In [105]:

```
import pandas as pd
import numpy as np
d=pd.read_csv("communities.csv")
d.head()
```

Out[105]:

	8	?	?.1	Lakewoodcity	1	0.19	0.33	0.02	0.9	0.12	...	0.12.2	0.26.1	0.2.
0	53	?	?	Tukwilacity	1	0.00	0.16	0.12	0.74	0.45	...	0.02	0.12	0.4
1	24	?	?	Aberdeentown	1	0.00	0.42	0.49	0.56	0.17	...	0.01	0.21	0.0
2	34	5	81440	Willingborotownship	1	0.04	0.77	1.00	0.08	0.12	...	0.02	0.39	0.2
3	42	95	6096	Bethlehemtownship	1	0.01	0.55	0.02	0.95	0.09	...	0.04	0.09	0.0
4	6	?	?	SouthPasadenacity	1	0.02	0.28	0.06	0.54	1.00	...	0.01	0.58	0.1

5 rows × 128 columns

In [106]:

```
d =d.drop(d.columns[[0,1,2,3,4]],axis=1)
```

In [107]:

```
d.isnull().sum()
```

Out[107]:

```
0.19      0
0.33      0
0.02      0
0.9       0
0.12      0
         ..
0.9.1     0
0.5.2     0
0.32.2    0
0.14.3    0
0.2.2     0
Length: 123, dtype: int64
```

In [108]:

```
from sklearn.impute import SimpleImputer
imp_mean = SimpleImputer(missing_values='?', strategy='constant',fill_valu
d=imp_mean.fit_transform(d)
d=pd.DataFrame(d)
d.head()
#SimpleImputer()
```

Out[108]:

	0	1	2	3	4	5	6	7	8	9	...	113	114	115	116	117
0	0.0	0.16	0.12	0.74	0.45	0.07	0.26	0.59	0.35	0.27	...	0.02	0.12	0.45	0.0	0.0
1	0.0	0.42	0.49	0.56	0.17	0.04	0.39	0.47	0.28	0.32	...	0.01	0.21	0.02	0.0	0.0
2	0.04	0.77	1.0	0.08	0.12	0.1	0.51	0.5	0.34	0.21	...	0.02	0.39	0.28	0.0	0.0
3	0.01	0.55	0.02	0.95	0.09	0.05	0.38	0.38	0.23	0.36	...	0.04	0.09	0.02	0.0	0.0
4	0.02	0.28	0.06	0.54	1.0	0.25	0.31	0.48	0.27	0.37	...	0.01	0.58	0.1	0.0	0.0

5 rows × 123 columns

In [109]:

```
x=d.iloc[:,0:121]
y=d.iloc[:,[122]]
from sklearn.model_selection import KFold
k=KFold(n_splits=10, random_state=1, shuffle=True)
```

In [110]:

```
from sklearn.linear_model import LinearRegression
l = LinearRegression()
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2)
#l=l.fit(x_train,y_train)
from sklearn.metrics import mean_squared_error,mean_absolute_error,median_
l=l.fit(x_train,y_train)
y_pred = l.predict(x_test)
mean = mean_squared_error(y_test,y_pred)
mean_ab = mean_absolute_error(y_test,y_pred)
med_ab = median_absolute_error(y_test,y_pred)
r = r2_score(y_test,y_pred)
print(mean)
print(mean_ab)
print(med_ab)
print(r)
```

```
3.079074724557988e+19
277794515.1474005
0.06312459732556969
-5.7579030509828486e+20
```

In [111]:

```
def fold(a):

  from sklearn.linear_model import LinearRegression
  l = LinearRegression()
  from sklearn.model_selection import KFold
  k = KFold(n_splits=a,shuffle=True)
  print("For Linear Regression with {} -fold cross validation.".format(a))
  for train,test in k.split(d):
    x_train=x.iloc[train,:]
    x_test=x.iloc[test,:]
    y_train=y.iloc[train,:]
    y_test=y.iloc[test,:]
    l = l.fit(x_train,y_train)
    y_pred = l.predict(x_test)
    mean = mean_squared_error(y_test,y_pred)
    mean_ab = mean_absolute_error(y_test,y_pred)
    med_ab = median_absolute_error(y_test,y_pred)
    r = r2_score(y_test,y_pred)
    print(mean)
    print(mean_ab)
    print(med_ab)
    print(r)
    print("\n")
    #print(train)
fold(5)
fold(10)
```

```
For Linear Regression with 5 -fold cross validation.
0.02063702194074034
0.10316326753246803
0.07547537182713612
0.6363191669044796


0.01726782738948942
0.09250683005640462
0.06694822120778166
0.6619545480013757


0.01850096011480582
0.09409298489918133
0.0644512463509497
0.6250970486425462


3.010358650538155e+18
86969622.39272705
0.06669190239885847
-5.832304170655905e+19


0.022812908182225956
0.10436081499539235
0.07075578842723548
0.6297303725810893


For Linear Regression with 10 -fold cross validation.
0.017981819916694543
0.09805083571947618
0.06983403020745162
0.747536378295794


0.0162554858023514
0.0895433039146904
0.059672956986216566
0.6994366329042725


2.608913240909865e+19
361172620.94676787
0.08379073188863087
-4.342919506803166e+20


0.019439350571902443
0.09892424809926295
0.07503747936456485
0.6438090355128242


0.020590070672865394
```

```
0.09812450678633867
0.06108299024579632
0.6185672273187638


0.019649915781526795
0.09669323085055838
0.06731900680481978
0.6105365463732058


0.026637833082055293
0.11095864646752014
0.07884410737032382
0.4317473270179447


0.017058235231306642
0.09274929791223516
0.06646422678316483
0.6696884257049506


0.01743326441107723
0.09490080684000186
0.06141189706245473
0.5859532343052777


0.01694639095499092
0.09801324242645523
0.07924048329252295
0.6995997276415584
```

In [113]:

```
from sklearn import datasets, linear_model
from sklearn.model_selection import cross_val_score
x=d.iloc[:,0:121]
y=d.iloc[:,[122]]
#lasso = linear_model.Lasso()
print(cross_val_score(l,x_test,y_test, cv=5))
print(cross_val_score(l,x_test,y_test, cv=10))
```

```
[ 0.05538798 -0.02693238  0.47118929 -0.76155384 -0.03062139]
[ 0.04377575  0.40196226  0.48214621 -0.16030471  0.4031748   0.50396491
 -2.37093409  0.51986956  0.53394094 -0.38503812]
```

In [ ]:

```
```

In [ ]:

```
```

In [ ]: