In [1]:

```
from sklearn.linear_model import LinearRegression
import pandas as pd
import numpy as np
import seaborn as sns
import scipy
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:

```
#list(pd.read_csv("region2013.csv", sep=";").columns.values)
```

In [3]:

```
dfUS13=pd.read_csv("C2013.csv", sep=";")
dfUS14=pd.read_csv("C2014.csv", sep=";")
dfNY13=pd.read_csv("NY2013.csv", sep=";")
dfNY14=pd.read_csv("NY2014.csv", sep=";")
dfTx13=pd.read_csv("texas2013.csv", sep=";")
dfTx14=pd.read_csv("Texas2014.csv", sep=";")
dfRel13=pd.read_csv("Rel2013.csv", sep=";")
dfRel14=pd.read_csv("Rel2014.csv", sep=";")
dfregion13=pd.read_csv("region2013.csv", sep=";")
```

In [4]:

```
del dfUS13['Unnamed: 20']
del dfUS13[' ']
del dfUS13[ 'Unnamed: 22']
del dfUS13[ 'Unnamed: 23']
```

In [58]:

```
dfTx14
```

Out[58]:

	City	Population	Violent crime	Murder and nonnegligent manslaughter	Robbery	Aggravated assault	Property crime
0	Abernathy	2 801	6	0	0	3	36
1	Abilene	120 686	571	7	128	343	5 344
2	Addison	16 061	67	0	15	44	686
3	Alamo	19 054	145	1	13	125	1 244
4	Alamo Heights	7 673	5	0	1	3	257
5	Alice	19 689	143	1	6	128	1 029
6	Allen	93 889	73	0	19	31	1 128
7	Alpine	6 058	7	1	0	6	49
8	Alton	15 311	10	0	4	5	253
9	Alvarado	3 821	18	0	3	10	95
10	Alvin	25 252	67	1	17	28	697
11	Amarillo	197 724	1 344	7	283	826	9 343
12	Andrews	13 153	80	1	4	62	310
13	Angleton	19 020	52	1	4	34	421
14	Anna	9 851	4	0	0	4	140
15	Anson	2 325	9	1	3	3	58
16	Anthony	5 365	13	0	3	9	152
17	Aransas Pass	8 367	37	0	8	29	444
18	Arcola	1 637	3	2	0	1	17
19	Argyle	3 644	1	0	0	0	19
20	Arlington	382 976	1 854	13	493	1 142	13 462
21	Arp	986	0	0	0	0	7
22	Athens	12 823	44	1	6	23	541
23	Atlanta	5 624	22	0	2	13	272
24	Aubrey	2 742	8	0	0	6	37
25	Austin	903 924	3 581	32	873	2 105	37 444
26	Azle	11 426	39	1	8	30	424
27	Baird	1 498	5	1	1	3	34
28	Balch Springs	25 305	233	2	30	186	1 079
29	Balcones	2 873	21	0	3	15	450

	Heights						
...	...	...	...	...	...	...	...
594	West University Place	15 509	2	0	1	1	169
595	Westworth	2 615	8	0	5	3	244
596	Wharton	8 738	43	0	8	35	207
597	Whitehouse	8 009	18	1	0	12	92
598	White Oak	6 372	4	0	0	4	76
599	Whitesboro	3 864	12	0	2	9	44
600	White Settlement	16 855	30	0	9	21	491
601	Whitewright	1 613	11	0	0	10	28
602	Whitney	2 081	3	0	1	2	64
603	Wichita Falls	104 949	426	2	130	207	4 258
604	Willis	6 127	47	0	8	30	173
605	Willow Park	4 595	1	0	0	1	46
606	Wills Point	3 511	5	0	1	4	69
607	Wilmer	3 829	9	0	3	6	95
608	Windcrest	5 668	8	0	3	2	326
609	Wink	1 024	0	0	0	0	11
610	Winnsboro	3 356	25	0	0	25	71
611	Winters	2 505	2	0	0	2	28
612	Wolfforth	4 070	2	0	0	1	34
613	Woodbranch	1 367	0	0	0	0	15
614	Woodville	2 495	3	0	0	3	21
615	Woodway	8 716	8	0	1	6	131
616	Wortham	1 053	1	0	0	1	23
617	Wylie	45 323	43	1	2	23	496
618	Yoakum	5 992	12	0	3	4	127
619	Yorktown	2 121	18	0	2	13	29
620	NaN	NaN	NaN	NaN	NaN	NaN	NaN
621	NaN	NaN	NaN	NaN	NaN	NaN	NaN
622	NaN	NaN	NaN	NaN	NaN	NaN	NaN
623		NaN	NaN	NaN	NaN	NaN	NaN

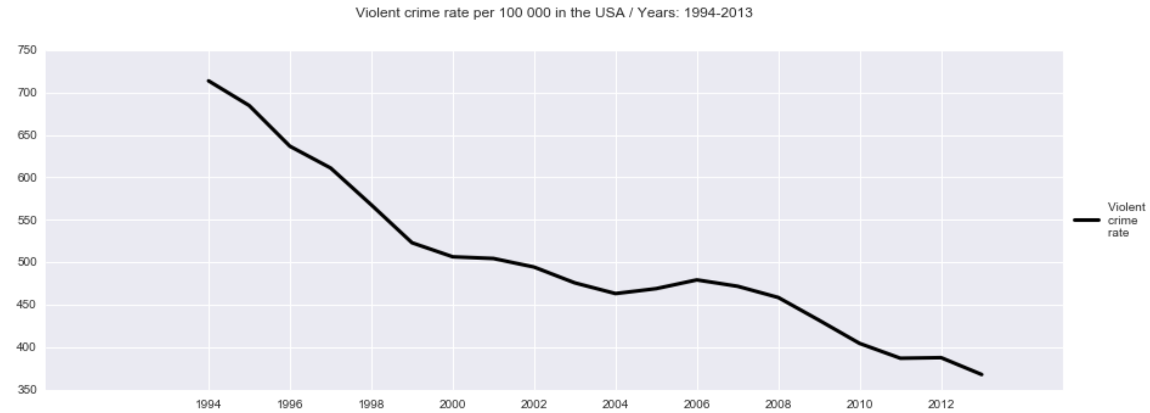
624 rows × 12 columns

In [6]:

```
dfUS13 = dfUS13.dropna()
dfUS13 = dfUS13.reset_index(drop=True)
kolumny13 =dfUS13.columns.values

#konwertuję zawartości komórek na liczby, by móc na nich dalej pracować

for col in kolumny13:
    for i in range(len(dfUS13)):
        if type(dfUS13.loc[i, col]) is str:
            dfUS13.loc[i, col] = dfUS13.loc[i, col].replace(" ", "").replace(",", ".")
        dfUS13.loc[i, col] = float(dfUS13.loc[i, col])

    dfUS13[[col]] = dfUS13[[col]].astype(float)
```

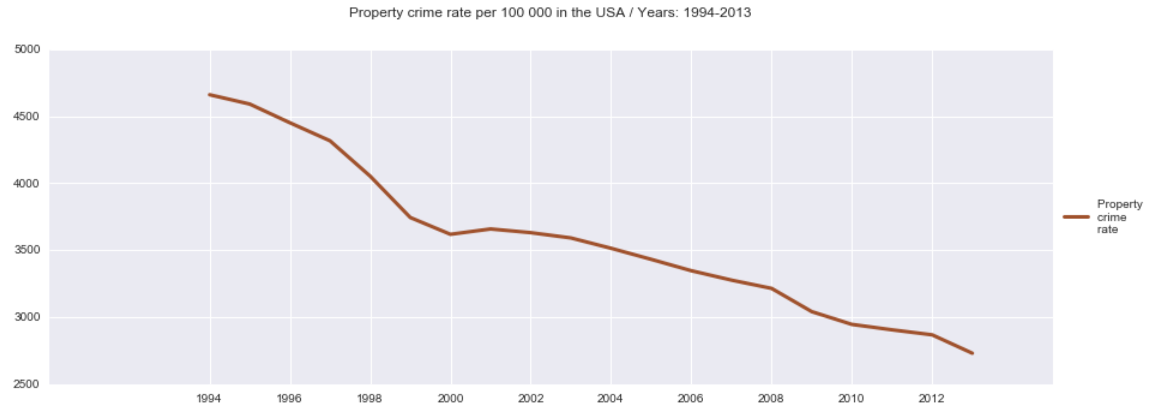
In [7]:

```
plt.figure(figsize=(15,5))
plt.plot(dfUS13['Year'], dfUS13['Violent \ncrime \nrate '], color = 'black', linewidth
= 3)
plt.xticks(np.arange(min(dfUS13['Year']), max(dfUS13['Year'])+1, 2.0))
plt.title('Violent crime rate per 100 000 in the USA / Years: 1994-2013', y=1.08)
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.show()
```



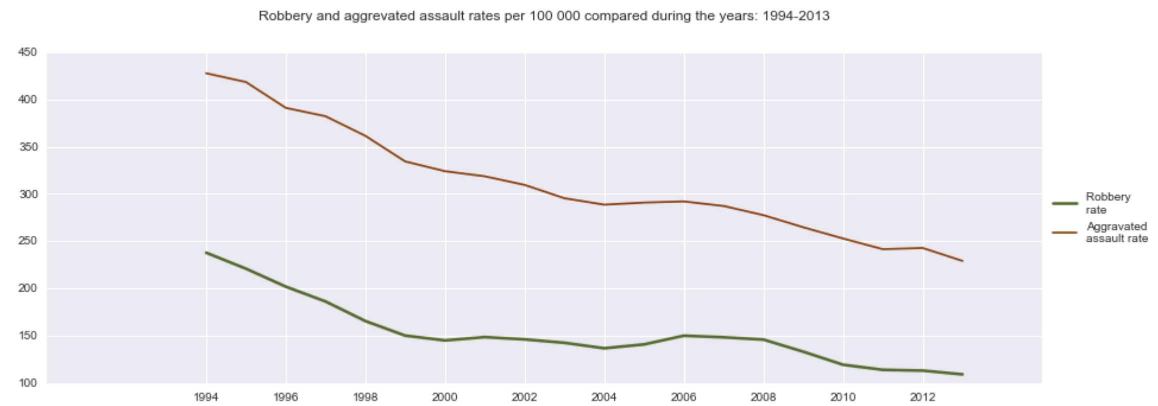
In [8]:

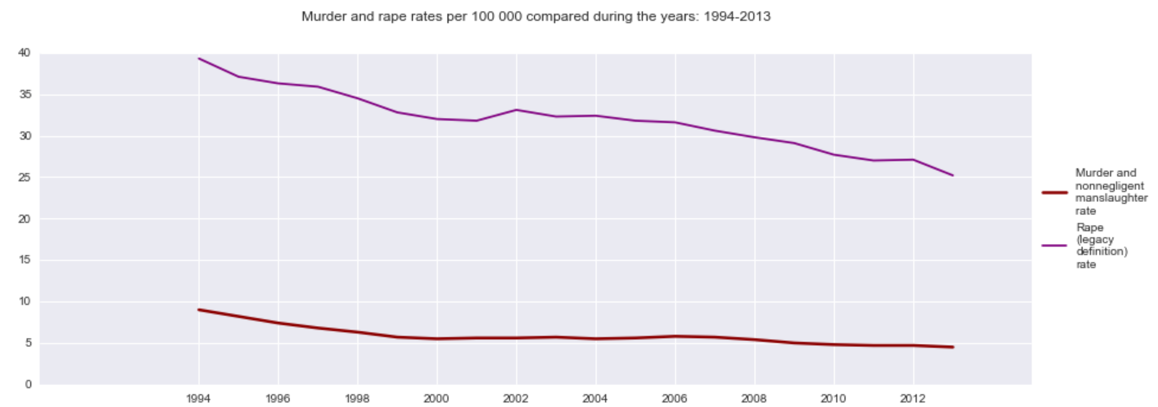
```
plt.figure(figsize=(15,5))
plt.plot(dfUS13['Year'], dfUS13['Property \ncrime \nrate '], color='sienna', linewidth
= 3)
plt.xticks(np.arange(min(dfUS13['Year']), max(dfUS13['Year'])+1, 2.0))
plt.title('Property crime rate per 100 000 in the USA / Years: 1994-2013', y=1.08)
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.show()
```



In [9]:

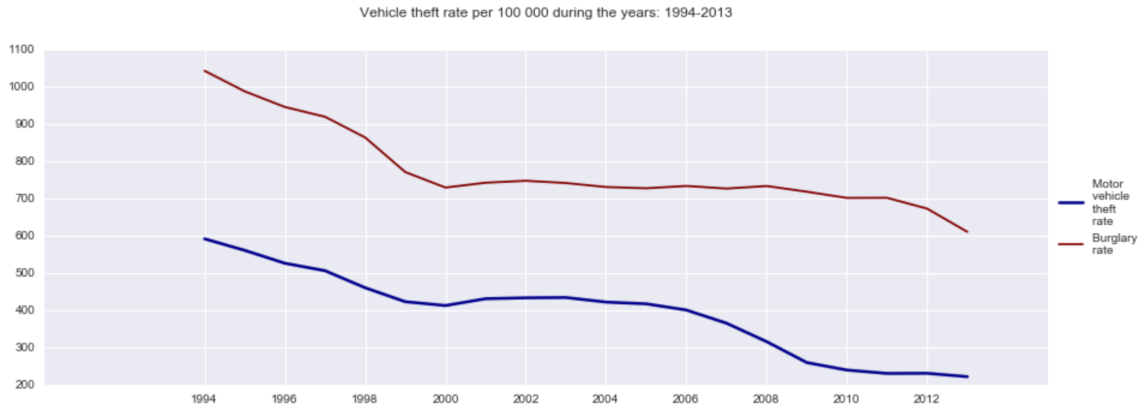
```
plt.figure(figsize=(15,5))

plt.plot(dfUS13['Year'], dfUS13['Robbery \nrate '], color = 'darkolivegreen', linewidth
 = 2.5)
plt.plot(dfUS13['Year'], dfUS13['Aggravated \nassault rate '], color = 'saddlebrown')
plt.xticks(np.arange(min(dfUS13['Year']), max(dfUS13['Year'])+1, 2.0))
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.title("Robbery and aggrevated assault rates per 100 000 compared during the years:
 1994-2013", y=1.08)
plt.show()

plt.figure(figsize=(15,5))
plt.plot(dfUS13['Year'], dfUS13['Murder and \nnonnegligent \nmanslaughter \nrate '], co
lor = 'darkred', linewidth = 2.5)
plt.plot(dfUS13['Year'], dfUS13['Rape\n(legacy\ndefinition)\nrate'], color = 'purple')
plt.xticks(np.arange(min(dfUS13['Year']), max(dfUS13['Year'])+1, 2.0))
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.title("Murder and rape rates per 100 000 compared during the years: 1994-2013",
y=1.08)
plt.show()
```





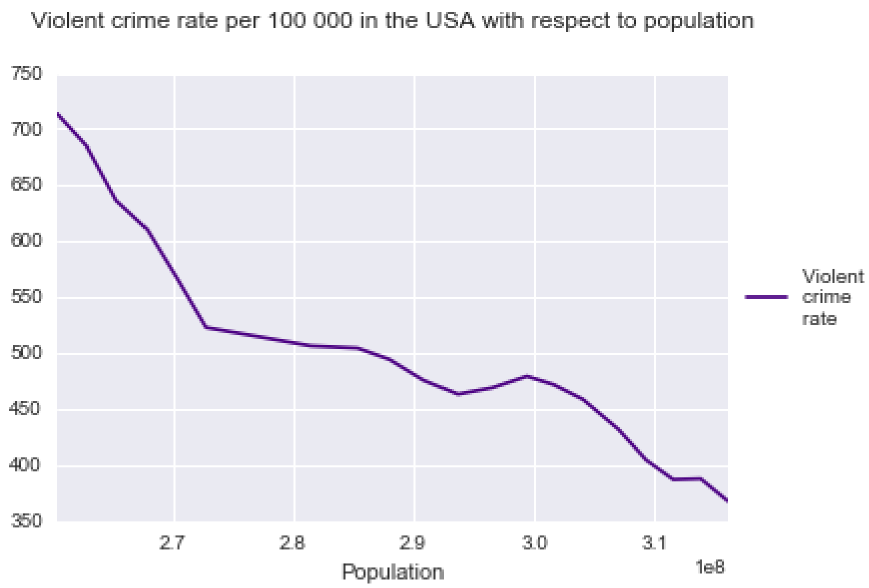
In [10]:

```
plt.figure(figsize=(15,5))
plt.plot(dfUS13['Year'], dfUS13['Motor \nvehicle \ntheft \nrate '], color = 'darkblue',
 linewidth = 2.5)
plt.plot(dfUS13['Year'], dfUS13['Burglary \nrate '], color = 'maroon')
plt.xticks(np.arange(min(dfUS13['Year']), max(dfUS13['Year'])+1, 2.0))
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.title( 'Vehicle theft rate per 100 000 during the years: 1994-2013', y=1.08)

plt.show()
```



In [11]:

```
dfUS13.plot(x='Population', y='Violent \ncrime \nrate ', color='indigo')
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.title( 'Violent crime rate per 100 000 in the USA with respect to population', y=1.
08)
plt.show()
```

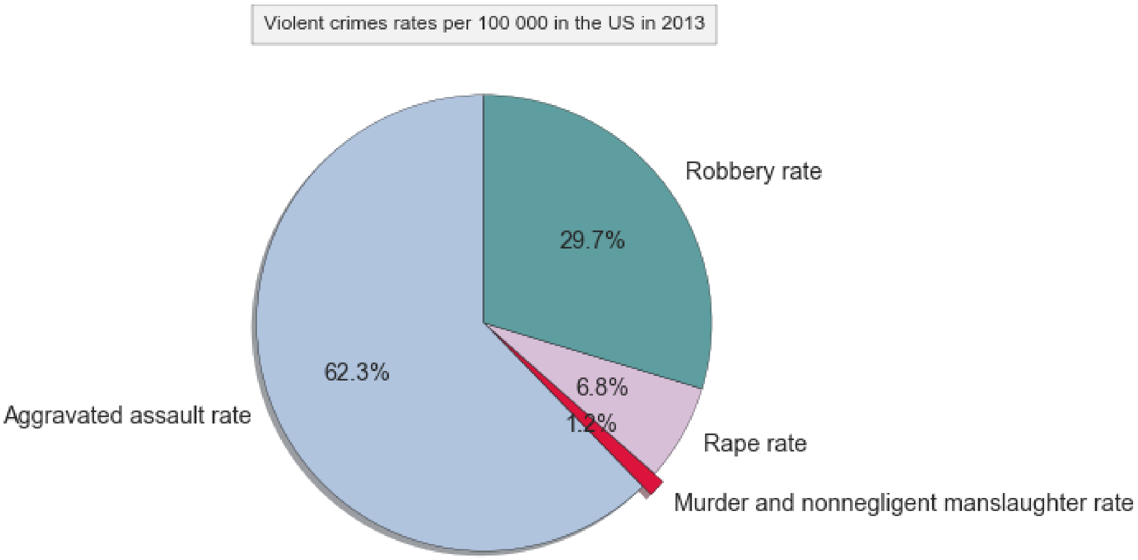


In [12]:

```
rokk = dfUS13.loc[19,:]
```

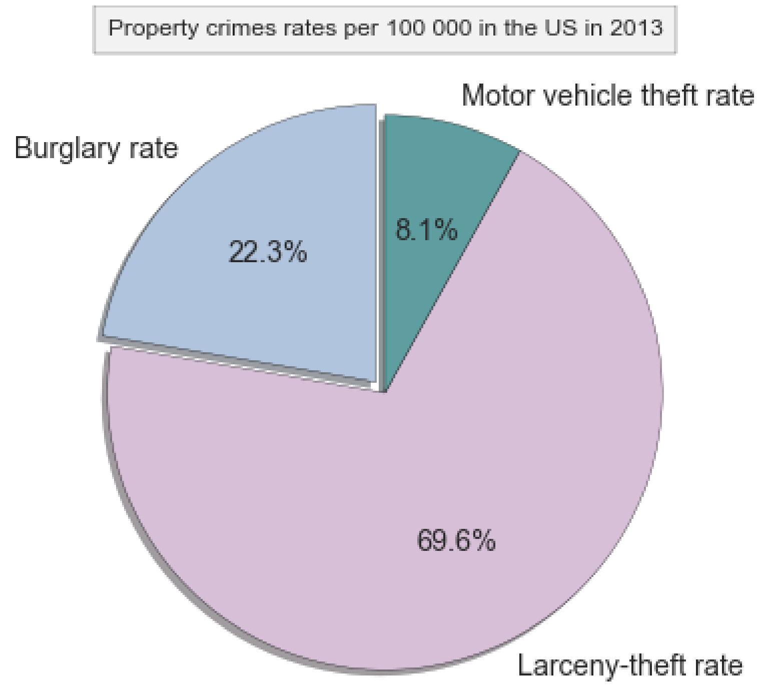
In [13]:

