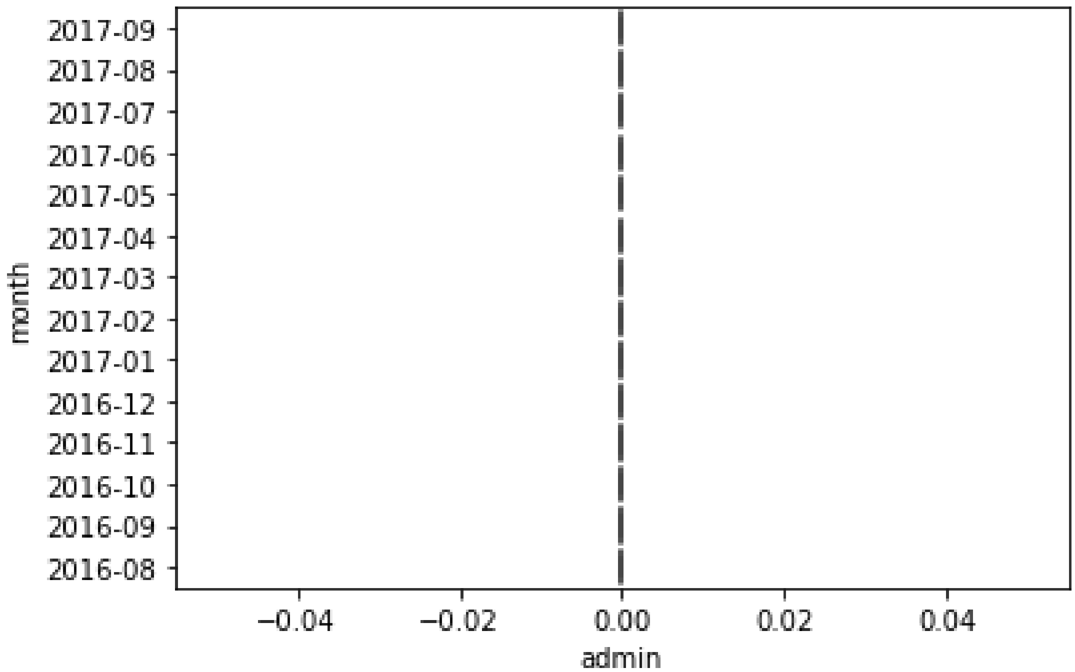
```
Suppression des valeurs externes(Conservation des valeurs interquartiles)

Int64Index([  2,   3,   6,  14,  16,  17,  24,  29,  34,  37,
            …
            732, 738, 739, 744, 752, 754, 760, 762, 766, 769],
```

```
dtype='int64', length=190)
```



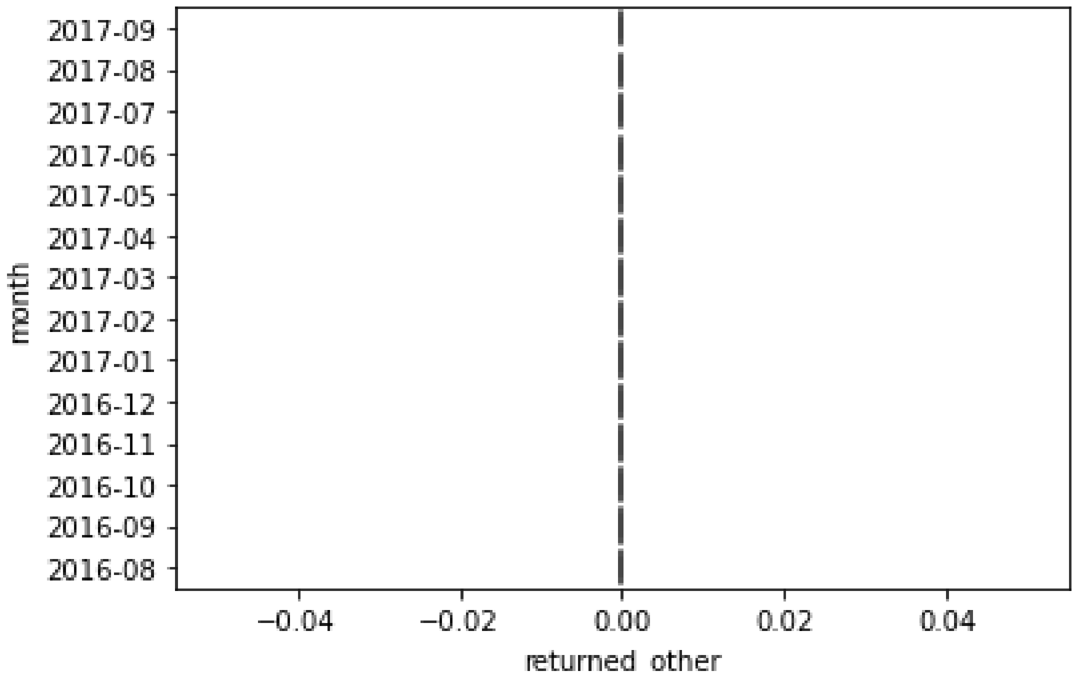
[85]:
```python
suppression_valeur('admin')
```

```
Suppression des valeurs externes(Conservation des valeurs interquartiles)

Int64Index([ 15,  28,  31,  36,  40,  52,  70,  78,  86,  91,
            …
            699, 700, 712, 721, 724, 730, 743, 746, 755, 767],
           dtype='int64', length=106)
```



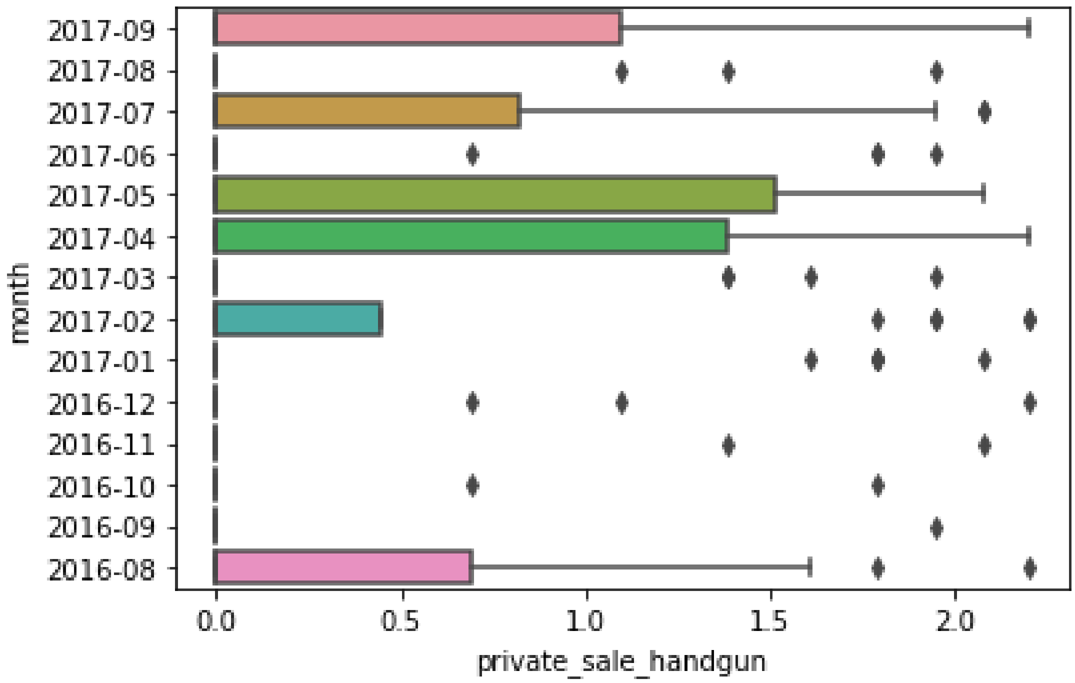
[86]:
```
suppression_valeur('returned_other')
```

```
Suppression des valeurs externes(Conservation des valeurs interquartiles)

Int64Index([ 42,  53,  68,  80,  82, 100, 135, 137, 155, 163, 210, 218, 247,
            265, 273, 280, 288, 302, 320, 328, 355, 357, 361, 372, 375, 427,
            430, 438, 445, 460, 465, 467, 485, 493, 522, 537, 540, 592, 595,
            614, 632, 647, 650, 677, 702, 705, 713, 742, 757, 768],
           dtype='int64')
```



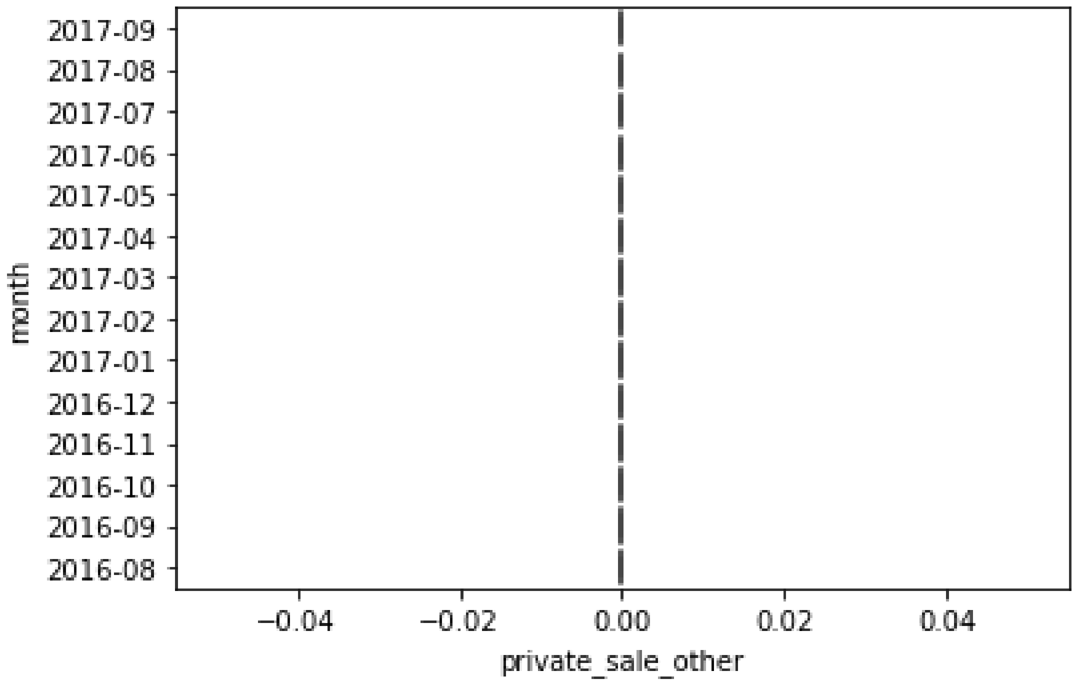
[87]:
```
suppression_valeur('private_sale_handgun')
```

```
Suppression des valeurs externes(Conservation des valeurs interquartiles)

Int64Index([  1,   7,   9,  10,  18,  19,  20,  23,  25,  26,
            …
            514, 518, 533, 588, 639, 640, 643, 694, 749, 761],
           dtype='int64', length=102)
```



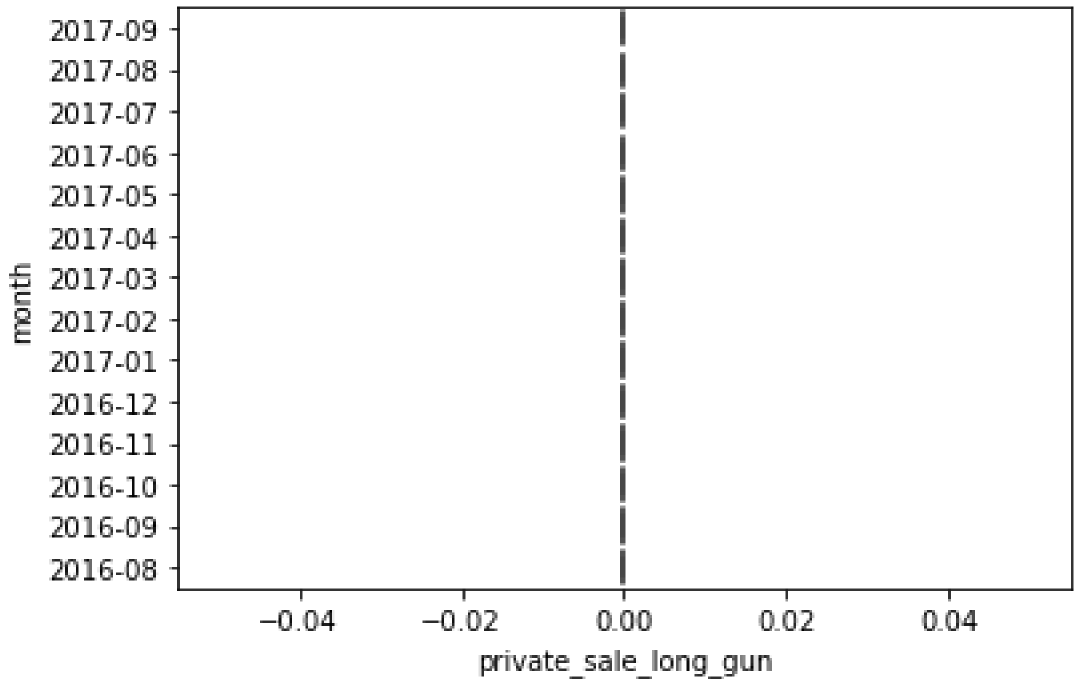
[88]:
```
suppression_valeur('private_sale_other')
```

```
Suppression des valeurs externes(Conservation des valeurs interquartiles)

Int64Index([  0,  35,  77, 123, 152, 187, 209, 246, 264, 276, 308, 352, 371,
            385, 407, 441, 462, 505, 517, 530, 572, 627, 682, 722],
           dtype='int64')
```



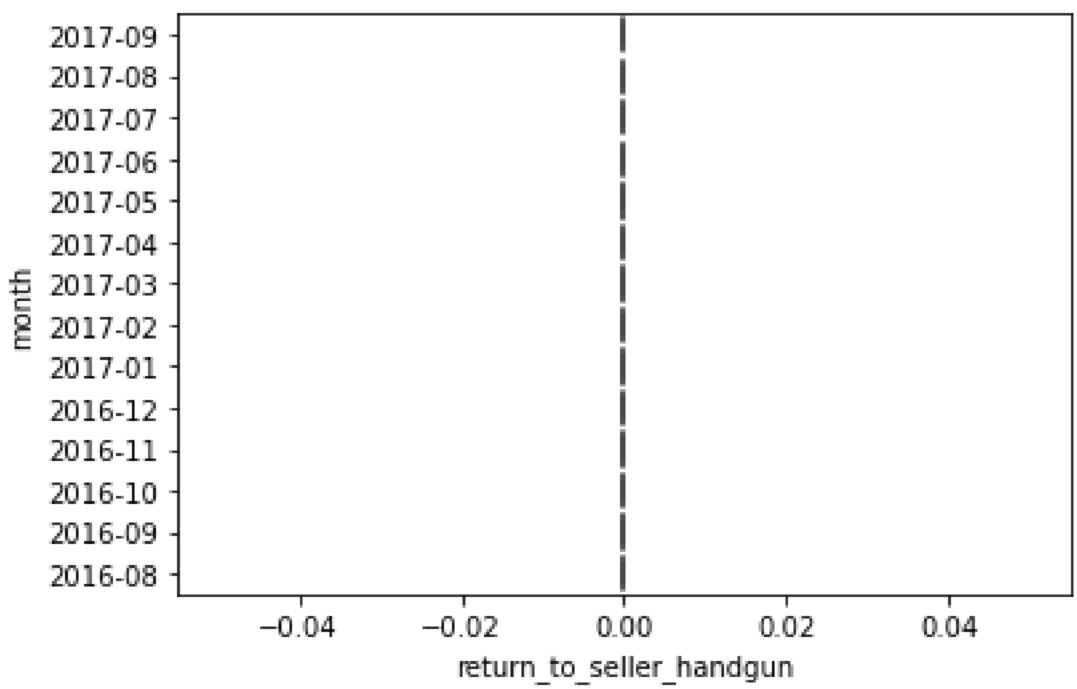
[89]:
```
suppression_valeur('private_sale_long_gun')
```

```
Suppression des valeurs externes(Conservation des valeurs interquartiles)

Int64Index([ 13,  22,  33,  44,  48,  72,  88,  97,  99, 108, 132, 143, 145,
            154, 191, 198, 240, 242, 253, 255, 284, 297, 301, 319, 331, 366,
            374, 378, 383, 386, 391, 405, 416, 429, 433, 453, 458, 459, 466,
            473, 475, 484, 495, 504, 508, 538, 569, 585, 593, 624, 648, 679,
            695, 715, 725, 728, 733, 734, 737, 740, 748, 750, 753],
           dtype='int64')
```



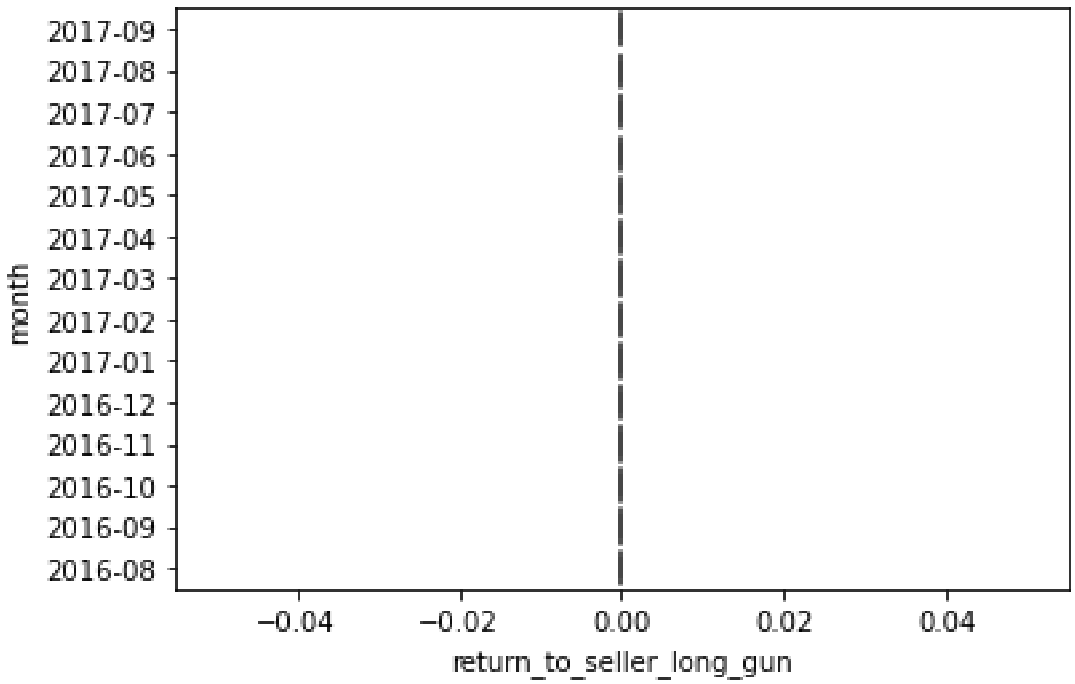
[90]:
```
suppression_valeur('return_to_seller_handgun')
```

```
Suppression des valeurs externes(Conservation des valeurs interquartiles)

Int64Index([], dtype='int64')
```



[91]:
```
suppression_valeur('return_to_seller_long_gun')
```

```
Suppression des valeurs externes(Conservation des valeurs interquartiles)

Int64Index([], dtype='int64')
```



Apres l'utilisation de l'ensemble des methodes pour la visualisation,l'observation et la suppression des valeurs extrêmes, On va passer à la suppression des colonnes inutiles où des colonnes que l'on peu s'en passer pour faire notre analyse.

Suppression des colonnes inutiles

- admin
- permit_recheck
- returned_other
- rentals_long_gun
- rentals_handgun
- return_to_seller_handgun
- return_to_seller_long_gun
- return_to_seller_other
- private_sale_handgun
- private_sale_long_gun
- private_sale_other

[92]:
```
df_gun_cp.
↪drop(['admin','permit_recheck','returned_other','rentals_long_gun','rentals_handgun',
              ␣
↪'return_to_seller_handgun','return_to_seller_long_gun','return_to_seller_other',
              ␣
↪'private_sale_handgun','private_sale_long_gun','private_sale_other'],axis=1,inplace=True)
```

La suppression des colonnes mentionner dans la cellules ci-dessus met fin notre étape de nettoyages des données. Ceci etant fait, nous allons passer a la phase de **l'analyses exploratoires des données**

5 Analyse Exploratoires des données

Nous allons commencer par faire une petite comparaison des données avant et apres le nettoyages

[93]:
```
#Avant netoyages
print('*********** Avant le nettoyages nous avions {} lignes et {} colonnes et␣
↪la tailles etait {} ********** '.format(df_gun.shape[0],df_gun.
↪shape[1],df_gun.size))
```

```
*********** Avant le nettoyages nous avions 12485 lignes et 27 colonnes et la
tailles etait 337095 **********
```

[94]:
```
# apres nettoyages
print('*********** Aprés le nettoyages nous avons {} lignes et {} colonnes et␣
↪la tailles est de {} ********** '.format(df_gun_cp.shape[0],df_gun_cp.
↪shape[1],df_gun_cp.size))
```

```
*********** Aprés le nettoyages nous avons 235 lignes et 16 colonnes et la
tailles est de 3760 **********
```

Les données nettoyer seront enregsitrer dans un nouveau fichiers CSV

[95]:
```
# Enregstrement d'un nouveau fichier csv
df_gun_cp.to_csv('data/gun_data_clean.csv',index=False)
```

[96]:
```
# Lecture du nouveau fichier
df=pd.read_csv('data/gun_data_clean.csv')
```

[97]:
```
# utilisation de la methode tail() qui affiche les 5 derniers lignes du dataset
df.tail()
```

[97]:
```
       month           state   permit  handgun  long_gun  other  multiple  \
230  2016-08  South Carolina  13000.0   8828.0    6906.0  494.0     330.0
231  2016-08    South Dakota   1259.0   2387.0    3559.0  202.0     133.0
232  2016-08         Vermont      0.0   1269.0    1296.0  108.0      47.0
233  2016-08  Virgin Islands     55.0    125.2     205.8    0.0       0.0
234  2016-08        Virginia    776.0  21108.0   15802.0  978.0       0.0
```

```
     prepawn_handgun  prepawn_long_gun  prepawn_other  redemption_handgun  \
230              4.0               7.0            0.0            6.924612
231              0.0               0.0            0.0            4.927254
232              0.0               0.0            0.0            0.000000
233              0.0               0.0            0.0            0.000000
234              0.0               0.0            0.0            0.000000

     redemption_long_gun  redemption_other  returned_handgun  \
230                743.0              10.0          3.555348
231                192.0               0.0          0.000000
232                  2.0               1.0          0.000000
233                  0.0               0.0          0.000000
234                  0.0               0.0          0.000000

     returned_long_gun   totals
230           1.386294  31389.0
231           0.000000   7873.0
232           0.000000   2724.0
233           0.000000   1424.8
234           0.693147  38667.0
```

Affichages des 5 derniers lignes du dataset avcc la methodes tail() de pandas; dans le celulles suivante nous allons voir les details du dataset

[98]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 235 entries, 0 to 234
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   month                235 non-null    object
 1   state                235 non-null    object
 2   permit               235 non-null    float64
 3   handgun              235 non-null    float64
 4   long_gun             235 non-null    float64
 5   other                235 non-null    float64
 6   multiple             235 non-null    float64
 7   prepawn_handgun      235 non-null    float64
 8   prepawn_long_gun     235 non-null    float64
 9   prepawn_other        235 non-null    float64
 10  redemption_handgun   235 non-null    float64
 11  redemption_long_gun  235 non-null    float64
 12  redemption_other     235 non-null    float64
 13  returned_handgun     235 non-null    float64
 14  returned_long_gun    235 non-null    float64
 15  totals               235 non-null    float64
```

```
dtypes: float64(14), object(2)
memory usage: 29.5+ KB
```

Les details sur les données; nous allons montrer les statistique descriptif des variables de types numériques;Ici on a 770 entrées (0 à 769).Vous avez aussi le nom des colonnes, leurs types et le nombres de valeurs non_null. Vous allez constater que l'on 770 entrées, toutes les colonnes on 770 valeurs non-null donc pas de valeurs null.

[99]:
```
df.describe()
```

[99]:

	permit	handgun	long_gun	other	multiple
count	235.000000	235.000000	235.000000	235.000000	235.000000
mean	4911.720000	7858.080000	6021.217021	399.281702	178.154043
std	8927.247264	9527.487204	6643.961247	510.377837	290.596633
min	0.000000	125.200000	205.800000	0.000000	0.000000
25%	0.000000	125.200000	205.800000	1.000000	0.000000
50%	819.000000	3992.000000	3040.000000	196.000000	8.000000
75%	2810.500000	12639.000000	12221.500000	492.000000	245.500000
max	26413.700000	28476.000000	17932.800000	1508.400000	973.400000

	prepawn_handgun	prepawn_long_gun	prepawn_other	redemption_handgun
count	235.000000	235.000000	235.000000	235.000000
mean	1.514894	1.623830	0.085106	2.438787
std	3.460830	3.561971	0.279636	2.941384
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%	0.500000	1.000000	0.000000	5.677820
max	12.000000	12.100000	1.000000	7.417040

	redemption_long_gun	redemption_other	returned_handgun
count	235.000000	235.000000	235.000000
mean	242.952340	1.855319	0.937726
std	449.494418	3.326327	1.552409
min	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000
75%	233.500000	2.000000	1.609438
max	1467.600000	10.000000	4.891101

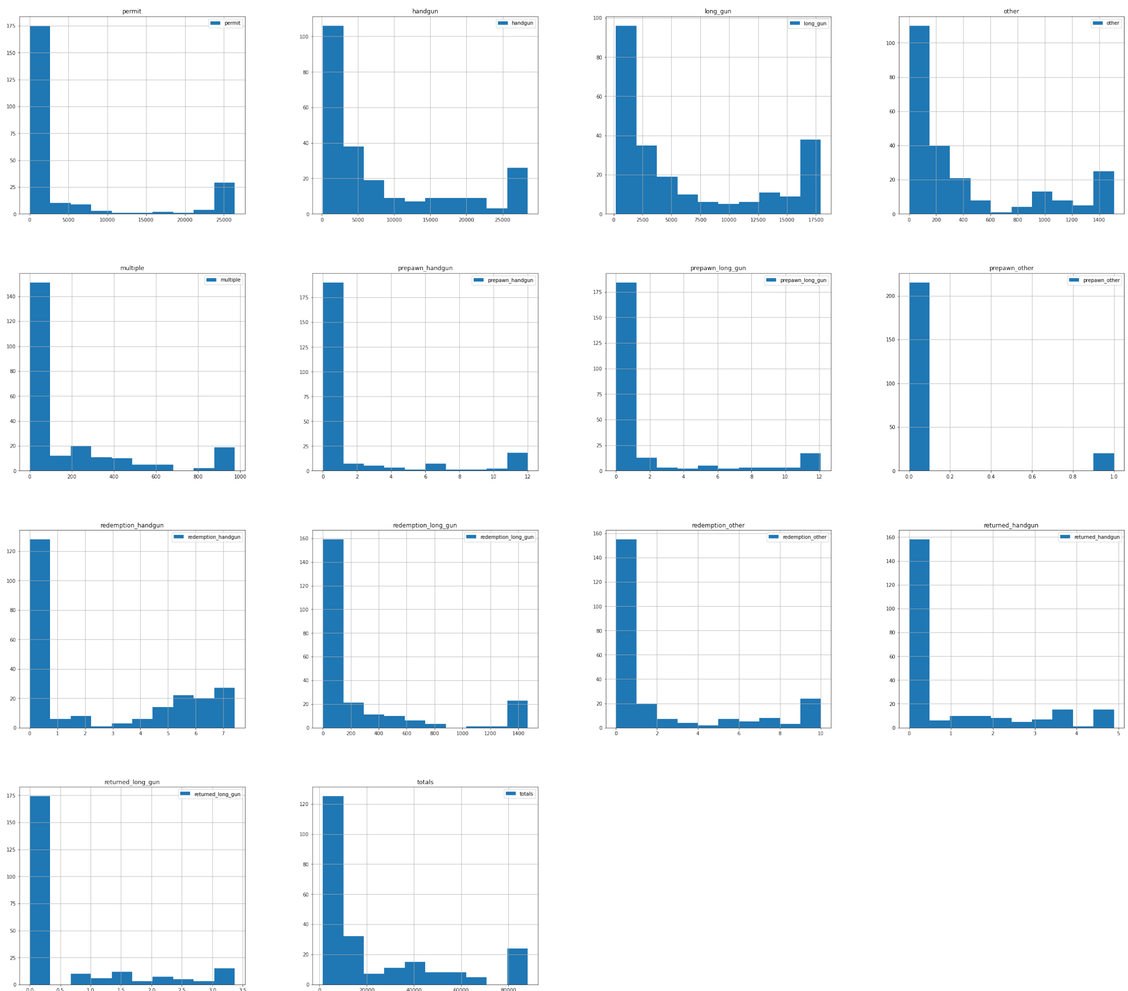
	returned_long_gun	totals
count	235.000000	235.000000
mean	0.524984	22116.445106
std	1.015590	28041.108319
min	0.000000	1424.800000
25%	0.000000	1424.800000
50%	0.000000	9284.000000
75%	0.693147	36314.500000

```
max               3.367296  88253.000000
```

Ici vous voyez les statistiques descriptives . La cellules suivantes vous montrera une vue globale sur la repartitins des données a l'aide d'un diagramme appelé histogramme de la methode **hist()** de pandas

[100]:
```python
# Histogramme
df.hist(figsize=(40,36),legend=True)
```

[100]:
```
array([[<AxesSubplot:title={'center':'permit'}>,
        <AxesSubplot:title={'center':'handgun'}>,
        <AxesSubplot:title={'center':'long_gun'}>,
        <AxesSubplot:title={'center':'other'}>],
       [<AxesSubplot:title={'center':'multiple'}>,
        <AxesSubplot:title={'center':'prepawn_handgun'}>,
        <AxesSubplot:title={'center':'prepawn_long_gun'}>,
        <AxesSubplot:title={'center':'prepawn_other'}>],
       [<AxesSubplot:title={'center':'redemption_handgun'}>,
        <AxesSubplot:title={'center':'redemption_long_gun'}>,
        <AxesSubplot:title={'center':'redemption_other'}>,
        <AxesSubplot:title={'center':'returned_handgun'}>],
       [<AxesSubplot:title={'center':'returned_long_gun'}>,
        <AxesSubplot:title={'center':'totals'}>, <AxesSubplot:>,
        <AxesSubplot:>]], dtype=object)
```



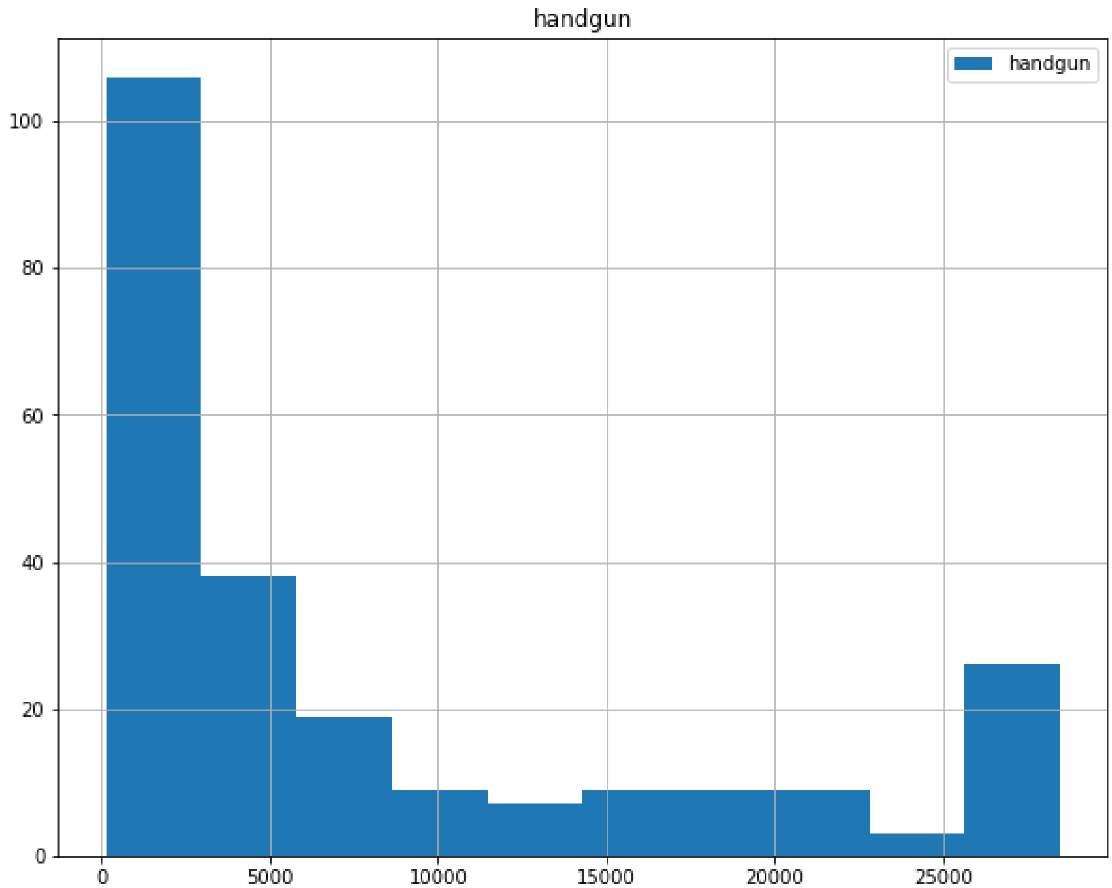
Ici On a une vue globales sur l'ensembles des variables du dataset. Sur les cellules qui vont suivre nous allons tenter des repondres au questions qui ont été poser dans la phase question un peu en haut.

5.1 Quelles sont les types d'armes les plus achetés en moyenne?

[102]:
```
# la moyenne
print(" {} d'armes de poing sont utilisées en moyenne ".format(df['handgun'].
 ↪mean()))
df.hist(legend=True,column='handgun',figsize=(10,8))
```

```
7858.079999999994 d'armes de poing sont utilisées en moyenne
```

[102]:
```
array([[<AxesSubplot:title={'center':'handgun'}>]], dtype=object)
```

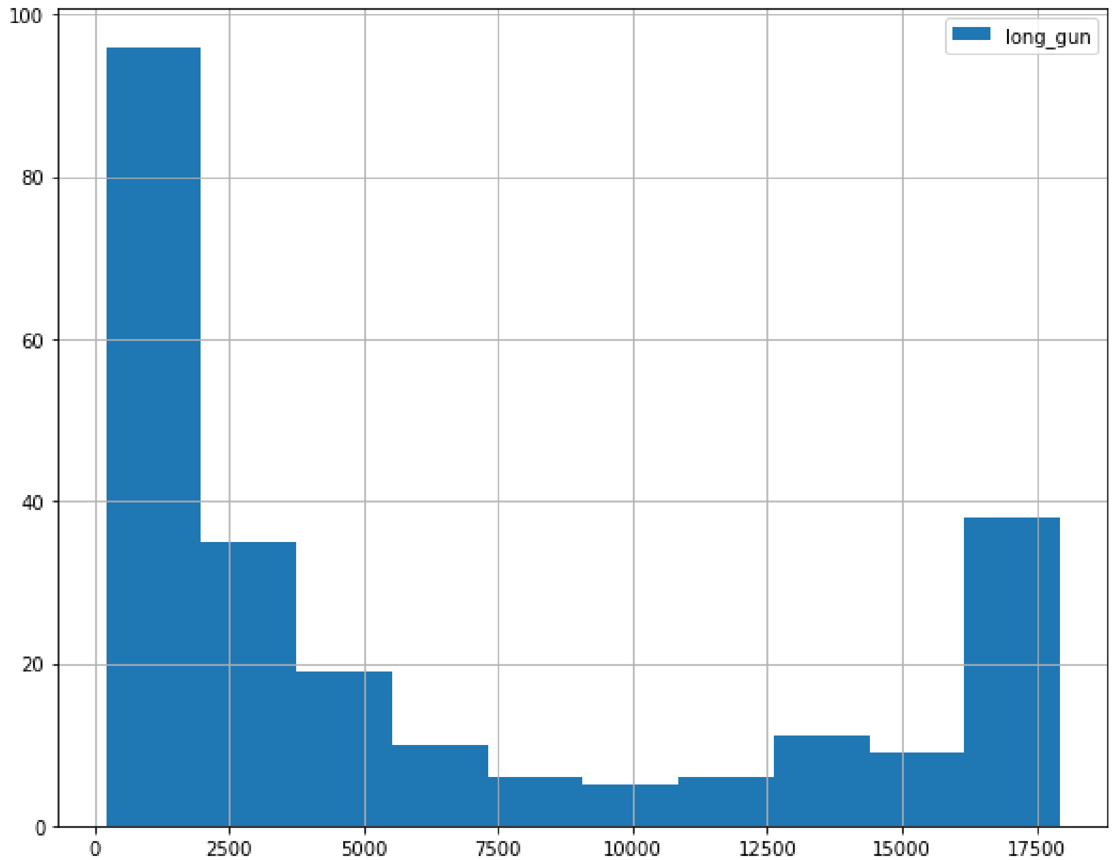


7858.079999999994 d'armes de poing sont utilisées en moyenne . La repartition des armes de poings en moyenne

[103]:
```
print(" {} d'armes d'épaules sont utilisées en moyenne ".format(df['long_gun'].
    mean()))
df['long_gun'].hist(figsize=(10,8),legend=True)
```

```
6021.217021276602 d'armes d'épaules sont utilisées en moyenne
```

[103]: <AxesSubplot:>

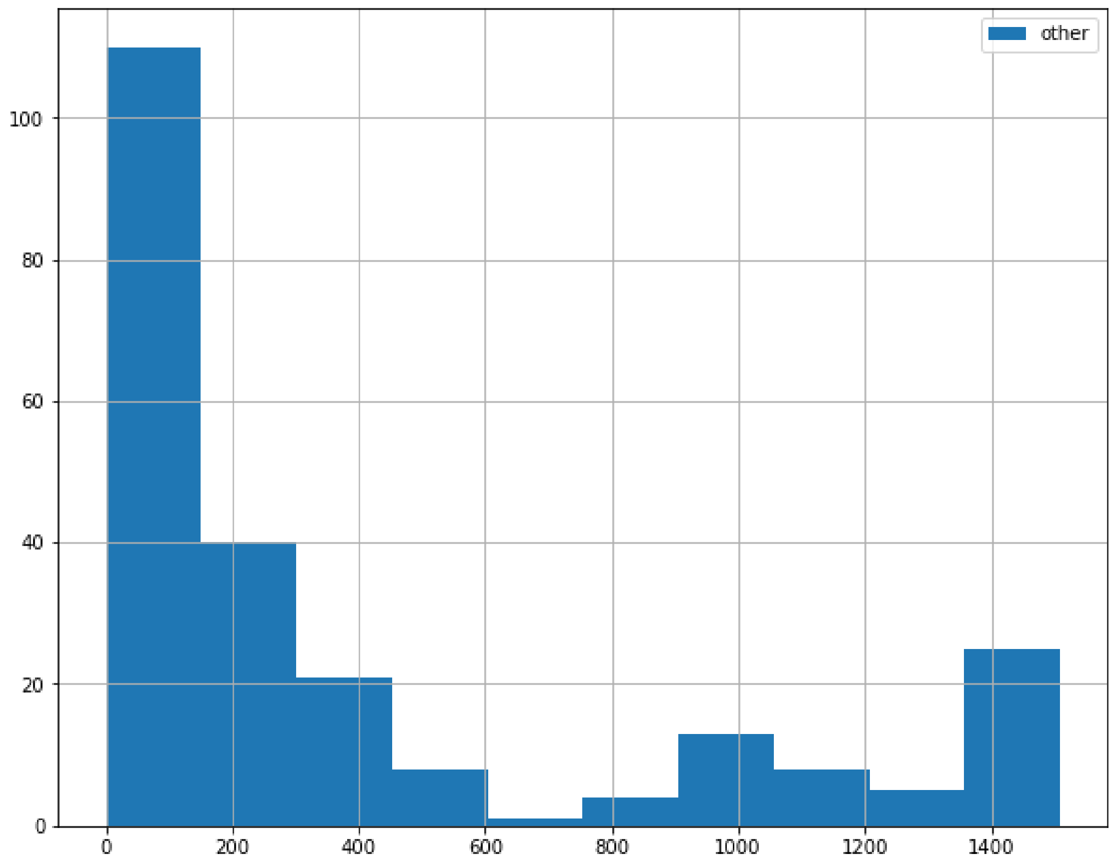


6021.217021276602d’armes d’épaules sont utilisées en moyenne. On a la repartitions d’armes d’épaules en moyenne

[104]:
```
print("{} autres types d'armes sont utilisées en moyenne ".format(df['other'].
    ↪mean()))
df['other'].hist(legend=True,figsize=(10,8))
```