```
from pylab import *
from matplotlib import font_manager as fm


figure(1, figsize=(6,6))
ax = axes([0.1, 0.1, 0.8, 0.8])


labels = 'Aggravated assault rate', 'Murder and nonnegligent manslaughter rate', 'Rape
 rate', 'Robbery rate'
fracs = [rokk[11]/rokk[3], rokk[5]/rokk[3], rokk[7]/rokk[3], rokk[9]/rokk[3]]
colors = ['lightsteelblue', 'crimson', 'thistle', 'cadetblue']
explode=(0, 0.05, 0, 0)

patches, texts, autotexts = ax.pie(fracs, explode=explode, labels=labels, colors = color
s, autopct='%1.1f%%', shadow=True, startangle=90)

title('Violent crimes rates per 100 000 in the US in 2013', bbox={'facecolor':'0.95',
'pad':7})


proptease = fm.FontProperties()
proptease.set_size('x-large')
plt.setp(autotexts, fontproperties=proptease)
plt.setp(texts, fontproperties=proptease)

plt.show()
```



In [14]:

```
figure(1, figsize=(6,6))
ax = axes([0.1, 0.1, 0.8, 0.8])


labels = 'Burglary rate', 'Larceny-theft rate', 'Motor vehicle theft rate'
fracs = [rokk[15]/rokk[13], rokk[17]/rokk[13], rokk[19]/rokk[13]]
colors = ['lightsteelblue', 'thistle', 'cadetblue']
explode=(0.05, 0, 0)

patches, texts, autotexts = ax.pie(fracs, explode=explode, labels=labels, colors = color
s, autopct='%1.1f%%', shadow=True, startangle=90)

title('Property crimes rates per 100 000 in the US in 2013', bbox={'facecolor':'0.95',
'pad':7})


proptease = fm.FontProperties()
proptease.set_size('x-large')
plt.setp(autotexts, fontproperties=proptease)
plt.setp(texts, fontproperties=proptease)

plt.show()
```



In [8]:

```
dfNY14 = dfNY14.dropna()
dfNY14 = dfNY14.reset_index(drop=True)
dfNY14.columns.values

kolumny = dfNY14.columns.values[1:]

for col in kolumny:
    for i in range(len(dfNY14)):
        if type(dfNY14.loc[i, col]) is str:
            dfNY14.loc[i, col] = dfNY14.loc[i, col].replace(" ", "")
        dfNY14.loc[i, col] = int(dfNY14.loc[i, col])

    dfNY14[[col]] = dfNY14[[col]].astype(int)
```

In [9]:

```
#dfNY14.dtypes
```

In [7]:

```
#print dfregion13.columns.values

dfregion13 = dfregion13.dropna()
dfregion13 = dfregion13.reset_index(drop=True)
dfregion13.columns.values

kolumny = dfregion13.columns.values[1:]

for col in kolumny:
    for i in range(len(dfregion13)):
        if type(dfregion13.loc[i, col]) is str:
            dfregion13.loc[i, col] = dfregion13.loc[i, col].replace(" ",
"").replace(",", ".")
        dfregion13.loc[i, col] = float(dfregion13.loc[i, col])

    dfregion13[[col]] = dfregion13[[col]].astype(float)
```

In [18]:

```
del dfRel13['Unnamed: 20']
del dfRel13[ 'Unnamed: 22']
del dfRel13[ 'Unnamed: 21']

dfRel13 = dfRel13.dropna()
dfRel13 = dfRel13.reset_index(drop=True)

#dfRel13.columns.values
```

In [10]:

```
#rozwazania dotyczace poszegolnych stanow (dfregion13)

#przekształcamy data frame w array

r13 = np.array(dfregion13)

# dostosowujemy wymiary tablic:
print
print  "MODEL 1: Population - Violent crime number"
X_train1 = r13[1:40,2].reshape(39,1)
y_train1 = r13[1:40,3].reshape(39,1)
X_test1 = r13[41:,2].reshape(10,1)
y_test1 = r13[41:,3].reshape(10,1)

mdl1 = LinearRegression().fit(X_train1,y_train1)
mdl1.predict(X_test1)

print "Wynik dla zbioru trenującego:"
print mdl1.score(X_train1,y_train1)
print "Wynik dla danych testowych:"
print mdl1.score(X_test1,y_test1)
print
print "MODEL 2: Pozwolenie na broń - violent crime rate"
X_train2 = r13[1:40,1].reshape(39,1)
y_train2 = r13[1:40,4].reshape(39,1)
X_test2 = r13[41:,1].reshape(10,1)
y_test2 = r13[41:,4].reshape(10,1)

mdl2 = LinearRegression().fit(X_train2,y_train2)
mdl2.predict(X_test2)

print "Wynik (testu R^2) dla zbioru trenującego:"
print mdl2.score(X_train2,y_train2)
print "Wynik dla danych testowych:"
print mdl2.score(X_test2,y_test2)
print
print "MODEL 3: Population - violent crime rate"
X_train3 = r13[1:40,2].reshape(39,1)
y_train3 = r13[1:40,4].reshape(39,1)
X_test3 = r13[41:,2].reshape(10,1)
y_test3 = r13[41:,4].reshape(10,1)

mdl3 = LinearRegression().fit(X_train3,y_train3)
mdl3.predict(X_test3)

print "Wynik dla zbioru trenującego:"
print mdl3.score(X_train3,y_train3)
print "Wynik dla danych testowych:"
print mdl3.score(X_test3,y_test3)

print
print "MODEL 4: Population - Larceny-theft rate"
X_train4 = r13[1:40,2].reshape(39,1)
y_train4 = r13[1:40,20].reshape(39,1)
X_test4 = r13[41:,2].reshape(10,1)
y_test4 = r13[41:,20].reshape(10,1)

mdl4 = LinearRegression().fit(X_train4,y_train4)
mdl4.predict(X_test4)

print "Wynik dla zbioru trenującego:"
```

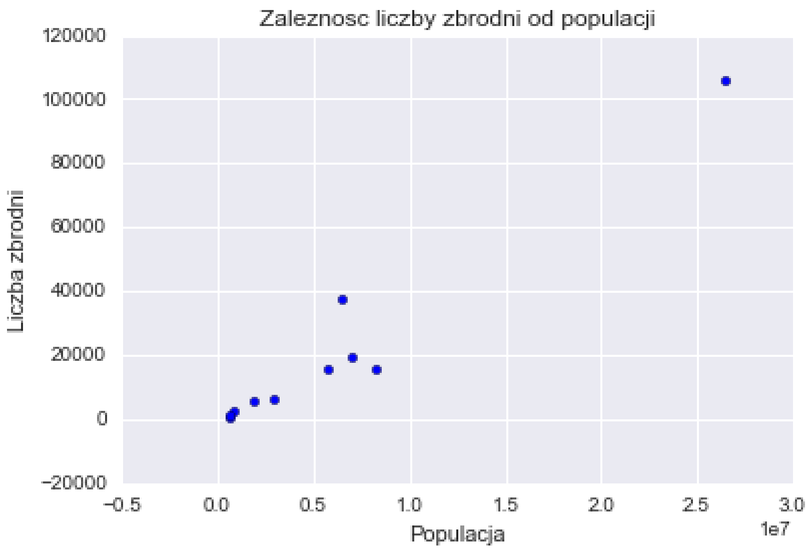
```
print mdl4.score(X_train4,y_train4)
print "Wynik dla danych testowych:"
print mdl4.score(X_test4,y_test4)
```

```
MODEL 1: Population - Violent crime number
Wynik dla zbioru trenującego:
0.969156540491
Wynik dla danych testowych:
0.94291475945

MODEL 2: Pozwolenie na broń - violent crime rate
Wynik (testu R^2) dla zbioru trenującego:
0.0090059104575
Wynik dla danych testowych:
-0.366347961263

MODEL 3: Population - violent crime rate
Wynik dla zbioru trenującego:
4.06537699627e-06
Wynik dla danych testowych:
-0.509830133196

MODEL 4: Population - Larceny-theft rate
Wynik dla zbioru trenującego:
0.0493176366619
Wynik dla danych testowych:
-0.409395132479
```

In [11]:

```
plt.scatter(list(X_test1[:,0]),list(y_test1))
plt.xlabel('Populacja')
plt.ylabel('Liczba zbrodni')
plt.title('Zaleznosc liczby zbrodni od populacji')
plt.legend()
plt.show()
print mdl1.score(X_train1,y_train1)
print mdl1.score(X_test1,y_test1)
print "Zatem jest zależność"
```

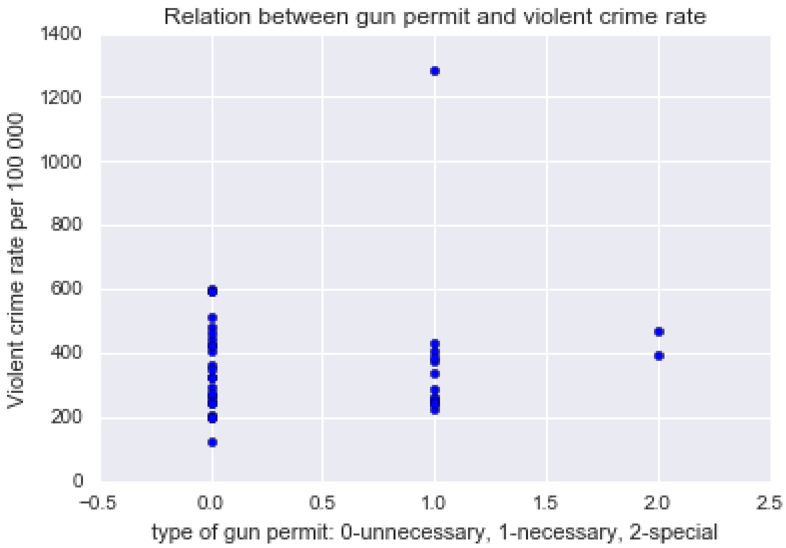
```
C:\Users\Edyta\Anaconda2\lib\site-packages\matplotlib\axes\_axes.py:519: U
serWarning: No labelled objects found. Use label='...' kwarg on individual
 plots.
  warnings.warn("No labelled objects found. "
```




```
0.969156540491
0.94291475945
Zatem jest zależność
```

In [12]:

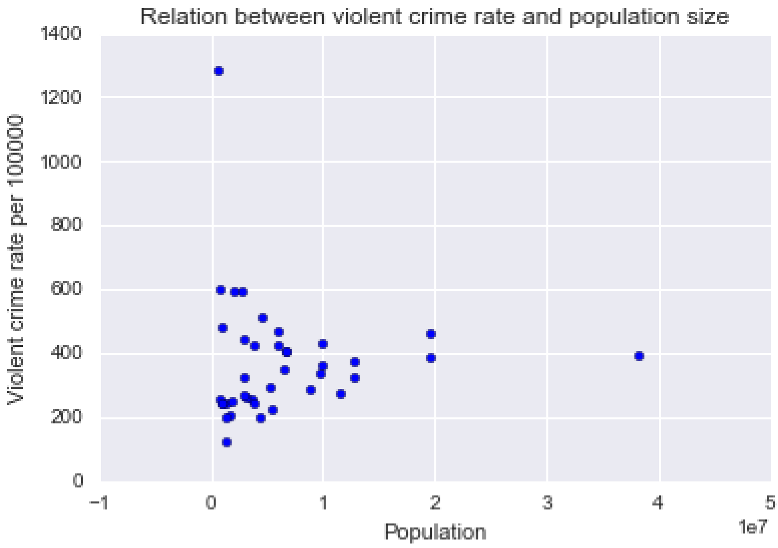
```
plt.scatter(list(X_train2[:,0]),list(y_train2))
plt.xlabel('type of gun permit: 0-unnecessary, 1-necessary, 2-special')
plt.ylabel('Violent crime rate per 100 000')
plt.title('Relation between gun permit and violent crime rate')
plt.legend()
plt.show()
print mdl2.score(X_train2,y_train2)
print mdl2.score(X_test2,y_test2)
print "Czyli nie ma zależności"
```




```
0.0090059104575
-0.366347961263
Czyli nie ma zależności
```

In [22]:

```
plt.scatter(list(X_train3[:,0]),list(y_train3))
plt.xlabel('Population')
plt.ylabel('Violent crime rate per 100000')
plt.title('Relation between violent crime rate and population size')
plt.legend()
plt.show()
print mdl3.score(X_train3,y_train3)
print mdl3.score(X_test3,y_test3)
print "Nie widać zaleznosci"
```




```
4.06537699627e-06
-0.509830133196
Nie widać zaleznosci
```

In [23]:

```
plt.scatter(list(X_train4[:,0]),list(y_train4))
plt.xlabel('Population')
plt.ylabel('Larceny theft rate per 100 000')
plt.title('Relation between larceny theft rate and population')
plt.legend()
plt.show()
print mdl4.score(X_train4,y_train4)
print mdl4.score(X_test4,y_test4)
print " Nie ma zaleznosci"
```


```
0.0493176366619
-0.409395132479
 Nie ma zaleznosci
```

In [24]:

```
dfTx13.head()
```

Out[24]:

	City	Population	Violent crime	Murder and nonnegligent manslaughter	Robbery	Aggravated assault	Property crime	Burg
0	Abernathy	2 821	0	0	0	0	12	12
1	Abilene	119 401	477	1	125	314	4 769	1 05
2	Addison	15 961	51	1	11	35	784	129
3	Alamo	18 876	164	0	27	126	1 336	203
4	Alamo Heights	7 443	9	0	2	5	235	36

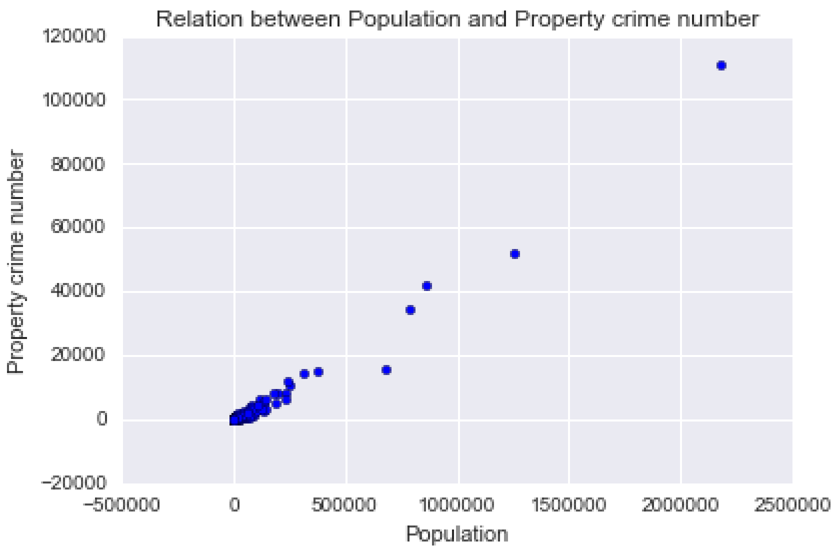
In [25]:

```
#rozwazania dotyczace Teksasu w latach 2013 i 2014 (dfTx13 i dfTx14)
t13 = np.array(dfTx13)
for i in xrange(604):
    t13[i,1] = t13[i,1].replace(" ", "")
    t13[i,6] = t13[i,6].replace(" ", "")


print "MODEL 5: Population and Property crime"
X_train5 = t13[1:400,1].reshape(399,1)
y_train5 = t13[1:400,6].reshape(399,1)
X_test5 = t13[401:,1].reshape(203,1)
y_test5 = t13[401:,6].reshape(203,1)
mdl5 = LinearRegression().fit(X_train5,y_train5)
mdl5.predict(X_test5)

print "Wynik dla zbioru trenującego:"
print mdl5.score(X_train5,y_train5)
print "Wynik dla danych testowych:"
print mdl5.score(X_test5,y_test5)

print

print "MODEL 6: Murder and nonnegligent manslaughter and Rape"
X_train6 = t13[1:400,3].reshape(399,1)
y_train6 = t13[1:400,11].reshape(399,1)
X_test6 = t13[401:,3].reshape(203,1)
y_test6 = t13[401:,11].reshape(203,1)
mdl6 = LinearRegression().fit(X_train6,y_train6)
mdl6.predict(X_test6)
print "Wynik dla zbioru trenującego:"
print mdl6.score(X_train6,y_train6)
print "Wynik dla danych testowych:"
print mdl6.score(X_test6,y_test6)
```

```
MODEL 5: Population and Property crime
Wynik dla zbioru trenującego:
0.971308324014
Wynik dla danych testowych:
0.948736877894

MODEL 6: Murder and nonnegligent manslaughter and Rape
Wynik dla zbioru trenującego:
0.771155434948
Wynik dla danych testowych:
0.599751291892
```

In [26]:

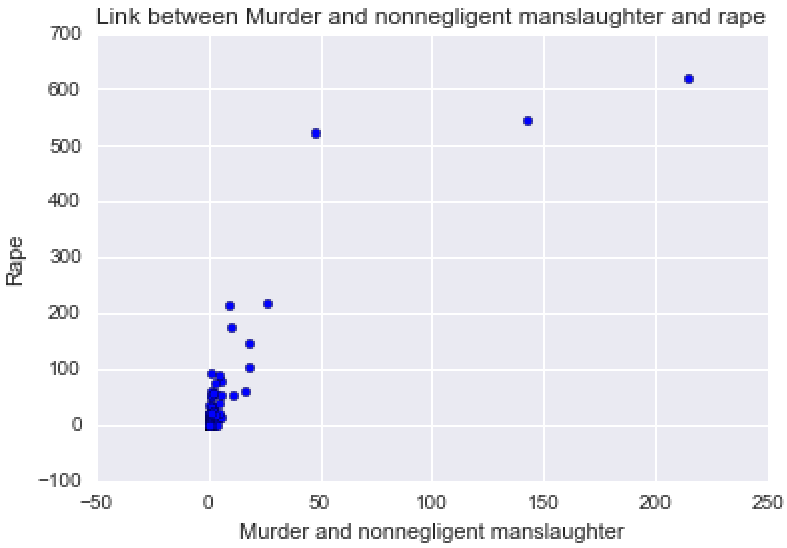
```
plt.scatter(list(X_train5[:,0]),list(y_train5))
plt.xlabel('Population')
plt.ylabel('Property crime number')
plt.title('Relation between Population and Property crime number')
plt.legend()
plt.show()
print mdl5.score(X_train5,y_train5)
print mdl5.score(X_test5,y_test5)
print "W końcu jest zależność :)"
```




```
0.971308324014
0.948736877894
W końcu jest zależność :)
```

In [27]:

```
plt.scatter(list(X_train6[:,0]),list(y_train6))
plt.xlabel('Murder and nonnegligent manslaughter')
plt.ylabel('Rape')
plt.title('Link between Murder and nonnegligent manslaughter and rape')
plt.legend()
plt.show()
print mdl6.score(X_train6,y_train6)
print mdl6.score(X_test6,y_test6)
print "Jest slaba zaleznosc :| "
```



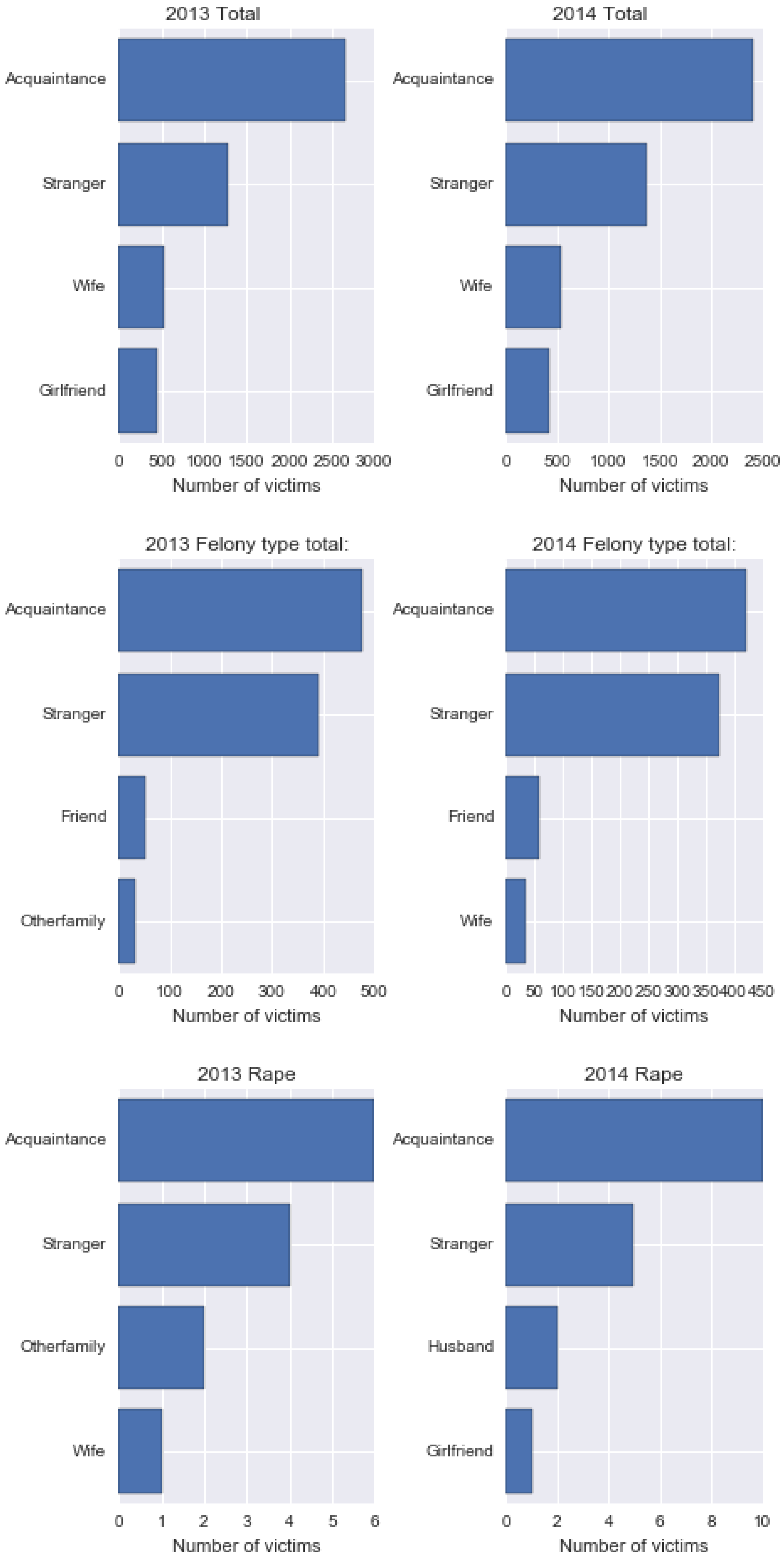