```
399.2817021276595 autres types d'armes sont utilisées en moyenne
```

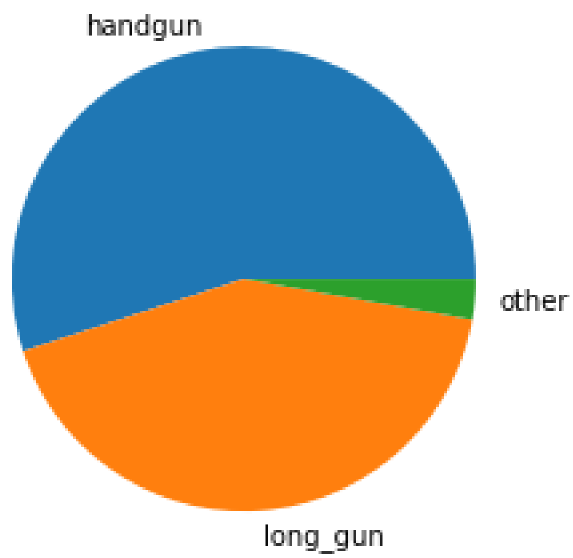
[104]: <AxesSubplot:>



399.2817021276595 autres types d'armes sont utilisées en moyenne .Repartition d'autres types d'armes en moyennes

[105]:
```
handgun = df['handgun'].mean()
long_gun = df['long_gun'].mean()
other = df['other'].mean()
arme = np.array([handgun,long_gun,other])
plt.pie(arme,labels=['handgun','long_gun','other'])
```

[105]:
```
([<matplotlib.patches.Wedge at 0x7fba7d222160>,
  <matplotlib.patches.Wedge at 0x7fba7d222640>,
  <matplotlib.patches.Wedge at 0x7fba7d222b20>],
 [Text(-0.17324012296324268, 1.0862724611236727, 'handgun'),
  Text(0.07726532057659248, -1.097283040166117, 'long_gun'),
  Text(1.095758017273731, -0.09651097129519559, 'other')])
```



On constate les types d'armes les plus achetés sont les armes de poing **(handgun)** ensuite vient les armes d'épaules **(long__gun)**

5.2 Quels États ont connu la plus forte croissance dans enregistrements d'armes à feu ?

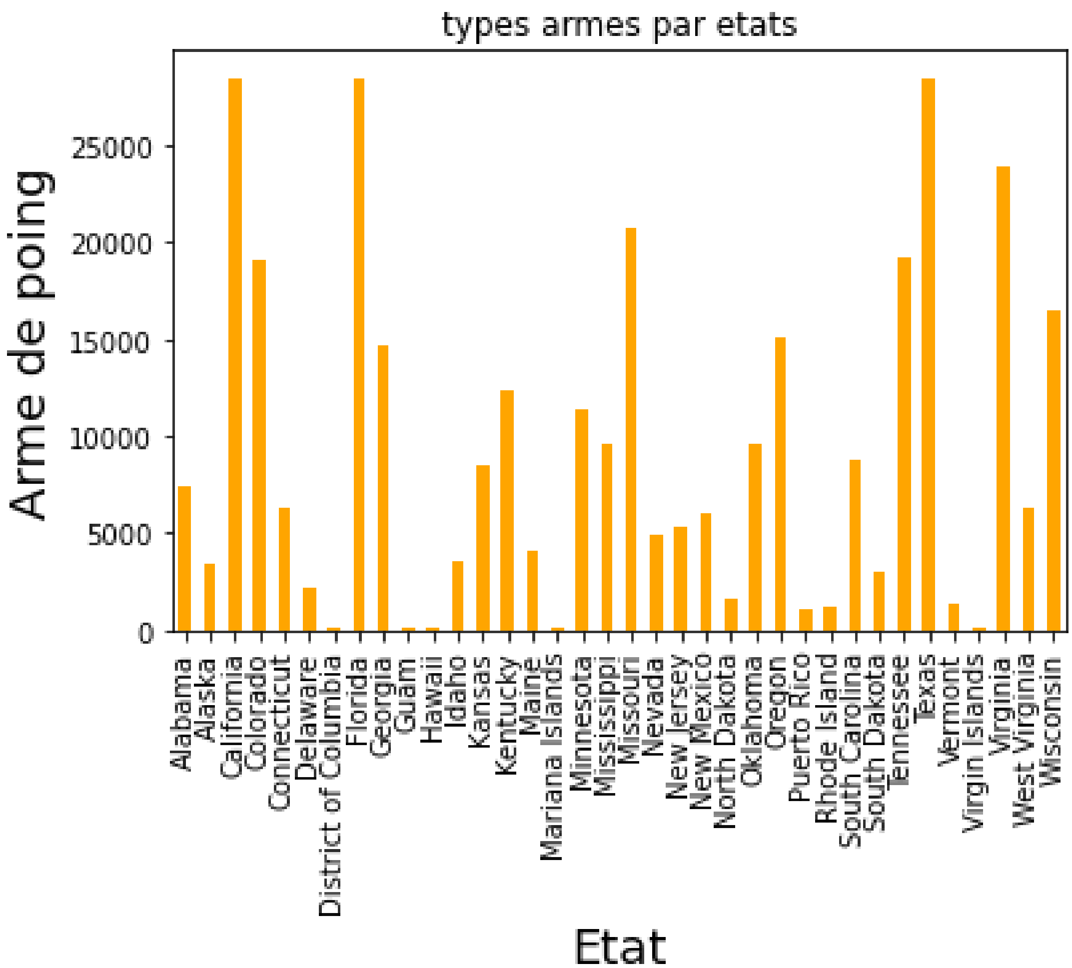
[106]:
```
df.groupby('state').handgun.mean()
```

[106]:
```
state
Alabama                  7441.000000
Alaska                   3353.600000
California              28476.000000
Colorado                19122.272727
Connecticut              6344.800000
Delaware                 2106.333333
District of Columbia      125.200000
Florida                 28476.000000
Georgia                 14689.333333
Guam                      125.328571
Hawaii                    125.200000
Idaho                    3518.500000
Kansas                   8494.000000
Kentucky                12296.500000
Maine                    4046.200000
Mariana Islands           125.200000
Minnesota               11427.750000
Mississippi              9642.200000
```

```
Missouri                20763.500000
Nevada                   4947.785714
New Jersey               5311.000000
New Mexico               5969.750000
North Dakota             1571.400000
Oklahoma                 9638.000000
Oregon                  15064.500000
Puerto Rico              1131.000000
Rhode Island             1196.000000
South Carolina           8776.500000
South Dakota             3005.000000
Tennessee               19219.000000
Texas                   28476.000000
Vermont                  1397.900000
Virgin Islands            125.200000
Virginia                23911.214286
West Virginia            6279.000000
Wisconsin               16538.333333
Name: handgun, dtype: float64
```

[107]:
```
hand=df.groupby('state').mean().handgun
hand.plot(kind='bar',title='types armes par etats',color='orange',alpha=1)
plt.xlabel('Etat',fontsize=18,)
plt.ylabel('Arme de poing',fontsize=18)
```

[107]:
```
Text(0, 0.5, 'Arme de poing')
```



Ce graphe visualise l'ensemble des armes de poing Enregsitré en moyenne dans chaque États

[108]:
```
df.groupby('state').mean().long_gun
```

[108]:
```
state
Alabama                  8657.000000
Alaska                   3655.200000
California              17932.800000
Colorado                14152.872727
Connecticut              2339.600000
Delaware                 2242.333333
District of Columbia      205.800000
Florida                 17932.800000
Georgia                 12929.000000
Guam                      205.800000
Hawaii                    205.800000
```

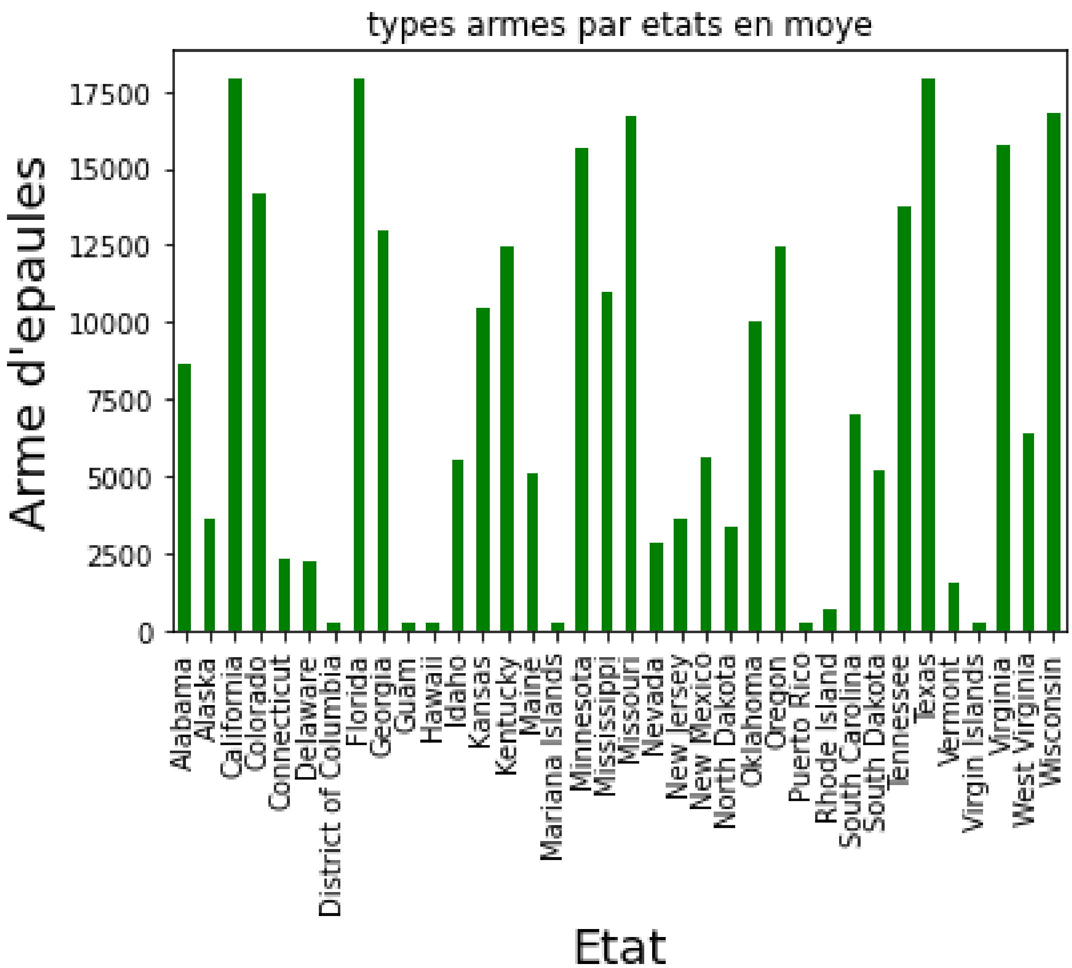
```
Idaho                      5508.500000
Kansas                    10473.000000
Kentucky                  12453.000000
Maine                      5097.000000
Mariana Islands             205.800000
Minnesota                 15648.450000
Mississippi               10942.400000
Missouri                  16730.400000
Nevada                     2875.500000
New Jersey                 3650.500000
New Mexico                 5571.500000
North Dakota               3381.200000
Oklahoma                  10003.000000
Oregon                    12458.500000
Puerto Rico                 215.276923
Rhode Island                687.000000
South Carolina             6997.000000
South Dakota               5177.800000
Tennessee                 13746.000000
Texas                     17932.800000
Vermont                    1524.200000
Virgin Islands              205.800000
Virginia                  15718.600000
West Virginia              6366.000000
Wisconsin                 16781.600000
Name: long_gun, dtype: float64
```

[109]:

```
long=df.groupby('state').mean().long_gun
long.plot(kind='bar',title='types armes par etats en␣
 ↪moye',color='green',alpha=1)
plt.xlabel('Etat',fontsize=18)
plt.ylabel('Arme d\'epaules',fontsize=18)
```

[109]:

```
Text(0, 0.5, "Arme d'epaules")
```



Ce graphe visualise l'ensemble des armes d'epaules Enregsitré en moyenne dans chaque États

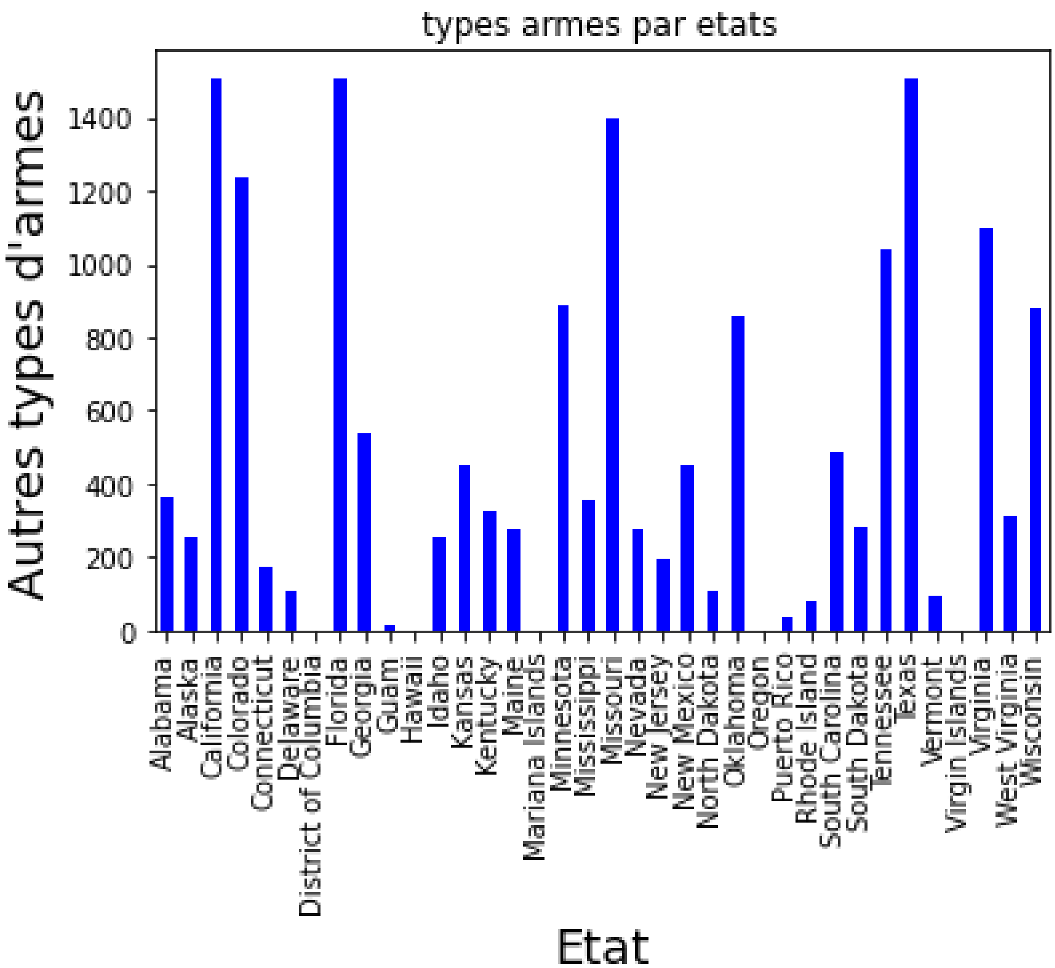
[110]:
```
df.groupby('state').mean().other
```

[110]:
```
state
Alabama                  361.000000
Alaska                   253.800000
California              1508.400000
Colorado                1234.290909
Connecticut              173.200000
Delaware                 106.333333
District of Columbia       0.000000
Florida                 1508.400000
Georgia                  534.666667
Guam                      11.714286
Hawaii                     0.000000
```

```
Idaho                  252.000000
Kansas                 450.000000
Kentucky               327.750000
Maine                  277.000000
Mariana Islands          0.285714
Minnesota              889.000000
Mississippi            352.000000
Missouri              1398.000000
Nevada                 273.357143
New Jersey             197.000000
New Mexico             448.750000
North Dakota           105.600000
Oklahoma               860.000000
Oregon                   0.000000
Puerto Rico             36.846154
Rhode Island            78.000000
South Carolina         488.500000
South Dakota           284.200000
Tennessee             1042.000000
Texas                 1508.400000
Vermont                 96.100000
Virgin Islands           0.071429
Virginia              1098.357143
West Virginia          312.500000
Wisconsin              882.333333
Name: other, dtype: float64
```

[111]:
```
other=df.groupby('state').mean().other
other.plot(kind='bar',title='types armes par etats',color='blue',alpha=1)
plt.xlabel('Etat',fontsize=18)
plt.ylabel('Autres types d\'armes',fontsize=18)
```

[111]:
```
Text(0, 0.5, "Autres types d'armes")
```



Ce graphe visualise les autres types d'armes Enregsitré dans chaque États en moyennne.

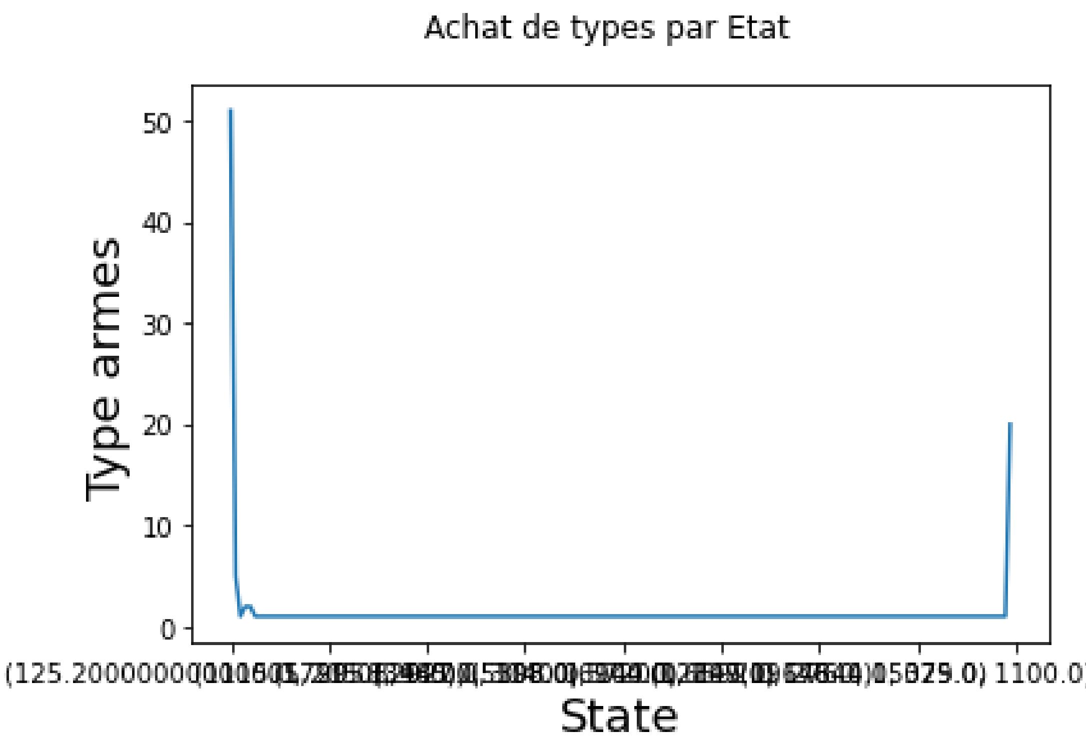
Ces graphes montre que les États comme **California,Colorado, Florida, Missouri, Texas,Tennessee,Virginia,Wisconsin** Ont connu le grandes nombres d'enregistrement d'armes de toutes types.

[112]:
```
a=df.groupby(['handgun','long_gun','other']).count()['state']
a
```

[112]:
```
handgun  long_gun  other
125.2    205.8     0.0       51
                   1.0        5
                   5.0        1
                   7.0        2
                   8.0        2
                             ..
28476.0  16844.0   1227.0     1
```

```
          17754.0   1193.0     1
          17932.8   1298.0     1
                    1308.0     1
                    1508.4    20
Name: state, Length: 160, dtype: int64
```

[113]:
```python
a.plot()
plt.suptitle('Achat de types par Etat')
plt.xlabel('State',fontsize=18)
plt.ylabel('Type armes',fontsize=18)
plt.show()
```

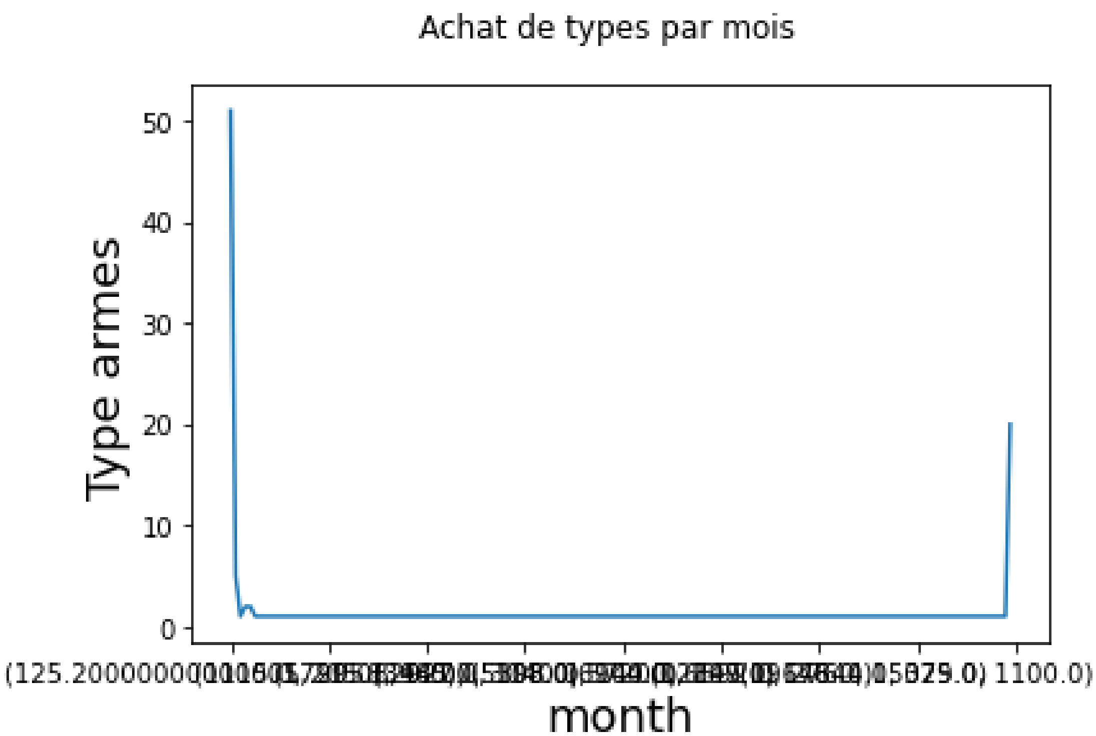


[118]:
```python
b=df.groupby(['handgun','long_gun','other']).count()['month']
b
```

[118]:
```
handgun  long_gun  other
125.2    205.8     0.0       51
                   1.0        5
                   5.0        1
                   7.0        2
                   8.0        2
                             ..
28476.0  16844.0   1227.0     1
```