```
0.771155434948
0.599751291892
Jest slaba zaleznosc :|
```

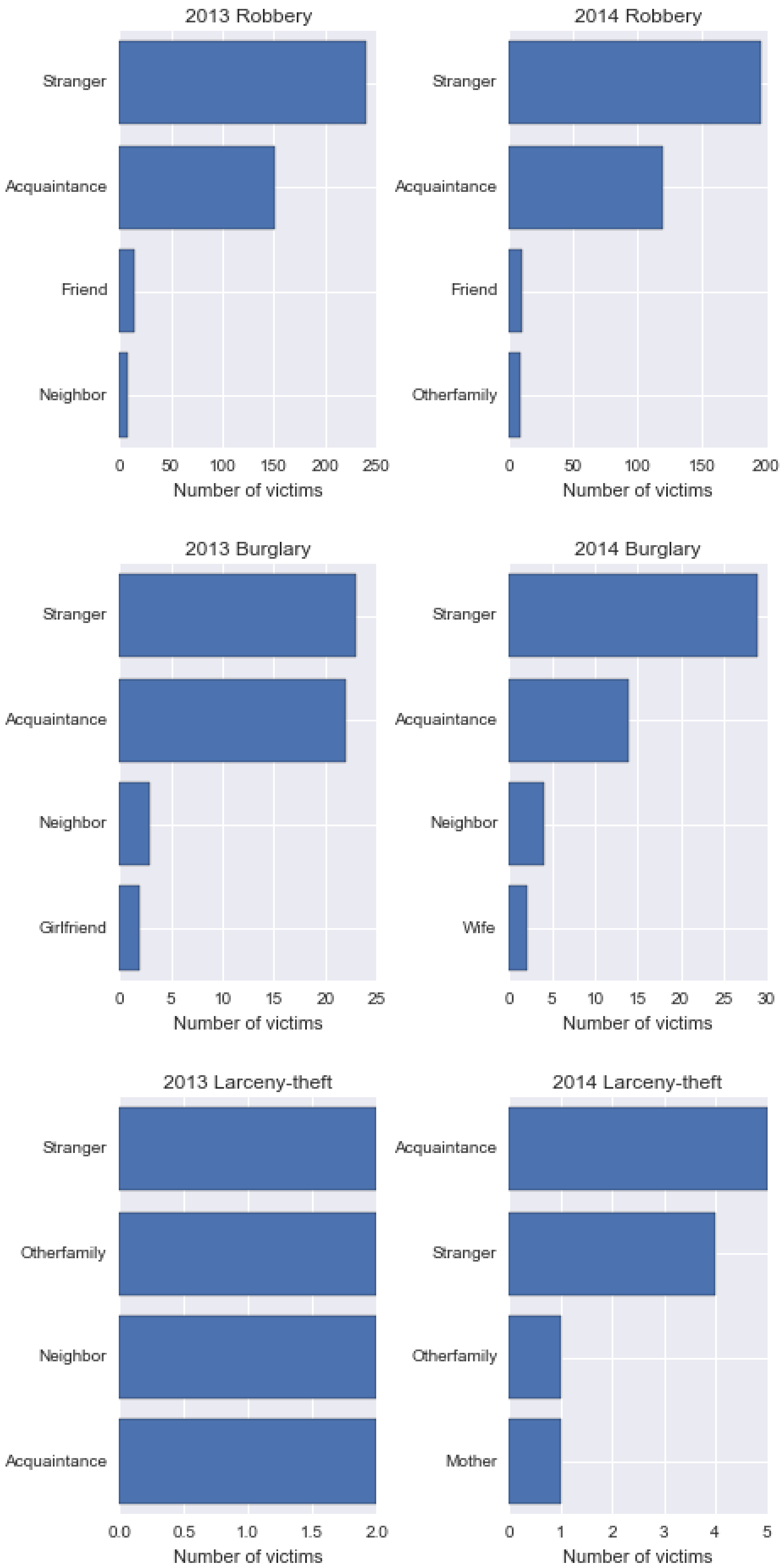
In [5]:

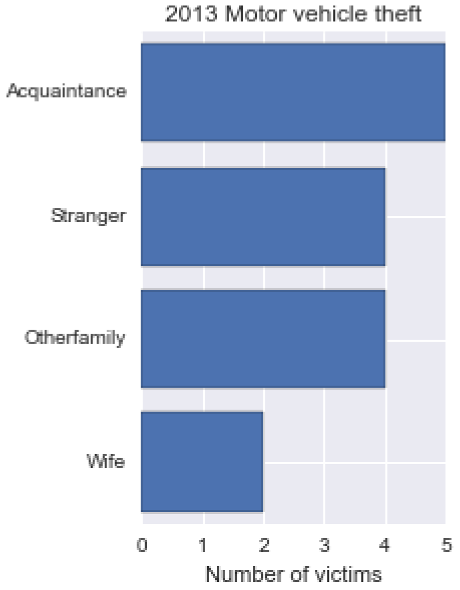
```
import csv

def prepare_data(name, isMax):
    with open(name, 'rb') as f:
        reader = csv.reader(f)
        a=True
        Head=[]
        Return_Table=[]
        for row in reader:
            row_table=row[0].split(";")
            tuple_table=[]
            if a:
                Head=row[0].split(";")
                a=False
                continue
            for i in xrange(2,19):
                tuple_table.append((int(row_table[i].replace(" ","")),Head[i]))
            if isMax:
                tuple_table.sort(reverse=True)
            else:
                tuple_table.sort()
            Return_Table.append((row_table[0], tuple_table[:4]))

        return Return_Table

def create_chart(Ret1,Ret2,isMax):

    for i in xrange(0,17):
        if (Ret1[i][1][3][0]==0 and Ret1[i][1][2][0]==0 and Ret1[i][1][1][0]==0 and Ret
1[i][1][0][0]==0
        and Ret2[i][1][3][0]==0 and Ret2[i][1][2][0]==0 and Ret2[i][1][1][0]==0 and Ret
2[i][1][0][0]==0):
            return
        f, (ax1,ax2) = plt.subplots(1,2, sharey=False)
        objects=(Ret1[i][1][3][1],Ret1[i][1][2][1],Ret1[i][1][1][1],Ret1[i][1][0][1])
        y_pos = np.arange(len(objects))
        quantity = (Ret1[i][1][3][0],Ret1[i][1][2][0],Ret1[i][1][1][0],Ret1[i][1][0][0])


        plt.tight_layout(w_pad=5)

        ax1.barh(y_pos, quantity, align='center', alpha=1)
        ax1.set_yticks(y_pos)
        ax1.set_yticklabels(objects)
        ax1.set_xlabel("Number of victims")
        ax1.set_title('2013'+" "+Ret1[i][0])

        objects=(Ret2[i][1][3][1],Ret2[i][1][2][1],Ret2[i][1][1][1],Ret2[i][1][0][1])
        y_pos = np.arange(len(objects))
        quantity = (Ret2[i][1][3][0],Ret2[i][1][2][0],Ret2[i][1][1][0],Ret2[i][1][0][0])


        ax2.barh(y_pos, quantity, align='center', alpha=1)
        ax2.set_yticks(y_pos)
        ax2.set_yticklabels(objects)
        ax2.set_xlabel("Number of victims")
        ax2.set_title('2014'+" "+Ret2[i][0])

        plt.show()

Ret1=prepare_data('Rel2013.csv',True)
```

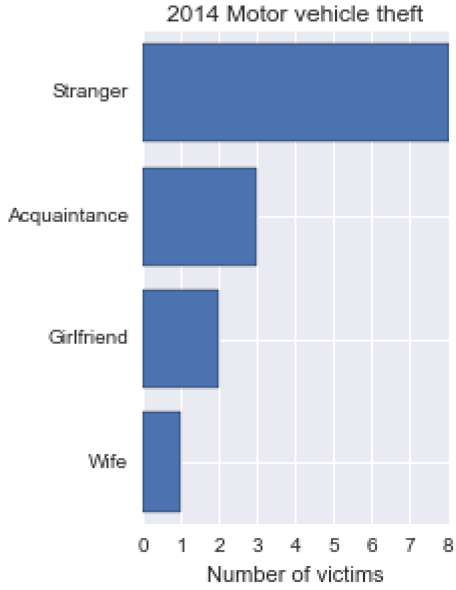
```
Ret2=prepare_data('Rel2014.csv',True)

print "The most frequent victims of murders in 2013-2014 and crime as a result of which
 there has been death"
create_chart(Ret1,Ret2,True)


Ret1=prepare_data('Rel2013.csv',False)
Ret2=prepare_data('Rel2014.csv',False)

print "Rarest victims of murders in 2013-2014 and crime as a result of which there has
 been death (non-zero)"
create_chart(Ret1,Ret2,False)
```

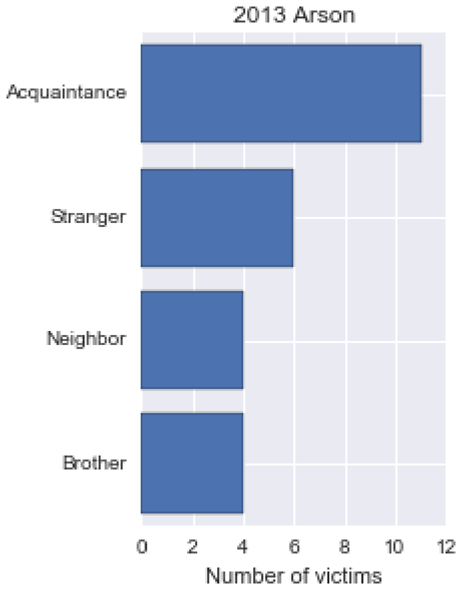
The most frequent victims of murders in 2013-2014 and crime as a result of which there has been death

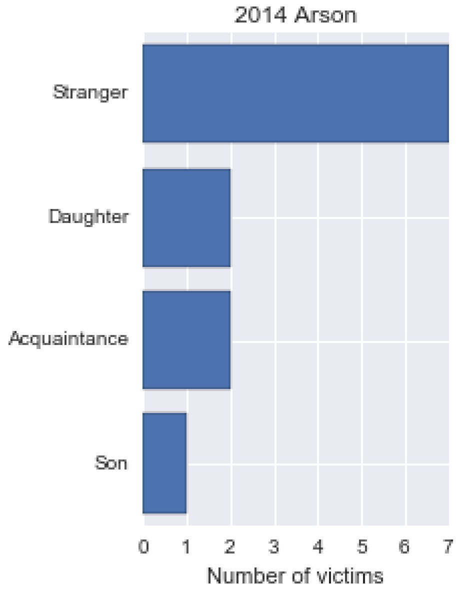


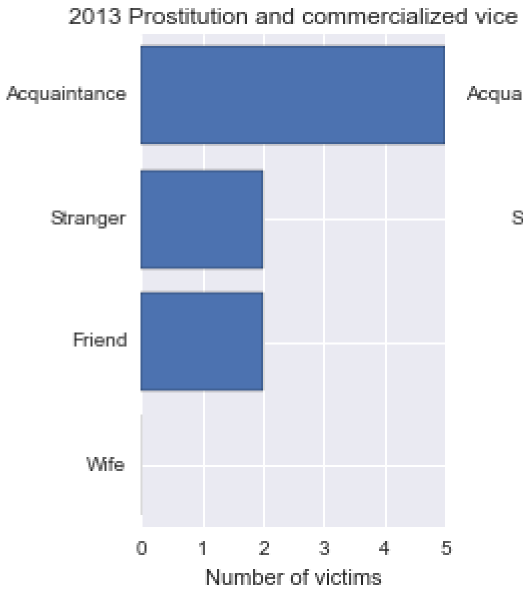

2013 Robbery
Stranger
Acquaintance
Friend
Neighbor
0 50 100 150 200 250
Number of victims
2014 Robbery
Stranger
Acquaintance
Friend
Otherfamily
0 50 100 150 200
Number of victims
2013 Burglary
Stranger
Acquaintance
Neighbor
Girlfriend
0 5 10 15 20 25
Number of victims
2014 Burglary
Stranger
Acquaintance
Neighbor
Wife
0 5 10 15 20 25 30
Number of victims
2013 Larceny-theft
Stranger
Otherfamily
Neighbor
Acquaintance
0.0 0.5 1.0 1.5 2.0
Number of victims
2014 Larceny-theft
Acquaintance
Stranger
Otherfamily
Mother
0 1 2 3 4 5
Number of victims

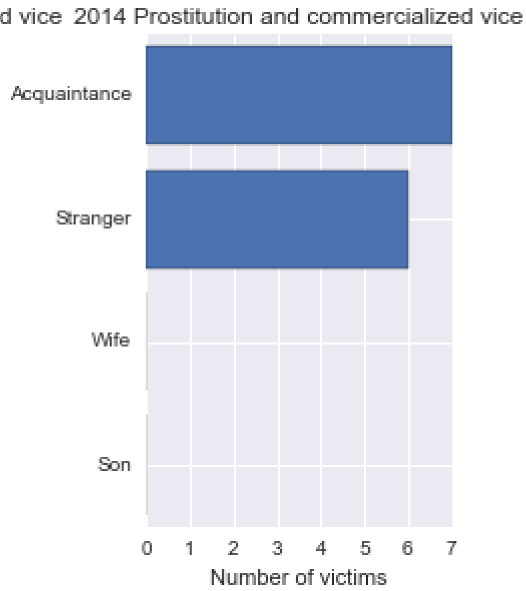

2013 Motor vehicle theft
Acquaintance
Stranger
Otherfamily
Wife
0 1 2 3 4 5
Number of victims

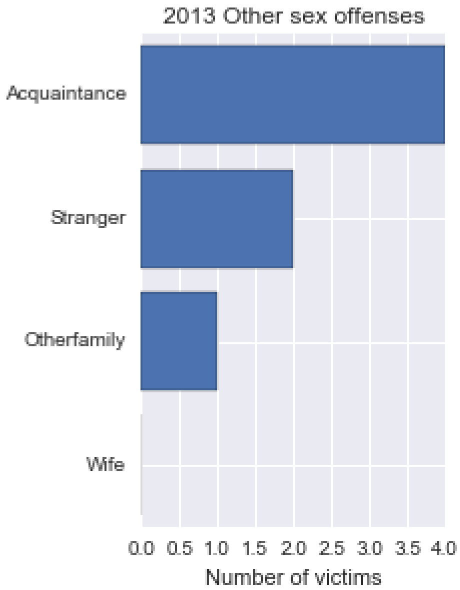

2014 Motor vehicle theft
Stranger
Acquaintance
Girlfriend
Wife
0 1 2 3 4 5 6 7 8
Number of victims

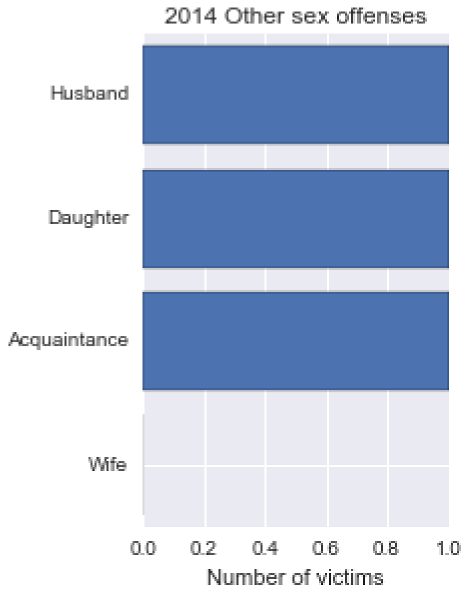

2013 Arson
Acquaintance
Stranger
Neighbor
Brother
0 2 4 6 8 10 12
Number of victims

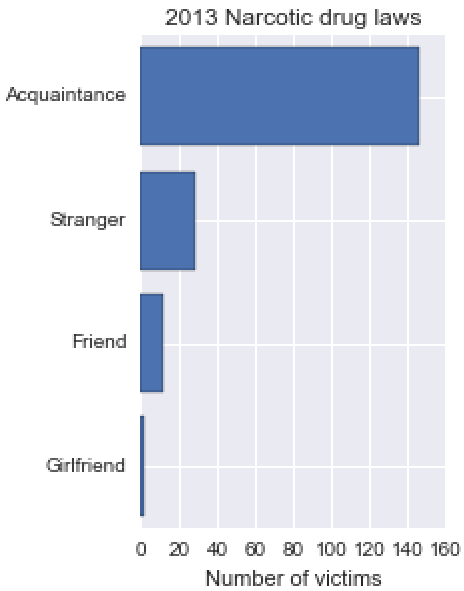

2014 Arson
Stranger
Daughter
Acquaintance
Son
0 1 2 3 4 5 6 7
Number of victims

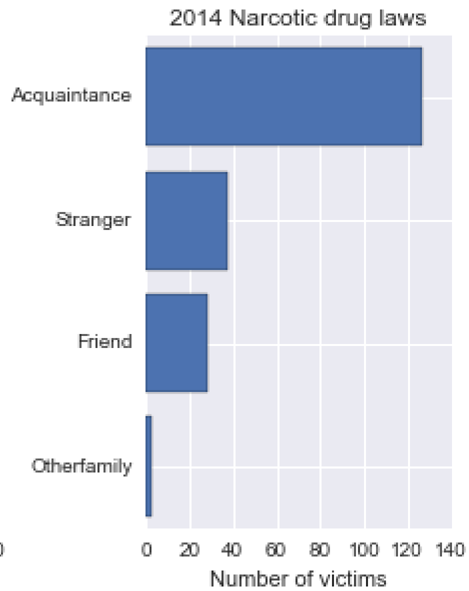

2013 Prostitution and commercialized vice
Acquaintance
Stranger
Friend
Wife
0 1 2 3 4 5
Number of victims

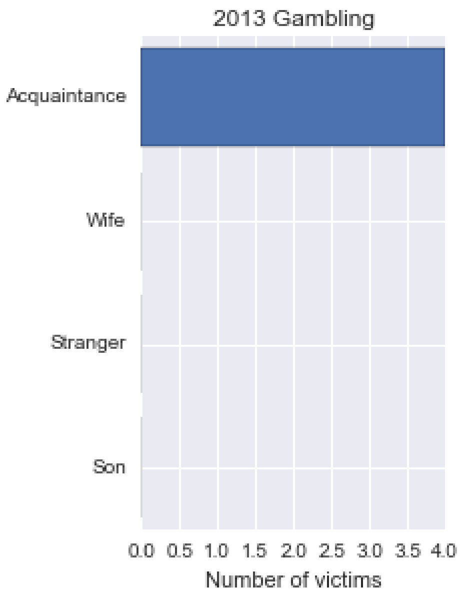

2014 Prostitution and commercialized vice
Acquaintance
Stranger
Wife
Son
0 1 2 3 4 5 6 7
Number of victims

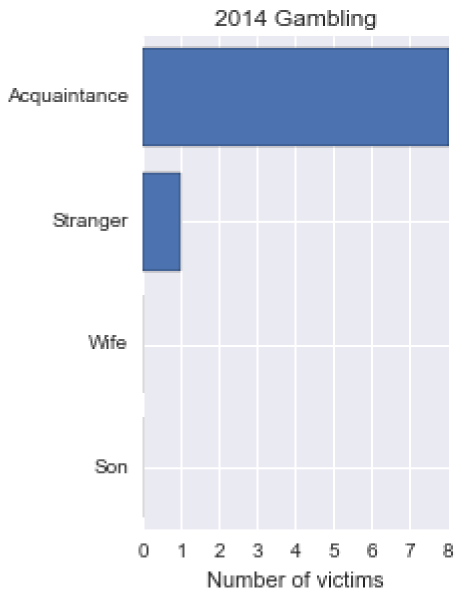