```
          17754.0    1193.0     1
          17932.8    1298.0     1
                     1308.0     1
                     1508.4    20
Name: month, Length: 160, dtype: int64
```

[119]:
```python
b.plot()
plt.suptitle('Achat de types par mois')
plt.xlabel('month',fontsize=18)
plt.ylabel('Type armes',fontsize=18)
plt.show()
```

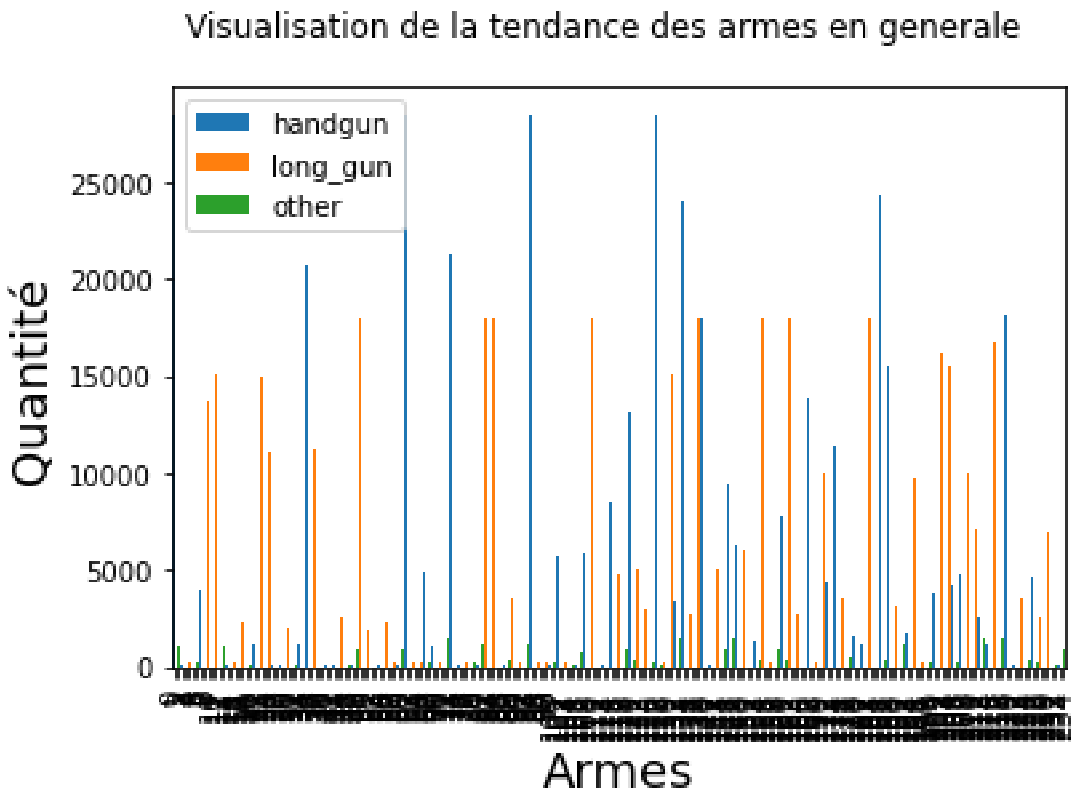


5.3 Quelle est la tendance générale des armes ?

[116]:
```python
df[['handgun','long_gun','other']].mean()
```

[116]:
```
handgun     7858.080000
long_gun    6021.217021
other        399.281702
dtype: float64
```

[117]:
```
df[['handgun','long_gun','other']].plot(kind='bar')
plt.suptitle('Visualisation de la tendance des armes en generale')
plt.xlabel('Armes',fontsize=18)
plt.ylabel('Quantité',fontsize=18)
plt.show()
```

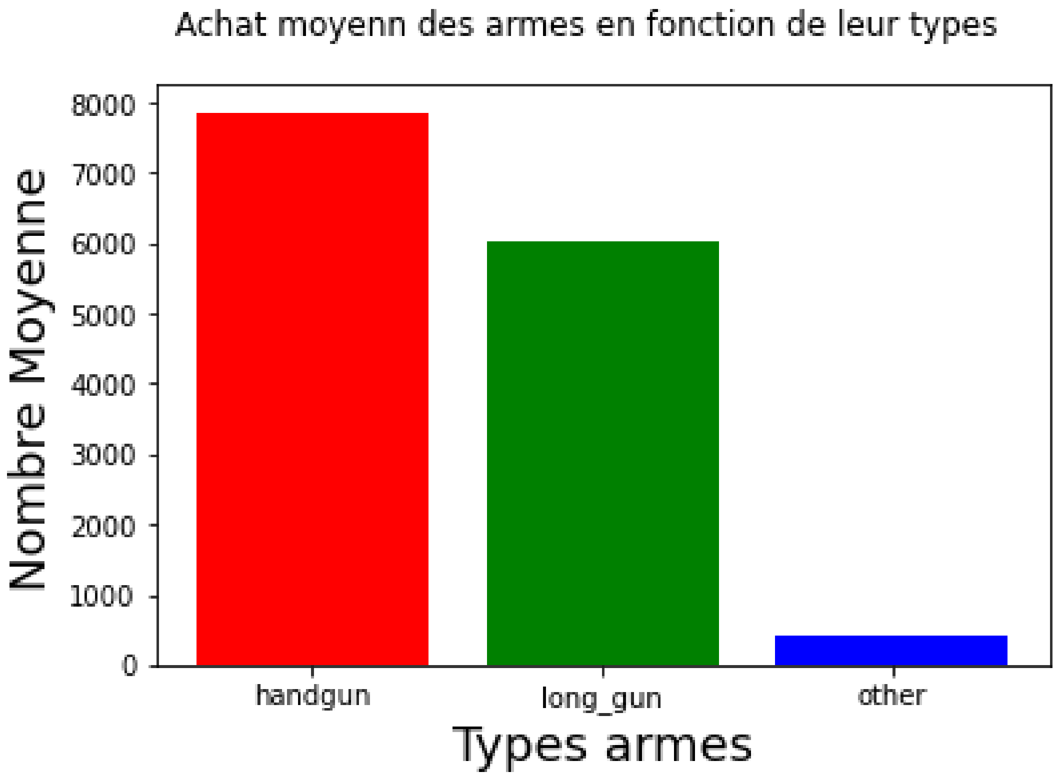


Ce Graphe explicite l'achat et le contrôle de l'ensemble des armes au sein du **NCIS** .Ceci montre la tendance generale des armes enregistrées

6 Conclusion

[120]:
```
# calcul la moyenne des types d'armes
lamda=np.array(df[['handgun','long_gun','other']].mean())
```

[121]:
```
plt.bar(['handgun','long_gun','other'],lamda,color=['red','green','blue'])
plt.suptitle('Achat moyenn des armes en fonction de leur types')
plt.xlabel('Types armes',fontsize=18)
plt.ylabel('Nombre Moyenne',fontsize=18)
plt.show()
```



L'etude montre que les types d'armes le plus achetés par les acheteurs sont les armes poing (**handgun**).Le diagramme en bar utilisé montre une visualisation clair et nette sur l'achat des types d'armes en moyenne.

[122]:
```
long = df["long_gun"].max()
long
```

[122]: 17932.8

[123]:
```
# utilisation de la fonction query() de pandas
df.query('long_gun == 17932.8')
```

[123]:

	month	state	permit	handgun	long_gun	other	multiple	\
0	2017-09	California	26413.7	28476.0	17932.8	1508.4	0.0	
12	2017-08	California	26413.7	28476.0	17932.8	1508.4	0.0	
24	2017-07	California	26413.7	28476.0	17932.8	1508.4	0.0	
36	2017-06	California	26413.7	28476.0	17932.8	1508.4	0.0	
49	2017-05	California	26413.7	28476.0	17932.8	1508.4	0.0	
61	2017-04	California	26413.7	28476.0	17932.8	1508.4	0.0	
72	2017-03	California	26413.7	28476.0	17932.8	1508.4	0.0	
82	2017-02	California	26413.7	28476.0	17932.8	1508.4	0.0	
84	2017-02	Florida	23617.0	28476.0	17932.8	1508.4	973.4	

```
94   2017-01  California  26413.7  28476.0   17932.8  1508.4       0.0
110  2016-12  California  26413.7  28476.0   17932.8  1508.4       0.0
111  2016-12    Colorado   5333.0  25905.0   17932.8  1428.0     973.4
127  2016-12       Texas  24494.0  28476.0   17932.8  1508.4     973.4
130  2016-12    Virginia   1145.0  28476.0   17932.8  1308.0       0.0
133  2016-11  California  26413.7  28476.0   17932.8  1508.4       0.0
134  2016-11    Colorado   7129.0  24052.0   17932.8  1508.4     973.4
138  2016-11     Florida  25377.0  28476.0   17932.8  1508.4     973.4
147  2016-11    Missouri   1504.0  23719.0   17932.8  1473.0     973.4
155  2016-11       Texas  23199.0  28476.0   17932.8  1508.4     973.4
158  2016-11    Virginia    695.0  28476.0   17932.8  1298.0       0.0
159  2016-11   Wisconsin  26413.7  17580.0   17932.8   883.0      34.0
162  2016-10  California  26413.7  28476.0   17932.8  1508.4       0.0
174  2016-10   Minnesota  26413.7  11394.0   17932.8  1024.0     511.0
183  2016-10       Texas  26413.7  28476.0   17932.8  1508.4     973.4
186  2016-10    Virginia    816.0  24337.0   17932.8  1269.0       0.0
191  2016-09  California  26413.7  28476.0   17932.8  1508.4       0.0
213  2016-09       Texas  26413.7  28476.0   17932.8  1508.4     973.4
218  2016-08  California  26413.7  28476.0   17932.8  1508.4       0.0

     prepawn_handgun  prepawn_long_gun  prepawn_other  redemption_handgun  \
0                0.0               0.0            0.0            6.282267
12               0.0               0.0            0.0            6.287859
24               0.0               0.0            0.0            5.973810
36               0.0               0.0            0.0            6.077642
49               0.0               0.0            0.0            6.070738
61               0.0               0.0            0.0            6.232448
72               0.0               0.0            0.0            6.415097
82               0.0               0.0            0.0            6.426488
84               6.0               3.0            0.0            7.417040
94               0.0               0.0            0.0            6.173786
110              0.0               0.0            0.0            6.463029
111              0.0               0.0            0.0            0.000000
127             12.0              12.1            1.0            7.417040
130              0.0               0.0            0.0            0.000000
133              0.0               0.0            0.0            6.253829
134              0.0               0.0            0.0            0.000000
138             11.0               5.0            0.0            7.417040
147              3.0              10.0            1.0            7.109879
155             12.0              12.1            1.0            7.417040
158              0.0               0.0            0.0            0.000000
159              0.0               0.0            1.0            4.844187
162              0.0               0.0            0.0            6.226537
174              1.0               2.0            0.0            5.347108
183             12.0              12.1            1.0            7.417040
186              0.0               0.0            0.0            0.000000
191              0.0               0.0            0.0            6.375025
```

213	12.0	12.1	1.0	7.417040
218	0.0	0.0	0.0	6.326149

	redemption_long_gun	redemption_other	returned_handgun	\
0	397.0	5.0	0.000000	
12	429.0	10.0	0.000000	
24	322.0	8.0	0.000000	
36	308.0	5.0	0.000000	
49	445.0	10.0	0.000000	
61	345.0	5.0	0.000000	
72	481.0	5.0	0.000000	
82	481.0	5.0	0.000000	
84	1467.6	10.0	4.891101	
94	381.0	7.0	0.000000	
110	831.0	10.0	0.000000	
111	0.0	0.0	4.891101	
127	1467.6	10.0	3.988984	
130	0.0	0.0	1.386294	
133	609.0	10.0	0.000000	
134	0.0	0.0	4.891101	
138	1428.0	7.0	4.891101	
147	1467.6	10.0	4.605170	
155	1467.6	10.0	3.891820	
158	0.0	0.0	0.693147	
159	477.0	9.0	3.496508	
162	534.0	6.0	0.000000	
174	667.0	10.0	2.197225	
183	1467.6	10.0	3.583519	
186	0.0	0.0	1.386294	
191	509.0	9.0	0.000000	
213	1467.6	10.0	3.465736	
218	580.0	10.0	0.000000	

	returned_long_gun	totals
0	0.000000	88253.0
12	0.000000	88253.0
24	0.000000	88253.0
36	0.000000	88253.0
49	0.000000	88253.0
61	0.000000	88253.0
72	0.000000	88253.0
82	0.000000	88253.0
84	3.367296	88253.0
94	0.000000	88253.0
110	0.000000	88253.0
111	3.367296	55428.0
127	0.693147	88253.0