2013 Other sex offenses
Acquaintance
Stranger
Otherfamily
Wife
0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0
Number of victims

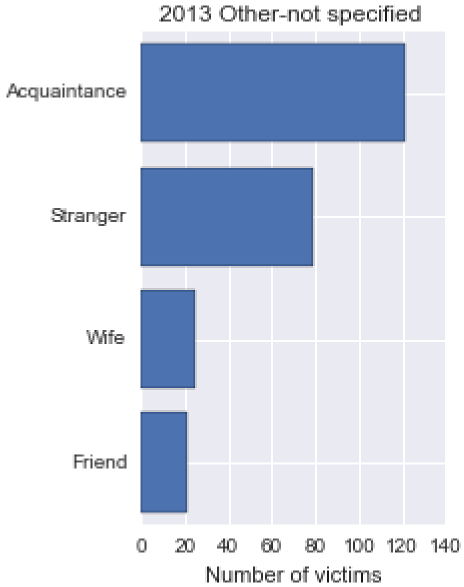

2014 Other sex offenses
Husband
Daughter
Acquaintance
Wife
0.0 0.2 0.4 0.6 0.8 1.0
Number of victims

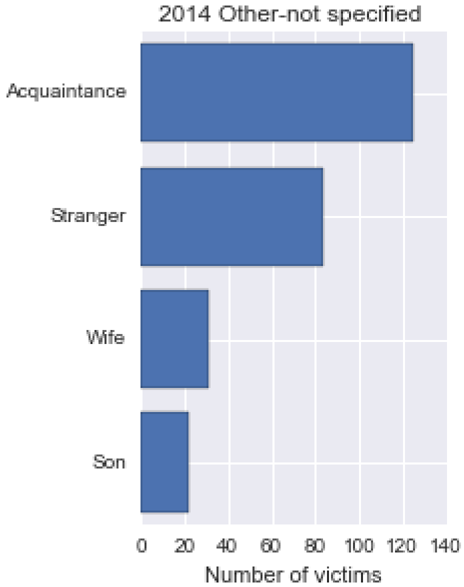

2013 Narcotic drug laws
Acquaintance
Stranger
Friend
Girlfriend
0 20 40 60 80 100 120 140 160
Number of victims

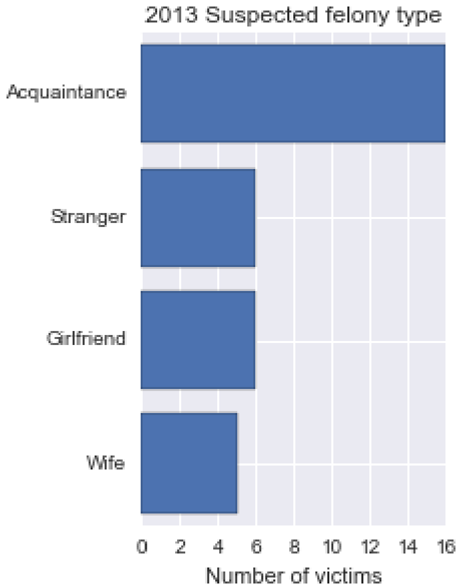

2014 Narcotic drug laws
Acquaintance
Stranger
Friend
Otherfamily
0 20 40 60 80 100 120 140
Number of victims

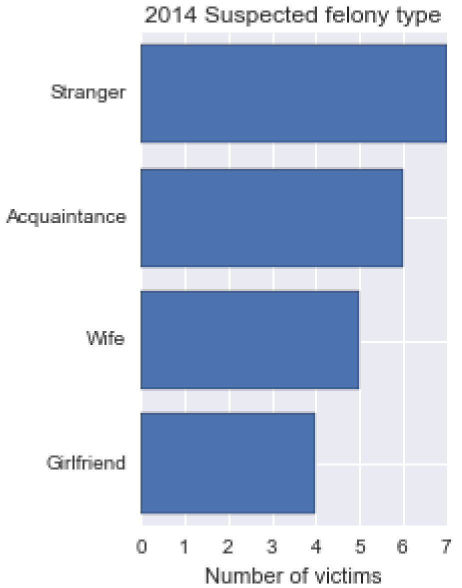

2013 Gambling
Acquaintance
Wife
Stranger
Son
0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0
Number of victims

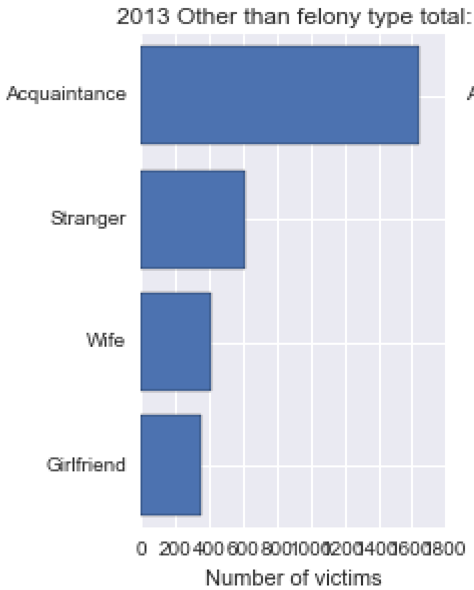

2014 Gambling
Acquaintance
Stranger
Wife
Son
0 1 2 3 4 5 6 7 8
Number of victims

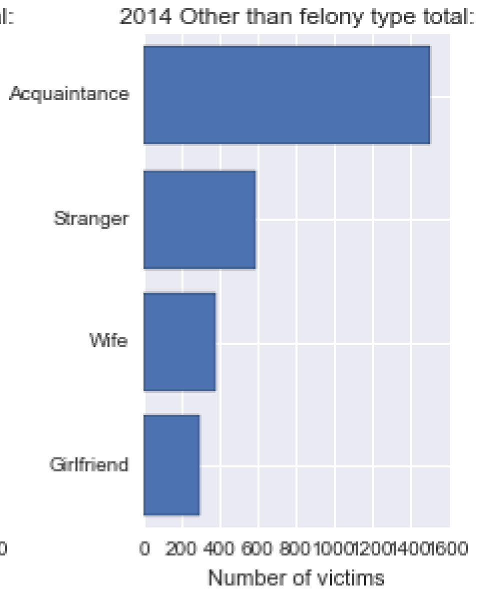

2013 Other-not specified
Acquaintance
Stranger
Wife
Friend
0 20 40 60 80 100 120 140
Number of victims

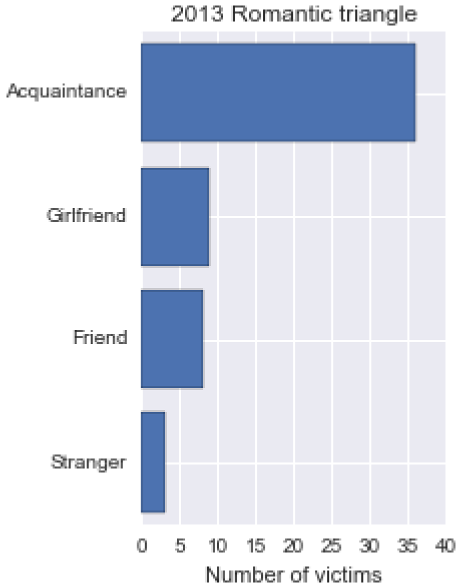

2014 Other-not specified
Acquaintance
Stranger
Wife
Son
0 20 40 60 80 100 120 140
Number of victims

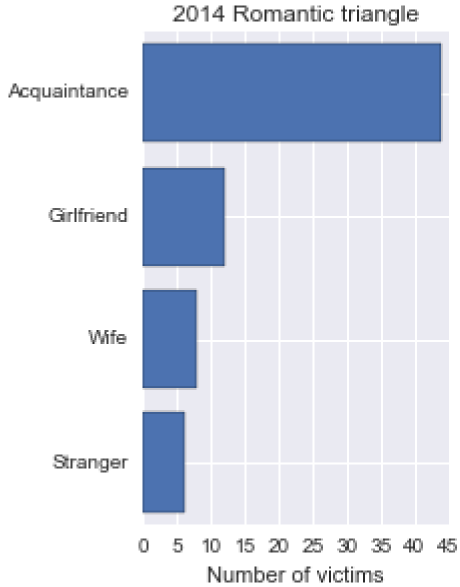

2013 Suspected felony type
Acquaintance
Stranger
Girlfriend
Wife
0 2 4 6 8 10 12 14 16
Number of victims

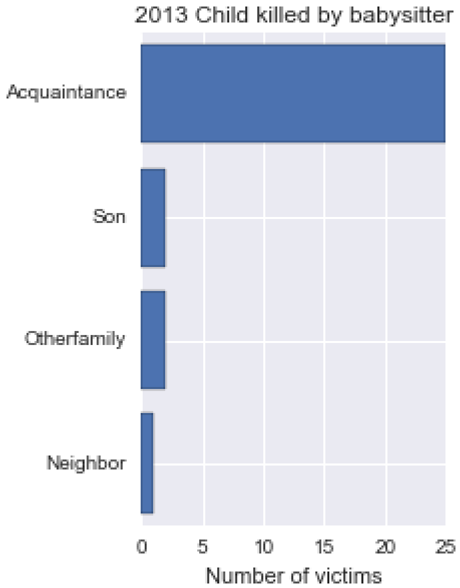

2014 Suspected felony type
Stranger
Acquaintance
Wife
Girlfriend
0 1 2 3 4 5 6 7
Number of victims

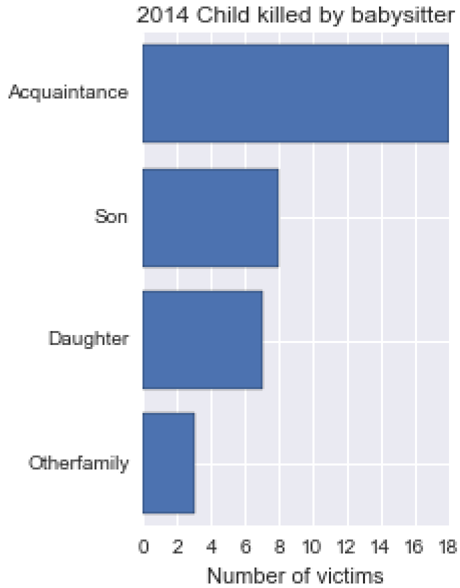

2013 Other than felony type total:
Acquaintance
Stranger
Wife
Girlfriend
0 200 400 600 800 1000 1200 1400 1600 1800
Number of victims


2014 Other than felony type total:
Acquaintance
Stranger
Wife
Girlfriend
0 200 400 600 800 1000 1200 1400 1600
Number of victims

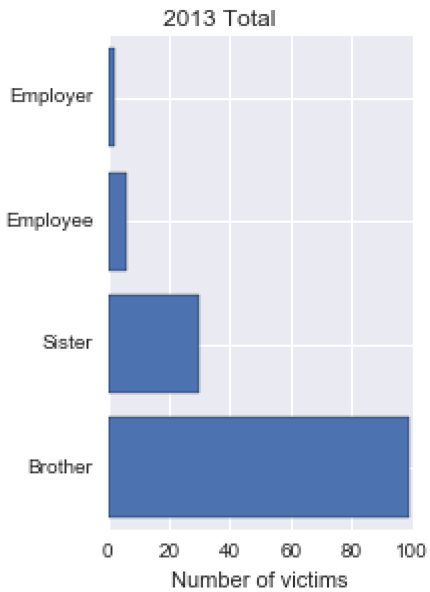


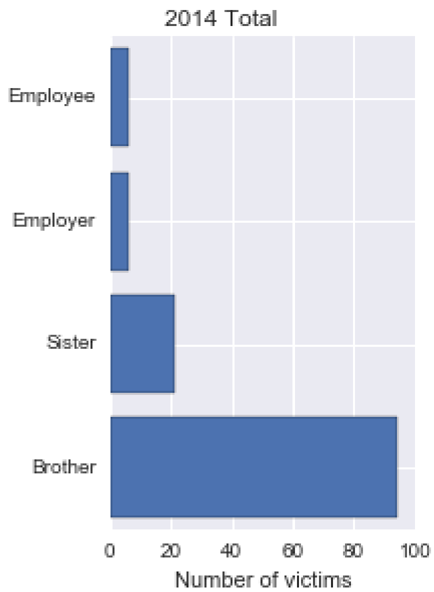


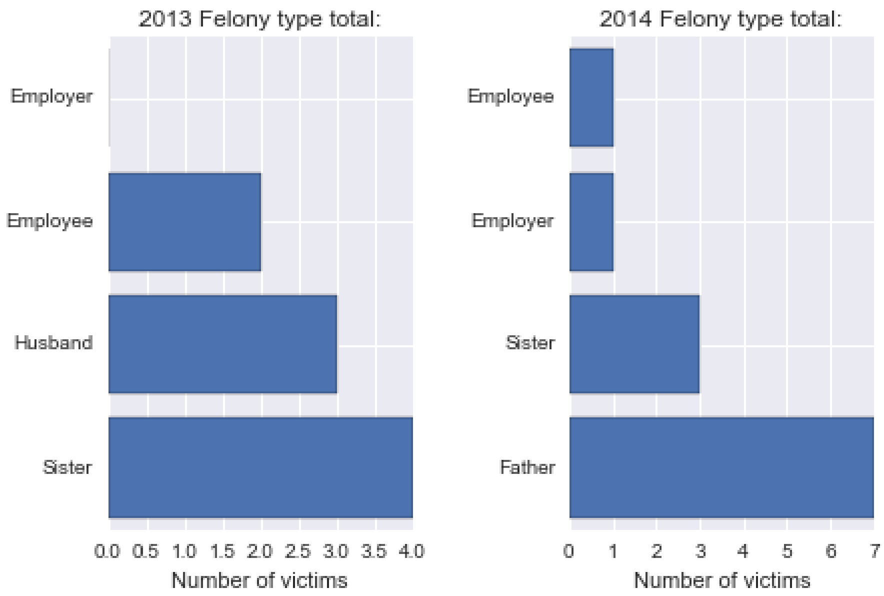