```
130            0.000000  62649.0
133            0.000000  88253.0
134            3.367296  54418.0
138            3.367296  88253.0
147            1.945910  57501.0
155            0.693147  88253.0
158            0.000000  58362.0
159            3.295837  65277.0
162            0.000000  88253.0
174            2.197225  70854.0
183            0.000000  88253.0
186            1.098612  46862.0
191            0.000000  88253.0
213            0.693147  88253.0
218            0.000000  88253.0
```

[124]:
```
hand =df["handgun"].max()
hand
```

[124]:
```
28476.000000000004
```

[125]:
```
df.query('handgun == 28476.000000000004')
```

[125]:
```
       month       state   permit  handgun  long_gun  other  multiple  \
0    2017-09  California  26413.7  28476.0   17932.8  1508.4       0.0
12   2017-08  California  26413.7  28476.0   17932.8  1508.4       0.0
24   2017-07  California  26413.7  28476.0   17932.8  1508.4       0.0
36   2017-06  California  26413.7  28476.0   17932.8  1508.4       0.0
49   2017-05  California  26413.7  28476.0   17932.8  1508.4       0.0
61   2017-04  California  26413.7  28476.0   17932.8  1508.4       0.0
71   2017-04    Virginia   1074.0  28476.0   16844.0  1227.0       0.0
72   2017-03  California  26413.7  28476.0   17932.8  1508.4       0.0
81   2017-03    Virginia    529.0  28476.0   17754.0  1193.0       0.0
82   2017-02  California  26413.7  28476.0   17932.8  1508.4       0.0
84   2017-02     Florida  23617.0  28476.0   17932.8  1508.4     973.4
94   2017-01  California  26413.7  28476.0   17932.8  1508.4       0.0
110  2016-12  California  26413.7  28476.0   17932.8  1508.4       0.0
127  2016-12       Texas  24494.0  28476.0   17932.8  1508.4     973.4
130  2016-12    Virginia   1145.0  28476.0   17932.8  1308.0       0.0
133  2016-11  California  26413.7  28476.0   17932.8  1508.4       0.0
138  2016-11     Florida  25377.0  28476.0   17932.8  1508.4     973.4
155  2016-11       Texas  23199.0  28476.0   17932.8  1508.4     973.4
158  2016-11    Virginia    695.0  28476.0   17932.8  1298.0       0.0
162  2016-10  California  26413.7  28476.0   17932.8  1508.4       0.0
183  2016-10       Texas  26413.7  28476.0   17932.8  1508.4     973.4
191  2016-09  California  26413.7  28476.0   17932.8  1508.4       0.0
213  2016-09       Texas  26413.7  28476.0   17932.8  1508.4     973.4
```

```
218  2016-08  California  26413.7  28476.0   17932.8  1508.4       0.0

     prepawn_handgun  prepawn_long_gun  prepawn_other  redemption_handgun  \
0                0.0               0.0            0.0            6.282267
12               0.0               0.0            0.0            6.287859
24               0.0               0.0            0.0            5.973810
36               0.0               0.0            0.0            6.077642
49               0.0               0.0            0.0            6.070738
61               0.0               0.0            0.0            6.232448
71               0.0               0.0            0.0            0.000000
72               0.0               0.0            0.0            6.415097
81               0.0               0.0            0.0            0.000000
82               0.0               0.0            0.0            6.426488
84               6.0               3.0            0.0            7.417040
94               0.0               0.0            0.0            6.173786
110              0.0               0.0            0.0            6.463029
127             12.0              12.1            1.0            7.417040
130              0.0               0.0            0.0            0.000000
133              0.0               0.0            0.0            6.253829
138             11.0               5.0            0.0            7.417040
155             12.0              12.1            1.0            7.417040
158              0.0               0.0            0.0            0.000000
162              0.0               0.0            0.0            6.226537
183             12.0              12.1            1.0            7.417040
191              0.0               0.0            0.0            6.375025
213             12.0              12.1            1.0            7.417040
218              0.0               0.0            0.0            6.326149

     redemption_long_gun  redemption_other  returned_handgun  \
0                  397.0               5.0          0.000000
12                 429.0              10.0          0.000000
24                 322.0               8.0          0.000000
36                 308.0               5.0          0.000000
49                 445.0              10.0          0.000000
61                 345.0               5.0          0.000000
71                   0.0               0.0          1.791759
72                 481.0               5.0          0.000000
81                   0.0               0.0          2.708050
82                 481.0               5.0          0.000000
84                1467.6              10.0          4.891101
94                 381.0               7.0          0.000000
110                831.0              10.0          0.000000
127               1467.6              10.0          3.988984
130                  0.0               0.0          1.386294
133                609.0              10.0          0.000000
138               1428.0               7.0          4.891101
155               1467.6              10.0          3.891820
```

158	0.0	0.0	0.693147
162	534.0	6.0	0.000000
183	1467.6	10.0	3.583519
191	509.0	9.0	0.000000
213	1467.6	10.0	3.465736
218	580.0	10.0	0.000000

	returned_long_gun	totals
0	0.000000	88253.0
12	0.000000	88253.0
24	0.000000	88253.0
36	0.000000	88253.0
49	0.000000	88253.0
61	0.000000	88253.0
71	0.000000	48631.0
72	0.000000	88253.0
81	1.386294	49368.0
82	0.000000	88253.0
84	3.367296	88253.0
94	0.000000	88253.0
110	0.000000	88253.0
127	0.693147	88253.0
130	0.000000	62649.0
133	0.000000	88253.0
138	3.367296	88253.0
155	0.693147	88253.0
158	0.000000	58362.0
162	0.000000	88253.0
183	0.000000	88253.0
191	0.000000	88253.0
213	0.693147	88253.0
218	0.000000	88253.0

Les statistiques montrent que les Etats **Texas, California,Colorado, Florida, Missouri,Tennessee,Virginia,Wisconsin,Minnesota…**, on connue une nombre importante d'enregistrement d'arme à feu par mois. Donc le taux de criminalité est tres importantes dans ces zones. L'analyse montre aussi que le nombre d'armes contrôlés aux sein du NCIS est tres éléves.

6.0.1 Limites

Dans notre processus d'analyse nous avons rencontré certaines probléme: Parexemple la correlation entre les variables du dataset et ceci peut avoir un impact sur l'analyse des l'ensembles des données.

[126]:
```
df.corr()
```

[126]:

```
                      permit   handgun  long_gun     other  multiple  \
permit              1.000000  0.672167  0.701614  0.671948  0.329030
handgun             0.672167  1.000000  0.953511  0.951049  0.404721
long_gun            0.701614  0.953511  1.000000  0.916299  0.492508
other               0.671948  0.951049  0.916299  1.000000  0.494089
multiple            0.329030  0.404721  0.492508  0.494089  1.000000
prepawn_handgun     0.344848  0.219758  0.320575  0.197066  0.574748
prepawn_long_gun    0.310000  0.192448  0.302260  0.175890  0.560207
prepawn_other       0.362310  0.198437  0.229225  0.178850  0.297128
redemption_handgun  0.598467  0.374575  0.465234  0.407584  0.470443
redemption_long_gun 0.583365  0.401815  0.504338  0.387496  0.598373
redemption_other    0.761509  0.530714  0.616228  0.551569  0.410624
returned_handgun    0.230721  0.464626  0.522184  0.538177  0.680665
returned_long_gun   0.192232  0.369076  0.444385  0.437713  0.576257
totals              0.891244  0.914011  0.921095  0.881746  0.443709

                     prepawn_handgun  prepawn_long_gun  prepawn_other  \
permit                      0.344848          0.310000       0.362310
handgun                     0.219758          0.192448       0.198437
long_gun                    0.320575          0.302260       0.229225
other                       0.197066          0.175890       0.178850
multiple                    0.574748          0.560207       0.297128
prepawn_handgun             1.000000          0.936910       0.444683
prepawn_long_gun            0.936910          1.000000       0.468616
prepawn_other               0.444683          0.468616       1.000000
redemption_handgun          0.647546          0.655950       0.360564
redemption_long_gun         0.903271          0.900747       0.471546
redemption_other            0.588582          0.588030       0.390034
returned_handgun            0.234309          0.228709       0.201285
returned_long_gun           0.052583          0.040400       0.072649
totals                      0.357116          0.323993       0.301990

                     redemption_handgun  redemption_long_gun  \
permit                         0.598467             0.583365
handgun                        0.374575             0.401815
long_gun                       0.465234             0.504338
other                          0.407584             0.387496
multiple                       0.470443             0.598373
prepawn_handgun                0.647546             0.903271
prepawn_long_gun               0.655950             0.900747
prepawn_other                  0.360564             0.471546
redemption_handgun             1.000000             0.797007
redemption_long_gun            0.797007             1.000000
redemption_other               0.740582             0.768540
returned_handgun               0.200241             0.255043
returned_long_gun              0.081903             0.087841
totals                         0.549923             0.585156
```

	redemption_other	returned_handgun	returned_long_gun	\
permit	0.761509	0.230721	0.192232	
handgun	0.530714	0.464626	0.369076	
long_gun	0.616228	0.522184	0.444385	
other	0.551569	0.538177	0.437713	
multiple	0.410624	0.680665	0.576257	
prepawn_handgun	0.588582	0.234309	0.052583	
prepawn_long_gun	0.588030	0.228709	0.040400	
prepawn_other	0.390034	0.201285	0.072649	
redemption_handgun	0.740582	0.200241	0.081903	
redemption_long_gun	0.768540	0.255043	0.087841	
redemption_other	1.000000	0.243092	0.138849	
returned_handgun	0.243092	1.000000	0.865363	
returned_long_gun	0.138849	0.865363	1.000000	
totals	0.700204	0.395774	0.308967	

	totals
permit	0.891244
handgun	0.914011
long_gun	0.921095
other	0.881746
multiple	0.443709
prepawn_handgun	0.357116
prepawn_long_gun	0.323993
prepawn_other	0.301990
redemption_handgun	0.549923
redemption_long_gun	0.585156
redemption_other	0.700204
returned_handgun	0.395774
returned_long_gun	0.308967
totals	1.000000

Ici on constate la correlation avec les variables du dataset est tres importantes et cela peu impacter sur les predictions à venir

6.1 Réferences

Pandas

Numpy

Matplotlib

Seaborn


```
[ ]:
```