```
Rarest victims of murders in 2013-2014 and crime as a result of which ther
e has been death (non-zero)
```







In [161]:

```
dfNY13 = dfNY13.dropna()
dfNY13 = dfNY13.reset_index(drop=True)
kolumnyNY13 =dfNY13.columns.values[1:]

for col in kolumnyNY13:
    for i in range(len(dfNY13)):
        if type(dfNY13.loc[i, col]) is str:
            dfNY13.loc[i, col] = dfNY13.loc[i, col].replace(" ", "").replace(",", ".")
        dfNY13.loc[i, col] = float(dfNY13.loc[i, col])

    dfNY13[[col]] = dfNY13[[col]].astype(float)
```

In [164]:

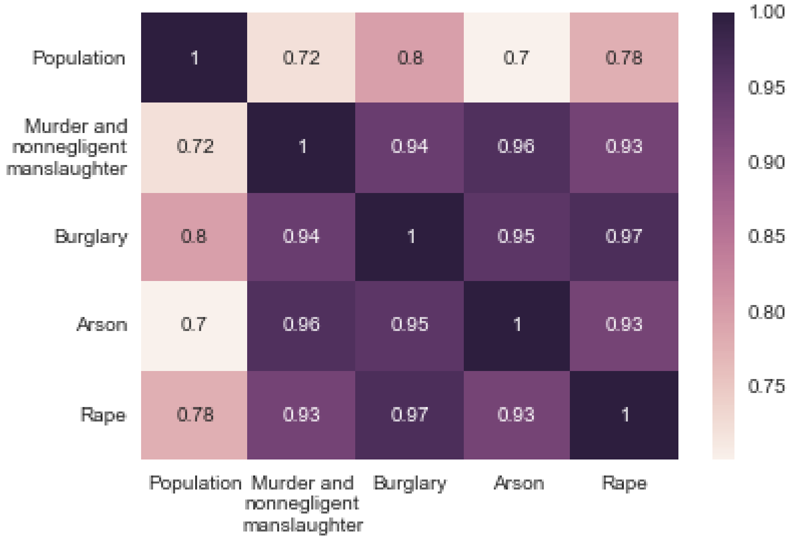
```
dfNY14 = dfNY14.dropna()
dfNY14 = dfNY14.reset_index(drop=True)
kolumnyNY14 =dfNY14.columns.values[1:]

for col in kolumnyNY14:
    for i in range(len(dfNY14)):
        if type(dfNY14.loc[i, col]) is str:
            dfNY14.loc[i, col] = dfNY14.loc[i, col].replace(" ", "").replace(",", ".")
        dfNY14.loc[i, col] = float(dfNY14.loc[i, col])

    dfNY14[[col]] = dfNY14[[col]].astype(float)
```

In [177]:

```
sns.heatmap(dfNY13.loc[:,['Population','Murder and\nnonnegligent\nmanslaughter','Burgla
ry','Arson','Rape']].corr(), annot=True)
```

Out[177]:

```
<matplotlib.axes._subplots.AxesSubplot at 0xd5ca0f0>
```

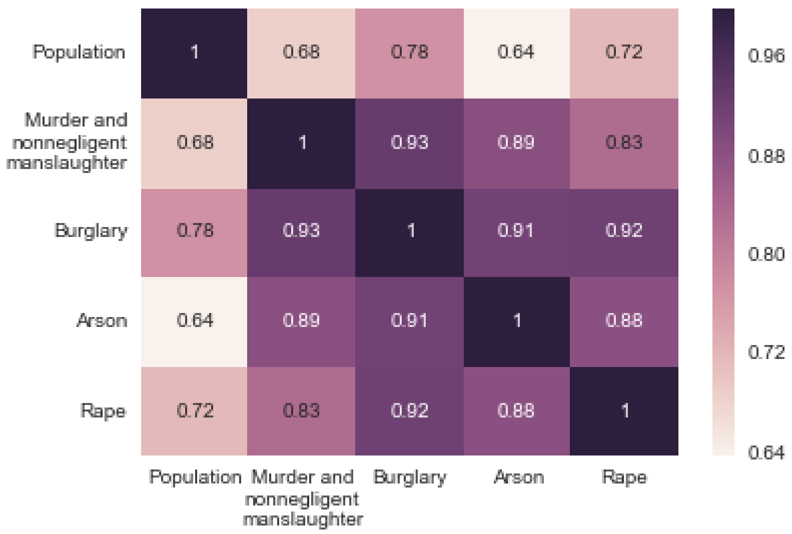


In [178]:

```
sns.heatmap(dfNY14.loc[:,['Population','Murder and\nnonnegligent\nmanslaughter','Burgla
ry','Arson','Rape']].corr(), annot=True)
```

Out[178]:

```
<matplotlib.axes._subplots.AxesSubplot at 0xedff278>
```



In [ ]:

```
#Na podstawie danych z 2013 i 2014 możemy powiedzieć, że najwięcej morderstw jest popeł
nianych na niezbyt bliskich znajomych.
#Okoliczności towarzyszące tym morderstwom mają związek z konkretną osobą (gwałt, prost
ytucja, narkotyki, hazard czy
#niezwrócenie pożyczonego samochodu). Wśród tych pojawia się także podpalenie, co może
 być powodowane bardzo różnymi powodami
#wzniecania pożaru jak np. wyłudzanie odszkodowania.

#Osobno można spojrzeć na trójkąty miłosne i dzieci zabijane przez niańki.
#Zarówno w jednym jak i w drugim przypadku giną znajomi czy dzieci znajomych, a liczba
 ich wcale nie jest tak mała.
#W wypadku dzieci nie giną przypadkowe osoby, a na drugim miejscu, co do częstości (po
 znajomych) są synowie.
#W kwestii miłości pojawiają się już losowe ofiary, ale jeśli ginie ktoś z tego związku
 to częściej jest to dziewczyna czy żona.
#Drugą najliczniejszą grupą ofiar są przypadkowi ludzie, co ma miejsce przede wszystkim
 w trakcie rozbojów, włamań i kradzieży.
#Najrzadziej zabijani są pracownicy i pracodawcy, co dość dobrze świadczy o USA.
#Biorąc pod uwagę wszystkie rodzaje morderstw stosunkowo rzadko morderstwo jest popełni
ane pomiędzy rodzeństwem.

#Na podstawie macierzy korelacji wnioskujemy, że w większe miasta stały się bardziej be
zpieczne w stosunku do mniejszych,
#ponieważ współczynnik korelacji populacji w stosunku do różnych rodzajów przestępstw.
#Widać również, że nie ma dużej zależności między populacją a liczbą podpaleń (współczy
nniki korelacji wynoszą odpowiednio .7 i .64)
```